IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON INNOVATION AND APPLICATION OF INTELLIGENT PROCESSING OF DATA,INFORMATION AND KNOWLEDGE AS RESOURCES IN EDGE COMPUTING

Received March 7, 2020, accepted March 15, 2020, date of publication March 18, 2020, date of current version March 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981632

Data Mining Algorithm for Cloud Network Information Based on Artificial Intelligence Decision Mechanism

YUAN HUANG[1], ZHE CHENG[1], QIANYU ZHOU[2], YUXING XIANG[1], AND RUIXIAO ZHAO[1]

[1]School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China
[2]School of Earth Science and Engineering, Hebei University of Engineering, Handan 056038, China

Corresponding author: Qianyu Zhou (zhouqianyu@hebeu.edu.cn)

This work was supported in part by the Project of the University Science and Technology Research Youth Fund of Hebei Province under Grant QN2019168 and Grant QN2018073.

ABSTRACT Due to the rapid development of information technology and network technology, there is a lot of data, but the phenomenon of lack of knowledge is becoming more and more serious. Data mining technology has developed vigorously in this environment, and it has shown more and more vitality. Based on Spark programming model, this paper designs the parallel extension of fuzzy c-means. In order to enhance the performance of fuzzy c-means parallel expansion, the improvement strategy of k-means during the initialization phase is borrowed, and k-means// is extended to fuzzy c-means to obtain better clustering performance. Combined with Spark's programming model, this paper can obtain extended parallel fuzzy c-means algorithm. Several experiments on the data set of the algorithm proposed in this paper have shown good scalability and parallelism, effectively expanding fuzzy c-means clustering to distributed applications, greatly increasing the scale of the data processed by the algorithm. This improves the robustness of the algorithm and the adaptability of the algorithm to the shape and structure of the data, so that the parallel and scalable clustering algorithm can more effectively perform cluster analysis on big data. Three algorithms were simulated on MATLAB platform. We use simple data sets and complex two-dimensional data sets, and compare with the traditional fuzzy c-means algorithm and fuzzy c-means algorithm based on fuzzy entropy. Experiments show that the scalable parallel fuzzy c-means algorithm not only greatly improves the anti-noise performance, but also improves the convergence speed, and it can automatically determine the optimal number of clusters.

INDEX TERMS Artificial intelligence, data mining, cluster analysis, scalable parallel fuzzy c-means, cloud computing.

I. INTRODUCTION

With the rapid development and increasing popularity of the Internet, modern society is generating data at unimaginable speeds. Mobile communication, website access, logistics transportation, scientific experiments, etc., and ubiquitous social and commercial activities are constantly generating various data, marking that people have entered a brand new era, the era of explosive growth in data big data. From the literal understanding, big data only seems to emphasize the size of the data, but in fact big data is not just ''big'', unpredictable data content and diverse data structures will be difficult problems that data analysis technology needs to solve [1]–[3]. This requires analytical technology to filter out low-value or low-density data, and then mine knowledge gold in high-value or high-density data [4], [5]. In recent years, the prosperity of the information industry has spawned a number of new concepts, technologies, and applications such as the Internet, massive data, massive storage, and analysis, all of which have contributed to the prosperity of big data.

After decades of changes and development, data mining has become an interdisciplinary discipline that integrates relevant knowledge from multiple disciplines such as statistics, databases, machine learning, pattern recognition, intelligence, and parallel computing [6]–[8]. Since the development of data mining, the data objects we have studied

The associate editor coordinating the review of this manuscript and approving it for publication was Ying Li.

have evolved from the original regular data to the current messy and huge data [9]. Therefore, the scope of research is getting wider and wider, and the technical requirements are getting higher. At present, research on large-scale data mining is mainly based on cloud computing platforms, and distributed and parallel processing of mining tasks [10]–[12]. As one of the most widely used cloud computing platforms, Hadoop is bound to have more research based on its Map Reduce programming framework [13]–[15]. Hadoop-based parallel data mining system-PDMiner (Parallel Distributed Miner) is a data mining cloud platform developed by the Institute of Computing Technology of the Chinese Academy of Sciences [16], [17]. PDMiner has integrated many data mining related algorithms and customized the user interface. Users can submit tasks and complete goals through the interface. Fuzzy cluster analysis, as one of the main techniques of unsupervised machine learning, is a method of analyzing and modeling important data with fuzzy theory [18]–[20]. It establishes the uncertainty description of the sample category, which can reflect the real world more objectively. Effectively used in large-scale data analysis, data mining, vector quantization, image segmentation, pattern recognition and other fields, it has important theoretical and practical application value [21]–[23]. The association rule method can be used to find the relationship or rule between the values of two or more variables. The ultimate goal is to find the association network in the entire data set. Associations can be divided into simple associations, temporal associations, and causal associations. With the further development of applications, the research of fuzzy clustering algorithms is constantly enriched [24]–[26]. Relevant scholars have proposed a clustering method based on fuzzy relation composition, but due to the disadvantages of this clustering method that is not suitable for large data sets, people rarely study it [27]–[29]. Slowly, people are trying to study fuzzy clustering using graph theory [30], [31]. In order to improve the classic fuzzy c-means' ability to cluster non-linear data, related scholars introduced a kernel function, which used the kernel function to map the data to a high-dimensional feature space and then calculate the inner product [32], [33]. The kernelization transformed the expressions of membership, distance, and objective function to get a kernelized version of fuzzy c-means [34]–[36]. Relevant scholars have proposed the realization of three different sampling techniques, which are based on non-iterative extended sampling [37]–[39]. Relevant scholars have performed parallelization of the k-Medoids clustering algorithm based on the Map Reduce model, allowing the algorithm to effectively use the Hadoop cluster, and greatly increasing the scale of the algorithm's processing data [40], [41].

This paper introduces the design of parallel expansion algorithm and experimental analysis of fuzzy c-means algorithm. By combining Spark's elastic distributed data operation for parallel expansion design of the algorithm, the algorithm can use cluster for data expansion and cluster expansion data analysis tasks. At the same time, a new method is used to improve the robustness caused by the initialization of the algorithm after expansion. Because cluster computing is costly in a distributed environment, the robustness of the algorithm is very important. An algorithm that is not stable is not suitable for parallel scaling in big data processing scenarios. Experimental comparison with existing Spark-based clustering algorithms proves that the scalable parallel algorithm can correctly and effectively perform cluster analysis of large-scale data.

Specifically, the technical contributions of this paper can be summarized as follows:

First, according to k-means//, an improved initialization process design is given, and the parallel design of the initialization part and the iteration part is given in conjunction with the distributed data set operation provided by Spark, so that the algorithm can be parallelized and highly expanded.

Second, the traditional fuzzy c-means clustering algorithm, fuzzy c-means clustering algorithm with fuzzy entropy, and scalable parallel fuzzy c-means clustering algorithm are applied to simple data sets and complex two-dimensional data sets. The simulation was performed on the MATLAB platform, and the clustering results were evaluated using performance indicators.

The rest of this paper is organized as follows. Section 2 analyzes cloud network information clustering in data mining technology. Section 3 studies the scalable parallel fuzzy c-means algorithm. In Section 4, the simulation of cloud network information data mining algorithm is simulated, and the simulation results are discussed. Section 5 summarizes the full text.

II. CLUSTERING ANALYSIS OF CLOUD NETWORK INFORMATION IN DATA MINING TECHNOLOGY

A. DATA MINING TECHNOLOGY

Data mining is the process of extracting the hidden and potentially useful knowledge and information from a large amount of incomplete, noisy, fuzzy and random data [42]–[44]. This contains two levels of meaning: (1) the data source must be real, large, and noisy; (2) this knowledge is implicit and potentially unknown useful information in advance, and the extracted knowledge is expressed as concepts and rules.

Data mining means a decision support process that looks for patterns in a collection of facts or observations. A pattern is an expression E expressed in the language L. It can be used to describe the characteristics of the data in the data set F. The data described by E is a subset Fe of the set F. E as a model requires it to be simpler than the description method of enumerating all elements in the data subset Fe. The object of data mining is not only a database, but also a file system, or any other data collection organized together, such as cloud network information resources.

Data mining systematic input is the data of the database, the guidance of the information analyst, and the knowledge and rules stored in the knowledge base of the mining system. The selected data is processed in various mining modules to generate auxiliary patterns and relationships, then evaluate

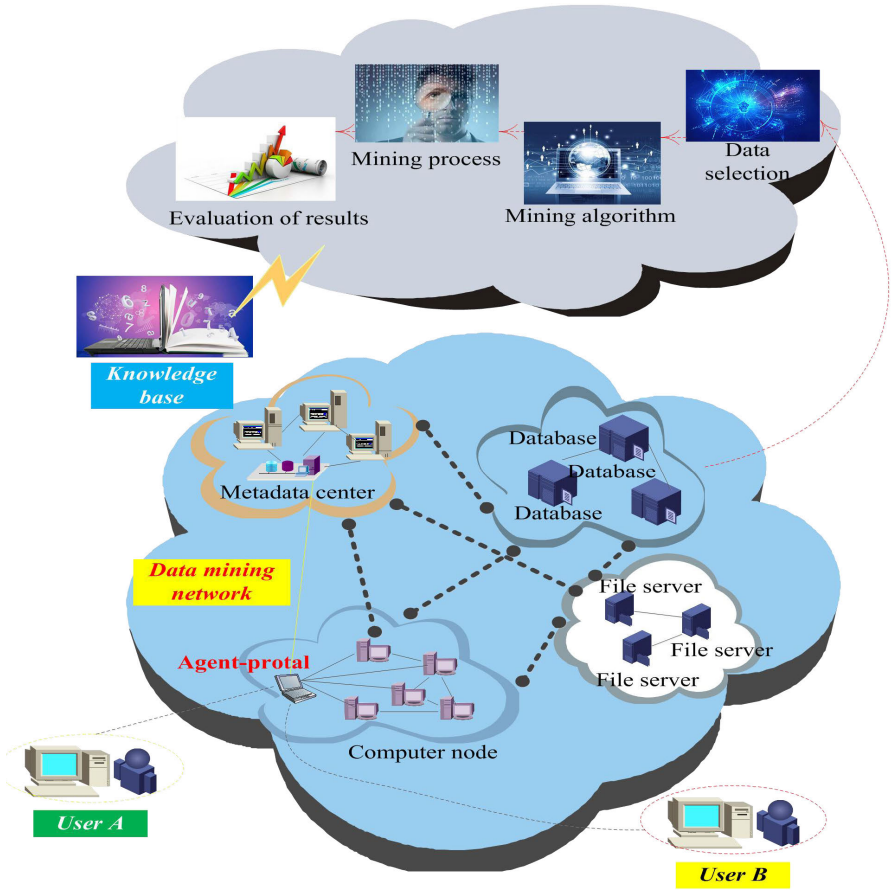


FIGURE 1. Schematic diagram of the topology of the data mining grid.

and interact with analysts to find interesting patterns. Some also need to be added to the knowledge base for subsequent extraction and evaluation [45], [46]. The topological structure of the data mining grid is shown in Figure 1.

Machine learning and data mining are most closely related. The main difference between the two is that the task of data mining is to discover understandable knowledge, while machine learning is concerned with improving the performance of the system. So training a neural network to control an inverted stick is a machine learning process, but not data mining. The main object of data mining is large data sets, such as data warehouses, but generally the data sets processed by machine learning are much smaller, so efficiency issues are crucial to data mining.

1) THE PROCESS OF DATA MINING

The data mining process generally consists of three main stages: data preparation, mining operations, result expression and interpretation.

(1) Data preparation stage: This stage can be further divided into 3 sub-steps: data integration, data selection, and data preprocessing. Data integration combines the data in multiple files or multiple database operating environments to resolve semantic ambiguities, handle omissions in the data, and clean dirty data. The purpose of data selection is to identify the data set to be analyzed, reduce the processing scope, and improve the quality of data mining. Preprocessing is to overcome the limitations of current data mining tools.

(2) Data mining stage: This stage performs actual mining operations.

(3) Results presentation and interpretation phase: They analyze the extracted information according to the decision purpose of the end user, distinguish the most valuable information, and submit it to the decision maker through a decision support tool. Therefore, the task of this step is not only to express the results (for example, using information visualization methods), but also to filter the information. If the decision maker cannot be satisfied, the above data mining process needs to be repeated.

2) THE MAIN PROBLEMS OF DATA MINING

The main problems of data mining are mainly in the following areas:

(1) Mining methods and user interaction issues: This reflects the type of knowledge mined, the ability to mine knowledge at multiple granularities, the use of domain knowledge, specific mining and knowledge display.

(2) Performance issues: This includes the effectiveness, scalability, and parallel processing of data mining algorithms. In order to effectively extract information from a large amount of data in a database, the data mining algorithm must be efficient and scalable. In other words, for large databases, the running time of the data mining algorithm must be predictable and acceptable. On the other hand, the large capacity of many databases, the widespread distribution of data, and the computational complexity of some data mining algorithms are factors that facilitate the development of parallel and distributed data mining algorithms. These algorithms divide the data into parts that can be processed in parallel and then combine the results of each part. In addition, the high cost of some algorithms in the data mining process has led to the need for incremental data mining algorithms. Incremental algorithms are combined with database updates without having to re-mine all data. This algorithm incrementally updates knowledge, modifies and strengthens previously discovered knowledge.

(3) Diversity of database types: There are various data storage methods, including relational databases, data warehouses, transaction databases, advanced database systems, spatial databases, text databases, multimedia databases, heterogeneous databases and heritage databases. It is important to develop data mining system on this basis. Due to the diversity of data types and the different goals of mining, it is unrealistic to expect a system to mine all types of data. In order to mine a specific type of data, a specific data mining system should be constructed, so that for different types of data, we may have different data mining systems. On the other hand, discovering knowledge from different structured, semi-structured, and unstructured data sources with different data semantics poses a huge challenge to data mining.

B. CLOUD COMPUTING ARCHITECTURE

The reason why the cloud computing platform is called ''cloud'' is that it has a huge ''cloud'' network, a powerful computer cluster to provide network computing and services, and it can manipulate virtualization technology to use various terminals to obtain services at anytime and anywhere to concentrate massive resources. Its cloud computing architecture is shown in Figure 2.

1) Cloud client: They login to the request service portal, and send requests to the server through the cloud client to achieve the interaction between the client and the server, realizing user account registration, resource configuration, custom services and other functions.

2) Service directory: After the user obtains the login right, he can customize the required services through this service directory, or cancel the existing service items. The cloud display interface generates corresponding service icons for users to browse based on the existing services.

3) Management system and deployment tools: They manage and deploy the entire cloud, optimize the allocation and utilization of resources, and schedule the effective allocation of resources.

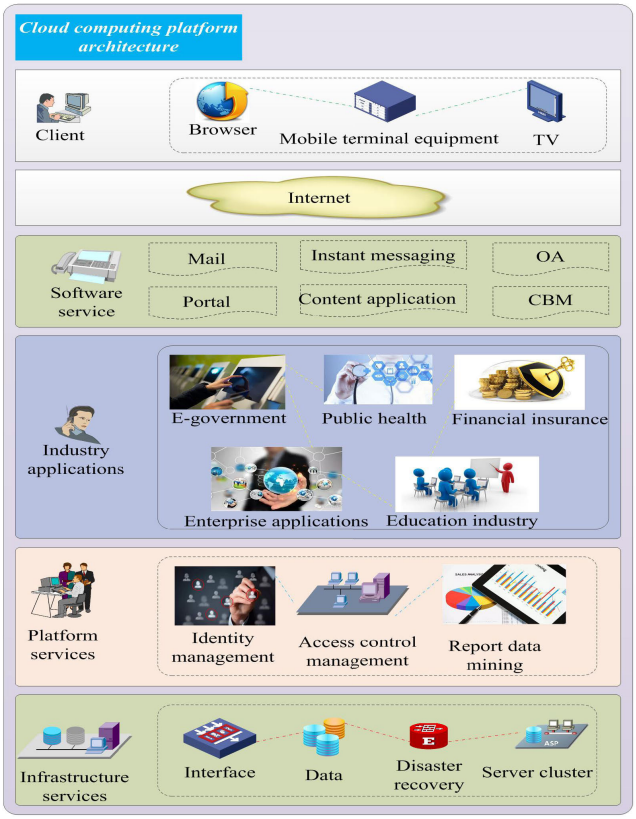


FIGURE 2. Cloud computing architecture.

4) Monitoring: Real-time monitoring of the data usage status of the cloud system is to ensure the reasonable allocation of resources.

5) Server cluster: They use multiple servers to implement parallel computing. Its main duties are to handle requests from multiple users, parallel processing of big data, and backup and storage of data.

Users can log in to the server through the cloud client, select the cloud service they need from the service catalog, and send the request to the server cluster. The server schedules the calculation of the response through the management system and deployment tools, and returns it to the cloud client. Deployment tools allocate resources, and configure web applications.

C. FORMAL DESCRIPTION OF CLUSTERING

Clustering is a major technique in data mining, which is to group a group of individuals into several categories according to similarity, that is, ''things are clustered in categories.'' Its main purpose is to make the distance between individuals belonging to the same category as small as possible, while the distance between individuals in different categories is as large as possible. The fundamental difference between clustering and classification is: in the classification problem, we know the classification attribute of the training example, and in clustering, we need to find this classification attribute value in

the training example. Clustering methods include statistical methods, machine learning methods, neural network methods, and database-oriented methods.

In statistical methods, clustering is called cluster analysis, and it is one of the three major methods of multivariate data analysis (the other two are regression analysis and discriminant analysis). It mainly studies clustering based on geometric distances, such as Euclidean distance and Minkowski distance. Traditional statistical clustering analysis methods include systematic clustering, decomposition, joining, dynamic clustering, and ordered sample clustering, overlapping clustering and fuzzy clustering.

Clustering in machine learning is called unsupervised or teacherless induction. Because compared with classification learning, examples of classification learning or data objects have class labels, while examples to be clustered are not labeled, and need to be automatically determined by the clustering learning algorithm. Conceptual clustering algorithms in the field of machine learning perform clustering through symbol attributes and derive a conceptual description of the clustering. When clustering objects can increase dynamically, concept clustering is called concept formation.

In neural networks, there is a class of unsupervised learning methods: self-organizing neural network methods, such as Kohonen self-organizing feature mapping networks, competitive learning networks, and so on. The SOM method in neural networks clusters data through repeated learning. It consists of an input layer and a competition layer. The input layer consists of N input neurons, and the competition layer consists of m × n = M output neurons, and forms a two-dimensional planar array. The neurons in the input layer are fully interconnected. The LBG method in the vector quantization VQ method can only cluster numerical attributes. The usual approach is to divide all the sets of vectors to be identified into several subsets, and the vectors in each subset have similar characteristics, so they can be represented by a representative quantity. This representative vector is called a codeword, and the set of all codewords is called a codebook.

The cluster analysis problem can be described as: given n vectors in m-dimensional space R_m, we assign each vector to one of the S clusters, so that the "distance" between each vector and its cluster center is the smallest. The essence of the cluster analysis problem is a global optimization problem. Here, m can be regarded as the number of attributes that the sample participates in clustering, n is the number of samples, and S is the number of classifications set by the user in advance.

The vectors X_i and X_j in the m-dimensional space R_m are:

$$X_i = \{X_{i1}, X_{i2}, \ldots, X_{im}\} \quad (1)$$

$$X_j = \{X_{j1}, X_{j2}, \ldots, X_{jm}\} \quad (2)$$

Then the distance between the vectors X_i and X_j can be defined as:

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2} \quad (3)$$

TABLE 1. Possibility of binary variables.

		Object j		
		1	0	Sum
Object i	1	q	r	q+r
	0	s	t	s+t
	Sum	q+s	r+t	p

D. SIMILARITY MEASUREMENT METHOD IN CLUSTER ANALYSIS

So how do you estimate the dissimilarity of different variables? Different variable estimation methods are different.

1) INTERVAL SCALE VARIABLES

The unit of measurement chosen will directly affect the results of the cluster analysis. In general, the smaller the unit selected, the larger the possible range of the variable, and the greater the impact on the clustering result. Therefore, in order to avoid the dependence of clustering results on unit selection, the data should be standardized. After normalization, the dissimilarity between objects is calculated based on the distance. The most commonly used distance metric is Euclidean distance, which is defined as:

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2} \quad (4)$$

Here $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects. When using Euclidean distance, special attention should be paid to the selection of the measured values of the samples, which should effectively reflect the characteristics of the category attributes. Two other well-known methods are Manhattan distance and Minkowski distance, which are:

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \quad (5)$$

$$d(i,j) = \sqrt[q]{|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q} \quad (6)$$

It can be seen from the Minkowski distance: when q = 1, it represents the Manhattan distance; when q = 2, it represents the Euclidean distance.

2) BINARY VARIABLES

A binary variable has only two states: 0 or 1. 0 indicates that the variable is empty, and 1 indicates that the variable exists. For example, given a variable smoker that describes a patient, 1 means that the patient smokes, and 0 means that the patient does not smoke. If the binary variables have the same weight, you can get a table of possibilities with two rows and two columns, as shown in Table 1.

Table 1 reflects the possibility of variable values for the two objects. In the table, q is the number of variables where object i and object j both have the value of 1, r is the number of variables where object i has the value of 1 and object j has the value of 0, s is the value of object i and the value of j is 0, 1 is the number of variables, and t is the number of variables,

whose objects i and j are both 0. The total number of variables is p, $p = q + r + s + t$.

A binary variable is symmetric, if its two states are of equal value and have the same weight. At this time, the evaluation of the dissimilarity between the two objects i and j is the most famous simple matching coefficient, which is defined as follows:

$$d(i,j) = \frac{r+s}{t+s+r+q} \tag{7}$$

That is, the difference is divided by the sum of the same point and the number of different points.

If the output of two states of a binary variable is not equally important, then the binary variable is asymmetric. According to the convention, we encode the more important output, usually a result with a small chance of occurrence, as 1 and the other as 0. Given two asymmetric binary variables, the case where both take 1 is considered more meaningful than the case where both take 0. The calculation of the dissimilarity at this time uses the evaluation coefficient Jaccard coefficient, which is defined as:

$$d(i,j) = \frac{s+r}{s+q+t} \tag{8}$$

E. CLUSTERING ALGORITHMS IN DATA MINING

1) DIVISION CLUSTERING ALGORITHM

Partition-based clustering is the most widely used clustering. The purpose is to divide the data set into several subsets, that is, given a data set with n tuples or records, we construct k groups, each group representing a cluster ($k < n$). For a given k, an initial grouping method can be given, and the grouping is changed by repeated iterations in the future, so that each improved grouping scheme is better than the previous one. Common clustering algorithms include K-means, K-center, CLARA (Clustering LARge Applications), CLARANS (Clustering LARge Applications based upon RAndomized Search), and so on.

Dividing clustering algorithms generally requires all data to be loaded into memory, limiting their application to large-scale data. They also require users to specify the number of clusters in advance, but in most practical applications, the final number of clusters is unknown. In addition, the partitioning clustering algorithm uses only a fixed principle to determine clustering. This makes the clustering result unsatisfactory when the shape of the cluster is irregular or the size is very different.

2) DENSITY-BASED CLUSTERING ALGORITHM

The density-based clustering method uses points with similar density as clusters according to the difference in spatial density, and can be extended in any direction as the density changes. The main idea is: as long as the number of objects or data points in the neighboring area exceeds a certain threshold, clustering continues.

The density-based clustering method treats clusters as high-density object regions divided by low-density regions in the data space. The advantage is that it can be scanned once and any shape and number can be found in the spatial database with ''noise'' clustering.

3) GRID-BASED CLUSTERING ALGORITHM

The grid-based clustering method refers to the use of a multi-resolution grid data structure, which transforms the processing of points into the processing of space, and achieves the purpose of data clustering by dividing the space. It divides the data space into a grid structure of a limited number of units, and all processing is targeted at a single unit. However, all grid clustering algorithms have the problem of quantization scale. In general, the division is too rough, which increases the possibility that objects of different clusters are divided into the same unit. Conversely, too detailed division will result in many small clusters. The usual method is to start by looking for clusters from small cells, then gradually increase the volume of the cells, and repeat this process until satisfactory clusters are found.

III. SCALABLE PARALLEL FUZZY C-MEANS ALGORITHM

A. IMPROVED FUZZY C-MEANS ALGORITHM

Since the fuzzy entropy H (x) is a strictly convex function, the fuzzy entropy can be added as an adjustment function to the objective function of fuzzy c-means.

The objective function added with fuzzy entropy is:

$$J_m(U,V) = w\sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}\log u_{ij} + \sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}^{m} d^2(x_j, v_i) \tag{9}$$

The derived membership is:

$$u_{ij} = \frac{\exp(-\frac{m u_{ij}^{m-1} d_{ij}^2}{w})}{\sum_{i=1}^{c}\exp(-\frac{m d_{ij}^2 u_{ij}^{m-1}}{w})} \tag{10}$$

The cluster center is:

$$v_i = \frac{\sum_{j=1}^{n} x_j u_{ij}}{\sum_{j=1}^{n} u_{ij}^{m}} (i = 1, 2, \ldots, c) \tag{11}$$

where n is the number of data objects, c is the number of clusters, m is the weight index, w is the adjustment factor, d_{ij} is the distance between the j-th data and the i-th cluster center, and u_{ij} is the j-th data belonging to the degree of the i-th cluster center, and it is a probability value.

The traditional fuzzy c-means clustering algorithm only considers the Euclidean distance between the data object and the cluster center, and ignores the interaction between the membership of the same data object and different cluster centers. And fuzzy entropy can just make up for the above shortcomings. At the same time, the introduction of the adjustment factor w can well reflect the distribution characteristics of the data set. The membership calculation formula with fuzzy entropy has a Gaussian distribution, so that data points near

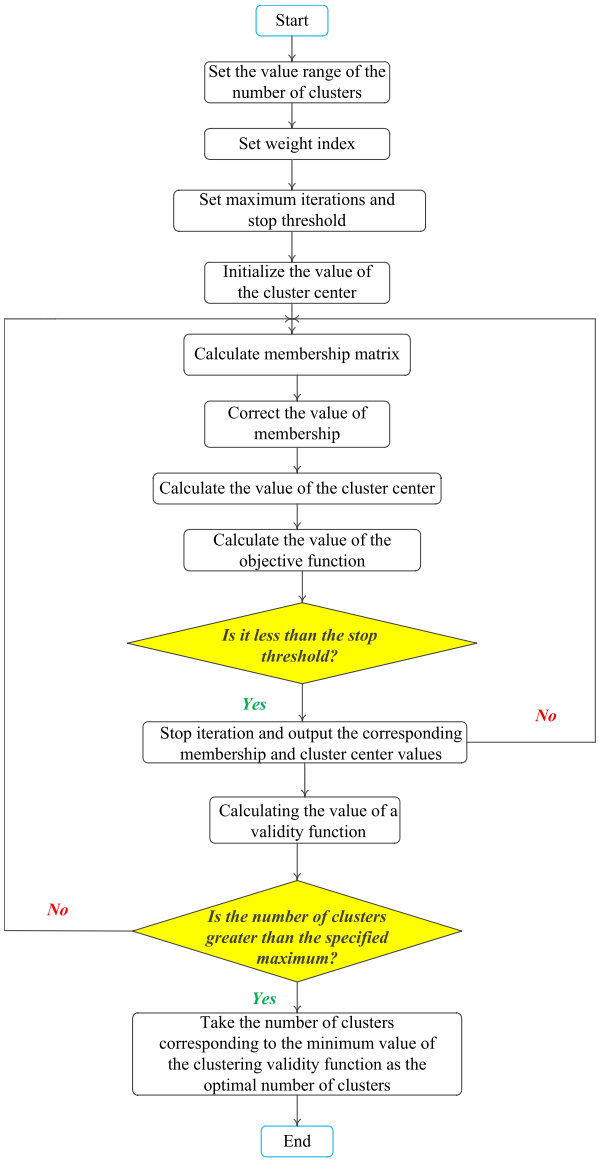


FIGURE 3. Algorithm flowchart.

the cluster center have a higher probability of belonging to the cluster center, and data points farther from the cluster center belong to the cluster. The probability of the center is relatively small, which can effectively suppress the noise data. The algorithm flow is shown in Figure 3.

B. IMPROVED K-MEANS INITIALIZATION ALGORITHM

k-means is a widely used clustering technique designed to minimize the average Euclidean distance between objects in the same genus. Its simple and fast characteristics are very attractive in practice. In practice, k-means usually requires fewer iterations, making it much faster than similar algorithms, but such fastness and simplicity comes at the cost of accuracy, and k-means is sensitive to initialization, but the algorithm uses unlimited randomness. Initialization, although it brings simplicity and efficiency to the execution of the algorithm, the obtained clustering effect varies greatly with the initial clustering center, so in practice, it is often run multiple times to average, which greatly reduces the practicality.

The k-means initialization seeding technology opens a new way to enhance the k-means clustering effect from the initialization stage. By adding an initialization process to the cluster center value based on probability, the speed of the k-means algorithm is significantly improved. And it gives a lower bound on precision guarantee. However, the operation steps of k-means ++ are inherently sequential. The entire sample set must be scanned multiple times and the subsequent operations depend on the previous results. As a result, the algorithm cannot be extended and is not suitable for large-scale data sets.

A parallel implementation of k-means ++, which includes an inherent execution order, is called k-means ++. The algorithm is simple and highly parallel, and it is easy to implement on any parallel computing model. Theoretically, it can be proved that k-means// approximates the optimal solution with a constant factor. Under the premise of ensuring accuracy, it can effectively reduce the number of sample scanning and algorithm iterations.

1) K-MEANS ++

k-means starts by randomly selecting a set of clustering centers, while k-means ++ proposes a special method for selecting these centers. Let $X = \{x_1, \ldots, x_n\}$ be a sample set in a d-dimensional Euclidean space, the number of generics is k, and any sample x ∈ X and a sample subset Y ⊆ X are defined to define a distance:

$$D(x, Y) = \min_{y \in Y} \|x - y\| \tag{12}$$

Let the cluster center set be $C = \{c_1, \ldots, c_k\}$, and define the cost of the sample set Y relative to the cluster center C as:

$$\phi_Y(C) = \sum_{y \in Y} \min_{i=1,\ldots,k} \|y - c_i\|^2 \tag{13}$$

k-means ++ is a fast and simple clustering initialization technology, but it can give an optimal clustering center set O (logk) times different from the optimal clustering center. If you sum the cost of all points in the sample, you let ϕ be the sum of the cost of all points under the optimal cluster center set condition as ϕ_{OPT}. It can be theoretically proved that under the condition of the cluster center set constructed using k-means ++ technology, the corresponding cost sum is E [ϕ] 8 or less (lnk + 2) ϕ_{OPT}. Compared with the original version of k-means, the initial clustering center is randomly selected, and the set obtained by k-means ++ is used as the initial clustering center for the iteration of the Lloyd body of k-means. It reduces the number of iterations and reduces the clustering uncertainty caused by initialization sensitivity, which greatly enhances the robustness of k-means.

2) K-MEANS//

The main disadvantage of k-means ++ is its inherent sequential execution characteristics. To obtain k cluster centers,

the data set must be traversed k times, and the calculation of the current cluster center depends on all the cluster centers obtained previously, which makes the algorithm unable to parallelize. The extension greatly limits the application of the algorithm on large-scale data sets. The main idea of k-means// is to change the sampling strategy during each traversal. After repeated sampling, a set of O (klogn) sample points is obtained. The set is approximated by an optimal solution with a constant factor, and then the O (klogn) points are clustered into k points. The k points are sent to the Lloyd iteration as the initial clustering center. Generally, 5 repeated sampling can get a good initial clustering center.

k-means// is largely inspired by k-means ++. The algorithm first randomly selects the same point as the cluster center, and calculates this point as the sum of the costs ψ of all sample points under the condition of the cluster center. The sampled samples are added to the clustering center set C, and the value of ϕX (C) is updated at the same time, and the sampling continues. The points are expected to be sampled in each cycle, and log ψ sample points are expected to be included in C after the end of the cycle, so the number of samples in C exceeds k. Finally, the weight is based on the number of samples divided into each cluster center in C, and ℓ log ψ points are weighted and then clustered into k points. Generally, the number of samples in C will be much smaller than the number of all samples. Clustering can be done quickly.

C. SCALABLE PARALLEL FUZZY C-MEANS

This section presents a parallel scalable fuzzy c-means algorithm that combines the Spark programming model and special initialization methods. Although Hadoop is now the most popular distributed processing framework, compared to Hadoop, Spark can provide a richer programming model and more effective support for iterative, interactive tasks. In order to get better initialization process and better algorithm performance, this paper introduces a specific initialization method developed from k-means ++ and k-means//. A series of elastic distributed data operations provided by Spark, that is, a series of transformation and behavior functions, are used to implement our parallel scalable fuzzy c-means algorithm.

The scalable parallel fuzzy c-means algorithm combines a special initialization process with the main iteration of fuzzy c-means. Both parts are designed based on the Spark programming model. Spark's distributed programming method uses multiple elastic distributed data operations to build distributed applications. Multiple map operations such as flat map and map partitions are used to distribute parallel tasks from the driver to each worker node. The operations are used to collect child run results from each worker node and return them to the driver. Because the algorithm pseudocode involves many operation functions of elastic distributed data, here we will first use a list of important elastic distributed data operations, as shown in Table 2.

Fuzzy c-means is sensitive to initialization. Initialization has an impact on the iterative process and results.

TABLE 2. Some important elastic data operations involved in the algorithm.

Elastic distributed data operations	Meaning
map(func)	Returns a new RDD after applying the function func to each element on the target RDD
flat Map(func)	Apply function func to each element on the target RDD, then flatten the resulting data into one dimension, and return a new RDD
reduce(func)	Reduce all elements in target RDD using function func
collect()	Returns an array of all the elements of the target RDD
map Partitions(func)	Returns a new RDD after applying the function func to each chunk of the target RDD
collect As Map()	Returns the key-value pairs in the target RDD as a Map to the driver node
reduce By Key(func)	Use function func to combine all values of the same key, and finally return a new RDD consisting of all the key values
take Sample(num)	The target RDD randomly samples num elements and returns the result array

Poor initialization can cause too many iterations and the result converges to a local optimum. Random initialization cannot guarantee a stable iterative process and clustering quality. In the scenario of big data analysis, because cluster computing is used to perform distributed calculations, the cost of each task analysis is relatively high. Therefore, the algorithms commonly used in big data analysis generally can guarantee a relatively stable algorithm process and quality. The algorithm of averaging multiple operations is usually not used for large-scale calculations in a distributed environment. The introduction of the popularized k-means// can get an approximate optimal initial cluster center, which improves the speed of algorithm convergence and reduces the number of iterations. It stabilizes the algorithm iteration process while ensuring the quality of the algorithm and avoiding the algorithm in a distributed environment.

The algorithm initialization part generalizes the algorithm idea of k-means// sampling by probability and subsampling to obtain a faster convergence and higher quality initialization result. Sampling by probability refers to sampling according to the probability that the individual's contribution to the clustering objective function accounts for the objective function and value of the entire sample. Sub-sampling refers to that during the initialization process, a number of samples are probabilistically sampled in each of the blocks in a distributed manner, and then the obtained samples are sampled locally to

TABLE 3. Values of membership for each data point.

Data point number	Traditional FCM			FCM with fuzzy entropy		Scalable parallel FCM	
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 1	Class 2
1	0.20	0.74	0.73	0.02	0.01	0.04	0.01
2	0.71	0.72	0.72	0.23	0.21	0.43	0.39
3	0.63	0.73	1.01	0.17	0.14	0.02	0.01
4	0.73	0.81	0.74	0.01	0.01	0.05	0.03
5	0.72	0.74	0.72	0.09	0.07	0.12	0.09
6	0.71	0.73	0.75	0.02	0.01	0.10	0.08
7	0.76	0.50	0.69	0.02	0.01	1.17	1.14
8	0.74	0.72	0.73	0.04	0.03	0.09	0.06
9	0.72	0.75	0.76	0.09	0.06	0.07	0.05
10	0.74	0.74	0.56	0.07	0.04	0.05	0.02
11	0.76	0.74	0.75	0.03	0.01	0.07	0.03
12	0.76	0.76	0.74	0.08	0.06	0.06	0.02
13	0.74	0.65	0.81	0.05	0.03	0.03	0.01
14	0.75	0.77	0.76	0.02	0.01	0.31	0.27
15	0.73	0.75	0.78	1.01	0.94	0.04	0.02
16	0.75	0.74	0.73	0.08	0.06	0.01	0.01
17	0.74	0.76	0.74	0.18	0.13	0.06	0.04
18	0.76	0.73	0.76	0.02	0.01	0.03	0.02

obtain c samples as the initial clustering center. The algorithm can be roughly divided into three stages. Firstly, the sampling is based on the percentage probability of the sample cost as a percentage of the total cost. Finally, a fuzzy c-means clustering is performed locally on the driver side, and the few samples obtained are clustered into c cluster centers as the output of the initialization process.

Because the algorithm design is based on the Spark programming model and uses the data operations and intermediate multiplexing provided by the model, the function primitives are directly programmed into Spark when the algorithm is expressed. The meaning of these functions can be seen in Table 2. The algorithm input includes the object file submitted to the shared file system, a parameter init for the number of iterations of the initialization process, and the sampling factor ℓ in k-means ℓ. The center set will be sent to the main iteration as the initial cluster center.

The algorithm steps are expressed using the model of elastic distributed data operation. The target file is created as elastic distributed data, and any point in the data set is taken as the first initial clustering center. The core iteration in the initialization process is developed from the k-means // algorithm. Given a constant value for the specific number of iterations, k-means // thinks that iterations can achieve good results after five or more iterations. In the iterative loop, when the current clustering center is C, the sum of the sample costs of all samples X is:

$$\phi = \sum_{i=1}^{n} \phi(p_i, C) \tag{14}$$

After explaining the meaning of each step in the algorithm initialization process, we describe the algorithm idea of the entire initialization process:

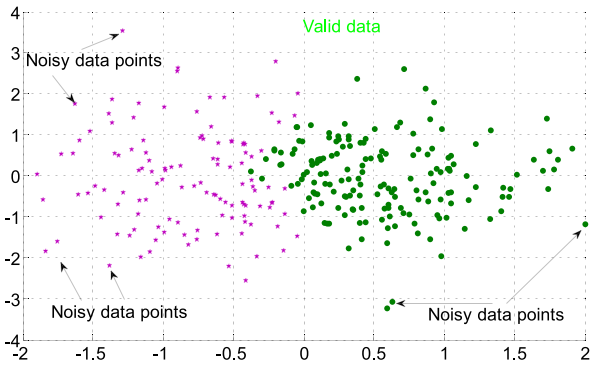


FIGURE 4. Simple data set.

(1) The first stage is distributed probability sampling of the entire sample. Each data set is first flattened into one dimension, the sample cost of each sample is calculated one by one, and then the total cost of the sample set is reduced and added. The total cost obtained is sent to each block, and samples are sampled according to probability independently in each block. The samples obtained from each block are collected, and a small number of samples with a large proportion of the total cost of the sample are collected.

(2) In the second stage, a small number of samples obtained are weighted and sampled according to weighted probability. We divide other samples into these sample categories, and calculate the number of samples divided into these small sample categories in a distributed manner, and use this as the weight when the sample is used as the cluster center, that is, consider these small samples as the cluster center.

(3) The third stage performs a local fast fuzzy c-means clustering. A smaller number of samples are quickly clustered

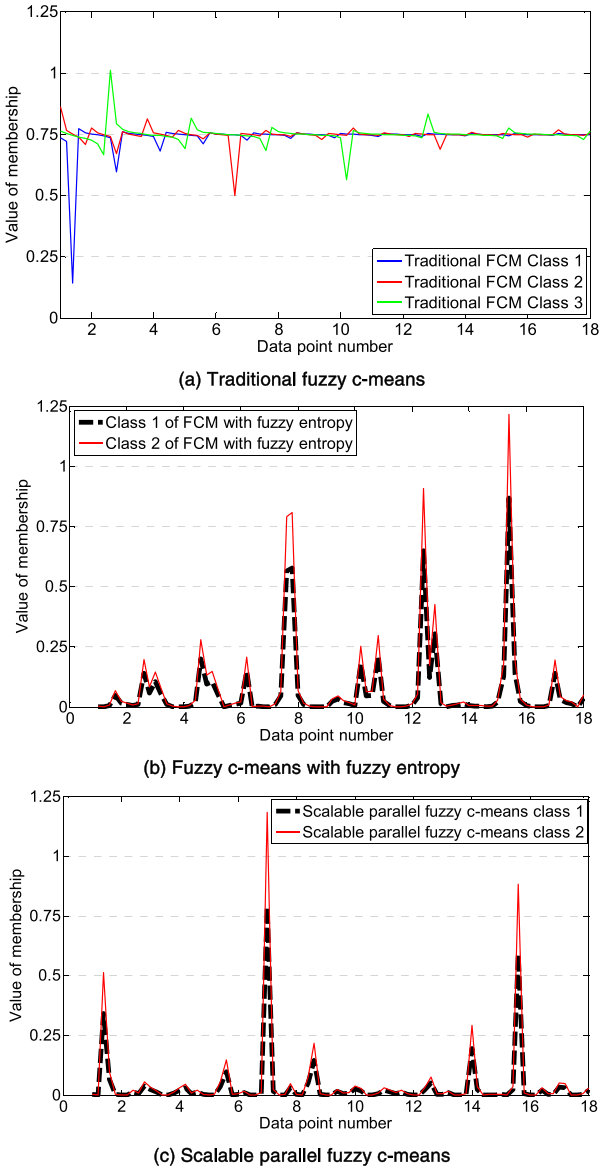


FIGURE 5. Trend graph of membership of each data point corresponding to the three algorithms.

into c, and these c samples are used as the initial cluster center set.

The algorithm is designed in parallel and implemented based on a distributed model. It has high scalability. Such scalability is mainly reflected in three aspects: processing scale expansion, vertical expansion, and horizontal expansion. Processing scale scalability mainly refers to the scalability of the algorithm in the size of the data that can be processed. The algorithm can effectively support data analysis tasks that are applied without any modification to the expansion of the data volume. Both vertical and horizontal scalability are related to the distributed architecture selected by the algorithm. For vertical scalability, if the cluster is upgraded by adding computing resources such as processors, the algorithm can effectively use the application performance

TABLE 4. Performance comparison of three algorithms.

Clustering algorithm	Traditional FCM	FCM with fuzzy entropy	Scalable parallel FCM
Cluster validity function value	9.01	6.12	5.11
Accuracy (%)	91.02	99.11	99.02
Precision (%)	89.10	99.04	99.06
Sensitivity (%)	99.04	98.23	99.28
Specificity (%)	52.03	99.18	98.30

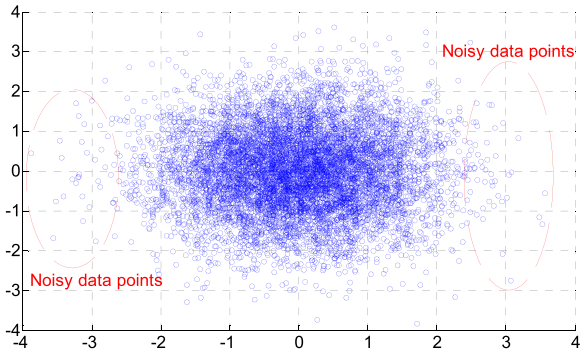


FIGURE 6. Two-dimensional data with Gaussian noise.

improvement brought by the cluster upgrade without modification. In terms of horizontal scalability, if more servers are added to the cluster to enhance the computing power of the cluster, there is no need to modify the algorithm, which can provide large-scale cluster analysis support.

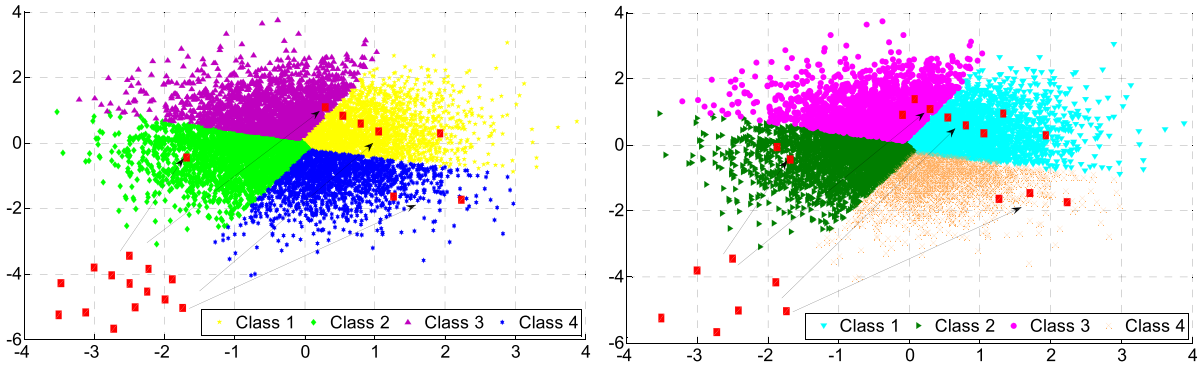
IV. SIMULATION AND RESULTS ANALYSIS OF DATA MINING ALGORITHMS

A. EXPERIMENTS ON A SIMPLE DATA SET

A set of artificial simple data sets is shown in Figure 4.

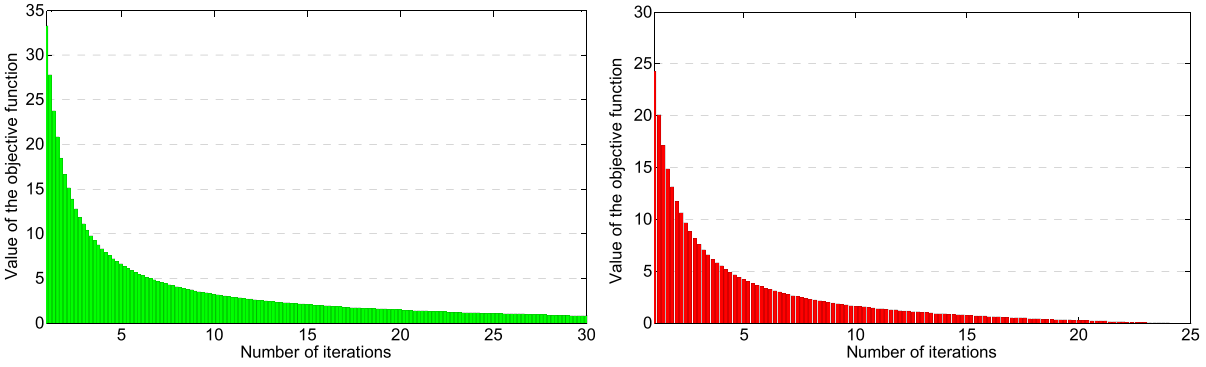
This data set includes valid data and noise data. Randomly we select 18 numbers in this set of data for numbering, from 1 to 18. Assume that the noisy data points are numbered 8 and 9, and then we apply the combined algorithm to this data set. The simulation results of the three algorithms are compared. Table 3 lists the membership values of each data point using the traditional fuzzy c-means clustering algorithm, fuzzy c-means clustering algorithm with fuzzy entropy, and scalable parallel fuzzy c-means clustering algorithm.

Figure 5 shows the trend graph of membership of each algorithm. It can be seen that the traditional fuzzy c-means clustering algorithm has poor anti-noise performance. The noise data is regarded as valid data during the clustering process. When the clustering validity function is added, the two types of data are aggregated into three categories, and the noise data is aggregated into one category as valid data, which leads to incorrect clustering results. The fuzzy c-means clustering algorithm with fuzzy entropy and scalable parallel membership of noisy data points 8 and 9 are close to 0. When



(a) Fuzzy c-means clustering algorithm without opponent suppression (b) Fuzzy c-means clustering algorithm without opponent suppression

FIGURE 7. Convergence trajectory of the cluster center.



(a) Fuzzy c-means clustering algorithm without opponent suppression (b) Fuzzy c-means clustering algorithm without opponent suppression

FIGURE 8. Convergence trend of the objective function.

adding the clustering validity function to optimize the number of clusters, it is not affected by the noisy data. The correct clustering results are obtained and the anti-noise performance is good.

Table 4 shows the values of the clustering effectiveness functions, accuracy, precision, sensitivity, and specificity corresponding to the three algorithms. The scalable parallel clustering efficiency function has the smallest value, and its performance is better than fuzzy c-means clustering algorithm based on fuzzy entropy, which greatly improves the performance of traditional fuzzy c-means algorithm. The traditional fuzzy c-means algorithm has deviations in accuracy, precision, sensitivity, and specificity, because the noise points are considered as valid data for clustering. But adding fuzzy entropy and scalable parallel c-means algorithm has better anti-noise performance.

B. EXPERIMENTS ON COMPLEX 2D DATASETS

A set of artificially complex two-dimensional data sets is shown in Figure 6.

The Gaussian noise with an average value of 0.51 and a variance of 0.216 was artificially added to this data set, and the combined algorithm was applied to this data set.

Figure 7 shows the clustering center convergence trajectory of the fuzzy c-means algorithm without the opponent suppression method and the fuzzy c-means algorithm with the opponent suppression method. It can be seen from the comparison that the convergence rate of the clustering center of the fuzzy c-means clustering algorithm with the adversary suppression method is significantly faster, and the optimal number of clusters c is automatically determined to be 4 due to the addition of the clustering validity function.

Figure 8 shows the convergence trend of the objective function of the fuzzy c-means algorithm without the opponent suppression method and the fuzzy c-means algorithm with the opponent suppression method. It can be clearly seen that the objective function of the fuzzy c-means clustering algorithm with opponent suppression converges to the minimum value quickly, while the traditional fuzzy c-means clustering algorithm has a relatively slower convergence of the objective function.

Figure 9 shows the clustering results of the combined fuzzy c-means clustering algorithm. It can be seen that due to the addition of the clustering validity function, each algorithm can automatically and accurately determine the optimal number of clusters $c = 4$. The traditional fuzzy c-means algorithm has poor anti-noise performance, and clusters the noisy data

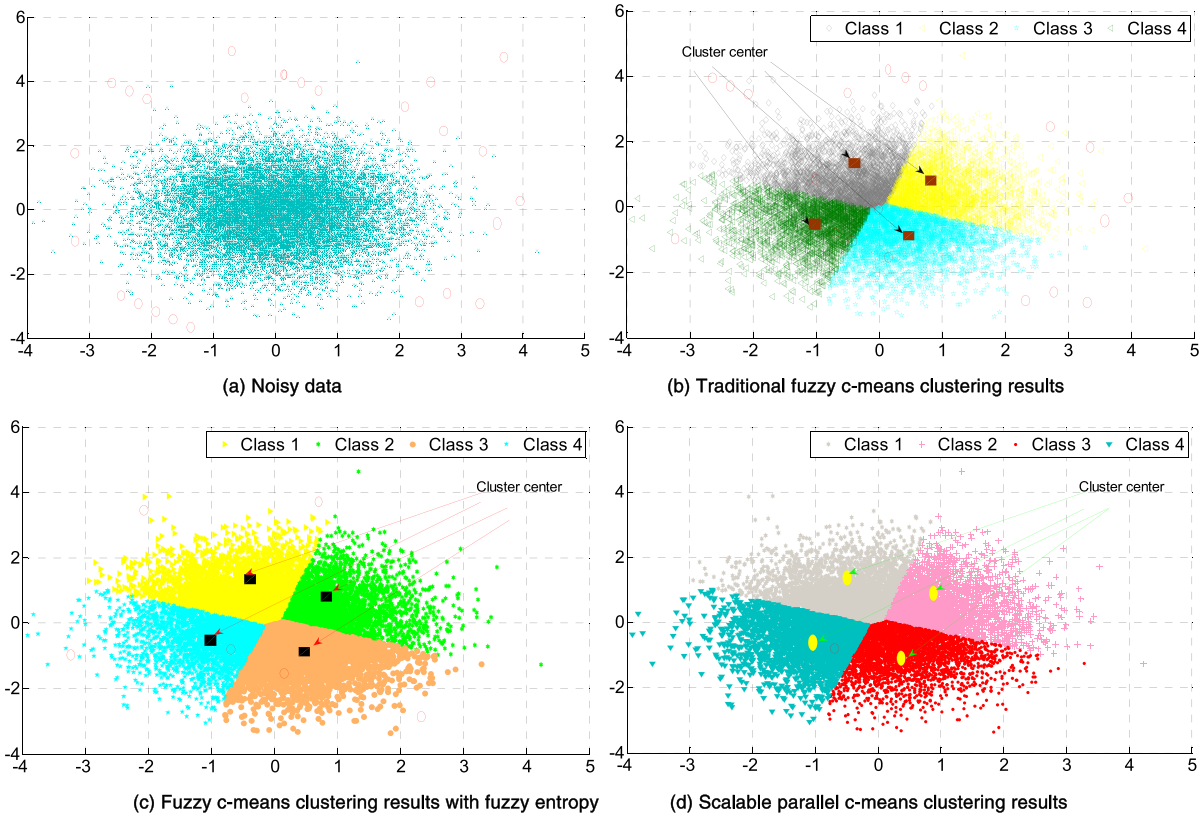


FIGURE 9. Clustering results of three algorithms for noise-reducing two-dimensional data.

as valid data. The addition of fuzzy entropy and scalable parallel fuzzy c-means clustering algorithm can eliminate the effect of some noise data on the valid data. Among them, the scalable parallel fuzzy c-means clustering algorithm has the best anti-noise performance, and the noise data has the least impact on the clustering results. At the same time, the convergence of the algorithm is fast due to the integration of adversarial suppression methods.

Table 5 lists the performance index values of the three algorithms. The scalable parallel clustering function has the smallest value, and its performance is better than the fuzzy c-means algorithm with fuzzy entropy. The fuzzy c-means algorithm based on scalable parallel constraints takes into account the differences between different classes, that is, it maximizes the dissimilarity between different classes, and has the ability to assign low membership values to noisy data points, so it has good anti-noise performance.

In order to further verify that the anti-noise performance of the scalable parallel fuzzy c-means algorithm is better than the traditional fuzzy c-means algorithm, and better than the fuzzy c-means algorithm with fuzzy entropy, the artificial IRIS data set is added artificially. The experimental results are shown in Figure 10.

From the picture above, we can get:

(1) Fuzzy entropy and scalable parallel fuzzy c-means algorithm have better anti-noise performance, because after the introduction of information entropy, the iterative process of the algorithm changes from the original uniform contraction to the uneven contraction. The shrinking direction shrinks, making the final clustering result more consistent with the actual distribution;

(2) The error rate of the fuzzy c-means algorithm with information entropy is low, because this algorithm not only considers the information of the data set, but also the influence of membership, and also introduces adjustments in the fuzzy entropy-based clustering algorithm. Factor w, based on the scalable parallel clustering algorithm, makes the membership calculation formula have the characteristics of Gaussian distribution. The probability of data points belonging to the cluster center is relatively small, thereby effectively suppressing the impact of noise data on the cluster center;

TABLE 5. Performance results of the three algorithms.

Clustering algorithm	Traditional FCM	FCM with fuzzy entropy	Scalable parallel FCM
Cluster validity function value	37.20	21.02	11.01
Accuracy (%)	68.21	82.30	82.21
Precision (%)	69.32	79.01	80.02
Sensitivity (%)	99.10	99.12	99.13
Specificity (%)	1.45	87.03	89.06

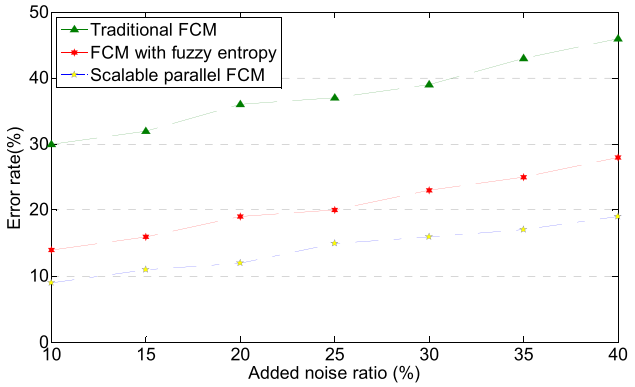


FIGURE 10. Curve of noise ratio and error rate.

(3) The scalable parallel fuzzy c-means clustering algorithm has the lowest error rate and the best anti-noise performance. This is because the scalable parallel fuzzy c-means clustering algorithm considers the same data object and different clusters. The influence of other classes on this class is also considered, making the clustering results more accurate.

V. CONCLUSION

As an important part of data mining, cluster analysis has been widely used in various fields. Although various clustering algorithms have been proposed, different algorithms have their own characteristics and are used in different environments and fields. This paper proposes a scalable parallel fuzzy c-means clustering algorithm, combined with the Spark programming model. It improves the fuzzy c-means algorithm based on special initialization methods. While ensuring the effective clustering ability of the algorithm, the fuzzy c-means can be applied to distributed scenarios in a parallel and highly scalable manner, so that the algorithm can effectively perform cluster analysis of large-scale data after parallelized design and initialization and improvement work. For the improvement of fuzzy c-means random initialization, the improvement strategy of k-means is used to improve the time performance and accuracy of the overall algorithm while obtaining a better initial cluster prototype. The improved analysis methods are integrated, and the combined improved algorithms are used for simple data sets and complex two-dimensional data sets, respectively. The clustering effectiveness function, accuracy, precision, sensitivity, and specificity were calculated to evaluate the clustering results, and a good clustering effect was obtained. However, the methods proposed in this paper are all researched and operated on numerical data. How to effectively apply the existing clustering methods to non-numerical attributes is a problem that needs to be studied in the next step.

REFERENCES

[1] N. Chen, T. Qiu, X. Zhou, K. Li, and M. Atiquzzaman, "An intelligent robust networking mechanism for the Internet of Things," *IEEE Commun. Mag.*, vol. 57, no. 11, pp. 91–95, Nov. 2019.
[2] M. Shengdong, X. Zhengxian, and T. Yixiang, "Intelligent traffic control system based on cloud computing and big data mining," *IEEE Trans Ind. Informat.*, vol. 15, no. 12, pp. 6583–6592, Dec. 2019.
[3] A. G. Polyakova, M. P. Loginov, and E. V. Strelnikov, "Managerial decision support algorithm based on network analysis and big data," *Int. J. Civil Eng. Technol.*, vol. 10, no. 2, pp. 291–300, 2019.
[4] H. A. Nsour, M. Alweshah, A. I. Hammouri, H. A. Ofeishat, and S. Mirjalili, "A hybrid grey wolf optimiser algorithm for solving time series classification problems," *J. Intell. Syst.*, vol. 29, no. 1, pp. 846–857, Dec. 2019.
[5] S. Ahmed, A. I. E. Seddawy, and M. Nasr, "A proposed framework for detecting and predicting diseases through business intelligence applications," *Int. J. Adv. Netw. Appl.*, vol. 10, no. 4, pp. 3951–3957, 2019.
[6] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang, and Y. Hao, "Ready player one: UAV-clustering-based multi-task offloading for vehicular VR/AR gaming," *IEEE Netw.*, vol. 33, no. 3, pp. 42–48, May 2019.
[7] X. Yuan and M. Elhoseny, "Intelligent data aggregation inspired paradigm and approaches in IoT applications," *J. Intell. Fuzzy Syst.*, vol. 37, no. 1, pp. 3–7, Jul. 2019.
[8] S. Wang, "Smart data mining algorithm for intelligent education," *J. Intell. Fuzzy Syst.*, vol. 37, no. 1, pp. 9–16, Jul. 2019.
[9] S. Narayan and J. Gobal, "Optimal decision tree fuzzy rule-based classifier for heart disease prediction using improved cuckoo search algorithm," *Int. J. Bus. Intell. Data Mining*, vol. 15, no. 4, pp. 408–429, 2019.
[10] P. Zhang, Q. Guo, S. Zhang, and H. H. Wang, "Pattern mining model based on improved neural network and modified genetic algorithm for cloud mobile networks," *Cluster Comput.*, vol. 22, no. 4, pp. 9651–9660, Jul. 2019.
[11] M. Tajamolian and M. Ghasemzadeh, "Analytical evaluation of an innovative decision-making algorithm for VM live migration," *J. AI Data Mining*, vol. 7, no. 4, pp. 589–596, 2019.
[12] J. Ren, "Cause analysis of haze based on big data of cloud computing," *Ekoloji*, vol. 28, no. 108, pp. 1095–1099, 2019.
[13] S. Tian, K. Tang, P. Yang, A. Jia, and H. Melvin, "Secure cloud computing model for communication network management," *J. Intell. Fuzzy Syst.*, vol. 37, no. 1, pp. 27–34, Jul. 2019.
[14] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 56–62, Mar. 2019.
[15] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, and I. Humar, "A dynamic service migration mechanism in edge cognitive computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–15, Apr. 2019.
[16] X. Yang, "Intelligent construction of English-Chinese bilingual context model based on CBR," *J. Intell. Fuzzy Syst.*, vol. 37, no. 1, pp. 95–101, Jul. 2019.
[17] F. Gürbüz, I. Eski, B. Denizhan, and C. Dağlı, "Prediction of damage parameters of a 3PL company via data mining and neural networks," *J. Intell. Manuf.*, vol. 30, no. 3, pp. 1437–1449, Mar. 2019.
[18] Y. Hu, L. Wu, C. Shi, Y. Wang, and F. Zhu, "Research on optimal decision-making of cloud manufacturing service provider based on grey correlation analysis and TOPSIS," *Int. J. Prod. Res.*, vol. 58, no. 3, pp. 748–757, Feb. 2020.
[19] A. Farouk and D. Zhen, "Big data analysis techniques for intelligent systems," *J. Intell. Fuzzy Syst.*, vol. 37, no. 3, pp. 3067–3071, Oct. 2019.
[20] K. Chung, H. Yoo, D. Choe, and H. Jung, "Blockchain network based topic mining process for cognitive manufacturing," *Wireless Pers. Commun.*, vol. 105, no. 2, pp. 583–597, Mar. 2019.
[21] Y. Lu, X. Hu, and Y. Su, "Framework of industrial networking sensing system based on edge computing and artificial intelligence," *J. Intell. Fuzzy Syst.*, vol. 38, no. 1, pp. 283–291, Jan. 2020.
[22] L. Zou, Q. Liu, S. Ma, and F. Ma, "Eliciting data relations of IOT based on creative computing," *Int. J. Performability Eng.*, vol. 15, no. 2, pp. 559–570, 2019.
[23] X. Li, D. Liu, H. Huang, and J. Wang, "Research on large data intelligent search engine based on multilayer perceptive botnet algorithm," *J. Intell. Fuzzy Syst.*, vol. 37, no. 3, pp. 3425–3434, Oct. 2019.
[24] M. K. Hassan, D. A. I. El, and M. M. Badawy, "EoT-driven hybrid ambient assisted living framework with naïve Bayes–firefly algorithm," *Neural Comput. Appl.*, vol. 31, no. 5, pp. 1275–1300, 2019.
[25] J. S. Malak, H. Zeraati, F. S. Nayeri, R. Safdari, and A. D. Shahraki, "Neonatal intensive care decision support systems using artificial intelligence techniques: A systematic review," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2685–2704, Dec. 2019.
[26] R. O. S. Juan and J. Kim, "Utilization of artificial intelligence techniques for photovoltaic applications," *Current Photovoltaic Res.*, vol. 7, no. 4, pp. 85–96, 2019.

[27] D. Wang, B. Song, D. Chen, and X. Du, "Intelligent cognitive radio in 5G: AI-based hierarchical cognitive cellular networks," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 54–61, Jun. 2019.

[28] S. Wu, J. Liu, and L. Liu, "Modeling method of Internet public information data mining based on probabilistic topic model," *J. Supercomput.*, vol. 75, no. 9, pp. 5882–5897, Sep. 2019.

[29] H.-Y. Lin and S.-Y. Yang, "A smart cloud-based energy data mining agent using big data analysis technology," *Smart Sci.*, vol. 7, no. 3, pp. 175–183, Jul. 2019.

[30] K. Sharmila, C. Shanthi, R. Devi, and T. K. Kannan, "A comprehensive study on novel hybrid approach for decision support system in disease diagnosis," *Indian J. Public Health Res. Develop.*, vol. 10, no. 3, pp. 200–204, 2019.

[31] J. Fan, S. Yu, J. Chu, F. Cheng, H. Fan, L. Wang, H. Wang, and J. Li, "A novel hybrid decision-making model for team building in cloud service environment," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 12, pp. 1134–1153, Dec. 2019.

[32] A. H. Rabie, S. H. Ali, H. A. Ali, and A. I. Saleh, "A fog based load forecasting strategy for smart grids using big electrical data," *Cluster Comput.*, vol. 22, no. 1, pp. 241–270, Mar. 2019.

[33] P. Kaur, R. Kumar, and M. Kumar, "A healthcare monitoring system using random forest and Internet of Things (IoT)," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 19905–19916, Jul. 2019.

[34] P. Sardar, J. D. Abbott, H. D. Aronow, J. F. Granada, J. Giri, and A. Kundu, "Impact of artificial intelligence on interventional cardiology: From decision-making aid to advanced interventional procedure assistance," *JACC, Cardiovascular Interventions*, vol. 12, no. 14, pp. 1293–1303, 2019.

[35] S. S. Jadhav and S. D. Thepade, "Fake news identification and classification using DSSM and improved recurrent neural network classifier," *Appl. Artif. Intell.*, vol. 33, no. 12, pp. 1058–1068, Oct. 2019.

[36] M. Jin, H. Wang, and Q. Zhang, "Association rules redundancy processing algorithm based on hypergraph in data mining," *Cluster Comput.*, vol. 22, no. S4, pp. 8089–8098, Jul. 2019.

[37] L. Cui, "Complex industrial automation data stream mining algorithm based on random Internet of robotic things," *Automatika*, vol. 60, no. 5, pp. 570–579, Nov. 2019.

[38] S. Li, G. Wang, and J. Yang, "Survey on cloud model based similarity measure of uncertain concepts," *CAAI Trans. Intell. Technol.*, vol. 4, no. 4, pp. 223–230, Dec. 2019.

[39] P. Deshpande, S. C. Sharma, S. K. Peddoju, and S. Junaid, "HIDS: A host based intrusion detection system for cloud computing environment," *Int. J. Syst. Assurance Eng. Manage.*, vol. 9, no. 3, pp. 567–576, Jun. 2018.

[40] P. Verma, S. K. Sood, and S. Kalra, "Cloud-centric IoT based student healthcare monitoring framework," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 5, pp. 1293–1309, Oct. 2018.

[41] L. Zhu and W. J. Zheng, "Informatics, data science, and artificial intelligence," *JAMA*, vol. 320, no. 11, pp. 1103–1104, Sep. 2018.

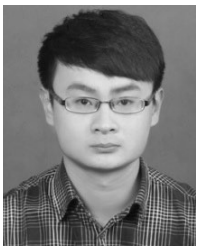
[42] D. D. Miller and E. W. Brown, "Artificial intelligence in medical practice: The question to the answer?" *Amer. J. Med.*, vol. 131, no. 2, pp. 129–133, Feb. 2018.

[43] D. Wang, D. Chen, B. Song, N. Guizani, X. Yu, and X. Du, "From IoT to 5G I-IoT: The next generation IoT-based intelligent algorithms and 5G technologies," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 114–120, Oct. 2018.

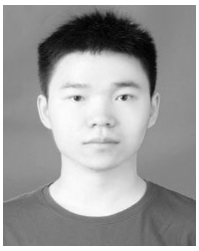
[44] S. Goswami, S. Chakraborty, S. Ghosh, A. Chakrabarti, and B. Chakraborty, "A review on application of data mining techniques to combat natural disasters," *Ain Shams Eng. J.*, vol. 9, no. 3, pp. 365–378, Sep. 2018.

[45] D. Li, L. Deng, Z. Cai, B. Franks, and X. Yao, "Intelligent transportation system in Macao based on deep self-coding learning," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3253–3260, Jul. 2018.

[46] M. Pushpalatha and S. Poornima, "A survey of predictive analytics using big data with data mining," *Int. J. Bioinf. Res. Appl.*, vol. 14, no. 3, pp. 269–282, 2018.



YUAN HUANG was born in Hebei, China, in 1987. He received the bachelor's and master's degrees from the Hebei University of Engineering, in 2010 and 2013, respectively, and the Ph.D. degree from Yanshan University, in 2017. Since 2017, he has been a Teacher with the School of Information and Electrical Engineering, Hebei University of Engineering. He has published 11 articles. His research interests include data mining and machine learning.



ZHE CHENG was born in Hebei, China, in 1995. He received the bachelor's degree from the Hebei Normal University of Science and Technology, in 2017. He is currently pursuing the master's degree with the Hebei University of Engineering. His research interests include machine learning and natural language processing.

QIANYU ZHOU was born in Hebei, China, in 1987. She received the bachelor's and master's degrees from the Hebei University of Engineering, in 2010 and 2013, respectively, and the Ph.D. degree from the China University of Mining and Technology, Beijing, in 2016. Since 2016, she has been a Teacher with the School of Earth Science and Engineering, Hebei University of Engineering. She has published four articles. Her research interests include data analysis and environmental geology.

YUXING XIANG was born in Chongqing, China, in 1997. He received the bachelor's degree from Chongqing Technology and Business University, in 2019. He is currently pursuing the degree with the School of Information and Electrical Engineering, Hebei University of Engineering. His research interests include data mining and machine learning.

RUIXIAO ZHAO was born in Henan, China, in 1996. She received the bachelor's degree from the Xinxiang University of Science and Technology, in 2018. She is currently pursuing the degree wit the School of Information and Electrical Engineering, Hebei University of Engineering. Her research interests include data mining and machine learning.