


Neighboring-Aware Hierarchical Clustering: A New Algorithm and Extensive Evaluation


Ali A. Amer, Zayed University, UAE*

 <https://orcid.org/0000-0002-2002-948X>


Muna Al-Razgan, King Saud University, Saudi Arabia

Hassan I. Abdalla, Zayed University, UAE

Mahfoudh Al-Asaly, King Saud University, Saudi Arabia

 <https://orcid.org/0000-0002-4558-5394>

Taha Alfakih, King Saud University, Saudi Arabia

 <https://orcid.org/0000-0003-0366-5932>

Muneer Al-Hammadi, Norwegian University of Science and Technology, Norway

ABSTRACT

In this work, a simple yet robust neighboring-aware hierarchical-based clustering approach (NHC) is developed. NHC employs its dynamic technique to take into account the surroundings of each point when clustering, making it extremely competitive. NHC offers a straightforward design and reliable clustering. It comprises two key techniques, namely, neighboring-aware and filtering and merging. While the proposed neighboring-aware technique helps find the most coherent clusters, filtering and merging help reach the desired number of clusters during the clustering process. The NHC's performance, which includes all evaluation metrics and run time, has been thoroughly tested against nine clustering rivals using four similarity measures on several real-world numerical and textual datasets. The evaluation is done in two phases. First, we compare NHC to three common clustering methods and show its efficacy through empirical analysis. Second, a comparison with six relevant, contemporary competitors highlights NHC's extremely competitive performance.

KEYWORDS

Hierarchical Clustering, Information Systems, K-means, Machine Learning, Neighboring Clustering, Partitional Clustering

INTRODUCTION

Clustering is an unsupervised method that divides the unlabeled data points into various groups based on distance metrics (or similarity measures). It is frequently utilized in a variety of fields, including knowledge discovery through machine learning (ML), information retrieval (IR), and data mining. By grouping related data points (or semantically important groupings of points or documents) into meaningful clusters and dispersing the dissimilar points over numerous clusters, clustering provides

DOI: 10.4018/IJSWIS.346377

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

intuitive navigation and browsing options. Several scientific applications have used a variety of clustering algorithms that were developed over the past 40 years, including mixed data (Kuwil et al., 2019), database system design (Abdalla et al., 2023; Fernández & Gómez, 2021), recommendation systems (Akilandeswari et al., 2022), data and text classification (Gilpin & Davidson, 2017; Hussain & Haris, 2019; Salem et al., 2018; Steinbach et al., 2000), high-dimensional data space (Chander et al., 2022), indexing (Zhu & Ma, 2018), and word embedding-based text clustering (Gong et al., 2018). As a general rule, while clustering data points (or documents) are divided into k clusters, related points (or documents) are placed in the same cluster, and different data points are placed in different clusters. In actuality, the complexity inherent in the clustering of documents, especially during the past 30 years, has caught the researchers' attention. Consequently, the search for semantically meaningful groups of documents is still in full gear for scholars.

Two clustering categories are widely popular, namely, partitional and hierarchical, which have achieved significant results in applications across multiple domains. Overall, partitional clustering is more dynamic and efficient than hierarchical clustering. Indeed, in partitional clustering, the points of data can be migrated from one cluster to another smoothly. Moreover, they can be linked with knowledge connected with clusters' size and shape via leveraging distance measurements with suitable prototypes (Kuwil et al., 2019). However, the demerits of partitional clustering are: (1) Most algorithms use optimization techniques to solve their problems with initialization and the number of clusters; (2) their iterative mechanism makes them susceptible to local minima and prone to cluster initialization, leading to the failure to find the best or optimal solutions; (3) they are highly sensitive to both noise and outliers, and determining the k clusters has long been expressed as challenging tasks; (4) their dimensionality negatively affects the efficacy of partitional category. Further, since performance is heavily dependent on the initial centroids and the number of clusters, the majority of them typically experience performance fluctuations (Gong et al., 2018; Kuwil et al., 2019). The k -means algorithm is one of the most used clustering algorithms (Abdalla et al., 2023; Arthur & Vassilvitskii, 2007). The k parameter in the k -means algorithm must be established as the number of clusters in the first phase. The k -means will then select k randomly chosen numbers as centroids using the k value. The initial data point for each cluster are these centroids, which are employed in the first phase. The computations to optimize the centroids' placements are carried out repeatedly until the centroids are stabilized by iteratively running k -means. The ultimate clustering solutions will then be produced using the stabilized centroids.

On the other side, without knowing the exact number of clusters (k), hierarchical algorithms are used to draw the hierarchy of the clusters (Kuwil et al., 2020; Liu & Frank, 2022; Zhang et al., 2015; Zhang P. et al., 2019; Zhang W. et al., 2019). The majority of these algorithms, including minimum-spanning tree-based clustering algorithms (MSTs), are suitable for massive, contemporary datasets with high dimensions (Wang et al., 2013). Due to the drawbacks of partitional methods described above, the results of hierarchical clustering are generally more reliable than those of partitional algorithms. The performance of the partitional algorithm, in particular k -means and its derivations, suffers from these flaws despite ongoing efforts to address them, such as k -means++ (Arthur & Vassilvitskii, 2017), particularly when data are sparse and high-dimensional. However, the building of the model requires a quadratic execution time for the hierarchical methods, which makes them slower than partitional clustering (Zhang P. et al., 2019). The resulting clusters may also contain various densities. Last but not least, outliers that affect cluster separations frequently result in a definite decline in clustering quality (Yang, 2022). One of the most widely used methods in this category is agglomerative hierarchical clustering (AHC) (Abdalla et al., 2023). The number of clusters for AHC, such as k -means, can be set at k , or the algorithm can be allowed to create any number of clusters it likes.

Even though extant literature includes many clustering studies, the majority of these have the following shortcomings:

1. They primarily focused on either IR datasets (e.g., documents) or non-IR datasets (e.g., those taken from the UCI repository as numerical ones). Therefore, when both IR and non-IR datasets are taken into account, the claim for the superiority of one algorithm over another over both data types is still under investigation, or at least not adequately convincing.
2. If previous studies did exist, they tended to employ tiny non-IR datasets and were rarely found to conduct clustering trials using more than one similarity metric. The Euclidean distance, or cosine similarity measure, was almost exclusively utilized in the vast majority of earlier publications. Thus, there is still strong evidence that any metric recorded will be the best fit in all circumstances. The choice of an appropriate distance metric or similarity measure is based on the situation and the dataset's characteristics.
3. Because of the lengthy execution time, some crucial measures, such as the silhouette, have hardly ever been leveraged to assess the effectiveness of clustering methods focused primarily on huge data. To the best of the authors' knowledge, besides presenting a new hierarchically-driven clustering algorithm, no other works have offered a thorough evaluation of multiple clustering algorithms along with several similarity measures over various datasets in the same pattern the authors did in this study. The main goal of summarizing the findings of such in-depth analysis is to offer comprehensive knowledge of the clustering technique on both IR and non-IR datasets.

In summary, the primary contributions of this work are the development of a novel neighboring-aware hierarchical-based clustering approach (NHC). The NHC distinguishes itself from its rivals with three features: (1) The proposed neighboring-aware strategy helps find the most coherent clusters; (2) the filtering and merging techniques help reach the desired number of clusters during the clustering process; (3) the robustness and efficiency of the NHC was thoroughly tested using multiple similarity measures over several real-world numerical and textual datasets (from various applications), which had not been done in any previous study in the literature; (4) the authors analyzed, compared, and benchmarked the clustering results of the proposed algorithm against well-known state-of-the-art clustering techniques. Based on the observed results, the NHC can be regarded as a solution that strikes a balance between the effectiveness of hierarchical clustering and the efficiency of partitional clustering. It combines the efficient clustering solutions attained by the hierarchy with the quick computation speed of partitional methods. Furthermore, as similarity metrics are important for clustering, the authors used four similarity measures to monitor the performance of the algorithms in different settings. Most importantly, the researchers compared all of the algorithms in terms of run time to ascertain their efficiency. The primary goal of the first phase was to compare the NHC with three popular clustering algorithms: K-means, Bisect K-means techniques, and AHC. The second phase focused on evaluating the NHC algorithm's performance against six cutting-edge clustering algorithms. The algorithms the authors used for comparison are incredibly relevant to this study. Throughout this research, the authors referred to numerical datasets (acquired from the UCI repository) as non-IR datasets, whereas they referred to document collections (text datasets) as IR datasets.

The rest of this paper is organized as follows: The second section provides the literature review; the third section offers a description of the proposed methodology of the NHC algorithm; the fourth section presents the experimental setup; the fifth section provides the outcomes of two evaluation phases; the sixth section consists in the authors' brief explanation and discussion of the behavior of all algorithms; finally, the seventh section offers the authors' conclusions and future research directions.

RELATED WORK

Over the past few decades, data clustering was extensively studied in a variety of fields. The two main categories of clustering algorithms are hierarchical and partitioning methods (Abdalla et al.,

2023; Arthur & Vassilvitskii, 2007; Fernández & Gómez, 2021; Hussain & Haris, 2019; Kuwil et al., 2019; Salem et al., 2018; Wang et al., 2013). Hierarchical algorithms function by grouping the data points (e.g., documents) into cluster trees (Fernández & Gómez, 2021). Depending on whether the hierarchical division is built from the bottom up or from the top down, the hierarchical algorithms can be further divided into agglomerative and divisive clustering. On the other hand, k-means and its variations (Abdalla et al., 2023; Kuwil et al., 2019) are the most widely used partitioning algorithms (Hussain et al., 2019), which is one-level partitioning.

Document clustering is still the most often used method in the fields of IR and natural language processing to assemble semantically linked data or documents that have a similar or closer topic of interest. In the same regard, some scholars put out studies for the clustering of numerical data (Hussain et al., 2019; Kuwil et al., 2019; Yang, 2022; Zhang et al., 2015). For instance, Kuwil et al. (2019) used the Euclidean distance to propose the critical distance clustering algorithm (CDC). The CDC features a straightforward but adaptable design. Comparable techniques, including MST-based clustering, k-means, and DbSCAN, were demonstrated to be slightly inferior to the CDC in some cases (e.g., education and oil datasets). The fundamental advantage of the CDC is that no predetermined guidelines are needed. Therefore, in this study, the authors included the CDC in their comparison analysis. Based on the local data gravity, Zhang P. et al. (2019) proposed the neighboring graphic hierarchical clustering (NGHC) as a hierarchical clustering of complicated design. Zhang P. et al. separated the dataset into groups using the gravity-based clustering method in the NGHC as intermediate findings. Then, they used a new linkage measure to combine those intermediary groups. They would combine the data gravitation between the two groups in this way until they obtained the appropriate findings. Nevertheless, the authors utilized only two datasets for evaluation, and they did not specify whether the gravity-based clustering method is the best option for either type of data.

On the other extreme, Hussain et al. (2019) developed the k-means-based co-clustering (kCC) algorithm as an improved variant of k-means through embedding higher-order statistics and data dualism principles. The researchers chose multiple points to represent each cluster's centers during the initialization phase. They also created neighborhood walk statistics as a semantic similarity for center re-estimation as well as cluster assignment in the iterative process. They evaluated kCC's effectiveness across several text datasets; the results revealed that it performed competitively against its rivals. Thus, in this study, the authors included kCC in their experimental investigation. On the other hand, some researchers focused on document clustering (Steinbach et al., 2000; Liu & Frank, 2022) and represented features in the term frequency-inverse document frequency (TF-IDF) matrix. Steinbach et al. (2000) published a technical study on the behavior of clustering algorithms in the clustering of IR datasets. They emphasized that bisecting k-means is superior to k-means and, in some situations, equal to or better than AHC. However, as the authors previously indicated, Steinbach et al. made these judgments only based on document collections (i.e., IR dataset). Nasim and Haider (2020) conducted an experimental investigation to determine the ideal clustering technique for Bahasa Indonesia. They tested three clustering algorithms, namely, k-means, k-means++, and AHC. The outcomes demonstrated that AHC was outperformed by the k-means and k-means++ algorithms. However, due to the short corpus Nasim and Haider employed (approximately 100 documents), their study is untrustworthy.

Other authors researched how similarity measures affect data clustering (Ljubešić et al., 2008; Patil & Thakur, 2018; Strehl et al., 2000). However, in these studies, they used only small datasets in the experiments; also, the findings showed that cosine and Jaccard similarity performed best, while Euclidean distance performed the worst for data clustering. On the same page is using the k-means clustering approach and Kullback-Leibler divergence (KLD) (Huang, 2008) that expanded (Strehl et al., 2000). Results showed that Jaccard similarity was more cohesive and KLD was more accurate for clustering, while Euclidean distance was the worst. This goes in direct and complete opposition to the overwhelming majority of clustering literature, which identified Euclidean as one of the most competitive metrics for clustering analysis (Abdalla & Amer, 2021). For example, Ljubešić et

al.'s (2008) trials using eight similarity measures demonstrated that Jenson-Shannon divergence, Manhattan, and Euclidean distances performed better than more common measures, such as cosine and Jaccard similarity. The authors' findings in this study come to support Ljubešić et al.'s (2008) and Abdalla and Amer's (2021) outcomes, in which Euclidean and Manhattan have been shown to be more competitive than cosine and KLD. Forsyth and Sharoff (2014) presented a method for assessing the effectiveness of similarity metrics using human judgment. Results showed that Pearson correlation outperformed both cosine and KLD. Moreover, the usage of tiny datasets with only about 43 documents in several works (Ahlgren & Colliander, 2009; Ahlgren & Jarneving, 2008), rendered them incomparable and less robust for large-scale document clustering.

Meanwhile, several recent studies assessed these algorithms while collecting tweets (Curiskis et al., 2020; Selvam et al., 2018). These studies demonstrated the superiority of TF-IDF-based k-means over AHC, bisecting k-means, and even k-medoids clustering methods (Shamir & Tishby, 2010). For instance, Nasim and Haider (2020) assessed three clustering algorithms (i.e., k-means, bisecting k-means, and affinity propagation) employing a variety of feature representations, such as TF-IDF and word embedding, across the corpus of Urdu tweets. The outcomes demonstrated that the k-means algorithm based on TF-IDF behaved the best. Nevertheless, Nasim and Haider examined a small corpus of tweets in Urdu, making their study unreliable. Shamir and Tishby (2010) applied k-means clustering to Urdu documents using several similarity measures, including cosine similarity, Jaccard similarity, and Levenshtein distance. Results showed that the Jaccard-based k-means algorithm behaved the best and was shown to be more effective, chiefly in terms of purity scores. Contrarily, Amalia et al. (2020) evaluated AHC and k-means algorithms using several similarity measures in document collection. Their study showed that AHC is better than k-means. As noted, some works showed k-means better, while others drew AHC as the best. Thus, results are conflicting and insights are contradicting, which makes it highly difficult for the user to choose the best-fit combination for the clustering analysis.

Meanwhile, Goyal et al. (2015) used the k-means clustering technique to examine two similarity measures, that is, cosine and fuzzy similarity measures. According to the findings, the fuzzy similarity measure performs better in terms of time and clustering solution quality than the cosine similarity measure. Xu and Tian (2015) outlined a few issues in text clustering. They examined the benefits and drawbacks of a few important algorithms. The foundation for the proposal of an efficient clustering algorithm was feature selection and similarity measures. Nguyen et al. (2019) presented variable entropy and variable mutability, which are two novel similarity measures based on the data's variability. The authors used a hierarchical approach to cluster data, and they compared their algorithm to 11 other ones. Exactly as in this study, Nguyen et al. (2019) proved that the properties of the dataset affect how well similarity-based algorithms function given the experimental findings. In other words, neither an optimal clustering technique nor a dominant similarity metric exist. The target application has a significant impact on how well the clustering method and similarity metric perform together. Finally, it is important to note that the authors did not include MST-based techniques in general (Şaar & Topcu, 2022) or those that use metaheuristics or evolutionary algorithms (Halim & Uzma, 2018), as a comprehensive analysis of data clustering algorithms is outside the purview of this work. Therefore, the authors included and discussed the research that is most important to the study in this paper.

As the authors discussed above, in extant literature, scholars consistently presented several clustering algorithms and similarity measures, notably the partitional and hierarchical methods, for the purposes of clustering analysis. Therefore, choosing an appropriate algorithm that best reflects the intended performance of the clustering process while taking similarity measures and dataset topology into consideration is a difficult challenge for the interested user. In addition, no comprehensive empirically focused investigation on the effectiveness and performance of clustering and the optimal combination (i.e., clustering method and similarity measure) has been conducted over a variety of datasets from various domains (IR and non-IR). Finally, the vast majority of older publications avoided

the run-time comparison, which is very alluring for gauging clustering effectiveness. Given the above-outlined challenges and limitations, in this research, the authors were motivated, to present a novel and highly competitive data clustering method, the NHC. Besides presenting the NHC, the authors conducted a thorough evaluation for the NHC vs. nine clustering algorithms utilizing four similarity measures over eight (small, middle, and large-sized) datasets. In addition, the authors employed four evaluation metrics, namely, purity, entropy, silhouette index (SI), and adjusted rand index (ARI), to assess performance. This research sought to provide a fair evaluation of all considered algorithms on datasets that are both textual (IR) and non-textual (non-IR).

METHODOLOGY

In this section, the authors describe the proposed NHC algorithm in detail and the tools needed to run experiments, including algorithms compared, similarity measures, dataset descriptions, and evaluation metrics.

The Proposed NHC Algorithm

The NHC method is built to act hierarchically while taking the surroundings of each data point into account at each stage of the clustering process. Through computational means, the target dataset's similarity measure matrix is used to obtain the neighborhood definition. Assuming a dataset D with a point collection $P = \{P_1, P_2, \dots, P_n\}$, where n is the total number of data points (i.e., size (D)), regardless of the types of the target points (e.g., text or numerical), after encoding data points in the viable system model, the first stage of the NHC is to compute the similarity matrix SM :

$$SSP = \sum_{i=1}^N \sum_{j=1}^{M-i} SM_{ij} \quad (1)$$

Then, using the similarity summation of all pairwise SSP , the next step is to compute the threshold Thv , using which the point's neighboring property is decided. The threshold value can be found using SSP as follows:

$$Thv = \frac{SSP}{X} \quad (2)$$

$$X = \sum_{i=0}^{N-1} N - i, \text{ where } 1 < X < N \quad (3)$$

Then, to construct the neighboring matrix, each data point's value P must exceed Thv to be defined as a neighboring point, and thus the point cell will be marked by value of "1;" otherwise, it is zero value. Equation 4 draws the neighboring matrix NM of points (i.e., P_i and P_j):

$$NM(P_i, P_j) = \begin{cases} 1, & P_{j.value} > Thv \\ 0, & \text{Otherwise} \end{cases} \quad (4)$$

Definition One: The neighborhood of point P_i is $NP(P_i)$, which is defined as follows:

$$NP(P_i) = \{P_j \mid P_i \in C(P_j) \wedge P_j \in C(P_i) \wedge NLM(P_i, P_j) \geq Thv; \text{where } h \geq j \geq 1\} \quad (5)$$

where $C(P_j)$ and $C(P_i)$ signify that both P_i and P_j have already been placed in the same cluster, and their NLM value is greater than Thv which is decided by Equation 4. The parameter h is the maximum number of neighbors of point P_i .

Definition Two: Let P be the group of N points in D ; for any point $P_i \in P$, there exists a specific number $h < n$ such that:

$$Neighbor(P_i) = \bigcup_{j=1}^h P_j \quad (6)$$

where $P_j \neq \phi$, for $j=1, \dots, h$; and $|P_j|=h \neq 0$ ($h < n$ as $|P_j| < |P|$) is the number of neighbors of P_i .

Definition Three: Using the NM matrix, the link matrix will be constructed by multiplying NM and its transpose matrix to create all possible links between each point in D and its neighbors, as given in Equation 7:

$$LM = NM \times NM^T \quad (7)$$

The link degree, in the LM matrix, further expresses the strength of the connection degree between each point P_i and its neighbors. In other words, it reflects the mutual neighbor degree between each point P_i and the points in its neighborhood. The maximum the link degree is between P_i and P_j , the highly likely to have P_j a neighbor of P_i . This step would help to group the most similar points in one cluster, and isolate in other clusters those points whose similarities are either weak or do not reach the data-driven threshold.

Remark One: $LM(P_i)$ reflects the connection strength between P_i and those points in its neighborhood.

The larger the $LM(P_i)$ is, the stronger the connection degree (and then the higher the similarity based on CSM) between P_i and all its neighbors.

Definition Four: The combined similarity matrix (CSM , see Equation 8) of link and similarity matrices (assuming LM is already normalized to have a normalized matrix NLM) is defined as follows:

$$CSM = \alpha \times SM + \beta \times NLM \quad (8)$$

The values of both parameters α and β are determined by the next definition:

$$values(\alpha \text{ and } \beta) = \begin{cases} 0.67, 0.33, SM \geq NLM \\ 0.33, 0.67, Otherwise \end{cases} \quad (9)$$

The authors did not choose α and β parameters' values (0.67 and 0.33) at random. To get the highest potential performance for clustering solutions, they conducted a detailed experimental research

on both numerical and textual datasets with all conceivable values (i.e., 0.5 and 0.5). As the findings below show, although values of (0.5, 0.5) give a competitive performance in certain circumstances, values (0.67 and 0.33) give a higher performance in most cases. According to the values of each component of the equation, both α and β are assigned the values (0.67 and 0.33) interchangeably, depending on which one is the maximum (0.67) or minimum part (0.33). The rationale behind this assignment is to maximize the similarity between each pair (so the max part of the equation is further maximized, “rewarded”) so that each region (clusters in the region) will contain only and only those points of the highest similarity. Each point pair in *CSM* can therefore be shown as having the highest nearest neighbor connection, as they are already closer and more similar.

Remark Two: The *CSM* of P_i is proportional to both or either part of the *CSM* equation. Both parts are either link connection strength or similarity degree, which Equations 8 and 9 seek to maximize.

Clustering Process

After finding the *CSM* matrix, the clustering process will go as follows: Assuming P data points, $P = \{P_1, P_2, \dots, P_n\}$, where n is the number of the whole data points under consideration, the key aim is to construct clusters $C = \{C_1, C_2, C_3, \dots, C_m\}$, where m is the number of all clusters. Initially, the value of m is dynamically left to the mechanism of the clustering algorithm without any restriction on the cluster size or width. Then, in the filtering phase, those clusters will be shrunk to m' , where $m' < m$. The goal is to find as few clusters as possible while maintaining them to be balanced, middle-sized, and highest density. Each P_i and P_j of the maximum value in the *CSM* matrix, in their respective vector of the *CSM*, will be selected from that vector (among N vectors) to establish clusters (as initial seeds) one by one. To consider the pair to establish a cluster, their *CSM* value must exceed the threshold value, which is set, according to the authors' experimental study, at 45%. In other words, the *CSM*'s value must be greater than, or equal to, 0.45, leading to having a smaller number of clusters than those produced by AHC. This step contributes effectively in reducing the hierarchical complexity, as the AHC suffers the initial number of clusters “ N ” which is equal to the size D . While the first pair (e.g., P_i and P_j) will create the first cluster, each successive pair will either join the first cluster or create a new one, just like in the AHC. These pairs are considered the centers of their clusters, using which all members/points will be pulled into those clusters hierarchically. On the other hand, these pairs and the already-allocated points are not allowed to be reconsidered as centers or even recontained in any succeeding cluster. It is a kind of rough clustering, as each point belongs to only one cluster. Such a procedure, besides the threshold mechanism and merging/filtering technique, contributes to reducing the clustering time to less than that consumed by the AHC. The authors' findings show that the NHC has a smaller runtime than most of the hierarchical clustering algorithms used in the comparison study.

Filtering Process

The ultimate aim of the filtering process is to produce m' clusters ($m' < m$) in which the data points have the strongest connection. This process seeks to merge some clusters while maintaining others unmerged. Yet, the decision to merge or leave the cluster unmerged needs to be made statistically to ensure that the NHC's performance is kept uncompromised. To merge cluster C_k over cluster collection C , C_k 's points will be distributed the same way initial clustering was done. Each point will be assigned to the cluster whose centers' connection is the strongest. A data-driven threshold called the split-cluster threshold *SCT* is used to declare the decision. The *SCT* is computed by taking the average size of all clusters produced. Then, for any cluster C_k , where $K < m$, whose size is less than *SCT*, C_k will be scattered over all other clusters whose size is bigger than *SCT* (Equation 10):

$$STC = \frac{1}{m} \sum_{i=1}^m C_k \cdot Size \quad (10)$$

EXPERIMENTAL SETUP

In this section, the authors draw the experimental setup, including the models compared (either hierarchical or partitional ones), the similarity measures, dataset descriptions, and finally evaluation metrics.

Algorithms Compared

The NHC algorithm is assessed against nine clustering algorithms; three of them are traditional ones (i.e., AHC, k-means, and bisect k-means) and others are state-of-the-art (SOTA) rivals, with some of them being hierarchical and others being partitional. The algorithms are listed as follows.

Hierarchical Clustering Group

AHC is the standard algorithm of its kind, which is widely known in the clustering literature (Abdalla et al., 2023; Fernández & Gómez, 2021).

The NBC (Zhang et al., 2015) is an advanced hierarchical clustering algorithm using which the large dataset is divided into distinct groups using the nearest-neighbor boundary clustering algorithm, which then locates each point's nearest neighbor inside the groupings. The authors' experimental analysis shows that the NBC is faster than the AHC and the NHC and close to partitional clustering.

The NGHC (Zhang P. et al., 2019), depending on the gravitation of the local data, presents the hierarchical clustering strategy. According to the results, the NGHC, NHC, and GCC function similarly in a few instances.

The CDC (Kuwil et al., 2019) features a straightforward but adaptable design. The CDC performed marginally better than its competitors, MST-based clustering, k-means, and DbSCAN. The fundamental advantage of the CDC is that no parameters in advance are required. However, the CDC has been ineffective and inefficient when dealing with large datasets. The CDC was created to function better with smaller datasets. Based on the authors' experimental results, the CDC has a competitive performance with NHC over small datasets, yet is inferior (even to AHC and k-means) over big datasets.

Partitional Clustering Group

K-means is the well-known approach from the clustering literature that the authors applied in this study.

Bisect k-means (Zhang W. et al., 2019), the k-means variant, is the bisecting k-means algorithm that starts with a single cluster of all the points, similar to the AHC. In this investigation, the authors: (1) Selected a cluster to split, which was the cluster with the highest SSE for the subsequent division; (2) using the fundamental k-means approach, they identified two subclusters during the bisecting step; (3) they performed step two 10 times; then, they took the division that produced the clustering of the highest similarity; (4) they repeated steps 1–3 till the k clusters were met.

It is important to note that there have been several methods for determining which cluster should be divided in each phase. For instance, the cluster with the greatest size, the cluster with the least overall similarity, or the cluster with the highest SSE, which the authors used in their approach. The authors investigated each situation and discovered that the greatest SSE scenario produced the best bisect k-means results.

The GCC (Kuwil et al., 2017) is a gravity center clustering presented as an advanced version of the CDC to tackle the CDC's limitation over middle-sized and big datasets. Experiments show

that the GCC behaves better than the k-means and bisect k-means in most cases. On the other hand, according to the results, the GCC has comparable efficacy with the AHC and the kCC in some cases.

The kCC (Hussain & Haris, 2019) is an advanced variant of k-means that comes to solve the initialization step of k-means. Results show that the kCC has highly competitive performance compared with the GCC and the NHC.

The Vector Space Model and Term Weighting

To make numerical datasets feasible to handle, the datasets are transformed into data vectors represented in matrices similar to the viable system model. The chosen numerical datasets, as published by UCI, do not have any missing values and do not need any preprocessing because they are made to allow for clear and direct clustering. On the other hand, the textual datasets are transformed into data vectors, which are represented as features in the vector space model (Abdalla & Amer, 2021). The weight of each feature is expressed using the TF-IDF weighting schema.

The Similarity Measures

Euclidean distance (Euc), using the frequency of N terms that would represent the N dimension, treats each feature as a point in 2D space. Based on Equation 1, Euclidean distance computes the relationship between each pair of points (x, y) in this space using their coordinates:

$$D_{Euc}(x, y) = \sum \sqrt{x_1 - y_1)^2 + x_2 - y_2)^2 + \dots + x_n - y_n)^2} \quad (11)$$

Cosine similarity (Equation 12), using the dot product and the magnitude of both vectors of both points (x, y), calculates the pairwise similarity between each pair of points:

$$Sim_{Cos}(x, y) = \frac{\sum_{i=1}^n (x_i * y_i)}{\sqrt{\sum_{i=1}^n x_i^2} * \sqrt{\sum_{i=1}^n y_i^2}} \quad (12)$$

Manhattan (Man) calculates the total absolute differences in the vector coordinates of the two points; it is how defined as follows:

$$Manhattan - distance(x, y) = \sum_{i=1}^n |x_{i1} - y_{i2}| \quad (13)$$

The KL in Equation 14 seeks to identify the difference between the probability distributions:

$$Sim_{KL}(x, y) = \sum_{i=1}^n P(x) * \log \left(\frac{P(x)}{P(y)} \right) \quad (14)$$

Datasets Description

Tables 1 and 2 show the characteristics of the corpus the authors used (IR and non-IR datasets) in this study.

Table 1. Non-IR Datasets Description, UCI Repository

Dataset	Dataset ID	Source	#attributes	#instances
Glass	D ₁	https://archive.ics.uci.edu/ml/datasets/glass+identification	10	214
Iris	D ₂	http://archive.ics.uci.edu/ml/datasets/Iris/	4	150
Wholesale	D ₃	https://archive.ics.uci.edu/ml/datasets/wholesale+customers	8	440
Wine	D ₄	https://archive.ics.uci.edu/ml/datasets/wine	13	178

Table 2. IR Datasets Description

Dataset	Dataset ID	Source	#documents	#classes	#words
BBC	D ₅	http://mlg.ucd.ie/datasets/bbc.html	2225	5	9635
Hitech	D ₆	http://sites.labic.icmc.usp.br/text_collections/	2301	6	12,942
Computers	D ₇	http://sites.labic.icmc.usp.br/text_collections/	9500	19	5011
Web-KB	D ₈	http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/index.html	4199	4	33025

Evaluation Metrics

Data clustering includes two kinds of evaluation metrics, namely, internal and external, or intrinsic and extrinsic metrics. The internal metrics are used to signal how well the clustering algorithm optimized a particular representation, as these metrics depend highly on the feature representations. Accordingly, internal metrics cannot be used for comparison between different representations. In other words, the internal metrics address the separation and cohesion of clusters. However, as advantages of these metrics, they do not ask for the labels. In this study, the authors used the SI as a representative of these metrics. The SI computes how much similarity each data point has to its cluster (cohesion) compared with the different clusters (separation). Simply put, the SI is computed using the mean intracluster distance and the mean nearest-cluster distance for each data point in the relevant cluster. Its value is restricted between 1 and -1, where 1 is the best value and -1 is the worst value. SI is defined by the next formula:

$$SI = \frac{i - n}{\max(i, n)} \quad (15)$$

where i is the mean intracluster distance and n is the mean nearest cluster distance. A silhouette cluster needs at least two clusters to be calculated.

In contrast, the external metrics compare the clustering to an external knowledge source such as labels. Entropy, purity, and ARI are examples of these metrics that the authors used in this research.

Purity (Pu) is defined as the fraction of the entire number of data points that are classified correctly and computed in the following Equation:

$$Pu = \frac{1}{N} \sum_{i=1}^k \max_j |c_i \cap t_j| \quad (16)$$

The best value of purity is 1, and the worst value is 0. The highest purity value is 1, which indicates that each document is already in its cluster, and the worst value is 0.

Entropy (Ent), on the other hand, is used to measure the extent to which a cluster contains a single class and not multiple classes. It is formulated as follows:

$$Ent = \sum_{i=1}^c c_i * \log(c_i) \quad (17)$$

Unlike purity, the best value of entropy is 0, and the worst value is 1. Finally, the ARI accounts for the adjustments in the rand index, which finds the similarity between the true labels and the predicted labels of clusters. It is defined as follows:

$$ARI = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}} \quad (18)$$

where n_{11} signals the number of data pairs which are in the same cluster, n_{00} gives the number of data pairs which are not in the same cluster(s), n_{10} gives the number of data pairs which are in the certain cluster (e.g., A), but in other cluster (e.g., B), and finally, n_{01} gives the number of data pairs which are in the certain cluster (e.g., B), but in other cluster (e.g., A). The ARI value is restricted between 0 and 1, where 1 signals the perfect match between true and predicted clusters.

RESULTS

Using $k \{2, 4, 8, 16, \text{ and } 32\}$ as the number of clusters, the following Tables 3—12 contain the experimental findings of all tests the authors performed over the target datasets. This section includes two subsections, which detail the first and second evaluation phase, respectively. In the first evaluation phase, the authors compared the NHC against the AHC, k-means, and bisect k-means (Tables 3–10). In the second evaluation phase, the authors compared the NHC against relevant SOTA algorithms (Tables 11 and 12). The bold values in Tables 3—12 represent the relative clustering algorithm's highest performance for each measurement on the associated dataset. In each table, if the relative algorithm gives the same value for any metric using both measures, all values are bolded in this case.

First Evaluation Phase

Table 3 shows that partitional clustering with purity is the best (the NHC is better than the AHC, though). Hierarchical with entropy is better than partitional (with the NHC being the best). While the AHC is the worst, the NHC has competitive performance with partitional concerns concerning ARI and SI.

Just like Table 3, Table 4 shows that partitional clustering with purity is the best (the NHC is still better than the AHC, though). However, hierarchical clustering with entropy is still outperforming partitional ones (with the NHC being the best). While the NHC is the best in most cases, the AHC has competitive performance with partitional concerns concerning ARI. With SI, k-means, followed by the NHC, have the best performance.

Unlike glass and IRIS, the results of wholesale in Table 5 show that hierarchical clustering with purity is better than partitional ones. However, partitional with entropy is the best.

Unlike glass, IRIS, and wholesale, Table 6 proves that hierarchical clustering with purity and entropy outperforms partitional clustering. The partitional clustering with the ARI is the best. In general, the NHC with Manhattan and Euclidean has outstanding performance. The NHC achieves

Table 3. Averaged Results of K (2, 4, 8, and 16) Over Glass

Metric/ Algorithm	AHC				K-means				Bi-sect Kmeans				NHC			
	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI
Euc	0.55	3.50	0.278	0.113	0.697	4.92	0.397	0.563	0.687	4.92	0.379	0.563	0.597	3.01	0.379	0.313
Cosine	0.50	4.65	0.244	-0.040	0.752	5.02	0.415	0.729	0.794	4.99	0.544	0.648	0.552	4.31	0.440	-0.091
KL	0.41	2.30	0.060	0.015	0.704	4.73	0.352	0.147	0.718	5.02	0.455	0.090	0.504	2.24	0.351	0.190
Man	0.51	3.55	0.236	0.148	0.711	4.90	0.418	0.532	0.691	4.93	0.383	0.531	0.511	3.22	0.279	0.221

Table 4. Averaged Results of K (2, 4, 8, and 16) Over IRIS

Metric/ Algorithm	AHC				K-means				Bi-sect K-means				NHC			
	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI
Euc	0.68	3.89	0.556	0.342	0.842	4.47	0.518	0.464	0.833	4.62	0.537	0.491	0.693	3.420	0.576	0.354
Cosine	0.67	4.31	0.547	0.619	0.864	4.632	0.630	0.682	0.794	4.99	0.544	0.648	0.691	4.337	0.550	0.648
KL	0.67	4.57	0.531	0.763	0.862	4.441	0.617	0.704	0.857	4.607	0.596	0.702	0.710	4.341	0.560	0.779
Man	0.69	3.92	0.557	0.349	0.842	4.706	0.547	0.498	0.691	4.93	0.383	0.531	0.699	3.901	0.550	0.360

Table 5. Averaged Results of K (2, 4, 8, and 16) Over Wholesale

Metric/ Algorithm	AHC				K-means				Bi-sect Kmeans				NHC			
	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI
Euc	0.682	0.981	0.007	0.730	0.018	5.818	0.001	0.395	0.017	5.863	1.308	0.355	0.699	0.930	0.012	0.751
Cosine	0.677	0.940	-0.01	0.013	0.021	5.587	0.001	0.604	0.020	5.710	2.290	0.512	0.679	0.922	-0.01	0.040
KL	0.677	0.940	-0.01	0.123	0.020	5.588	0.001	0.508	0.021	5.708	0.000	0.492	0.672	0.909	-0.01	0.120
Man	0.682	0.969	0.007	0.719	0.017	5.291	0.000	0.401	0.019	5.85	0.000	0.324	0.687	0.952	0.031	0.730

Table 6. Averaged Results of K (2, 4, 8, and 16) Over Wine

Metric/ Algorithm	AHC				K-means				Bi-sect Kmeans				NHC			
	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI	Pu	Ent	ARI	SI
Euc	0.398	2.66	-0.0	0.368	0.056	4.621	0.001	0.368	0.058	4.848	-4.61	0.378	0.402	2.002	-0.0	0.379
Cosine	0.404	1.26	-0.0	-0.13	0.056	4.607	0.001	0.455	0.056	4.663	0.001	0.381	0.410	1.503	-0.0	-0.17
KL	0.40	1.63	-0.1	0.03	0.06	4.84	0.01	0.46	0.06	4.89	0.005	0.4141	0.4082	1.6881	-0.0	0.0320
Man	0.422	0.94	-0.0	0.43	0.059	4.61	0.01	0.33	0.06	4.84	0.001	0.3194	0.4310	0.9045	-0.0	0.4405

the hybrid performance of both hierarchical and partitional clustering algorithms in most cases. Over small datasets, the NHC shows good performance compared with its traditional rivals.

Excluding entropy in which the NHC has outstanding performance, Tables 7 and 8 allow to conclude that partitional clustering is better than hierarchical clustering. With text datasets, in general, partitional clustering outperforms hierarchical clustering in terms of purity and SI, yet it outperforms hierarchical clustering with entropy. In both cases, the NHC and bisect k-means are better than the AHC and k-means in most cases.

Table 7. Averaged Results of K (4, 8, 16, and 32) Over BCC

Metric/ Algorithm	Purity				Entropy				SI			
	AHC	k-means	Bisect- k-means	NHC	AHC	k-means	Bisect- k-means	NHC	AHC	k-means	Bisect- k-means	NHC
Euclidean	0.2356	0.8561	0.7966	0.2366	2.1552	7.3761	7.4535	2.0052	-0.0039	0.0135	0.0121	-0.0131
Cosine	0.2356	0.8539	0.8747	0.2396	2.1552	7.3987	7.4319	2.0052	-0.0079	0.0261	0.0225	0.0272
KL	0.2346	0.2351	0.2357	0.2376	2.1552	2.2047	7.7055	2.1542	0.0	0.0	0.0	0.0
Manhattan	0.2360	0.4348	0.6081	0.2368	2.1709	7.6329	7.5702	2.1712	0.2577	0.0481	0.0083	0.2577

Table 8. Averaged Results of K (4, 8, 16, and 32) Over Computers

Metric/ Algorithm	Purity				Entropy				SI			
	AHC	k-means	Bisect- k-means	NHC	AHC	k-means	Bisect- k-means	NHC	AHC	k-means	Bisect- k-means	NHC
Euclidean	0.0541	0.1453	0.1733	0.0582	2.1584	6.8965	8.9251	2.0280	0.4634	-0.0225	-0.0207	0.4901
Cosine	0.0541	0.3126	0.3074	0.0590	2.1552	8.8794	8.8845	2.0276	0.0	0.0	0.0	0.0
KL	0.0541	0.0542	0.0542	0.0542	2.1552	2.3268	9.1585	2.1370	0.0	0.0	0.0	0.0
Manhattan	0.0542	0.1151	0.1166	0.0566	2.1689	9.1259	9.0461	2.1690	0.5677	0.0399	0.0497	0.5677

Unlike Table 8, results in Table 9 show that the NHC is highly competitive with partitional clustering in terms of purity and SI. However, the NHC is still superior regarding entropy.

Results in Table 10 show that the NHC is highly competitive with partitional clustering in terms of purity, yet inferior regarding SI. However, just like in Tables 7—9, the NHC is still superior regarding entropy. It is worth noting that, over IR datasets, the NHC is inferior to partitional algorithms with any metric over any dataset, and it is possible to note that the NHC is still better than its hierarchical rivals in most cases.

Second Evaluation Phase

To evade overloading this work with detailed results, the authors used Euclidean for the second phase of evaluation as all SOTA algorithms used Euclidean in their works. For the number of clusters, the authors used the same k values over each corresponding dataset as given in the first phase of evaluation. It is worth indicating that the authors used the parameter settings in all compared algorithms as drawn in their published papers.

Table 11 evidences that the NHC and the NGHC have similar performance trends, chiefly over small datasets. In general, the NGHC is better than the GCC and the NBC, yet slightly inferior to the

Table 9. Averaged Results of K (4, 8, 16, and 32) Over Hitech

Metric/ Algorithm	Purity				Entropy				SI			
	AHC	k-means	Bisect- k-means	NHC	AHC	k-means	Bisect- k-means	NHC	AHC	k-means	Bisect- k-means	NHC
Euclidean	0.2664	0.2854	0.3307	0.2783	2.1552	7.7083	7.6827	2.1002	0.5838	-0.0589	-0.0083	0.5903
Cosine	0.2676	0.5571	0.5693	0.2770	2.1920	7.5228	7.4589	2.1919	-0.0071	0.0203	0.0176	-0.0040
KL	0.2681	0.2664	0.2664	0.2801	2.1552	2.1552	7.7393	2.0098	0.0	0.0	0.0	0.0
Manhattan	0.2660	0.2610	0.2600	0.2660	2.1552	7.7379	7.7373	2.1544	0.6555	0.1878	0.1944	0.6555

Table 10. Averaged Results of K (4, 8, 16, and 32) Over Web-KB

Metric/ Algorithm	Purity				Entropy				SI			
	AHC	k-means	Bisect-k-means	NHC	AHC	k-means	Bisect-k-means	NHC	AHC	k-means	Bisect-k-means	NHC
Euclidean	0.391	0.652	0.677	0.401	2.158	7.999	8.104	2.083	0.001	0.009	0.008	0.001
Cosine	0.391	0.679	0.683	0.401	2.158	7.997	8.036	2.159	0.001	0.019	0.012	0.011
KL	0.393	0.393	0.392	0.401	2.155	3.084	8.339	2.026	0.002	0.002	0.001	0.000
Manhattan	0.393	0.397	0.399	0.408	2.256	5.140	8.338	2.054	0.428	0.134	0.124	0.430

Table 11. NHC Against SOTA Clustering Algorithms Using Euclidean Over Small Datasets

	Purity						Entropy						SI					
	NBC	NGHC	CDC	GCC	KCC	NHC	NBC	NGHC	CDC	GCC	KCC	NHC	NBC	NGHC	CDC	GCC	KCC	NHC
D1	0.55	0.59	0.55	0.69	0.69	0.60	3.39	3.01	3.39	4.878	4.90	3.001	0.121	0.30	0.17	0.56	0.56	0.31
D2	0.68	0.69	0.69	0.82	0.82	0.69	3.69	3.42	3.41	4.49	4.55	3.42	0.320	0.35	0.31	0.44	0.52	0.35
D3	0.68	0.68	0.68	0.01	0.01	0.70	0.98	0.93	0.98	5.93	5.99	0.93	0.731	0.74	0.73	0.39	0.31	0.75
D4	0.39	0.40	0.39	0.05	0.06	0.49	2.66	2.44	2.65	4.50	4.590	2.004	0.35	0.36	0.35	0.37	0.38	0.38
#Win	0	0	0	1	1	2	0	1	0	0	0	3	0	0	0	0	2	2

Table 12. NHC Against SOTA Clustering Algorithms Using Euclidean Over IR Datasets

	Purity						Entropy						SI					
	NBC	NGHC	CDC	GCC	KCC	NHC	NBC	NGHC	CDC	GCC	KCC	NHC	NBC	NGHC	CDC	GCC	KCC	NHC
D ₅	0.22	0.23	0.21	0.85	0.78	0.25	2.13	2.00	4.39	7.20	7.53	2.01	-0.00	-0.01	-0.03	0.014	0.010	-0.01
D ₇	0.05	0.05	0.04	0.14	0.18	0.06	2.21	2.31	5.64	6.12	9.03	2.13	0.458	0.461	0.411	-0.02	-0.01	0.490
D ₆	0.26	0.28	0.27	0.28	0.33	0.28	2.15	2.10	3.01	7.10	6.09	2.10	0.589	0.590	0.584	-0.05	-0.01	0.590
D ₈	0.39	0.39	0.37	0.64	0.65	0.40	2.14	2.17	4.65	7.10	7.01	2.08	0.000	0.001	0.001	0.009	0.008	0.001
#Win	0	2	0	1	3	0	0	2	0	0	0	3	0	0	0	2	0	2

NHC in most cases. On the other hand, the kCC has competitive performance compared to the NHC in terms of SI and purity, yet is greatly inferior to the NHC in entropy.

Table 12 shows that the NGHC is competitive with the GCC and the kCC regarding purity. The NGHC and the NHC are superior to all rivals regarding entropy. Finally, both the GCC and the NHC have the best performance with SI. The authors could also infer that there is no dominant clustering algorithm over all datasets with all evaluation metrics. Nevertheless, based on the discussion of results in Tables 11 and 12, it is possible to note that the NGHC, NHC, and kCC are superior algorithms in the great majority.

Overall, the NBC has almost equivalent performance to the AHC. These findings are consistent with Zhang et al.'s (2015) findings. The authors' experimental analysis shows that the NBC is by far faster than the AHC and significantly faster than the NHC (Figures 3 and 4). Further, according to the results, the NGHC, NHC, and GCC function similarly in a few instances. On the other hand, except for Hitech, the CDC is inferior to all hierarchical clustering algorithms. The degraded performance

of the the CDC, chiefly over big datasets, is due to the fact that the CDC was devised to function better with smaller datasets. Based on the authors' experimental results, the CDC has had competitive performance with the NHC over small datasets but is inferior (even to the AHC and k-means) over big datasets. The authors also noted that, unlike partitional algorithms, hierarchical clustering has stable results over all datasets.

Moreover, experiments show that the GCC behaves better than k-means and bisect k-means, in most cases. Finally, given its competitive results, the kCC (Hussain & Haris, 2019) has experimentally proven its power to solve the initialization step of k-means. Results show that the kCC has comparable performance compared with both the GCC and the NHC. Overall, the best clustering variations are the NHC as a hierarchical-based algorithm and the kCC as a partitional type.

DISCUSSION

Based on their experimental study, the authors found that hierarchical clustering has a competitive performance; however, in most cases, it has shown poor performance (excluding the NHC) on both non-IR and IR datasets compared with the GCC and the kCC. That is because of the hierarchical nature, which makes it unable to fix any mistakes that would happen while clustering the documents, as Fernández & Gómez. (2021) and Zhu et al. (2018) discussed. On the other hand, partitional clustering (e.g., k-means and bisect-k-means) also has some deficits, which the GCC and the kCC come to cover. Among these deficits are the centroid initialization, and in practice, k-means sometimes fails to produce the desired number of clusters or secure the clusters that meet the document classes. On the other extreme, according to the authors' experiments, the bisecting k-means and kCC have been able to produce clusters of almost uniform sizes when k-means and the GCC produced clusters of different sizes.

These key characteristics make bisecting k-means and the kCC the best options for big data clustering, when partitional clustering is under consideration. As a compromised solution, the authors empirically found that the NHC algorithm is highly attractive, combining the robust architecture of hierarchical clustering with the good performance of partitional ones. Experimental results obtained in this work confirm the authors' claims, as the NHC has produced a competitive performance that outperforms both hierarchical partitional algorithms in most cases. It is worth indicating that, during experimental evaluation, the authors found that both K-means and bisecting k-means were faster than both the NHC and the AHC, with bisecting k-means being the fastest clustering algorithm. This is another characteristic that makes the bisecting k-means a better option for clustering big data. Thus, one of the authors' ultimate aims for the next work is to make the NHC as efficient as possible.

Run Time Comparison

In this subsection, the authors briefly draw a comparison between all algorithms in terms of their run time on the non-IR datasets. If the algorithm has been slow or fast on this type of dataset, this would surely mean that this algorithm is slow or fast on IR datasets as well. As a result, the authors restricted the comparison to non-IR datasets to robustly conclude the fast and slowest algorithm. They selected the IRIS and glass datasets for this comparison.

First, they conducted a comparative analysis of the NHC against its traditional rivals (i.e., AHC, k-means, and bisect k-means). With regard to the first phase of evaluation, Figures 1 and 2 illustrate the averaged run time in seconds of four algorithms (i.e., AHC, NHC, k-means, and bisect k-means) for three k values (i.e., 2, 4, and 8). The results show that the NHC could be used to address the quadratic run time of hierarchical clustering. The NHC seems to be significantly close to partitional ones, chiefly k-means. However, bisect k-means is the fastest algorithm when the AHC is the lowest one. As to the impact of similarity measure, it is possible to conclude that cosine contributes significantly to the degraded speed of all algorithms, which is in contrast to the Euclidean and Manhattan contributions.

Figure 1. Run Time in Second, Glass Dataset

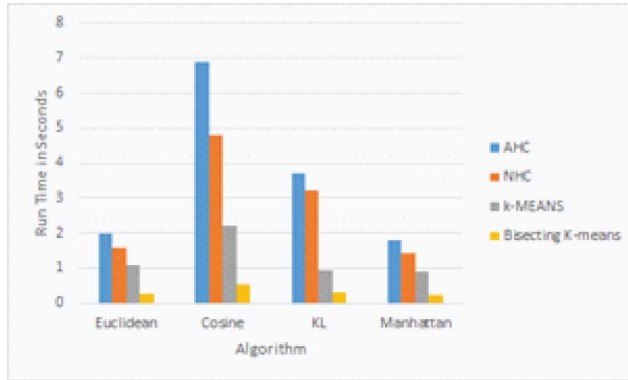
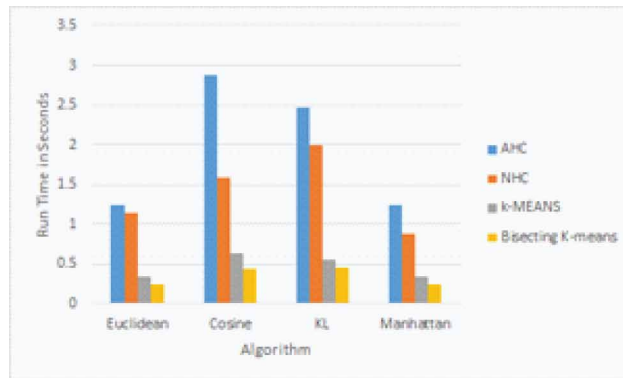


Figure 2. Run Time in Second, IRIS Dataset



While Manhattan has the most positive role in the efficiency of algorithms, KL, on the other hand, has been able to draw itself as a compromised solution between cosine and Euclidean.

On the other hand, with regard to the second phase of evaluation, Figures 3 and 4 show the averaged run time in seconds of nine algorithms (i.e., AHC, NBC, NHGC, CDC, NHC, k-means, bisect k-means, GCC, and kCC) for three k values (i.e., 2, 4, and 8). The findings in Figures 1–4 showcase that the AHC, followed by the CDC, NGHC, and NHC, are the slowest algorithms when the bisecting k-means, followed by k-means and the kCC, are the fastest. Surprisingly, the NBC is seen as the compromised solution between hierarchical and partitional algorithms, as it is the second-fastest one after the bisect k-means. The bisect k-means has been seen as the fastest because of its attractive mechanism, as it always works in one single cluster in each successive iteration. This comes in contrast to the k-means and kCC, which always work on the whole dataset in each iteration. Finally, it is noted that Euclidean-based algorithms are faster than cosine-based algorithms, since Euclidean is faster than cosine.

In brief, hierarchical clustering has shown competitive performance on non-IR datasets. However, in most cases, excluding the NHC, hierarchical clustering has shown poor performance on the IR dataset, compared with both the kCC and bisecting K-means. This is because of the nature of the AHC, which makes it unable to fix any mistakes that would happen while clustering the documents, as Fernández & Gómez. (2021) and Zhu et al. (2018) discussed. Such a limitation has been addressed by considering both local and global information about each point in the NHC using the NLM matrix.

Figure 3. Run Time in Second, Glass Dataset

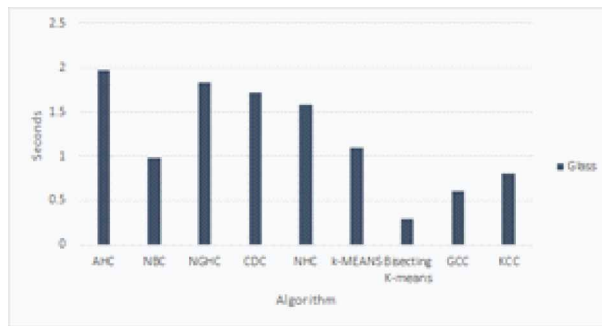
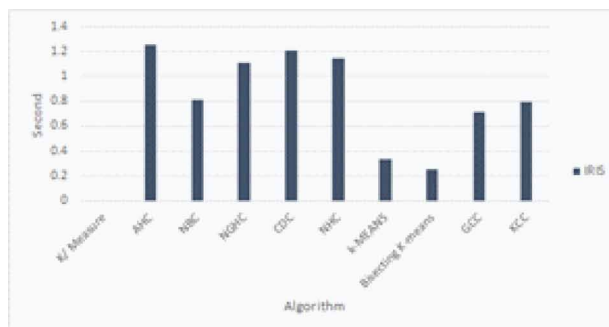


Figure 4. Run Time in Second, IRIS Dataset



This is one reason behind the competitive performance of the NHC. On the other hand, the k-means and bisect k-means also have some deficits, which the GCC and kCC come to address. Among these deficits are the centroid initialization, and, in practice, k-means sometimes fails to either produce the desired number of clusters or the clusters that meet the document classes. On the other extreme, according to the authors' experiments, the bisecting k-means were able to produce clusters of almost uniform sizes when k-means produced clusters of different sizes. These key characteristics make the bisecting k-means even better than the kCC and GCC for document clustering when partitional clustering is under consideration. Finally, it is worth indicating that, during experimental evaluation, the authors found that partitional clustering was faster than hierarchical ones, with the bisect k-means being the fastest clustering algorithm. This is another characteristic that makes the bisecting k-means a better option for clustering big data in particular.

On the other hand, unlike hierarchical clustering algorithms, which only address the local neighbors at each clustering step, the NHC, via the NLM matrix, considers both the local and global neighbors, making it effectively successful in separating the overlapping clusters with both small and big datasets. Finally, the authors conducted a rigorous run-time comparison on two small non-IR datasets between all algorithms; the statistics of the comparisons showed that the bisecting clustering was the fastest algorithm, while the AHC algorithm was the slowest. Moreover, the Euclidean-based algorithms were faster than those that used the cosine similarity measure.

Comment on the Behavior of Clustering Algorithms

In this study, hierarchical clustering demonstrated competitive performance on non-IR datasets. However, in most situations, except for the NHC, hierarchical clustering performed poorly

on the IR dataset, when compared to both the kCC and bisecting k-means. This is due to the nature of the AHC, which makes it unable to correct any errors that may occur during clustering documents (Fernández & Gómez, 2021; Zhu et al., 2018). The authors used the NLM matrix to account for both local and global information regarding each point in the NHC, addressing this constraint. This is one of the reasons for the NHC's strong competitive success. On the other hand, the k-means and bisect k-means have several shortcomings that the GCC and kCC attempt to overcome. Among these shortcomings is centroid initialization, and, in reality, the k-means occasionally fails to yield either the appropriate number of clusters or clusters that meet the document classes. On the other hand, the authors' investigations showed that the bisecting k-means might yield clusters of nearly uniform sizes, when the k-means produced clusters of varying sizes (Steinbach et al., 2000). These fundamental qualities make the bisecting k-means superior to the kCC and GCC for document clustering, when partitional clustering is taken into account. Finally, during the experimental evaluation, the authors discovered that partitional clustering was faster than hierarchical clustering, with the bisect k-means being the fastest clustering algorithm. This is another feature that makes the bisecting k-means a superior choice for clustering large datasets.

Most notably, hierarchical clustering is commonly regarded as clustering of higher quality, although it is limited due to its "quadratic" time complexity. In contrast, the k-means and its derivatives have a linear time complexity, but are depicted as having inferior clusters. Driven by these arguments, this work delivers a detailed comparative empirically-oriented analysis on both document and nondocument data, setting it apart from previous efforts, particularly when nine clustering algorithms and four similarity metrics are used. The study intends to confirm or deny these assertions about which similarity measure will give the highest performance on these algorithms based on the results and discussion below. It is worth indicating that the ultimate aim of this study was to enrich the literature with a new, simple, yet robust, and dynamic clustering algorithm to further enhance and simplify the entire process of data clustering.

Merits and Limitations of the Neighboring-Aware Hierarchical-Based Clustering Approach

Merits

Unlike hierarchical clustering algorithms, which only address the local neighbors at each clustering step, the NHC via the NLM matrix considers both the local and global neighbors, making it effectively successful in separating overlapping clusters with both small and big datasets. It is worth considering that, overall, the NHC achieves the maximum silhouette coefficient and lowest entropy (with low standard deviations), which means that the NHC has the power to find the optimal number of clusters. Moreover, the maximum silhouette coefficient value denotes the well-separated and compact clusters with the highest "possible" density.

Limitations

The NHC is similar to hierarchical clustering, which is usually thought of as the clustering of the better quality; it is restricted due to its "quadratic" time complexity. However, the NHC's run time is still better than that of the AHC, NGHC, and CDC algorithms. Therefore, one of the authors' future goals is to make the NHC as efficient as possible. Moreover, given that the scope of this study was limited to numerical and textual datasets, the authors did not run the NHC on different-topology datasets (i.e., image and gene datasets). Such implementation is out of the scope of this research, as all considered algorithms run only on either numerical or textual datasets, or even both. Thus, the authors plan to record its performance on as many different topology datasets as to make a universally applicable competitive clustering algorithm.

Brief Insights and Recommendations

Based on the results and discussion on the target datasets above, the authors generally recommend the best similarity measure, according to the general averaged results, to be used so the corresponding evaluation metric would get its maximum performance.

Recommendations on non-IR datasets are: Cosine and Manhattan for purity, and Manhattan and KL for entropy; Euclidean and KL for SI; for the ARI, Euclidean, and cosine with k-means variations.

Recommendations on IR datasets are: Cosine and Euclidean for purity; KL and Manhattan for entropy; for SI, Euclidean and Manhattan have the best performance, chiefly with the NHC.

FUTURE RESEARCH DIRECTIONS

In their follow-up work, the authors plan to expand this study by leveraging parallel and distributed computing (Zhang W. et al., 2019) to make the NHC maximally efficient, while maintaining its efficacy, making the NHC effectively applicable for big datasets. Further, given that the performance of their proposed algorithm showed to be promising, the authors established it can be used in the future for several applications, including distributed database applications (Amer et al., 2020). Finally, despite using big datasets such as computers and Web-KB, the authors plan to further evaluate the NHC on bigger datasets.

CONCLUSION

In this study, the authors aimed to enrich the clustering literature with a new effective hierarchical clustering variation. The introduced variation is a straightforward, yet reliable, NHC. To determine the NHC's effectiveness, the authors investigated the performance of the NHC along with nine clustering algorithms belonging to the hierarchical and partitional clustering types using four similarity measures and distance metrics (i.e., Euclidean distance, cosine similarity measure, KL, and Manhattan) over eight datasets. The authors diversified the datasets by taking them from different resources, including IR and ML repositories (e.g., UCI). Over these datasets of various sizes from various applications, the researchers carried out a thorough two-phase evaluation of the NHC, using various similarity metrics and distances. They first compared the NHC against three conventional clustering algorithms and empirically observed it was highly effective. Then, the authors tested the NHC against six relevant rivals, demonstrating its extremely competitive performance.

The experimental study showed that hierarchical clustering, including the NHC, has competitive performance on non-IR datasets. The authors developed the NHC to be universally applicable to any dataset of any kind. However, in most cases, excluding the NHC, hierarchical clustering has shown poor performance on the IR dataset, compared with both the kCC and bisecting k-means. Nevertheless, due to its effective design, the NHC, which is a hierarchical-driven algorithm, showed to be powerful and highly competitive, compared to all clustering algorithms, including the traditional and SOTA ones. On the other extreme, according to the authors' experiments, besides being the fastest algorithm, the bisecting k-means was able to produce clusters of almost uniform sizes, when the k-means produced clusters of different sizes (Steinbach et al., 2000). These key characteristics make the bisecting k-means even better than the kCC and GCC for document clustering, when partitional clustering is under consideration. The results also evidenced that the dataset's characteristics and similarity measures have a great impact on the performance of any clustering algorithm. For example, on small IR datasets (Tables 3—6), the k-means and bisect-k-means behaved better than both the AHC and NHC, excluding entropy. On the other hand, on the IR dataset, the NHC behaved the best in most cases, chiefly with entropy. The NHC generally behaved the best.

ETHICS APPROVAL

The study was classified as nonhuman subject research, and informed permission was not required.

COMPETING INTERESTS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

FUNDING

This research has been supported by Research Project Number (RSP2024R206), King Saud University, Riyadh, Saudi Arabia.

ACKNOWLEDGMENT

The authors would like to thank and appreciate the support they received from the Researchers Supporting Project number (RSP2024R206), King Saud University, Riyadh, Saudi Arabia.

CONSENT TO PARTICIPATE

Not applicable.

CONSENT TO PUBLICATION

Not applicable.

DATA AVAILABILITY

The datasets the authors used in this work are publicly available.

PROCESS DATES

Received: February 9, 2024, Revision: May 2, 2024, Accepted: April 16, 2024

CORRESPONDING AUTHOR

Correspondence should be addressed to Ali Amer (United Arab Emirates, aliaaa2004@yahoo.com)

REFERENCES

- Abdalla, H. I., & Amer, A. A. (2021). Boolean logic algebra driven similarity measure for text-based applications. *PeerJ. Computer Science*, 7, e641. doi:10.7717/peerj-cs.641 PMID:34401474
- Abdalla, H. I., Amer, A. A., & Ravana, S. D. (2023). On hierarchical clustering-based approach for RDBS design. *Journal of Big Data*, 10(1), 172. doi:10.1186/s40537-023-00849-7
- Ahlgren, P., & Colliander, C. (2009). Document-document similarity approaches and science mapping: Experimental comparison of five approaches. *Journal of Informetrics*, 3(1), 49–63. doi:10.1016/j.joi.2008.11.003
- Ahlgren, P., & Jarneving, B. (2008). Bibliographic coupling, common abstract stems, and clustering: A comparison of two document-document similarity approaches in the context of science mapping. *Scientometrics*, 76(2), 273–290. doi:10.1007/s11192-007-1935-1
- Akilandeswari, J., Jothi, G., Dhanasekaran, K., Kousalya, K., & Sathiyamoorthi, V. (2022). Hybrid firefly-ontology-based clustering algorithm for analyzing tweets to extract causal factors. *International Journal on Semantic Web and Information Systems*, 18(1), 1–27. doi:10.4018/IJSWIS.295550
- Amalia, A., Sitompul, O. S., Nababan, E. B., & Mantoro, T. (2020). A comparison study of document clustering using DOC2VEC versus TFIDF combined with LSA for small corpora. *Journal of Theoretical and Applied Information Technology*, 98(17), 3644–3657.
- Amer, A. A., & Mohamed, M. H., & Al_Asri, K. (. (2020). ASGOP: An aggregated similarity-based greedy-oriented approach for relational DDBSs design. *Heliyon*, 6(1). Advance online publication. doi:10.1016/j.heliyon.2020.e03172 PMID:31938750
- Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* (Vol. 07-09-January-2007, pp. 1027–1035). Association for Computing Machinery.
- Chander, S., Vijaya, P., & Dhyani, P. (2022). A parallel fractional lion algorithm for data clustering based on MapReduce cluster framework. *International Journal on Semantic Web and Information Systems*, 18(1), 1–25. doi:10.4018/IJSWIS.297034
- Curiskis, S. A., Drake, B., Osborn, T. R., & Kennedy, P. J. (2020). An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Information Processing & Management*, 57(2), 102034. Advance online publication. doi:10.1016/j.ipm.2019.04.002
- Fernández, A., & Gómez, S. (2021). mdendro: Extended Agglomerative Hierarchical Clustering.
- Forsyth, R. S., & Sharoff, S. (2014). Document dissimilarity within and across languages: A benchmarking study. *Literary and Linguistic Computing*, 29(1), 6–22. doi:10.1093/lilc/fqt002
- Gilpin, S., & Davidson, I. (2017). A flexible ILP formulation for hierarchical clustering. *Artificial Intelligence*, 244, 95–109. doi:10.1016/j.artint.2015.05.009
- Gong, S., Hu, W., Li, H., & Qu, Y. (2018). Property clustering in linked data: An empirical study and its application to entity browsing. *International Journal on Semantic Web and Information Systems*, 14(1), 31–70. doi:10.4018/IJSWIS.2018010102
- Goyal, M. M., Agrawal, N., Sarma, M. K., & Kalita, N. (2015). Comparison clustering using cosine and fuzzy set based similarity measures of text documents. *ArXiv*, abs/1505.00168.
- Halim, Z., & Uzma, . (2018). Optimizing the minimum spanning tree-based extracted clusters using evolution strategy. *Cluster Computing*, 21(1), 1–15. doi:10.1007/s10586-017-0868-6
- Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008)* (pp. 49–56).
- Hussain, S. F., & Haris, M. (2019). A k-means based co-clustering (kCC) algorithm for sparse, high dimensional data. *Expert Systems with Applications*, 118, 20–34. doi:10.1016/j.eswa.2018.09.006
- Kuwil, F. H., Atila, Ü., Abu-Issa, R., & Murtagh, F. (2020). A novel data clustering algorithm based on gravity center methodology. *Expert Systems with Applications*, 156, 113435. Advance online publication. doi:10.1016/j.eswa.2020.113435

- Kuwil, F. H., Shaar, F., Topcu, A. E., & Murtagh, F. (2019). A new data clustering algorithm based on critical distance methodology. *Expert Systems with Applications*, 129, 296–310. doi:10.1016/j.eswa.2019.03.051
- Liu, R. G., & Frank, M. J. (2022). Hierarchical clustering optimizes the tradeoff between compositionality and expressivity of task structures for flexible reinforcement learning. *Artificial Intelligence*, 312, 312. doi:10.1016/j.artint.2022.103770 PMID:36711165
- Ljubešić, N., Boras, D., Bakarić, N., & Njavro, J. (2008). Comparing measures of semantic similarity. In *Proceedings of the 30th International Conference on Information Technology Interfaces, 2008 (ITI 2008)* (pp. 675–682). IEEE doi:10.1109/ITI.2008.4588492
- Nasim, Z., & Haider, S. (2020). Cluster analysis of urdu tweets. *Journal of King Saud University. Computer and Information Sciences*. Advance online publication. doi:10.1016/j.jksuci.2020.08.008
- Nguyen, T.-H. T., Dinh, D.-T., Sriboonchitta, S., & Huynh, V.-N. (2019). A method for kmeans-like clustering of categorical data. *Journal of Ambient Intelligence and Humanized Computing*, ●●●, 1–11.
- Patil, H., & Thakur, R. S. (2018). Document clustering. In *Information retrieval and management* (pp.). <https://doi.org/doi:10.4018/978-1-5225-5191-1.ch003>
- Şaar, F., & Topcu, A. E. (2022). Minimum spanning tree-based cluster analysis: A new algorithm for determining inconsistent edges. *Concurrency and Computation*, 34(9), e6717. Advance online publication. doi:10.1002/cpe.6717
- Salem, S. B., Naouali, S., & Chtourou, Z. (2018). A fast and effective partitional clustering algorithm for large categorical datasets using a k-means based approach. *Computers & Electrical Engineering*, 68, 463–483. doi:10.1016/j.compeleceng.2018.04.023
- Selvam, S., Balakrishnan, R., & Ramakrishnan, B. S. (2018). Ontology with hybrid clustering approach for improving the retrieval relevancy in social event detection. *International Journal on Semantic Web and Information Systems*, 14(4), 33–56. doi:10.4018/IJSWIS.2018100102
- Shamir, O., & Tishby, N. (2010). Stability and model selection in k-means clustering. *Machine Learning*, 80(2-3), 213–243. doi:10.1007/s10994-010-5177-8
- Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. *KDD Workshop on Text Mining*, 400, 1–2. doi:10.1109/ICCCYB.2008.4721382
- Strehl, A., Ghosh, J., & Mooney, R. (2000). Impact of similarity measures on web-page clustering. In *Proceedings of the Workshop on Artificial Intelligence for Web Search (AAAI 2000)* (pp. 58–64).
- Wang, X., Wang, X. L., Chen, C., & Wilkes, D. M. (2013). Enhancing minimum spanning tree-based clustering by removing density-based outliers. *Digital Signal Processing: A Review Journal*, 23(5), 1523–1538. 10.1016/j.dsp.2013.03.009
- Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2), 165–193. doi:10.1007/s40745-015-0040-1
- Yang, C. (2022). A path-clustering driving travel-route excavation. *International Journal on Semantic Web and Information Systems*, 18(1), 1–16. doi:10.4018/IJSWIS.306750
- Zhang, P., Xiang, X., & She, K. (2019). A novel hierarchical clustering approach based on data gravitation model. *Journal of Physics: Conference Series*, 1325(1), 012106. Advance online publication. doi:10.1088/1742-6596/1325/1/012106
- Zhang, W., Zhang, G., Chen, X., Liu, Y., Zhou, X., & Zhou, J. (2019). DHC: A distributed hierarchical clustering algorithm for large datasets. *Journal of Circuits, Systems, and Computers*, 28(4), 1950065. doi:10.1142/S0218126619500658
- Zhang, W., Zhang, G., Wang, Y., Zhu, Z., & Li, T. (2015). NBC: An efficient hierarchical clustering algorithm for large datasets. *International Journal of Semantic Computing*, 9(3), 307–331. doi:10.1142/S1793351X15400085
- Zhu, E., & Ma, R. (2018). An effective partitional clustering algorithm based on new clustering validity index. *Applied Soft Computing*, 71, 608–621. doi:10.1016/j.asoc.2018.07.026

Muna Al-Razgan received the Ph.D. degree in information technology from George Mason University, VA, USA. She is currently a Full Professor in software engineering with the College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. Her research interests include data mining, machine learning, artificial intelligence, educational data mining, and assistive technologies.

Hassan Abdalla has been an associate professor of Information Systems at the College of Technological Innovation since 2018. He holds a PhD in Information Systems from London, UK. Prior to joining Zayed University, Dr. Abdalla, who is an Oracle Certified Professional, worked as an associate professor at the College of Computer and Information Sciences at King Saud University (KSU), Riyadh, Saudi Arabia. He also served there as a head of the Quality Unit. Dr. Abdalla has published his research work in many reputable refereed international journals and conference proceedings, and has also served as a reviewer for many top-ranked Journals. Dr. Abdalla refereed international journal distributed database systems, information retrieval, knowledge management, and enterprise computing.

Mahfoudh Al-Asaly received the Ph.D. degree from the Information System Department, King Saud University (KSU), Riyadh, Saudi Arabia. M.Sc. degree from the Computer Science Department, King Saud University (KSU), Riyadh, Saudi Arabia. He also works as a Researcher with the Computer Science College, KSU. His research interests include machine learning, cloud computing, and artificial intelligence.

Taha Alfakih is an assistant professor at King Saud University. He received his B.S. degree in computer science from the Computer Science Department, Hadhramout University, Yemen, M.Sc. degree from the Computer Science Department, King Saud University (KSU), Riyadh, Saudi Arabia, and Ph.D. degree from the Information System Department, King Saud University (KSU), Riyadh, Saudi Arabia. He also works as a researcher with the Computer Science College, KSU. His research interests include machine learning, mobile edge computing, and the Internet of things.

Muneer Al-Hammadi (Member, IEEE) received the Ph.D. degree in computer engineering from King Saud University (KSU), Saudi Arabia, in 2020. He is currently working as a postdoctoral fellow with the Department of Civil and Environmental Engineering (IBM), Faculty of Engineering, Norwegian University of Science and Technology (NTNU). His research interests include image and video processing, computer vision, and machine learning.