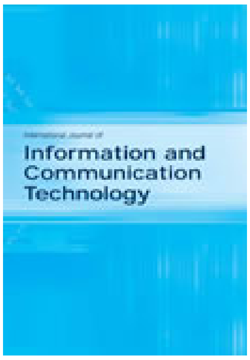


International Journal of
Information and Communication Technology

International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

A distributed two-stage clustering method based on node sampling

Baolong Zhang, Haiyan Huang

DOI: 10.1504/IJICT.2025.10073373

Article History:

Received:	10 June 2025
Last revised:	26 June 2025
Accepted:	27 June 2025
Published online:	26 September 2025

A distributed two-stage clustering method based on node sampling

Baolong Zhang*

Office of Development Planning and Quality Evaluation,
Jiyuan Vocational and Technical College,
Jiyuan, 459000, China
Email: jyzy_zbl@163.com
*Corresponding author

Haiyan Huang

School of Artificial Intelligence,
Jiyuan Vocational and Technical College,
Jiyuan, 459000, China
Email: 0001238@jyvtc.edu.cn

Abstract: To address the issues of high computational resource consumption and low clustering efficiency in big data clustering, this paper first proposes the density deviation sampling improvement algorithm (EDDS). Then, each cluster node independently performs clustering on a subset of the big data to generate initial local clustering results. Next, using the EDDS algorithm on each node, representative data subsets are extracted, and these subsets are aggregated into a sample set that reflects the characteristics of the entire big dataset. Finally, further clustering analysis is performed on this sample set. By integrating the local clustering information from each node using the clustering results, a comprehensive clustering result for the entire big dataset is output. Experimental results demonstrate that, compared to traditional clustering methods, the suggested approach effectively combines the efficiency of parallel processing with the accuracy of integrated analysis.

Keywords: big data clustering; distributed computing; density deviation sampling; node sampling; two-stage clustering.

Reference to this paper should be made as follows: Zhang, B. and Huang, H. (2025) ‘A distributed two-stage clustering method based on node sampling’, *Int. J. Information and Communication Technology*, Vol. 26, No. 34, pp.100–115.

Biographical notes: Baolong Zhang received his Master’s degree from Nanjing University of Technology in June 2013. He is currently working in the Jiyuan Vocational and Technical College. His research interests include computer application technology and the big data technology.

Haiyan Huang received her Master’s degree from Nanjing University of Technology in June 2013. She is currently working in the Jiyuan Vocational and Technical College. Her research interests include computer application technology and the big data technology.

1 Introduction

In the quick growth of information technique, the amount of data is growing explosively. As a core data mining methodology, clustering enables automated grouping of data elements according to their natural affinities, finding extensive utility in diverse domains (Zou, 2020). Nevertheless, as the volume of data surges rapidly, traditional clustering algorithms encounter a substantial rise in computational complexity when handling large-scale datasets, leading to diminished operational efficiency (Suganya et al., 2018). To tackle the challenges associated with clustering large-scale data, distributed computing technology has emerged as a viable solution. By dispersing large-scale data to multiple computing nodes for parallel processing, distributed computing can fully harness the computing capabilities of the cluster, resulting in a notable boost in computational efficiency (Shukur et al., 2020). At the same time, sampling techniques have come to the fore in big data processing. Sampling techniques can reduce the scale and complexity of data processing to a certain extent by selecting representative samples from large-scale datasets to be analysed, reducing the amount of computation while retaining the chief features of the data (Rajendra Prasad et al., 2021). Integrating distributed computing technology with sampling technology offers fresh perspectives and methodologies for addressing the challenge of large-scale data clustering.

Current research on clustering algorithms has seen substantial progress, yet these algorithms still confront hurdles in big data scenarios, including the efficient handling and processing of large-scale data collections, high-dimensional data, and the dynamic alterations in data flows (Shafi et al., 2024). By providing a solution to clustering problems in extensive datasets, distributed clustering algorithms are highly adaptable for environments where data is spread out over numerous computing nodes. The universal clustering algorithm can run effectively on datasets of different scales. For small-scale datasets, the algorithm can complete the clustering task quickly and provide accurate clustering results. For large-scale datasets, algorithms can complete clustering within a reasonable time by optimising the computing process and adopting technical means such as distributed computing. Thus, on account of the contrasts in data type structures and the various application scenarios, there are different clustering approaches, including partition-based clustering (Prasad et al., 2023), hierarchical clustering (Ran et al., 2023), density-based clustering (Aliguliyev, 2009), etc. Before initiating the clustering process, partition-based clustering methods require the predefined number of clusters or cluster centres to be specified. Representative algorithms of this type of clustering include the K-means clustering algorithm (Celebi et al., 2013), the fuzzy C-means (FCM) algorithm (Hashemi et al., 2023), and the Kmodes algorithm (Cao et al., 2012). Yang and Nataliani (2017) proposed using the attribute reduction theory of rough sets to select data features, then compute the weighted Euclidean distance of the selected features, and select cluster centre points, but the clustering effect was not satisfactory. Lletı et al. (2004) combined genetic algorithms with the K-means approach to enhance the clustering efficiency and accuracy of the K-means method. Saha et al. (2019) proposed a method in light of an optimised forest optimisation approach to solve the K-Modes clustering centre. An attenuation factor was introduced as an adaptive step size to accelerate the clustering speed of the approach, and the arithmetic crossover operation was combined to improve the traditional forest optimisation algorithm's disadvantages of easily falling into local optimal solutions and slow convergence, thereby improving the clustering effect and clustering accuracy.

Clustering algorithms that rely on centre points encounter constraints in distributed settings, including issues related to data distribution and the presence of outliers. It is arduous to establish a comprehensive distributed computing framework for clustering algorithms that do not depend on centre points. The balanced iterative reducing and hierarchical clustering (BIRCH) clustering algorithm is a typical hierarchical clustering algorithm (Li et al., 2021). The algorithm merges multiple clustered feature trees and considers how to maintain the consistency and integrity of the clustering structure, which involves complex merging logic. Density-based clustering techniques are capable of managing clusters that possess intricate shapes and varying sizes, and they exhibit strong resilience against noise and outliers. Representative methods contain density-based spatial clustering of applications with noise (DBSCAN) (Ienco and Bordogna, 2018) and density-based clustering (DENCLUE) (Cai et al., 2024), among others. Latifi-Pakdehi and Daneshpour (2021) studied a hierarchical DBSCAN algorithm, generating a cluster hierarchy to improve clustering performance. Pandey and Shukla (2021) proposed a clustering method based on random sampling (RS) and probability density, which first reduces computational complexity through RS, then proposes a variable density function, and extends it to density-based cluster detection in complex networks. Ding et al. (2023) proposed a method for generating skewed density level samples, aiming to reduce the time required to extract all clusters.

For the goal of better integrating the characteristics of clustering algorithms in distributed environments and address the issues of high computational resource consumption and low clustering efficiency in big data clustering, this paper proposes a two-stage distributed clustering framework for big data, leveraging EDDS-based sampling. He proposed method initially conducts localised clustering computations at individual nodes, subsequently utilising these partial clustering outcomes, extracts representative data samples from each node, then transmits the selected sample data from each node to the central node. Afterwards, further clustering analysis is conducted on the merged sample data at the central node, and the clustering results of the samples are sent back to each local node. Finally, each local node combines its own local clustering results with the sample clustering results from the central node to complete the final clustering label integration. Through the above process, the proposed method achieves a distributed transformation of centralised clustering algorithms, enabling rapid and consistent clustering analysis of global data. Theoretical analysis and numerical experiments show that compared with traditional full-data centralised clustering methods, the two-phase clustering method effectively combines the efficiency of parallel processing and the accuracy of integrated analysis, significantly reducing computational resource consumption while ensuring clustering quality. It is a feasible distributed solution for big data clustering.

2 Relevant technologies

2.1 Introduction to clustering algorithm

Clustering algorithms represent an unsupervised learning approach designed to partition objects within a dataset into distinct groups or clusters, where objects within the same cluster exhibit a high degree of similarity, whereas objects from different clusters show notable dissimilarities (Sisodia et al., 2012). Common clustering approaches include partition-based clustering methods, hierarchical clustering methods, and density-based clustering algorithms. Compared with the other two types of clustering algorithms, density-based clustering algorithms can detect clusters of arbitrary shapes, and they have good robustness to noise and outliers. The representative algorithm is DBSCAN.

DBSCAN stands out as one of the most effective clustering algorithms in machine learning. A key strength of DBSCAN lies in its capacity to detect arbitrarily-shaped clusters while effectively filtering out noise points. DBSCAN scans the entire dataset D and checks whether each element $d \in D$ has a density higher than a certain threshold *minPts*. The density of any given element is found by tallying the count of elements within a range less than *eps* from that element. When the density exceeds *minPts*, the element is deemed a dense pattern. Conversely, if it does not, the element is provisionally labelled as noise. Should the element be dense, it is allocated to a new cluster C, and a breadth-first search (BFS) is initiated on the dense neighbouring elements of d that remain unexamined. After all the iterations within the inner loop, the elements that are neither assigned to a cluster nor temporarily categorised as noise are designated to cluster C.

2.2 Principle of density-based sampling

The basic principle of the divide-and-diverge sampling (DDS) algorithm is to determine the sampling probability based on the distribution characteristics of each node in the original dataset to be studied and the mining task, so that the distribution characteristics of the final generated sample dataset are similar to those of the original dataset (Ros and Guillaume, 2016). Compared with RS, DDS can reduce the sampling ratio in high-density areas and increase the sampling ratio in low-density areas, thus obtaining samples that better reflect the distribution of the dataset.

Given an original dataset T, which is divided into D groups, denoted as $D_i = \{x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{im}\}$, where x_{ij} is the j^{th} data in the i^{th} group, $j = 1, 2, \ldots, m$. Assuming the weight of data x_{ij} in the sample is Q_j, its probability is $P(x_{ij} \mid x_{ij} \in D_i)$, there is the following equation, where c is a constant.

$$\sum_{j=1}^{m_i} Q_j P(x_{ij}) = cm_i \tag{1}$$

Since the weights of data in D groups in the sample are all Q_j, the sampling probabilities are equal, defined as follows. By combining equation (1), equation (2), and equation (3), equation (4) can be obtained.

$$P(x_{ij} \mid x_{ij} \in D_i) = w(j) \tag{2}$$

$$Q_j = \frac{1}{w(j)} \tag{3}$$

$$\sum_{j=1}^{m_i} Q_j P(x_{ij} \mid x_{ij} \in D_i) = \sum_{j=1}^{m_i} w(m_i) \frac{1}{w(m_i)} = m_i \tag{4}$$

Define the data sampling probability in D_i as $w(m_i) = b/m_i^k \ (0 \leq k \leq 1)$, where k is a constant, $k \in [0, 1]$. When $k = 0$, the sampling is RS; when $k = 1$, it indicates the same sample size is drawn from D_i. The sample size m is the sum of the number of samples drawn from D_i, as shown below:

$$m = \sum_{i=1}^{D} m_i w(m_i) = \sum_{i=1}^{D} m_i \frac{b}{m_i^k} = b \sum_{i=1}^{D} m_i^{1-k} \tag{5}$$

The data sampling probability can be obtained from equation (5), as shown in equation (6).

$$w(m_i) = \frac{b}{m_i^k} = \frac{m}{m_i^k} m_i^k \sum_{i=1}^{D} m_i^{1-k} \tag{6}$$

The DDS algorithm divides the original dataset T into different groups D_i, where the number of data in each group D_i is the density of the group. After division, each data point within the same group is equally likely to be sampled, and the sampling probability of different groups is determined by the density of each group (Tabandeh et al., 2022).

3 Improved density deviation sampling algorithm based on variable grid

To address the issue that the DDS algorithm cannot better reflect the characteristics of data distribution, this paper proposes and implements an improved density bias sampling algorithm based on uneven data (EDDS). By introducing grid cell density and trigonometric functions, a better density bias sampling effect can be achieved. In the DDS principle, parameter k is a global parameter, and its value has a vital impact on the final sampling outcome. According to the research results of existing relevant literature, parameter k is usually set to a fixed value. The fixed setting of parameter k has great limitations, mainly because it is difficult to adapt to fixed parameters for various datasets. In this paper, the parameter k is optimised by introducing grid density and trigonometric functions. Compared with the fixed value parameter k, the improved parameter k has a better sampling effect. By setting the corresponding function, under the condition that s and t remain unchanged, the parameter k is adjusted according to the grid density m_i, that is, the higher the grid cell density, the more grid sampling samples. The modified parameter k is as follows:

$$k = s - t\cos\left(\frac{m_i \pi}{2m}\right) \tag{7}$$

where s is a constant, $s \in [0, 1]$; t is a constant, $t \in [0, 1]$, m_i is the grid density, m is the number of samples.

The data sampling probability in D_i is shown in equation (8), where $b = m \Big/ \sum_{i=1}^{D} m_i^{1-\left(\left(s - t\cos\left(\frac{m,\pi}{2m}\right)\right)\right)}$.

$$w(m_i)=\frac{b}{m_i^k}=\frac{b}{m_i^{s-t\cos\left(\frac{m_i\pi}{2m}\right)}} \tag{8}$$

The number of samples extracted in D_i can be obtained from equation (8), as implied in equation (9).

$$m_i w(m_i)=m_i\frac{b}{m_i}=m_i\frac{m}{m_i^k\sum_{i=1}^{D}m_i^{1-k}}=\frac{m m_i^{1-\left(s-t\cos\left(\frac{m_i\pi}{2m}\right)\right)}}{\sum_{i=1}^{D}m_i^{1-\left(s-t\cos\left(\frac{m_i\pi}{2m}\right)\right)}} \tag{9}$$

To better reflect the characteristics of data distribution, this paper proposes the EDDS algorithm. This algorithm normalises the data of each dimension, then introduces the mean square error to calculate the dispersion of each dimension's data. The dimension with the minimum mean square error is selected as the grid division dimension, and then density deviation sampling is performed.

$$\overline{x_{ij}}=\frac{x_{ij}-x_{ij_{\min}}}{x_{ij_{\max}}-x_{ij_{\min}}} \tag{10}$$

where x_{ij} is the jth data in the ith dimension, $x_{ij_{\min}}$ is the minimum value of the jth data in the ith dimension, $x_{ij_{\max}}$ is the maximum value of the jth data in the ith dimension, and $\overline{x_{ij}}$ is the normalised jth data in the ith dimension.

$$\overline{x}=\sum_{1}^{M}\frac{\overline{x_{ij}}}{M} \tag{11}$$

where M is the number of single-dimensional data, $\overline{x}$ is the average value of all data in a certain dimension.

$$\sigma_i=\sum_{1}^{M}\frac{\left(\overline{x_{ij}}-\overline{x}\right)^2}{(M-1)} \tag{12}$$

where σ_i is the mean square error of all data in a certain dimension.

4 Two-phase clustering method based on the improved density deviation sampling algorithm

4.1 Local clustering

To better integrate the characteristics of clustering algorithms in a distributed environment and address the challenges faced, a two-phase clustering method for large-scale data based on the EDDS algorithm is designed. First, clustering of a large data subset is independently performed on each cluster node to obtain preliminary local clustering results. Then, representative data subsets are extracted on each node and aggregated into a sample set that reflects the characteristics of the entire large dataset. Next, further clustering analysis is performed on this sample set. Finally, the clustering results from the above steps are used to integrate the local clustering information from each node and output the comprehensive clustering result for the entire large dataset.

First, define the data block in the distributed big data storage environment: the dataset D is decomposed into a set of multiple data blocks $\{D_1, D_2, \ldots, D_k\}$. If the conditions $D_i \neq \varnothing, \forall i \in \{1, 2, ..., k\}, \bigcup_{i=1}^{k} D_i = D$ are satisfied, then $D_1, D_2, \ldots, D_k$ is called a complete data block of D. If there is also $D_i \cap D_j = \varnothing$, $\forall i, j \in \{1, 2, \ldots, k\}$, $i \neq j$, it is called a completely a-overlapping data block of D.

In distributed clustering, local clustering is an important step, which allows effective management and analysis of data in a distributed environment. The core idea of local clustering is to distribute a large dataset across different nodes, and each node independently performs clustering analysis on its managed data subset. In a distributed computing framework, especially when using Hadoop and its file system HDFS, distributed clustering algorithms can be effectively deployed, where each HDFS data block can be regarded as a unit for local clustering. On each node, clustering analysis can be independently performed on the data block D_i, obtaining the local clustering result of the data block M_i. This step does not involve data exchange across nodes, which can significantly reduce the burden of network communication and improve the processing speed of big data clustering problems.

4.2 Node sampling based on the improved density deviation sampling algorithm

Conducting node sampling in a distributed big data environment is a key data processing step that enables efficient data processing and analysis without having to deal with the entire big dataset. Nodal sampling is usually used to reduce the burden of data processing and to ensure the consistency and stability of computational results through the representativeness of the sampled samples. In this paper, data blocks on different nodes are regarded as different layers of data, and the EDDS algorithm is utilised to perform a stratified sampling method on data blocks on different nodes. Specifically, in a distributed data environment, the dataset held by each node can be regarded as a separate layer. These layers may be defined based on geographic location, data type, user group, or any other logical approach. Stratified sampling allows for parallel processing, speeding up data preprocessing and initial analysis. Sampling each stratum independently results in a sample that is both representative of the population and consistent with the global data characteristics. This technique is adaptable to a multitude of data and node configurations, granting the flexibility to modify the sampling strategy to suit different data strata.

It's important to highlight that node sampling should only be carried out once local clustering has been finalised. Since local clustering results may contain outliers, the sampling method must be designed to ensure that the sample reflects data from each local class and captures all outliers. As a result, along with the EDDS sampling strategy, extra sampling rules are also essential. In this paper, each outlier is treated as an independent class. First, a random data point is sampled from each class into the sample, and then the remaining data is sampled proportionally using the EDDS sampling strategy to ensure the sample's inclusiveness of outliers and classes.

4.3 Sample clustering

In a distributed big data environment, using a central node to perform global clustering analysis and then returning the results to each node is a complex process. It requires completing global clustering by merging the centre points of local clustering, processing boundary points, and calculating overlapping areas, according to different local clustering algorithms. This method obtains local clustering results through local clustering; obtains local representative points through node sampling; and obtains sample clustering results by clustering the fused representative points, which can also be regarded as a global clustering result. Since this method includes two clustering steps, one for local and one for samples, it is called a two-phase clustering method.

1 After completing local clustering and node sampling, the local samples extracted from each node need to be transmitted to a central node. Here, all sample data will be aggregated together, obtaining the sample of the global data S_{total}.

2 Performing a clustering algorithm on the aggregated sample data S_{total} at the central node obtains the two-phase clustering result of the sample M_{sample}. The purpose of this step is to identify the global patterns and structures of data in the entire distributed system.

3 Returning the sample clustering result M_{sample} to the original nodes. Each node will use these sample clustering results to label its local data

4.4 Mapping of two-phase clustering results

After completing the two-phase clustering of the samples at the central node, the cluster labels of the samples are returned to their original nodes. The local nodes need to perform further analysis to integrate the local clustering results M_i and the sample clustering results M_{sample}, with the ultimate goal of mapping the two-phase clustering results of the samples to all the local data, obtaining the final clustering results of the local data M_{final}. Therefore, an effective strategy needs to be constructed to integrate the local and global clustering results, ensuring the consistency of the local data with the global model.

This section proposes a mapping method that depends on the mode of the sample clustering labels. For each local category on the node, the mode of the two-phase clustering labels of the sampled instances is selected as the new label for that category. During the mapping process, since all the local clustering results will be mapped to the sample clustering labels, the final number of clusters mainly depends on the sample clustering results. Therefore, the final number of clusters must be less than or equal to the number of clusters in the sample clustering. It is worth noting that the mode mapping method causes the clusters generated by the local clustering to only merge and become larger during the final integration, but cannot become smaller.

This mapping method can effectively integrate clustering results in a distributed environment, ensuring that the global clustering results are correctly reflected on each local node, which is crucial for ensuring the accuracy and reliability of data-driven decisions. The final clustering mapping actually integrates the results of local clustering and sample clustering, so the two-phase clustering approach in light of node sampling is, to some extent, also a clustering ensemble method.

5 Specific implementation of the two-phase clustering method

As introduced in detail about the process of the above two-phase clustering method, the two-phase clustering approach in light of EDDS sampling proposed in this paper includes four steps: local clustering, node sampling, sample two-phase clustering, and clustering result mapping, as shown in Figure 1.

During the local clustering process, there is no need to restrict the distribution of data blocks, nor is it required that the amount of data in each data block is the same. Data blocks can have significant differences between them. Since the final clustering labels are mainly generated by the results of the sample two-phase clustering, there is no need to restrict the results of local clustering. The clustering results of each data block can be different, and it is not necessary to force each data block to be clustered into the same number of classes. Therefore, after local clustering of the data blocks, multiple classes will exist on the local node that are divided by the local clustering algorithm.

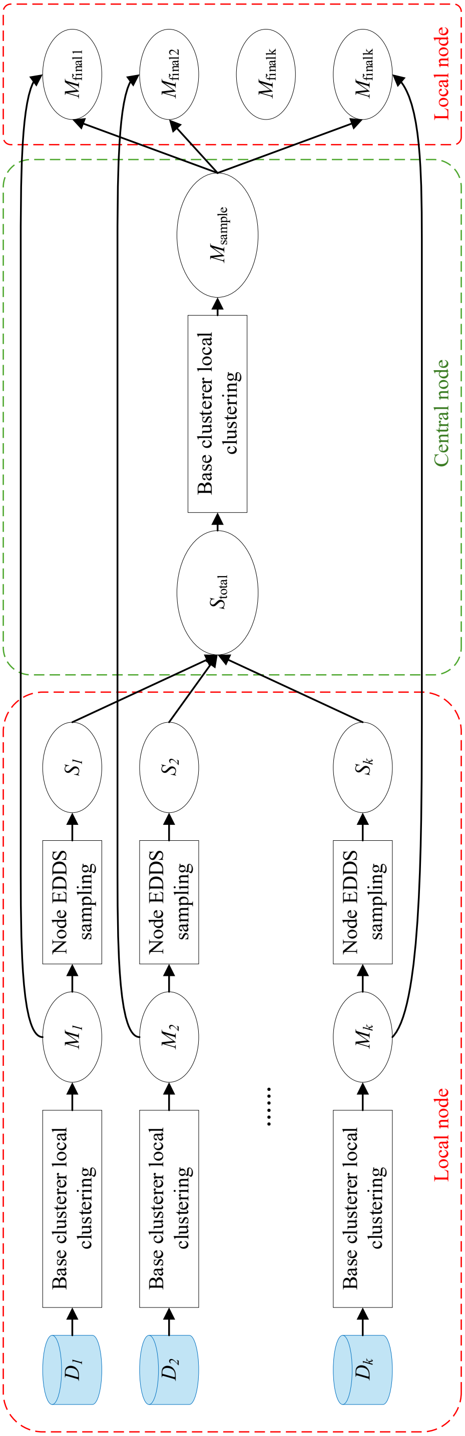
After local clustering, EDDS sampling needs to be performed on the results of local clustering. It should be ensured that at least one sample is extracted from each local class, so that each local cluster has at least one representative point in the sample, which can guarantee that the local clusters can be mapped to the final results. When using some local clustering algorithms, outliers may appear in the resulting clusters. To guarantee that the final clustering results can be traced back to every single point in the dataset, it is necessary to incorporate all outliers identified during local clustering into the sampled data. In pursuit of this aim, two strategies are outlined to ensure the complete coverage of outliers.

1 Independent classification of outliers: Every outlier constitutes an individual class, consisting of a solitary element. The sampling process mandates that at least one sample be obtained from each local class, with the remaining data being sampled according to the EDDS sampling strategy. This approach ensures that all outliers are encompassed within the final sample.

2 Aggregation of outliers into a large class: All outliers can be regarded as a large class, and this large class can be directly included in the sample set to simplify the sampling process and maintain the integrity of the sample. In addition, the remaining samples can be sampled according to the given sampling strategy.

The two-phase clustering method performs clustering analysis on the aggregated sample data at the central node, which can identify the global patterns and structures of data in the entire distributed system. This method uses sampled data instead of traditional central points or complex local cluster representative points, simplifying the process of merging or dividing local clustering results in traditional methods through secondary sample clustering. Therefore, it is suitable for different base clustering algorithms and provides a general distributed processing framework for large-scale data clustering.

Figure 1 Flowchart of distributed two-stage clustering method based on node sampling (see online version for colours)

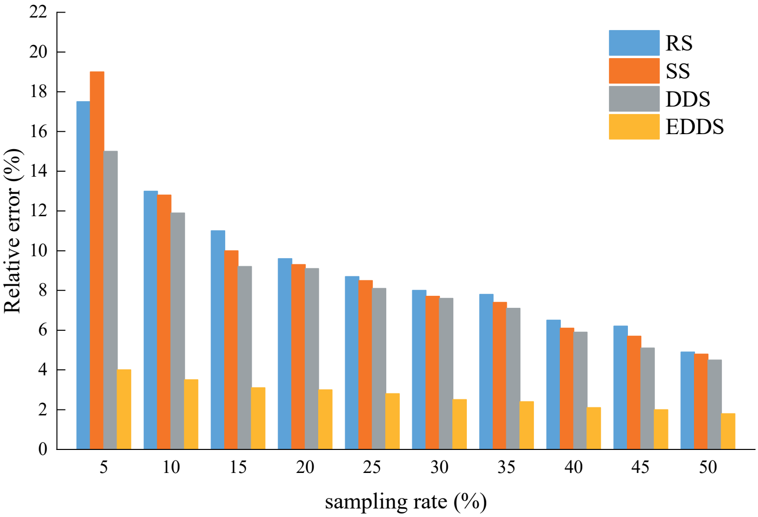


Return the global clustering results to each node. Each node uses these clustering results to label its local data, thereby obtaining a consistent global clustering result. Since each outlier in the local clustering is regarded as a separate class and included in the sample, the local outliers will also be mapped into the two-phase clustering results of the final sample. By using this method, it is ensured that local outliers are both identified and processed within the global clustering results. The mode mapping technique applied in this work effectively mitigates the impact of rare events or outliers on the concluding clustering results. Through plurality mapping, local clusters are tagged by selecting the samples that have the highest frequency of appearance in the cluster, effectively curbing the interference and skewness induced by small-probability events or outliers.

6 Experimental results and analyses

This paper uses the dataset from the literature (Luchi et al., 2019) as a simulation dataset to validate the feasibility and effectiveness of the two-stage clustering method based on node EDDS sampling. A Python 3.8 environment, equipped with an Intel i7-10700 CPU and 32 GB of RAM, was utilised to build the simulation environment. The distributed environment was constructed on a Spark cluster that utilised the YARN scheduler. This cluster comprised five computing nodes, each outfitted with a 48-core Intel (R) Xeon (R) Platinum 8168 CPU operating at 2.70 GHz, 256 GB of RAM, and 2 TB of external storage. Packages used by the cluster include Spark 2.4.0, HDFS 3.0.0, YARN 3.0.0, and JDK 1.8.0 on CentOS Linux release 7.9.2009 (core).

Figure 2 Relative error of different sampling methods (see online version for colours)



Firstly, the performance of EDDS method is analysed, and in this paper, RS method, systematic (SS) sampling method, and DDS method are selected as the comparison methods to evaluate the relative errors of different sampling methods, as shown in Figure 2. It can be observed that the relative error of the method in this paper has a performance improvement of 12%~15% compared to the benchmark algorithm at a sampling rate of 5%. The sampling effect of the EDDS algorithm is significantly better than that of the other three methods, with a higher quality of samples taken, the distributional characteristics of the original dataset effectively preserved, and a better ability to resist noise.

The DBSCAN algorithm was used as the base algorithm for overall clustering (overall DBSCAN) and sample-based two-stage clustering (two-stage DBSCAN). Each data point will contain two clustering labels: an overall clustering label and a two-stage clustering label. When the clustering label of a data point matches the true label, it is considered as a correct clustering result, otherwise it is marked as an error. The accuracy of clustering results for overall clustering and two-stage clustering in the dataset were calculated respectively, and the experiments were repeated for 1000 times to obtain the experimental results as listed in Table 1. It can be found that the clustering accuracy based on the two-stage DBSCAN is much higher than that of the overall DBSCAN, which indicates that the proposed method does not reduce the accuracy of the clustering algorithm in this experiment, but on the contrary, it improves the accuracy of the clustering.

Table 1 Clustering effect of two-stage DBSCAN

Method	*Average accuracy*	*Standard deviation of accuracy*	*Accuracy of 95% confidence interval*
Overall DBSCAN	90.24%	0.1544	(0.8945, 0.9158)
Two-stage DBSCAN	98.51%	0.0631	(0.9668, 0.9792)

Figure 3 and Figure 4 show representative clustering results for one of the 1,000 experiments. It can be observed that due to the random nature of the data generation, there are points where the blue clusters and the green clusters partially overlap. As a result, in the overall data clustering, DBSCAN recognises blue clusters and green clusters as a single category, leading to misidentification of the entire category, which reduces the accuracy of the overall clustering. However, in the second-stage DBSCAN clustering, since only a portion of the data is used, the density of overlapping parts is reduced, and thus the second-stage DBSCAN is able to recognise blue clusters and green clusters separately. This is the reason why the accuracy of the second-stage clustering method is much higher than the overall clustering in this experiment.

This result also illustrates that two-stage clustering is the integration of clustering results from multiple data blocks, which is equivalent to aggregating information from different local perspectives. When mapping the two clustering outcomes, by taking into account both the sample clustering results and the local clustering results, the final category for each sample can be pinpointed more precisely, thereby enhancing the overall clustering performance of the dataset. Simultaneously, conducting local clustering and sample clustering separately facilitates the independent adjustment of parameters for various data distributions. In contrast to using uniform parameters for all data in global clustering, the integration of block clustering and sample clustering offers a better fit for the data's inherent distribution patterns. As a result, employing different parameter settings for local clustering and sample clustering can further boost the accuracy of the clustering process.

Figure 3 The clustering results of the overall DBSCAN (see online version for colours)

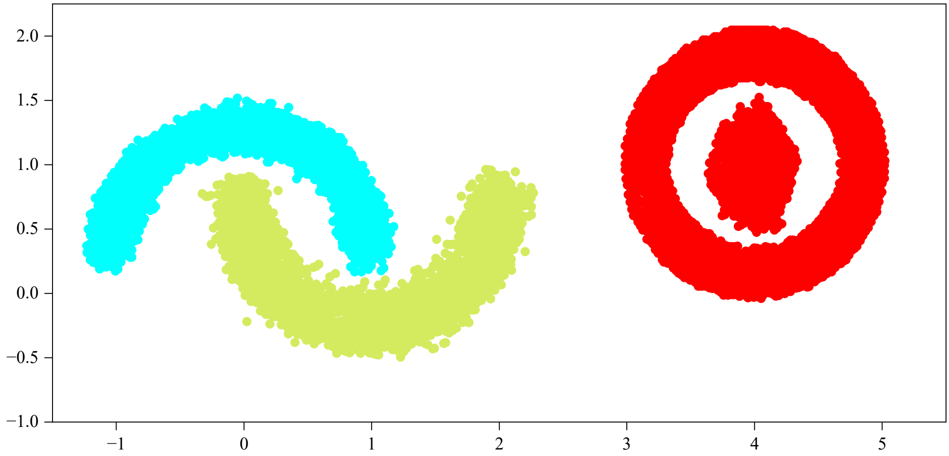
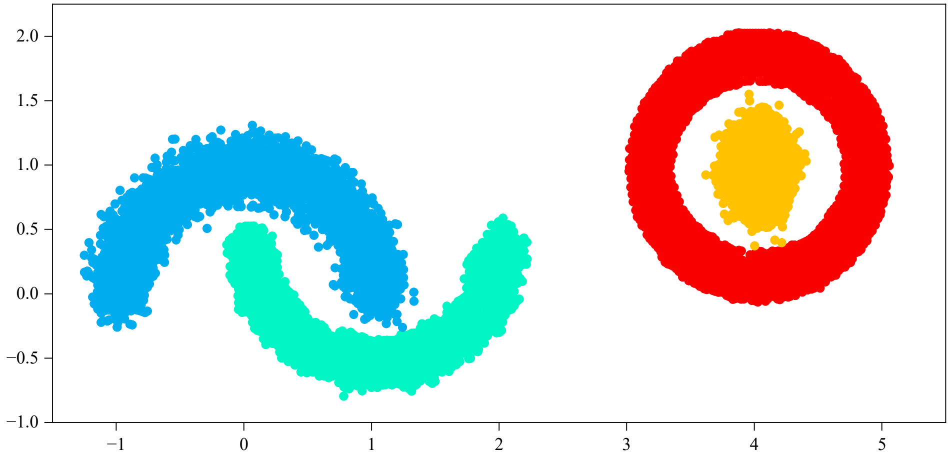


Figure 4 The clustering results of the two-stage DBSCAN (see online version for colours)



Considering that the distributions in the dataset may have a large number of overlapping regions, adjusted mutual information (AMI) (Jiang et al., 2020) was chosen as an assessment metric for the clustering effect. The AMI metric can more accurately assess the consistency between clustering results and the true labels, offering a rational evaluation despite the presence of overlapping clusters. A higher AMI score corresponds to a higher level of similarity between the clustering results and the genuine labels. As demonstrated by the experimental results in Table 2, the mean AMI scores for the overall DBSCAN clustering exhibit a slight edge over those for the two-stage DBSCAN clustering, yet the variance between the two techniques stands at a negligible 0.09%. In order to further substantiate whether the difference was statistically meaningful, a hypothesis test was carried out on the AMI scores pertaining to the two groups. The test yielded a p-value of 0.98, suggesting that the observed difference was highly statistically insignificant. Consequently, it can be inferred that there is no significant difference in the AMI scores between the two-stage DBSCAN clustering and the overall DBSCAN clustering. This finding suggests that although the two-stage DBSCAN clustering employs a different strategy in processing the dataset than the overall DBSCAN clustering, its final clustering results are comparable to the overall DBSCAN clustering. This result is important for understanding the performance of different clustering algorithms on specific datasets, especially when the data distribution is complex and overlapping.

Table 2 Two-stage DBSCAN clustering performance analysis

Method	*Average AMI/%*	*AMI standard deviation*	*AMI 95% confidence interval*
Overall DBSCAN	90.26%	0.1532	(0.7403, 0.7685)
Two-stage DBSCAN	97.95%	0.1546	(0.7429, 0.7679)

7 Conclusions

In this paper, a second-order clustering method based on node sampling is proposed to solve the big data clustering problem with distributed storage. The method first performs local clustering on the data. Then samples are drawn from each local clustering result and second-order clustering is performed on the samples. The final clustering results are obtained by mapping the local clustering results and the second-order clustering results of the samples. This method has the following advantages.

1. Improved sampling effect. Aiming at the problem that the traditional sampling method cannot reflect the data distribution characteristics, an improved algorithm for density deviation sampling based on inhomogeneous data is proposed and realised, which achieves a better density deviation sampling effect by introducing the density of the grid cells and the trigonometric function.

2. A universal distributed clustering framework is provided. This method provides a unified distributed computing model for different clustering algorithms. This means that clustering algorithms, whether centroid-based, density-based, or graph-based, can be efficiently run in distributed environments using this framework, without the need for large-scale modifications or customisations to the algorithms themselves.

3. Improve the computational efficiency of big data clustering. The method in this paper can significantly reduce the real-time data exchange in the network, lower the communication cost and improve the computational efficiency of the clustering algorithm without losing the accuracy of the algorithm. In addition, the method greatly saves computational resources and speeds up processing, making the clustering algorithm more suitable for handling large-scale datasets.

4. Better applicability of big data clustering. Through local clustering and sample clustering of different data blocks, it is convenient to select clustering algorithms according to different data distributions and adjust the parameters of clustering algorithms. This kind of targeted adjustment can better adapt to the characteristics of data distribution and further improve the accuracy of clustering.

The experimental results show that not only the relative error of the EDDS method has a performance improvement of 12%~15% compared with the baseline sampling algorithm, but also the proposed clustering method improves the clustering accuracy.

Acknowledgements

This work is supported by the Henan Province Higher Education Teaching Reform Research and Practice Project (No. 2024SJGLX0691) and the Henan Province Higher Education Teaching Reform Research and Practice Project (No. 2021SJGLX675).

Declarations

All authors declare that they have no conflicts of interest.

References

Aliguliyev, R.M. (2009) 'Performance evaluation of density-based clustering methods', *Information Sciences*, Vol. 179, No. 20, pp.3583–3602.

Cai, T., Lv, J., Ye, Z., Li, X., Zhou, W. and Kochan, O. (2024) 'A streaming data clustering method based on dual strategies improved DENCLUE', *IEEE Access*, Vol. 12, pp.153709–153726.

Cao, F., Liang, J., Li, D., Bai, L. and Dang, C. (2012) 'A dissimilarity measure for the k-modes clustering algorithm', *Knowledge-Based Systems*, Vol. 26, pp.120–127.

Celebi, M.E., Kingravi, H.A. and Vela, P.A. (2013) 'A comparative study of efficient initialization methods for the k-means clustering algorithm', *Expert Systems with Applications*, Vol. 40, No. 1, pp.200–210.

Ding, S., Li, C., Xu, X., Ding, L., Zhang, J., Guo, L. and Shi, T. (2023) 'A sampling-based density peaks clustering algorithm for large-scale data', *Pattern Recognition*, Vol. 136, pp.38–54.

Hashemi, S.E., Gholian-Jouybari, F. and Hajiaghaei-Keshteli, M. (2023) 'A fuzzy C-means algorithm for optimizing data clustering', *Expert Systems with Applications*, Vol. 227, pp.12–26.

Ienco, D. and Bordogna, G. (2018) 'Fuzzy extensions of the DBScan clustering algorithm', *Soft Computing*, Vol. 22, No. 5, pp.1719–1730.

Jiang, H., Jang, J. and Lacki, J. (2020) 'Faster DBSCAN via subsampled similarity queries', *Advances in Neural Information Processing Systems*, Vol. 33, pp.22407–22419.

Latifi-Pakdehi, A. and Daneshpour, N. (2021) 'DBHC: A DBSCAN-based hierarchical clustering algorithm', *Data & Knowledge Engineering*, Vol. 135, pp.19–31.

Li, Y., Jiang, H., Lu, J., Li, X., Sun, Z. and Li, M. (2021) 'MR-BIRCH: a scalable MapReduce-based BIRCH clustering algorithm', *Journal of Intelligent & Fuzzy Systems*, Vol. 40, No. 3, pp.5295–5305.

Lletı, R., Ortiz, M.C., Sarabia, L.A. and Sánchez, M.S. (2004) 'Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes', *Analytica Chimica Acta*, Vol. 515, No. 1, pp.87–100.

Luchi, D., Rodrigues, A.L. and Varejão, F.M. (2019) 'Sampling approaches for applying DBSCAN to large datasets', *Pattern Recognition Letters*, Vol. 117, pp.90–96.

Pandey, K.K. and Shukla, D. (2021) 'Euclidean distance stratified random sampling based clustering model for big data mining', *Computational and Mathematical Methods*, Vol. 3, No. 6, pp.12–26.

Prasad, R.K., Chakraborty, S. and Sarmah, R. (2023) 'Impact of distance measures on partition-based clustering method – an empirical investigation', *International Journal of Information Technology*, Vol. 15, No. 2, pp.627–642.

Rajendra Prasad, K., Mohammed, M., Narasimha Prasad, L. and Anguraj, D.K. (2021) 'An efficient sampling-based visualization technique for big data clustering with crisp partitions', *Distributed and Parallel Databases*, Vol. 39, No. 3, pp.813–832.

Ran, X., Xi, Y., Lu, Y., Wang, X. and Lu, Z. (2023) 'Comprehensive survey on hierarchical clustering algorithms and the recent developments', *Artificial Intelligence Review*, Vol. 56, No. 8, pp.8219–8264.

Ros, F. and Guillaume, S. (2016) 'DENDIS: a new density-based sampling for clustering algorithm', *Expert Systems with Applications*, Vol. 56, pp.349–359.

Saha, I., Sarkar, J.P. and Maulik, U. (2019) 'Integrated rough fuzzy clustering for categorical data analysis', *Fuzzy Sets and Systems*, Vol. 361, pp.1–32.

Shafi, I., Chaudhry, M., Montero, E.C., Alvarado, E.S., Diez, I.D.L.T., Samad, M.A. and Ashraf, I. (2024) 'A review of approaches for rapid data clustering: challenges, opportunities and future directions', *IEEE Access*, Vol. 12, pp.138086–138120.

Shukur, H., Zeebaree, S.R., Ahmed, A.J., Zebari, R.R., Ahmed, O., Tahir, B.S.A. and Sadeeq, M.A. (2020) 'A state of art: survey for concurrent computation and clustering of parallel computing for distributed systems', *Journal of Applied Science and Technology Trends*, Vol. 1, No. 2, pp.148–154.

Sisodia, D., Singh, L., Sisodia, S. and Saxena, K. (2012) 'Clustering techniques: a brief survey of different clustering algorithms', *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, Vol. 1, No. 3, pp.82–87.

Suganya, R., Pavithra, M. and Nandhini, P. (2018) 'Algorithms and challenges in big data clustering', *International Journal of Engineering and Techniques*, Vol. 4, No. 4, pp.40–47.

Tabandeh, A., Jia, G. and Gardoni, P. (2022) 'A review and assessment of importance sampling methods for reliability analysis', *Structural Safety*, Vol. 97, pp.1–16.

Yang, M-S. and Nataliani, Y. (2017) 'A feature-reduction fuzzy clustering algorithm based on feature-weighted entropy', *IEEE Transactions on Fuzzy Systems*, Vol. 26, No. 2, pp.817–835.

Zou, H. (2020) 'Clustering algorithm and its application in data mining', *Wireless Personal Communications*, Vol. 110, No. 1, pp.21–30.