Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 560 – 567

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

DENCLUE-IM: A New Approach for Big Data Clustering

Hajar REHIOUI*, Abdellah IDRISSI, Manar ABOUREZQ, Faouzia ZEGRARI

Computer Science Laboratory (LRI)
Computer Science Department, Faculty of Sciences
Mohammed V University in Rabat

Abstract

Every day, a large volume of data is generated by multiple sources, social networks, mobile devices, etc. This variety of data sources produce an heterogeneous data, which are engendered in high frequency. One of the techniques allowing to a better use and exploit this kind of complex data is clustering. Finding a compromise between performance and speed response time present a major challenge to classify this monstrous data. For this purpose, we propose an efficient algorithm which is an improved version of DENCLUE, called DENCLUE-IM. The idea behind is to speed calculation by avoiding the crucial step in DENCLUE which is the Hill Climbing step. Experimental results using large datasets proves the efficiency of our proposed algorithm.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Big Data; Clustering; Density based clustering; DENCLUE.

1. Introduction

The technological revolution has generated tens of terabytes of heterogeneous data every day. According to an investigation made by the institute IDC[1], 1.8 zettabyte data was created in 2011, 2.8 zettabytes in 2012 and it will increase to 40 zettabytes in 2020. This large quantity of complex data, which can be part of Big data, needs a more developed technology to better store, use and analyse.

There is several propositions to define Big Data[2]. The most widely used definition is that proposed by Gartner[3], it is based on the notion of the 3 Vs: Volume, Velocity and Variety.

- **Volume:** The volume of data in the world increases exponentially. For instance, according to statistics made in 2012[4], Twitter generated 7 terabytes of data each day and Facebook 10 terabytes. This massive amount of data will present major challenges in the future.

*Corresponding author. Tel.: +212-674-206-349.
E-mail address: rehioui.hajar@gmail.com

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.265

- **Velocity:** The speed and high frequency of data creation represent a real challenge, especially in real time applications. We must note that the thousand of terabytes of data are generated every hour[5].
- **Variety:** The huge amounts of data are generated from the social networks, the smart phones, the sensors and more. As a result, heterogeneous data are produced.

In this context, researchers have gone in the direction to add other Vs for big data definition. H.U Buhl et al.[6] suggest that the study of the correctness and accuracy of information is necessary and then include the fourth V for veracity. J. Gantz and D. Reinsel[7] take into account another V which is value. The value of the data extracted after analyse are more important than the data itself. The variability is the sixth V which has been proposed by the NIST[8], it refers to the transformation that affect the meaning of the same information, referenced in other contexts.

As mentioned above, the big data produce a heterogeneous data (Variety) which makes it difficult to exploit. To overcome these limits, clustering methods can be considered as a promising solution. Generally, The challenge in large data is to provide a clustering method that produce an acceptable quality of clustering in a reasonable runtime. For this end, the density-based clustering methods are widely used thanks to their ability to classify the large databases (Volume), and to their efficiency to omit noisy data (Veracity).

In this context, M. Ester et al.[9] proposed a DBSCAN method as a new density-based clustering algorithm for large spacial databases. Based on this work, a lot of variant of DBSCAN are proposed in literature such as OPTICS[10], ST-DBSCAN[11], MR-DBSCAN[12], etc. However DBSCAN and its variants operate efficiently only on spatial data, and have their limitations with respect to large dimensional data. To overcome these shortcoming, the DENCLUE algorithm was proposed by A. Hinneburg and D. A. Keim in[13].
Nevertheless, the DENCLUE algorithm suffers in term of the execution time. This is due to the hill climbing method which slows down the convergence to the local maximum. That is why in the present contribution we aim to improve this algorithm in order to obtain clusters in a reasonable response time.

The paper is organized as follows. We present in the next section, the DENCLUE algorithm. We propose its improvement in section 3. We expose experimental results in section 4 and conclude in section 5.

2. Data clustering

Clustering called also unsupervised classification is a process of categorizing a set of data into homogeneous groups (clusters). The elements in each cluster should be similar. Thus, the similarity between individuals in the same cluster (intra-class) must be small and high between the different clusters (inter-class). This similarity is considered as a distance measure. Mathematically, the ultimate goal of data clustering is to partition a set of unlabeled objects $O = \{o_1, o_2, ..., o_n\}$ into k clusters. Each object is characterized by a feature vector $X = \{x_1, x_2, ..., x_p\}$, where p is its dimension. A large number of clustering methods are developed in literature, which can be grouped[14,15] based on some specific criteria[16]. But generally they are divided into five families[17,18] as illustrated in figure 1.

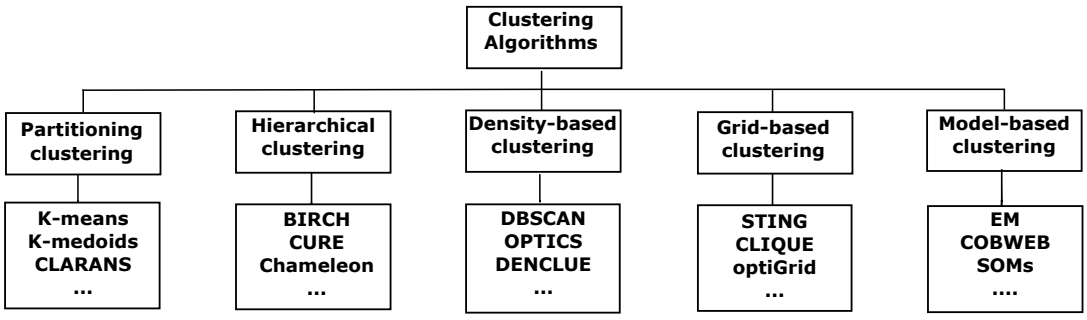

Fig. 1: Clustering taxonomy[17].

- **Partitioning clustering:** This type of clustering is the simplest one, its function is to divide data into many clusters. To reach this objective, initial groups are formed and assembled in order to have the final clusters.

- **Hierarchical clustering:** This type group objects into a tree of clusters, the algorithms in this type are divided into two categories, divisive (top to bottom) and agglomerative (bottom to top)[19]. The first type puts all the data into a single cluster, then divided it hierarchically until forming the final clusters. The second type puts each object of the database in one cluster, and merges them after that recursively until the last clusters are formed.
- **Density-based clustering:** In this type, the objects are classified based on their regions of density. The density-based algorithms have the ability to discover classes of arbitrary shapes and omit noisy objects.
- **Grid-based clustering:** In this type of clustering, data are divided into grid of objects. This type applied the algorithm on the grid, rather than applied it directly on the database.
- **Model-based clustering:** This type is based on the hypothesis that the data is generated by a probability distributions. These methods aimed to emit a model assumption for each cluster, then find the best fit of the data to the model.

In this work, we give special attention to density based clustering algorithms. This family of methods has proved its efficiency in term of clustering thanks to its possibility to find clusters of arbitrary shapes and also to detect noisy objects. In this context, we focus on DENCLUE algorithm. This method is characterized by its fast response time compared with other density based algorithms as demonstrated in the literature[17,13].

2.1. DENCLUE

DENCLUE[13] (DENsity-based CLUstEring) is considered as a special case of the Kernel Density Estimation (KDE)[20,21,22]. The KDE is a non-parametric estimation technique, which aimed to find dense regions points. The authors of DENCLUE developed this algorithm to classify large multimedia databases, because this type of database contains large amounts of noise, and requires clustering high-dimensional feature vectors.
Principally, DENCLUE operates through two stages, the pre-clustering step and the clustering step as illustrated in figure 2. The first step is for constructing a map (a hyper-rectangle) of the database. This map is used to speed the calculation of the density function. As for the second step, it allows identifying clusters from highly populated cubes (the cubes of which the number of points exceeds a threshold ξ determined in parameters), and theirs neighbouring populated cubes.

DENCLUE is based on the calculation of the influence of points between them. The total sum of these influence functions represent the density function. There exist many influence functions, based on the distance between two points x and y; but we will focus in this work on the Gaussian function.
The equation (1), derived from[13], shows the influence function between two points x and y.

$$f_{Gauss}(x, y) = \exp^{-\frac{d(x,y)^2}{2\sigma^2}}, \tag{1}$$

where $d(x, y)$ is an euclidean distance between x and y, and σ represents the radius of the neighbourhood containing x.
Equation (2), extracted from[13], represents the density function.

$$f_D(x) = \sum_{i=1}^{N} f_{Gauss}(x, x_i), \tag{2}$$

where D represents the set of points on the database, and N its cardinal.
To determine the clusters, DENCLUE calculate the density attractor for each point in the database. This attractor is considered as a local maximum of the density function. This maximum is found by the Hill Climbing algorithm, which is based on gradient ascent approach[22] as shown in equation (3), presented in[13].

$$x = x^0,\ x^{i+1} = x^i + \delta \frac{\nabla f^D_{Gauss}(x^i)}{\|\nabla f^D_{Gauss}(x^i)\|} \tag{3}$$

The calculation ends when $f^D(x^k) < f^D(x^{k+1})$ with $k \in N$, then we take $x^* = x^k$ as a density attractor.
The points forming a path with the density attractor, are called attracted points. Clusters are made by taking into account the density attractors and its attracted points.

The strength of this algorithm resides in the choice of the structure with which the data are presented. A. Hinneburg and D. A. Keim [13] have chosen to work with the concept of hyper-rectangle. A hyper-rectangle is constituted by hyper-cubes. Each hyper-cube is represented by the dimension of the feature vector points (i.e., the number of criteria) and by a key. This structure allows to DENCLUE an easy manipulation for the data, by using the cubes keys, and considering only populated cubes.

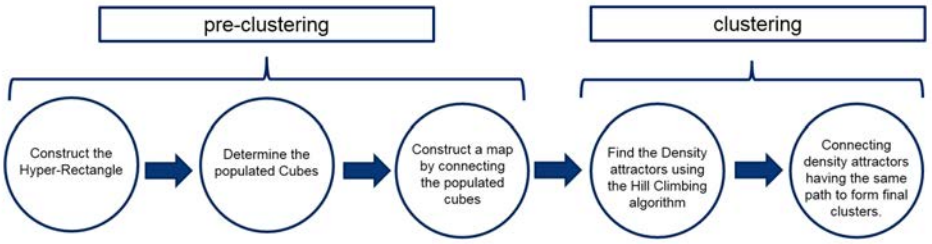

Fig. 2: DENCLUE process.

However, the use of Hill Climbing in DENCLUE presents limitations, in terms of the quality of clustering and the execution time. We highlight that the hill climbing doesn't converge exactly to the maximum, which just comes close. To overcome these limits, we have implemented in previous work [23] two algorithms : DENCLUE-SA and DENCLUE-GA. They are based on replacing the hill climbing algorithm by two promising metaheuristics algorithms : simulated annealing (SA) and genetic algorithm (GA). Despite that the two algorithms have a good clustering performance, they suffer in terms of runtime execution. In order to efficiently adapt DENCLUE algorithm in big data framework, we develop in this work an improved version of DENCLUE algorithm which we will call DENCLUE-IM. It is presented hereafter.

3. Proposed clustering method : DENCLUE-IM

As mentioned above, DENCLUE-IM is our amelioration of DENCLUE. This improvement consists in modifying the step based on the Hill Climbing algorithm. This step considered as crucial in DENCLUE algorithm is based on gradient calculations. These calculations are done for each point in order to find its density attractor. Make calculations for each point is not obvious to achieve results in a reasonable time, especially when it comes to operate on large databases. Our approach, as presented in algorithm 1, allows to find an equivalent item to the density attractor, which will represent all the points contained in a hyper-cube, instead of the calculations made for each point in the dataset (see equation 3). This representative of hyper-cube denoted x_{Hcube}, will be considered as the point having the highest density in this hyper-cube as shown in equation 4.

$$\forall x \in C_p \quad f_D(x) \leq f_D(x_{Hcube}), \tag{4}$$

where C_p is a given populated Hyper-cube in the constructed hyper-rectangle.
Thus each hyper-cube constitute an initial cluster represented by its x_{Hcube}. These clusters x_{Hcube} will be merged if and only if there exists a path between their representatives.

Algorithm 1. DENCLUE-IM algorithm

Input: The dataset, σ, and ξ
Step 1. Take dataset in a map whose each side is of 2σ, consider only populated cubes.
Step 2. Calculate the mean of each populated cubes.
Step 3. Find highly populated cubes.
Step 4. Determine the connection between each highly populated cube, and other cubes (Highly or just populated cubes) by the distance between their means. If $d(mean(c_1), mean(c_2)) < 4\sigma$, then the two cubes are connected.
Step 5. Only the highly populated cubes and cubes which are connected to a highly populated cube are considered in determining clusters.
Step 6. Find the representative of the hyper-cube.
Step 7. Connecting the representatives of hyper-cubes having the same path to form a cluster.
Output: Assignment of data values to clusters.

4. Results and discussion

4.1. Experimental setting

4.1.1. Experimental data

To evaluate the efficiency of DENCLUE-IM to cluster Voluminous data, we have used three datasets, Page Blocks, Spambase and Cloud services.

Page Blocks: This dataset, extracted from the UCI Machine Learning Repository, presents classified blocks of the page layout in a document that has been detected by a segmentation process[24].

Spambase: This dataset, also extracted from the UCI Machine Learning Repository illustrate classified Email as Spam or Non-Spam[25].

Cloud services: This data consists of 50 000 Cloud services, each one composed by 10 attributes[26].

The description of the used datasets is shown in Table 1.

Table 1: Description of datasets used in the experiments.

Dataset	# instances	# attributes	# classes
Page Blocks	5472	10	5
Spambase	4601	57	2
Cloud Services	50000	10	Unlabelled dataset

4.1.2. Validity metrics

There exist several validity metrics mentioned in the literature[27,17], which aim to evaluate the performance of clustering methods. These measures are including the Dunn Index[28], the Davies-Bouldin Index[29] and the Cluster Accuracy Index.

Dunn Index (DI): This index assesses the separation degree between individuals of the same cluster, that is to say the intra-cluster similarity. A high value indicates a better clustering.

Davies-Bouldin Index (DBI): This index, as DI, evaluates also the separation degree between clusters (inter-cluster dissimilarity), the smallest value indicates the better clustering.

Cluster Accuracy (CA): CA measures the percentage of correctly classified objects in a cluster, based on the pre-defined class labels. This index doesn't operate on unlabelled database, the high value indicates the best clustering quality.

4.2. Clustering performance

In this part we point out the interest of using DENCLUE-IM for big data clustering. For doing so, we compare the proposed approach against DENCLUE, DENCLUE-SA and DENCLUE-GA. For the first dataset, DENCLUE-IM has the second best DI, the best DBI, and very closed CA compared to the other algorithms results. In the Spambase dataset, DENCLUE-IM has the best DI behind DENCLUE-GA.
As for the DBI and CA, DENCLUE-IM has pretty good values, which are so closed to the three other algorithms. In the third dataset, our methods exceeds other algorithms in term of DI, and have the second best DBI comparing to other algorithms.
All these results conclude that our algorithm has an acceptable clustering performance.

Table 2: The comparison of the four algorithms according to their validity metrics.

Measures	Algorithms	Page Blocks	Spambase	Cloud Services
DI	DENCLUE	**0.721**	0.789	0.898
	DENCLUE-SA	**0.721**	0.821	0.898
	DENCLUE-GA	**0.721**	**0.835**	0.846
	DENCLUE-IM	0.693	0.831	**0.899**
DBI	DENCLUE	0.563	0.867	**1.254**
	DENCLUE-SA	0.474	**0.764**	1.714
	DENCLUE-GA	0.639	0.968	2.413
	DENCLUE-IM	**0.412**	1.041	1.262
CA	DENCLUE	**0.920**	**0.805**	—
	DENCLUE-SA	**0.920**	0.789	—
	DENCLUE-GA	0.916	0.718	—
	DENCLUE-IM	0.911	0.701	—

4.3. Runtime measurement

We study here the performance of our approach in terms of executions time. All algorithms are implemented in JAVA environment, on a Core i5 (2.70 GHz) PC with 8 GB of memory. Table 3 records the runtime of each method. It is observed that DENCLUE-IM is faster than the three other methods for the all used datasets.
For example, in the first dataset, DENCLUE-IM runtime is minimized by 12 times compared to the DENCLUE. As for the DENCLUE-SA and DENCLUE-GA, they require a runtime multiplied approximatively by 19 and 27 respectively, compared to the DENCLUE-IM.
In the other hand, in the third dataset which is Cloud Services, to classify all services, our approach requires around 28 minutes. Regarding to the three other algorithms, they require approximatively 32 hours which is equivalent to a runtime multiplied by 68 compared to our approach.

Table 3: Comparison between the algorithms according to their execution time (s).

Algorithms	Page block	Spambase	Cloud Services
DENCLUE	71.028	1285.911	116202.129
DENCLUE-SA	107.852	1347.818	115906.865
DENCLUE-GA	158.055	574.382	113650.162
DENCLUE-IM	**5.749**	**27.54**	**1675.508**

5. Conclusion

In this work, we have addressed the problem of clustering large dimensional datasets. We proposed a new density-based clustering algorithm, named DENCLUE-IM. It was developed to improve the capacity of the existing DEN-

CLUE algorithm, to operate on the massive data, which ensure the first V characterizing Big Data, Volume. By applying our approach on different large dimensional datasets, DENCLUE-IM has proved its efficiency by outperforming the run time of DENCLUE, DENCLUE-SA and DENCLUE-GA. Our new method gives also a pretty good quality of clustering according to the three used clustering validity metrics. We can underline that DENCLUE-IM guarantee the three Vs of the characteristics of Big Data, namely Volume, Variety and Veracity. Thus we proved that this approach found a trade-off between the quality of clustering and the runtime.
In our future work we will seek to develop a new clustering algorithm that satisfy all the criteria of big data, namely Vs mentioned in the literature.
It would be also interesting to study the parameters tuning of the proposed algorithm, as well to adapt the DENCLUE-IM to classify: 1) Entities forming the Mobile Ad-hoc Networks[30,31] and 2) Aircrafts in the Airports[32,33] or in the sky[34].

References

1. J. Gantz, D. Reinsel, The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east, IDC iView: IDC Analyze the Future 2007 (2012) 1–16.
2. M. Abourezq, A. Idrissi, Database-as-a-service for big data: An overview, International Journal of Advanced Computer Science and Applications (IJACSA) 7 (1).
3. Big Data definition in the Gartner IT Glossary, retrieved from, `http://www.gartner.com/it-glossary/big-data`, accessed: 27-January-2016.
4. J. Kalibjian, " big data" management and security application to telemetry data products, in: International Telemetering Conference Proceedings, International Foundation for Telemetering, 2013.
5. Geoinformatics, Department of Civil Engineering, IIT Kanpur, retrieved from, `http://gi.iitk.ac.in/gi/geoinformatics`, accessed: 17-January-2016.
6. H. U. Buhl, M. Röglinger, D.-K. F. Moser, J. Heidemann, Big data, Business & Information Systems Engineering 5 (2) (2013) 65–69.
7. J. Gantz, D. Reinsel, Extracting value from chaos, IDC iview (1142) (2011) 9–10.
8. N. B. D. PWG, Draft nist big data interoperability framework, Reference Architecture.
9. M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise., in: Kdd, Vol. 96, 1996, pp. 226–231.
10. M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure, in: ACM Sigmod Record, Vol. 28, ACM, 1999, pp. 49–60.
11. D. Birant, A. Kut, St-dbscan: An algorithm for clustering spatial–temporal data, Data & Knowledge Engineering 60 (1) (2007) 208–221.
12. Y. He, H. Tan, W. Luo, H. Mao, D. Ma, S. Feng, J. Fan, Mr-dbscan: An efficient parallel density-based clustering algorithm using mapreduce, in: Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on, IEEE, 2011, pp. 473–480.
13. A. Hinneburg, D. A. Keim, An efficient approach to clustering in large multimedia databases with noise, in: KDD, Vol. 98, 1998, pp. 58–65.
14. P. Berkhin, A survey of clustering data mining techniques, in: Grouping multidimensional data, Springer, 2006, pp. 25–71.
15. R. Xu, D. Wunsch, et al., Survey of clustering algorithms, Neural Networks, IEEE Transactions on 16 (3) (2005) 645–678.
16. A. K. Jain, A. Topchy, M. H. Law, J. M. Buhmann, Landscape of clustering algorithms, in: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, Vol. 1, IEEE, 2004, pp. 260–263.
17. A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, A. Bouras, A survey of clustering algorithms for big data: Taxonomy and empirical analysis, Emerging Topics in Computing, IEEE Transactions on 2 (3) (2014) 267–279.
18. G. H. Shah, C. Bhensdadia, A. P. Ganatra, An empirical evaluation of density-based clustering techniques, International Journal of Soft Computing and Engineering (IJSCE) ISSN (2012) 2231–2307.
19. C. Ding, X. He, Cluster merging and splitting in hierarchical clustering algorithms, in: IEEE International Conference on Data Mining (ICDM'02), IEEE, 2002, pp. 139–146.
20. E. Parzen, On estimation of a probability density function and mode, The annals of mathematical statistics (1962) 1065–1076.
21. M. Rosenblatt, et al., Remarks on some nonparametric estimates of a density function, The Annals of Mathematical Statistics 27 (3) (1956) 832–837.
22. M. J. Zaki, W. Meira Jr, Data mining and analysis: fundamental concepts and algorithms, Cambridge University Press, 2014.
23. A. Idrissi, H. Rehioui, A. Laghrissi, S. Retal, An improved denclue algorithm for data clustering, in: IEEE 2015 International Conference on Information and Communication Technology and Accessibility, 2015.
24. F. Esposito, D. Malerba, G. Semeraro, Multistrategy learning for document recognition, Applied Artificial Intelligence an International Journal 8 (1) (1994) 33–84.
25. M. Hopkins, E. Reeber, G. Forman, J. Suermondt, Spam base dataset, Hewlett-Packard Labs.
26. M. Abourezq, A. Idrissi, Integration of Qos aspects in the cloud service research and selection system, International Journal of Advanced Computer Science and Applications 6 (6).
27. X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, in: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, AAAI Press, 2013, pp. 2598–2604.
28. J. C. Dunn, Well-separated clusters and optimal fuzzy partitions, Journal of cybernetics 4 (1) (1974) 95–104.
29. D. L. Davies, D. W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence (2) (1979) 224–227.

30. A. Idrissi, How to minimize the energy consumption in mobile ad-hoc networks, International Journal of Artificial Intelligence and Applications (IJAIA) 3 (2).
31. A. Idrissi, C. M. Li, J. F. Myoupo, An algorithm for a constraint optimization problem in mobile ad-hoc networks, in: 18th IEEE International Conference on Tools with Artificial Intelligence, ICTAI06, Washington, USA, 2006.
32. A. Idrissi, Some methods to treat capacity allocation problems, Journal of Theoretical and Applied Information Technology 37 (2) (2012) 141–158.
33. A. Idrissi, C. M. Li, Modeling and optimization of the capacity allocation problem with constraints, in: 4th IEEE Proceedings of International Conference on Computer Science, RIVF'06, Ho Chi Min City, 2006.
34. A. Idrissi, Casc: Decision support for aerial controller by constraint satisfaction, 3rd International Conference on Computer Science, RIVF'05, Hanoi, Vietnam.