Folia Oeconomica
Acta Universitatis Lodziensis
ISSN 0208-6018 e-ISSN 2353-7663
www.czasopisma.uni.lodz.pl/foe/
2(341) 2019

DOI: http://dx.doi.org/10.18778/0208-6018.341.03

Jerzy Korzeniewski
University of Łódź, Faculty of Economics and Sociology, Department of Statistical Methods
jurkor@wp.pl

Abridged Symbolic Representation of Time Series for Clustering

Abstract: In recent years a couple of methods aimed at time series symbolic representation have been introduced or developed. This activity is mainly justified by practical considerations such memory savings or fast data base searching. However, some results suggest that in the subject of time series clustering symbolic representation can even upgrade the results of clustering. The article contains a proposal of a new algorithm directed at the task of time series abridged symbolic representation with the emphasis on efficient time series clustering. The idea of the proposal is based on the PAA (piecewise aggregate approximation) technique followed by segmentwise correlation analysis. The primary goal of the article is to upgrade the quality of the PAA technique with respect to possible time series clustering (its speed and quality). We also tried to answer the following questions. Is the task of time series clustering in their original form reasonable? How much memory can we save using the new algorithm? The efficiency of the new algorithm was investigated on empirical time series data sets. The results prove that the new proposal is quite effective with a very limited amount of parametric user interference needed.

Keywords: clustering, time series, symbolic representation, data mining

JEL: C22

1. The research problem

Time series data occur frequently in business and economics. It may be of interest to analyse time series with respect to their classification and discovering patterns or similarities. Clustering algorithms generally do not work directly with original data. Usually time series data are pre-processed before clustering. There are many methods used to transform original data. Among them we should mention principle component analysis (Gavrilov et al., 2000), piecewise aggregate approximation (PAA) (Yeh, Dai, Chen, 2007), discrete Fourier transformation (Agrawal, Faloutsos, Swami, 1993), discrete wavelet transformation (Struzik, Siebes, 1999; Yin, Gaber, 2008), clipping (Bagnall, Janacek, 2005). Some techniques of time series periodisation were also proposed by Grabiński (1992). They are based on the idea of finding some potential thresholds (e.g.: by clustering), and then testing them by means of comparing the parameters (usually expected values). The transformed data are inputs to clustering algorithms. The general belief is that the transformations of original data usually improve the efficiency by reducing data dimensionality and stressing typical features. An important component of clustering is the distance measure used. The most popular measures are: the Euclidean distance, Pearson's correlation coefficient and the short time series distance (Möller-Levet et al., 2003). One can choose from a wide variety of methods and algorithms, however, not many of them are designed with a particular emphasis on prospective clustering of time series data. The problem lies in the question of the possibility of upgrading the PAA technique of symbolic time series representation with a view to further improving the efficiency of time series clustering.

2. New algorithm proposal

In Figure 1, the idea of the PAA approach is presented. It consists in segmenting the whole time series into a predetermined number of equal length parts and assigning 1 if this is an upward movement in the two average values of the two adjacent parts or 0 otherwise. In the effect, we get an extremely shortened notation of the whole time series consisting of an array of ones and zeros. This approach is very popular in the community of time series statisticians, however, we believe, its basic form can be upgraded for two reasons.

There is no particular consideration for the context of possible clustering of the resulting shortened time series notation. Another reason is, probably, the user determined value of the number of segments. We propose the number of segments to be determined in the algorithm and the selection of some of the ones and zeros, those which are more prominent and valuable for clustering, thus shortening the notation as well. To this end, we will use the idea of the distance based cor-

relation coefficient proposed by Korzeniewski (2012). This coefficient effectively captures the clustering notion and has been used successfully many times in traditional stationary cluster analysis. Formally, the distance based correlation coefficient (*DBCorr*) between two sets *A*, *B* of variables is given by the formula:

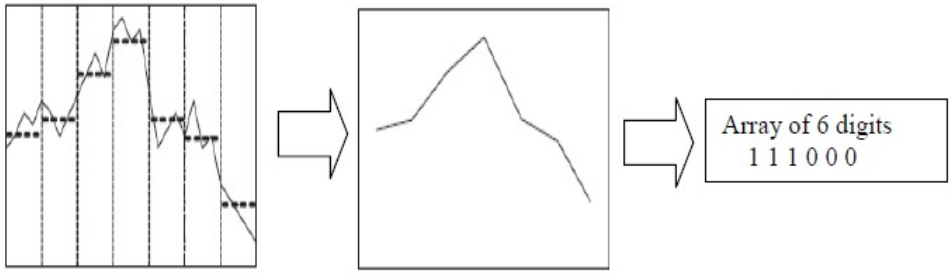

Figure 1. The idea of the PAA technique. The first three ones symbolise three rises and the last three zeros represent three falls

Source: own work on the basis of Fu, 2011

$$DBCorr\left(A,B,l\right)=\frac{\frac{1}{l}\sum_{i=1}^{l}d_i^A d_i^B-\overline{d}^A\overline{d}^B}{s^A s^B}, \tag{1}$$

where $1 \le l \le n$ denotes the number of observation pairs drawn without replacement from all pairs of observations; d_i^A, d_i^B denote distances for the *i*-th pair of objects based on the variables from sets *A, B,* respectively; $\overline{d}^A$, $\overline{d}^B$, s^A, s^B denote arithmetic means and standard deviations computed from all *l* distances on both sets of variables, respectively. This kind of correlation measure is extremely useful when applied in stationary cluster analysis (Korzeniewski, 2012) because if there is a cluster structure and both sets of variables *A* and *B* participate in creating it, then any substantial changes in distances between objects in set *A* should cause changes of distances in set *B*. To fix all technicalities, let us establish that we will apply formula (1) only to sets *A* and *B* consisting of single variables and *l* = 20 with the value of *DBCorr* (*u*, *v*) (*l* is skipped) being the arithmetic mean from 100 repetitions when the number of the data set items is bigger than 30. If the number of data set items is smaller than 30, the number *l* is constituted by the number of all pairs of objects one can create from the whole data set. This coefficient should capture efficiently those pairs of variables which add more to the creation of the cluster structure than others because if both variables contribute something to such a structure, possible transitions from one cluster to another are always connected with "jumps" of distances on both variables. In our context, the variables are binary variables corresponding to consecutive segments (except for the first segment) and one should not expect the Sokal-Michener binary distance to be very precise,

however, even small differences of values of the *DBCorr* can be meaningful. Another feature of this coefficient that can prove useful is the fact that we can use the numerical value of this coefficient as a guidance pointing to a reasonable choice of alternative in the context of determining the number of segments for the PAA analysis. To be more precise, we can choose the number of segments corresponding to the greatest value of average *DBCorr* for a specified set of binary variables.

The considerations presented above let us formulate the following algorithm aimed at reducing the time series to an array of ones and zeroes and, subsequently, to extracting from this array half of all binary symbols/variables describing the series best in the clustering context.

1. Run the PAA algorithm for a reasonable number of segments (binary variables), say $d = 12, \ldots, 30$.
2. For each d, find *DBCorr* (A, B) for all possible pairs of binary variables A, B resulting from step 1.
3. For each A, find the sum of *DBCorr* (A, B) for all B and choose half of all binary variables with the highest values of the sum.
4. Choose d corresponding to the highest value of the sum resulting from step 3. Record all the time series from the whole time series data set in the form of the chosen half of all binary variables resulting initially from step 1.

Table 1. General characteristics of the time series data sets investigated

Set	Number of classes	Size of training set	Size of testing set	Time series length d_i	1-NN Euclidean distance
50Words	50	450	455	270	0.369
Adiac	37	390	391	176	0.389
ArrowHead	3	36	175	251	0.200
Beef	5	30	30	470	0.333
BeetleFly	2	20	20	512	0.250
BirdChicken	2	20	20	512	0.450
Car	4	60	60	577	0.267
CBF	3	30	900	128	0.148
Chlorine	3	467	3840	166	0.35
Coffee	2	28	28	286	0.000
Cricket_X	12	390	390	300	0.423
Cricket_Y	12	390	390	300	0.433
Cricket_Z	12	390	390	300	0.413
Haptics	5	155	308	1092	0.630
Yoga	2	300	3000	426	0.170

Source: UCR Time Series Classification Archive web site [accessed: 1.08.2017]

3. Experiment description

We decided to assess the quality of the new proposal by means of an experiment based on empirical data sets. The UCR Time Series Classification Archive was chosen as the source of data sets. From that base, we sampled 20 data sets whose (and some others') general characteristics are presented in Table 1.

Table 2. Results concerning the new algorithm clustering efficiency assessment.

Set	Optimal number k of segments	Average value of *DBCorr*	Similarity to proper clustering based on d_i original variables	Similarity to proper clustering based on k selected segments
Adiac_train	30	0.024	0.919	0.736
Adiac_test	29	0.026	0.927	0.714
ArrowHead_train	14	0.077	0.123	0.257
ArrowHead_test	14	0.058	0.089	0.015
Beef_train	20	0.068 103 –74	0.103	–0.074
Beef_test	14	0.096	0.072	0.067
BeetleFly_train	12	0.046	0.040	–0.011
BirdChicken_train	16	0.103	–0.013	–0.044
Car_train	17	0.043	0.010	0.002
CBF_train	13	0.051	0.000	0.147
Chlorine_train	17	0.152	0.436	0.427
Coffee_train	29	0.009	0.009	–0.028
Cricket_X_train	12	0.024	0.818	0.793
Cricket_Y_train	12	0.024	0.775	0.798
Cricket_Z_train	16	0.023	0.803	0.075
Haptics_train	14	0.045	0.005	0.004
Words_train	29	0.017	0.926	0.932
Words_test	13	0.017	0.922	0.912
Yoga_train	15	0.130	–0.003	0.009

Source: own investigations

We ran our algorithm on those 20 data sets using the the Sokal-Michener distance in formula (1). In order to assess the results, we computed and can now present:

1) the average value of *DBCorr* among all pairs of the better half of all variables;
2) similarity to proper clustering based on d_i original variables;
3) similarity to proper clustering based on k selected segments.

Similarities were measured by means of the adjusted Rand index (see e.g.: Gatnar, Walesiak, 2004), which is a popular and widely accepted measure of sim-

ilarity of two divisions. Clusterings were always made by the PAM algorithm (see e.g.: Gatnar, Walesiak, 2004) with the Sokal-Michener distance as the distance measure. The PAM algorithm has a much better opinion than the classical *k*-means in the statistical community, as it is much better suited to discover clusters of different shapes. The interpretation of the Rand index is simple – the closer to 1 it is, the more similar two divisions are. The interpretation of the *DBCorr* (*A*, *B*) coefficient is very similar – one would want this coefficient to be as close to 1 as possible if one thinks that both variables *A*, *B* contribute to creating the cluster structure. However, in the case of such a weak scale as the binary scale, even the value of 0.05 suggests that at least one of the variables contributes something to creating the cluster structure.

4. Results and conclusions

To start commenting on the results presented in Table 2, let us mention that the arithmetic mean of the numbers of segments eventually established by the algorithm is 17.7. This seems to be a good result answering the question of possible savings on storage memory rather in the affirmative, because 17.7 is not much more than half of the maximum number of segments used in this experiment i.e. 30 and, what is more, the time series are recorded in the form of half of *k* binary variables (in this way clustering results are much faster to obtain).

As far as the question about the reasonability of using original data is concerned, the answer rather confirms the widespread belief that such an approach does not do much good. Out of 20 sets, only 8 (including *Chlorine_train*) were, more or less, properly clustered on original, untransformed data.

The main target of this research seems to be achieved, too. If the algorithm chooses the better half of *k* binary variables, this choice guarantees, more or less, equal quality clustering as the whole set of *di* original variables. There was only one spectacular loss in the case of *Cricket_Z_train* (0.075 against 0.803), however there were two quite spectacular wins *ArrowHead_train* (0.255 against 0.123) and *CBF_train* (0.147 against 0.000). In the rest of the data sets, whenever the clustering on the shortened binary variables form was bad, so was the clustering on all original variables, or if the last was of good quality, so was the clustering on the shortened binary variables form.

References

Agrawal R., Faloutsos C., Swami A. (1993), *Efficient similarity search in sequence databases*, "Lecture Notes in Computer Science", vol. 730, pp. 69–84.

Bagnall A., Janacek G. (2005), *Clustering time series with clipped data*, "Machine Learning", vol. 58(2–3), pp. 151–178.

Fu T. (2011), *A review on time series data mining*, "Engineering Applications of Artificial Intelligence", vol. 24, Issue 1, pp. 164–181.

Gatnar E., Walesiak M. (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych*, Wydawnictwo Akademii Ekonomicznej we Wrocławiu, Wrocław.

Gavrilov M., Anguelov D., Indyk P., Motwani R. (2000), *Mining the stock market: which measure is best*, Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining, Boston, pp. 487–496.

Grabiński T., (1992), *Metody taksonometrii*, Wydawnictwo Akademii Ekonomicznej w Krakowie, Kraków.

Korzeniewski J. (2012), *Metody selekcji zmiennych w analizie skupień. Nowe procedury*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź.

Möller-Levet C.S., Klawonn F., Cho K., Wolkenhauer O. (2003), *Fuzzy clustering of short time-series and unevenly distributed sampling points*, "Lecture Notes in Computer Science", vol. 2811, pp. 330–340.

Struzik Z.R., Siebes A. (1999), *Measuring time series' similarity through large singular features revealed with wavelet transformation*, Proceedings of tenth international workshop on database & expert systems applications, Berlin, pp. 12–22.

Yeh M.Y., Dai B.R., Chen M.S. (2007), *Clustering over multiple evolving streams by events and correlations*, "IEEE Transactions on Knowledge and Data Engineering", vol. 19(10), pp. 1349–1362.

Yin J., Gaber M.M. (2008), *Clustering distributed time series in sensor networks*, Proceedings of the eighth IEEE international conference on data mining, Washington, pp. 678–687.

Skrócona reprezentacja symboliczna szeregów czasowych dla analizy skupień

Streszczenie: W ostatnich latach pojawiły się metody symbolicznego reprezentowania szeregów czasowych. Te badania są zasadniczo motywowane względami praktycznymi, takimi jak oszczędzanie pamięci lub szybkie przeszukiwanie baz danych. Niektóre wyniki w temacie symbolicznego reprezentowania szeregów czasowych sugerują, że zapis skrócony może nawet poprawić wyniki grupowania. Artykuł zawiera propozycję nowego algorytmu ukierunkowanego na zagadnienie skróconej symbolicznej reprezentacji szeregów czasowych, a w szczególności na efektywne grupowanie szeregów. Idea propozycji polega na wykorzystaniu techniki PAA (*piecewise aggregate approximation*) z następną analizą korelacji otrzymanych segmentów szeregu. Podstawowym celem artykułu jest modyfikacja techniki PAA ukierunkowana na możliwość dalszego grupowania szeregów w ich skróconym zapisie. Próbowano również znaleźć odpowiedzi na następujące pytania: „Czy zadanie grupowania szeregów czasowych w ich oryginalnej postaci ma sens?", „Ile pamięci można oszczędzić, stosując nowy algorytm?". Efektywność nowego algorytmu została zbadana na empirycznych zbiorach danych szeregów czasowych. Wyniki pokazują, że nowa propozycja jest dość efektywna przy bardzo nikłym stopniu parametryzacji wymaganym od użytkownika.

Słowa kluczowe: analiza skupień, szereg czasowy, reprezentacja symboliczna, data mining

JEL: C22

Received: 2018-01-21; verified: 2019-02-01. Accepted: 2019-05-06