Clustering Algorithm for Time Series with Similar Shapes

Jungyu Ahn[1] and Ju-Hong Lee[2]
[1] Dept. of Computer Engineering, Inha University
Namgu, Incheon - Korea
[e-mail: ahnjungyu320@gmail.com]
[2] Dept. of Computer Engineering, Inha University
Namgu, Incheon - Korea
[e-mail:juhong@inha.ac.kr]
*Corresponding author: Ju-Hong Lee

*Received September 25, 2017; revised January 2, 2018; accepted January 23, 2018;
published July 31, 2018*

Abstract

Since time series clustering is performed without prior information, it is used for exploratory data analysis. In particular, clusters of time series with similar shapes can be used in various fields, such as business, medicine, finance, and communications. However, existing time series clustering algorithms have a problem in that time series with different shapes are included in the clusters. The reason for such a problem is that the existing algorithms do not consider the limitations on the size of the generated clusters, and use a dimension reduction method in which the information loss is large. In this paper, we propose a method to alleviate the disadvantages of existing methods and to find a better quality of cluster containing similarly shaped time series. In the data preprocessing step, we normalize the time series using z-transformation. Then, we use piecewise aggregate approximation (PAA) to reduce the dimension of the time series. In the clustering step, we use density-based spatial clustering of applications with noise (DBSCAN) to create a precluster. We then use a modified K-means algorithm to refine the preclusters containing differently shaped time series into subclusters containing only similarly shaped time series. In our experiments, our method showed better results than the existing method.

Keywords: Time Series, Clustering, Similar Shape, Modified K-Means

This work was supported by INHA UNIVERSITY Research Grant.

http://doi.org/10.3837/tiis.2018.07.008 **ISSN : 1976-7277**

1. Introduction

A time series is a set of data that is sequentially observed over time. Due to the advancement of information devices, time series data observed in real time in various fields, such as finance, communications, medicine, health, and transportation, are used in each field. Time series data mining includes query by content, anomaly detection, motif discovery, predication, clustering, classification, and segmentation [11]. Since time series clustering is performed without prior information, it is used for exploratory data analysis [14]. Clustering is especially important in time series analysis because it can find hidden patterns by finding clusters of similarly shaped time series [14].

The existing time series clustering algorithms [1,3,12,15,16] have a problem where the generated clusters contain many time series having different shapes. The causes of the problem are as follows. Time series clusters are created in consecutively dense data spaces [3] [12]. Therefore, the size of the cluster can be very small or very large. If the size of the generated clusters is too large, the clusters include not only time series having similar shapes but also time series having different shapes. In general, density-based clustering causes a problem where time series having different shapes are included in the cluster.

Aghabozorgi et al. [16] and Lai et al. [1] used symbolic aggregate approximation (SAX) [7] as a dimension-reduction method for time series data with high-dimensional characteristics. SAX reduces the dimension of the time series via the piecewise aggregate approximation (PAA) method, and then quantizes it to convert the time series data into a string. However, there are problems with SAX quantization. In SAX, since the length of the quantization interval for quantizing the time series data is non-uniform, this causes an error in the distance calculation of the two time series. Also, if the number of quantization intervals is not sufficiently large, there is a problem where the error in the distance calculation becomes larger.

In cases 1, 2, and 3 of **Fig. 1**, although the actual distances between two data are the same, a SAX-based method calculates the distance differently, as 0, 1, and 2. In cases 4 and 5 of **Fig. 1**, although the actual distance between the two data in case 4 is smaller than that in case 5, the SAX-based distance is 1 in case 4, and the SAX-based distance is 0 in case 5.

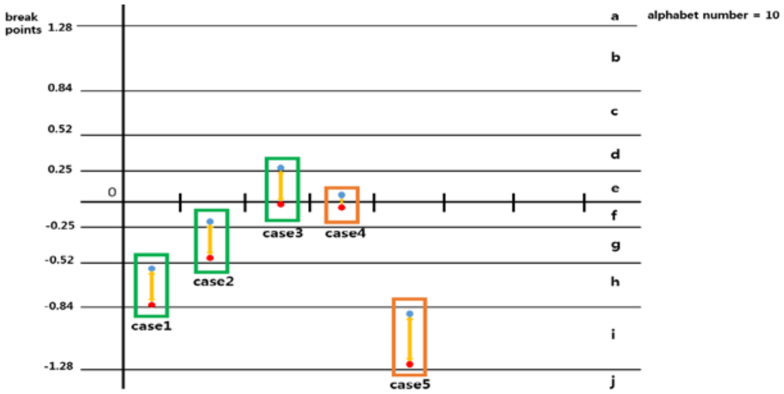

Fig. 1. Error in distance calculation of SAX

This quantization of SAX is a factor in lowering the quality of clusters. Although the quantization of SAX can be advantageous for retrieval purposes, it is not suitable for finding clusters of time series having similar shapes, since it is not sufficiently accurate for measuring distances among time series in the clustering problem [15]. The quantization of SAX causes a problem where time series having similar shapes are included in different clusters, or time series having different shapes are included in the same cluster. Moreover, quantization is not very efficient in terms of computational efficiency. Therefore, quantization of SAX is unnecessary. That is, the PAA step in SAX is enough for dimension reduction of the time series. The PAA reduces the dimension of time series as follows. The PAA divides the time series into frames with a certain length. It replaces the frame with an average of the time series value of the frame. In this method, there is no quantization process. Therefore, when calculating the distance between time series, the distance calculation error does not occur like the distance calculation in SAX.

Aghabozorgi et al. [15,16] used a general partition-based clustering algorithm in the refining process. The partition-based clustering algorithm creates clusters by dividing the data space into K partitions. However, if the data space is not sufficiently divided by the K value selected by the user, there is the problem of the cluster being excessively large. This can cause the quality of clusters to deteriorate, because time series with different shapes are included in the same cluster. For example, if the cluster size generated by the partition-based clustering algorithm is too large, the distance between two time series in the cluster may be very large. This means that the shape between the two time series can be different.

In this paper, we propose a method to alleviate the disadvantages of the existing methods so that similarly shaped time series are included in the cluster. And our method improves the quality of the cluster. The scale of the analyzed time series may be different. This can have a significant impact on the results of the time series analysis. For example, there may be a problem where two time series with large scale differences and similar shapes are not assigned to the same cluster, and two time series with small scale ranges and different shapes are assigned to the same cluster. Therefore, time series should be normalized. Time series are normalized by applying Z-transformation. And we use PAA to reduce the dimension of the time series.

In the preclustering step, preclusters are generated by applying a density-based spatial clustering of applications with noise (DBSCAN) algorithm [9] to the time series data of the reduced dimension. Although DBSCAN has the disadvantage of creating an excessively large time series cluster, we used DBSCAN in the preclustering step because DBSCAN has the following advantages. First, DBSCAN regards consecutive high-density parts as clusters in the data space, and low-density parts as noise, so DBSCAN can distinguish noise efficiently. Second, DBSCAN has an advantage in that it does not need to set the number of clusters in advance. It is very difficult for us to know the number of valid clusters in advance. However, DBSCAN automatically creates clusters according to the distribution of data. Third, since the time complexity of the algorithm is considerably low [11], DBSCAN meets the purpose of preclustering to improve the clustering speed.

In the refinement step, a modified K-means algorithm is used. Since preclusters generated by DBSCAN may contain time series that do not have similar shapes, they are divided into subclusters of appropriate size. The K-means algorithm must input the number of clusters in advance. However, instead of estimating the number of clusters in advance, we estimate the size of the clusters so that only time series having similar shapes are included. Modified

K-means improves the quality of the final cluster by dividing the precluster into clusters of the estimated size.

The main contribution of this paper can be described as follows.

1. The proposed method has a higher similarity in shape among the time series included in the cluster than other existing algorithms.
2. Unlike the conventional time-series clustering algorithms, the clustering method proposed in this paper includes a cluster tuning process. Therefore, our proposed clustering method refines clusters containing time series with different shapes so that the final clusters contain only time series of similar shape.
3. The proposed method solves the fundamental problem of a time-series clustering algorithm that estimates the number of clusters. In general, existing time series clustering algorithms must estimate the number of clusters in the data space. If the number of estimated clusters is not correct, there is a problem that the cluster contains time series data of different shapes. However, the proposed method does not use the number of clusters as an algorithm parameter. Instead, it sets the density of the data in the data space and the cluster size as parameters. By doing so, this method successfully creates clusters that contain similar shaped time series.
4. We analyzed the efficiency of SAX and PAA, which are frequently used for time series analysis among time series dimension reduction methods.

This paper is composed as follows. Section 2 describes related work. Section 3 describes the clustering method proposed in this paper. Section 4 presents the experiments. Section 5 describes conclusions and future work.

2. Related Work

A time series has high-dimensional characteristics. The characteristics can cause a dimensionality problem and a computation time problem in time series analysis [10]. Therefore, dimensional reduction, which transforms high-dimensional time series data into a low-dimensional representation, is essential for time series analysis. The dimension reduction method of a time series can be either a data adaptive method or a non-data adaptive method [13]. The data adaptive method is used when the size of the data is variable. It includes symbolic aggregate approximation [7], and piecewise aggregate approximation [4]. The non-data adaptive method is used when the size of the data is fixed. It includes discrete fourier transform (DFT) [2], discrete cosine transformation (DCT) [5], and discrete wavelet transformation (DWT) [8].

Ding et al. proposed a density-based clustering algorithm (YADING) for large-volume time series data analysis [12]. It reduces the execution time of the algorithm by using the sampling method, and estimates the density radii suitable for the time series data distribution by using inflection points. And it creates clusters using the estimated density radii as a parameter for DBSCAN.

Aghabozorgi et al. proposed a time series data clustering method (ThreePTC) to evaluate co-movement of the stock market [15]. It converts time series data using SAX. A K-mode algorithm is applied to the transformed time series to create preclusters. The preclusters are refined using a PCS algorithm, and subclusters are created. It merges highly similar subclusters into a final cluster. Here, a shape similarity method is used as the similarity calculation method.

Lai et al. proposed a two-level clustering method for time series data analysis [1]. It converts time series data using SAX. And it creates the clusters using a CAST algorithm. Each cluster is divided into subclusters using subsequence information of the time series data in the cluster.

Aghabozorgi et al. proposed a hybrid clustering algorithm for time series clustering [16]. This method is executed as follows. Clusters are generated by applying a 'similar in time' similarity method and the CAST algorithm to the time series data. The generated clusters are merged by applying a k-medoid algorithm and a similar-in-shape similarity method.

Zechao Li et al.[17] proposed a new unsupervised feature selection algorithm by integrating cluster analysis and sparse structure analysis. In particular, Nonnegative Spectral Clustering is used to learn the label of the input sample more accurately and is also responsible for the function of feature selection. Also, for the optimization of the algorithm, the authors of this paper proposed an efficient iterative algorithm.

Zechao Li et al.[18] propose a novel Robust Structured Subspace Learning (RSSL) algorithm by integrating image understanding and feature learning into a joint learning framework. The learned subspace is adopted as an intermediate space to reduce the semantic gap between the low-level visual features and the high-level semantics. To guarantee the subspace to be compact and discriminative, the intrinsic geometric structure of data, and the local and global structural consistencies over labels are exploited simultaneously in the proposed algorithm.

Zechao Li et al.[19] have argued that the proposed method is able to directly identify a distinct subset of the most useful and redundant functions. In particular, the non-negative spectral analysis performed here is developed to learn the exact cluster label of the input image, and feature selection is performed simultaneously.

3. Proposed Method

3.1 Data Preprocessing

Generally, time series have a very different scale for each piece of data. In addition, time series generally have high-dimensional characteristics. Therefore, data preprocessing is necessary to analyze the time series. In this paper, we use a normalization method and a dimension-reduction method to preprocess time series data.

3.1.1 Data Normalization

The unit of measurement for time series data can have a significant impact on the results of time series data analysis. Therefore, it is necessary to express the value of the time series data as a value within a certain range. Normalization means representing the value of the data as a value within a certain range [6]. In this paper, Z-transformation is used as a time series data normalization method. Z-transformation is defined as follows.

Definition. Z-Transformation

$$Y_t = \frac{X_t - \mu_t}{\sigma_X} \tag{1}$$

Where Y_t represents a normalized time series at time t, X_t represents a time series that is not normalized at time t, μ_t represents the average of the X data, and σ_X represents the standard deviation of the X data.

3.1.1 Dimension Reduction

Since time series data have high-dimensional characteristics, the use of a raw time series for time series clustering can cause various problems, such as high similarity computation time, the curse of dimensionality, and long clustering running time. Therefore, in time series clustering, high-dimensional time series data need to be converted into low-dimensional data. In this paper, we chose PAA as the dimension-reduction method instead of SAX, which is used in most existing time series clustering. In general, PAA has advantages of the simple and quick calculation of dimension reduction. The validity of selecting PAA is demonstrated by comparing the loss of information in PAA and SAX in an experiment.

3.2 Clustering

The proposed clustering algorithm consists of two steps: Preclustering and refinement.

3.2.1 Preclustering

Preclustering is creating time series clusters by using a time series of reduced dimension. Since a time series has high-dimensional characteristics, calculation of the distance using the raw time series has a disadvantage in that it takes a long time. Therefore, we reduce the execution time of clustering by using a time series of reduced dimension in the time series distance calculation. A DBSCAN algorithm is used in the preclustering step. DBSCAN finds core objects with dense neighborhoods, and creates clusters by connecting these core objects and neighborhoods.

3.2.2 Refinement

In the preclustering step, we use the low-dimensional data and perform a density-based clustering algorithm on this data. Therefore, clusters generated in the preclustering step may include time series with different shapes. Fig. 2 illustrates the problems that may occur during the preclustering step.

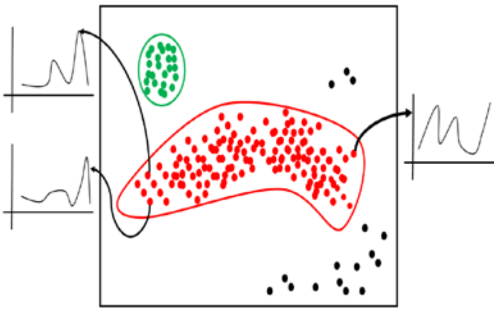

Fig. 2. The problem of density-based clustering

Therefore, it is necessary to refine the clusters generated in the Preclustering step. In this step, the precluster generated in the preclustering step is refined by applying a modified K-means algorithm. The modified K-means algorithm uses the normalized data of the original data instead of the time series data of the reduced dimension. The modified K-means algorithm does not pre-determine the number of clusters, as in K-means or partition-clustering algorithms. It uses the Maximum Standard Deviation (MSD) value of the cluster to limit its size. MSD is defined as follows.

Definition. Maximum Standard Deviation(MSD)

$$MSD = \max_{i \in cluster} \sqrt{\frac{1}{n}\sum_{t=1}^{n}(f_i(t) - p(t))^2} \tag{2}$$

Where p(t) represents the prototype time series of the cluster, $f_i(t)$ represents the i-th time series datum of the data.

Definition. Prototype Time Series

$$p(t) = \frac{1}{|cluster|}\sum_{i \in cluster} f_i(t) \tag{3}$$

MSD is the maximum distance between the cluster's internal time series and the cluster's prototype time series. A large value of MSD means that the size of the cluster is large, and a small value of MSD means that the size of the cluster is small. Therefore, we set a specific value of MSD as a cluster size parameter. The modified K-means algorithm uses the experimentally determined MSD value as an MSD threshold for the algorithm, and is performed as follows. K-means (k=2) is performed to divide the data space into two clusters. If the MSD of the cluster generated in the previous step is larger than the MSD threshold, K-means (k=2) is performed on the generated cluster, and the cluster is divided again. The above cluster partitioning is repeated until the MSD value of the cluster becomes smaller than the MSD threshold. The modified K-means algorithm is described in **Fig. 3**.

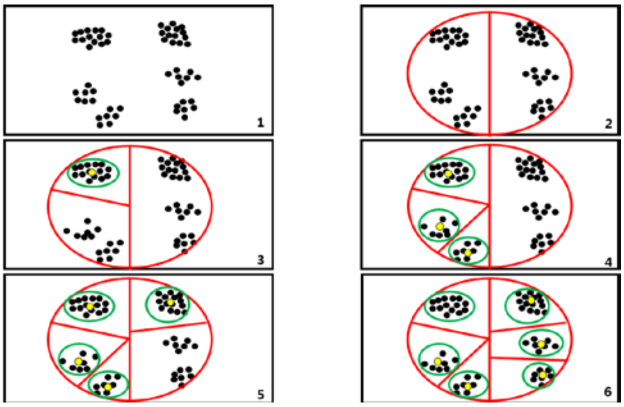

Fig. 3. Modified K-means steps. The red dot represents the prototype of the generated cluster. The green circle represents a cluster for which the MSD value is less than the MSD threshold.

Fig. 4 shows the estimation method of the MSD threshold used in modified K-means. We sequentially include N neighbors from the randomly selected reference time series data in the neighborhood set. We then use the graph of the time series data contained in the neighbor set sequentially to find the appropriate MSD threshold. A time series with significantly less similarity of shapes in neighbor sets exceeding a certain MSD value is included. Then, we remove the time series from the neighbor set and set the MSD value in the neighbor set to the MSD threshold.

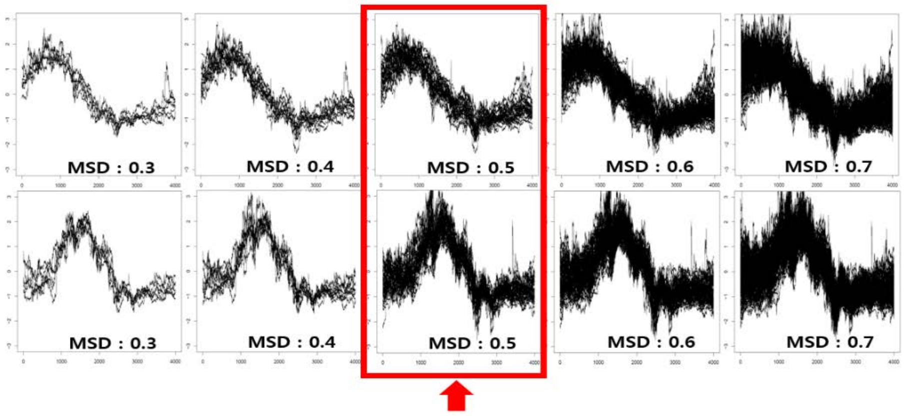

Fig. 4. Estimating MSD threshold, 0.45

The following algorithm tables represent the clustering method and modified K-means algorithm proposed in this paper.

Algorithm 1. Main Algorithm

```
Input: Time series Data(S)
Output: clusters

NDS ← Z-Transformation(S)
$\sigma$ ← estimateMSDthreshold(NDS)
RDS ← PAA(NDS)
preclusters ← DBSCAN(RDS)
refinedclusters ← Ø
for P ∈ preclusters do
    refinedclusters ← refinedclusters ∪ Modified_K-Means(P in NDS, $\sigma$)
end for
return refinedclusters
```

Algorithm 2. Modified_K-Means(S, σ)

```
Input: Normalized Time series Data(S), $\sigma$
Output: clusters
$\rho$ ← calculateMSD(S)
if $\rho < \sigma$ then return S
{$C_1, C_2$} ← K-means(S,2)
result ← Ø
for each $C$ ∈ {$C_1, C_2$}
    result ← result ∪ Modified_K-Means($C$, $\rho$)
end for
return result
```

4. Experimental Classification Results and Analysis

In this paper, we performed the following experiments:

- An experiment on the efficiency of dimension reduction
- An experiment to compare our method with other algorithms

4.1 Experiment Setup

We used stock data and the University of California, Riverside (UCR) Time Series as experimental data. The stock data consist of 1200 stocks listed in 2015, and each stock has 4000 time dimensions. The experiment was performed using an Intel i5 microprocessor with 4GB of RAM.

4.2 Efficiency of Dimension Reduction

In this paper, PAA and SAX are compared to select the dimension-reduction method for the time series data. Each dimension-reduction method was evaluated using Dimension Reduction Efficiency (DRE). DRE is defined based on the amount of information loss as follows.

Definition. Dimension Reduction Efficiency

$$DRE = \frac{SSEO}{SSER} x100(\%) \tag{4}$$

Definition. Sum of Square Error for Original Space

$$SSEO = \sum_{k \in SetO} \sum_{t=1}^{n} (f_k(t) - \mu(t))^2 \tag{5}$$

Where μ represents a reference time series arbitrarily selected from the original data space, SetO is a set of n time series nearest to the reference time series, and f_k is the nearest k-th time series in the original data space from the reference time series.

Definition. Sum of Square Error for Reduced Space

$$SSEO = \sum_{k \in SetR} \sum_{t=1}^{n} (f_k(t) - \mu(t))^2 \tag{5}$$

Where SetR is a set of n time series nearest to the reference time series in the reduced space, f_k is a representation in the original space of the k-th nearest time series from the reference time series in the reduced space.

DRE has the following meanings. Sum of Square Error for Reduced Space (SSER) is an SSE value obtained by using k data close to the reference data in the reduced space. Therefore, SSER is always larger than Sum of Square Error for Original Space (SSEO) obtained by using k data close to the reference data in the original space. The dimension-reduction method with the least loss of information has a larger DRE value than the dimension-reduction method with a large loss of information. **Fig. 5** shows the result of DRE for each dimension-reduction method.

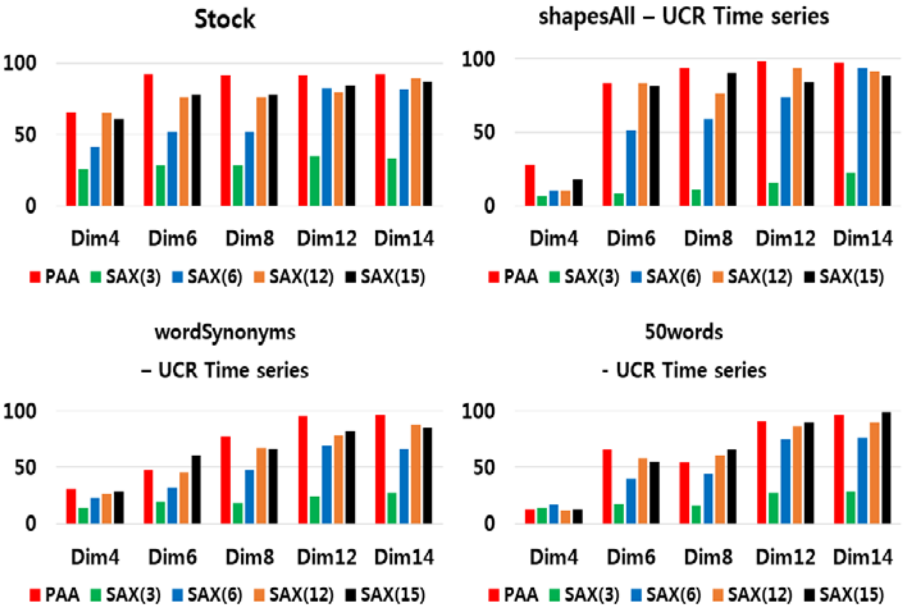

Fig. 5. DRE results for each dimension-reduction method. For SAX, the numbers in parentheses indicate the number of alphabets

We confirmed that the DRE value of PAA is higher than that of SAX. This has the following meaning. Because SAX performs dimension reduction in two steps, the loss of information of the SAX method is greater than that of PAA. This causes the SSER value of SAX to be larger than that of PAA in the reduced dimension space. And this causes the DRE value of SAX to be smaller than that of PAA. Therefore, in this paper, we chose PAA as the dimension reduction method.

4.3 Comparing our Method with Other Algorithms

We compared the proposed method with the K-means algorithm [13], YADING [12], and ThreePTC [15]. Root Mean Square Error (RMSE) and maximum standard deviation were used to evaluate clusters generated by the clustering algorithm. RMSE is defined as follows.

Definition. Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{|Cluster|}\sum_{i \in cluster}\sum_{t=1}^{n}(f_i(t)-\mu(t))^2} \tag{3}$$

Where |Cluster| represents the number of time series in the cluster, f_k represents the i-th time series in the cluster, and μ represents the prototype time series of the cluster.

RMSE was used to evaluate the coherence of the time series within the cluster. However, RMSE averages the SSE value of the cluster. Therefore, even if the generated clusters have similar RMSE values, the deviation of the time series from the reference time series in the cluster may be greatly different. In order to consider the degree of deviation of the time series, we use RMSE and MSD together to evaluate the similarity of the shape of the time series within the cluster. The following figures show the RMSE and MSD values of the clusters generated by our method and the compared algorithms.

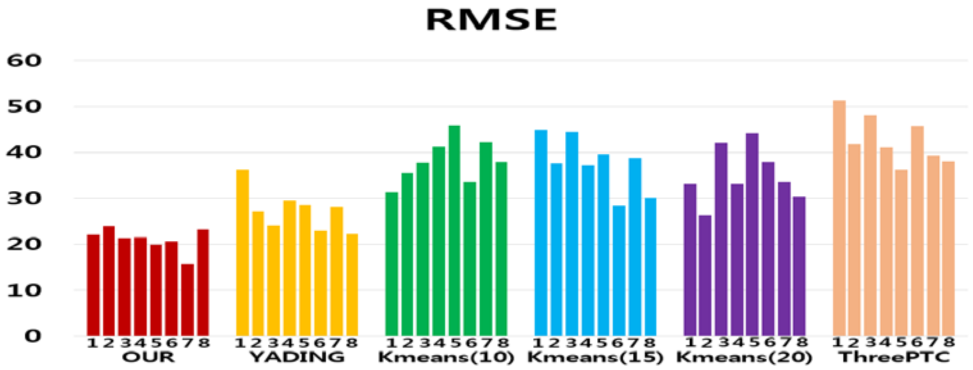

Fig. 6. RMSE values of each clustering algorithm

Fig. 6 shows that our method has a smaller RMSE than other algorithms. In terms of distance from the prototype time series, the time series inside the cluster generated by our method has a smaller value than that of other algorithms. Therefore, we can confirm that the clusters generated by our method have better cohesion.

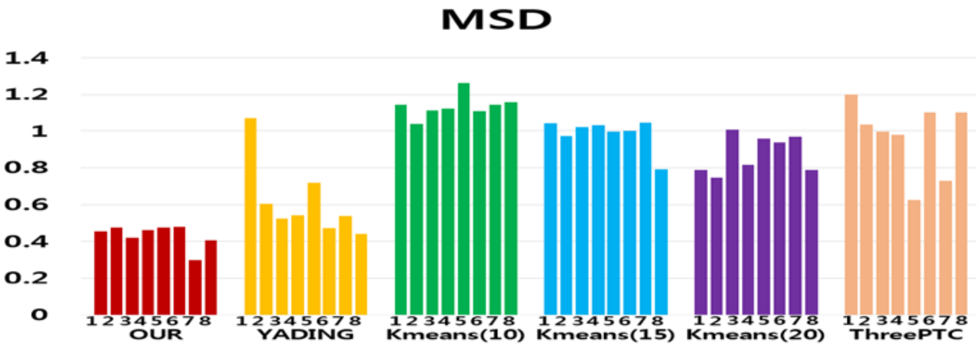

Fig. 7. MSD values of each clustering algorithm

In **Fig. 7**, we can see that our method has a smaller MSD value than other algorithms. Clusters generated by our method have a smaller value than the other algorithms in the degree of deviation from the prototype time series.

As shown in **Fig. 6** and **Fig. 7** above, our method has smaller RMSE and MSD than the compared algorithms. This means that clusters generated by our method are more coherent than clusters generated by other algorithms. And this means that the time series within the cluster generated by our method do not deviate much from the prototype time series of the cluster. Therefore, we show that the time series within the cluster generated by our method are similar in shape to each other.

The following figures show clusters generated by each algorithm.

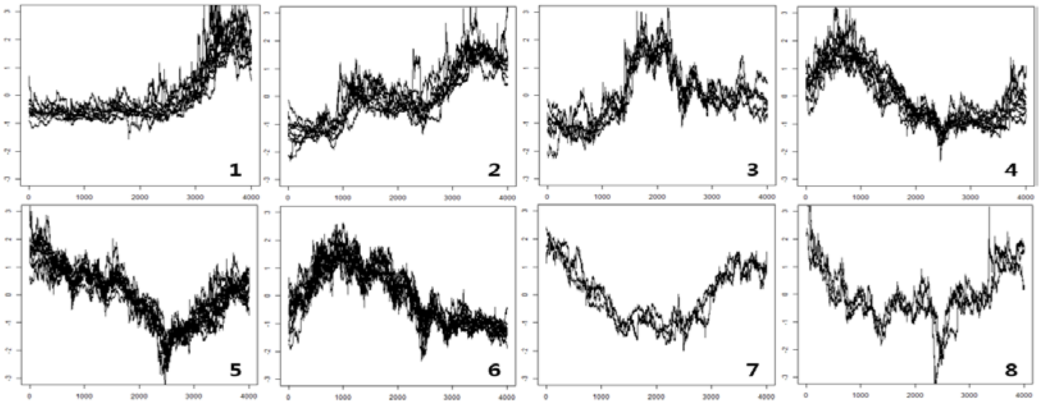

Fig. 8. Clusters generated by our method

In our previous experiment, our method has smaller RMSE and MSD than other clustering algorithms. As shown in **Fig. 8**, the time series within the cluster generated by our method have large coherence and a small deviation from the prototype time series.

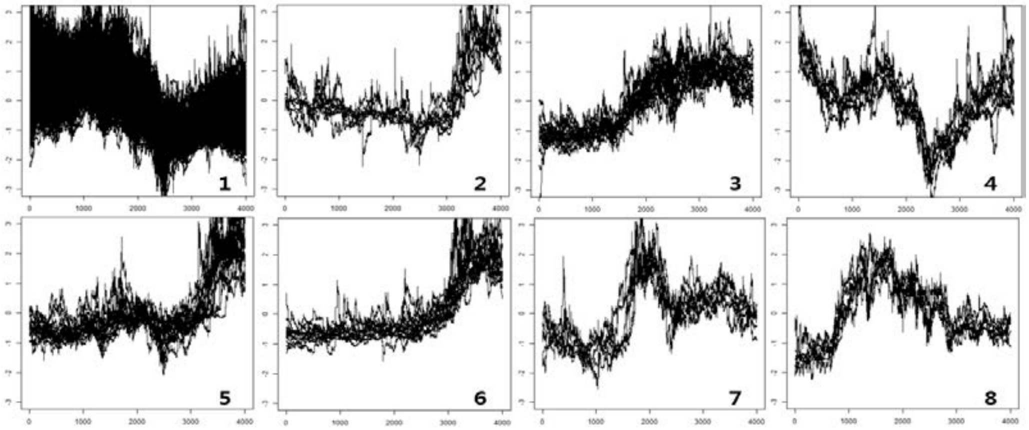

Fig. 9. Clusters generated by YADING

Fig. 9 shows the clusters generated by YADING. Because YADING uses a density-based clustering method, it often creates an excessively large cluster. Therefore, we can see that time series with different shapes are included in the cluster generated by YADING. Because the clusters generated by YADING often have too large a size, they have high MSD, RMSE in previous experiments.

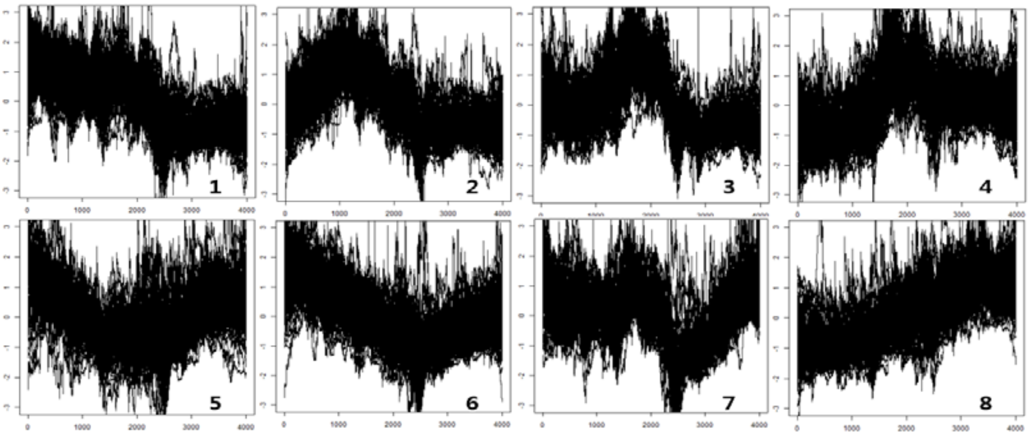

Fig. 10. Clusters generated by K-means (10)

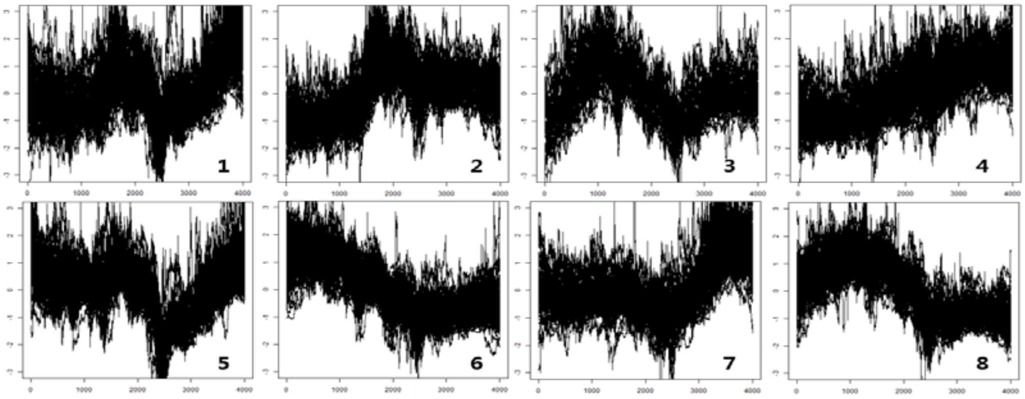

Fig. 11. Clusters generated by K-means(15)

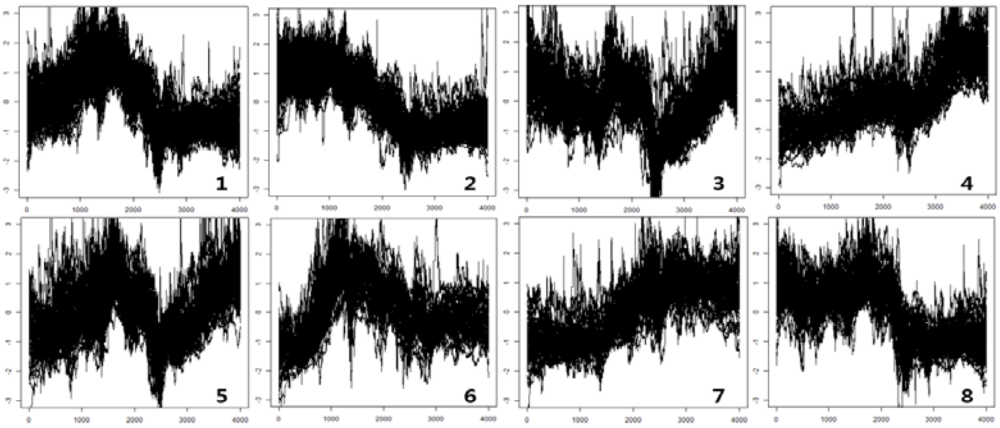

Fig. 12. Clusters generated by K-means(20)

Fig. 10, **Fig. 11**, and **Fig. 12** show clusters generated by K-means. K-means is a partition-based clustering method. If the number of clusters selected by the user is not correct, the size of the clusters generated by the algorithm is not appropriate. This causes the cluster to include time series that are not similar in shape. In previous experiments, clusters generated by K-means generally had high MSD and RMSE values.

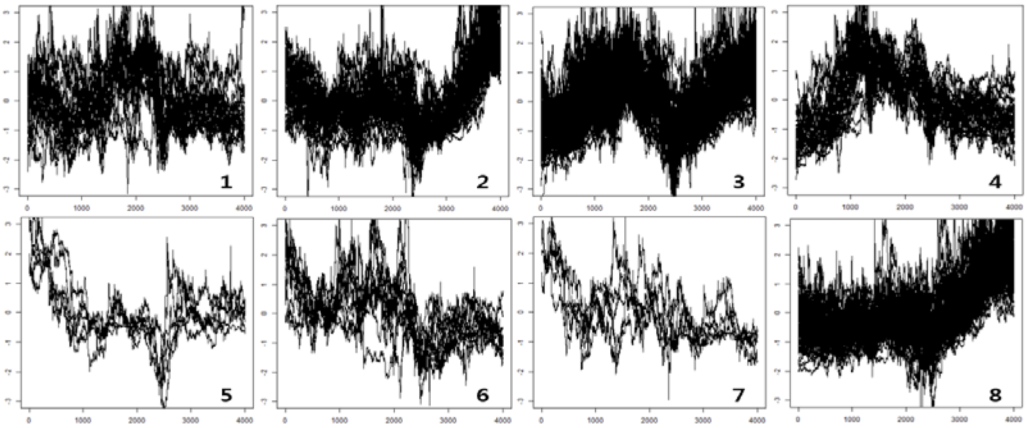

Fig. 13. Clusters generated by ThreePTC

Fig. 13 shows the clusters generated by ThreePTC. The ThreePTC algorithm reduces the dimension of the time series using SAX, which has a large loss of information in the dimension reduction step. In the refinement step, the clusters are refined according to the number of clusters selected by the user using the partition-based clustering algorithm. These two features cause the performance of ThreePTC algorithm to deteriorate. Therefore, the RMSE and MSD values of the clusters generated by ThreePTC are significantly larger than those of other algorithms. In **Fig. 13**, we confirmed that the time series in the clusters generated by ThreePTC have different shapes. Although the ThreePTC algorithm includes a refinement step, we can see that the quality of the clusters is not improved much through the refinement step.

5. Conclusion

In this paper, we propose a clustering method that allows time series with similar shapes to be included in the same cluster. An important disadvantage of existing time series clustering methods is that time series of similar shapes are included in different clusters, or time series of different shapes are included in the same cluster, which degrades the quality of the cluster. We solve this problem by using a dimension-reduction method with less loss of information, and by limiting the size of the clusters. In our experiments, we have shown that our proposed method performs better than the existing methods. The clustering algorithm proposed in this paper can be used as a basic method for applying various time series applications such as time series prediction and robot control. In the application of a financial time series, it can be used as a basic method in portfolio construction and risk management. As future work, we plan to apply this clustering method to finance and economics.

References

[1] Cheng-Ping Lai, Pau-Choo Chung, Vincent S. Tseng, “A novel two-level clustering method for time series data analysis,” *Expert Systems with Applications*, Vol. 37, pp. 6319-6326, 2010. Article (CrossRef Link)

[2] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, “Fast subsequence matching in time-series databases,” *ACM SIGMOD*, Vol. 23, Issue. 2, pp. 419–429, 1994. Article (CrossRef Link)

[3] Daxin Jiang, Jian Pei, Aidong Zhang, "DHC: A Density-based Hierarchical Clustering Method for Time Series Gene Expression Data," in *Proc. of Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering*, 2003. Article (CrossRef Link)

[4] E. Keogh, M. Pazzani, K. Chakrabarti, S. Mehrotra, “A simple dimensionality reduction technique for fast similarity search in large time series databases,” *Pacific-Asia Conference on Knowledge Discovery and Data Mining.*, Vol. 1805, pp. 122–133., 2000. Article (CrossRef Link)

[5] F. Korn, H.V. Jagadish, C. Faloutsos, “Efficiently supporting ad hoc queries in large datasets of time sequences,” *ACM SIGMOD*, Vol. 26, pp. 289–300, 1997. Article (CrossRef Link)

[6] Jiawei Han, Micheline Kamber, Jian Pei, "Data Mining Concepts and Technique 3rd," *Morgan Kaufmann*, 2011.

[7] J. Lin, E. Keogh, L. Wei, S. Lonardi, “Experiencing SAX: a novel symbolic representation of time series,” *Data Min. Knowl. Discov.*, Vol. 15, Issue. 2, pp. 107–144, 2007. Article (CrossRef Link)

[8] K. Chan, A.W. Fu, “Efficient time series matching by wavelets,” in *Proc. of IEEE International Conference on Data Engineering*, vol. 15, no. 3, pp. 126 – 133, 1999. Article (CrossRef Link)

[9] Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. of International Conference on Knowledge Discovery and Data Mining*, pp. 226-231, 1996

[10] Michel Verleysen, Damien Fracois, “The Curse of Dimensionality in Data Mining and Time Series Prediction,” *IWANN*, vol. 3512, pp. 758-770., 2005. Article (CrossRef Link)

[11] Philippe esling and Carlos agon, “Time-Series Data Mining,” *ACM Computing Surveys*, Vol. 45, No. 1, Article 12, 2012. Article (CrossRef Link)

[12] Rui Ding, Qiang Wang, Yingnong Dang, Qiang Fu, Haidong Zhang, Dongmei Zhang, “YADING: Fast Clustering of Large-Scale Time Series Data,” *Proceedings of the VLDB Endowment*, Vol 8. Issue 5, PP. 473-484, 2015. Article (CrossRef Link)

[13] R. C. Dubes and A. K. Jain. “Algorithms for Clustering Data,” *Prentice Hall*, 1988

[14] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, The Ying Wah, ”Time series clustering – A decade review,” *Information Systems*, Vol. 53, pp. 16-38., 2015. Article (CrossRef Link)

[15] Saeed Aghabozorgi,Ying Wah The, “Stock market co-movement assessment using a three-phase clustering method,” *Expert Systems with Applications*, Vol. 41, pp.1301-1314, 2014. Article (CrossRef Link)

[16] Saeed Aghabozorgi, The Ying Wah, Tutut Herawan, Hamid A.Jalab, Mohammad Amin Shaygan, and Alireza Jalali, “A Hybrid Algorithm for Clustering of Time Series Data Based on Affinity Search Technique,” *The Scientific World Journal*, Vol. 2014, 12pages, 2014. Article (CrossRef Link)

[17] Zechao Li, Jing Liu, Yi Yang, Xiaofang Zhou, Hanqing Lu, “Clustering-Guided Sparse Structural Learning for Unsupervised Feature Selection,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, Issue. 9, 2014. Article (CrossRef Link)

[18] Zechao Li, Jing Liu, Jinhui Tang, Hanqing Lu, “Robust Structured Subspace Learning for Data Representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, Issue. 10, pp. 2138-2150 2015. Article (CrossRef Link)

[19] Zechao Li, Jinhui Tang, “Unsupervised Feature Selection via Nonnegative Spectral Analysis and Redundancy Control,” *IEEE Transactions on Image Processing*, Vol. 24, Issue. 12, 2015. Article (CrossRef Link)

Jungyu Ahn received the M.S. degree in 2018 and B.S. degree in 2015 from Inha University. His research interests are Reinforcement learning, Time series Clustering, Financial Time series Data Mining.

Juhong Lee is a professor of computer engineering at Inha University. He received the Ph.D degree in 2001 from KAIST and He received M.S. degree in 1985 and B.S. degree in 1983 from Seoul National University. His research intersts are Reinforcement learning, Financial Time series Data Mining.