Detecting Hidden Energy Consumption Patterns

Suma Dodmani
Department of Computer Science
University of Colorado Boulder
suma.dodmani@colorado.edu

Toshiko Sesselmann
Department of Computer Science
University of Colorado Boulder
tose5851@colorado.edu

Abstract

In the wake of global climate change and increase in temperature over years, one significant area that has gained much of an attention is energy consumption at household level. There has been a significant amount of debate about the fact that making people realize their energy consumption patterns, helps them better understand their environmental footprint. Both producing and using electricity more efficiently reduces amount of fuel needed to generate electricity and the amount of greenhouse gases emitted. Thus, in this project we show that the patterns emerging from the energy consumption are different for every household and talk about methods to detect the energy consumption clusters by applying K-Means and Hierarchical clustering and topic modeling to differentiate the patterns of energy consumption from high end users to low end users.

***Keywords* —** Climate Change, Energy Consumption, Environmental Footprint, K-Means, Topic Modelling

1 Introduction

There are patterns in how people use electricity throughout the day. Depending on the energy usage patterns, utility companies can personalize the behavioral advice and motivate their customers to save energy.

Smart Meters can monitor electric usage in homes hourly or even more frequently. We will use the data of 5,567 London Households between November 2011 and February 2014 of half hourly and daily electricity consumption. (https://www.kaggle.com/jeanmidev/smart-meters-in-london/home)

Using the data above, we will use unsupervised clustering techniques to find the patterns in how people use electricity throughout the day.

2 Data Preparation

The dataset from kaggle as mentioned above contains the energy consumption readings for a sample of 5567 London households that took part in UK Power Networks led Low Carbon London Project. The data contains the block files with half hourly smart meter measurement.

We start with averaging energy usage at every half-hour of the day for each household, then calculate the proportion of usage at each half-hour. For the purpose of this project, we picked the same set of data from 500 households for all experiments and only used weekdays usage, since weekend usage patterns are most likely different compared to weekdays. There are 48-features (percentage at every half hour of the day) on each sample. Any samples with non-numbers or the average usage of 0 are removed from the data set. Plotting this data all at once, you get the Figure 1:

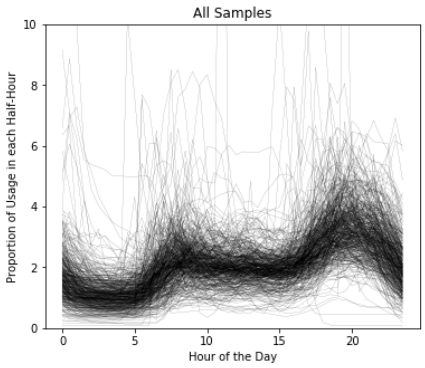

Figure 1: Load Curves from 500 Households

3 K-Means^{++} Clustering

K-Means^{++} Clustering is one of the unsupervised learning algorithms. We used KMeans library in Scikit-Learn to find similar patterns in the data set.

3.1 Algorithm

K-Means^{++} clustering algorithm is an improvement to the K-Means clustering algorithm that guarantees to find a solution that is O(log k) competitive to the optimal k-means solution. [4] Instead of picking initial centroids randomly in K-Means, K-Means^{++} picks a first centroid randomly, then the next point is picked based on a probability that depends on the distance to the first point. The further apart the point is the more probable it is. The process is repeated to pick K centroids. This introduces an overhead in the initialization of the algorithm, but solves the NP-hard K-Means problem.

In K-Means^{++} clustering, the number of clusters k needs to be predefined. We picked the number of clusters to be 4 clusters, as the result of analyzing Silhouette value, squared Euclidian distance and Calinski-Harabasz index mentioned in 3.2.1, 3.2.2, 3.2.3. The algorithm works as follows:
1. k centroids are initially picked by K-Means^{++} algorithm procedure
2. Take each sample and associate it to the nearest centroid.
3. Once all samples have been associated, re-calculate k new centroids as barycenters of the clusters resulting from the previous step.
4. Repeat Steps 2 and 3 until the centroids no longer move.

This algorithm aims to minimize a squared error function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left| x_i^{(j)} - c_j \right|^2$$

where $\left| x_i^{(j)} - c_j \right|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers. [3]

We applied the K-Means^{++} clustering technique to the data set to group into 4 clusters and the plot from each cluster is shown in Figure 2. We detected the outliers in Cluster 2.

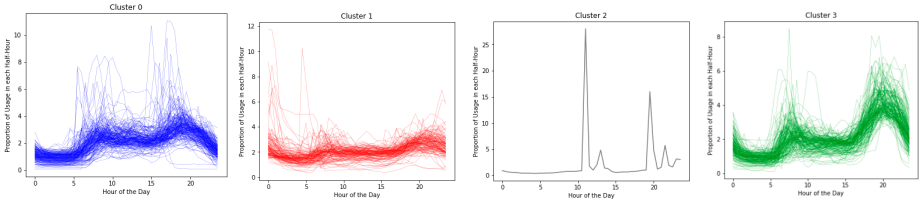

Figure 2: Samples Associated to each Cluster

Taking the mean from each cluster, the final centroid from each cluster is shown in Figure 3.

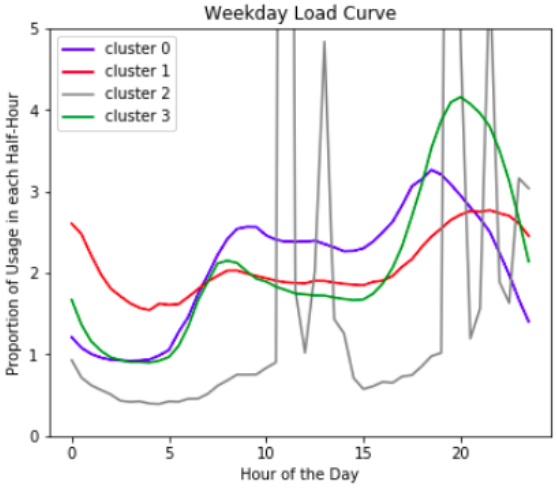

Figure 3: KMean Cluster Centers

There are three distinct energy use patterns throughout the course of the day, excluding outliers Cluster 2. Cluster 0 has two peaks; one peak in the morning and another peak in the evening. Cluster 1 has flat usage throughout the day. Cluster 3 has a sharp peak in the evening.

3.2 Evaluating Quality of Clusters

3.2.1 Silhouette Value

In order to use K-Means^{++} Clustering algorithm, the desired number of clusters needs to be known beforehand. One of the measures used to determine the optimal number of cluster is Silhouette Coefficient to study the separation distance between the resulting clusters. The Silhouette Coefficient for a sample s(i) can be calculated as

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

where for each datum i, a(i) is the average distance between i and all other data within the same cluster and b(i) be the smallest average distance of i to all points in any other cluster, of which i is not a member. [6]

This measure has a range of [-1, 1]. Silhouette coefficients near +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters and negative values indicate that those samples might have been assigned to the wrong cluster. [5]

The mean Silhouette Coefficient over all samples is plotted in Figure 4 for the number of 2-29 clusters, k. Observing just the number of clusters 2 to 7, the silhouette plot shows that 3 or 4 would be a good pick whose value is above average silhouette scores. 4 is a slightly better choice.

3.2.2 Total within Sum of Squares

Another criteria to evaluate the quality of clusters is using a basic Euclidean distance metric. The sum of square distance between each data point and its respective cluster centers for all samples is plotted in Figure 5 for the number of cluster, k, from 2 clusters to 29 clusters. The drop in distance at k=4 is observed and when k is equals or greater than 5, the distance continues to increase. This is another indication that k=4 is a good choice for this given data set.

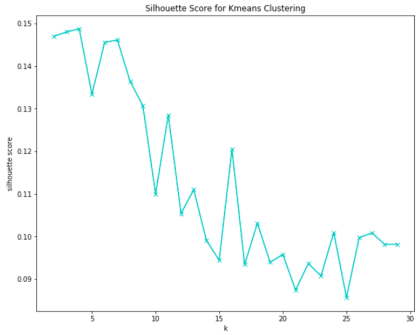

Figure 4: Mean Silhouette Coefficient

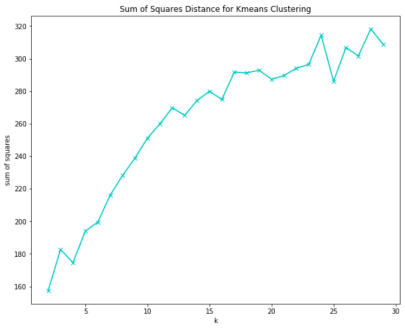

Figure 5: Euclidean distance

3.2.3 Calinski Harabasz Index

Another method that is based on the concept of dense and well-separated clusters is the Calinski-Harabasz index. The math formula to the measure is:

$$\frac{SS_B}{SS_W} \times \frac{N-k}{k-1}$$

Where k is the number of clusters, and N is the total number of observations (data points), SS_W is the overall within-cluster variance (equivalent to the total within sum of squares calculated above), SS_B is the overall between-cluster variance. A big SS_B value means that the centroid of each cluster will be spread out and they are not too close to each other. The Calinski-Harabasz Index, the ratio of $\frac{SS_B}{SS_W}$, should be the biggest that at the optimal clustering size. [7]

The Calinski-Harabasz Index plot below for this data set does not show the biggest value at k=4, but biggest at cluster 2. Knowing that all measures will not able to suggest the optimal number of clusters, we will determine the reasonable cluster size to be 4 using other methods.

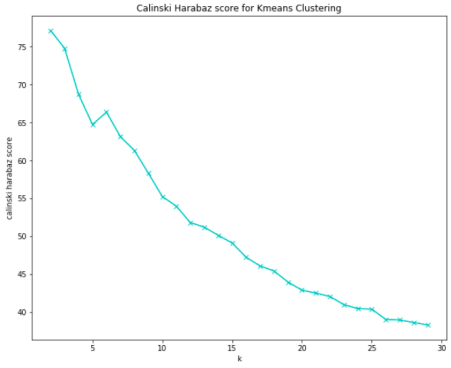

Figure 6: Calinski Harabasz

4 Hierarchical Clustering

Hierarchical Clustering is another unsupervised learning algorithm. The key difference with K-Means^{++} Clustering is that you do not have to pre-determine the number of clusters. We used hierarchical clustering library in Scikit-Learn.

4.1 Algorithm

In agglomerative Hierarchical Clustering, data points are clustered using a bottom-up approach starting with individual data points. The algorithm works as follows: [8]
1. At the start, treat each data point as one cluster. Therefore, the number of clusters at the start will

be K, while K is an integer representing the number of data points.
2. Form a cluster by joining the two closest data points resulting in K-1 clusters.
3. Form more clusters by joining the two closest clusters resulting in K-2 clusters.
4. Repeat the above three steps until one big cluster is formed.

In this experiment, Euclidean distance was used to find the closest points.

4.2 Dendrogram

The resulting cluster tree (a dendrogram) from the algorithm to represent our data is shown in Figure 7.

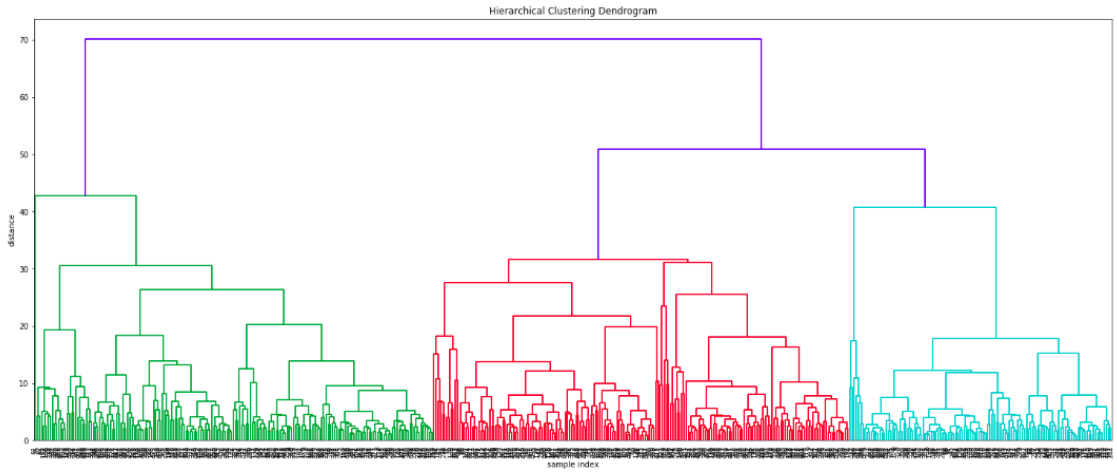

Figure 7: Dendrograms

The greater the difference in height in the dendrogram, the more dissimilarity. From the dendrogram, any k of 3, 4 or 5 should produce distinct load curves with enough distance. With k=4, taking the mean from each cluster, it produced similar load curves in Figure 8 as K-Means^{++} Clustering.

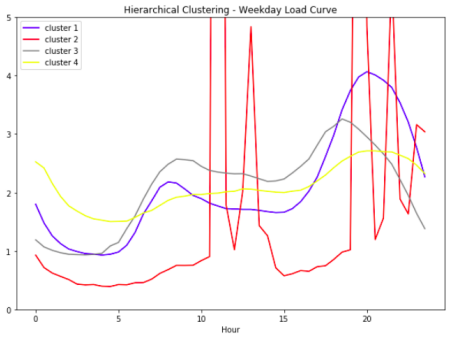

Figure 8: Hierarchical Cluster Means

The biggest advantage with Hierarchical Clustering is that you do not have to predetermine the number of clusters and you can visually make a decision using dendrograms. The result of Hierarchical and K-Means^{++} clustering are very similar with our data.

5 Topic Modeling

Topic Modeling is a generative statistical model predominantly used to identify hidden topics in text documents. It is known to automatically identify topics present in a text object and derive hidden

patterns exhibited by a text corpus. However, for the scope of this project, we try and explore the possible hidden features topic modeling techniques can extract by modeling the relations between energy variables.

There has been very little work regarding application of topic modeling in areas outside Natural Language Processing(NLP) domain. But, because clustering, dimension reduction, and HMMs aren't suitable for finding the relations between variables as stated in [16] we observe topic models help identify relation among the variables and thus extract the patterns.

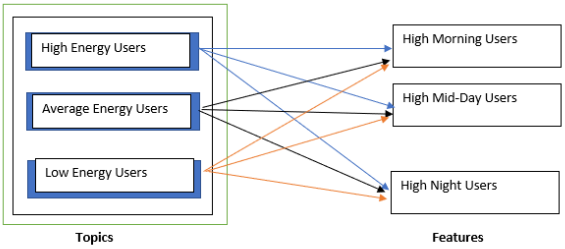

Figure 9: Inferred energy topic models and features considered

Figure 9 shows the way topics are modeled. Each of the topic has the same set of features but differ in percentile. This is because the usage of electricity in every household is done all through the day with variation only in the percentile of usage at different times. Although clustering techniques can cluster the users into categories, it is hard to identify the relations between variables.

5.1 LDA- Model

Latent Dirichlet allocation (LDA) is a generative model and belongs to the class of topic models.In this model, a topic is a discrete distribution over the data with probability vector ϕ_t [17]. We implement LDA for our topic modeling problem. Each of the energy topic is defined as a mixture of the features considered. The greatest challenge is to define the features which are all numeric values as topic models consider categorical data. We followed a similar approach as stated in [16] to overcome this problem. Each of the feature category is considered as bins which represent the distribution of values and energy consumption value of each household were categorized based on the values at different times of the day.

The results are based on two set of experiments:

- We apply topic modeling(LDA) on each block data separately to identify the patterns with respect to each block.
- LDA is applied on different kinds of users (high energy consumption, average consumption and low consumption) to identify what combination of features each type comprise of.

5.2 Results

1. The data comprised of 10 blocks each block had approximately 100 houses and their electricity consumption at different times of the day. Through LDA model we identified the probability mass assigned to high morning consumers, high mid-day consumers and high night consumers for each block.

   The result of this experiment shows that almost all the blocks were high mid-day users as the probability mass for these is high across all the blocks. There is great amount of variation with respect to the night energy users. This might correspond to the night owl cluster. However, the model assigned comparatively low probability mass to morning users.

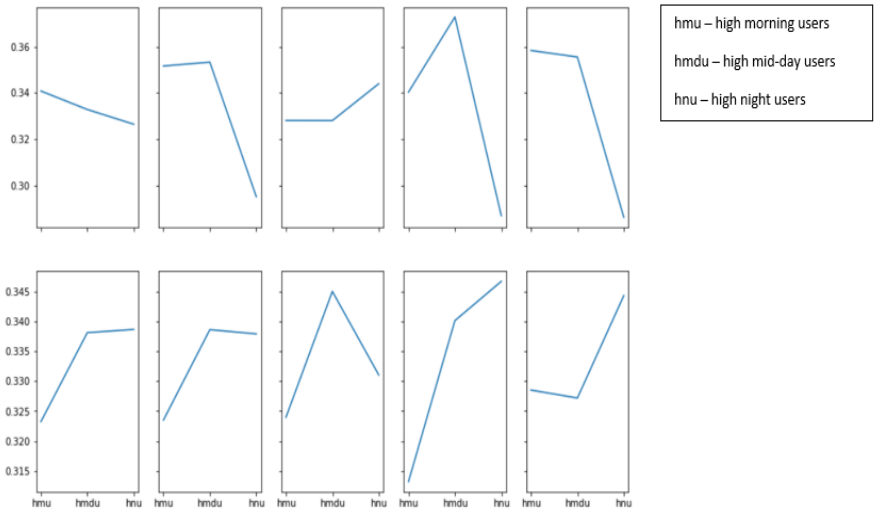

Figure 10: Usage pattern for each block

2. The second experiment was more insightful about the usage patterns of high energy consumers, average energy consumers and low energy consumers. We considered three features: high morning users, high mid-day users and high night users. For the average energy users, the model assigned more probability mass to high mid-day users. For Maximum energy users, it was opposite. The probability mass is high for high morning users and high night users. For the minimum energy users probability mass was more for the high night users and very less for high morning and mid-day users.

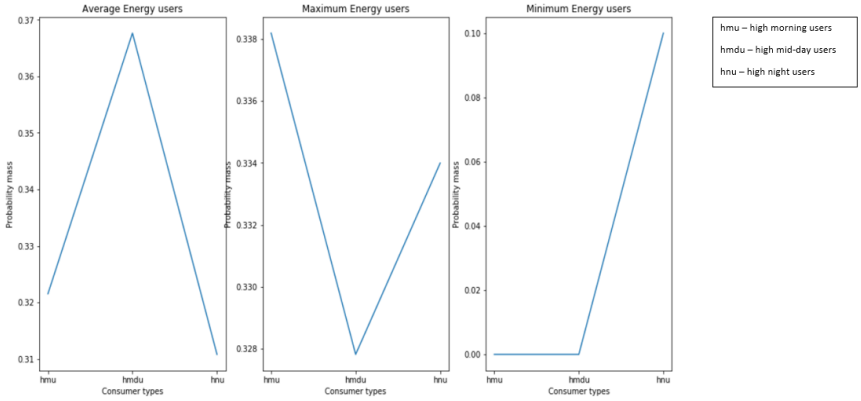

Figure 11: Usage patterns of different types of consumers

6 Classification using Dynamic Time Warping

Once clustering technique is applied and good load curves are obtained, we might have a need to classify a new set of samples to one of the clusters. For the time-series data like this one, just getting the L_2 norm, the Euclidean distance, is not be able to identify the load curve similarity. It often produces bad similarity measures when it encounters distortion in the time axis. [10] For example, two similar curves in Figure 12 are offset by 2 in x-axis. The Euclidean distance for the 2 curves are 7.48331477355, whereas the dynamic time warping distance is 0.

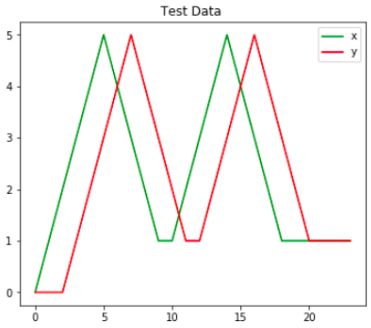

Figure 12: Two Similar Curves

This warping path can be found using dynamic programming to evaluate the following recurrence. $\gamma(i,j) = d(x_i, y_j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$ where d(i,j) is the distance found in the current cell, and $\gamma(i,j)$ is the cumulative distance of $d(i,j)$ and the minimum cumulative distances from the three adjacent cells. [13]

Using the dynamic time warping distance, we can use the k-NN algorithm for classification. Using only centroids data resulted from one of the unsupervised learning clustering techniques, 1-NN algorithm that uses dynamic time warping distance can find the closest load curve pattern. The Figure 13 shows the match between the test data and the closest cluster centroid using the algorithm.

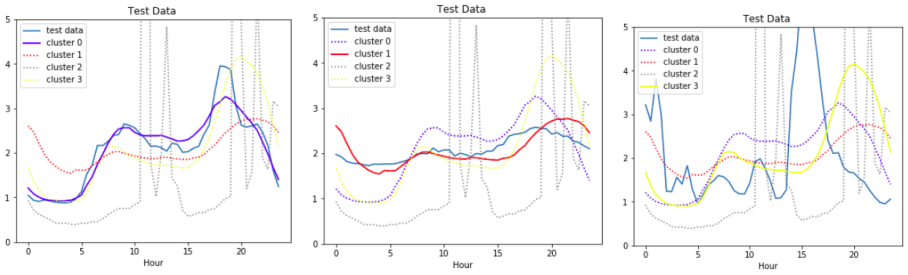

Figure 13: Finding a Similarity using 1-NN with Dynamic Time Warping Distance

7 Conclusion

Using the smart meter data and unsupervised clustering techniques, we were able to discover electricity consumption patterns and evaluated the quality of the clustering results. We can classify new sets of data into one of the identified patterns, using dynamic time warping distance.

While KMeans and Hierarchical clustering techniques helped us identify the different categories of consumers, through topic modeling technique we identified the pattern of usage for different type of consumers.

Understanding their pattern of usage helps the consumers know their contribution to the global environment footprint. Also, utility companies can make use of energy consumption behaviors of consumers. With Time Of Use pricing (TOU), consumers are charged more during the peak energy hours and the knowledge of their energy usage patterns can help to design time-variant prices in order to maximize profit. [14]

Utility companies can also target customers for various Energy Efficiency and Demand Response programs. Energy Efficiency reduces kilowatt-hours used, while Demand Response reduces kilowatts of demand during peak hours of the day. Demand Response programs often include rewards or penalties to encourage behavior change. Customers who save energy during peak hours may receive credit, or may avoid getting rate increases that utilities may employ during a Demand Response event.[15] Customers with large relative peaks during peak hours benefit more from

Demand Response programs and utility companies can put their marketing money targeting just the group of people.

Many utility companies offer a variety of rebates to consumers for upgrading your home to energy-efficient appliances and equipment. Programmable thermostat Energy Efficiency program may give most benefit for people who are not home during the day, that translates to the consumption patterns with two peaks, one in the morning and one in the evening.

The knowledge of the consumer's energy consumption patterns can help both the consumers and utility producers to better understand the patterns of energy consumption, profit both parties and help analyze its effect on the global climatic conditions.

References

[1] Knowing your "energy personality" can save you a lot of money
https://www.washingtonpost.com/news/energy-environment/wp/2015/03/03/why-knowing-your-energy-personality-could-help-save-you-a-lot-of-money/?noredirect=on&utm_term=.01fb7a7fd1a8

[2] Seven reasons why utilities should be using Machine Learning
https://blogs.oracle.com/utilities/utilities-machine-learning

[3] A Tutorial on Clustering Algorithms
https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

[4] K-means++ Wikipedia
https://en.wikipedia.org/wiki/K-means%2B%2B

[5] Selecting the number of clusters with silhouette analysis on KMeans clustering
https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html#sphx-glr-auto-examples-cluster-plot-kmeans-silhouette-analysis-py

[6] Selecting the number of clusters with silhouette analysis on KMeans clustering
https://en.wikipedia.org/wiki/Silhouette_(clustering)

[7] Calinski-Harabasz Index and Boostrap Evaluation with Clustering Methods
http://ethen8181.github.io/machine-learning/clustering_old/clustering/clustering.html

[8] Theory of Hierarchical Clustering
https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/

[9] Shape matching with time series data
https://roamanalytics.com/2016/11/28/shape-matching-with-time-series-data/#Motivations-for-shape-matching

[10] Time Series Classification and Clustering with Python
http://alexminnaar.com/time-series-classification-and-clustering-with-python.html

[11] Programatically understanding dynamic time warping (DTW)
https://nipunbatra.github.io/blog/2014/dtw.html

[12] Wikipedia - Dynamic time warping
https://en.wikipedia.org/wiki/Dynamic_time_warping

[13] Everything you know about dynamic time warping is wrong
https://www.researchgate.net/publication/216301292_Everything_you_know_about_dynamic_time_warping_is_wrong

[14] Yi Wang, Qixin Chen, Chongqing Kang, Mingming Zhang, Ke Wang, and Yun Zha, *Load Profiling and Its Application to Demand Response: A Review*, Tsinghua Science and Technology, April 2015, 20(2): 117-129

[15] What is Demand Response?
https://simpleenergy.com/what-is-demand-response/

[16] Cheng Tang and Claire Monteleoni *Can Topic Modeling Shed Light on Climate Extremes?*, George Washington University, November 2015, Computing in Science and Engineering

[17] Wallach et al., *Evaluation Methods for Topic Models*, 26th International Conference on Machine Learning, Montreal, Canada, 2009.