Communications for Statistical Applications and Methods
2020, Vol. 27, No. 6, 589–602

https://doi.org/10.29220/CSAM.2020.27.6.589
Print ISSN 2287-7843 / Online ISSN 2383-4757

# Comparison of time series clustering methods and application to power consumption pattern clustering

Jaehwi Kim[a], Jaehee Kim[1,b]

[a]Korea Rural Economic Institute, Korea;
[b]Department of Statistics, Duksung Women's University, Korea

**Abstract**

The development of smart grids has enabled the easy collection of a large amount of power data. There are some common patterns that make it useful to cluster power consumption patterns when analyzing s power big data. In this paper, clustering analysis is based on distance functions for time series and clustering algorithms to discover patterns for power consumption data. In clustering, we use 10 distance measures to find the clusters that consider the characteristics of time series data. A simulation study is done to compare the distance measures for clustering. Cluster validity measures are also calculated and compared such as error rate, similarity index, Dunn index and silhouette values. Real power consumption data are used for clustering, with five distance measures whose performances are better than others in the simulation.

Keywords: complexity distance, model-free distance, model-based distance, power consumption, time series clustering, silhouette

## 1. Introduction

Smart grids have many advantages over traditional conventional power grids because they enhance the way electricity is generated, distributed, and consumed by using advanced sensing devices and controllers that depend on power consumption profiles. With the development of the smart grid technology, electrical data is accumulating into big data in real time. According to Haben *et al.* (2015), the clustering method of the electrical data is changing from a method that has been applied to a large customer group or the energy data of a high/medium voltage customer to low voltage clustering such as household level. Clustering algorithms are used to profile individuals' power consumption and analyzing their patterns (Al-Jarrah *et al.*, 2017; Tsekouras *et al.*, 2007). This analyzes what energy suppliers and distribution network operators need to understand how customers use energy and its impact on voltage networks. Clustering issues are at the heart of many knowledge discovery and data mining tasks that are also useful in identifying data patterns (Xiong and Yeung, 2004). Clustering is an unsupervised process such as turning grouping data patterns into clusters, therefore the patterns within a cluster are similar but different from between the other clusters. The goal of clustering is to identify structures in an unlabeled data set by objectively organizing data into homogeneous clusters where the within-cluster-object similarity is minimized and the between-cluster-object dissimilarity is maximized. Time series clustering is a research area applicable to a wide range of fields (Liao, 2005). There seems to be an increased interest in time series clustering as a part of temporal data mining

[1] Corresponding author: Department of Statistics, Duksung Women's University, Samyang-ro 144-gil 33, Dobong-gu, Seoul 01369, Korea. E-mail: jaehee@duksung.ac.kr

Published 30 November 2020 / journal homepage: http://csam.or.kr

research. Note that time series clustering may cause some multivariate clustering algorithms to fail, because the similarity notation at higher dimensions becomes obscure when the time series is very long (Wang *et al.*, 2006). Therefore, it is necessary to consider clustering using a suitable distance measure between time series incorporating their dependency.

There are various types of distances used in clustering such as model-free, model-based, and complexity methods (Montero and Vilar, 2014). The model-free method is used to measure the proximity between two time series based on the closeness of their values at specific points in time. One way is to compare the autocorrelation function (ACF) between two time series (Bohte *et al.*, 1980; Galeano and Peña, 2000; Caiado *et al.*, 2006; D'Urso and Maharaj, 2009). Fréchet (1906) calculates the distance by considering all combinations of the time points between two time series.

Model-based approaches consider that each time series is generated by some kind of model or by a mixture of underlying probability distributions. Time series are considered similar when model parameters characterizing individual series or the remaining residuals after fitting the model are similar. For example, the distance using the correlation of two time series (Golay *et al.*, 1998) is used. The greater the correlation between two time series, the closer the distance is given. Chouakria and Nagabhushan (2007) propose a distance that covers both existing measures of the correlation between the proximity of observations and the behavior proximity estimates between two time series. Caiado *et al.* (2006) compared the periodograms of two time series and expressed them in Euclidean distance form. For example, the model-based method uses the assumption that each time series follows the ARIMA model. Piccolo (1990) calculates the distance between two time series using the Euclidean distance between parameters of the AR models. Maharaj (1996, 2000) compare two time series by setting the chi-square test statistic using the parameters of the AR models. Kalpakis *et al.* (2001) calculate the linear predictive coding (LPC) cepstrum and Euclidean distance between the LPC cepstrums.

Complexity-based approaches compare the levels of complexity of time series. The similarity of two time series does not rely on specific serial features or the knowledge of underlying models, but on measuring the level of information shared by both time series. The mutual information between two series can be formally established using the Kolmogorov complexity concept; however this measure cannot be computed in practice and must be approximated. There are two ways to consider the complexity: calculate the complexity of each time series and compare them with each other (Li *et al.*, 2004; Keogh *et al.*, 2004; Keogh *et al.*, 2007), or to set the weighting function for complexity (Batista *et al.*, 2011).

In this paper we compare clustering methods for time series and apply them to power consumption times series as power consumption profiling. This practical application is important for load forecasting, bad data correction, optimal energy resources scheduling. This paper is organized as follows. Section 2 introduces 10 distance measures considered, and Section 3 provides hierarchical and K-means clustering methods and clustering comparative measures. Section 4 shows the clustering results of simulation data and its application to electricity consumption data. Discussions are in Section 5. Clustering analysis is implemented with R program and TSclust package.

## 2. Dissimilarity measure

Let $\mathbf{X} = (X_1, \ldots, X_T)'$ and $\mathbf{Y} = (Y_1, \ldots, Y_T)'$ denote subsets from two real-valued processes $X = \{X_t, t \in \mathbb{Z}\}$ and $Y = \{Y_t, t \in \mathbb{Z}\}$, respectively. Here $\mathbf{X}$ and $\mathbf{Y}$ are both time series. There are $N$ cases of time series whose time-length is $T$ in the data set.

### 2.1. Autocorrelation-based distance

The autocorrelation function is used as a dissimilarity measure and some authors have studied this type of measure (Bohte *et al.*, 1980; Galeano and Peña, 2000; Caiado *et al.*, 2006; D'Urso and Maharaj, 2009). Galeano and Peña (2000) define a distance between **X** and **Y** with the estimated autocorrelations vectors as

$$d_{\mathrm{ACF}}(\mathbf{X},\mathbf{Y}) = \sqrt{(\hat{\boldsymbol{\rho}}_{\mathbf{X}} - \hat{\boldsymbol{\rho}}_{\mathbf{Y}})'\,\boldsymbol{\Omega}\,(\hat{\boldsymbol{\rho}}_{\mathbf{X}} - \hat{\boldsymbol{\rho}}_{\mathbf{Y}})}, \tag{2.1}$$

where $\hat{\boldsymbol{\rho}}_{\mathbf{X}} = (\hat{\rho}_{1,\mathbf{X}}, \dots, \hat{\rho}_{L,\mathbf{X}})'$ and $\hat{\boldsymbol{\rho}}_{\mathbf{Y}} = (\hat{\rho}_{1,\mathbf{Y}}, \dots, \hat{\rho}_{L,\mathbf{Y}})^{\top}$ are the estimated autocorrelation vectors of **X** and **Y** respectively. And $\boldsymbol{\Omega}$ is a weight matrix. For some $L$, $\hat{\rho}_{i,\mathbf{X}} \approx 0$ and $\hat{\rho}_{i,\mathbf{Y}} \approx 0$ for $i > L$.

If $\boldsymbol{\Omega} = \mathbf{I}$ with the uniform weights,

$$d_{\mathrm{ACFU}}(\mathbf{X},\mathbf{Y}) = \sqrt{\sum_{i=1}^{L} (\hat{\rho}_{i,\mathbf{X}} - \hat{\rho}_{i,\mathbf{Y}})^2}. \tag{2.2}$$

If the geometric weights are decaying according to the autocorrelation lag,

$$d_{\mathrm{ACFG}}(\mathbf{X},\mathbf{Y}) = \sqrt{\sum_{i=1}^{L} p(1-p)^i\,(\hat{\rho}_{i,\mathbf{X}} - \hat{\rho}_{i,\mathbf{Y}})^2}, \quad \text{with } 0 < p < 1. \tag{2.3}$$

Likewise this measure evaluates the dissimilarity between the corresponding spectral representations of the series.

## 2.2. Correlation-based distance

Correlation measures the similarity of two series. Golay *et al.* (1998) propose correlation-based distances such as

$$d_{\mathrm{Cor}}(\mathbf{X},\mathbf{Y}) = \sqrt{2\,(1 - \mathrm{Cor}\,(\mathbf{X},\mathbf{Y}))} \tag{2.4}$$

based on the Pearson correlation between **X** and **Y**

$$\mathrm{Cor}(\mathbf{X},\mathbf{Y}) = \frac{\sum_{t=1}^{T}\left(X_t - \bar{X}\right)\left(Y_t - \bar{Y}\right)}{\sqrt{\sum_{t=1}^{T}\left(X_t - \bar{X}\right)^2}\sqrt{\sum_{t=1}^{T}\left(Y_t - \bar{Y}\right)^2}},$$

where $\bar{X}$ and $\bar{Y}$ are the averages of the time series **X** and **Y** respectively.

## 2.3. An adaptive dissimilarity index covering both proximity on value and on behavior

Chouakria and Nagabhushan (2007) propose a dissimilarity measure that covers both existing measures of the proximity on observations and temporal correlations for the behavior proximity estimation. The proximity between the behaviors of the series is evaluated by the first-order temporal correlation coefficient as follows:

$$\mathrm{CorT}(\mathbf{X},\mathbf{Y}) = \frac{\sum_{t=1}^{T-1}(X_{t+1} - X_t)(Y_{t+1} - Y_t)}{\sqrt{\sum_{t=1}^{T-1}(X_{t+1} - X_t)^2}\sqrt{\sum_{t=1}^{T-1}(Y_{t+1} - Y_t)^2}}.$$

CorT($\mathbf{X},\mathbf{Y}$) has a value from $-1$ to 1. The value CorT($\mathbf{X},\mathbf{Y}$) $= 1$ means that both series have similar temporal behavior with a similar direction and instantaneous growth rate. If CorT($\mathbf{X},\mathbf{Y}$) equals 0, there is no monotonicity between $\mathbf{X}$ and $\mathbf{Y}$ and their growth rate is stochastically linearly independent (different behaviors). $\mathrm{CorT}(\mathbf{X}, \mathbf{Y}) = -1$ means that the rate of growth is similar but the direction is opposite (opposite behaviors). The dissimilarity measure as a function of CorT is

$$d_{\mathrm{CorT}}(\mathbf{X}, \mathbf{Y}) = \phi_k \left[\mathrm{CorT}(\mathbf{X}, \mathbf{Y})\right] \cdot d(\mathbf{X}, \mathbf{Y}), \tag{2.5}$$

where $\phi_k(\cdot)$ is an adaptive tuning function to automatically regulate an existing raw-data distance d($\mathbf{X}$,$\mathbf{Y}$) according to the temporal correlation with $\phi_k(u) = 2/(1+\exp(ku))$, $k \geq 0$. The value $\mathrm{CorT}(\mathbf{X}, \mathbf{Y}) = 0$ implies $d_{\mathrm{CorT}}(\mathbf{X}, \mathbf{Y}) = d(\mathbf{X}, \mathbf{Y})$.

## 2.4. Periodogram-based distance

Caiado *et al.* (2006) propose the Euclidean distance between the periodogram ordinates as

$$d_{\mathrm{Per}}(\mathbf{X}, \mathbf{Y}) = \frac{1}{n}\sqrt{\sum_{k=1}^{n} \left(I_{\mathbf{X}}(\lambda_k) - I_{\mathbf{Y}}(\lambda_k)\right)^2}, \tag{2.6}$$

where $I_X(\lambda_k) = T^{-1}|\sum_{t=1}^{T} X_t e^{-i\lambda_k t}|^2$ is the periodogram of $\mathbf{X}$ and $I_Y(\lambda_k) = T^{-1}|\sum_{t=1}^{T} Y_t e^{-i\lambda_k t}|^2$ is the periodogram of $\mathbf{Y}$, at frequencies $\lambda_k = 2\pi k/T$, $k = 1, \ldots, n$, with $n = [(T-1)/2]$.

## 2.5. Fréchet distance

Fréchet (1906) proposed a method for measuring proximity between continuous curves. Fréchet distance is widely used in the discrete case (Eiter and Mannila, 1994) and time series framework. Let $M$ be the set of all possible sequences of $m$ pairs that are preserved in the form of observation order $r$ denoted as

$$r = \left((X_{a_1}, Y_{b_1}), \ldots, (X_{a_m}, Y_{b_m})\right)$$

while $a_i, b_j \in \{1, \ldots, T\}$ such that $a_1 = b_1 = 1$, $a_m = b_m = T$, and $a_{i+1} = a_i$ or $a_i + 1$ and $b_{i+1} = b_i$ or $b_i + 1$, for $i \in \{1, \ldots, m-1\}$.

The Fréchet distance is defined by

$$d_F(\mathbf{X}, \mathbf{Y}) = \min_{r \in M}\left(\max_{i=1,\ldots,m} \left|X_{a_i} - Y_{b_i}\right|\right). \tag{2.7}$$

Fréchet distance not only treats the series as two point sets, but also considers the order of observation. Note that $d_F(\mathbf{X}, \mathbf{Y})$ can also be computed on series of different lengths.

## 2.6. Piccolo distance

Piccolo (1990) argues that autoregressive expansions convey all the useful information about the stochastic structure of processes except for initial values. If the series are non-stationary, differencing is carried out to make them stationary. If the series have seasonality, seasonality is removed before further analysis, then they are fitted with the truncated AR($\infty$) models of order $k_1$ and $k_2$ approximating the generation processes of $\mathbf{X}$ and $\mathbf{Y}$, respectively, using criteria such as Akaike information criterion (AIC) and Bayesian information criterion (BIC). This approach solves the problem of obtaining ad hoc ARMA approximation for each series subjected to clustering.

The Piccolo's distance with $k = \max(k_1, k_2)$ takes the following form as

$$d_{\text{Pic}}(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{j=1}^{k} \left(\hat{\pi}^*_{j,\mathbf{X}} - \hat{\pi}^*_{j,\mathbf{Y}}\right)^2}, \tag{2.8}$$

where $\hat{\mathbf{\Pi}}_{\mathbf{X}} = (\hat{\pi}_{1,\mathbf{X}}, \ldots, \hat{\pi}_{k_1,\mathbf{X}})'$ and $\hat{\mathbf{\Pi}}_{\mathbf{Y}} = (\hat{\pi}_{1,\mathbf{Y}}, \ldots, \hat{\pi}_{k_2,\mathbf{Y}})'$ denote the vectors of AR($k_1$) and AR($k_2$) parameter estimations for $\mathbf{X}$ and $\mathbf{Y}$, respectively.

If $j \leq k_1$, and $\hat{\pi}^*_{j,\mathbf{X}} = 0$ otherwise, and analogously $\hat{\pi}^*_{j,\mathbf{Y}} = \hat{\pi}_{j,\mathbf{Y}}$, if $j \leq k_2$, and $\hat{\pi}^*_{j,\mathbf{Y}} = 0$ otherwise. In addition to satisfying the distance properties such as non-negativity, symmetry and triangularity, $\sum \pi_j$, $\sum \|\pi_j\|$, $\sum \pi_j^2$ are well-defined quantities. Therefore $d_{\text{Pic}}$ always exists in all invertible ARIMA processes.

## 2.7. Maharaj distance

A major feature of the Maharaj method is the introduction of hypothesis tests to see if the two time series are significantly different. Maharaj (1996, 2000) uses hypothesis testing to determine whether the two time series have significantly different generating processes for the invertible and stationary ARIMA classes.

The test statistic is

$$d_{\text{Mah}}(\mathbf{X}, \mathbf{Y}) = \sqrt{T}\left(\hat{\mathbf{\Pi}}^*_{\mathbf{X}} - \hat{\mathbf{\Pi}}^*_{\mathbf{Y}}\right)' \hat{\mathbf{V}}^{-1} \left(\hat{\mathbf{\Pi}}^*_{\mathbf{X}} - \hat{\mathbf{\Pi}}^*_{\mathbf{Y}}\right), \tag{2.9}$$

where $\hat{\mathbf{\Pi}}^*_{\mathbf{X}}$ and $\hat{\mathbf{\Pi}}^*_{\mathbf{Y}}$ are the AR($k$) parameter estimations of $\mathbf{X}$ and $\mathbf{Y}$, respectively, with $k$ selected as in the Piccolo's distance. $\hat{\mathbf{V}}$ is an estimator of $\mathbf{V} = \sigma^2_{\mathbf{X}}\mathbf{R}^{-1}_{\mathbf{X}}(k) + \sigma^2_{\mathbf{Y}}\mathbf{R}^{-1}_{\mathbf{Y}}(k)$. $\sigma^2_{\mathbf{X}}$ and $\sigma^2_{\mathbf{Y}}$ are the variances of the white noise processes associated with $\mathbf{X}$, $\mathbf{Y}$. $\mathbf{R}_{\mathbf{X}}$ and $\mathbf{R}_{\mathbf{Y}}$ are the sample covariance matrices of both series. $d_{\text{Mah}}$ is asymptotically $\chi^2$ distributed under the null hypothesis $\mathbf{\Pi}_{\mathbf{X}} = \mathbf{\Pi}_{\mathbf{Y}}$.

Therefore, the dissimilarity measure between $\hat{\mathbf{\Pi}}_{\mathbf{X}}$ and $\hat{\mathbf{\Pi}}_{\mathbf{Y}}$ through associated $p$-value is

$$d_{\text{Mah},p}(\mathbf{X}, \mathbf{Y}) = P\left(\chi^2_k > d_{\text{Mah}}(\mathbf{X}, \mathbf{Y})\right). \tag{2.10}$$

If a hierarchical algorithm starting from the pairwise matrix of $p$-values $d_{\text{Mah},p}$ is developed, then a clustering homogeneity criterion is provided by pre-specifying a threshold significance level $\alpha$ such as 5% or 1%. Those series with associated $p$-values greater than $\alpha$ will be grouped together. It implies that only those series whose dynamic structures are not significantly different at level $\alpha$ will be placed in the same group. The test statistic $d_{\text{Mah}}$ and the associated $p$-value $d_{\text{Mah},p}$ satisfy the properties of non-negativity and symmetry; therefore they can be used as dissimilarity of $\mathbf{X}$ and $\mathbf{Y}$. $d_{\text{Mah}}$ evaluates dissimilarity by comparing autoregressive approximations of two series, similar to Piccolo distance $d_{\text{Pic}}$. But $d_{\text{Mah}}$ can be affected by the scale unit because it uses variance of white noise processes, unlike $d_{\text{Pic}}$.

## 2.8. Cepstral-based distance

The linear predictive coding (LPC) cepstrum is proposed by Kalpakis *et al.* (2001) to be useful for clustering ARIMA time series, and provides good properties for distinguishing between ARIMA time series. The cepstrum is defined as the inverse Fourier transform of the logarithm of a signal spectrum. The LPC cepstrum is a cepstrum constructed using autoregression coefficients from the linear model of the signal. The LPC cepstrum is so named because the cepstrum is derived from the LPC of the signal.

Cepstral-based distance is calculated as the Euclidean distance between the LPC cepstral coefficients of $\mathbf{X}$ and $\mathbf{Y}$,

$$d_{\text{LPC.Cep}}(\mathbf{X},\mathbf{Y}) = \sqrt{\sum_{i=1}^{T} \left(\psi_{i,\mathbf{X}} - \psi_{i,\mathbf{Y}}\right)^2}, \tag{2.11}$$

where a time series $\mathbf{X}$ following an AR($p$) structure such as $X_t = \sum_{r=1}^{p} \phi_r X_{t-r} + \epsilon_t$ with the autoregression coefficients $\phi_r$, a white noise process $\epsilon_t \sim (0, \sigma^2)$. Here the LPC cepstral coefficients are defined as

$$\psi_h = \begin{cases} \phi_1, & \text{if } h = 1, \\ \phi_h + \sum_{r=1}^{h-1} \left(\phi_r - \psi_{h-r}\right), & \text{if } 1 < h \le p, \\ \sum_{r=1}^{p} \left(1 - \frac{r}{h}\right) \phi_r \psi_{h-r}, & \text{if } p < h. \end{cases}$$

## 2.9. Compression-based dissimilarity measure

The Kolmogorov complexity of an object $x$, $K(x)$ is the shortest program length that can produce $x$ on a universal computer such as a Turing machine. $K(x)$ is the minimum amount of information for generating $x$ by an algorithm. The larger $K(x)$, the greater the complexity. Similarly, given two objects x and y, the conditional Kolmogorov complexity of $x$ given $y$, $K(x|y)$ is the length of the minimum program that produces $x$ when $y$ is given as an auxiliary input on the program. Therefore, $K(x) - K(x|y)$ is the amount of information that $y$ generates for $x$.

Based on this theory, Li *et al.* (2004) present the normalized information distance (NID) as:

$$d_{\text{NID}}(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}. \tag{2.12}$$

Here $d_{\text{NID}}$ is a metric with a value of $[0, 1]$. The biggest problem with $d_{\text{NID}}$ is that Kolmogorov complexity is uncomputable. So, approximate $K(\cdot)$ by the length of the compression object obtained from data compressors such as gzip and bzip2 (compression program).

Let $C(\mathbf{X})$ and $C(\mathbf{Y})$ be the compressed size of $\mathbf{X}$ and $\mathbf{Y}$, respectively. The denominator of $d_{\text{NID}}$ is easily approximated to $\max\{C(\mathbf{X}), C(\mathbf{Y})\}$, but the numerator is difficult to approximate because it contains conditional Kolmogorov complexities. Li *et al.* (2004) solve this problem by using the theory that $K(x|y)$ is roughly equal to $K(xy) - K(y)$, where $K(xy)$ is the minimum program length to calculate the sequence of $x$ and $y$. A normalized compression distance (NCD) approximating the NID is expressed as

$$d_{\text{NCD}}(\mathbf{X},\mathbf{Y}) = \frac{C(\mathbf{XY}) - \min\{C(\mathbf{X}), C(\mathbf{Y})\}}{\max\{C(\mathbf{X}), C(\mathbf{Y})\}}. \tag{2.13}$$

A metric $d_{\text{NCD}}$ has a value from 0 to $1 + \epsilon$, where $\epsilon$ is the error due to the deficiency of the compression techniques.

Keogh *et al.* (2004) (see also Keogh *et al.* (2007)) propose a simplified version of the NCD called a compression-based dissimilarity measure (CDM) as

$$d_{\text{CDM}}(\mathbf{X},\mathbf{Y}) = \frac{C(\mathbf{X},\mathbf{Y})}{C(\mathbf{X})C(\mathbf{Y})}. \tag{2.14}$$

This metric $d_{\text{NCD}}$ ranges from $1/2$ to 1, where $1/2$ means pure identity and 1 means maximum discrepancy.

### 2.10. Complexity-invariant dissimilarity measure

Batista *et al.* (2011) argue that highly complex time series pairs often tend to be further apart than simple series pairs. This means that complex series can be incorrectly assigned to a class with less complexity. To reduce this effect, Batista *et al.* (2011) use information on the difference in complexity between the two series as a correction factor for conventional dissimilarity measures. A complexity-invariant dissimilarity measure (CID) is defined as

$$d_{\text{CID}}(\mathbf{X}, \mathbf{Y}) = \text{CF}(\mathbf{X}, \mathbf{Y}) \cdot d(\mathbf{X}, \mathbf{Y}) \tag{2.15}$$

where an existing raw-data distance $d(\mathbf{X}, \mathbf{Y})$ and a complexity correction factor $\text{CF}(\mathbf{X}, \mathbf{Y})$ with

$$\text{CF}(\mathbf{X}, \mathbf{Y}) = \frac{\max\{\text{CE}(\mathbf{X}), \text{CE}(\mathbf{Y})\}}{\min\{\text{CE}(\mathbf{X}), \text{CE}(\mathbf{Y})\}}, \qquad \text{CE}(\mathbf{X}) = \sqrt{\sum_{t=1}^{T-1} (X_t - X_{t+1})^2}.$$

Here $\text{CE}(\cdot)$ is a complexity estimator of series. If all the complexity of the series is the same, then $d_{\text{CID}}(\mathbf{X}, \mathbf{Y}) = d(\mathbf{X}, \mathbf{Y})$.

## 3. Clustering methods

### 3.1. Hierarchical clustering

Hierarchical clustering works by combining individual objects with similar objects or groups sequentially and hierarchically using a tree model. It uses a dendrogram, a tree-like structure that shows the order in which objects are joined, so we can do this without having to define a number of clusters in advance. After creating the dendrogram, the tree can be cut at the appropriate level to divide the entire data into several clusters. The distance or similarity between objects is required to perform hierarchical clustering. We calculate group objects that are close together in sequence along with the distance between the newly bundled cluster and another object (or another cluster). Several linkage methods can be used and we use the complete linkage method in this paper. The complete linkage method uses the farthest distance of all pairs of objects in two clusters when calculating the distance between two clusters.

### 3.2. K-means clustering

K-means clustering is a method of forming clusters by gathering individuals close to the center of each cluster. K-means clustering must first set the center of each cluster. Unlike hierarchical clustering, it can only work by specifying the number of clusters in advance. Let $X = C_1 \cup C_2 \cup \cdots \cup C_K$, and $C_i \cap C_j = \phi$. Clusters are determined by

$$\arg\min_{C} \sum_{i=1}^{K} \sum_{x_j \in C_i} \left\| x_j - c_i \right\|^2, \tag{3.1}$$

where $c_i$ is the center of the cluster. The operation of K-means clustering is similar to the EM algorithm. Initially, each object is assigned to a cluster based on the centers of the cluster given at random.

The center value is calculated in each cluster. The objects are then assigned to the cluster again, according to the new center. The operation stops when the center converges in a certain place or fills a predetermined number of iterations.

### 3.3. Clustering validity

To evaluate clustering validity we consider four measures. Error rate (ER) and similarity (sim) index compare the actual clusters with the estimated clusters. The Dunn index and silhouette measure are computed to compare within cluster connectedness.

(i) Error rate (ER)

Let $H$ be a clustering map defined as

$$H(f,g) = \begin{cases} 1, & \text{if } f \text{ and } g \text{ are in the same cluster,} \\ 0, & \text{otherwise.} \end{cases}$$

Regarding the estimation error, the clustering estimation error rate $\eta(K)$ is defined by Serban and Wasserman (2005) as

$$\eta(K) = \frac{1}{{}_N\mathrm{C}_2} \sum_{r<s} I\left(H_K(f_r, f_s)\right) \neq \hat{H}_K\left(\hat{f}_r, \hat{f}_s\right), \tag{3.2}$$

where $C = \{f_1, \ldots, f_N\}$ denote the true curves and $\hat{C} = \{\hat{f}_1, \ldots, \hat{f}_N\}$ denote the estimated curves.

(ii) Similarity index (Sim index)

$$\mathrm{Sim}\left(\mathcal{G}, C\right) = \frac{1}{K} \sum_{i=1}^{K} \max_{1 \le j \le K} \mathrm{Similarity}\left(\mathcal{G}_i, C_j\right), \quad \text{where } \mathrm{Similarity}\left(\mathcal{G}_i, C_j\right) = \frac{\left[\mathcal{G}_i \cap C_j\right]}{\left[\mathcal{G}_i\right] + \left[C_j\right]}, \tag{3.3}$$

where $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$ is the true partition, and $C = \{C_1, \ldots, C_K\}$ is the partition obtained via clustering algorithm. $[\cdot]$ denotes the cardinality of the elements in the set.

(iii) Dunn index

Dunn (1974) proposed an internal measure of clustering as

$$D(C) = \frac{\min_{C_k, C_l \in C, C_k \neq C_l}\left(\min_{i \in C_k, j \in C_l} \mathrm{dist}(i,j)\right)}{\max_{C_m \in C} \mathrm{diam}\left(C_m\right)}, \tag{3.4}$$

where $\mathrm{diam}\left(C_m\right)$ is the maximum distance between observations in cluster $C_m$, and $d(i,j)$ is the distance between data points $x_i$ and $x_j$ in the cluster $C_i$. A larger value means it is tighter in the same cluster.

(iv) Silhouette

Rousseeuw (1987) proposed silhouette width as the average of each observation's silhouette value.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{3.5}$$

where

$$a(i) = \frac{1}{[C_i] - 1} \sum_{j \in C_i, i \neq j} d(i, j) \quad \text{and} \quad b(i) = \min_{k \neq i} \frac{1}{[C_k]} \sum_{j \in C_k} d(i, j).$$

The silhouette value is from −1 to 1 and means how tightly grouped an internal evaluation is. A larger value means it is tighter.

## 4. Data analysis

### 4.1. Simulation

We design a set of 100 time series in five clusters. Each cluster has 20 time series. Each time series has 96 time points. The errors are generated independently from $N(0, 1)$. The description of each cluster is :

Cluster 1. Step up and down

$$\begin{cases} X_t \sim N(1, 1), & t = 1, \dots, 20, \\ X_t \sim N(8, 2), & t = 21, \dots, 70, \\ X_t \sim N(1, 1), & t = 71, \dots, 96. \end{cases}$$

Cluster 2. Combined AR model

$$\begin{cases} X_t = 1.5 + 0.8X_{t-1} + \epsilon_t, & t = 1, \dots, 48, \\ X_t = -1 + 0.6X_{t-1} + \epsilon_t, & t = 49, \dots, 96, \end{cases} \quad \text{where } \epsilon_t \sim N(0, 1).$$

Cluster 3. ARMA model

$$X_t = 4 + 0.5X_{t-1} + \epsilon_t + \epsilon_{t-1}, \quad t = 1, \dots, 96, \quad \text{where } \epsilon_t \sim N(0, 1).$$

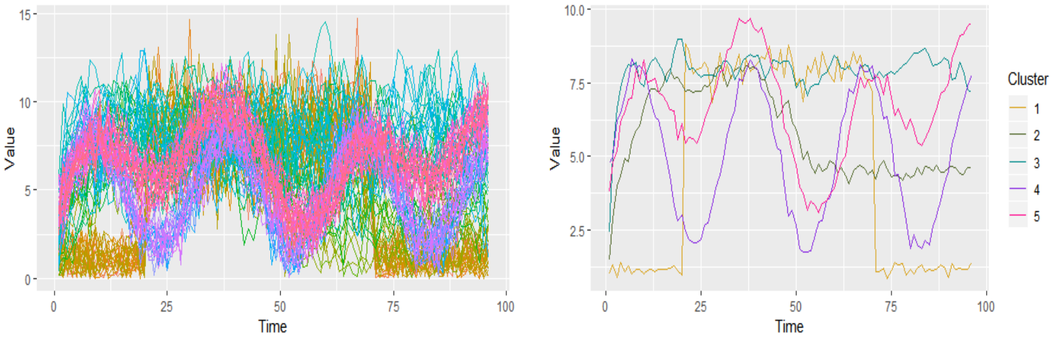
Cluster 4. Periodic model

$$X_t = 3 \sin\left(\frac{20(t-1)}{95}\right) + 5 + \epsilon_t, \quad t = 1, \dots, 96, \quad \text{where } \epsilon_t \sim N(0, 1).$$

Cluster 5. Complex periodic model

$$X_t = 2\left[\sin\left(\frac{20(t-1)}{95}\right) + 2\left|\sin\left(\frac{20(t-1)}{4 \times 95}\right)\right|\right] + 4 + \epsilon_t, \quad t = 1, \dots, 96, \quad \text{where } \epsilon_t \sim N(0, 1).$$

Figure 1 shows one data set (a) in the simulation and mean lines (b) of each cluster. The simulation results are obtained from 100 repetitions. We used 10 distance measures with hierarchical complete linkage and K-means clustering algorithm respectively. Table 1 and Table 2 show the results. Hierarchical and K-means clustering methods with each distance measure provide some similar measures of cluster validity. According to ER, the best performance is done with $d_{\text{CID}}$ in both hierarchical and K-mean clustering. Either clustering with $d_{\text{Per}}$ has the highest silhouette values. Based on the Sim index, $d_{\text{CorT}}$ is the best in hierarchical clustering and $d_{\text{CID}}$ is the best in K-means clustering. The methods are substantially different; however, they lead to acceptable models in the sense of cluster validity measures.



(a) (b)

Figure 1: *One simulation data set of 100 cases (a) and average lines of each cluster (b).*

Table 1: Hierarchical clustering performance with simulation data

| Distance | Measure | | | |
|---|---|---|---|---|
| | ER | Sim index | Dunn index | Silhouette |
| ACF | 0.2991 | 0.5388 | 0.4173 | 0.1006 |
| Cor | 0.1995 | 0.6547 | 0.5747 | 0.2477 |
| CorT | 0.1189 | 0.8002 | 0.3547 | 0.3022 |
| Per | 0.2108 | 0.6954 | 0.3188 | 0.4807 |
| Frechet | 0.1834 | 0.7124 | 0.2723 | 0.2212 |
| AR.Pic | 0.2852 | 0.5771 | 0.1312 | 0.2802 |
| AR.Mah | 0.1797 | 0.7213 | 0.0464 | 0.4702 |
| AR.LPC.CEP | 0.1764 | 0.7117 | 0.1880 | 0.2902 |
| CDM | 0.2545 | 0.5547 | 0.9776 | 0.0009 |
| CID | 0.1055 | 0.8200 | 0.4502 | 0.3623 |

Table 2: K-means clustering performance with simulation data

| Distance | Measure | | | |
|---|---|---|---|---|
| | ER | Sim index | Dunn index | Silhouette |
| ACF | 0.2991 | 0.5388 | 0.4173 | 0.1006 |
| Cor | 0.0886 | 0.8472 | 0.4630 | 0.2282 |
| CorT | 0.0865 | 0.8556 | 0.2889 | 0.2949 |
| Per | 0.1087 | 0.8138 | 0.1562 | 0.4991 |
| Frechet | 0.1059 | 0.8261 | 0.3016 | 0.2649 |
| AR.Pic | 0.1840 | 0.6780 | 0.0696 | 0.2612 |
| AR.Mah | 0.1730 | 0.7314 | 0.0114 | 0.4436 |
| AR.LPC.CEP | 0.1423 | 0.7475 | 0.1119 | 0.2784 |
| CDM | 0.1530 | 0.7089 | 0.9762 | 0.0014 |
| CID | 0.0778 | 0.8708 | 0.3361 | 0.3278 |

In practice, attention should be given to the choice of the distance and clustering algorithm. Each property should be considered to make proper clusters. A range of feature-, model-, and complexity-based dissimilarities are included in this paper. For instance, the K-means clustering algorithm moves each series to the cluster whose centroid is closest in order to recalculate the cluster centroid and repeats the procedure until no more are assigned. The range of proper methods become limited and careful implementation is required once the clustering objectives are made clear and the characteristics of time series are considered.

Table 3: List of 90 devices and buildings in 16 companies located in South Korea

| Company | Type | Company | Type | Company | Type |
|---|---|---|---|---|---|
| D1 | LV2 | S1 | L-B | T1 | Welding Line Distribution Board |
| | LV5 | | LP-A1 | P1 | Unit1_1 |
| | Chiller No.4 | | LP-M | | Unit1_2 |
| | Chiller No.6 | | PA-2 | | Unit1_3 |
| | Chiller No.7 | | PA-3 | | Unit1_4 |
| | Chiller No.8 | | PB-1 | | Unit1_5 |
| | Water treatment | S2 | E_V | | Unit1_coiler |
| D2 | L-1M public | | Water supply | | Unit2_1 |
| | L-2 main | S3 | 1F | | Unit2_2 |
| | L-14 main | | 2,3,4F | | Unit2_3 |
| | LP-CAR parking tower | | 5F | | Unit2_4 |
| M1 | Main device | | Public | | Unit2_coiler |
| | Main office | | Main | | Unit3_1 |
| | Grooving machine | B1 | F-B3 | | Unit3_2 |
| G1 | Public | | L-F | | Unit3_3 |
| | Main | | L-O | | Unit3_incidental equipment |
| | Shopping area | | P-CAR | | Unit3_coiler |
| N1 | 250KVA_Main | | P-E-1 | | Unit4_1 |
| | 300KVA_Main | | P-EHP | | Unit4_2 |
| | Load1 | K1 | Main circuit breaker | | Unit4_3 |
| | Load2 | | Circuit breaker1 | | Unit4_4 |
| | Load3 | | Circuit breaker4 | | Unit4_incidental equipment |
| | Lathe1 | C1 | Load1 | | Unit4_coiler |
| | Lathe2 | | Load2 | H1 | L-1A |
| | New material | | Load3 | | L-M |
| | Grinder | | Load4 | | Public |
| | Urethane1 | T1 | A-04 | H1 | LC-1 office main |
| | Urethane2 | | B-01 | | Main MCCB |
| D3 | Rooftop | | CO2 welding machine | | Sub1 light |
| S1 | L-A | | Main press | | Main1 main |

## 4.2. Electricity consumption data analysis

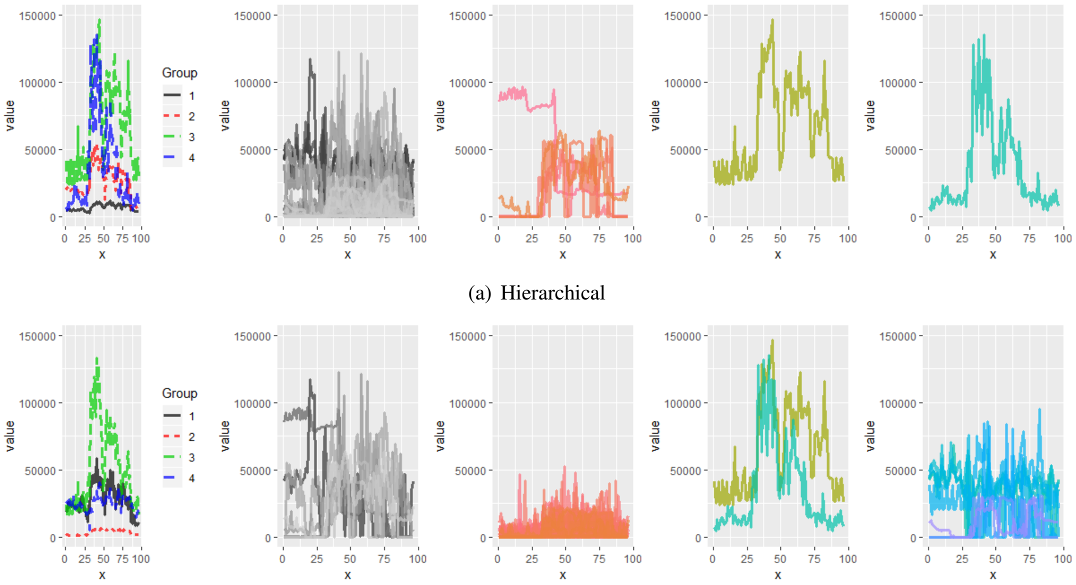
For a clustering application, we use a power consumption data set from a total of 90 devices and buildings in 16 companies located in South Korea on June 19, 2018 (Table 3). Power consumption data measured every 15 minutes is a time series with a total of 96 time points per day. Each power consumption pattern varies widely, but can be divided into three types: continuous consumption, on-off repetition, and turning on and off once a day. For example, machine A repeats on-off every 1 hour from 9 am, but machine B can repeat on-off every 2 to 3 hours even if it starts at 9 am. Various continuous patterns are also possible according to working periods and workloads.

According to comparing the silhouette values from three to ten clusters based on $d_{\text{CID}}$, the silhouette value of resulting four clusters was the highest. Therefore we determined four clusters from $d_{\text{CID}}$ which was the best in the simulation. The distances used for cluster analysis are $d_{\text{AR.Mah}}$, $d_{\text{CID}}$, $d_{\text{Cor}}$, $d_{\text{CorT}}$, and $d_{\text{Per}}$. The missing values in the data are estimated using the Stineman (1980) interpolation which creates a curve with no more inflection points than clearly required by the given set of points. We used this method by na.interpolation of imputeTS (Morits and Bartz-Beielstein, 2017) R packages.

Table 4 provides the clustering validity measures of this real power consumption data. Figure 2 gives clusters and their averaging patterns using the $d_{\text{Per}}$ in hierarchical complete linkage and K-means clustering. The patterns in each cluster are shown in the far left figure with the average lines. Figure

Table 4: Hierarchical & K-means clustering with electricity consumption data with 4 clusters

| Method | Measure | Cor | CorT | Per | AR.Mah | CID |
|---|---|---|---|---|---|---|
| Hierarchical | Dunn index | 0.6077 | 0.2632 | 0.5383 | 0.0059 | 0.0235 |
| | Silhouette | 0.1972 | 0.6823 | 0.7799 | 0.3408 | 0.6142 |
| K-means | Dunn index | 0.4076 | 0.0066 | 0.1415 | 0.0004 | 0.0008 |
| | Silhouette | 0.1002 | 0.4646 | 0.6568 | 0.2940 | 0.2793 |



(a) Hierarchical

(b) K-means

Figure 2: *A real data clustering result by using $d_{\text{Per}}$. (a) Hierarchical, (b) K-means.*

3 provides cluster members by $d_{\text{Per}}$ in K-means. Cluster 1 has some continuous pattern. Cluster 2 has many devices of on-off form, Cluster 3 has some main devices, and chiller devices are in Cluster 4. There is some limit to device interpretation since the exact device information is not allowed.

## 5. Discussion

The development of smart grids has enabled the easy collection of vast amount of power data. In order to efficiently analyze this huge data, it is very useful to quickly catch and cluster power consumption patterns. If you can understand power consumption patterns, there is an advantage in analysis such as prediction. We compared 10 distance measures using hierarchical clustering and K-means clustering. Simulation provides that there is no one best clustering. The time series structure should be reflected in doing clustering analysis. There should be more measures to evaluate time series clustering except Dunn index and silhouette width. This work could be extended to meet the requirements of real-time data processing applications, such as clustering the power consumption of appliances and controlling usage patterns, and possibly the detection of appliances with anomalies that indicate faulty or compromised appliances. We hope that this research will help serve others interested in advancing time series clustering research. For power consumption big data, clustering is a challenging problem considering local and global profiles with computational burden and effective complexity modelling.

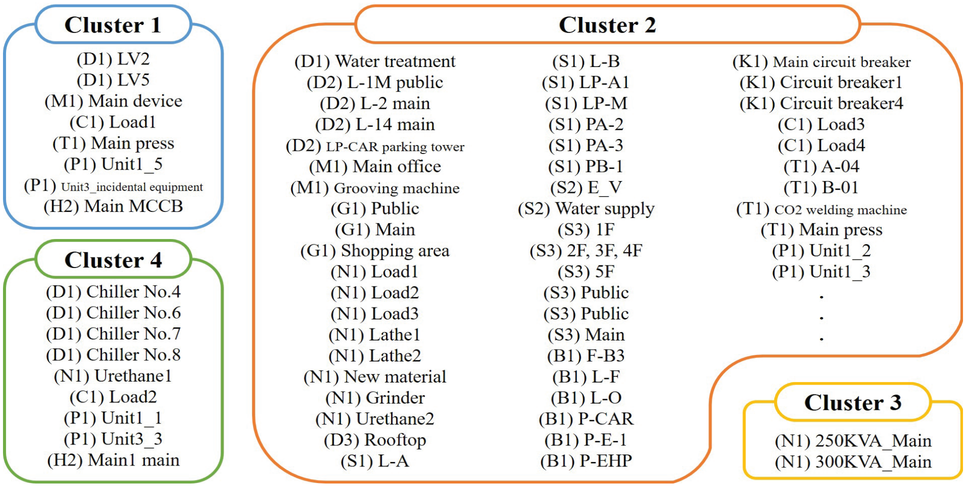


Figure 3: *Cluster members obtained by $d_{\text{Per}}$ in K-means.*

## Acknowledgement

This research was supported by Korea Electric Power Corporation (Grant number: R18XA01). Also it is supported by the Korea Research Foundation (KNRF) (No. 2018R1A2B26001664).

## References

Al-Jhrrah OY, Al-Hammadi Y, and Muhaidat S (2017). Multi-layered clustering for power consumption profiling in smart grids, *IEEE Access*, Digital Object Identifier /ACCESS.2017.2712258.

Batista GE, Wang X, and Keogh EJ (2011). A complexity-invariant distance measure for time series. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, 699–710.

Bohte Z, Čepar D, and Košmelj K (1980). Clustering of time series. In *Compstat 1980: Proceeding in Computational Statistics*, (MM Barritt, D Wishart (eds), 587–593), Physica-Verlag, Heidelberg.

Caiado J, Crato N, and Peña D (2006). A periodogram-based metric for time series classification, *Computational Statistics & Data Analysis*, **50**, 2668–2684.

Chouakria AD and Nagabhushan PN (2007). Adaptive dissimilarity index for measuring time series proximity, *Advances in Data Analysis and Classification*, **1**, 5–21.

Dunn J (1974). Well separated clusters and optimal fuzzy partitions, *Journal of Cybernetics*, **4**, 95–104.

D'Urso P and Maharaj EA (2009). Autocorrelation-based fuzzy clustering of time series, *Fuzzy Sets and Systems*, **160**, 3565–3589.

Eiter T and Mannila H (1994) *Computing discrete frechet distance* (Technical Report CD-TR 94/64), Information Systems Department, Technical University of Vienna, Vienna, Austria.

Fréchet MM (1906). Sur quelques points du calcul fonctionnel, *Rendiconti del Circolo Matematico di Palermo (1884–1940)*, **22**, 1–72.

Galeano P and Peña D (2000). Multivariate analysis in vector time series, *Department de Estadística y Econometría, Universidad Carlos III de Madrid*, Working Paper 01-24 Statistics and Econometrics Series 15.

Golay X, Kollias S, Stoll G, Meier D, Valavanis A, and Boesiger P (1998). A new correlation-based fuzzy logic clustering algorithm for fMRI, *Magnetic Resonance in Medicine*, **40**, 249–260.

Haben S, Singleton C, and Grindrod P (2015). Analysis and clustering of residential customers energy behavioral demand using smart meter data, *IEEE Transactions on Smart Grid*, **7**, 136–144.

Kalpakis K, Gada D, and Puttagunta V (2001). Distance measures for effective clustering of ARIMA time-series. In *Proceedings 2001 IEEE International Conference on Data Mining*, 273–280.

Keogh E, Lonardi S, and Ratanamahatana CA (2004). Towards parameter-free data mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 206–215.

Keogh E, Lonardi S, Ratanamahatana CA, Wei L, Lee SH, and Handley J (2007). Compression-based data mining of sequential data, *Data Mining and Knowledge Discovery*, **14**, 99–129.

Li M, Chen X, Li X, Ma B, and Vitányi PM (2004). The similarity metric, *IEEE Transactions on Information Theory*, **50**, 3250–3264.

Liao TW (2005). Clustering of time series data—a survey, *Pattern Recognition*, **38**, 1857–1874.

Maharaj EA (1996). A significance test for classifying ARMA models, *Journal of Statistical Computation and Simulation*, **54**, 305–331.

Maharaj EA (2000). Cluster of time series, *Journal of Classification*, **17**, 297–314.

Montero P and Vilar JA (2014). TSclust: An R package for time series clustering, *Journal of Statistical Software*, **62**, 1–43.

Moritz S and Bartz-Beielstein T (2017). imputeTS: time series missing value imputation in R, *The R Journal*, **9**, 207–218.

Piccolo D (1990). A distance measure for classifying ARIMA models, *Journal of Time Series Analysis*, **11**, 153–164.

Rousseeuw PJ (1987). Silhouettes: graphical aid to the interpretation and validation of cluster analysis, *Journal of Computation and Applied Mathematics*, **20**, 53–65.

Serban N and Wasserman L (2005). CATS: clustering after transformation and smoothing, *Journal of American Statistical Association*, **471**, 990–999.

Stineman RW (1980). A consistently well-behaved method for interpolation, *Creative Computing*, **6**, 54–57.

Tsekouras GJ, Hatziargyriou ND, and Dialynas EN (2007). Two-stage pattern recognition of load curves for classification of electricity customers, *IEEE Transactions on Power Systems*, **22**, 1120–1128.

Wang X, Smith K, and Hyndman R (2006). Characteristic-based clustering for time series data, *Data Mining and Knowledge Discovery*, **13**, 335–364.

Xiong Y and Yeung DY (2004). Time series clustering with ARMA mixtures, *Pattern Recognition*, **37**, 1675–1689.

*Received July 14, 2020; Revised September 8, 2020; Accepted September 26, 2020*