Application of Big Data Mining System Integrating Spectral Clustering Algorithm and Apache Spark Framework

Yuansheng Guo
China Mobile Communications Group, Hunan Co. Ltd, Changsha, Hunan, 410001, China

***Abstract*—Spectral clustering algorithm is a highly effective clustering algorithm with broad application prospects in data mining. To improve the efficient data processing capability of big data mining systems, a big data mining system that integrates spectral clustering algorithm and Apache Spark framework is proposed. It is applied in the big data mining system by combining Hdoop, Spark framework, and spectral clustering algorithm. The research results indicated that after 300 iterations of spectral clustering algorithm, the error value tended to stabilize and drops to 0.123. In different datasets, different error values were displayed, indicating that spectral clustering algorithm had better performance in discrete data processing and smaller testing errors. The minimum time consumed by the comparative system was 37.83 seconds, the maximum time was 55.26 seconds, and the average time was 51.65 seconds. The minimum time consumed by the research system was 18.93 seconds, the maximum time consumed was 32.22 seconds, and the average time consumed was 28.14 seconds. Compared with the comparative system, the research system consumed less time, trained faster, and was more conducive to shortening the clustering running time. The algorithm framework and system raised in the research have good operational efficiency and clustering ability in data mining processing, which promotes the reliability and development of big data mining systems.**

Keywords*—Spectral clustering algorithm; apache spark; big data; data mining

I. INTRODUCTION

The advent of the big data era has led to a proliferation of big data mining technology across a range of industries. Big data technology takes a critical parts in multiple fields with its massive data information and high-intensity processing capabilities. It not only enables efficient analysis of complex data modules, but also has foresight and predictability, and can extract valuable data in a timely manner [1]. Data mining technology, as an emerging discipline, originated in the 1980s with the initial aim of promoting the development of artificial intelligence technology. Modern data mining technologies focus on in-depth exploration of hidden and valuable data to discover new data patterns and valuable information, which has critical guiding significance for enterprise decision-making. Spark, as a big data processing framework, has the merits of high efficiency, scalability, and high fault tolerance, and is therefore broadly utilized in the field of big data mining [2]. This study will explore big data mining techniques from the perspective of Spark. Spectral Clustering (SC), as a classic data mining algorithm, is a clustering algorithm used in graph theory. It achieves node clustering by analyzing the eigenvalues and eigenvectors of the Laplacian matrix of the graph. Many experts and researchers have put forward their own opinions on the research and implementation of big data systems. SC is an unattended clustering algorithm that has been broadly applied in the fields of pattern matching and computer vision due to its excellent clustering capabilities. However, the conventional SC algorithms are ill-suited for large-scale data classification such as that required for hyperspectral remote sensing images. This is due to their high computational complexity and the difficulty of representing the inherent uncertainty of the images [3]. Li et al. employed fuzzy anchor points for the processing of hyperspectral image classification and proposed an SC algorithm based on fuzzy similarity measurement. The findings of the experiment on the datasets of hyperspectral remote sensing images demonstrated the efficacy of the enhanced algorithm. The incorporation of a fuzzy likelihood measure led to the generation of a more resilient similarity matrix. The kappa coefficient obtained by the raised algorithm was 2% higher than that of the traditional algorithm. Furthermore, the raised algorithm achieved superior classification results on hyperspectral remote sensing images when compared with existing methods [4]. The advent of wireless communication technology has led to the generation of a substantial corpus of spatio-temporal user tracking data, which is recorded by wireless communication networks as users utilize these networks to meet a range of needs. To enhance the healthy development of students and facilitate the construction of campus-wide information, Guo Y et al. put forth an SC algorithm based on a multi-level threshold and density combined with common nearest neighbors. Several clustering algorithms were used for detecting anomalies, and four assessment indicators were applied to assess the clustering results. The results indicated that the MSTDSNN-C algorithm exhibited better clustering performance [5]. However, the fact that the clustering model is defined only for the original data and not explicitly extended to out-of-sample data is one of the main drawbacks of SC. To improve its efficiency, Shen D et al. proposed a new modular SC method with out of sample extension, combining a new spectral mapping algorithm based on modular similarity measurement and out of sample extension. The experiment outcomes denoted that the research method had better findings compared to other related algorithms on several data sets [6]. A block distributed Chebyshev-Davidson algorithm was developed by Pang Q et al. to solve the problem of large leading eigenvalues in SC. Through the analysis of the Laplacian matrix or normalized

Laplacian matrix in SC, a scalable distributed parallel version was developed. The results demonstrated its efficiency in SC and its advantage in scalability compared to existing feature solvers used for SC in parallel computing environments [7].

Most existing multi-view clustering methods may be affected by data corruption in terms of technology, leading to a sharp decline in clustering performance. Pan Y et al. put forth a multi-pattern SC method which uses robust bar space segmentation. To address the optimization issue of the weak sparse segmentation, an optimization procedure based on the extended Lagrangian multiplier method was developed. The experiment findings on various benchmark sets showed that the raised method performed well relative to several recent advances in clustering methods [8]. High utility itemset mining is a common utilized data mining method for finding useful patterns. Sethi K et al. proposed a new way to mine itemsets using Spark. They tested it on six real data sets and found that it outperformed other algorithms [9]. When managing very large datasets, the high processing cost of mining data for fuzzy rules increased considerably, and in many cases memory overrun faults are triggered. Fernandez-Basso C et al. used the Spark algorithm to process large amounts of heterogeneous data and find interesting rules. They proposed a measure of interest decomposition based on Alpha cuts and demonstrated through experiments that only 10 equidistant Alpha cuts were sufficient to find all the important fuzzy rules. The efficiency and speed of all proposals were compared and analyzed [10]. Ji L et al. proposed an improved SC-based method of detecting anomalies for anomalous data mining in dam safety monitoring, which introduced natural eigenvalues to select data point edges based on traditional SC. The results showed that this method could avoid the algorithm from becoming bogged down in local topology and improve the efficiency of clustering and anomaly detection. It further confirmed that the method could adjust itself well to the case of discrete distribution datasets, and was more accurate than classical SC methods in both the case of labeling and detecting the data points with unusual anomalies [11].

In summary, regarding data mining, existing researchers in the literature review have some involvement and research on data processing, algorithm classification, and dataset clustering improvement. However, the design and application of clustering algorithms for implementing system data mining are not deep enough, such as data relationship description, architecture design of data processing systems, etc. In order to achieve more efficient and large-scale data processing efficiency, a big data mining method that combines spectral clustering algorithm and Apache Spark framework is proposed compared with literature review. It combines distributed computing framework (such as Spark) to optimize spectral clustering algorithm, realizing parallel processing and fast clustering of large-scale datasets. This is similar to the distributed block Chebyshev Davidson algorithm developed by Pang Q et al. And innovatively introduced spectral clustering algorithm applied to data mining systems, designed a big data mining system architecture, and provided a technical foundation for massive data mining and processing.

The article structure of this study is as follows. Introduction is given in Section I. Section II of this study is dedicated to the integration of the SC algorithm with the Apache Spark framework for the purpose of facilitating the mining of large data sets. This represents a significant area of focus and innovation within the field of big data analytics. Section III presents the experimental verification and analysis of the results obtained from the data set, based on the algorithm designed in the first part. Section IV presents conclusions regarding the experimental results and discusses the limitations of the design, as well as avenues for future research.

II. Methods and Materials

The study adopts spectral clustering algorithm as the core clustering method, which can identify sample spaces of any shape and converge to the global optimal solution, especially suitable for clustering convex structured data. And by constructing a similarity matrix, calculating eigenvalues and eigenvectors, and using classical clustering algorithms such as K-means to cluster the eigenvectors, data clustering analysis is achieved. Firstly, this study combines Hadoop and Apache Spark to investigate the processing techniques of big data. Secondly, the SC algorithm is introduced and combined with the Apache Spark framework to design a framework for a big data mining system.

A. Big Data Technology based on Hadoop and Apache Spark Computing Framework

As the advent of the digital age, big data has become a fundamental element for enterprises to compete. Apache Spark has gained widespread attention in terms of processing speed, fault tolerance, and ease of use. Apache Spark is a high-performance, flexible computing engine that is optimized for processing large datasets. Compared to the traditional big data processing framework MapReduce, Spark has a faster processing speed. This is because Spark stores data in memory instead of traditional storage on disk. Another feature of Spark is that it can perform iterative calculations based on memory. Hadoop Distributed File System (HDFS) can work well on inexpensive hardware and is designed to be fault-tolerant. It provides high throughput for accessing application data and enables fast access to large datasets [12]. Hadoop is a distributed system built by the Apache Foundation, and the HDFS is one of its components [13]. The big data ecosystem of Hadoop is shown in Fig. 1.

HDFS is capable of accessing data in the file system in the form of streams, and the fundamental design of this framework is based on HDFS and MapReduce. HDFS provides storage for substantial quantities of data, while MapReduce offers computational capabilities for similarly large data sets [14]. The MapReduce feature of Hadoop can decompose a large and complex task, allocate scattered subtasks to multiple nodes, and then load them as a single dataset into a data warehouse. The distributed architecture of Hadoop enables the big data processing engine to be situated as proximate to the storage facility as possible. This makes the system relatively suitable for batch operations such as ETL, given that the results of such operations may be transmitted directly from the processing engine to storage. The popularity of Hadoop in the area of big data processing can be attributed to its efficacy in data extraction, transformation, and loading.

Spark is an open-source project under the Apache foundation that provides a distributed computing framework for fast processing of large-scale datasets [15]. Compared to traditional MapReduce, Spark uses memory storage to read and write data faster, avoiding frequent disk I/O operations and improving data processing speed. Spark supports multiple programming languages, such as Scala, Java, Python, and R, making it easy for users to choose their familiar programming language for development. It also provides a resilient distributed dataset (RDD), as shown in Fig. 2 for its structure and running process.

Fig. 2 (a) showcases the structure of the RDD dataset, and Fig. 2 (b) showcases the operational flowchart of RDD. RDD is composed of multiple partitions, each of which is a subset of data that can be distributed across multiple machines for parallel computing. Partitioning is the process of grouping data records with the same attributes together according to specific rules, where each partition is equivalent to a segment of the dataset. This partitioning mechanism enables RDD to support parallel processing and improve computational efficiency.

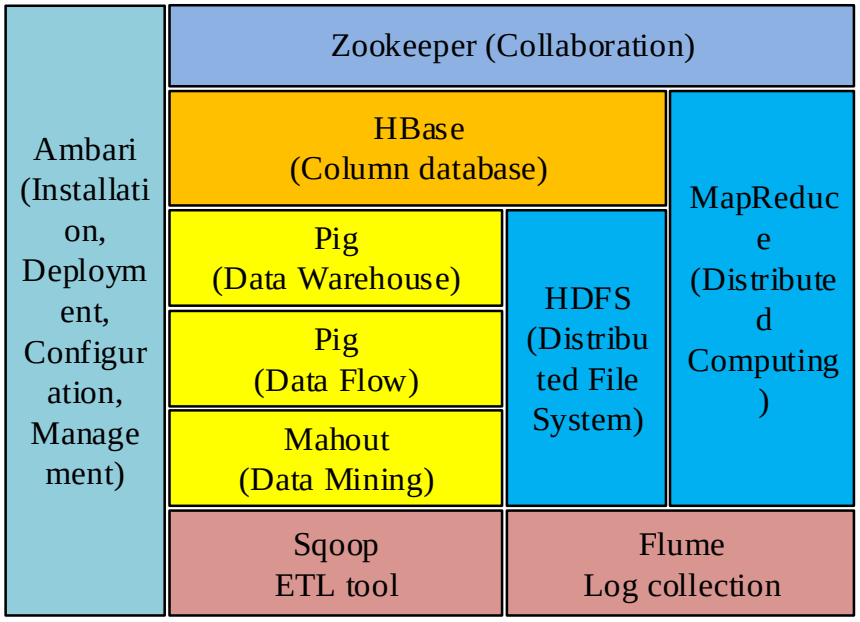
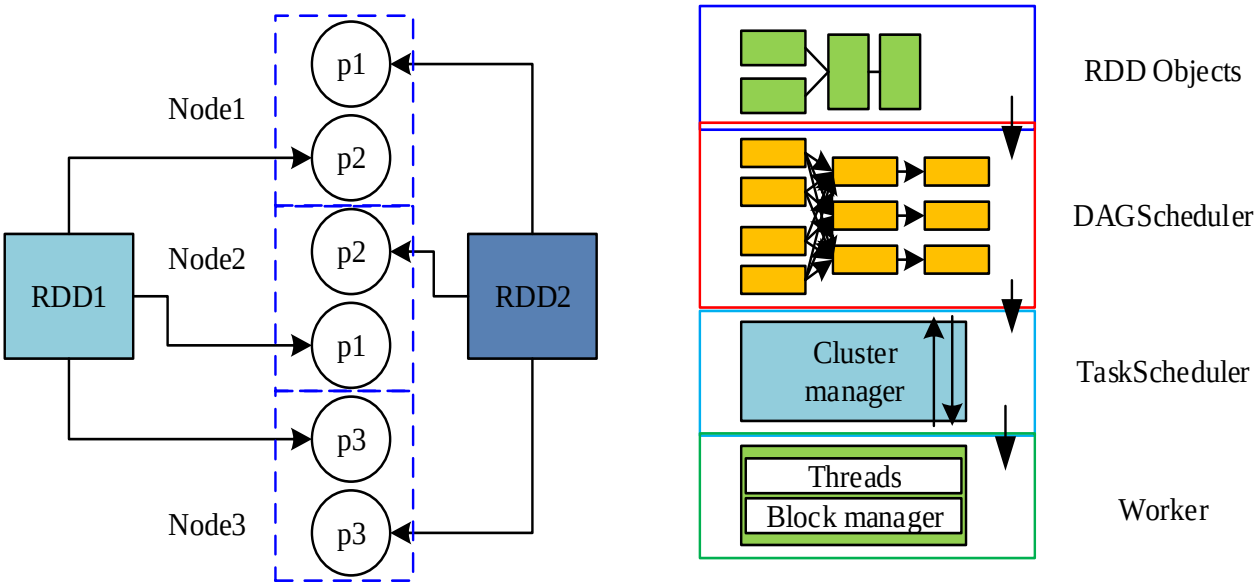


Fig. 1. Hadoop big data ecosystem.



(a) Structure diagram of elastic distribution dataset RDD

(b) Operation flowchart of RDD

Fig. 2. Structure diagram and operation flowchart of RDD dataset.

The running process of RDD in Spark architecture mainly includes the following steps. Firstly, it is necessary to create an RDD object. Secondly, the dependency relationships between RDDs are calculated and a Directed Acyclic Graph (DAG) is constructed. SparkContext is responsible for calculating the dependency relationships between RDDs and building the DAG. DAG represents the structure of the entire computing task, including the conversion and computation between various RDDs. Then the DAG is decomposed into multiple stages, and the DAGScheduler is responsible for decomposing the DAG graph into multiple stages, each stage containing multiple tasks [16]. The tasks in each stage are executed in order of their dependency relationships to ensure the correctness of the calculation results. Afterwards, each task will be distributed by the task scheduler to the Executors on each work node for execution. After receiving the task, the Executor will occupy corresponding resources such as CPU and memory and perform calculations. The calculation results will be returned to the Driver for summarization and processing. Finally, there is the summary and output of the results. After all tasks are completed, the Driver will collect all the results, perform necessary summarization and processing, and finally output the results. This can be done by pulling all data back to the driver end using the collect () method.

This process involves the core mechanisms of Spark's distributed computing framework, including resource allocation, task scheduling and execution, as well as result aggregation and output. In this way, Spark can efficiently process large-scale datasets, achieve parallel and distributed computing, and the running process is shown in Fig. 3.

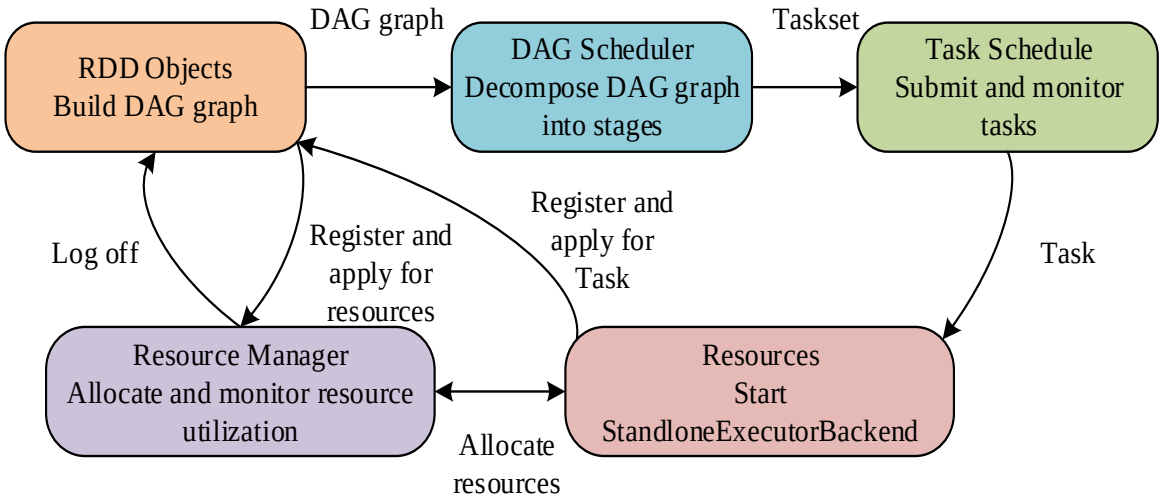


Fig. 3. Spark running process.

The running process of Spark involves environment construction, resource allocation, task decomposition and scheduling, as well as specific behaviors in different running modes, ensuring efficient execution of distributed computing. Firstly, the DAG graph created in the RDD object is decomposed into stages, and Task Schedule is formed through Taskset to submit and monitor tasks.

B. A Big Data Mining System Integrating Spectral Clustering Algorithm and Apache Spark Framework

To achieve efficient mining and analysis of big data, a high-performance SC algorithm is adopted in the study, which can provide better clustering for convex structured data. SC is a clustering method based on graph theory that divides a weighted undirected graph into two or more optimal subgraphs. This is achieved by ensuring that the subgraphs are as similar as possible internally while maximizing the distance between subgraphs [17]. The underlying principle of the SC method is the transformation of the initial clustering problem into an optimal graph partitioning problem. The selection of appropriate eigenvectors for clustering is achieved by calculating the eigenvalues and eigenvectors of the similarity matrix of the sample data points. This method is capable of identifying sample spaces of any shape and converging upon the global optimal solution [18]. The implementation process of SC includes constructing a similarity matrix, calculating eigenvalues and eigenvectors, and using K-means or other classical clustering algorithms to cluster eigenvectors. The SC algorithm has a wide range of applications, including computer vision, pattern recognition, information retrieval, and other fields. Spectral clustering algorithm treats all data as points in space during the clustering process. By slicing the graph composed of all data points, the edge weights between different subgraphs are minimized, while the edge weights within subgraphs are maximized, thus achieving the purpose of clustering. This method overcomes the disadvantage of traditional clustering algorithms (such as K-Means) that may not be able to obtain the global optimal solution on any shaped sample space.

The study will use a directed unweighted graph to represent the dataset $G=(V,E)$, and describe its relationships using a matrix to transform it into a graph/matrix problem. The similarity of data points will be described using functions, and the relationship equation will be constructed as shown in Eq. (1).

$$w_{i,j}=\exp\left(-\frac{\left\|x_i-x_j\right\|^2}{2\sigma^2}\right) \quad (1)$$

In Eq. (1), $w_{i,j}$ denotes the similarity between x_i and x_j corresponding to the i row and j column, and the dataset is represented as $\{v_1, v_2, ..., v_n\}$. x_j and x_j are the data points. A matrix is constructed with a size of n * n based on the relationships between data points. A set matrix that represents the sum of similarity relationships between data points and other points through a degree matrix, as shown in Eq. (2).

$$\begin{cases} d_i = \sum_{j=1}^{n} w_{ij} \\ D = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \cdots & \\ & & & d_n \end{pmatrix} \end{cases} \quad (2)$$

In Eq. (2), D denotes the degree matrix, and d_i represents the degree of data point x_i. In this study, the similarity matrix is constructed using fully connected connections, and a Gaussian kernel function is utilized to construct the similarity distance, as shown in Eq. (3).

$$W_{ij} = S_{ij} = \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \quad (3)$$

In SC algorithms, graph problems involve partitioning problems. From the perspective of graph theory, clustering problems are equivalent to partitioning problems of a graph. The similarity between subgraphs is described by dividing them into different subgraphs. the partitioning principles include minimum cut criterion, normative cut criterion, and proportional cut criterion. The objective of partitioning is to reduce the sum of edge weights that are removed, as a smaller sum of edge weights results in a greater dissimilarity between the subgraphs connected by them, and therefore a greater distance between them. Subgraphs with low similarity can be easily cut off from them [19].

The Laplacian matrix is an important component of SC algorithms and is a matrix used to represent a graph. Given a graph $G = (V, E)$ with n vertices, the vertex set V represents each sample, and the weighted edge E represents the similarity between each sample. The non normalized Laplacian matrix is represented by Eq. (4).

$$L = D - W \quad (4)$$

The properties of the non normalized Laplacian matrix are shown in Eq. (5).

$$f^T L f = \frac{1}{2} \sum_{ij=1}^{n} w_{ij} (f_i - f_j)^2 \quad (5)$$

In Eq. (5), $f = (f_1, f_2, ..., f_n)^T, f \in R^n$ is an arbitrary vector. The normalized Laplacian matrix can be divided into two forms: symmetric and random walk normalized matrices, as shown in Eq. (6).

$$\begin{cases} L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \\ L_{rw} = D^{-1} L = I - D^{-1} L \end{cases} \quad (6)$$

In Eq. (6), L_{sym} represents the symmetric normalization matrix, L_{rw} represents the normalization moment of random walks, W represents the adjacency matrix, I represents the identity matrix, and L represents the non normalized Laplacian matrix. The properties of the normalized Laplacian matrix are shown in Eq. (7).

$$f^T L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 \quad (7)$$

In Eq. (7), d_i and d_j represent the element values of the matrix. The acquisition of SC algorithm requires the partitioning of the graph, transforming discrete problems into continuous problems. The SC algorithm's acquiring process is shown in Fig. 4.

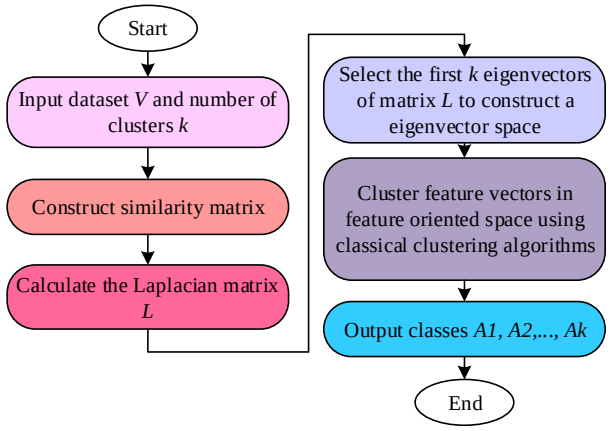


Fig. 4. Spectral clustering algorithm process.

The process mainly includes the following steps. Firstly, it will calculate the similarity between given datasets, and select an appropriate similarity calculation method based on the characteristics of the datasets to build a similarity matrix. On the basis of the similarity matrix, a Laplacian matrix is constructed through regularization processing. The Laplacian matrix can be constructed in two ways: diagonal matrix and adjacency matrix. The eigenvalue decomposition is performed on the Laplacian matrix to obtain a series of eigenvalues and corresponding eigenvectors. The corresponding eigenvectors are selected based on the first K smallest eigenvalues, which form a low dimensional space, and project the original dataset into this low dimensional space [20-21]. The clustering analysis is performed on the projected dataset using the K-means algorithm to get the final clustering results. In addition, to assess the efficacy of SC algorithms, this study uses algorithm time complexity. Firstly, a dataset of n with each data dimension d is set up to construct a corresponding similarity map. After calculating the time complexity, the eigenvalues and eigenvectors of the similarity matrix are calculated. Finally, the corresponding eigenvectors are obtained through dimensionality reduction for clustering. The calculated time map is shown in Fig. 5.

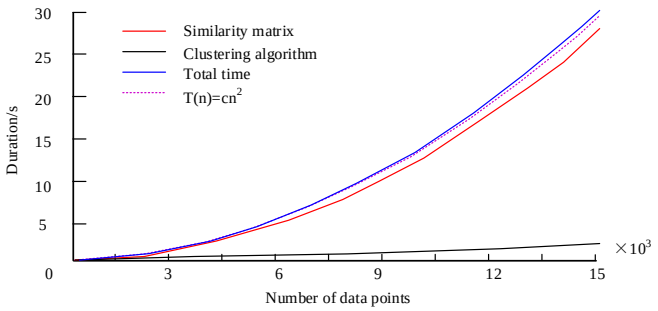


Fig. 5. Calculation time chart of spectral clustering.

Fig. 5 shows the time required for each step of spectral clustering in the dataset. The total time of the algorithm is basically consistent with the fitting function f (n)=cn^2, so the total time complexity of the algorithm is O (n^2), and the construction of the similarity matrix stage consumes the most time. This study is based on SC algorithm and Apache Spark framework to design a big data mining system. The system is broken into three layers of architecture, each layer has interface connections, and from bottom to top are the data layer, business layer, and interaction layer. The data layer accesses files in the data system during the homework process to perform read and save operations on data in the database. The main function of the interaction layer is to display data and receive and transmit user data, providing an interactive operating interface for the website's system operation. The business layer identifies and processes user input information, saves it separately, establishes a new data storage method, reads the data during the storage process, and saves the business logic description code. The system architecture is shown in Fig. 6.

The research first uses the Hadoop and Apache Spark computing frameworks for data processing, and utilizes the distributed computing capabilities of the Apache Spark framework to allocate the computing tasks of the spectral clustering algorithm to multiple nodes for parallel execution, thereby improving the efficiency of the algorithm. By utilizing Spark's RDD (Elastic Distributed Dataset) mechanism, distributed storage and parallel processing of data can be achieved, reducing disk I/O operations and accelerating data processing speed.

This study combines spectral clustering algorithm with Apache Spark framework, which not only optimizes the computational efficiency of spectral clustering algorithm, but also enhances the ability of big data processing. This technological fusion provides new ideas and methods for research in related fields, promoting innovation and development of algorithm technology. By utilizing the distributed computing capabilities of the Apache Spark framework, this study achieved efficient processing and analysis of large-scale datasets. This helps to address the limitations of traditional big data processing techniques in terms of processing speed and fault tolerance, providing strong support for the further development and application of big data technology.

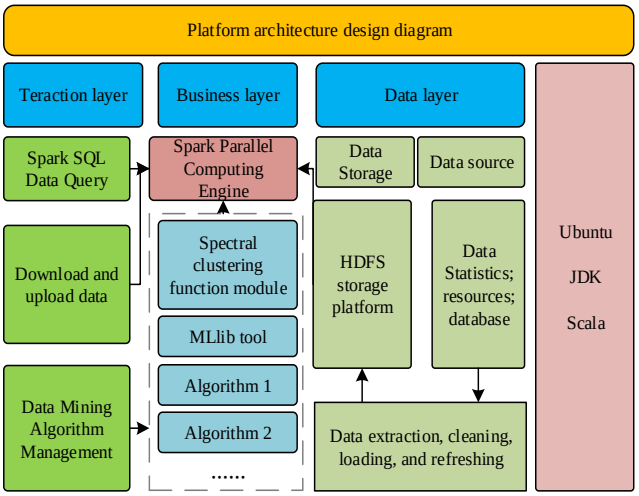


Fig. 6. System architecture design.

III. Results

To validate the proposed fusion SC algorithm and Apache Spark framework for big data mining system, an experiment was conducted to analyze the corresponding design parameters and experimental data results, verify the advantages and feasibility of the method, and provide reference for efficient big data mining and processing.

A. Data Mining System Platform and Environment

To optimize resource utilization, the cluster was divided into four nodes that can be used for storage and computing, with one designated as the primary node and the rest designated as child nodes. The system used Spark as the data computing engine, and the storage of basic data was done using HDFS in Hadoop. It promoted resource coordination between the two through YARN. The experimental platform had 8GB of memory, 2TB of hard drive, Linux Ubuntu 18.04 system, and a 2.9GHz Intel i5 processor. The specific parameters are indicated in Table I.

TABLE I. Specific Parameters

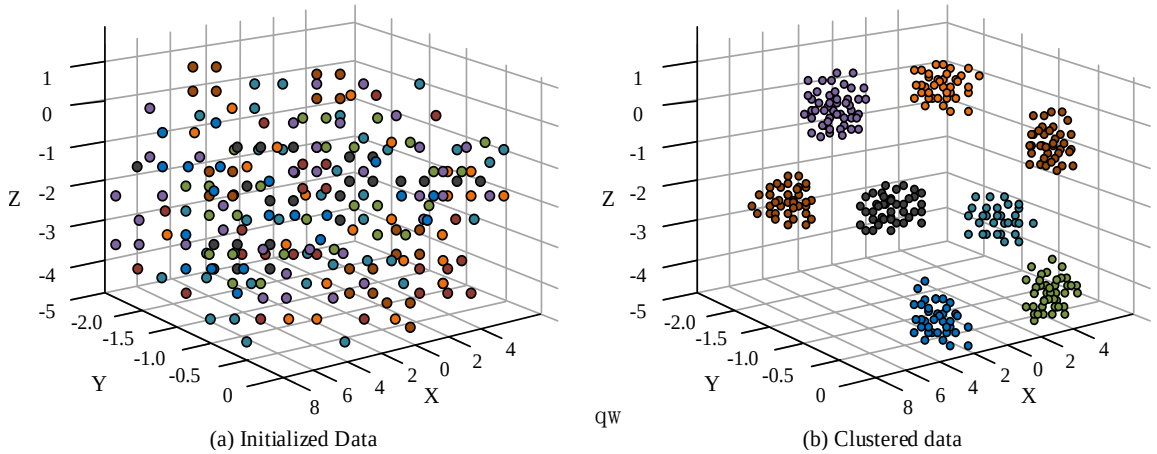
Project	Parameter	Host Name	Address	Node type
CPU	Intel®Core(TM) i7-4790 @3.60GHz	Master	192.168.60.150	NameNode/Master/Worker
Memory	8GB	Slavel	192.168.60.151	DataNode/ Worker
Hard drive	2TB	Slave2	192.168.60.152	DataNode/Worker
Bandwidth	100Mb/s	Slave3	192.168.60.153	DataNode/Worker
Operating system	Linux Ubuntu 1 8.04	/	/	/

B. Data Mining Processing Results and Analysis

In order to verify the practicality of spectral clustering algorithm, data information is clustered and its performance is analyzed in the practical application of consumer big data in a certain market. The cluster diagram is shown in Fig. 7. The 8 clusters in Fig. 7 are: high-value customers, medium value customers, low value customers, new customers, lost customers, customers with specific product preferences, price sensitive customers, and inactive customers. As shown in Fig. 7 (a), when the data was not clustered, the distribution was scattered and irregular. As shown in Fig. 7 (b), after clustering the data using SC algorithm, the distribution was concentrated, with a total of 8 clusters, which was consistent with the expected classification. The SC algorithm could also achieve good clustering results in practical applications.

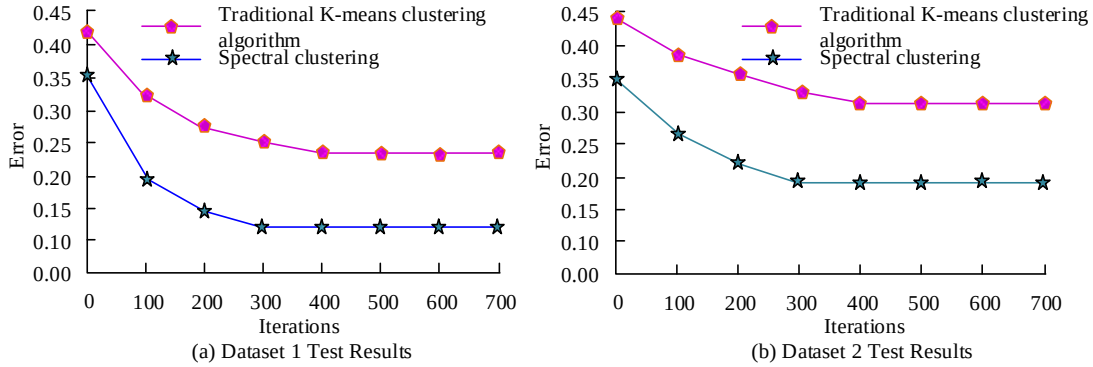
SC algorithm is more effective in processing large amounts of discrete data and is also more suitable for data mining and classification processing. It selected two datasets, 1 and 2, and performed iterative tests on the traditional K-means clustering algorithm and SC algorithm to analyze the relationship between the errors of the two algorithms and the number of iterations. The result is denoted in Fig. 8. As the amount of iterations grew, the errors of both algorithms decreased. In Fig. 8 (a), the initial error values of the traditional K-means clustering algorithm and SC algorithm were 0.425 and 0.356, respectively. After 500 iterations of the traditional K-means clustering algorithm, the error value tended to stabilize and decreased to 0.254. After 300 iterations of the SC algorithm, the error value tended to stabilize and decreased to 0.123. In Fig. 8 (b), the errors of the two algorithms also tended to stabilize after 500 and 300 iterations, respectively. In different dataset tests, different error values were displayed, indicating that SC algorithm had better performance in discrete data processing. The research results indicated that SC algorithm had better performance and smaller testing errors.

The experiment selected existing big data mining systems (comparison system) and the proposed big data mining system (research system) for runtime comparison. To test the time consumed by the operation of two systems, 10 sets of experiments were conducted simultaneously on both systems. The findings are indicated in Fig. 9. From Fig. 9, the minimum time consumed by the comparative system was 37.83 seconds, the maximum time was 55.26 seconds, and the average time was 51.65 seconds. The minimum consumption time of the research system was 18.93 seconds, the maximum consumption time was 32.22 seconds, and the average consumption time was 28.14 seconds. Compared with the comparative system, the research system consumed less time, trained faster, and was more conducive to shortening the clustering running time.



(a) Initialized Data (b) Clustered data

Fig. 7. Data information clustering diagram.



(a) Dataset 1 Test Results (b) Dataset 2 Test Results

Fig. 8. Relationship between error and iteration times.

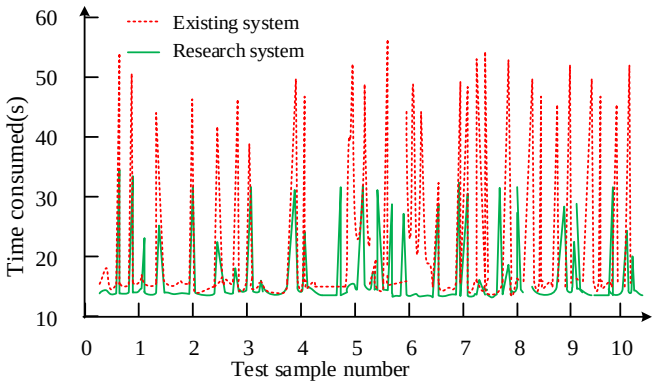
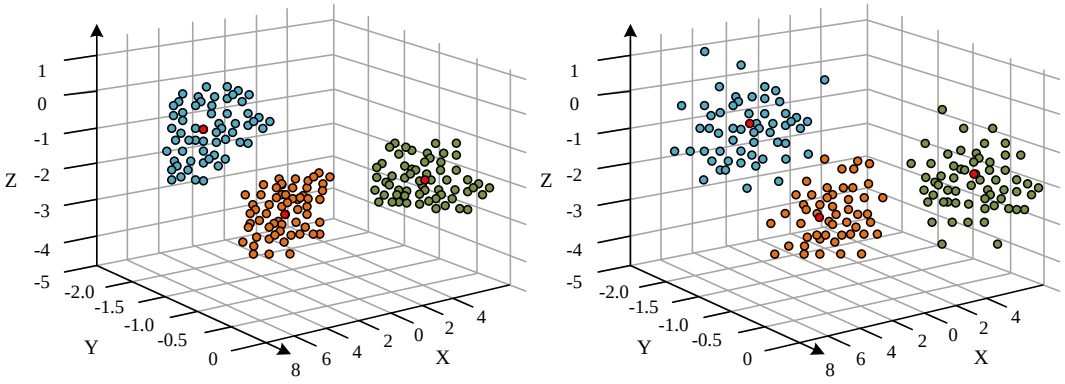


Fig. 9. Comparison of consumption time.

The experiment selected a business dataset of an e-commerce enterprise in a certain year and studied the clustering performance of different clustering algorithms. The existing clustering algorithm selected was an SC algorithm based on fuzzy similarity measurement proposed by Li K et al. Fig. 10 shows the clustering outcomes of the two algorithms. Among them, Fig. 10 (a) showcases the clustering diagram of the SC algorithm. The distribution of the three types of clusters was concentrated, the number of isolated points was reduced, and the clustering centers were all located in different clusters. Fig. 10 (b) showcases the clustering diagram of the original model. The clustering effect of the model on the data was not ideal. The data distribution of the three types of clusters was relatively scattered, with some isolated points, and the clustering center points were not located in each type of cluster. From the clustering graph, the SC algorithm significantly improved the clustering effect of the data.

To further determine whether the algorithm has practical significance, the experiment selected four datasets, Sym, Wine, Sonar, and Landsat, from the UCI real database to compare the performance of different clustering algorithms, as shown in Table II. Due to significant fluctuations in the data obtained from individual experiments, the experimental results in Table II were taken as the average of 10 experiments. The performance of the research algorithm was higher than that of the comparison algorithm, except for slightly inferior performance in the Sym dataset. Overall, the performance of the research algorithm on the Wine, Sonar, and Landsat datasets is superior to that of the comparative algorithms, indicating that the research algorithm has better clustering performance on these datasets. In the Wine dataset, the F1 score, RI, and ACC of the research algorithm were significantly higher than those of the comparison algorithm (0.8259 vs. 0.7447, 0.5034 vs. 0.3816, 0.7022 vs. 0.6185). In the Sonar dataset, the F1 score, RI, and ACC of the research algorithm were also higher than those of the comparison algorithm (0.7328 vs. 0.6551, 0.6184 vs. 0.2836, 0.6745 vs. 0.5337). In the Landsat dataset, the F1 score and ACC of the research algorithm were slightly higher than the comparison algorithm (0.7422 vs. 0.6602, 0.6219 vs. 0.6438), but the RI was slightly lower than the comparison algorithm (0.4403 vs. 0.4072). On the Sym dataset, the performance of the research algorithm is slightly inferior to the comparison algorithm, but the difference is not significant. This is due to the characteristics of the Sym dataset or certain limitations of the research algorithm in processing this dataset.



(a) Comparison algorithm (b) Research algorithm

Fig. 10. Cluster comparison chart.

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT CLUSTERING ALGORITHMS

Algorithm	**Research algorithm**			**Comparison algorithm**		
Index	F1	RI	ACC	F1	RI	ACC
Sym	0.6874	0.4203	0.6397	0.6972	0.4368	0.6515
Wine	0.8259	0.5034	0.7022	0.7447	0.3816	0.6185
Sonar	0.7328	0.6184	0.6745	0.6551	0.2836	0.5337
Landsat	0.7422	0.4403	0.6219	0.6602	0.4072	0.6438

IV. Discussion and Conclusion

A. Discussion

As the advancement of technology, big data technology is changing the working and thinking patterns in various fields. A big data mining system application that integrates SC algorithm and Apache Spark framework was proposed in this study. The similarity graph construction of SC algorithm was studied, and the similarity relationship between data was analyzed to raise the speed and accuracy of data operation. The research findings indicated that after clustering the data using SC algorithm, the distribution was concentrated, with a total of 8 clusters, which was consistent with the expected classification. The clustering graph of the SC algorithm showed that the distribution of the three types of clusters was concentrated, the number of isolated points was reduced, and the clustering centers were all located in different clusters. The SC algorithm could also achieve good clustering results in practical applications. The minimum consumption time of the research system was 18.93 seconds, the maximum consumption time was 32.22 seconds, and the average consumption time was 28.14 seconds. Compared with the comparative system, the research system consumed less time, trained faster, and was more conducive to shortening the clustering running time. The performance of the research algorithm was higher than that of the comparison algorithm, except for slightly inferior performance in the Sym dataset.

B. Conclusion

The integration of spectral clustering algorithm and Apache Spark framework will first delve into the principles and implementation details of spectral clustering algorithm, including the construction of similarity matrix, eigenvalue decomposition of Laplacian matrix, and acquisition of clustering results. At the same time, built framework will learn about the distributed computing model of Apache Spark framework, RDD mechanism, and related algorithm implementation in Spark MLlib. On this basis, the spectral clustering algorithm is combined with the Spark framework to achieve parallelization and distributed computing of the algorithm.

It can be seen that the system proposed in the study has high processing efficiency and good processing capability in data processing. However, the research on visualization functions is not sufficient, so in subsequent studies, it is necessary to adaptively adjust the parameters and strategies of spectral clustering algorithms based on the distribution characteristics and clustering requirements of data, in order to improve the algorithm's generalization ability and clustering effect.

References

[1] Li L, Luo D, Yao W.Analysis of transmission line icing prediction based on CNN and data mining technology.Soft Computing, 2022, 26(16):7865-7870.

[2] Ramalingeswara Rao T, Ghosh S K, Goswami A.Mining user–user communities for a weighted bipartite network using spark GraphFrames and Flink Gelly.The Journal of Supercomputing, 2021, 77(6):5984-6035.

[3] Belcastro L,Salvatore Giampà, Marozzo F, Talia D, Trunfio P, Badia R M.Boosting HPC data analysis performance with the ParSoDA-Py library.The Journal of Supercomputing, 2024, 80(8):11741-11761.

[4] Li K, Xu J, Zhao T, Liu Z.A fuzzy spectral clustering algorithm for hyperspectral image classification.IET Image Processing, 2021,15(12):2810-2817.

[5] Guo Y, Liu M.Spatial-temporal trajectory anomaly detection based on an improved spectral clustering algorithm.Intelligent data analysis, 2023,27(1):31-58.

[6] Shen D, Li X, Yan G.Improve the spectral clustering by integrating a new modularity similarity index and out-of-sample extension.Modern Physics Letters B, 2020,34(11):1-12.

[7] Pang Q, Yang H.A Distributed Block Chebyshev-Davidson Algorithm for Parallel Spectral Clustering.Journal of scientific computing, 2024, 98(3):1-24.

[8] Pan Y, Huang C Q, Wang D. Multiview Spectral Clustering via Robust Subspace Segmentation.IEEE Transactions on Cybernetics, 2020, 52(4):2467-2476.

[9] Sethi K K, Ramesh D, Trivedi M C.A Spark-based high utility itemset mining with multiple external utilities.Cluster computing, 2022,25(2):889-909.

[10] Fernandez-Basso C, Ruiz M D, Martin-Bautista M J.Spark solutions for discovering fuzzy association rules in Big Data.International Journal of Approximate Reasoning, 2021,137(3):94-112.

[11] Ji L, Zhang X, Zhao Y, Li Z.Anomaly Detection of Dam Monitoring Data based on Improved Spectral Clustering.Journal of Internet Technology, 2022, 23(4):749-759.

[12] Wen X, Wu Z, Wu W L.Economic mining of thermal power plant based on improved Hadoop-based framework and Spark-based algorithms.Journal of supercomputing, 2023, 79(18):20235-20262.

[13] Tran D T, Huh J H.Building a model to exploit association rules and analyze purchasing behavior based on rough set theory.The Journal of Supercomputing, 2022, 78(8):11051-11091.

[14] Li J, Shi J, Feng L C.A parallel and balanced S VM algorithm on spark for data-intensive computing.Intelligent data analysis, 2023, 27(4):1065-1086.

[15] Lin L, Tang C, Dong G, Chen Z, Pan Z, Liu J, Yang Y, Shi J, Ji R, Hong W.Spectral Clustering to Analyze the Hidden Events in Single-Molecule Break Junctions.The Journal of Physical Chemistry C, 2021, 125(6):3623-3630.

[16] Yang Q, Li Z, Han G, Gao W, Zhu S, Wu X.An improvement of spectral clustering algorithm based on fast diffusion search for natural neighbor and affinity propagation.The Journal of Supercomputing, 2022, 78(12):14597-14625.

[17] Zhou X, Liu H, Wang B, Zhang Q, Wang Y.Novel Convolutional Restricted Boltzmann Machine manifold learning inspired dynamic user clustering hybrid precoding for millimeter-wave massive multiple-input multiple-output systems:.International Journal of Distributed Sensor Networks, 2021, 17(11):2777-2790.

[18] Wu Y, Chen Y, Ling W.Audit Analysis of Abnormal Behavior of Social Security Fund Based on Adaptive Spectral Clustering Algorithm.Complexity, 2021, 2021(2):1-11.

[19] Zheng C,Zhao J,Guan Q, Zheng C C Q.ADSVAE: An Adaptive Density-aware Spectral Clustering Method for Multi-omics Data Based on Variational Autoencoder.Current Bioinformatics,2023,18(6):527-536.

[20] Zhao J, Guan Q, Zheng C C Q. ADSVAE: An Adaptive Density-aware Spectral Clustering Method for Multi-omics Data Based on Variational Autoencoder.Current Bioinformatics, 2023, 18(6):527-536.

[21] G Mehdi, H Hooman, Y Liu, S Peyman and R. Arif Data Mining Techniques for Web Mining: A Survey. Artificial Intelligence and Applications, 2022, 1(1):3-10.