Mining Load Profile Patterns for Australian Electricity Consumers

Vanh Khuyen Nguyen[1](✉), Wei Emma Zhang[1], Quan Z. Sheng[1], and Jason Merefield[2]

[1] Department of Computing, Macquarie University, Sydney, Australia
thi-vanh-khuyen.nguyen@students.mq.edu.au,
{w.zhang,michael.sheng}@mq.edu.au
[2] Mojo Power Company, Sydney, Australia
jmerefield@mojopower.com
https://www.mojopower.com.au

Abstract. The transformation from centralized and fossil-based electricity generation to distributed and renewable energy sources is an inevitable trend in the energy industry. One of the prime challenges in this transformation is the task of load/battery management, especially at the residential level. In solving this task, it is critical that a good strategy for analyzing and grouping residential electricity consumption patterns is in place so that further optimization strategies can be devised for different groups of consumers. Based on the real data from an Australian electricity retailer, we propose a clustering process to determine typical customer load profiles. It can be served as a standard framework for dealing with real-world unsupervised problems. In addition, some statistical techniques, including cumulative sum and calculation of the most frequent value in dataset by using *mode*, are integrated into our data preprocessing and analysis. CUSUM chart is a graphical method to clearly visualize as well as detect changes in time-series data and then using *mode* values is to replace missing values in the dataset. Furthermore, in our framework, more practical Elbow method is conducted to determine appropriated number of clusters for k-centers algorithm. We then apply multiple state-of-the-art clustering methods for time series data and benchmark their respective performance. We found that k-centers clustering techniques produces better results compared to exemplar-based methods. Additionally, choosing appropriated number of clusters for *k-means* can improve performance of clustering model. For example, *k-means++* with $k = 2$ has significantly outperformed other methods in our experiment.

Keywords: Time series clustering · Residential electricity consumption · Data mining

1 Introduction

According to Australian Energy Market Operator (AEMO), renewable energy resources, especially residential battery storage, have been significantly

G. Cong et al. (Eds.): ADMA 2017, LNAI 10604, pp. 781–793, 2017.
https://doi.org/10.1007/978-3-319-69179-4_55

increasing because of diverse driving factors such as improved technology in solar battery with affordable prices, increase in retail electricity prices, and environmental impacts [1]. Taking these opportunities, electricity retailers have attempted to build a new business model to minimize electricity costs for both industry and individual household [1]. Recently, there is significant development of micro solar management system for households, including smart meters and applications of demand-side management. Consequently, it is essential to keep track electricity consumption in order to maintain balance between demand side and supply side [1,3]. More importantly, segmenting consumer load profiles into separated groups has received more attention in research area [10]. Hino et al. also stated that determining energy consumption behaviors is useful for selecting and designing optimal tariff for different consumers groups [10].

In recent years, time-series clustering has received strong interest [8,14,23] due to its effectiveness for discovering data in many real-world applications [11], especially for clustering energy consumption patterns [10]. There are different types of clustering algorithms and some statistical techniques applied to determine typical load patterns as well as measure similarities between them [3,10,12]. For instance, the work of [12] segmented domestic electricity load profiles based on applying the most common methods included *k-means*, *k-medoid*, and Self Organizing Maps (SOM) whereas the study also carried out comparisons to investigate which algorithm is outperforming. On the other hand, Hino et al. proposed Gaussian mixture model for data representation before conducting hierarchical algorithm to cluster energy load profile patterns [10]. Similarly, Zakaria et al. [23] established another approach based on subsequent time-series clustering approach [11] to resolve unsupervised problems in real-world time series data.

However, most of the existing techniques focus more on technical or theoretical issues rather than practical aspect to resolve our realistic problems in real-world time series data. Yet, despite good performance of some complex algorithms, simple methods with less time consumption and cost effectiveness might be more invaluable and constructive in real applications of small business. Besides, how to combine all potential techniques going through various processes to obtain expected results for business purposes that is more challenging. Therefore, we propose a practical process to serve as a standard framework for resolving clustering problems in industries, especially for electricity industry. First of all, we utilize statistical techniques to analyze and visualize data to demonstrate the data characteristics. Based on our analysis, we found that missing or unknown values occur and they are often shown as `NaN` in the datasets or interruptions in time series chart. We then replace `NaN` with rational values by adopting *mode* method in statistics that indicates the highest frequent value in the data. After that, we can perform clustering on the user's load profiles by applying k-centroids approach. However, determining appropriated number of clusters in k-centers algorithms is typically challenging [3,22]. Therefore, we suggest to utilize a graphical and practical method, called Elbow method [5]. This technique is used by iterating *k-means* with the range of number of clusters

k from 1 to 10 and it effectively works for small range of number of clusters. Based on the Elbow chart, we can identify appropriated clusters number for our *k-means* model. In our work, we will consider original *k-means* and one of its variations (*k-means++*), and exemplar-based method (*affinity propagation*). As ground truth labels of data are often unknown in real world, Calinski-Harabaz and Silhouettes metrics are used for modeling evaluation. The more higher values of these metrics, the better quality of the clustering model.

There are significant findings in our experiment to prove that *k-means++* is outperforming compared to original *k-means* and *affinity propagation*, while the suitable selection of $k = 2$ also improves the performance of our approach. There are three main contributions in our work:

- Introducing a set of practical steps for clustering problems based on real-world dataset. It can be demonstrated as a standard framework for all steps involved in mining electricity load profile patterns.
- Suggesting more practical techniques in statistics that have been effectively applied in data preprocessing and data demonstration.
- Providing significant evidences on how different clustering approaches and methods of predefining number of clusters could impact on performance of clustering models.

The rest of the paper is organized as follows: Sect. 2 summarizes some studies related to time-series clustering problems. Then, Sect. 3 provides the fundamental theories applied in our research. In Sect. 4, we explain in more details of our experimental process as well as evaluating the application of clustering models. Finally, Sect. 5 concludes our study.

2 Related Work

Recently, the development of smart meters technology has provided the opportunity of collecting and storing consumer electricity load profiles in the form of time-series data [10]. Accordingly, there are many existing studies proposed to explore typical patterns of energy consumption of households.

One approach is to predefine typical load profiles (*TLPs*) for each group; and then cluster a specific consumer to a particular group by measuring and comparing individual consumer's load patterns with the predefined *TLPs*, namely pattern-recognition methods [9]. For instance, the research project of [9] applied FCM clustering algorithm in order to identify *TLPs* of each class and group consumers with similar load curves together. However, that approach might cause expensive costs and high time consumption due to *TLPs*-determination.

Very recently, it is claimed that subsequence time-series techniques has gained more interests in data mining area, namely *shapelets* [11]. Specifically, Zakaria et al. proposed a method based on shapelet-based time series classification to resolve unsupervised problems in real world [23]. In this study, it showed that subsequence time-series clustering technique does not only deal with unequal time series in length, but also improve accuracy of clustering results [23].

Our approach adopts similar ideas based on distances measurement to analyze the similarities between time series sequences, that has been widely used in both practice and literature, including *k-means*, *fuzzy k-means*, and hierarchical clustering [6,10]. Those studies aim to investigate and identify consumer behaviors in using electricity and group them into similar classes. In the study of [10], they introduced the method for daily consumption data based on Gaussian Mixture Model and then applied hierarchical clustering to specify typical patterns of consumption. However, performing distance measurement method, including Euclidean or Dynamic Time Warping, might become difficult when there exists missing values or unequal time-series lengths [23]. There are some improved versions in *k-means* algorithm that has been established. For example, Wagstaff proposed improved version of *k-means* with soft constraints (KSC) to handle missing values in datasets without using any imputation techniques [21]; in addition, Mesquita added one more soft constraint into KSC to deal with imputed values [14].

3 The Methodology

The current clustering approaches has focused on improving existing clustering algorithms and then evaluating their modeling by using some common and public datasets. However, there are few studies proposed the completely clustering process to resolve the realistic problems based on real-world data collection. Our approach, which is illustrated in Fig. 1, serves as a framework for practical applications in industrial and scientific fields.

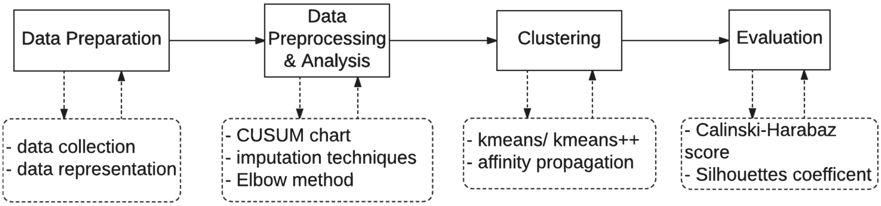


Fig. 1. The main clustering process steps

3.1 Data Preparation

In this work, we use the real data from an Australian electricity provider that contains 280 consumers records along with their total energy loads in 30-minute interval between September 2015 and October 2016. The data collection contains approximately 1% unknown values that produced automatically by some errors in system. Those values moreover are meaningless for our investigation, thereby removing them from our datasets.

Following the research of [10], we represent our data collection as a daily consumption data and then transform it to time series matrix $Q_{m \times n}$ defined as follows:

$$Q_{m\times n} = \begin{pmatrix} v_{1,1} & v_{1,2} & \cdots & v_{i,j} & \cdots & v_{1,n} \\ v_{2,1} & v_{2,2} & \cdots & v_{i,j} & \cdots & v_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ v_{m,1} & v_{m,2} & \cdots & v_{i,j} & \cdots & v_{m,n} \end{pmatrix}$$

where: $v_{i,j}$ is the daily consumption of a customer c_i at the time interval t_j; $i \in C$ and $j \in T$ with $C = \{c_1, c_2, \cdots, c_m\}$ is the list of account number and $T = \{t_1, t_2, \cdots, t_n\}$ is the list of time series by date, respectively.

Please note that the matrix might contain some NaN value as each record sequence does not often have the same length in real-world data [17]. Missing values is one of the most obstacles in data modeling since most of existing clustering algorithms do not allow any NaN in data inputs [21,23]. In our proposed approach, some imputation methods are applied to handle with this problem in the following sections.

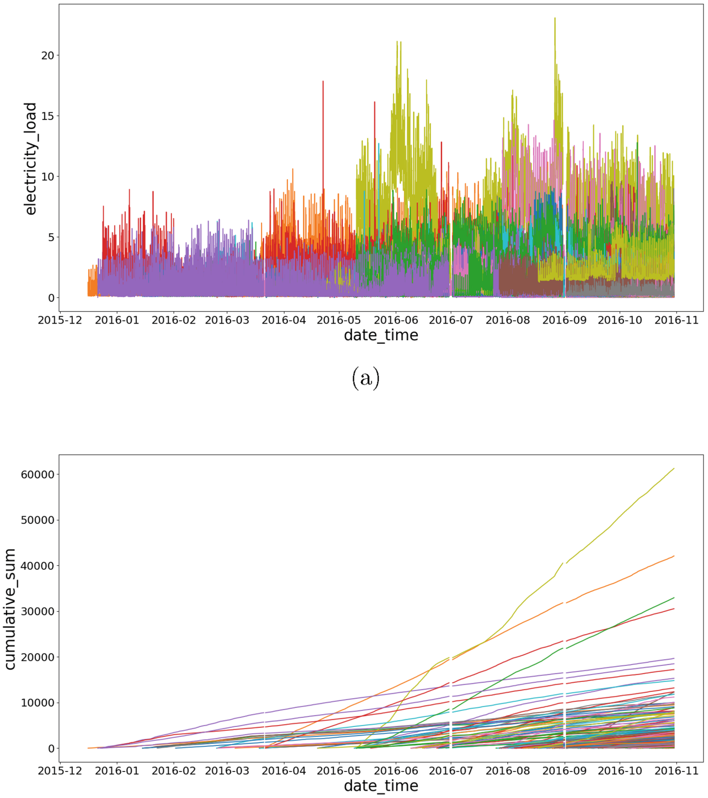
3.2 Data Preprocessing and Analysis

Observing and analyzing data is an important step in order to have an overview and understanding data structure. In our clustering framework, we suggest to leverage *CUSUM* technique for detecting changes in time series data. This is one of the most popular statistical tool applied in various fields such as manufacturing process, signal anomaly detection in control system, and others [13].

Cumulative sum is defined as a sequence of partial sums of a given sequence $\{a_k\}_{k=1}^n$. The partial sum of the first N terms of the given sequence is defined by $S_N = \sum_{k=1}^N a_k$. For example, the cumulative sums of the sequence $\{a_1, a_2, a_3, ...\}$ are $\{a_1, a_1 + a_2, a_1 + a_2 + a_3, ...\}$. However, we excluded NaN values when implementing CUSUM chart so that we can effectively detect missing values. As seen in Fig. 2(b), it shows more clearly interruptions caused by missing values compared to time-series chart without using cumulative sum in Fig. 2(a). We then interpolated those missing values by computing the mean of the values before and after the NaN values. Furthermore, electricity data is consistently organized by time order; thereby using this method being more reasonable for missing values interpolation. The result is obtained as seen in Fig. 3(a).

However, another challenge in our existing data collection is unequal time series instances in length as seen in Fig. 3(a). In order to handle this issue, we proposed less costly imputation technique to replace those missing values by the values with the highest frequency in the specific time-series instances. As a result, dataset has obtained the same length (Fig. 3(b)) and that is well-prepared for our clustering models.

The next obstacle is to identify suitable number of clusters. In this case, the common Elbow method is applied to find the appropriated clusters for our model. In Fig. 4, sum of distances is significantly dropped down at k = 2 and it continues decreasing at k = 3 and k = 4. However, it is quite steady when $k > 4$. Therefore, the potential number of clusters k is the range between 2 and 4. In this work, we also discover how different k clusters chosen may affect on the quality of clustering model that will discuss further in the next section.



(a)

(b)

Fig. 2. Real-world data collection plotting

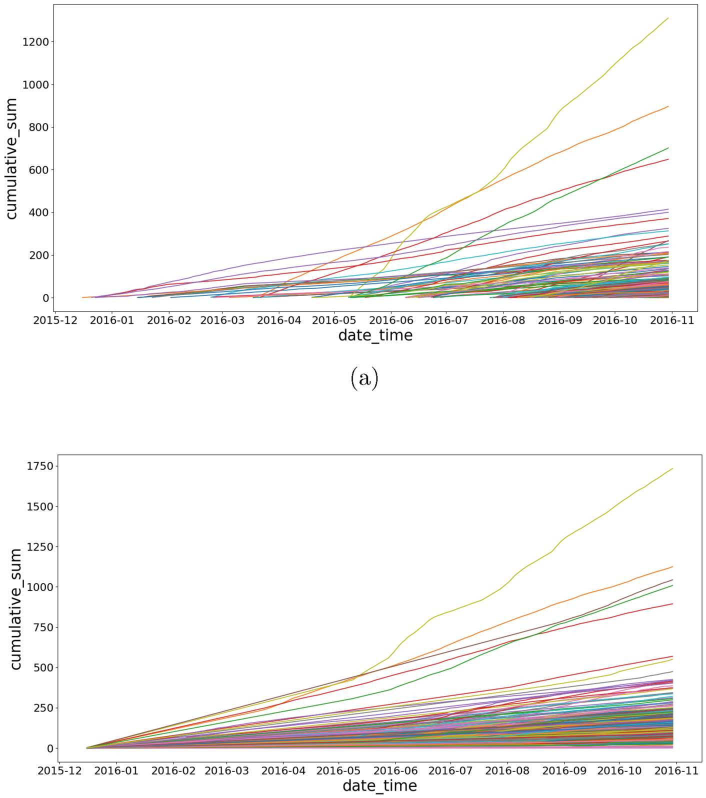
3.3 Clustering

In the literature, many existing or on-going developed clustering methods have been proposed in past decade [17]. In our application, we apply two different approaches, including *exemplar-based clustering* and *k-centers clustering* methods, which will be described in more detail in this section.

Clustering with K-means and K-means++. The classical *k-means* is one of the most well-known techniques in dealing with real-world clustering problems [4]. In this *k-means* approach, its simple idea is to select k clusters to minimize the total squared distance between each data point and its nearest centroids, defined by [4]:

$$\phi = \sum_{x \in X} min_{c \in C} (||x - c||)^2 \tag{1}$$

Another reason of utilizing *k-means* in this project is because of its less expensive computational cost and ease of use for our real-world problem [4].



(a)

(b)

Fig. 3. CUSUM chart after interpolating missing values (a) and applying imputation technique (b)

However, the Algorithm 1 proposed by [4], we can not guarantee selected centers distributed optimally within data since those centers are chosen randomly. Furthermore, size of each cluster might not be equal in real-world data; thereby, leading to the largest clusters being dominated. Accordingly, Arthur and Vassilvitskii proposed improved version called *k-means++* in order to prevent these limitations in classical *k-means* [4]. The principal ideas of *k-means++* is to optimize the chosen centroids in k-means algorithm by calculating "D^2 weighting", given by: $P = \frac{D(x)^2}{\sum_{x \in X} D(x)^2}$. Here, $D(x)$ indicates the shortest distance from a specific data point to the nearest centroid that has been selected in prior step. Unlike random centroids selection in original *k-means*, new centroids in *k-means++* will be taken by measuring the probability P as above.

Clustering with Affinity Propagation. Unlike random initialization of centers selection in k-centers clustering techniques, all data points can be considered

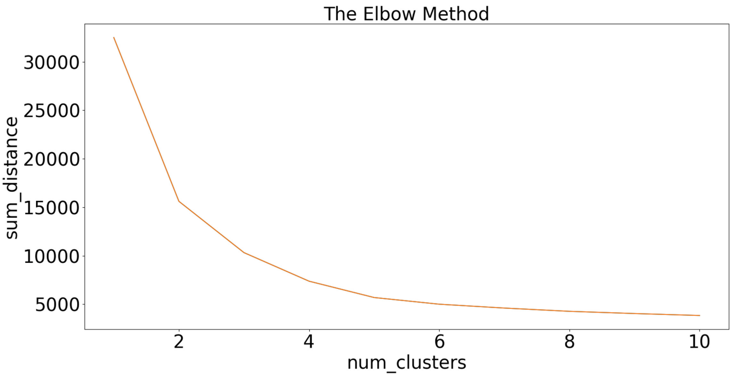


Fig. 4. Elbow chart for determining the number of clusters in *k-means*

Algorithm 1. k-means Algorithm

```
1: Choose randomly an initial k centers $C = \{c_1, c_2, ..., c_k\}$
2: For each $i \in \{1...k\}$, assign $C_i$ to $x \in X$ if $D_{x \in X, c_i} < D_{x \in X, c_j}$, $\forall i \neq j$, $D_{x,c}$ is
       distance between data point $x_i$ to centroid $c_i$
3: For each $i \in \{1...k\}$, recalculate and set new centroid for each group $c_i = \frac{1}{c_i} \sum_{x \in C_i} x$
4: Repeat (2) and (3) until convergence.
```

as potential centers in *affinity propagation (AP)* [7]. Hence, AP has received more considerable attention recently [22]. Let $X = \{x_1, x_2, ..., x_n\}$ be a set of data points and using $s(x_i, x_j)$ function is to measure similarity between two data points. The goal of *affinity propagation* technique is to minimize the negative squared distance [7] between x_i and x_j, given by:

$$s(i, j) = -||x_i - x_j||^2 \tag{2}$$

In other words, $s(i, j)$ is used to indicate how well-suited data point j can be an exemplar of data point i. An exemplar is defined as a center selected from actual data points. Then, AP take $s(i, i)$ as an input preference to determine how likely a particular input can be chosen as an exemplar [7].

The main process of AP algorithm is to exchange messages between two data points that belong to two different categories [7,22]. Firstly, the responsibility $r(i, k)$, which is accumulated evidence that indicates how well data point k should likely serve as exemplar of data point i. Secondly, the availability $a(i, k)$, which is accumulated evidence that reflects how appropriated data point i should take data point k as its exemplar.

4 Experimental Evaluations

The goal of this section is to express further details of our experimental implementation on the real-world data as well as evaluate performance of chosen clustering models. This section will first explain setting of our experiment and will then demonstrate metrics for model evaluation; finally, it will show results and comparisons between clustering models.

4.1 Experiment Setting

Recently, Python is a programming language in computer science which has been widely used in data mining and data analysis due to its clear and simple syntax. Moreover, it has gradually nominated in scientific field since a huge amount of extension libraries and packages developed for machine learning such as PyBrain [19], mlpy [2], and scikit-learn [16]. Taking advantages of scikit-learn library and pandas package[1], we effectively performed the clustering modeling to resolve our real-world problems in the electricity industry.

4.2 Evaluation Metrics

The ground truth labels in our dataset are unknown; thereby Calinski-Harabaz and Silhouettes metrics proposed to evaluate how well our models perform.

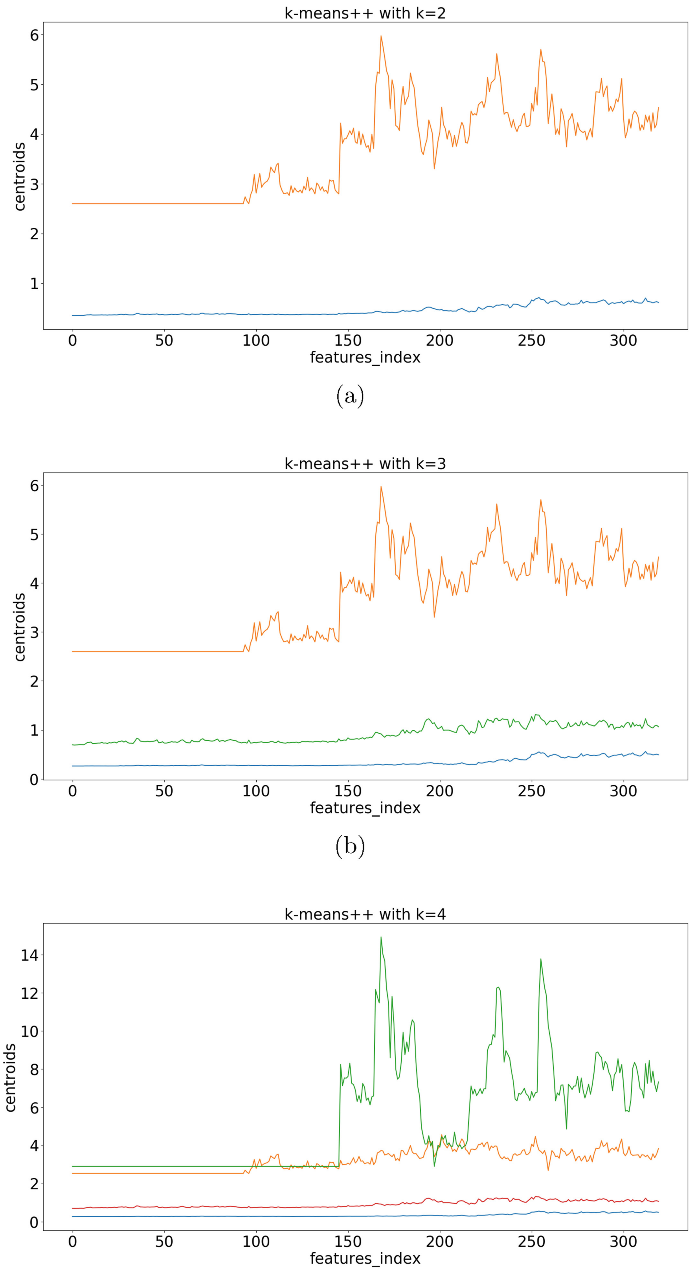
The Calinski-Harabaz score indicates how better clusters are defined by calculating ratio of within-cluster dispersion and between-cluster dispersion [18]. The higher Calinski-Harabaz score defines the better clustering model that the clusters are well separated.

Similarly, Silhouettes coefficient is conducted for dataset with unknown ground truth labels to evaluate model performance. The Silhouettes score s is simply computed [18] by $s = \frac{b-a}{max(a,b)}$, where: a is mean distance of a specific data point and other data points in same cluster, b is mean distance of a specific data point and other data points in neighboring clusters. Obviously, Silhouettes coefficient must belong to range $[-1, 1]$. When the score s is closer to 1, it means the clusters are well defined; otherwise, the clustering is incorrect. When the score s however equals to zero, it indicates that clusters might be overlapping [18].

Table 1. Clustering results of real world electricity data set.

Technique	k	Time(s)	Calinski-Harabaz(CH)	Silhouettes(S)
k-means++	$k = 4$	0.06	312.475	0.490
k-means++	$k = 3$	0.03	296.648	0.487
k-means++	$k = 2$	0.02	299.715	0.858
k-means	$k = 4$	0.07	253.400	0.356
MiniBatchKMeans	$k = 4$	0.02	250.855	0.317
Affinity propagation	*Automatic*	0.05	133.042	0.145

[1] https://pypi.python.org/pypi/pandas/.



(a)

(b)

(c)

Fig. 5. Plots of *k-means++* results (Color figure online)

4.3 Results and Comparisons

In Table 1, it shows that the outperforming algorithm is *k-means++* with k = 2. In fact, its computing speed is shorter than others while the Silhouettes score is also double higher than other methods. In general, *k-means++* with $k = 2$ gains better results even though its Calinski-Harabaz score is lower approximately 13 points than *k-means++* with $k = 4$.

On the other hand, *Affinity Propagation* estimated automatically the number of clusters k shows lowest performance with lowest scores in Calinski-Harabaz and Silhouettes. The classical *k-means* by choosing random centers furthermore has lower score of Calinski-Harabaz (253.400) and Silhouettes (0.356) compared to the average scores of *k-means++* ($\bar{CH} = 302.946$ and $\bar{S} = 0.612$). In this study, we also attempt to test the computation time between *k-means++* and *Mini-batch k-means* proposed by [20]. As shown in Table 1, by choosing the same $k = 4$ for both *k-means++* and *Mini-batch k-means*, the computation time of *Mini-batch k-means* performed better than *k-means++*. However, when comparing between *k-means++* with $k = 2$ and *Mini-batch k-means*, both of them have the same computation time $t = 0.02s$ (Table 1).

Furthermore, by plotting centroids set of each cluster by time-series features, we can visibly recognize that the major difference in distance and pattern between clusters. As shown in Fig. 5, the two clusters in 5(a) are well separated with the average of centroids in the first cluster (orange line) that is double higher than another cluster (blue line). However, when choosing $k = 3$ in 5(b), the second and third clusters represented respectively by green line and blue line are very close in distance in spite of their slightly different patterns. On the other hand, 5(c) demonstrates the overlap between four clusters. It can be concluded that *k-means++* with $k = 2$ has provided outperforming results others in this study.

5 Discussion and Conclusion

The common challenge in time-series clustering problem is how to handle data with unequal length to improve the accuracy of the model [17]. Accordingly, there are many improved algorithms proposed in the literature such as *k-Shape* for shape-based clustering to replace classic clustering methods based on distance measurement [15], *k-means with soft constraint (KSC)* in [21] without requiring imputation for missing values, and *KSC-OI* algorithm proposed by [14] for the improvement of *KSC*. However, there is lack of practical proofs in these clustering algorithms.

Our current study might involve some limitations relevant to data bias due to imputation process for missing values and fill-in method for `NaN` values in data collection. However, the research project has been successfully implemented to segment consumption dataset provided by the Australian electricity retailer. The experiment has shown the significant performance of *k-means++* in time-computing cost and the quality of clustering groups compared to other methods like classical *k-means*, *Mini-batch k-means*, and *affinity propagation* for generating optimal number of clusters within itself. Furthermore, it has been proved that selecting the number of clusters k might also impact the quality of clusters. For example, applying *k-means++* with $k = 2$ obtained better results in our study. For our further research, we aim to design an appropriated algorithm to handle missing values as well as unequal time-series length based on current dataset in order to improve the accuracy of clustering model.

Acknowledgement. This study was funded by Capital Markets Cooperative Research Centre (CMCRC) (https://www.cmcrc.com) and supported for data collection by Mojo Power, Australia.

References

1. AEMO. Emerging Technologies Information Paper (2015)
2. Albanese, D., Visintainer, R., Merler, S., Riccadonna, S., Jurman, G., Furlanello, C.: mlpy: Machine Learning Python. CoRR (2012)
3. Anuar, N., Zakaria, Z.: Electricity load profile determination by using fuzzy C-means and probability neural network. Energ. Procedia **14**, 1861–1869 (2012)
4. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding (2007)
5. Bholowalia, P., Kumar, A.: EBK-means: a clustering technique based on elbow method and K-means in WSN. IJCA **105**(9), 17–24 (2014)
6. Chicco, G., Napoli, R., Piglione, F.: Comparisons among clustering techniques for electricity customer classification. IEEE Trans. Power Syst. **21**, 933–940 (2006)
7. Frey, B., Dueck, D.: Clustering by passing messages between data points. Science **315**(5814), 972–976 (2007). (Washington)
8. Fu, T.C.: A review on time series data mining. Eng. Appl. Artif. Intell. **24**(1), 164–181 (2011)
9. Gerbec, D., Gasperic, S., Smon, I., Gubina, F.: Allocation of the load profiles to consumers using probabilistic neural networks. IEEE Trans. Power Syst. **20**(2), 548–555 (2005)
10. Hino, H., Shen, H., Murata, N., Wakao, S., Hayashi, Y.: A versatile clustering method for electricity consumption pattern analysis in households. IEEE Trans. Smart Grid **4**(2), 1048–1057 (2013)
11. Hou, L., Kwok, J.T., Zurada, J.M.: Efficient learning of timeseries shapelets. In: Proceedings - 30th AAAI on Artificial Intelligence, pp. 1209–1215 (2016)
12. Mcloughlin, F., Duffy, A., Conlon, M.: A clustering approach to domestic electricity load profile characterisation using smart metering data. Appl. Energ. **141**, 190–199 (2015)
13. Mesnil, B., Petitgas, P.: Detection of changes in time-series of indicators using CUSUM control charts. Aquat. Living Res. **22**(2), 187–192 (2009)
14. Mesquita, D., Gomes, J., Rodrigues, L.: K-means for datasets with missing attributes: building soft constraints with observed and imputed values. In: 24th ESANN, pp. 27–29 (2016)
15. Paparrizos, J., Gravano, L.: k-shape. ACM SIGMOD Rec. **45**(1), 69–76 (2016)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)

17. Rani, S., Sikka, G., Liao, T.W.: Recent techniques of clustering of time series data: a survey. Pattern Recognit. **52**(15), 1–9 (2005)
18. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**(C), 53–65 (1987)
19. Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., Sehnke, F., Ruckstiess, T., Schmidhuber, J.: PyBrain. J. Mach. Learn. Res. **11**, 743–746 (2010)
20. Sculley, D.: Web-scale k-means clustering. In: Proceedings - 19th WWW, p. 1177 (2010)
21. Wagstaff, K.: Clustering with missing values: no imputation required. In: Banks, D., McMorris, F.R., Arabie, P., Gaul, W. (eds.) Classification, Clustering, and Data Mining Applications. Studies in Classification, Data Analysis, and Knowledge Organisation, pp. 649–658. Springer, Heidelberg (2004)
22. Wang, C.D., Lai, J.H., Suen, C.Y., Zhu, J.Y.: Multi-exemplar affinity propagation. IEEE Trans. Pattern Anal. Mach. Intell. **35**(9), 2223–2237 (2013)
23. Zakaria, J., Mueen, A., Keogh, E.: Clustering time series using unsupervised-shapelets. In: Proceedings - IEEE ICDM, pp. 785–794 (2012)