Lappeenranta University of Technology
School of Engineering Science
Degree Programme in Computational Engineering and Technical Physics
Intelligent Computing

Juho-Pekka Koponen

ANALYSING ELECTRICITY CONSUMPTION DATA BY FAST SEARCH AND FIND OF DENSITY PEAKS

Bachelor's Thesis

Examiner:	Professor Lasse Lensu
Supervisors:	Doctor Xiaozhi Gao Associate Professor Arto Kaarna

Abstract

Lappeenranta University of Technology
School of Engineering Science
Degree Programme in Computational Engineering and Technical Physics
Intelligent Computing

Juho-Pekka Koponen

Analysing electricity consumption data by fast search and find of density peaks

Bachelor's Thesis

2017

33 pages, 15 figures, 1 table, and 1 appendix.

Examiner: Professor Lasse Lensu

Supervisors: Doctor Xiaozhi Gao
Associate Professor Arto Kaarna

Keywords: clustering, k-means, DBSCAN, electricity consumption, load profile

The aim of this thesis is to compare a clustering method called Clustering by fast search and find of density peaks (CFSFDP) (Rodriguez et al. 2014) to two traditional clustering methods in analysis electricity consumption. These traditional methods are k-means and Density-based clustering of applications with noise (DBSCAN). The profiles of the typical days can be described as cluster centroids. The methods were compared by the easiness of the selection of parameter values and the qualities of the resulting clusters.

Based on the comparison, it was more difficult to select the parameter values with CFSFDP than with the other methods with the used dataset. The resulting clusters of CFSFDP were more spread out than those of the other methods. Between k-means and DBSCAN the results were highly similar.

Tiivistelmä

Lappeenrannan teknillinen yliopisto
School of Engineering Science
Laskennallisen tekniikan koulutusohjelma
Älykäs laskenta

Juho-Pekka Koponen

Sähkönkulutusdatan analysointi nopean haun ja tiheyshuippujen avulla

Kandidaatintyö

2017

33 sivua, 15 kuvaa, 1 taulukko ja 1 liite.

Tarkastaja:	Professori Lasse Lensu
Ohjaajat:	Tohtori Xiaozhi Gao Apulaisprofessori Arto Kaarna

Hakusanat: klusterointi, k:n keskiarvo, DBSCAN, sähkönkulutus, kuormitusprofiili
Keywords: clustering, k-means, DBSCAN, electricity consumption, load profile

Tämän kandidaatintyön tavoitteena on vertailla Klusterointia nopean haun ja tiheyshuippujen avulla (CFSFDP) (Rodriguez et al. 2014) kahteen perinteiseen klusterointimenetelmään sähkönkulutuksen analysoinnissa. Perinteiset menetelmät, joihin CFSFDP:tä verrataan, ovat k:n keskiarvon klusterointimentelmä ja DBSCAN. Tyypillisten päivien kuormitusprofiileja voidaan kuvata klustereiden keskipisteinä. Menetelmiä vertailtiin parametriarvojen valitsemisen helppoudella sekä tuloksena saatujen klustereiden ominaisuuksilla.

Kyseisen datan kanssa CFSFDP:tä käytettäessä oli vaikeampi valita parametriarvot kuin muilla menetelmillä. Tuloksena olevat klusterit olivat levittyneempiä CFSFDP:llä kuin muilla menetelmillä. Tulokset olivat hyvin samanlaisia k:n keskiarvon ja DBSCAN:n välillä.

Contents

List of symbols

C	A set of cluster centroids.
$\mathbf{c}$	A cluster centroid.
d	A distance between two points.
d_c	A cutoff distance in CFSFDP.
J	An objective function.
k	The number of clusters in k-means clustering.
$minPts$	A parameter in DBSCAN representing the minimum number of points to form a dense region.
n	The number of data samples.
P	The average number of neighbors as portion of the whole data set.
S	A set containing separate subsets of samples for each cluster.
δ	The distance from point p to the nearest point with greater local density than that of p.
ε	A parameter in DBSCAN representing the minimum distance to consider two points as connected to each other.
κ	The curvature of a line.
$\boldsymbol{\mu}$	A mean of vectors.
ρ	The local density of a data point.
ρ_b	The highest density of the points in a border region of a cluster in CFSFDP.

List of abbreviations

CFSFDP	Clustering by fast search and find of density peaks.
DBSCAN	Density-based spatial clustering of applications with noise.
MSE	Mean squared error.
PCA	Principal component analysis.

1 Introduction

1.1 Background

The forecasting of electricity consumption is important in today's competitive electricity market. The forecasting can be made easier if we can classify the consumers into different segments according to their consumption habits. As the processing performance of computers and machine learning algorithms have developed, nowadays it is possible to automate the segmentation (Zhong et al. 2015). Automated customer segmentation helps the electricity distribution providers in determining dedicated tariffs for different classes of electricity customers (Chicco, Napoli, and Piglione 2003). Many kinds of clustering algorithms can be used in classifying the consumption habits. There is no universal agreement on which algorithm is the best for this problem, and there is plenty of research done on the subject around the world.

The methods suggested for the classification of load profiles include hierarchical classification in frequency domain (Zhong et al. 2015), self-organizing maps (Verdu et al. 2006, Chicco, Napoli, Piglione, et al. 2004), artificial neural networks (Varga et al. 2015, Gerbec et al. 2005), subspace clustering (Piao et al. 2014), fuzzy statistics (Li et al. 2007), mixture model clustering (Labeeuw et al. 2013, Haben et al. 2016, Coke et al. 2010), ant colony clustering (Chicco, Ionel, et al. 2013), support vector clustering (Chicco and Ilie 2009), Renyi entropy-based classification (Chicco and Akilimali 2010) as well as combinations of different methods (G. Tsekouras et al. 2008, G. J. Tsekouras et al. 2007).

1.2 Research objective and scope of the thesis

Clustering by fast search and find of density peaks (CFSFDP) is a new kind of clustering method proposed by Rodriguez et al. (2014). The method claims to find the number of clusters intuitively as well as spot outliers automatically. The object of this thesis is to compare the performance of this method to traditional clustering methods in finding typical daily load profiles of electricity consumers. The traditional methods used in comparison are k-means clustering and Density-based spatial clustering of applications with noise (DBSCAN).

1.3 Research methodology

The data to be used is anonymous hourly electricity consumption data gathered from the customers of a Finnish solar power system company. The different clustering methods will be applied to the data and the results will be compared by using Mean squared error (MSE) as an adequacy measure. Furthermore, the aspects that will be looked at are the detection of outliers and the difficulty of selecting the proper parameters for each method.

The data will be pre-processed so that each data point represents one day of one customer's electricity consumption. As the data is hourly, this will create a dataset of 24 dimensions. Because the clustering methods take a lot of time computing data with that many dimensions, the number of dimensions will be reduced by using Principal component analysis (PCA).

1.4 Structure of the thesis

In the following section, the theory behind the used clustering methods is explained. After that we go through the experiments and results for each clustering method. We do this by explaining the encountered problems and solutions to them, as well as the programs used. Finally we discuss the differences of the results and make proposals for further experiments.

2 Clustering methods

k-Means clustering and DBSCAN were selected as the methods to use in comparison because they both are rather simple and well-known clustering methods that are easy to implement. In addition to that, k-means has been succesfully used in the electricity consumer segmentation problem (G. J. Tsekouras et al. 2007), so it is a good baseline to compare new methods to. DBSCAN was chosen specifically because it is a density-based algorithm similarly to CFSFDP.

2.1 k-Means clustering

In k-means clustering, we determine a single parameter k, and the data set with n samples will be divided into k sets $S = \{S_1, S_2, \ldots, S_k\}$ with cluster centroids $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$ (Wu 2012). The aim of k-means clustering is to minimize the variance of each cluster. This gives us an objective function

$$J = \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2, \tag{1}$$

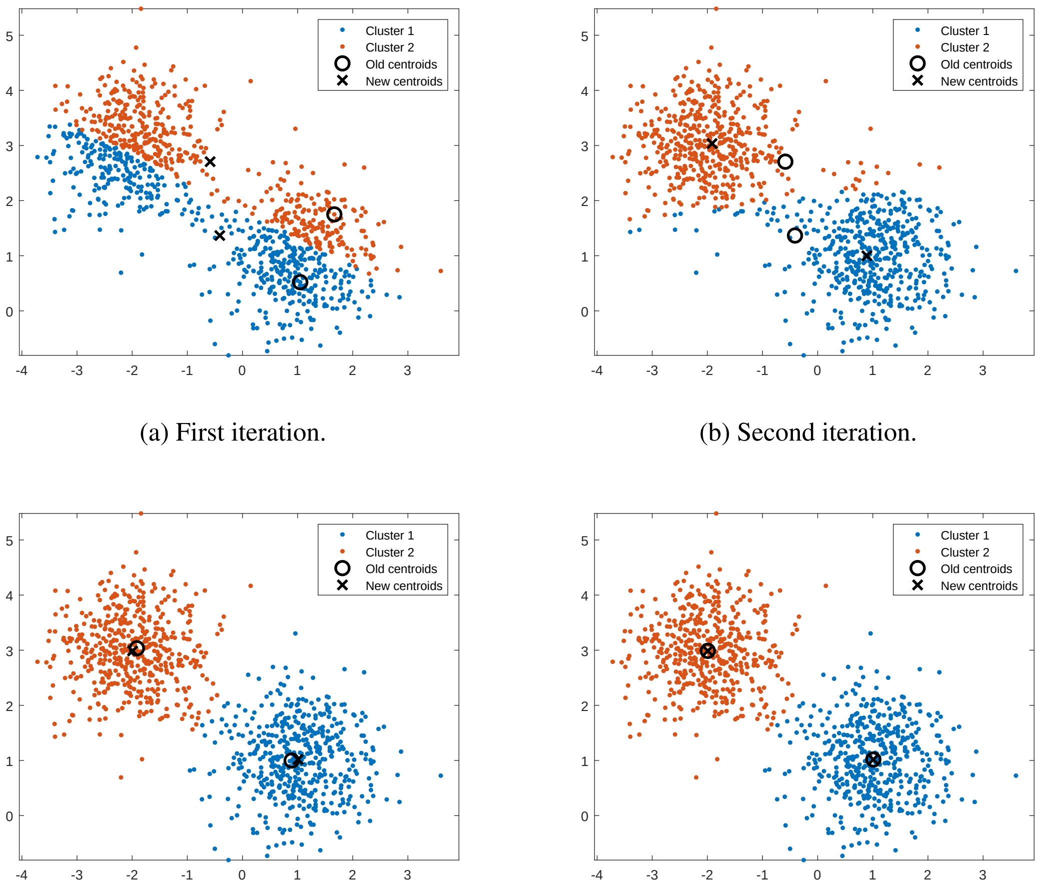
where $\boldsymbol{\mu}_i$ is the mean of the points in S_i, and so represents the centroid of cluster i.

The initial cluster centroids $C^{(1)} = \{\mathbf{c}_1^{(1)}, \mathbf{c}_2^{(1)}, \ldots, \mathbf{c}_k^{(1)}\}$ are selected to be k random samples from the data set. The objective function is minimized iteratively by alternating between two steps. In the first step, each data point is assigned to the cluster centroid that it is closest to. Euclidean distance is used as the distance measure. In the second step, the cluster centroids $C^{(t+1)}$ are updated to correspond to the new sets $S^{(t)}$ by calculating the means with the formula

$$\mathbf{c}_i^{(t+1)} = \boldsymbol{\mu}_i^{(t)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x} \in S_i^{(t)}} \mathbf{x}, \tag{2}$$

where $S_i^{(t)}$ is the ith set in the tth iteration. Figure 1 shows an example of how the clusters and cluster centroids change through four iterations of k-means for two clusters in two dimensions.

When the assignments no longer change, the algorithm has converged to a local minimum of the objective function J. As there is no guarantee that this is the global minimum, k-means is often run multiple times with different initial values for $C^{(1)}$, and the result with the smallest value for J is selected as the final result.



(a) First iteration.

(b) Second iteration.

(c) Third iteration.

(d) Fourth iteration.

Figure 1. Clusters and cluster centroids of subsequent iterations of k-means.

2.2 Density-based spatial clustering of application with noise

DBSCAN uses the local densities of the data points to determine clusters and outliers. This allows for clusters of arbitrary shapes, unlike k-means. When using DBSCAN, the user has to determine two parameters: ε and $minPts$.

The density ρ of point p is defined as the number of points that are within a distance of ε from p, including p itself. Those points are directly reachable from p. If a point has a density higher than $minPts$, it is a core point. A point p is reachable from point q if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly reachable from p_i. If a point is not reachable from any other point, it is an outlier. (Bäcklund et al. 2011)

A cluster is a group of core points that are reachable from each other, and the non-core points that are reachable from any of those core points. By these definitions, all clusters contain at least one core point, all core points are part of a cluster and non-core points are either outliers or the edge points of a cluster.

Figure 2 visualizes a simple case of DBSCAN. The green points are core points. The blue point is a non-core point that is part of the cluster because it is directly reachable from one of the core-points. The red point is an outlier.

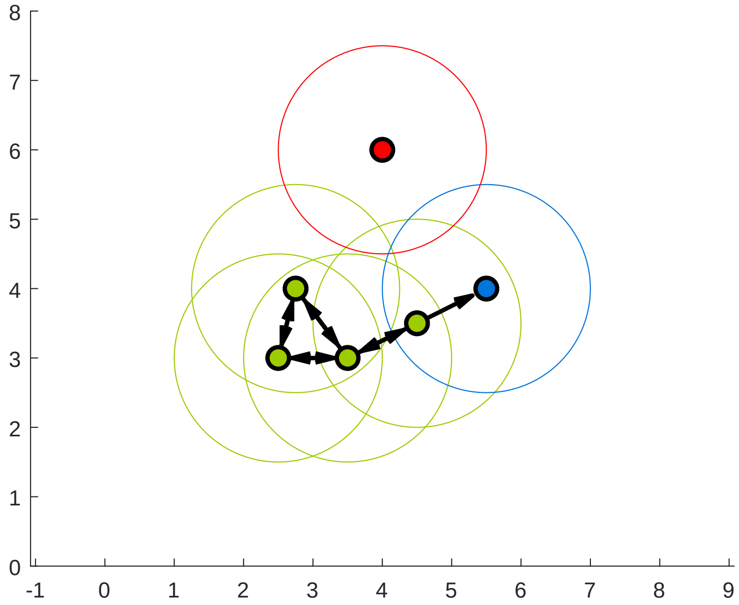


Figure 2. Density-based spatial clustering of application with noise with $minPts = 3$ and $\varepsilon = 1.5$.

2.3 Clustering by fast search and find of density peaks

CFSFDP is a density-based algorithm similar to DBSCAN. Whereas DBSCAN ignores cluster centroids, CFSFDP comes with an assumption that if a point has larger local density ρ than its neighbors, and is at a large distance δ from points with a higher local density, it can be considered a cluster centroid.

With this assumption, Rodriguez et al. (2014) define a measure

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}), \tag{3}$$

where d_{ij} is the distance between points p_i and p_j. For the point with the highest density, $\delta_i = \max_j(d_{ij})$. In Figure 3, we can see that when the data points are plotted in a chart with ρ and δ as axes, the cluster centroids, which are represented by the colorful points in the graph, are separate from the rest of the points. Furthermore, the points in the upper left of the chart have a low density and a large distance to the nearest point with larger density, and can thus be considered as outliers. After the user has selected the cluster centroids from the chart, each of the remaining points are assigned to the same cluster as their nearest neighbor with a higher density.

When determining the outliers, we first define a border area for each cluster as the set of points assigned to that cluster but being within distance d_c from points assigned to other clusters. Then the highest density ρ_b of those points is found, and all of the points in the cluster with a density lower than ρ_b are defined as outliers.

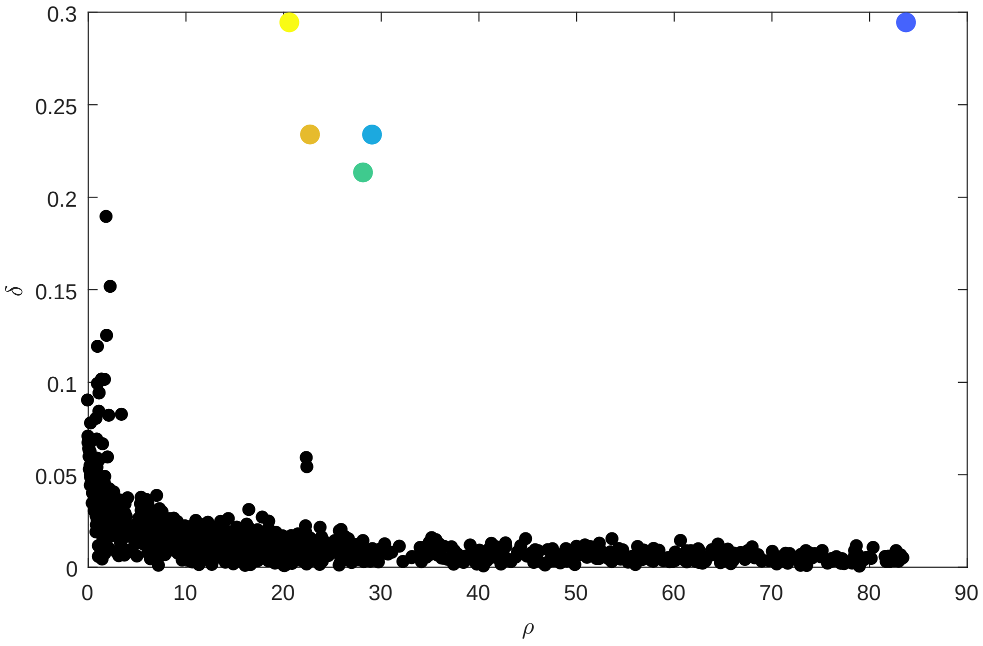


Figure 3. Selecting cluster centroids with CFSFDP.

The parameters needed for CFSFDP are the cutoff distance d_c, and the user input when selecting the cluster centroids, which can also be considered as a parameter. Rodriguez et al. (2014) state that as a rule of thumb, d_c can be selected so that the average number of neighbors is around 1 to 2 % of the total number of points in the data set.

3 Experiments and results

3.1 Data pre-processing

The original data is one-dimensional hourly data where each data point represents electricity consumption of one consumer in one hour. We want to cluster the daily load profiles, so the data was transformed into 24-dimensional where each data point represents the electric load profile of one consumer in one day. A large number of dimensions makes clustering methods slow, so the dimension of the data needed to be reduced. PCA is a traditional way of dimensionality reduction, so that was used. Figure 4 shows how the variance explained by each number of principle components. Using the elbow method, we see that up until four principal components, the added components help preserve much more information of the data, but after that the benefit decreases. With this in mind, and the fact that four principal components explain 99.7 % of the variance, we concluded that this would be the ideal number of dimensions to reduce the data to.

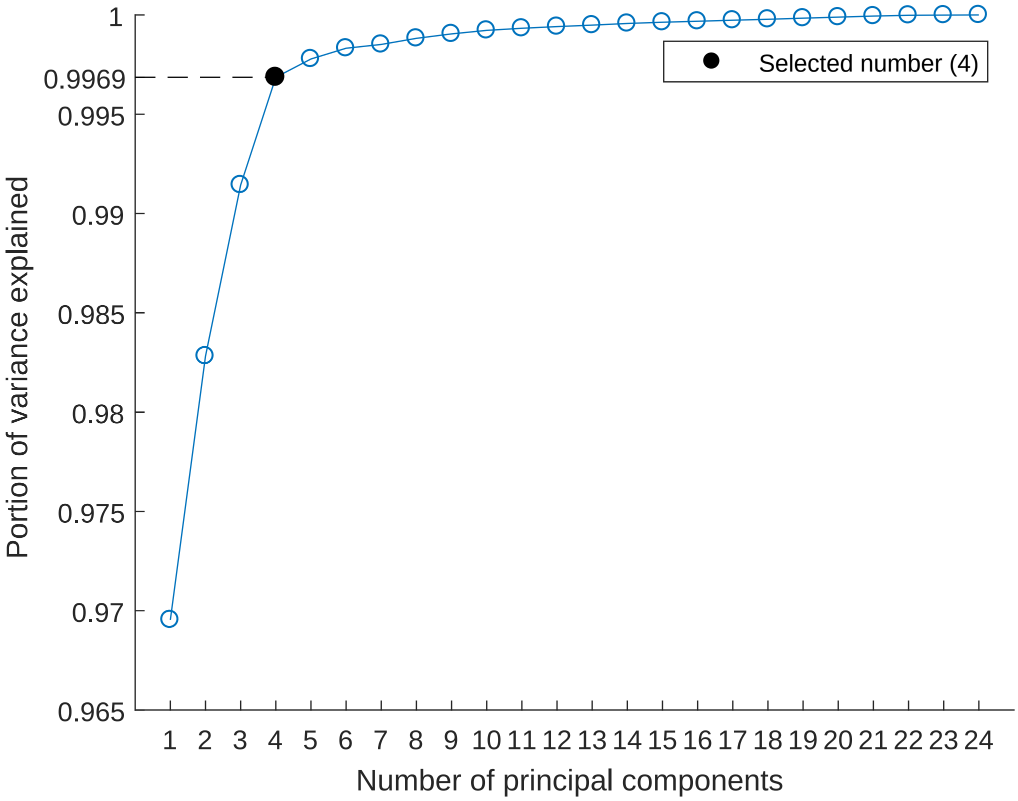


Figure 4. Variance explained by principal components.

A thing to notice is that even with one principal component, 97 % of the variance is explained. This is because the data has very high differences in the magnitude of data points due to some consumers being, for example, detached houses whereas others are big retail stores or farms. The intricacies of the data would emerge more if the data was normalized, but as the goal is to differentiate different kinds of consumers, the magnitude can be thought as one important feature. Therefore the data has not been normalized, but in future research, normalization should be studied.

3.2 k-Means clustering

When using k-means, the amount of clusters k is the only parameter needed. To determine the ideal value, k-means was performed with different values for k, and MSE for each k was plotted in Figure 5. From the figure using the elbow method, we can see that three clusters would be ideal. The specific function that was used to perform k-means was MATLAB's `kmeans()` (*MATLAB k-means* 2017).

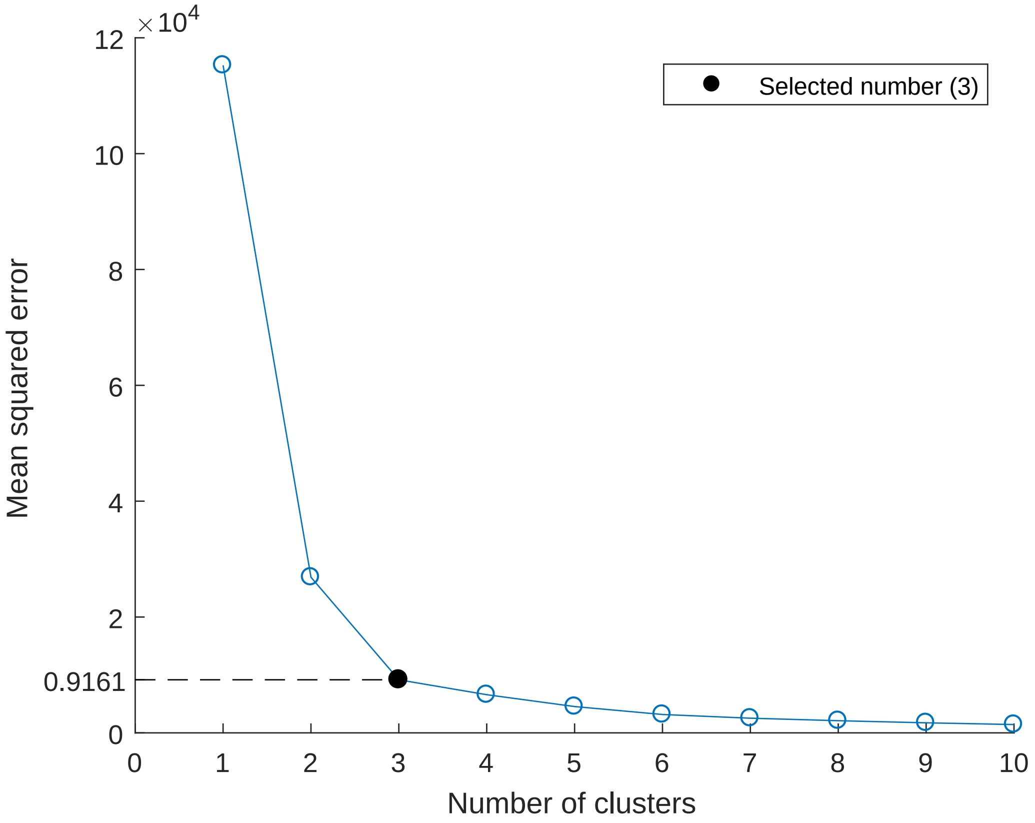


Figure 5. Mean of squared errors for k-means with different numbers of clusters

In Figure 6, we can see the medians and quartiles of the clusters. As the dimensions represent hours in the original data, the medians represent the typical load profiles for each cluster.

As there are three groups with vastly different amounts of energy consumption, k-means divides those groups apart. There are 747 members in cluster 1, 344 members in cluster 2 and 7 114 members in cluster 3. MSE is 9 160.

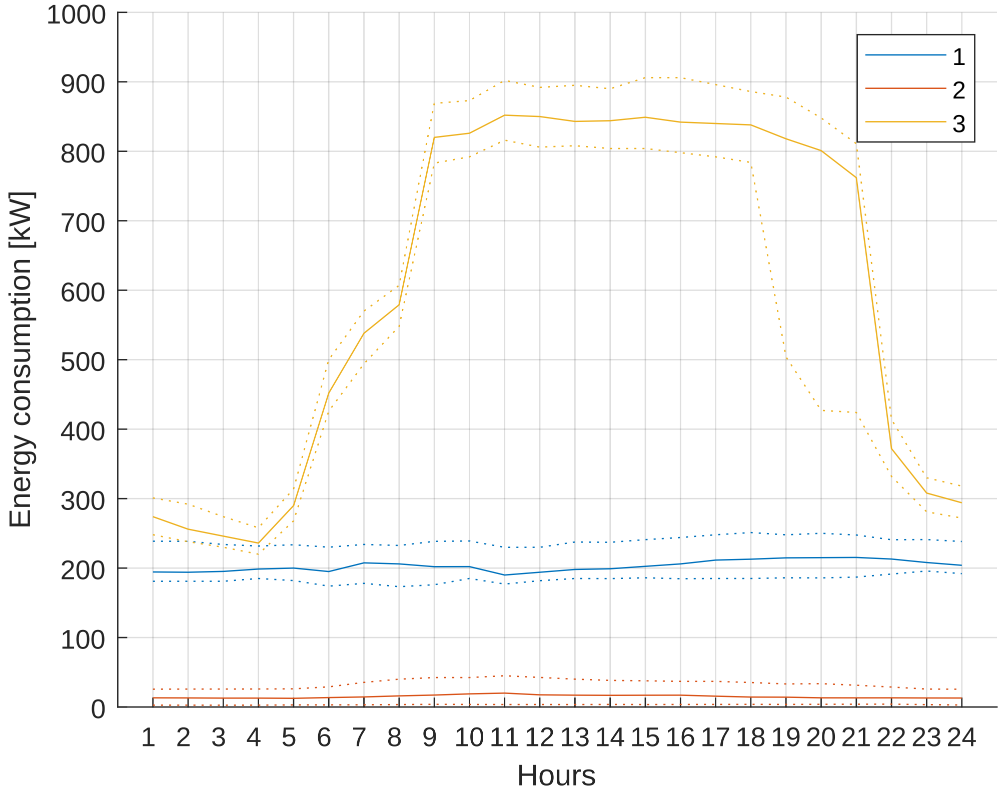


Figure 6. Medians and quartiles for the 3 clusters determined by k-means.

3.3 Density-based spatial clustering of applications with noise

When implementing DBSCAN, a function from MATLAB's file exchange was used (*MATLAB DBSCAN* 2017). DBSCAN takes two parameters, $minPts$ and ε. Determining ideal values for these was a problem. When $minPts$ is selected, a good value for ε can be selected by plotting the distance to the nearest $minPts$ for each point in the ascending order and then using the elbow method to select ε. However, the suitable $minPts$ was not known, so it was decided to plot many curves with different values for $minPts$, select many (ε, $minPts$) pairs and select the ideal pair using MSE. However, manually choosing the pairs would take a large amount of time, so a method was implemented that can detect the elbow point automatically.

If the data was smooth without noise, the elbow point would be in the point with most curvature κ, which is defined by formula

$$\kappa = \frac{y''}{(1 + y'^2)^{3/2}}, \tag{4}$$

where y is the function of the line in relation to the horizontal axis.

However, the lines are not smooth, so they were smoothed with a moving average filter using MATLAB's `smooth()` function (*MATLAB smooth* 2017). After that, the discrete version of (4) was used for $minPts = 1, 2, \ldots, 100$. A subset of the results is in Figure 7. The method seems to correctly detect the elbows.

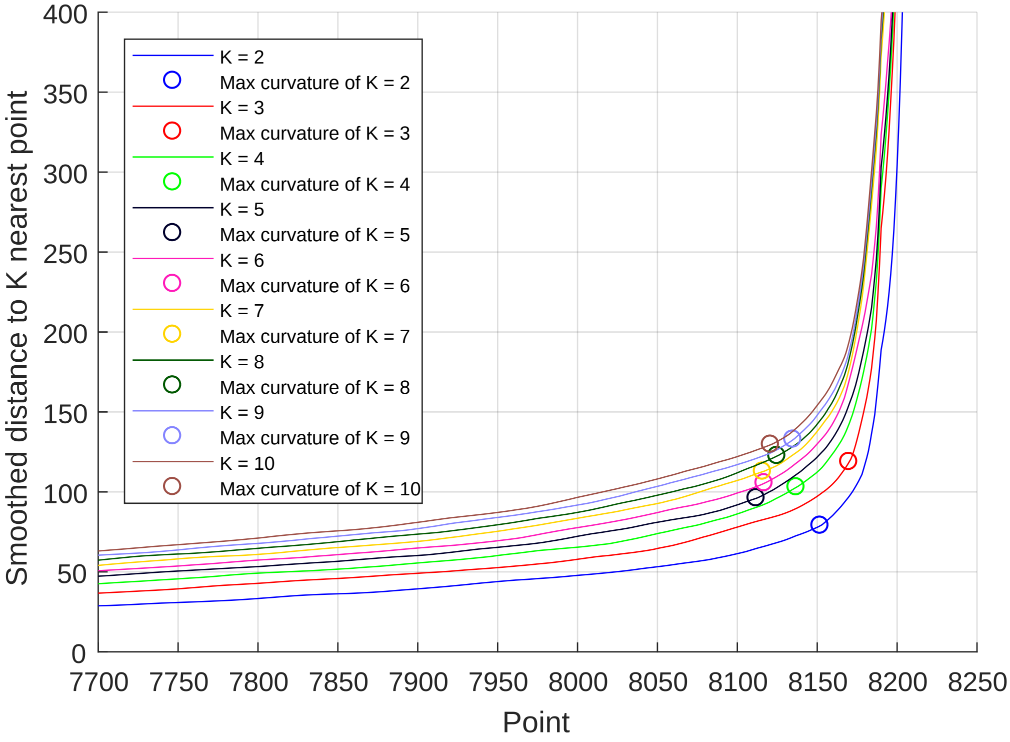


Figure 7. Distance to the nearest point and points with the most curvature.

Next DBSCAN was performed with all of the selected parameter pairs and MSE was computed for each of them. The results were compared with different parameter pairs and it became clear that with more than three clusters, some of the clusters had less than ten members. As clusters with so few members can be considered outliers, it was determined that, similar to k-means, the number of clusters to be used would be three. Same number of clusters also allows for better comparison between the different methods. In Table 1 is listed MSE for all of the parameter pairs where number of clusters is three. The parameter pair ($minPts = 46$, $\varepsilon = 160.9$) produced the lowest MSE $= 5327$, so those were chosen as the parameters to be used.

Table 1. Mean squared error for DBSCAN with parameters that produce three clusters.

$minPts$	ε	MSE
20	134.6	6 297
26	177.7	26 531
38	161.6	5 413
39	163.4	5 396
43	158.6	5 342
46	160.9	5 327
59	183.5	5 354

With the selected parameters DBSCAN results in three clusters that are much like the ones in k-means. The difference is that DBSCAN finds 173 outliers, that do not belong in any cluster. This affects especially cluster 2. There are 680 members in cluster 1, 243 members in cluster 2 and 7 109 members in cluster 3. MSE is 5 327, which is better than for k-means with the same amount of clusters. The medians and quartiles are plotted in Figure 8.

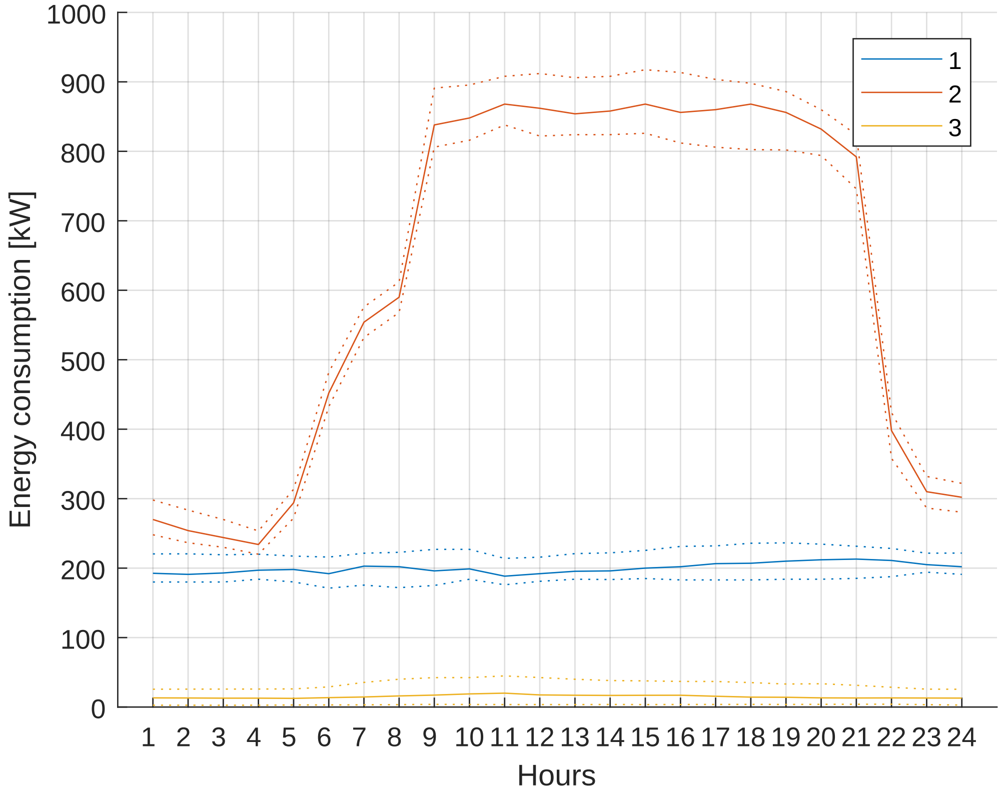


Figure 8. Medians and quartiles for clusters determined by DBSCAN with $\varepsilon = 160$ and $minPts = 46$.

Generally MSE does not give meaningful results for DBSCAN, as DBSCAN can find clusters that are not hyperspherical. However, as the results presented above suggest that DBSCAN clusters the data in much the same way as k-means, which defines only hyperspherical clusters (Jain et al. 1999), we can be confident that the clusters defined by DBSCAN are also hyperspherical.

3.4 Clustering by fast search and find of density peaks

Rodriguez et al. have provided a MATLAB script of their proposed method. For this thesis the script was modified to have faster performance by utilizing preallocation and vectorization (*MATLAB vectorization* 2017). The script was also modified to be a function so that it's easier to use later on. The validity of the modified script was tested by using the example data provided by Rodriguez et al. and confirming that the results are identical for the modified and unmodified scripts. As a result the execution speed of the script increased considerably.

CFSFDP requires determining the cutoff distance d_c. In the original script, this parameter has been modified so that instead of d_c, the user selects the average number of neighbors as a portion P of the dataset. Rodriguez et al. (2014) state that as a rule of thumb, P should be from 1 to 2 %. In these experiments, we varied the value from 1 to 3 % as in some cases if $P \leq 2\%$, the algorithm does not detect any outliers. The script also has an option to use Gaussian kernel as the density measure instead of the cutoff density measure defined earlier. Changing the density measure did not affect the results much, so the cutoff density was used in these experiments.

During runtime, the function produces a decision graph with ρ and δ as axes, where the user is supposed to draw a rectangle that contains the desired cluster centroids, that are separated from the rest of the points. In Figure 9 is the decision graph for the electricity consumption data used in this thesis. The arrows and text boxes were added later, but otherwise the graph in the figure is exactly as produced by the program. From the graph we can see that there are two points that are clearly separated from the rest of the points. This would tell us that there are two distinct clusters in the data set. However, in the lower left corner there are two points that are also somewhat separated from the data set, but not clearly as much as the other two. This proves to be a problem in this seemingly intuitive selection of the cluster centroids. When there are one or more very distinct outliers, the scale of the graph becomes dominated by them. This makes it difficult to distinguish the cluster centroids visually.

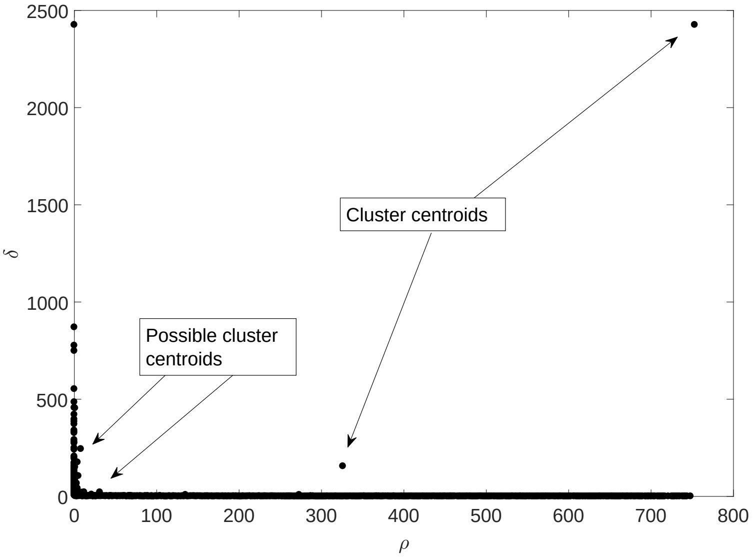


Figure 9. Decision graph with linear scale for Clustering with fast search and find of density peaks where the average number of neighbors is 1.5 % of the whole data set.

To reduce the difference in magnitudes, an option was added to the function that allows the δ-axis to be plotted in logarithmic scale. Figure 10 shows the decision graph with this option selected. Again, it's not exactly clear which points should be selected as the centroids, but the aforementioned four points are more separated than the others.

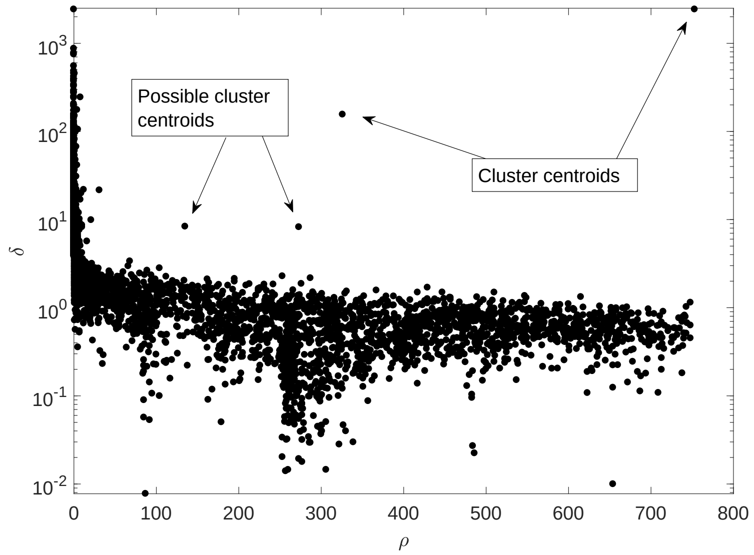


Figure 10. Decision graph with a logarithmic scale for Clustering with fast search and find of density peaks where the average number of neighbors is 1.5 % of the whole data set.

To further make the decision easier in case of dominating outliers, another option was added to the function that allows to ignore the outliers with highest δ in the chart. The user can choose the number of outliers to be ignored. When using this option, δ of the most dense point is assigned to be the same as the highest δ in this new data set without the most distinct outliers. The decision graph with linear scale and 20 distinct outliers ignored is in Figure 11. The same graph with logarithmic scale is in Figure 12.

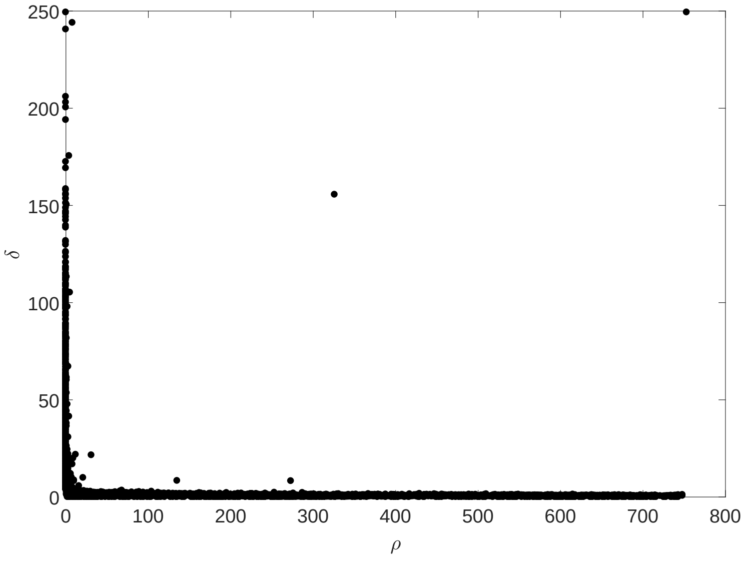


Figure 11. Decision graph with linear scale for Clustering with fast search and find of density peaks where the average number of neighbors is 1.5 % of the whole data set, and 20 of the most distinct outliers have been ignored.

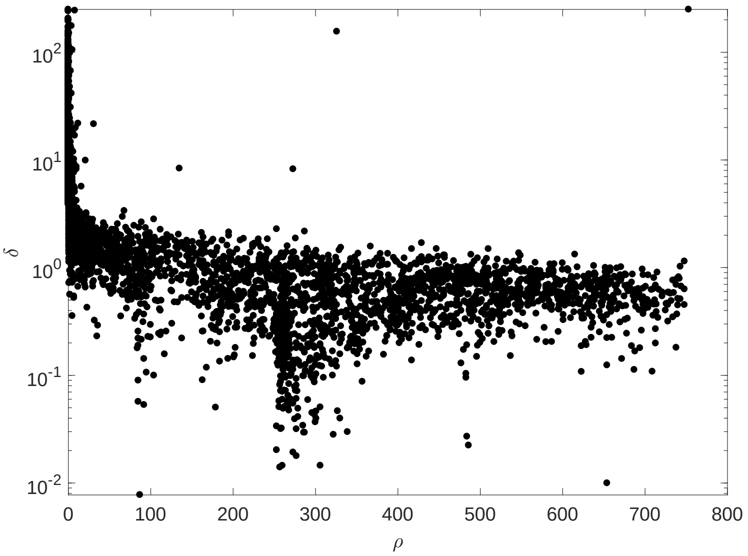


Figure 12. Decision graph with logarithmic scale for Clustering with fast search and find of density peaks where the average number of neighbors is 1.5 % of the whole data set, and 20 of the most distinct outliers have been ignored.

In their article, Rodriguez et al. (2014) also talk about a measure $\gamma = \rho\delta$, which could be used to select the cluster centroids. In Figures 13 and 14, γ is plotted for each point in descending order, with linear and logarithmic scales respectively. The idea is to select the points which have much larger γ than the other points as the cluster centroids.

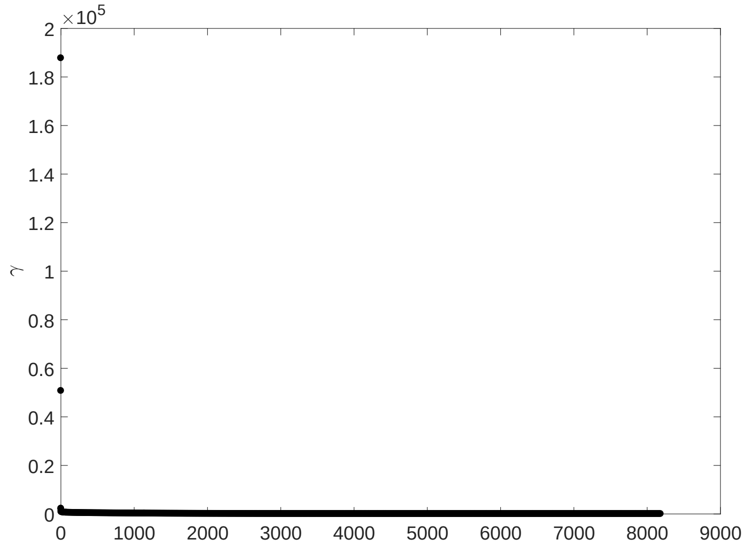


Figure 13. Alternative decision graph with linear scale for Clustering with fast search and find of density peaks where the average number of neighbors is 1.5 % of the whole data set, and 20 of the most distinct outliers have been ignored.

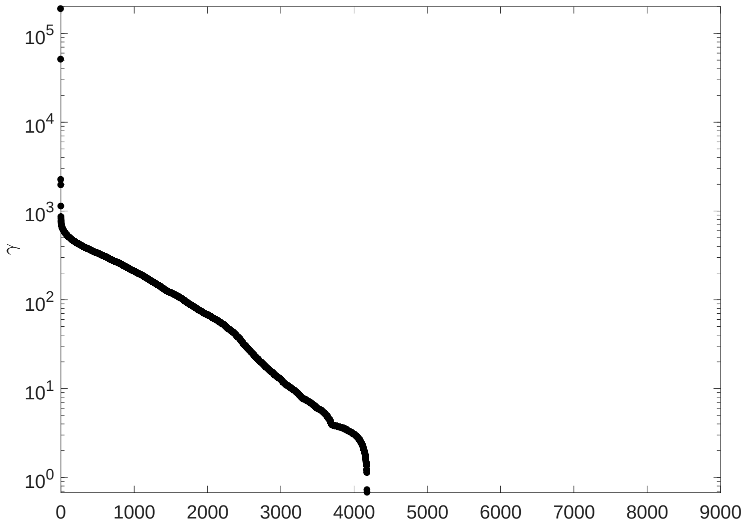


Figure 14. Alternative decision graph with logarithmic scale for Clustering with fast search and find of density peaks where the average number of neighbors is 1.5 % of the whole data set, and 20 of the most distinct outliers have been ignored.

The selection of the number of clusters was difficult as different parameters and visualization options gave different results, so it was decided to use three clusters so that it would be easy to compare the results to k-means and DBSCAN. The value for P was selected to be 3.0 % as its decision graph produced three cluster centroids somewhat separated from the rest of the points. Medians and quartiles with different parameters can be found in Appendix 1.

The result which was selected for comparison is in Figure 15. We can see that the results are very different than the results from k-means and DBSCAN in Figures 6 and 8 on pages 15 and 18. There are 748 members in cluster 1, 1 847 members in cluster 2, 3 249 members in cluster 3, and 2 361 outliers. It seems that CFSFDP tries to distribute the data points to the clusters more equally and it detects much more outliers than DBSCAN. MSE for these clusters is 87 878, which is an order of magnitude greater than with the other two methods.

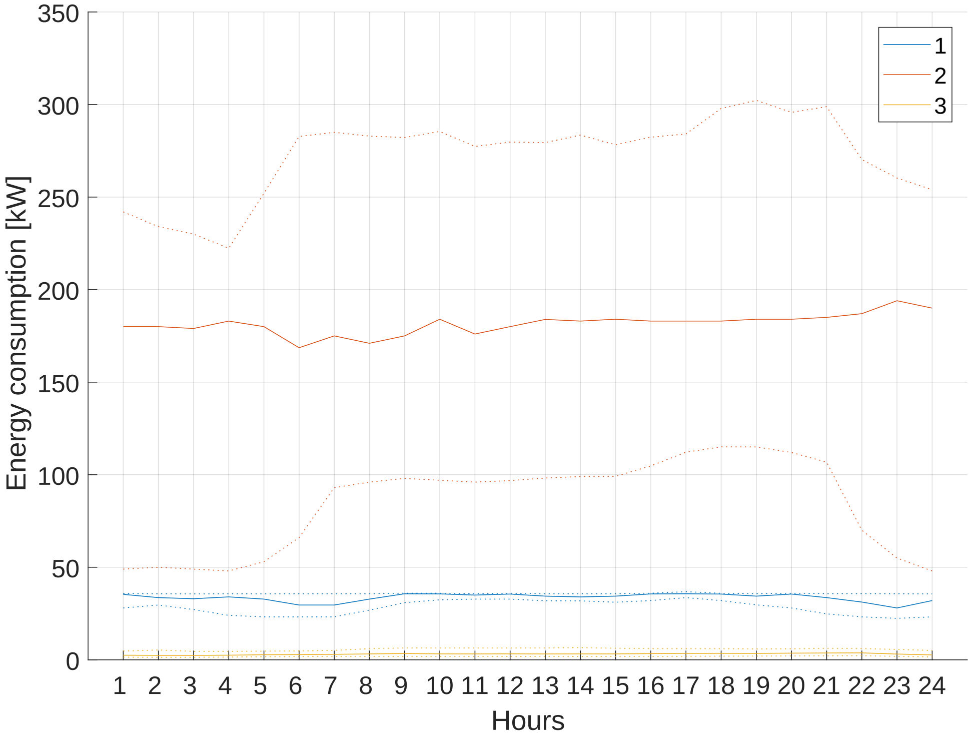


Figure 15. Medians and quartiles for two clusters determined by Clustering with fast search and find of density peaks with $P = 3.0\%$.

4 Conclusions

The aim of the thesis was to compare CFSFDP to k-means and DBSCAN in the analysis of electricity consumption. The goal was reached, although the preprocessing of the data needs further studying. The amount of the data would also have to be larger in order to make concrete conclusions.

CFSFDP does not find typical daily load profiles of electricity consumers as effectively as k-means and DBSCAN with the used data set. The data set proved to be a difficult one to cluster as there were different kinds of customers with largely varying magnitudes of consumption. Therefore, the magnitude ended up being the dominating factor in the clustering.

The intuitive selection of the number of clusters was one of the claimed strong points in CFSFDP, but with this data set the selection was very difficult. In addition to that, the selection of cutoff distance d_c was difficult. With k-means the parameter selection was straightforward by using the elbow method, and with DBSCAN we created an automated method for parameter selection utilizing the elbow method. The MATLAB script for CFSFDP made by Rodriguez et al. (2014) was made into a function and developed further by optimizing the performance and adding options to help with the selection of the number of clusters.

With more data and utilization of the ground truth information of the data, we could make more certain and general conclusions of the differences between the clustering methods in the problem at hand. Normalization of the data could also make the clustering more efficient.

References

Wu, Junjie (2012). *Advances in K-means Clustering*. Springer Theses. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-29806-6 978-3-642-29807-3. DOI: 10.1007/978-3-642-29807-3. URL: http://link.springer.com/10.1007/978-3-642-29807-3 (visited on 11/13/2017).

Bäcklund, Henrik, Anders Hedblom, and Niklas Neijman (2011). "A density-based spatial clustering of application with noise". In: *Data Mining TNM033*, pp. 11–30.

Chicco, G. and J. S. Akilimali (2010). "Renyi entropy-based classification of daily electrical load patterns". In: *Transmission Distribution IET Generation* 4.6, pp. 736–745. ISSN: 1751-8687. DOI: 10.1049/iet-gtd.2009.0161.

Chicco, G. and I. S. Ilie (2009). "Support Vector Clustering of Electrical Load Pattern Data". In: *IEEE Transactions on Power Systems* 24.3, pp. 1619–1628. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2009.2023009.

Chicco, G., O. M. Ionel, and R. Porumb (2013). "Electrical Load Pattern Grouping Based on Centroid Model With Ant Colony Clustering". In: *IEEE Transactions on Power Systems* 28.2, pp. 1706–1715. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2012.2220159.

Chicco, G., R. Napoli, and F. Piglione (2003). "Application of clustering algorithms and self organising maps to classify electricity customers". In: *2003 IEEE Bologna Power Tech Conference Proceedings,* 2003 IEEE Bologna Power Tech Conference Proceedings, vol. 1, 7 pp. Vol.1–. DOI: 10.1109/PTC.2003.1304160.

Chicco, G., R. Napoli, F. Piglione, P. Postolache, M. Scutariu, and C. Toader (2004). "Load pattern-based classification of electricity customers". In: *IEEE Transactions on Power Systems* 19.2, pp. 1232–1239. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2004.826810.

Coke, Geoffrey and Min Tsao (2010). "Random effects mixture models for clustering electrical load series". In: *Journal of Time Series Analysis* 31.6, pp. 451–464. ISSN: 1467-9892. DOI: 10.1111/j.1467-9892.2010.00677.x. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9892.2010.00677.x/abstract (visited on 03/30/2017).

MATLAB DBSCAN (2017). *DBSCAN Clustering Algorithm - File Exchange - MATLAB Central.* URL: http://se.mathworks.com/matlabcentral/fileexchange/52905-dbscan-clustering-algorithm (visited on 11/27/2017).

Gerbec, D., S. Gasperic, I. Smon, and F. Gubina (2005). "Allocation of the load profiles to consumers using probabilistic neural networks". In: *IEEE Transactions on Power Systems* 20.2, pp. 548–555. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2005.846236.

Haben, S., C. Singleton, and P. Grindrod (2016). “Analysis and Clustering of Residential Customers Energy Behavioral Demand Using Smart Meter Data”. In: *IEEE Transactions on Smart Grid* 7.1, pp. 136–144. ISSN: 1949-3053. DOI: 10.1109/TSG.2015.2409786.

Jain, A. K., M. N. Murty, and P. J. Flynn (1999). “Data clustering: a review”. In: *ACM Computing Surveys* 31.3, pp. 264–323. ISSN: 03600300. DOI: 10.1145/331499.331504. URL: http://portal.acm.org/citation.cfm?doid=331499.331504 (visited on 04/12/2018).

MATLAB k-means (2017). *k-means clustering - MATLAB kmeans - MathWorks Nordic*. URL: https://se.mathworks.com/help/stats/kmeans.html (visited on 11/27/2017).

Labeeuw, W. and G. Deconinck (2013). “Residential Electrical Load Model Based on Mixture Model Clustering and Markov Models”. In: *IEEE Transactions on Industrial Informatics* 9.3, pp. 1561–1569. ISSN: 1551-3203. DOI: 10.1109/TII.2013.2240309.

Li, W., J. Zhou, X. Xiong, and J. Lu (2007). “A Statistic-Fuzzy Technique for Clustering Load Curves”. In: *IEEE Transactions on Power Systems* 22.2, pp. 890–891. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2007.894851.

Piao, M., H. S. Shon, J. Y. Lee, and K. H. Ryu (2014). “Subspace Projection Method Based Clustering Analysis in Load Profiling”. In: *IEEE Transactions on Power Systems* 29.6, pp. 2628–2635. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2014.2309697.

Rodriguez, Alex and Alessandro Laio (2014). “Clustering by fast search and find of density peaks”. In: *Science* 344.6191, pp. 1492–1496. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1242072. URL: http://science.sciencemag.org/content/344/6191/1492 (visited on 04/03/2017).

MATLAB smooth (2017). *Smooth response data - MATLAB smooth - MathWorks Nordic*. URL: https://se.mathworks.com/help/curvefit/smooth.html (visited on 11/27/2017).

Tsekouras, G. J., N. D. Hatziargyriou, and E. N. Dialynas (2007). “Two-Stage Pattern Recognition of Load Curves for Classification of Electricity Customers”. In: *IEEE Transactions on Power Systems* 22.3, pp. 1120–1128. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2007.901287.

Tsekouras, G.J., P.B. Kotoulas, C.D. Tsirekis, E.N. Dialynas, and N.D. Hatziargyriou (2008). “A pattern recognition methodology for evaluation of load profiles and typical days of large electricity customers”. In: *Electric Power Systems Research* 78.9, pp. 1494–1510. ISSN: 03787796. DOI: 10.1016/j.epsr.2008.01.010. URL:

http://linkinghub.elsevier.com/retrieve/pii/S0378779608000278 (visited on 03/30/2017).

Varga, E. D., S. F. Beretka, C. Noce, and G. Sapienza (2015). "Robust Real-Time Load Profile Encoding and Classification Framework for Efficient Power Systems Operation". In: *IEEE Transactions on Power Systems* 30.4, pp. 1897–1904. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2014.2354552.

MATLAB vectorization (2017). *Vectorization - MATLAB & Simulink - MathWorks Nordic*. URL: https://se.mathworks.com/help/matlab/matlab_prog/vectorization.html (visited on 10/30/2017).

Verdu, S. V., M. O. Garcia, C. Senabre, A. G. Marin, and F. J. G. Franco (2006). "Classification, Filtering, and Identification of Electrical Customer Load Patterns Through the Use of Self-Organizing Maps". In: *IEEE Transactions on Power Systems* 21.4, pp. 1672–1682. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2006.881133.

Zhong, S. and K. S. Tam (2015). "Hierarchical Classification of Load Profiles Based on Their Characteristic Attributes in Frequency Domain". In: *IEEE Transactions on Power Systems* 30.5, pp. 2434–2441. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2014.2362492.

Appendix 1. Clustering by fast search and find of density peaks

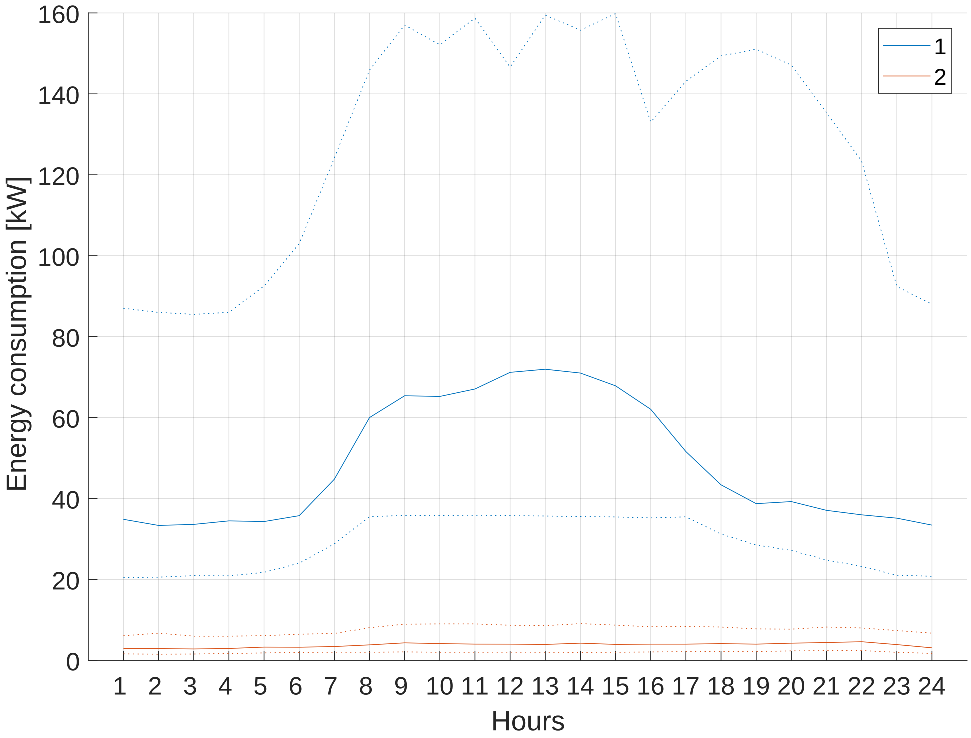


Figure A1.1. Medians and quartiles for two clusters determined by Clustering with fast search and find of density peaks with $P = 1.5\%$.

(continues)

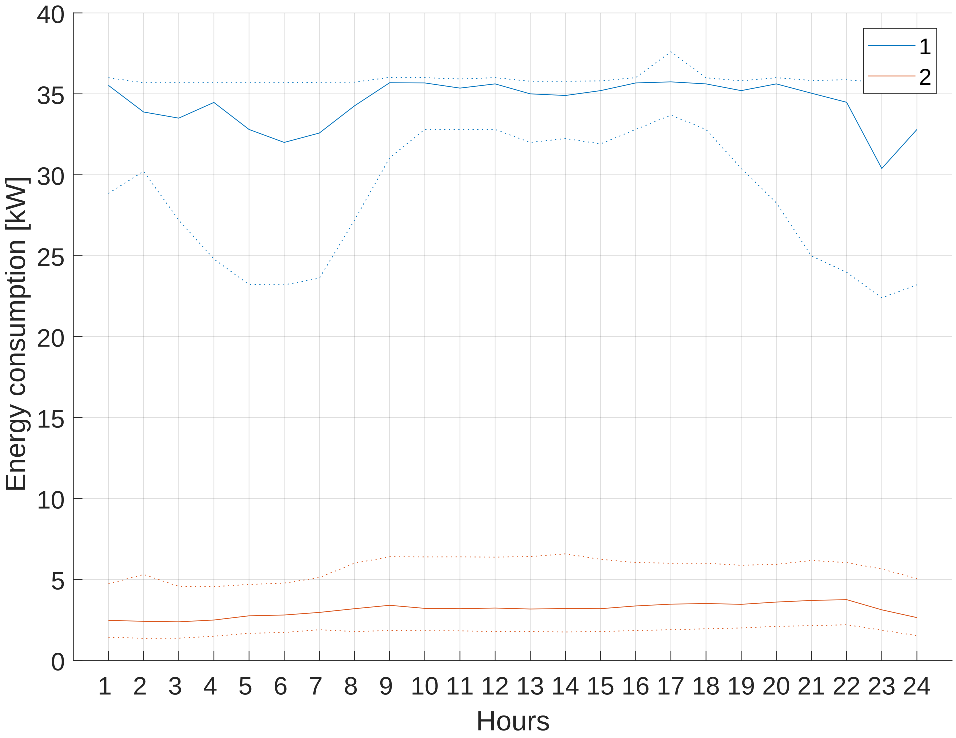


Figure A1.2. Medians and quartiles for two clusters determined by Clustering with fast search and find of density peaks with $P = 3.0\%$.

(continues)

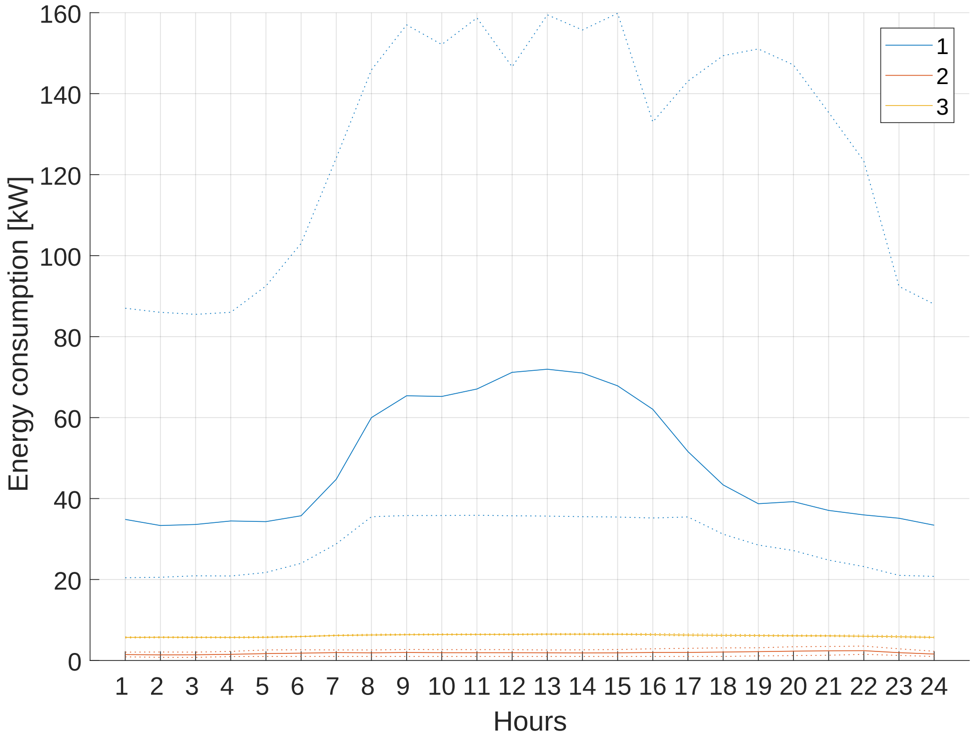


Figure A1.3. Medians and quartiles for three clusters determined by Clustering with fast search and find of density peaks with $P = 1.5\%$.

(continues)

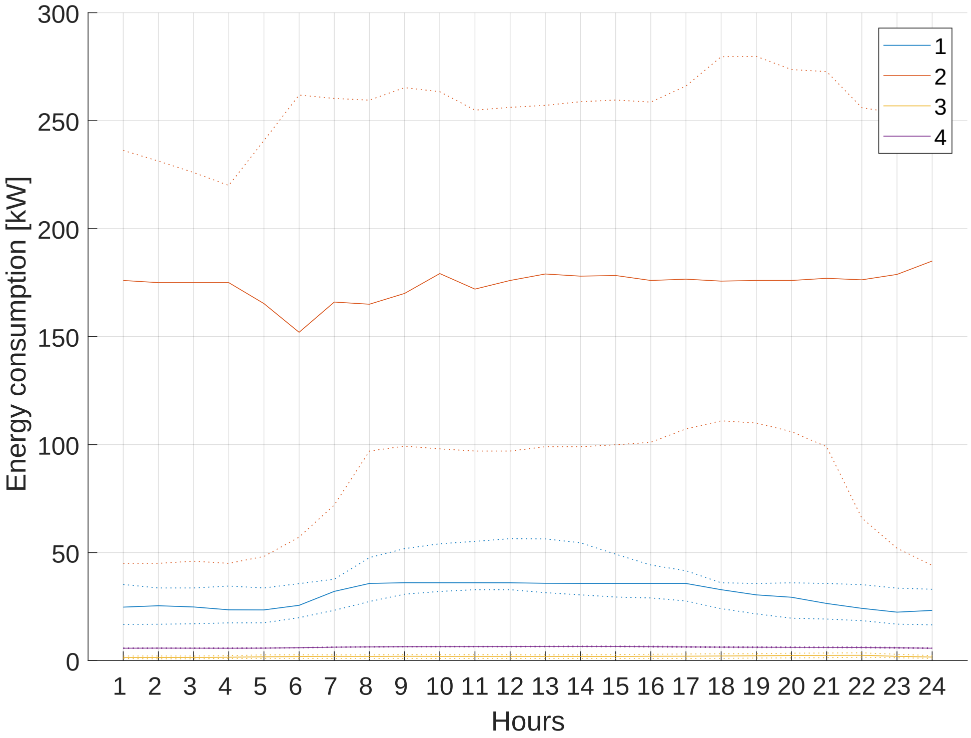


Figure A1.4. Medians and quartiles for four clusters determined by Clustering with fast search and find of density peaks with $P = 1.5\%$.

(continues)

Appendix 1. (continued)

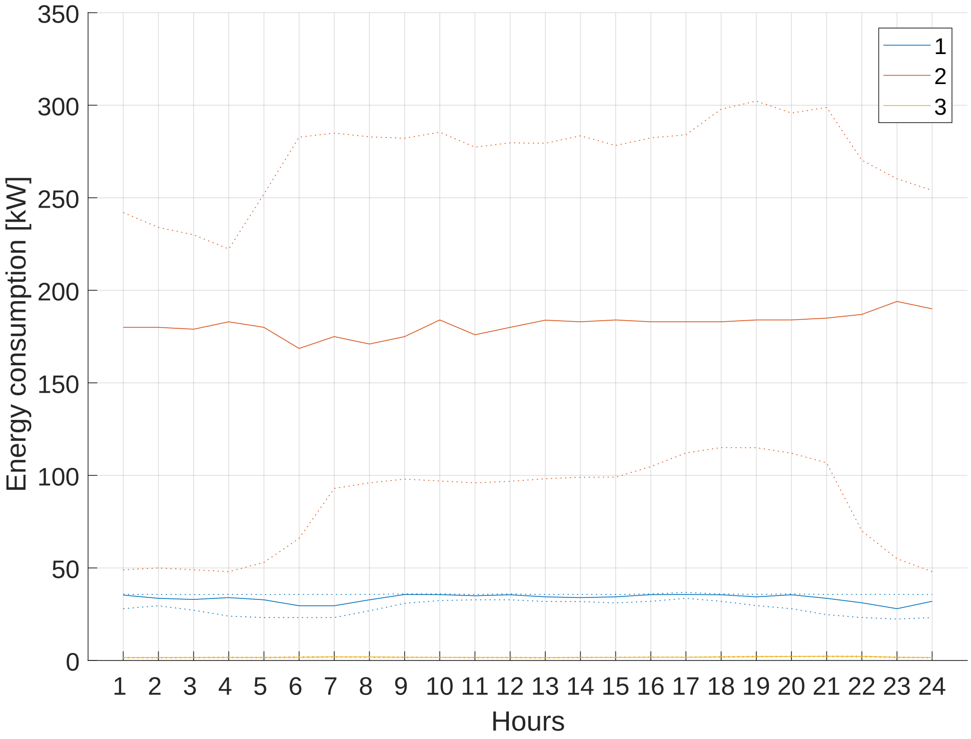


Figure A1.5. Medians and quartiles for four clusters determined by Clustering with fast search and find of density peaks with $P = 3.0\%$. There are only three clusters visible as all of the points in one cluster were assigned as outliers.