CLUSTERING METHODOLOGY FOR TIME SERIES MINING

KLASTERIZĀCIJAS METODOLOĢIJA LAIKRINDU IZPĒTĒ

P. Grabusts, A. Borisov

Keywords: time series, similarity measures, k-means clustering, hierarchical clustering, LCSS method

Abstract - A time series is a sequence of real data, representing the measurements of a real variable at time intervals. Time series analysis is a sufficiently well-known task; however, in recent years research has been carried out with the purpose to try to use clustering for the intentions of time series analysis. The main motivation for representing a time series in the form of clusters is to better represent the main characteristics of the data. The central goal of the present research paper was to investigate clustering methodology for time series data mining, to explore the facilities of time series similarity measures and to use them in the analysis of time series clustering results. More complicated similarity measures include Longest Common Subsequence method (LCSS). In this paper, two tasks have been completed. The first task was to define time series similarity measures. It has been established that LCSS method gives better results in the detection of time series similarity than the Euclidean distance. The second task was to explore the facilities of the classical k-means clustering algorithm in time series clustering. As a result of the experiment a conclusion has been drawn that the results of time series clustering with the help of k-means algorithm correspond to the results obtained with LCSS method, thus the clustering results of the specific time series are adequate.

Introduction

Considerable amount of scientific and business data is represented in the form of time series. Recently, the interest in time series analysis has grown. There are two relevant directions in time series research: time series similarity detection and time series clustering. In the first case the importance lies on the detection of different time series similarity measures, in the second case – clustering can be used to group time series according to definite features. Both of these directions supplement each other and allow describing regularities in time series more effectively.

The present research paper is organized in the following way. In the next section of the paper two time series similarity measures will be described: Euclidean distance and Longest Common Subsequence measure. Then, two classical clustering algorithms will be examined that will later on be used in the research activities: k-means clustering and hierarchical clustering. In the section concerning clustering for time series the essence and the most characteristic features of time series clustering will be given. In the last section the analysis of the experimental research carried out for the purposes of the present research paper will be provided – similarity values have been defined to the time series under consideration, k-means clustering algorithm has been adjusted to them and the results obtained have been compared.

Time Series Similarity Measures

A time series is a sequence of real numbers that represent the measurements of a real variable at equal time intervals, whereas a time series database is a collection of time series [1, 2, 3]. Time series data can be analyzed in many different ways. The first step in investigating the time series is often plotting of the samples of time series data under consideration (see Fig.1).

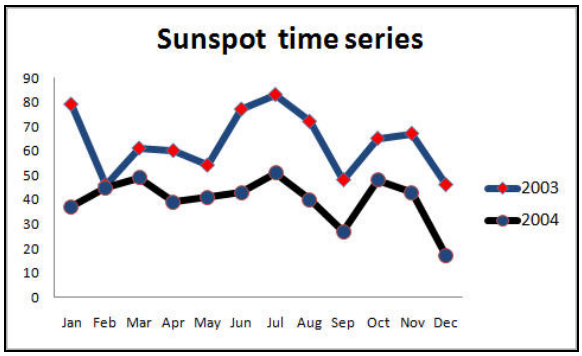


Fig.1. Typical sample of time series

Time sequences appear in many applications, to be more precise, in any applications that involve a value that changes over time. There are several important aspects of mining time series that include trend analysis, similarity search and mining of sequential and periodic patterns in time related data. Recently, a lot of attention has been paid to the problem of similarity retrieval of time sequences in databases, or so called "query by example" [4].

The measure of similarity between objects is an important aspect in many data mining applications. Two time series are considered to be similar if they

have enough non-overlapping time ordered subsequences that are similar. The two subsequences are considered to be similar if one is enclosed within an envelope of a user defined width around the other.

The problem of retrieving similar time sequences may be manifested in the following way: provided there is a sequence q, a set of time sequences X, a distance measure d and a tolerance threshold ε, the task is to find the set of sequences closer to q than ε [5, 6].

Formally, provided there is a pair of time series, the similarity between them is usually measured by their correlation or distance. If a time series is treated as a high dimensional point, the Euclidean distance becomes a natural choice for distance between time series due to the fact that the Euclidean distance is commonly used as a basic similarity measure for time series. According to this model, if the Euclidean distance D between two time sequences Q and S of length n is less than a threshold ε, then the two sequences are said to be similar. Euclidean distance D between two time series $Q=\{q_1,q_2,..., q_n\}$ and $S=\{s_1,s_2,..., s_n\}$ is defined in the following way:

$$D(Q,S)=\sqrt{\sum_{i=1}^{n}(q_i-s_i)^2} \qquad (1)$$

It is problematic to perform similarity search in time series databases: provided there is a query time series and all the time series in the database being similar to the query. Similarity queries in time series databases can be divided into two categories [7]:

- Whole sequence matching. In whole sequence matching, all the time series in the database have the same length n. The length of the query time series q is also n. The Euclidean distance between the query time series and any time series in the database can be calculated in linear time. Provided a query threshold is ε, the result to a whole sequence similarity query search for q are all the time series in the database whose Euclidean distance with q is inferior to the threshold ε.
- Subsequence matching. In subsequence matching, the time series in the database are of different lengths, where the lengths of the candidate time series are usually larger than the length of the query time series. The result to a subsequence query search is any subsequence of any candidate time series whose distance with q is lower than ε.

The most familiar way to deal with the problem of similarity query in time series databases is linear scan in which one calculates the Euclidean distance between the query time series and all the candidate time series in the database. The results of only those time series with the distance less than ε are reported.

Lately, more complicated time series distance measures have been introduced. Some of them are as follows: the Dynamic Time Warping (DTW) measure, the Longest Common Subsequence measure (LCSS) and probabilistic distance measure [4].

DTW recovers optimal alignments between sample points in the two time series. The alignment is optimal in the sense that it minimizes a cumulative distance measure consisting of "local" distances between aligned samples. This procedure is called time warping due to the fact that it warps the time axes of the two time series in such a way that corresponding samples appear at the same location on a common time axis.

Other technique being worthy of attention to describe the similarity is to find the LCSS of two sequences and then to define the distance using the length of this subsequence. The LCSS indicates how well the two sequences can match one another in the case if it is allowed to stretch them without rearranging the sequence of values. Since the values are real numbers, approximate matching rather than exact matching is typically allowed.

The similarity between time series A and B defined as $LCSS(A,B) / max(|A|, |B|)$, where $LCSS[i, j]$ denotes the longest common subsequence between the first i elements of sequence A and the first j elements of sequence B:

$$LCSS[i,j]=\begin{cases}0 & \text{if } i=0 \text{ or } j=0\\ 1+LCSS[i-1,j-1] & \text{if } a_i=b_i\\ \max(LCSS[i-1,j],LCSS[i,j-1]) & \text{otherwise}\end{cases} \qquad (2)$$

A series of experiments has been made by applying the LCSS that showed this method's advantages over the Euclidean measure. Typical synthetic data comprising 100 elements in the time sequence is shown in Figure 2.

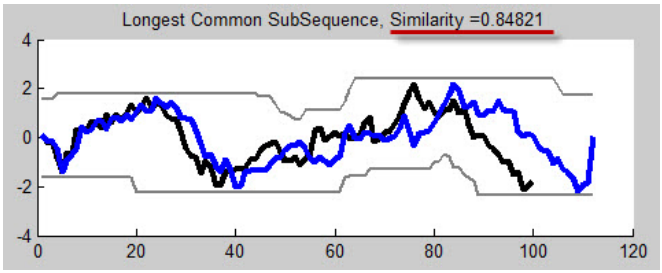


Fig.2. Demonstration of the LCSS method

In the end, $LCSS[n,m]$ will give the length of the longest common subsequence between the two sequences A and B.

Cluster Analysis Method

Cluster analysis is used to automatically generate a list of patterns by a training set [8]. All the objects of this sample are presented to the system without the indication to which pattern they belong. The cluster

analysis is based on the hypothesis of compactness. It means that methods of cluster analysis enable one to divide the objects under investigation into groups of similar objects frequently called clusters or classes. Given a finite set of data *X*, the problem of clustering in *X* is to find several cluster centres that can properly characterize relevant classes of *X*. In classic cluster analysis, these classes are required to form a partition of *X* such that the degree of association is strong for data within blocks of the partition and weak for data in different blocks.

As a data mining function, clustering can be used as a standalone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, and to focus on a particular set of clusters for further analysis. Clustering is one of the most fundamental issues in data recognition. It plays a significant role in searching for structures in data. It may serve as a pre-processing step for other algorithms, which will operate on the detected clusters.

In general, clustering algorithms are used to group some given objects defined by a set of numerical properties in such a way that the objects within a group are more similar than the objects in different groups. Therefore, a particular clustering algorithm needs to be given a criterion to measure the similarity of objects, how to cluster the objects into groups. One of the most widely used k-means clustering algorithms uses the Euclidean distance to measure the similarities between objects. K-means clustering algorithms need to assume that the number of groups (clusters) is known a priori. Table 1 outlines the k-means clustering algorithm [7].

Table 1

An outline of k-means algorithm

K-means clustering
1. Decide on a value for *k*.
2. Initialize the *k* cluster centers (randomly, if necessary).
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.
4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.
5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise go to 3.

Another popular clustering algorithm is called hierarchical clustering. Hierarchical clustering algorithms find successive clusters by using previously established clusters. These algorithms can be either agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters, whereas divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Table 2 outlines the basic hierarchical clustering algorithm [7].

Table 2

An outline of hierarchical clustering algorithm

Hierarchical clustering
1. Calculate the distance between all objects. Store the results in a distance matrix.
2. Search through the distance matrix and find the two most similar clusters/objects.
3. Join the two clusters/objects to produce a cluster that now has at least 2 objects.
4. Update the matrix by calculating the distances between this new cluster and all other clusters.
5. Repeat step 2 until all cases are in one cluster.

Clustering algorithms have shown their best in different data mining tasks, consequently it would be efficient to evaluate their facilities in the analysis of time series.

Clustering for Time Series

Clustering is one of the most frequently used data mining algorithms considered to be useful both in its own right as an exploratory technique, and as a subroutine in more complex data mining algorithms. Conditionally, time series clustering can be divided into two large groups [7]:

- Whole clustering: In this group the notion of clustering is similar to that of conventional clustering of discrete objects. Provided there is a set of individual time series data, the purpose is to classify similar time series into the same cluster.
- Subsequence clustering: Provided there is a single time series, individual time series (subsequences) are extracted with a sliding window. Clustering is then performed on the extracted time series.

Classical clustering algorithms perform whole clustering. For instance, k-means is a heuristic algorithm, and the cluster centers found may not be optimal. It means that the algorithm is guaranteed to converge on a local, but not necessarily on a global optimum. The quality of results may be affected by the choices of the initial centers. One technique that can be applied in order to decrease this problem is to do multiple restarts, and choose the best set of clusters. An obvious question raised here is how much variability in the shapes of cluster centers one gets between multiple runs. The essence of k-means clustering for time series is shown in Figure 3.

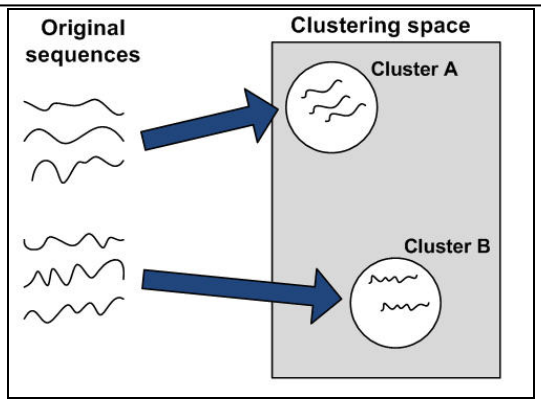


Fig.3. K-means clustering for time series

When the clustering is complete, each cluster center represents a similar group.

Unlike k-means, hierarchical clustering is a deterministic algorithm. Hierarchical clustering can be modified into a partitional clustering by cutting the first k links. The idea of the hierarchical clustering is illustrated in Figure 6. The subsequent time series in each of the k sub-trees can then be unified into single cluster prototypes. When performing hierarchical clustering, one has to make a choice about how to define the distance between two clusters, where this choice is called the linkage method.

The clustering algorithms under consideration have their advantages and disadvantages. The wide application of these methods is restricted by the length of time series, as a result of which clustering algorithms do not provide precise results. This is the reason why in recent years algorithm group, called STS (Subsequence Time Series) clustering has received a wide attention (see Figure 4) [9].

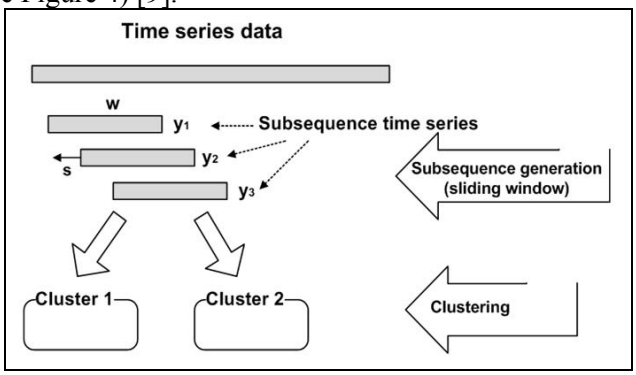


Fig.4. Model of subsequence time series clustering

Subsequence can be defined in the following way: provided there is a time series T of length m, a subsequence C_p of T is a sampling of length $w < m$ of contiguous positions from T, that is, $C=t_p, \dots, t_{p+w-1}$ for $1 \le p \le m-w+1$. Sliding windows can be defined as follows: provided there is a time series T of length m, and a user-defined subsequence length of w, a matrix S of all possible subsequences can be built by "sliding a window" across T and placing subsequence C_p in the p-th row of S. The size of matrix S is $(m - w + 1)$ by w.

Thus, a time series can be modified into a discrete representation by first forming subsequences (using a sliding window) and then clustering these subsequences by using a suitable measure of time series similarity. Subsequence clustering is normally used as a subroutine in many other algorithms, including rule discovery, indexing, classification and prediction.

Experimental Results

The goal of the experiment was to examine the LCSS method suitability in the detection of the real time series similarity. The data has been taken from GCOS-AOPC/OOPC "Working group on surface pressure" climate time series – monthly sunspots numbers, the time coverage of which is 1749 to 2006 [10]. The role of the Global Climate Observing System Working Group on Surface Pressure is to promote the development of long-term high-quality analyses of atmospheric surface pressure. For the experimental research activities the time series by the years 2000-2005 have been chosen, the data for the last years is not complete (see Fig. 5).

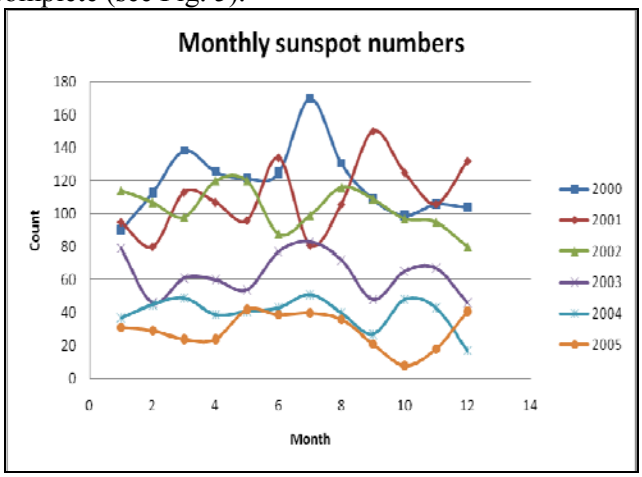


Fig.5. Monthly sunspot numbers

Sunspot time series by the years under consideration can be seen in Table 3 that shows how the time series look in the particular year. The experiments have been carried out to detect time series similarity by applying the LCSS method. The similarity value between two time series has been calculated.

Table 4 displays the results of the application of the LCSS method – time series have been compared in pairs as a result of which the similarity values have been obtained.

Similarity value of the time series with itself equals to 1; in the case when the similarity has not been established, the corresponding cell of the table is left empty. It can be concluded that in this case time series

2000 is slightly similar to 2001 (0.17) and 2002 (0.08). Time series 2003 is slightly similar to 2004 (0.16), but 2004 – 2005 (0.25).

Analyzing data from Table 4, it could be assumed that time series 2000, 2001 and 2002 are located in one cluster, but time series 2003, 2004 and 2005 in another cluster. In the following group of experiments the supposition that clustering could provide analogical results will be verified.

Table 3

Sunspot time series by years

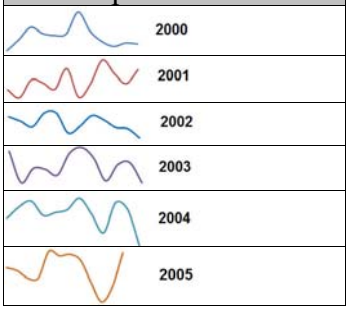
Sunspot time series


Table 4

Similarity values of time series comparison in pairs

	2000	2001	2002	2003	2004	2005
2000	1	0.17	0.08			
2001	0.17	1				
2002	0.08		1			
2003				1	0.16	
2004				0.16	1	0.25
2005					0.25	1

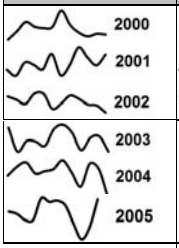
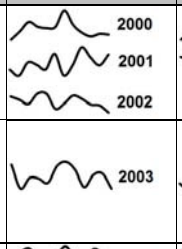
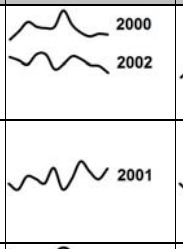
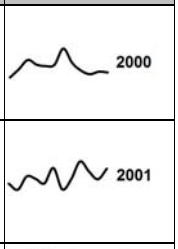
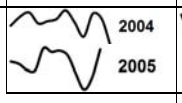
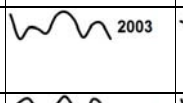
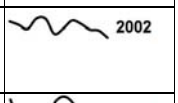
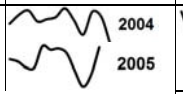
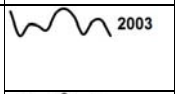
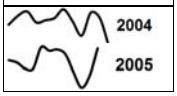
During the second set of experiments in the analysis of time series the k-means clustering algorithm (whole clustering) has been applied. Successively, 2, 3, 4 and 5 clusters have been chosen. The clusters and time series within them obtained as a result of the performance of the algorithm are shown in Table 5.

Analyzing data from the table, it can be concluded that time series in two clusters validate previously mentioned intuitive supposition on the division of the time series. In the case of three, four and fine clusters it can be seen that time series 2004 and 2005 are located inside one cluster that corresponds to Table 4 similarity values.

Thus, it can be reasoned that in this example the results of the time series clustering with the help of the k-means algorithm correspond to the results obtained by using the LCSS method. It gives assurance that the results of time series clustering are adequate that has also been confirmed by the results of the hierarchical clustering (see Figure 6).

Table 5

The number of clusters obtained and time series in clusters

Time series in clusters			
2	3	4	5
			
			
			
			

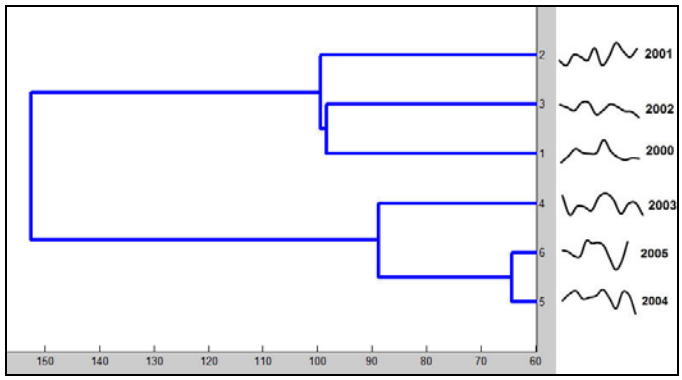


Fig.6. Hierarchical clustering results for sunspot time series

Conclusions and Future Works

A time series is a sequence of real data, representing the measurements of a real variable at time intervals. Time series analysis is a sufficiently well-known task; however, currently research activities are being carried out with the purpose to try to use clustering for the intentions of time series analysis. The main motivation for representing a time series in the form of clusters is to better represent the main characteristics of the data.

In the present research paper two tasks have been completed. The first task was to define time series similarity measures. It has been established that LCSS method gives better results in the detection of time series similarity than the Euclidean distance. The second task was to explore the facilities of the classical k-means clustering algorithm in time series clustering. As a result of the experiment a conclusion has been drawn that the results of time series clustering with the help of k-means algorithm correspond to the results

obtained with LCSS method, thus the clustering results of the specific time series are adequate.

In previous research different clustering algorithms and their application have been analyzed, for instance, the application of clustering method in the Radial Basic Function neural networks or clustering methods in neuro-fuzzy modelling. The main aim of these applications is to extract knowledge from data through rule extraction. The direction of further research activities will be related to rule discovery from clustered time series.

References

1. Kirchgassner G., Wolter J. Introduction to modern time series analysis. – Berlin:Springer, 2007, 274 p.
2. Lutkepohl H. New introduction to multiple time series analysis. – Berlin:Springer, 2005, 764 p.
3. Tsay R.S. Analysis of financial time series. – John Wiley & Sons, 2002, 448 p.
4. Vlachos M., Gunopulos D. Indexing time series under condition of noise. Data mining in time series database: Series in machine perception and artificial intelligence. – World Scientific Publishing, 2004. Vol.57, pp. 67-100.
5. Agrawal R., Faloutsos C., Swami A. Efficient similarity search in sequence databases. Proc. 4th Int. Conf. On Foundations of Data Organizations and Algorithms, 1993. – Chicago. pp. 69-84.
6. Faloutsos C., Ranganathan M., Manolopoulos Y. Fast subsequence matching in time-series databases. Proc. ACM SIGMOID Int. Conf. on Management of Data, 1994. – Minneapolis. pp. 419 – 429.
7. Keogh E., Lin J., Truppel W. Clustering of time series subsequences in meaningless implications for previous and future research. Proc. of the 3rd IEEE International Conference on Data Mining, 2003. – pp. 115 – 122.
8. Xu R., Wunch D.C. Clustering. – John Wiley & Sons, 2009, 358 p.
9. Fujimaki R., Hirose S., Nakata T. Theoretical analysis of subsequence time-series clustering from a frequency-analysis viewpoint. SIAM International Conference on Data Mining, 2008. – Atlanta. pp. 506 – 517.
10. Working Group of Surface Pressure - http://www.cdc.noaa.gov/gcos_wgsp/Timeseries/SUNSPOT/ - Visit date – September, 2009.

Pēteris Grabusts was born in Rezekne, Latvia. He received his Dr.sc.ing. degree in Information Technology from Riga Technical University in 2006. Since 2006 he is Associate Professor in the Department of Computer Science in Rezekne Higher Educational Institution.

His research interests include data mining technologies, neural networks and clustering methods. His current research focuses on techniques for time series clustering and fuzzy clustering.

Arkady Borisov is Professor of Computer Science in the Faculty of Computer Science and Information Technology at Riga Technical University (Latvia). He holds a Doctor of Technical Sciences degree in Control in Technical Systems and the Dr.habil.sci.comp. degree. His research interests include fuzzy sets, fuzzy logic and computational intelligence. He has 205 publications in the area.

Pēteris Grabusts, Arkādijs Borisovs. Klasterizācijas metodoloģija laikrindu izpētē

Vispārīgā gadījumā laikrinda tiek traktēta kā datu virkne noteiktā laika intervālā. Laikrindu analīze ir pietiekami labi pazīstams uzdevums, taču pēdējā laikā tiek veikti pētījumi ar nolūku mēģināt pielietot klasterizāciju laikrindu analīzē – laikrindu datu sadalīšanu līdzīgās grupās. Laikrindu attēlošanas klasteru formā galvenā motivācija ir laikrindu raksturlielumu labāka izpratne. Darba galvenais mērķis bija izpētīt laikrindu klasterizācijas izmantošanas metodoloģiju, apzināt laikrindu līdzības novērtējumus un izmantot tos laikrindu klasterizācijas rezultātu novērtēšanā. Par līdzības novērtējumu tradicionāli tiek izmantots Eiklīda attālums, taču pēdējā laikā līdzības mēra noteikšanai izmanto vairāk komplicētas metodes, piemēram, garākās kopīgās virknes metodi (LCSS). Dotajā darbā tika veikti divi uzdevumi. Pirmais bija noteikt laikrindu līdzības novērtējuma iespējas. Eksperimenta mērķis bija pārbaudīt LCSS metodes piemērotību reālu laikrindu līdzības novērtēšanai. Tika konstatēts, ka LCSS metode dod labākus rezultātus laikrindu līdzības noteikšanā nekā Eiklīda distance. Otrais uzdevums bija izpētīt klasterizācijas algoritmu iespējas laikrindu klasterizācijā. Tika izmantoti divi klasiskie klasterizācijas algoritmi: k-vidējo un hierarhiskās klasterizācijas algoritms. Eksperimentu rezultātā tika izdarīts secinājums, ka konkrēto laikrindu klasterizācijas rezultāti ar k-vidējo algoritma palīdzību atbilst iegūtajiem rezultātiem ar LCSS metodi. Tas deva pārliecību, ka laikrindu klasterizācijas rezultāti ir adekvāti.

Петерис Грабуст, Аркадий Борисов. Методология использования кластеризации в исследовании временных рядов

Формально временной ряд это последовательность данных во временном интервале. Анализ временных рядов достаточно хорошо известная задача, но в последнее время производятся исследования с целью использовать кластеризацию для анализа временных рядов. Главная мотивация для отображения временных рядов в кластерном виде это лучшее представление основных характеристик временных рядов. Целью данной работы является исследование методологии применения кластеризации временных рядов, определение возможных способов оценки сходства или близости временных рядов и применение этих оценок для анализа результатов кластеризации. В качестве меры сходства традиционно используется Эвклидово расстояние, но существуют и более специфические методы, в частности, метод наиболее длинной общей последовательности (LCSS). Были произведены исследования определения возможностей оценки меры сходства временных рядов. Для этой цели применялся метод LCSS, и в экспериментах этот метод показал лучшие результаты, чем Эвклидово расстояние. На следующем этапе экспериментов производилась кластеризация временных рядов. В качестве алгоритмов кластеризации использовались классические алгоритмы к-средних и иерархической кластеризации. По итогам экспериментов сделан вывод, что результаты кластеризации конкретного набора временных рядов соответствуют результатам, полученным с помощью метода LCSS, то есть результаты адекватны.