International Journal of Power Engineering and Engineering Thermophysics (2017) Vol. 1, Num. 1
Clausius Scientific Press, Canada

A Fast Clustering Algorithm for Power Data

Shuqin Zeng[1, a, *], Haizhou Du[1, b] and Tingting Dou[1, c]

[1]School of Computer Science and Technology, Shanghai University of Electric Power
a. zengshuqin21@163.com, b. duhaizhou@shiep.edu.cn, c. dttdoutingting@163.com
*corresponding author

Keywords: Power data clustering, Density-based Clustering, Energy saving.

Abstract: Energy conservation is an urgent issue to solve on a global scale. A more and more widely used method for energy saving and emission reduction is the applications of data mining technology including data clustering in power system. However, power data has characteristics of large volume, high dimensions, discrete and complex datasets which lead to poor clustering results when we choose common classic clustering algorithm. In our paper, we proposed D-CFSFDP algorithm which is suitable for power data clustering. We do experiments compared with DBSCAN algorithm and K-means algorithm. We demonstrate the power of the algorithm on the power data from Shanghai Energy Conservation Supervision Center.

1. Introduction

Energy is an important foundation for human society to survive and develop. At present, Energy consumption is growing rapidly and demand for energy is greatly increased. Increasing energy efficiency and saving energy on a global scale is imminent. In 2016, the endorsement of G20 Energy Efficiency Leading Programme (EELP) of Hangzhou G20 Summit stated that energy conservation and efficient energy consumption are one of the best ways to rationalize the use of energy resources and the most important measure for climate change in the medium and long term for every countries. Energy saving and emission reduction of power industries are fundamental to the resources and environmental safety, which are directly linked to the overall goal of achieving energy saving and emission reduction [1].

In recent days, a new idea for energy saving and emission reduction is the applications of using data mining technology in power system [2]. We joined a project of the Shanghai Energy Conservation Supervision Center, aiming at finding the methods to save energy by mining the large amounts of power data. However, power data has characteristics of large volume, high dimension, discrete and complex datasets which leads to poor clustering results of power data when we choose common classic clustering algorithm.

In order to solve this problem, in this paper, we will propose a clustering algorithm suitable for power data clustering, D-CFSFDP algorithm which is based on the algorithm proposed in 2014 of Alex Rodriguez et al [3].

The rest of this paper is organized as follows. In Section 2,we will discuss related work of CFSFDP algorithm. In Section 3, we are going to describe our proposed algorithm in detail. Experiments and evaluation will be discussed in Section 4.In the end, Section 5 concludes our paper.

2. Related Work

Alex Rodriguez and Alessandro Laio[3] proposed a clustering algorithm in 2014.Their paper, clustering by fast search and find of density peaks (CFSFDP) is proposed to cluster data by finding of density peaks. CFSFDP is based on two assumptions that: a cluster center is a high dense data point as compared to its surrounding neighbors, and it lies at a large distance from other cluster centers. Based on these assumptions, CFSFDP supports a heuristic approach, known as decision graph to manually select cluster centers. However manual selection of cluster centers is big limitation of CFSFDP in intelligent data analysis. Rongfang Bie, Rashid Mehmood et al [4]proposed a fuzzy- CFSFDP method for adaptively selecting the cluster centers effectively. fuzzy-CFSFDP uses the fuzzy rules based on aforementioned assumption for the selection of cluster centers, compared the resulting clusters with the state of the art methods.
Zhang WenKai and Li Jing [5]proposed an extension of CFSFDP, E_CFSFDP inspired by the idea of a hierarchical clustering algorithm CHAMELEON because that CFSFDP performs not well when there are more than one density peak for one cluster, namely "no density peaks". They used the original CFSFDP to generating initial clusters first, then merge the sub clusters in the second phase. They have conducted the algorithm to several data sets, of which, there are "no density peaks".
Shihua Liu, Bingzhong Zhou[6] proposed DPC_M algorithm based on CFSFDP.DPC algorithm constructs a Decision Graph by computing a local density and a relative distance to discover the cluster center in a dataset. The remaining data points in the dataset are allocated at once to the cluster to which the nearest cluster center belongs. The key issue for the DPC algorithm proposed in literature is how to define the distance measurement between data points in the mixed dataset. Therefore, the DPC_M algorithm designed for the clustering of the mixed data proposed in this paper is constructed by using a new unified dissimilarity metric between the mixed data points.

3. Proposed Algorithm

First of all, we will introduce the background of D-CFSFDP algorithm, and then we will describe it detailedly in part of this chapter behind.

3.1. Background

The algorithm was based on the assumptions that cluster centers are surrounded by neighbours with lower local density and that they are at a relatively large distance from any points with a higher local density. For each data point, we only need to compute two quantities: its local density and its distance from points of higher density. The local density of data points is defined as:

$$\rho_i = \sum_j X(d_{ij} - d_c) \quad , x(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \qquad (1)$$

Where d_c is a cut off distance. δ_i is measured by computing the minimum distance between point x_i and any other point x_j with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \qquad (2)$$

For point with highest density, we conventionally take $\delta_j = \max_j (d_{ij})$.

Generally, ρ_i is equal to the number of points that are closer than d_c to point x_i. The algorithm is highly correlated with the distance between points, thus the results of the analysis are robust with respect to the choice of d_c for large data sets.

Then select data points with large local density ρ and large distance δ as cluster center. In order to determine the number of cluster centers quantitatively the author gives a definition of $\gamma_i = \rho_i \delta_i$. Hence data points with a higher value of γ are more likely to be cluster centers. Sort γ in descending order and choose data points with relatively large value of γ.
This algorithm has many advantages and it is suitable for power data clustering. However, it may exist some problems at some special cases and we would like to make improvements to get a better power data clustering result.

3.2. D-CFSFDP Algorithm

As mentioned above, local density of data point x_i is $\rho_i = \sum_j X(d_{ij} - d_c)$. Problems do exist with this formula in some special circumstances. For example, if ρ_i is equal to ρ_j and they are the largest local density points, what's more, x_i is very close to point x_j .These two points belonging to the same cluster will be divided into two clusters according to distance formula. This case will be discussed in Step 5 in detail.
In our algorithm, for the above case, we improve the distance formula by adding:
If $\rho_i = \rho_j$ and $d_{ij} < d_c$,Then $\delta_j = d_{ij}$.
Before calculating $\gamma_i = \rho_i \delta_i$,we will take a z score scale for ρ and δ respectively and then calculate γ. It will be described in detail later in step 6.

Step 1: Data Preprocessing
Step 2: Calculating distance d_{ij}
In field of data mining including clustering analysis, similarity between data points is generally calculated by distance.The popular distance formula, Minkowski-form distance is defined based on the norm:

$$d(x,y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} \qquad (3)$$

When p is equal to 2, $d(x,y)$ is Euclidean distance L_2 .The Euclidean distance between two points is the length of the path connecting them. Euclidean distance is the most common use of distance [8, 9]. In our algorithm, Euclidean distance is also our best choice.
Step 3: Selection of d_c
The choice of parameter d_c is very important that too big or too small may degrade the performance of algorithm. If d_c is too large, local density of each data point will be consequently larger than it should be, which may result in a decrease in the number of clusters. And if it is too small, data points that originally belong to one cluster may be divided into several clusters, resulting in a significant increase in the number of clusters. For the choice, the author gives a suggestion: one can choose d_c so that the average number of neighbours is around 1% to 2% of the total number of points in data set. We own large power data sets, and the choice for our project is about 2%.
Step 4: Calculating local density
Step 5: Calculating cut off distance

As mentioned earlier, distance formula of calculating δ has problems in some special circumstances. For example, when ρ_i is equal to ρ_j and they are the largest local density points, in the meantime, x_i is very close to point x_j .

As we can see in Figure 1, the serial number represents density of data points. Number 6 data point has a neighbour of number 7 data point and they own the same density which is the largest local density. Since ρ_{x_6} is equal to ρ_{x_7} ,from distance equations we can see, δ_6 =min (d_{61} , d_{62} , d_{63} , d_{64} , d_{65})= d_{62} , similarly δ_7 =min (d_{71} , d_{72} , d_{73} , d_{74} , d_{75})= d_{72} . Both d_{62} and d_{72} is large enough that data point of number 6 and number 7 may be considered as cluster centers. Obviously this result is wrong because it is against its prerequisite assumption that cluster center is far from all higher local density points. However we need to distinguish such case from the situation of Figure 2.

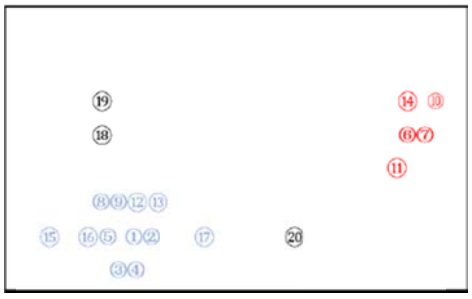

Figure 1 Unexpected case.

In Figure 2, data points of number 6 and number 7 have the same density and they both have large distance from points of higher density. Different from the above situation, data points of number 6 and number 7 really belong to two different clusters respectively.

Figure 2 Normal circumstances.

In order to distinguish between those two cases, when we have two points with same largest local density and rather large distance, firstly we will check the distance between these points. If their distance d_{ij} is much larger than cut off distance, it is clear that it belongs to the second case, which matches the original formula. Otherwise, d_{ij} is less than d_c , namely, the first case that we want to emphasize. Under these circumstances, we would like to select one of these points as cluster center meanwhile assign a very small distance to other points if there are two or more adjacent local density maximum points with the same density. Hence, we add the equations to solve this kind of problem.

If $\rho_i = \rho_j$ and $d_{ij} < d_c$,then we choose data point x_i as cluster center, and assign d_c which is a very small distance to data point x_j .

Step 6: Calculating the value of γ

From the front of this paper we can see $\gamma_i = \rho_i \delta_i$. Hence data points with a higher value of γ are more likely to be cluster centers. In some special cases there may exist problems. The first case is that some points that are of large local density but have a small distance value that are selected as center point, which may cause two centers in the same cluster. The second case is that ρ is small, but δ is large, so that some of the abnormal points will be considered as cluster centers.

For this case, we take a z score scale for ρ and δ respectively and then calculate γ.

The standard score of a raw score $x^{[1]}$ is

$$z = \frac{x - \mu}{\delta} \qquad (4)$$

Where μ is mean value of the population. σ is standard deviation of the population. The absolute value of z represents the distance between raw score and population mean in units of the standard deviation [9,10,11].

Step 7: plot decision graph

Take ρ for abscissa axis and δ for the ordinate, we then plot ρ-δ decision graph. Determine the number n of cluster centers according to decision graphs and output the cluster centers data set C.

4. Experiments

Our algorithm has its basis in distance between data points, thus, choice of parameter d_c is very important that too big or too small d_c may degrade the performance of algorithm. If too large, local density of data points will be consequently larger than they should be, which may result in a decrease in the number of clusters. And if the parameter is too small, data points that originally belong to one cluster may be divided into several clusters, resulting in a significant increase in the number of clusters. We choose 2% as t which means that we select d_c so that the average number of neighbors is around 2of the total number of points in the data set.

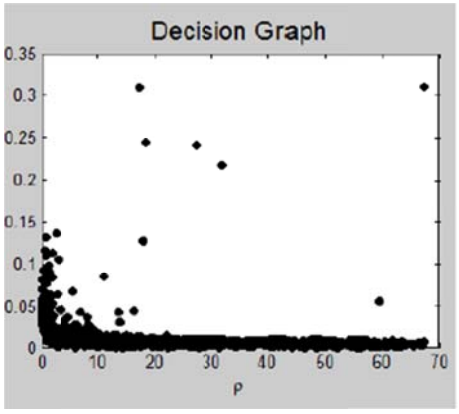

Figure 3 t=1%

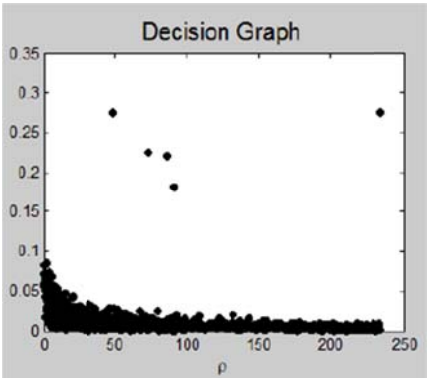

Figure 4 t=5%

In Figure 3, Figure 4 and Figure 5, the value of t is 1%,5%and 10% respectively. From the figure of our experiments, we can see the choice of t is robust that it ranges from 1% to 10% which shows good robustness of our algorithm.

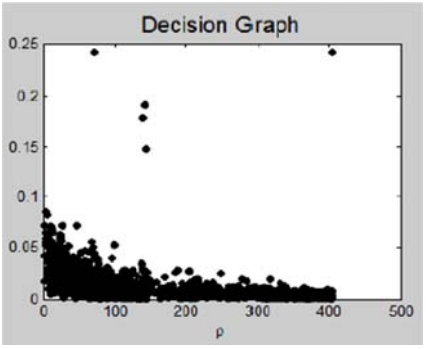

Figure 5 t=10%

DBSCAN and K-means algorithms are mostly used clustering algorithms. K-means is a simple algorithm and it usually works fast but has an important drawback. The performance of K-means depends on the value of K, that is, we need to find a value of K in advance .DBCAN allows the finding of nonspherical clusters but works only for data defined by a set of coordinates and is computationally costly [3]. Figure 6 shows the runtime of these three algorithms. Generally speaking, DBSCAN algorithm is the slowest one. As we know, when the amount of data increases, DBSCAN algorithm requires a larger memory support and I / O consumption [12,13,14]. Thus, DBSCAN is a poor choice for us to choose when we decide to cluster for large data sets. However, when the number of data sets is lower than 1800, DBSCAN may operate faster than K-means. The runtime of our algorithm shows its superiority compared with other two algorithms since we have simple calculations and fast searching.

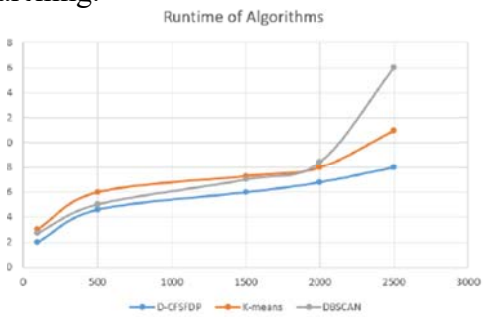

Figure 6 Runtime of three algorithms

5. Conclusion

We have introduced a fast clustering algorithm, D-CFSFDP algorithm. The D-CFSFDP algorithm has many advantages. It does not need to specify the number of clusters in advance compared with K-means algorithm. It is robust with respect to changes in the metric change. It has the ability to identify any shape cluster. And we prove the robustness, fast speed and accuracy of our algorithm, respectively by experiments. Generally speaking, the algorithm achieves expected goal. Experimental results on data sets of the power plant demonstrate that the algorithm is very effective.

Acknowledgements

We would like to appreciate Shanghai Energy Conservation Supervision Center.

References

[1] Chao Jing,Tianlong Guet al,“An Energy-Saving Clustering Algorithm Based on LEACH”,Knowledge Acquisition and Modeling Workshop,2008,DOL:10.1109/2008.4810509.

[2] J. Han and M. Kamber, “Data mining: Concepts and techniques,“Data Mining Concepts Models Methods and Algorithms Second Edition, vol. 5, no. 4, pp. 1 – 18, 2006.
[3] Alex Rodriguez, Alessandro Laio, “Clustering by fast search and find of density peaks”, Vol.344,no.6191,pp.1492-1496, Science 27, June 2014.
[4] Rongfang Bie, Rashid Mehmood, Shanshan Ruan, Yunchuan Sun et al, “Adaptive fuzzy clustering by fast search and find of density peaks”, Volume 20, Issue 5, pp 785–793, October 2016.
[5] Zhang WenKai, Li Jing, “Extended fast search clustering algorithm：Widely density cluster,no density peaks”,
[6] Shihua Liu,Bingzhong Zhou et al,“Clustering Mixed Data by Fast Search and Find of Density Peaks”, Mathematical Problems in Engineering, Volume 2017 (2017), Article ID 5060842, 7 pages.
[7] P. Berkhin, “A survey of clustering data mining techniques,” in Grouping Multidimensional Data, J. Kogan, C. Nicholas, and M. Teboulle, Eds.Springer Berlin Heidelberg, 2006, pp. 25–71.
[8] Jimeng Sun, Fei Wang, Jianying Hu, Shahram Edabollahi,“Supervised Patient Similarity Measure of Heterogeneous Patient Records”,Volume 14, Issue 1, June 2012.
[9] Emre Karakoc, Artem Cherkasov, S. Cenk Sahinalp ,”Novel Approaches for Small Biomolecule Classification and Structural Similarity Search”, Sigkdd , Volume 9, Issue 1, June 2007.
[10] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, E. Simoudis, J. Han, U. Fayyad, Eds. (AAAI Press, Menlo Park, CA, 1996), pp. 226–231.
[11] Y. Cheng, Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. 17, 790 (1995). CrossRefWeb of Science Search Google Scholar .
[12] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation. ACM Trans. Knowl. Discovery Data 1, 4 , es (2007). CrossRef Search Google Scholar .
[13] P. Fränti, O. Virmajoki, Iterative shrinking method for clustering problems. Pattern Recognit. 39, 761–775 (2006). CrossRefWeb of Science Search Google Scholar .
[14] Biant D,Kut A(2007).“ST-DBSCAN:An algorithm for clustering spatial-temporal data.”Data and Knowledge Engineering,60(1),208-221.