STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LIX**, Number 1, 2014

MULTIOBJECTIVE APPROACH OF MULTI-DIMENSIONAL TIME SERIES CLUSTERING

RAMONA STOICA

Abstract. The multidimensional time series are a generalization of the single time series and are more difficult to cluster due to the higher number of parameters used to characterize a data instance. In this work we formulate the multidimensional time series clustering problem as a multiobjective problem and implement several distance measures in the k-means clustering algorithm in order to see the effect of the similarity measure in the clustering process. All the measures are geometrical distances. We used four data sets in order to validate the results. The Euclidean distance which is the most used one does not seem to be the most adequate measure in multidimensional clustering.

1. Introduction

Time series data is a sequence of real numbers that represent the measurements of a real variable at equal time intervals. A data stream is an ordered sequence of points $x_1, \ldots, x_n$. These data can be read or accessed only once or a small number of times. A time series is a sequence of real numbers, each number indicating a value at a time point. Data flows continuously from a data stream at high speed, producing more examples over time in recent real world applications. Most of the time series encountered in cluster analysis are discrete time series. When a variable is defined at all points in time the time series is continuous. Clustering of time series data has applications in an extensive assortment of fields and has attracted a large amount of research [1, 2, 3, 4, 5, 6, 7]. Multidimensional time series are an extension and generalization of regular time series. They have more impact nowadays as most of the data consists of more parameters which are measured over time and decision has to be made considering the behavior of all these parameters together. We

Received by the editors: January 27, 2014.

2010 *Mathematics Subject Classification.* 68P15, 68T05.

1998 *CR Categories and Descriptors.* I.2.6[**Computing Methodologies**]: Artificial Intelligence – *Learning*; I.2.8[**Computing Methodologies**]: Problem Solving, Control Methods, and Search – *Heuristic methods*.

Key words and phrases. Bioinformatics, Dynamic clustering.

propose to investigate in this paper the behavior of k-means algorithm for several multidimensional time series data. We compare versions of k-means for several distance measures. The paper is organized as follows: Section 2 introduces the clustering problems, some similarity distances and some approaches. Section 3 describes the multidimensional time series data, Section 4 presents the k-means for multidimensional time series data clustering and the distance measures that we used, Section 5 contains experiments and comparisons and Section 6 presents the conclusions of this work.

2. Clustering: basic notions

Clustering refears to grouping together data samples that are similar in some way, according to some criteria. It is a form of unsupervised learning because there are no examples showing how the data should be grouped together.

A *cluster* is a collection of data objects that are:

- similar to one another within the same cluster
- dissimilar to the objects in the other clusters.

There are several ways to define similarity and dissimilarity between clusters. These definitions depend on:

- the type of the data considered
- what kind of similarity we are looking for.

Similarity and dissimilarity between objects is often expressed in terms of a distance measure $d(x, y)$. Ideally, every distance measure should be a metric, i.e., it should satisfy the following conditions [8]:

(1) $d(x, y) \geq 0$
(2) $d(x, y) = 0$ *iff* $x = y$
(3) $d(x, y) = d(y, x)$
(4) $d(x, z) \leq d(x, y) + d(y, z)$

2.1. **Similarity and dissimilarity measures.** In this section we briefly review the concepts of similarity and dissimilarity in the context of clustering and we present the similarity measures used later on in the paper.

2.1.1. *Similarity.* The *similarity measure* indicates the strength of the relationship between two data points. The more the two data points resemble one another, the larger the similarity coefficient is [1]. A *metric* is a distance function f that satisfies the following four properties [9]:

(1) non negativity: $f(x, y) \geq 0$
(2) reflexivity: $f(x, y) = 0 \Leftrightarrow x = y$
(3) commutativity: $f(x, y) = f(y, x)$

(4) triangle inequality: $f(x,y) \leq f(x,z) + f(y,z)$

where x, y, and z are arbitrary data points.

Several similarity distances exist. We present some of them which are further used in our implementation.

Euclidean distance. For two data points **x** and **y** in d-dimensional space, the Euclidean distance between them is defined by:

$$d_{euc}(x,y) = [\sum_{j=1}^{d}(x_j - y_j)^2]^{\frac{1}{2}} = [(x-y)(x-y)^T]^{\frac{1}{2}},$$

where x_j and y_j are the values of the j_th attribute of x and y, respectively. The squared Euclidean distance is defined as:

$$d_{euc}(x,y) = d_{euc}(x-y)^2 = \sum_{j=1}^{d}(x_j - y_j)^2 = (x-y)(x-y)^T$$

Manhattan distance. Manhattan distance is defined to be the sum of the distances of all attributes. That is, for two data points **x** and **y** in a d-dimensional space, the Manhattan distance between them is given by:

$$d(x,y) = \sum_{j=1}^{d}|x_j - y_j|$$

Maximum distance. Maximum distance is defined to be the maximum value of the distances of the attributes; that is, for two data points **x** and **y** in d-dimensional space, the maximum distance between them is given by:

$d_{max}(x,y) = max_{1<i<j<d}|x_j - y_j|$

Average distance. The average distance is derived from the Euclidean distance. Given two data points **x** and **y** in a d-dimensional space, the average distance is defined by:

$d_{ave}(x.,y) = (\frac{1}{d}\sum_{j=1}^{d}(x_j - y_j)^2)^{\frac{1}{2}}$

2.2. **Clustering algorithms: k-means.** Clustering algorithms can be divided into two main classes:

(1) hierarchical algorithms: divide the data set into a sequence of partitions
(2) partitioning algorithms: divide the data set into a single partition

In this paper we deal with a variation of the k-mean clustering algorithm.

The k-means algorithm [10] is one of the most used clustering algorithms. It was designed to cluster numerical data in which each cluster has a center called the mean. The k-means algorithm is classified as a partitional or non-hierarchical clustering method [11]. In this algorithm, the number of clusters k is assumed to be fixed.

The algorithm has the following main steps:

(1) Pick a random number k of cluster centers

(2) Assign every item to its nearest cluster center using a similarity or distance measure (e.g. Euclidean distance)
(3) Move each cluster center to the mean of its assigned items
(4) Repeat steps 2 and 3 until change in cluster assignments is less than a threshold

There is an error function in this algorithm which, for given initial k clusters, allocates the remaining data to the nearest clusters and then repeatedly changes the membership of the clusters according to the error function until the error function does not change significantly or the membership of the clusters no longer changes. The k-means algorithm [12, 13, 8] is described below.

K-means algorithm
Require: Data set D, Number of Clusters k, Dimensions d:
{Ci is the i^{th} cluster}
{*1. Initialization Phase*}
1: $(C1, C2,\ldots,Ck)$ = Initial partition of D.
{*2. Iteration Phase*}
2: **repeat**
2.1: d_{ij} = distance between data i and cluster j;
2.2: $n_i = \arg\min_{1\leq j\leq k} d_{ij}$;
2.3: Assign case i to cluster n_i;
2.4: Recompute the cluster means of any changed clusters above;
3: **until** no further changes of cluster membership occur in a complete iteration
4: Output results.

The computational complexity of the algorithm is $O(nkd)$ per iteration [14, 8], where d is the dimension, k is the number of clusters, and n is the number of data points in the data set.

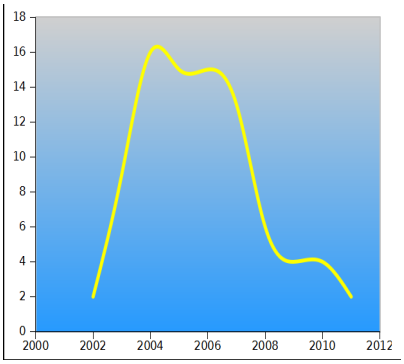
There are some drawbacks with the k-means algorithm:

- result can vary significantly depending on initial choice of seeds (both number and position);
- can get trapped in local minimum – it often terminates at a local optimum;
- to increase the chance of finding the global optimum: restart with different random seeds;
- must pick number of clusters before hand;
- all items are forced into a cluster;
- it is too sensitive to outliers;
- it does not perform well on high dimensional data;
- it only works with numerical data.

3. Time series and multidimensional time series clustering

Time series data is a sequence of real numbers that represent the measurements of a real variable at equal time intervals. Figure 1 shows an example of a time series that has years as unit time intervals.

Figure 1. A time series data example.



A data stream is an ordered sequence of points $x_1 ... x_n$. These data can be read or accessed only once or a small number of times. A time series is a sequence of real numbers, each number indicating a value at a time point. Data flows continuously from a data stream at high speed, producing more examples over time in recent real world applications.

Most of the time series encountered in cluster analysis are discrete time series. When a variable is defined at all points in time the time series is continuous. In general, a time series can be considered as a mixture of the following four components [15, 8]:

(1) a trend (the long-term movement);
(2) fluctuations about the trend of greater or less regularity;
(3) a seasonal component;
(4) a residual or random effect.

Clustering time series is a problem that has applications in an extensive assortment of fields and has recently attracted a large amount of research. Time series data are frequently large and may contain outliers. In addition, time series are a special type of data set where elements have a temporal ordering. Therefore clustering of such data stream is an important issue in the data mining process. Numerous techniques and clustering algorithms have been proposed earlier to assist clustering of time series data streams. The clustering algorithms and their effectiveness on various applications are compared to developing a new method to solve the existing problem.

Clustering of time series data has applications in an extensive assortment of fields and has attracted a large amount of research [16, 15, 17, 18, 19, 20, 21, 22].

3.1. **Multidimensional time series clustering.** A time series is defined as an array X = (x_1, x_2,..., x_n) of measurements in time for a given parameter (or variable).

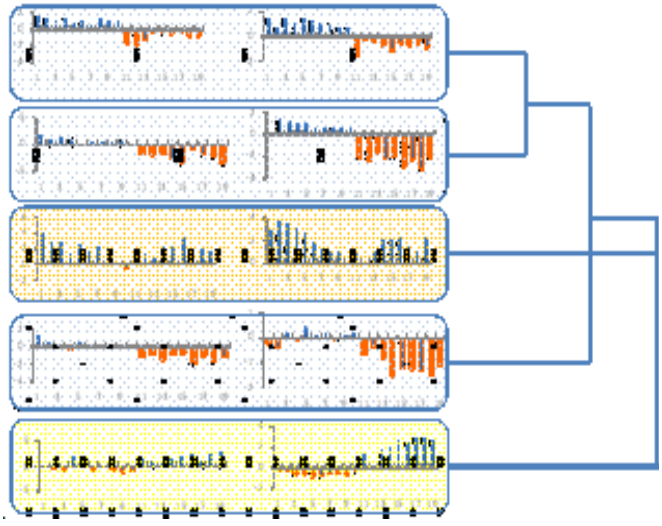
A *multidimensional time series* is defined as:

$$\mathrm{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{pmatrix}$$

where each X_i, $1 \leq \quad i \quad \leq \quad N$ is a time series on its on. The size of these time series can vary.

In the multidimensional case, clustering involves grouping entities of the form X. Figure 2 shows an example of hierarchical clustering for 2-dimensional time series (there are 5 entries or instances that are clustered).

FIGURE 2. Two dimensional time series: example of hierarchical clustering.



Multi-dimensional time series occur if one deals with multiple measurements on some objects, phenomena, or variables.

In time series clustering, each item in the set of items to be clustered is a series of records in time. For instance, the temperatures measured each

day, over the course of 2 years in a certain city are a time series. The same measurements for a number of cities represent the set of the time series which are to be clustered, based on the temperature values recorded in two years.

Each data in this case consists of 730 points (two years of 365 days each) in a two dimensional space. In the multidimensional case, each data consists of more than just one time series. For instance, we want to clusters cities which are not similar only with respect to temperature values over the course of two years, but also the wind speed, pressure, precipitations volume, etc, each of them measured daily. The figures below show some examples of items having two time series each. Some of them may be more similar with respect to one of the time series, while the other will be more similar with respect to the other. In multidimensional clustering we want to cluster together items which are similar with respect to all the time series, regarded in general. This example is illustrated in Figure 3.

Many times, the multidimensional time series data are converted into a single time series by concatenating all the time series into a single one. But this will conduct to loss of generality. The advantage of dealing with a multidimensional time series as such without transforming them is that, on the one hand, it offers a global point of view and shows some critical pathologies arising from evident discrepancies, whereas, on the other hand, it permits to integrate the information contained in each one-dimensional time series of X and therefore it is useful when each array is sparse and short [23].

4. A variant of multidimensional time series data clustering

The similarity between two time series is usually calculated using a distance or a similarity measure. In this section we consider the difference between each time series (of a multidimensional time series instance) as an objective function which has to be minimized. Thus, for comparing how similar two objects X and Y are, where X and Y are given by:

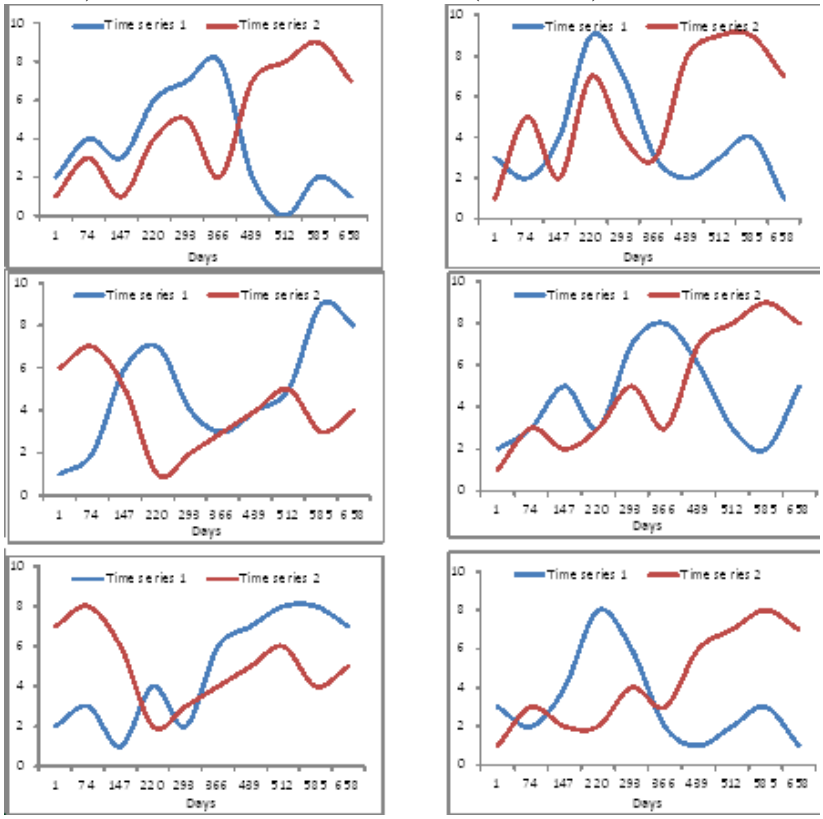
$$\mathrm{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{pmatrix}, \mathrm{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix}$$

we define an N dimensional objective function F= (f_1, f_2, ..., f_N) as:

$$\mathrm{F} = \begin{pmatrix} f_1 = d\,(X_1,\, Y_1) \\ f_2 = d\,(X_2,\, Y_2) \\ \vdots \\ f_N = d\,(X_N,\, Y_N) \end{pmatrix}$$

where d($\cdot$) defines a similarity measure.

FIGURE 3. Example of two sets of measurements (two time series) over the course of two years (730 days).



We use k-means for clustering multidimensional time series data. In our case, each item is assigned to a cluster based on the values of the F function. We consider a weighted combination of all f_i,1≤ i ≤ N as a result of the similarity and denote this by d_{sim}:

$$d_{sim} = \sum_{i=1}^{N} w_i f_i$$

where w is a vector of weights denoting the importance of that particulate time series in the clustering. For our experiments we considered all time series as having equal importance and in this case w_i =1, 1≤ i ≤ N.

We implemented four different distances d(·):

- Euclidean distance
- Manhattan distance

TABLE 1. Weather records

Algorithm	Number of clusters	Silhouette coefficient
k-means with Euclidean distance	8	0.2105
k-means with Manhattan distance	10	0.1574
k-means with Maximum distance	12	0.0200
k-means with Average distance	8	0.2105

- Maximum distance
- Average distance

Setting the value of k. One of four distance measures (Euclidean distance, Manhattan distance, Maximum distance, Average distance) is selected from the main menu, and sent as parameter for the algorithm to use while computing. Also a Maximum Distance Percent can be introduced before running the algorithm; the default value for this variable is 0.6 in our experiments.

The algorithm starts with a large k (equal to the no. of items to cluster) which is decreased step-by-step (by moving data, if convenient, from initial clusters - containing only one item from the data set - to new clusters - containing similar items) until it reaches a value that satisfies the stability of each cluster (small distance between data belonging to same cluster, large distance between data belonging to distinct clusters).

4.1. **Numerical experiments.** We perform experiments considering three datasets from various domains. Silhouette coefficient [2] is used to compare the performance of k-means for various distance measures.

4.2. **Weather records data set.** This data set contains data about countries with respect to temperature, precipitation level, atmospheric pressure and humidity. The countries have to be clustered based on the records over time for all these parameters together.

The details of the data set are:

- 14 (Countries);
- No. of parameters: 5 (Precipitations Level (L/mˆ2), Wind Speed (m/s), Temperature (grC), Atm. Pressure (mmHg), Humidity (%RH));
- No. of time points: 77.

The results obtained by k-means are presented in Table 1.

From the experiments we observe that:

- Best average silhouette coefficient: Euclidean distance and Average distance;

- Better average silhouette coefficient for cluster 0 is obtained using Manhattan distance (0.745) not Euclidean/Average distance (0.181) or Maximum distance (0.240);
- Better average silhouette coefficient for cluster 1 is obtained using Euclidean distance or Average distance (0.674);
- Best average silhouette coefficient obtained for a cluster is 0.828 using Euclidean, Average or Manhattan distance.

4.3. **Sensors recordes data set.** This data set if from the Machine Learning Repository. The file contains 19 activities (like sitting, lying on back and on right side, ascending and descending stairs, running on a treadmill with a speed of 8 km/h, etc). Data is acquired from one of the sensors (T_xacc) of one of the units (T) over a period of 5 sec, for each subject and for each of the activities.

Results obtained by k-means are presented in Table 2. In this case we tested the algorithm with two values for the maximum Distance Percent parameter (used to decide which k (number of clusters) is best): 0.6 and 0.9.

We observed that:

- Best average silhouette coefficient: Manhattan distance using Max Distance Percent 0.9;
- The same average silhouette coefficient for cluster 1 is obtained using Manhattan distance, Euclidean distance or Average distance and the default Max Distance Percent (0.6) or Average distance and a Max Distance Percent = 0.9 (0.521)
- The same average silhouette coefficient for cluster 0 is obtained using Maximum distance and the default Max Distance Percent or Euclidean distance or Average distance and a Max Distance Percent = 0.9 (0.491);
- Best average silhouette coefficient obtained for a cluster is 0.613 using Manhattan distance and Max Distance Percent 0.9;
- For Max Distance Percent lower than default (0.6) worse clustering results have been obtained.

4.4. **KEGG biological data set.** The third dataset if from the KEGG [24]database and is not a time series dataset. We wanted to test the algorithm for this kind of data as well, in order to validate the findings. The data is a Metabolic Relation Network (Directed) Data Set. It has 8 attributes such as: Nodes (min:2, max:116), Edges (min:1, max:606), Connected Components (min:1, max:13), Network Diameter (min:1, max:30), Network Radius (min:1, max:2), Shortest Path (min:1, max:3277), Characteristic Path Length (min:1), Average number of Neighbors (min:1))

TABLE 2. Sensors recordes

Algorithm	Number of clusters	Silhouette coefficient
Max Distance Percent = 0.6		
k-means with Euclidean distance	18	0.028
k-means with Manhattan distance	18	0.028
k-means with Maximum distance	17	0.028
k-means with Average distance	18	0.028
Max Distance Percent = 0.9		
k-means with Euclidean distance	17	0.028
k-means with Manhattan distance	16	0.038
k-means with Maximum distance	17	0.028
k-means with Average distance	18	0.028

The data set has 1,000 instances.
The results obtained by k-means are given in Table 3.

TABLE 3. KEGG data set

Algorithm	Number of clusters	Silhouette coefficient
k-means with Euclidean distance	618	0.0014
k-means with Manhattan distance	618	0.0014
k-means with Maximum distance	618	0.0014
k-means with Average distance	618	0.0014

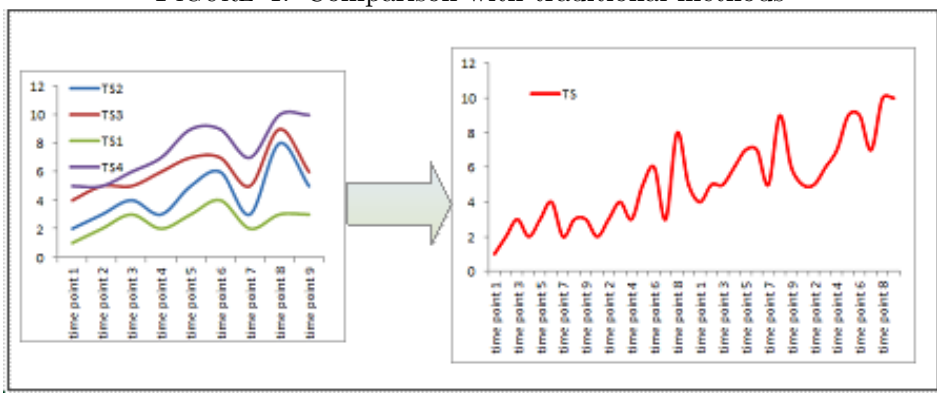
We can observe that:

- The same average silhouette coefficient is obtained for all distance measures (0.0014);

- Using different values for Max Distance Percent (0.2, 0.6, 0.9) hasn't improved the results;
- The best average silhouette coefficient obtained for a cluster is 0.920.

4.5. **Comparison with traditional methods.** In the traditional methods the data is usually pooled, that is a single parameter is inferred for all time series. In this way, a multi-dimensional time series item is transformed into a single dimensional one as it can be seen in Figure 4. We have implemented this approach and tested it using the same settings and under the same conditions as for our approach. For the first data set, for two of the similarity metrics – Manhattan and Average – the number of clusters obtained by the traditional methods was higher that the real one. For the average measure in the second data set the number of clusters obtained by the traditional methods was again higher.

FIGURE 4. Comparison with traditional methods



5. CONCLUSIONS

This paper investigates the role of various distance measures in k-means algorithm for clustering multidimensional time series data. Euclidean distance is the most frequent used and most common measure. Our experiments on – three different data sets – reveal that Manhattan distances (and sometimes the average distance) are better candidates for similarity between two multidimensional time series instances. This work only investigates geometrical distances, but as future work, geometric distances presented here will be compared with other similarity measures (such as descriptive measures, pattern finding measures, etc.).

References

[1] Evertitt B. *Cluster analysis 3rd edition* New York, Toronto: Halsted Press, 1993

[2] Kaufman L., Rousseeuw P. *Finding Groups in Data: An Introduction to Cluster Analysis* Wiley Series in Probability and Mathematical Statistics. New York: John Wiley and Sons, Inc.,1990.

[3] Williams W., Lammbert J. *Multivariate methods in plant ecology: V. similarity analyses and information-analysis* Journal of Ecology, 54(2):427445, 1966.

[4] Duran B., Odell P. *Cluster Analysis: A Survey*, volume 100 of Lecture Notes in Economics and Mathematical Systems Berlin, Heidelberg, New York: Springer-Verlag,1974.

[5] Lance G., Williams W. *A general theory of classificatory sorting strategies I. Hierarchical systems* The Computer Journal , 9(4):373380.

[6] Florek K., Lukaszewicz J., Steinhaus H., Zubrzycki S. *Sur la liaison et la division des points d'un ensemble fini* Colloquium Mathematicum, 2:282285.

[7] Johnson S. *Hierarchical clustering schemes* Psychometrika, 32(3):241254.

[8] Guojun G., Chaoqun M., Jianhong W. *Data Clustering: Theory, Algorithms and Applications* SIAM Publishers, 2007.

[9] Zhang B., Srihari S. *Properties of Binary Vector Dissimilarity Measures* Technical Report, CEDAR, Department of Computer Science and Engineering, University of Buffalo, The State University of New York, 2003. http://www.cedar.buffalo.edu/papers/publications.html.

[10] Macqueen J. *Some methods for classification and analysis of multivariate observations* Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, volume 1, pages 281–297. Berkeley, CA: University of California Press.

[11] Jain A., Dubes R. *Algorithms for Clustering Data* Englewood Cliffs, NJ: PrenticeHall.

[12] Hartigan J., Wong M. *Algorithm AS136: A k-means clustering algorithm* Applied Statistics, 28(1):100–108.

[13] Hartigan J. *Clustering Algorithms* Toronto: JohnWiley & Sons.

[14] Phillips S. *Acceleration of k-means and related clustering algorithms. In Mount, D. and Stein, C., editors,* ALENEX: International workshop on algorithm engineering and experimentation, LNCS, volume 2409, pages 166–177. San Francicsco: Springer-Verlag.

[15] Kendall S., Ord J. *Time Series, 3rd edition* Seven Oaks, U.K.: Edward Arnold.

[16] Tsay R. *Analysis of Financial Time Series* Wiley Series in Probability and Statistics. New York: JohnWiley & Sons.

[17] Shadbolt J., Taylor J. *Neural Networks and the Financial Markets* London: Springer**20**.

[18] Azoff E. *Neural Network Time Series Forecasting of Financial Markets* New York: John-Wiley & Sons.

[19] Gunopulos D., Das G. *Time series similarity measures (tutorial PM-2)* Tutorial notes of the sixth ACM SIGKDD international conference on knowledge discovery and data mining, pages 243–307. Boston, MA: ACM Press .

[20] Bollobás B., Das G., Gunopulos D., and Mannila H. *Time-series similarity problems and well-separated geometric sets* SCG '97: Proceedings of the thirteenth annual symposium on computational geometry, pages 454–456. New York: ACM Press.

[21] Das G., Gunopulos D., Mannila H. *Finding similar time series* Proceedings of the first European symposium on principles of data mining and knowledge discovery, pages 88–100. New York: Springer-Verlag.

[22] Warren Liao T. *Clustering of time series data—a survey* Pattern Recognition, Volume 38, Issue 11, Pages 1857-1874,2005.

[23] Franciosi M., Menconi M. *Multi-dimensional sparse time series: feature extraction* CoRR abs/0803.0405 (2008).
[24] Kanehisa, M. *The KEGG database. In Novartis Found Symp* 2002, January, Vol. 247, pp. 91-101.

Babeş-Bolyai University, Department of Computer Science, 1, M. Kogălniceanu street, 400084 Cluj-Napoca, Romania
E-mail address: ramona@cs.ubbcluj.ro