Zayed University
ZU Scholars

All Works

1-1-2022

A Brief Comparison of K-means and Agglomerative Hierarchical Clustering Algorithms on Small Datasets

Hassan I. Abdalla
Zayed University

Follow this and additional works at: https://zuscholars.zu.ac.ae/works

 Part of the Computer Sciences Commons

Recommended Citation

Abdalla, Hassan I., "A Brief Comparison of K-means and Agglomerative Hierarchical Clustering Algorithms on Small Datasets" (2022). *All Works*. 5203.
https://zuscholars.zu.ac.ae/works/5203

This Conference Proceeding is brought to you for free and open access by ZU Scholars. It has been accepted for inclusion in All Works by an authorized administrator of ZU Scholars. For more information, please contact scholars@zu.ac.ae.

A Brief Comparison of K-means and Agglomerative Hierarchical Clustering Algorithms on Small Datasets

Hassan I. Abdalla(✉)

College of Technological Innovation, Zayed University, P.O. Box 144534, Abu Dhabi, UAE
hassan.abdalla@zu.ac.ae

Abstract. In this work, the agglomerative hierarchical clustering and K-means clustering algorithms are implemented on small datasets. Considering that the selection of the similarity measure is a vital factor in data clustering, two measures are used in this study - cosine similarity measure and Euclidean distance - along with two evaluation metrics - entropy and purity - to assess the clustering quality. The datasets used in this work are taken from UCI machine learning depository. The experimental results indicate that k-means clustering outperformed hierarchical clustering in terms of entropy and purity using cosine similarity measure. However, hierarchical clustering outperformed k-means clustering using Euclidean distance. It is noted that performance of clustering algorithm is highly dependent on the similarity measure. Moreover, as the number of clusters gets reasonably increased, the clustering algorithms' performance gets higher.

Keywords: Clustering · K-means · Hierarchical clustering · Clustering comparison · Cosine · Euclidean

1 Introduction

Clustering algorithms are a vital techniques of machine learning, and are widely used in almost all scientific application including databases [1, 2], collaborative filtering [3], text classification [4], indexing, etc. The clustering is an automatic process of assembling of data points into similar assembles so that points in the same cluster are highly similar to each other, and maximally dissimilar to points in other assembles. With the constantly-increasing volumes of daily data and information, clustering is being undeniably helpful technique in organizing collections of data for an efficient and effective navigation [1]. However, with the dynamic characteristics of the collected data, the clustering algorithms have to be able to cope and deal with the newly-added data in every second so it would help in discovering knowledge effectively and timely. As one of the most commonly known techniques for the unsupervised learning, clustering comes with the main objective finding the natural clusters among the assigned patterns. It simply groups data points into categories of similar points.

This paper is organized as follows: in Sect. 2, related work is briefly covered. Section 3 covers methodology including clustering algorithms and similarity measures used in

© The Author(s) 2022
Z. Qian et al. (Eds.): WCNA 2021, LNEE 942, pp. 623–632, 2022.
https://doi.org/10.1007/978-981-19-2456-9_64

this work. Section 3 introduces performance evaluation including experimental setup, datasets description, evaluation metrics and results. Discussion is concisely covered in Sect. 4. Finally, conclusions and future work is given in Sect. 5.

2 Related Work

In literature, the Hierarchical clustering is often seen to give solutions of better quality than k-means. However, it is limited due to its complexity in terms of quadratic time. Opposed to hierarchical, K-means has a linear time complexity. It is linear in the number of points to be assigned. However, it is seen to give inferior clusters comparing with hierarchical. Most of earlier works used both algorithms with K-means algorithm (with Euclidean distance) is used more frequently to assemble the given data points. In its nature, K-means is linked with the finding of centroids. The centroids comes from the Euclidean Geometry itself. K-means also enjoys its being scalable and more accurate than hierarchical clustering algorithm chiefly for document clustering [5].

In [5], on the other hand, the experimental results of agglomerative hierarchical and K-means clustering techniques were presented. The results showed that hierarchical is better than k-means in producing clusters of high quality. In [6] authors compared two similarity measures - cosine and fuzzy similarity measures - using the k-means clustering algorithm. The results showed that fuzzy similarity measure is better than cosine similarity in terms of time and clustering solutions quality. In [7], several measures for text clustering were described approaches using affinity propagation. In [8] different clustering algorithms were explained and implemented on text clustering. In [9] some problems that that text clustering have been facing was explained. Some key algorithms, and their merits and des-merits were discussed in details. The feature selection and the similarity measure were the corner stones for proposing an effective clustering algorithm.

3 Methodology

3.1 Term Weighting

The Term Frequency (TFIDF) technique, as the most widely used, of weighting is adapted in this work.

3.2 K-Means Clustering Algorithm

The k-means clustering algorithm is widely used in data mining [1, 4] for its being more efficient than hierarchical clustering algorithm. It is used in our work as follows;

1. The number of clusters is one of these K values [2, 4]. That means K-means is run three times with one different K value each time.
2. The centroids has been chosen at first step randomly.
3. The standard k-means is run by getting all the data points involved in the first loop. The results are saved for next iteration and centroids are modified. Then, the clustering process run over for successive iteration by setting all points of clusters free, and randomly selecting new centroids.
4. Step 3 is iteratively continued till either number of iterations reach 30 iterations or each cluster has been seen in stable state.

3.3 The Hierarchical Clustering (HC)

Initialization: Given a set of points N, the data point matrix between points, initial clusters were initiated by randomly picking head for each cluster [10]. Then, in each loop, for any new data point, the data point cost between the new point and each cluster is calculated. The cluster whose average cost is the lowest would contain the relative point at hand. The step (1) is repeated till all points were clustered. Like K-means, number of clusters is selected to be one of these K values [2, 4]. That means hierarchical clustering is run three times with one different K value each time.

3.4 Similarity Measures

The similarity measures, used in this study, are Cosine and Euclidean [1].

Euclidean Distance (ED). In ED, each document is seen as a point in 2D space based on the term frequency of N terms that would represent the N dimension. ED measures the similarity between each point pair in this space using their coordinate based on the following equation:

$$D_{Euc}(x, y) = \sum \sqrt{x1 - y1)^2 + x2 - y2)^2 + \cdots xn - yn)^2} \tag{1}$$

Cosine Similarity Measure. The Cosine similarity, as one of the most widely-used measure, computes the pairwise similarity between ach document pair using the dot product and the magnitude of both vectors of both documents. It is computed as follows:

$$Sim_{Cos}(x, y) = \frac{\sum_{i=1}^{n} (x * y)}{\sqrt{\sum_{i=1}^{n} x^2} * \sqrt{\sum_{i=1}^{n} y^2}} \tag{2}$$

The union is used to normalize the inner product. Where x and y are the point pair needed to be clustered.

3.5 Experimental Setup

Machine Description. Table 1 displays the machine and environment descriptions used to perform this work.

Table 1. Machine and environment description.

Task	Tool	Specification
Clustering	Language	Python 3, Development Software: Jupyter Notebook
	OS	Windows 8 (64 bit)
	Memory	RAM 4 GB
	CPU	Intel I Core ™ (i5)
	Dataset	Glass & Iris

3.6 Dataset Description

Tables 2, 3 hold the datasets description which is taken literally from UCI (Machine Learning Repository).

Table 2. Iris dataset

Dataset characteristics:	Multivariate	Number of instances:	150	Area	Life
Attribute characteristics:	Real	Number of attributes:	4	Date donated	1988–07-01
Associated tasks:	Classification	Missing values?	No	Number of web hits:	3536252

Table 3. Glass identification dataset

Data set characteristics:	Multivariate	Number of instances:	214
Attribute characteristics:	Real	Number of attributes:	10
Associated tasks:	Classification	Missing Values?	No

3.7 The Clustering Evaluation Criterions

The evaluation metrics used to assess clustering quality are Entropy and Purity.

Purity (also known as Accuracy): It determines how large the intra-cluster is, and how less the inter-cluster is [1]. In other words, it is use to evaluates how much coherent the clustering solution is, and is formulated as follows;

$$\text{Purity} = \frac{1}{N}\sum_{i=1}^{k} max_j |c_i \cap t_j| \tag{3}$$

where N is the number of objects (data points), k is the number of clusters, ci is a cluster in C, and tj is the classification which has the max count for cluster ci.

Entropy. It is used to measure the extent to which a cluster contain single class and not multiple classes. It is formulated as follows:

$$Entropy = \sum_{i=1}^{c} ci * log(ci) \tag{4}$$

Unlike purity, the best value of entropy is "0" and the worst value is "1".

4 Results and Discussion

In this section, we provide the obtained results of running both algorithms on both datasets using both measures – Cosine and Euclidean. Three K values for clusters – 2, 4, and 8 – along with using two evaluation metrics.

Table 4. Iris dataset - Cosine

AHC			
Metric/K	2	4	8
Entropy	4.60517	4.60937	**3.70626**
Purity	0.66667	0.66667	0.68
K-means			
Metric/K	2	4	8
Entropy	4.60517	**4.47621**	4.81686
Purity	0.66667	**0.97333**	**0.95333**

Table 5. Iris dataset - Euclidean

AHC			
Metric/K	2	4	8
Entropy	**3.91202**	**3.93659**	**3.82572**
Purity	0.66667	0.68667	0.7
K-means			
Metric/K	2	4	8
Entropy	3.97029	4.68630	4.7789
Purity	0.66667	**0.88667**	**0.97333**

For Iris dataset, k-means with cosine outperformed AHC. However, AHC with Euclidean outperformed k-means. On the other hand, for Glass dataset, AHC with cosine

Table 6. Glass dataset - Cosine

AHC			
Metric/K	2	4	8
Entropy	**4.72739**	**4.60619**	**4.62534**
Purity	0.48131	0.49065	0.53738
K-means			
Metric/K	2	4	8
Entropy	4.96284	4.99857	5.09285
Purity	**0.67757**	**0.71963**	**0.85981**

Table 7. Glass dataset - Euclidean

AHC			
Metric/K	2	4	8
Entropy	0.69315	**4.93907**	**4.85886**
Purity	0.36449	0.62617	0.67290
K-means			
Metric/K	2	4	8
Entropy	**4.68213**	4.98090	5.09710
Purity	**0.51402**	**0.74766**	**0.83178**

and Euclidean outperformed k-means in terms of entropy. In contrast, k-means outweighed AHC in terms of purity for both cosine and Euclidean. If we took this analysis as points for both algorithm, Table would hold these points.

Table 8. K-means and AHC in points

AHC		
Dataset/Measure	Cosine	Euclidean
Iris	**0**	**1**
Glass	1	1
K-means		
Dataset/Measure	Cosine	Euclidean
Iris	1	0
Glass	1	**1**

From Table 8, it can be noted that both algorithms have similar trend performance on both datasets. However, AHC preferred giving smaller entropy than k-mean, when k-means preferred giving higher purity.

In next Tables 9, 10, 11 and 12, Mean and Standard Deviation (STD) of both Entropy and Purity were taken in an average of all K values (2, 4, and 8) of each algorithm with respect to each evaluation metric -Entropy and Purity. Booth Mean and STD are interpreted using the basic values of entropy and purity that are drawn in Tables 4, 5, 6 and 7).

Table 9. Iris dataset - Cosine

AHC		
	Mean	STD
Entropy	4.30693	0.42474
Purity	0.67111	0.00629
K-means		
Metric/K	Mean	STD
Entropy	4.63275	0.14043
Purity	0.86444	0.14009

Table 10. Iris dataset - Euclidean

AHC		
	Mean	STD
Entropy	3.89144	0.04754
Purity	0.68444	0.01370
K-means		
Metric/K	Mean	STD
Entropy	4.47851	0.36135
Purity	0.84222	0.12908

Table 11. Glass dataset - Cosine

AHC		
	Mean	STD
Entropy	4.65297	0.05320
Purity	0.50312	0.02453

(*continued*)

Table 11. (*continued*)

K-means		
Metric/K	Mean	STD
Entropy	5.01809	0.05484
Purity	0.75234	0.07791

Table 12. Glass dataset – Euclidean

AHC		
	Mean	STD
Entropy	3.49703	1.98291
Purity	0.55452	0.13572
K-means		
Metric/K	Mean	STD
Entropy	4.92035	0.17509
Purity	0.69782	0.13443

Mean (Purity) in k-means is always better than AHC. However, Mean (Entropy) in AHC is always better than K-means. This confirms our previous analysis that AHC always produces solutions of lower entropy and K-means always gives solutions of higher purity. However, STD in AHC is better than K-means on both Iris and Glass datasets for both Euclidean and Cosine respectively. On the other hand, K-means is better than AHC on both Iris and Glass datasets for both Cosine and Euclidean respectively. As a rule of thumb, when STD is >=1, that implies a relatively high variation. However, when STD <=1, it is seen low. This means that the distributions with STD higher than 1 are seen of high variance whereas those with STD lower than 1 are seen of low-variance. In General, STD is better when it is kept as much low as possible which means that data has less variations around the mean with different K values for clusters.

5 Conclusions and Future Work

In this paper, we tried to briefly investigate the behavior of hierarchical and k-means clustering algorithms using cosine similarity measure and Euclidean distance along with using two evaluation metrics – Entropy and Purity. In general, AHC produced clustering solution of lower entropy than k-means. In contrast, k-means produced clustering solution of higher purity than AHC. Both algorithms look to have a similar performance trend on both datasets with AHC being slightly superior in terms of clustering solution quality. On the other hand, although we have not discussed the run time, we found from experiments that AHC suffers from the computational complexity comparing with K-means which was faster. However, the hierarchical clustering produced a clustering

solutions of slightly high-quality than K-means. As a matter of fact, the performance of both algorithms on both "small" datasets could not be taken as a decisive factor for the report on behavior of both algorithm.

Therefore, the future work is directed towards extending this study significantly by: (1) Proposing new clustering algorithm, (2) including medium-sized and big datasets, (3) investigating more similarity measures [12], (4) considering more evaluation metrics, and finally, (5) studying one more clustering algorithm [13]. The ultimate aim of future work is to draw a valuable comparison study between all algorithms on target datasets so that the best combination of clustering algorithm and the relative similarity measure is captured. Moreover, the effect of using a different incremental number of clusters "K" is investigated.

Acknowledgments. The author would like to thank and appreciate the support received from the Research Office of Zayed University for providing the necessary facilities to accomplish this work. This research has been supported by Research Incentive Fund (RIF) Grant Activity Code: R20056–Zayed University, UAE.

References

1. Amer, A.A.: On K-means clustering-based approach for DDBSs design. J. Big Data **7**(1), 1–31 (2020). https://doi.org/10.1186/s40537-020-00306-9
2. Amer, A., Mohamed, M., Al_Asri, K.: ASGOP: an aggregated similarity-based greedy-oriented approach for relational DDBSs design. Heliyon **6**(1), e03172 (2020)
3. Amer, A., Abdalla, H., Nguyen, L.: Enhancing recommendation systems performance using highly-effective similarity measures. Knowl.-Based Syst. **217**, 106842 (2021)
4. Amer, A.A., Abdalla, H.I.: A set theory based similarity measure for text clustering and classification. J. Big Data **7**(1), 1–43 (2020). https://doi.org/10.1186/s40537-020-00344-3
5. Lee, C., Hung, C., Lee, S.: A comparative study on clustering algorithms. In: 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Honolulu, HI, pp. 557–562 (2013)
6. Scheunders, P.: A comparison of clustering algorithms applied to color image quantization. Pattern Recogn. Lett. **18**(11–13), 1379–1384 (1997)
7. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining, vol. 400, pp. 1–2 (2000)
8. Goyal, M., Agrawal, N., Sarma, M., Kalita, N.: Comparison clustering using cosine and fuzzy set based similarity measures of text documents. arXiv, abs/1505.00168 (2015)
9. Kumar, S., Rana, J., Jain, R.: Text document clustering based on phrase similarity using affinity propagation. Int. J. Comput. Appl. **61**(18), 38–44 (2013)
10. Kamble, R., Sayeeda, M.: Clustering software methods and comparison. Int. J. Comput. Technol. Appl. **5**(6), 1878–1885 (2014)
11. Xu, D., Tian, Y.: A comprehensive survey of clustering algorithms. Ann. Data Sci. **2**(2), 165–193 (2015). https://doi.org/10.1007/s40745-015-0040-1
12. Abdalla, H., Amer, A.: Boolean logic algebra driven similarity measure for text based applications. PeerJ Comput. Sci. **7**, e641 (2021)
13. Abdalla, H., Artoli, A.: Towards an efficient data fragmentation, allocation, and clustering approach in a distributed environment. Information **10**(3), 112 (2019)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.