
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 5, pp. 3614–3625, DOI 10.12694/scpe.v25i5.3186

SIMULATION OF SEGMENTED CLUSTERING OF CLOUD STORAGE DATA BASED ON NEURAL NETWORK MODELS AND PYTHON

GUOQING XIA* AND HUAZHEN CHEN†

Abstract. In order to improve the operational efficiency of traditional cloud storage data segmentation clustering methods, the author proposes a machine learning based cloud storage data segmentation clustering method. Reasonably extract multiple small datasets from the cloud storage database, which contain all natural clusters in the cloud storage database. Construct a similarity matrix based on the definition of similarity. Using nonlinear kernel principal component algorithm to measure the similarity of data in the similarity matrix, data with the same features are grouped together through similarity measurement, and a mixed Gaussian distribution probability density model is used to calculate the posterior probability of different categories of data, implement segmented clustering of cloud storage data by comparing probability sizes. The experimental results show that the proposed method can shorten the clustering running time, reduce the clustering variation to 29%, and effectively improve the smoothness of the clustering results.

Key words: Neural network models, Cloud storage, Segmented clustering of data, A mixed Gaussian distribution probability density model

1. Introduction. Neural networks are widely used in nonlinear system identification and control due to their strong learning ability. However, practical objects are full of uncertain factors, and many problems cannot be described with accurate mathematical models. Fuzzy systems utilize the knowledge and experience of experts to solve mathematical problems through natural language. Since the proposal of Adaptive Network Based Fuzzy Reference System (ANFIS), the research and application of this theory have made significant progress [1]. For a long time, fuzzy design networks were based on traditional Type 1 fuzzy logic systems. With the development of fuzzy set theory and the shortcomings of Type 1 fuzzy logic systems in describing the uncertainty of the objective world, the theory and application of Type 2 fuzzy logic systems have become a research hotspot in fuzzy theory in recent years. With the deepening of research on the identification and control of type 2 fuzzy logic systems in nonlinear systems, research on the identification and control of type 2 fuzzy neural networks in nonlinear systems is gradually increasing. At present, the structure of type-2 fuzzy neural networks mainly includes fixed network structure, self adjusting type-2 fuzzy neural network, self evolving type-2 fuzzy neural network, self-organizing interval type-2 fuzzy neural network, etc. Once the structure of interval type-2 fuzzy neural networks is determined, the next step is to learn the network parameters. Currently, the most commonly used parameter learning algorithm is the backpropagation (BP) algorithm. However, the BP algorithm is sensitive to initial values, and inappropriate initial values can cause the algorithm to diverge or converge to non optimal solutions. The structure of interval type-2 fuzzy logic systems is similar to that of traditional fuzzy logic systems, but it requires a key step, which is order reduction. The order reduction process first reduces the type-2 fuzzy set to a type-1 fuzzy set, and then obtains the precise output of the final interval type-2 fuzzy system using the conventional method of defuzzification of type-1 fuzzy sets. Currently, most interval type-2 fuzzy neural network systems use the Karnik Mendel (KM) reduction algorithm, which is an iterative optimization algorithm, firstly, it is necessary to sort the discrete points by size, so that the corresponding membership degree needs to be modified accordingly. The order can be reduced to obtain two switching points, and each switching point may not be the same. Therefore, when using the BP algorithm to learn parameters, the process is relatively cumbersome and there is no unified learning formula.

*School of Information Engineering, Guangdong Polytechnic, Foshan, Guangdong, 528041, China (Corresponding author, 13926104089@163.com)
†Department of Electronics, Software Engineering Institute of Guangzhou, Guangzhou, Guangdong, 510990, China

Clustering is the process of distinguishing and classifying things according to certain requirements and rules. In this process, there is no guidance from teachers or any prior cognitive information about classification, but only the similarity between things is used as the standard for their classification. Therefore, it belongs to the research content of unsupervised learning. Cluster analysis is the use of mathematical methods to process and study things that need to be classified [2-3]. Birds of a feather flock together. Clustering method is the process of grouping a collection of physical or abstract objects into multiple classes composed of similar objects, and clustering is an ancient problem. Since the emergence of human society, with the continuous development of human society, research on clustering has also been deepening. The continuous exploration of the world by humans requires distinguishing things that belong to different categories and recognizing the similarities of things in the same category. Multivariate statistical analysis, as a branch developed from classical statistics, is also an important branch of mathematical statistics, and cluster analysis belongs to multivariate statistical analysis. As one of the important research directions in statistics, cluster analysis has a profound theoretical foundation and has formed a systematic methodological system. In the field of pattern recognition, cluster analysis is also an important research direction for unsupervised pattern recognition.

Unlike classification, clustering does not rely on pre-defined classes and signed training practices, so clustering analysis is observational learning rather than example based learning. Through cluster analysis, a sample set without any prior knowledge is divided into several subsets based on specific classification rules. The samples within these subsets maintain high similarity, while the samples between subsets maintain low similarity as much as possible. In other words, samples in the same cluster should be as close as possible, while the distance between different cluster centers should be as far as possible. In many applications, data objects in certain classes can be treated as a whole. There are many clustering methods, and their principle is to divide the sample set that needs to be classified into several different categories based on similarity to represent the different characteristics of the system.

The minimum overlap between categories should be used to avoid repetition, which means that each category should contain as many similar samples as possible and have significant differences from each other. The purpose of clustering algorithms is to find several least similar sample centers that contain a set of similar samples, in order to maximize the representation of different features of the system. At the same time, each cluster center should contain a sufficient number of samples to ensure that it uses as few cluster centers as possible to represent the system. Clustering is a technique that studies the logical or physical interrelationships between data. Its analysis results not only reveal the inherent connections and differences between data, but also provide important basis for further data analysis and knowledge discovery. The purpose of clustering is to discover the essential clustering properties between samples, and it is an important component of data mining techniques. Data clustering is flourishing, and contributing fields include data mining, statistics, machine learning, spatial database technology, biology, and marketing. Nowadays, data clustering analysis has become a very active research topic [4]. Traditional clustering analysis is a hard division that strictly divides the objects to be analyzed into certain categories, with the property of either this or that. Therefore, the boundaries of this type of classification are clear. However, in reality, most objects do not have strict attributes, and they have intermediary properties in terms of form and category, with the nature of "this is also that". For example, people are classified according to their height as "tall people", "short people", and "not tall but not short people". However, as tall as possible, and as short as possible, this classification discrimination is a problem that classical classification cannot solve, so it is suitable for soft partitioning. The proposal of fuzzy set theory provides a powerful analytical tool for this soft partitioning, and people have begun to use fuzzy methods to handle clustering problems, namely fuzzy clustering analysis. Fuzzy clustering analysis extends the values of membership relationships from binary logic of 0,1 to the interval of [0,1], thereby more reasonably representing the mediating nature between things. Due to the uncertainty level of the sample belonging to each category obtained by fuzzy clustering, which expresses the "this is also that" property of the sample belonging to different categories, that is, the fuzziness of the sample's membership relationship to different categories, the description and expression of the real world are more reasonable, and have made significant progress in the theory of clustering analysis. The implementation process of data clustering is shown in Figure 1.1.

Fuzzy clustering theory has been widely used in the real world, promoting the improvement of social productivity. With the continuous development of practice, fuzzy clustering theory is also constantly improving

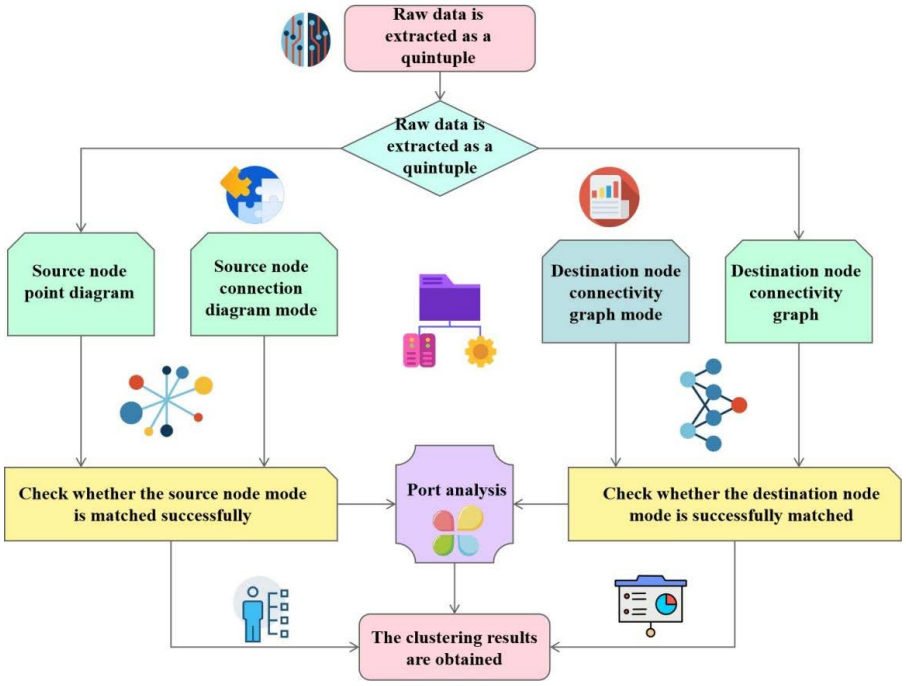

Fig. 1.1: Implementation process of data clustering

and enriching in practice. With the development of practice and the improvement of theory, fuzzy clustering has been widely applied in many fields, and has achieved satisfactory results and objective benefits. Its application scope involves many fields such as channel equalization in communication systems, codebook design in vectorized coding, time series prediction, neural network training, nonlinear system identification, parameter estimation, medical diagnosis, weather forecasting, food classification, water quality analysis, etc, the author mainly studies its application in nonlinear system identification. In nonlinear system identification, fuzzy clustering algorithms can be used for feature space partitioning and fuzzy rule extraction to construct fuzzy classifiers based on fuzzy if then rules.

The amount of data on the internet is showing an explosive growth trend with the rapid development of computers, leading to increased data storage costs, reduced reliability of data storage, and difficulties in managing large amounts of data, which have long plagued users. Users hope to obtain useful information from complex and diverse data, therefore, cloud storage data segmentation and clustering technology has emerged. However, in traditional cloud storage data segmentation and clustering methods, the use of Ethernet and TCP/IP network communication protocols improves and simplifies network protocols to reduce clustering delays, however, neglecting the diversity of data types can lead to problems such as long clustering time and large errors. In this context, researching accurate and efficient cloud storage data segmentation clustering methods has become a widely concerned focus in the current data clustering field, receiving widespread attention from industry insiders. At the same time, many good methods have also emerged. In response to the above issues, the author proposes a machine learning based segmented clustering method for cloud storage data. The experimental results show that the proposed method can shorten the running time of data segmentation clustering and improve the smoothness of clustering results.

2. Methods.

2.1. Overview of Spatial Data Cloud Storage. Space cloud storage mainly refers to the distributed storage and read/write of spatial data based on cloud computing technology. Currently, most of its research focuses on raster data storage and management, while there is relatively little research in the field of vector

data cloud storage. At the same time, research on spatial database systems and spatial databases as services (SDBaaS) in cloud environments based on cloud storage of spatial data is also in its early stages. We will elaborate on the theoretical research of spatial data cloud storage from two aspects: Raster and vector [5].

(1) Overview of Cloud Computing. Cloud computing, as a new type of distributed parallel computing model, can integrate computing power, storage space, and network resources from different regions, allowing any terminal (PC, Web, iOS, Android) user to access the cloud platform at any time and effectively use any resources on the platform. It fully utilizes the good scalability, powerful computing power, and cross regional information resource sharing function of distributed computing systems, by establishing a virtual and single system image for applications and data, users can easily and easily access all shared resources on the entire cloud platform, which is an excellent strategy to solve the deep sharing of current geospatial data and data processing services. Large cloud computing providers (Amazon, Google, Microsoft) encapsulate their infrastructure, platforms, frameworks, software, applications, and even data into network services, providing users and developers with standardized interfaces and on-demand billing models. Google's cloud computing platform provides basic technical support for applications such as Google App Engine, Google Map, and Google Trend. Google Cloud Platform was founded in 1996, when the founder of Google began attempting to combine multiple inexpensive PCs into a powerful computing platform to index billions of web pages; Nowadays, Google has a cloud computing platform consisting of over one million servers, providing different levels of services including IaaS, PaaS, and more.

Finally, Amazon created an elastic computing cloud EC2 based on this platform, providing users with on-demand online rental services for computing resources with surplus hardware resources. Microsoft also released its cloud computing platform product Azure in November 2009, which is a cloud application platform built on top of Microsoft's data center. It can manage and hook cloud application systems and provide a set of tools to facilitate developers to develop and debug cloud applications locally. So far, few have conducted research and development on cloud computing platforms as an integrated platform for spatial cloud storage and third-party service release and deployment. In order to achieve the goals of cloud computing and its high performance, low cost, and strong universality, cloud computing has developed a series of key technologies that support its functions such as data storage, data management, parallel computing, and concurrency control, including computer system virtualization technology, massive distributed storage technology, parallel programming mode, large-scale data management technology, distributed resource management technology, etc.

Virtualization technology is the underlying key to building an IaaS cloud platform. It quickly integrates and decomposes physical resources in a specific way, can dynamically organize multiple heterogeneous hardware, and achieve isolation between underlying physical hardware and other virtual machines on specific computers, achieving loose coupling between computing cluster hardware and software, and relieving severe dependencies between these architectures, thus meeting the scalability and scalability requirements of cloud computing for clusters. Its basic strategy is to build independent virtual machines, flexibly respond to the sudden increase or decrease in resource demands of cloud users, and improve the efficiency of platform resource utilization. Simulation is a process of continuously extracting dependencies, and its guest virtual machines will be managed and operated by virtual machine administrators (Hypervisors), providing personalized and diverse computing environments.

(2) Spatial database. Spatial databases optimize the storage and querying of spatial objects, including points, lines, and surfaces, based on traditional relational databases. A typical relational database typically only includes various numbers and characters, while a spatial database adds spatial data types and adds database functionality to handle these types of data [6-7]. OGC (OpenGeospatial Consortium) has developed the Simple Features Specification for geospatial data and corresponding standards to standardize the functionality of spatial databases in data processing. Due to the fact that the indexes of relational databases are not optimized for spatial queries, spatial databases need to develop their own spatial indexes to improve the efficiency of spatial database operations. Universal spatial databases typically support spatial operations such as spatial metrics, spatial functional functions, spatial predicates, constructors, and more. However, currently many NoSQL databases such as MongoDB and CouchDB, although they support spatial data types, do not fully support the aforementioned spatial operation functions.

Table 2.1: Comparison of NoSQL Database Types

Type	Product	Characteristic
Key value	Redis	Simple and easy to use, with direct values
Column Family	Bigtable, HBase	Flexible mode, allowing for arbitrary addition or deletion of column families
Document	MongoDB, CouchDB	Arbitrary pattern query, nested documents
Chart	Neo4, GraphLab	Suitable for complex data structures

(3) NoSQL database. Popular Web 2.0 applications typically have hundreds of millions of users, and these applications generate massive amounts of user data in a short period of time (ranging from a few months to a year), causing server loads to grow exponentially, resulting in extremely high demands for data storage scalability. In order to meet the requirements of data storage and management for such applications, web data must be partitioned and stored on thousands of processors. These new storage systems aim to provide distributed data storage, good horizontal scalability, and high-performance read and write operations. At the same time, traditional relational databases severely lack horizontal scalability, which limits the performance of single machines to handle data of such scale. NoSQL is a thriving non relational database technology in this context, which is usually classified into various types such as key values, column families, documents, and graphs based on different data models, as shown in Table 2.1.

The NoSQL database adopts a looser consistency model to provide a simple, lightweight, and efficient storage and retrieval mechanism to support better scalability and availability than traditional relational databases. The key technology for NoSQL to handle massive data is that it pre-set partitioning functions for the data NoSQL typically automatically divides data into relatively small table units (MB level), stored on multiple different physical servers, and user programs access the servers where these table blocks are located through indexes or metadata. All updates generated by the user program will be aggregated to the main server, and then the updates to the data will be propagated to each replica server storing the table block through synchronization services.

Due to the high cost of the two-stage commit protocol used in traditional distributed databases, it may also fail during commit, leading to cluster congestion. NoSQL databases follow the CAP theorem and mostly provide a BASE concurrent transaction model that is looser than ACID, achieving basic availability, flexible state, and final consistency of the database in specific application areas. Therefore, NoSQL basically includes the following characteristics:

The ability to expand horizontally. NoSQL database can dynamically expand to multiple servers as business and data grow; Copy distribution capability. NoSQL database easily, effectively, and accurately replicates and propagates data to child node servers; Simple query interface and protocol. Compared to the complete SQL syntax of relational databases, NoSQL databases only provide lower level data query interfaces; Data storage can effectively utilize distributed indexes and memory; Ability to dynamically add new attributes to data records.

(4) Figure Database. In many fields including semantic analysis, geographic information systems, image processing, social networks, and biochemical informatics, graphs are a natural and suitable data model for domain data features. Semantic web information can be viewed as a collection of graphs that represent entities and explicit relationships; The transportation network in GIS is a typical graph; In chemical informatics, graphs can be used to represent the atoms and chemical bonds of compounds. The data in these fields are highly complex and large-scale, and existing data models, query languages, and database systems are difficult to support the modeling, querying, and management of these data. A graph database is a database system that represents and stores data in a graph structure with vertices, edges, and attributes, providing adjacency operations without indexing. The existing graph databases mainly use PropertyGraph as the data model, and some graph databases support HyperGraphModel. A property graph is a multigraph in which both vertices and edges have attributes stored in key value pairs, and all edges of the property graph are directed and asymmetric, as shown in Figure 2.1. A hypergraph is a superset of a graph, whose edges can be associated with any number

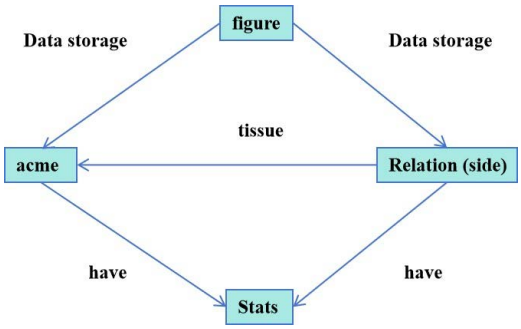

Fig. 2.1: Attribute Graph Data Model

of vertices.

Due to the fact that graph databases typically use attribute graphs as data models, native graph based data management systems have natural advantages in data management in the aforementioned fields. They can be used to store highly complex and large-scale non relational data in many fields, including semantic web, GIS, social network, bioinformatics, chemical informatics, and more. The author will map vector data that follows the OGC Simple Feature Specification (SFS) to an attribute graph model, where the objects and relationships of the vector data correspond to the vertices and edges of the graph data, respectively. Due to the fact that both the item points and edges in the attribute graph have attributes, there will be some storage redundancy. The spanning tree of the attribute graph model itself can serve as a natural database index, but it is not entirely applicable to spatial data. Spatial indexing can effectively improve the efficiency of spatial data queries, and in-depth research is needed on the theory of spatial indexing in order to select the appropriate spatial index.

(5) Spatial index. Spatial index is a spatial data structure that can be used in spatial databases to optimize spatial query indexing techniques by preserving the position, shape, and positional relationships of spatial entities. The components of spatial indexing include pointer objects pointing to spatial entities, bounding polygons of spatial entities, and object identifiers of spatial entities, which have very similar implementation principles. The spatial index will divide the spatial query area based on the principle of spatial segmentation by dividing the query dimension of the target. The spatial dimension will identify these partitioned subspaces with a tree structure and ensure index uniqueness through hash identification. The principle of spatial segmentation mainly includes two methods: Rule-based segmentation and object based segmentation. The former is based on the idea of computational geometry, while the latter is based on the independence of spatial objects. Rule based segmentation method, viewing a geographic plane as a multi-dimensional geometric body, segmentation is achieved through regular rectangles or irregular concave polygons. The integrity of a single spatial feature entity is ignored, and it will be divided into multiple parts of different units. However, this method of segmentation does not disrupt the logical consistency of spatial entities, but only changes the pointer object of the spatial index. The object segmentation method first determines the minimum bounding rectangle of the spatial entity, and then separates it according to the degree of entity independence, while ensuring the spatial and attribute consistency of the entity. However, this method has a high time complexity and produces a large number of spatial entities, resulting in significant storage and computational consumption. Therefore, it can be seen that spatial indexing, through preprocessing, can exclude objects unrelated to user query targets and quickly locate spatial entities that meet query syntax requirements.

2.2. Machine learning based segmented clustering method for cloud storage data.

(1) Building a similarity matrix for cloud storage datasets. Reasonably extract multiple small datasets from the cloud storage database, which should contain all natural clusters in the database. Divide the numerical data in the small-scale dataset into other types of data, extract them separately, and obtain independent datasets. Based on the data in each column, construct numerical and symbolic matrices, and combine them to obtain similarity matrices. Various types of data are mixed and stored in the database, and the dataset forms multiple natural clusters in the database. Extracting small-scale datasets from the database and performing segmented

clustering on the selected dataset can effectively reduce the computational error of numerical data clustering algorithms and simplify the algorithm steps. When extracting small-scale datasets, it is important to ensure the validity of the dataset and determine the number of selected natural clusters. Assuming that there are a total of n natural clusters in the selected dataset, in order to effectively simplify the complexity of mixed large-scale databases, a small-scale dataset sampling method is adopted to cluster the selected dataset, and a sample estimation method is designed to obtain the ideal sampling sample. Use S to represent the sampling sample, and let $\xi, \xi \in [0,1]$ be the data extraction ratio of the dataset, due to the presence of n natural clusters in the dataset, the size of the extracted natural clusters is n. T represents the probability of extracting $\xi \times n$ data from the natural clusters in the dataset, and the resulting data samples are represented as:

$$S = \xi \times n + \frac{n}{n_i} \times log\frac{1}{\tau} + \frac{n}{n_i} \sqrt{log(\frac{1}{\tau})^2 + 2\xi \times n \times log\frac{1}{\tau}} \tag{2.1}$$

If the size and number of categories of the natural cluster are small or equal, it indicates that the size of the natural cluster is more than one layer, and the size of the natural cluster meets the requirements for extracting the dataset [8]. Set the extraction ratio of the dataset to ξ in such cases where the natural cluster size and number of categories are small, the extracted dataset size will also be smaller. Adopting equal scale sampling for mixed large-scale dataset A can enhance the convenience of dataset sampling in cloud storage databases and ensure the rationality of sampling. Set the small-scale dataset size as A_i, undergo m sampling, and meet the sampling control conditions as follows:

$$\begin{cases} A_i \cap A_j = \varnothing \\ m_i = m_j \end{cases} \tag{2.2}$$

Because the size of the sampled dataset is relatively small, when clustering the dataset, the aggregation level will quickly complete the clustering, which greatly improves the clustering speed. Moreover, due to the small size, the clustering accuracy is also improved. Further cluster the numerical data extracted from the dataset, strip out other types of data, and in order to simplify the algorithm steps, all other types of data are uniformly recorded as symbolic data. Extract numerical and symbolic data separately, construct two independent datasets, and construct the approximation matrix of the dataset as follows:

$$W_i = \frac{C_i}{\sum_{i=1}^{n} C_i} \tag{2.3}$$

In the formula, W represents an independent dataset.

Calculate the similarity between numerical and symbolic data separately, and construct a matrix of numerical data using a Gaussian function, assuming AA represents a numerical data matrix, and d represents the euclidean distance between data points, λ represents the characteristic parameter of the Gaussian function, then T_{ij} can be expressed as:

$$T_{ij} = exp(-\frac{d}{2\lambda^2}), i, j = 1, 2, ..., n \tag{2.4}$$

Symbolic data attributes can be set to:

$$T'(x_i, x_j) = \begin{cases} 0, x_i \neq x_j \\ 1, x_i = x_j \end{cases}, i, j = 1, 2, ..., n \tag{2.5}$$

The numerical matrix $T_{i,j}$ and symbolic data $T'_{i,j}$ can be represented as follows:

$$T_{i,j} = \begin{bmatrix} 1, 0.331, 0.475, 0.358 \\ 0.331, 1, 0.331, 0.135 \\ 0.475, 0.331, 1, 0.216 \end{bmatrix} \tag{2.6}$$

$$T'_{i,j} = \begin{bmatrix} -1,0,0,1,0,1 \\ 0,1,1,0,1,0 \\ 0,1,1,0,1,0 \\ 1,0,0,1,0,1 \\ 0,1,1,0,1,1 \end{bmatrix} \tag{2.7}$$

The similarity matrix constructed by combining numerical matrix $T_{i,j}$ with symbolic data $T'_{i,j}$ is:

$$T' = T_{i,j} + \sum_{i=1}^{n} P_i \times T'_{i,j} i, j = 1, 2, ..., n \tag{2.8}$$

In the formula, P represents the similarity weight.

(2) Segmented clustering of cloud storage data based on machine learning. Based on the similarity matrix provided in (1), a nonlinear kernel principal component algorithm is used to measure the similarity of data in the similarity matrix. A mixed Gaussian distribution probability density model is used, combined with the comparison of similarity measurement probabilities, to achieve segmented clustering of cloud storage data.

Based on the similarity matrix, a non-linear kernel principal component algorithm is used to obtain the matrix similarity measure, taking into account a set of variables with non-linear correlation in the cloud storage database $X_{i,j}(t), i = 1, 2, ..., N, j = 1, 2, ..., m, t = 1, 2, ..., T$, N represents the data capacity in the cloud storage database, n represents the number of variables, and T represents the length of the time series. Assuming that the N×m sample data in cloud storage is represented as $X(t)(x_1(t), x_2(t), ..., x_n(t))$, a nonlinear mapping function ω is used to project the sample data X_i(t) from the input space R^N to the high-dimensional feature space F^N:

$$\omega : R^N \to F^N \tag{2.9}$$

The sample data projected onto the high-dimensional and high-dimensional feature space F^N is represented by $\omega(x_i(t), i = 1, 2, ..., N)$, and the mapping process needs to satisfy the centralization condition of the feature space as follows:

$$\sum_{i=1}^{N} \omega(x_i(t)) = 0 \tag{2.10}$$

After projection, the covariance matrix of the sample data that meets the centralization condition is set as:

$$C = \frac{1}{n} \sum_{i=1}^{n} \omega(x_i(t)) \omega^T(x_i(t)) \tag{2.11}$$

In the formula, C represents the covariance matrix.

Assuming that the eigenvalues of C satisfy $0 \leqslant \lambda_1 \leqslant \lambda_2 \leqslant ...\lambda_N$, the corresponding feature data is $V(\nu_1, \nu_2, ..., \nu_N)$. Due to the fact that the non-zero feature data V and the projection data $\omega(X(t))$ belong to the high-dimensional feature space F^N, there exists a set of coefficients $\alpha = [\alpha_1, \alpha_2, ..., \alpha_N]^T$ that enable V to be represented by a linear combination of $\omega(X(t))$ as:

$$V = \sum_{i=1}^{N} \alpha_i \omega(x_i(t)) \tag{2.12}$$

Set a kernel matrix K for N×N, the specific process is as follows:

$$K_{i,j} = k(x_i, x_j) = \omega^T(x_i(t)) \omega(x_j(t)) \tag{2.13}$$

The vector form of its sum matrix is:

$$K\alpha = N\lambda\alpha \tag{2.14}$$

In the above equation, K is the kernel function matrix of $N \times N$; N is the data characteristic value of K; $\alpha_1, \alpha_2, ..., \alpha_N$ is the eigenvector corresponding to the eigenvalues of each data. The projection $P(x_j(t))$ of sample data $x_j(t)$ in the direction of feature vector $V_r(r = 1, 2, ..., k)$:

$$P(x_j(t)) = V_r^T \omega(x_i(t)) \tag{2.15}$$

In the formula, the vector (xi (t)) represents the j-th sample data.

After setting the non-linear kernel function $k(x_i, x_j)$, it is possible to perform non-linear projection of data in cloud storage databases based on mapping rules, reduce data dimensions, and extract data feature information according to non-linear related indicators. The weight of the extracted features is:

$$p_i = \frac{\lambda_i}{\sum_{i=1}^{k} \lambda_i} \tag{2.16}$$

In the formula, pi represents the weight of the eigenvalues [9].

Based on similarity measurement, a mixed Gaussian distribution model is used to segment and cluster cloud storage data. The mixed Gaussian distribution model is a weighted linear combination of a finite number of multivariate Gaussian distributions. The probability density function g (x) of the mixed Gaussian distribution is calculated as:

$$g(x) = \sum_{i=1}^{k} \alpha_i f_i(x; \mu_i) \tag{2.17}$$

In the formula, K represents the number of multivariate Gaussian distributions; $f_i(x; \mu_i), \alpha_i$ is the probability density function and weight of the i-th multivariate Gaussian distribution. In the parameter estimation of the model, due to the large number of parameters to be estimated, the moment estimation method is obviously difficult to implement. Therefore, consider maximum likelihood estimation and construct the corresponding likelihood function L;

$$L = \prod_{j=1}^{N} g(x_j; \theta) \tag{2.18}$$

According to the above equation, when there is a large amount of data, maximum likelihood estimation is more difficult.

Assuming the number of clusters k is a known exogenous variable, initialize all parameters of the mixed Gaussian model, assuming $\theta_0 = (\alpha i_0, \mu i_0)$, among them, the mean vectors and covariance matrices of K multivariate Gaussian distributions can be obtained through other statistical algorithms, and the weights are initially set to $\frac{1}{k}$.

In the first step of updating weights, for any sample xj, the probability $\omega_{j1}(k)$ of the k-th class is used, and the $\omega_{j1}(k) \times x_j$ part of its value is treated as generated by the k-th multivariate gaussian model, the k-th multivariate gaussian model produces $\omega_{j1}(k) \times x_j, (j = 1, 2, ..., N)$, with a total of N data points, and the parameters belonging to this multivariate gaussian distribution replace the initial parameters. After the first iteration, the parameters of the k-th single Gaussian model are:

$$N_{k1} = \sum_{j=1}^{N} \omega_{j1}(k) \tag{2.19}$$

After a complete EM calculation, the updated value $\theta_1 = (\alpha_{i1}, \mu_{i1})$ of all cloud storage segment data can be obtained. Using θ_1 as the cloud storage segmentation data for the mixed Gaussian model, a second EM iteration can be performed. Given a sufficiently small threshold, after multiple iterations, when $|ln(L)^{[n-1]} - \ln(L)^{[n]}| < threshold$, the EM loop iteration can be exited. Obtain convergent model parameters.

After achieving global convergence of model parameters, clustering of cloud storage segmented data samples xj can be achieved by comparing the probability value $\omega_j(k), \omega_j(k)$ of any sample xj belonging to different categories.

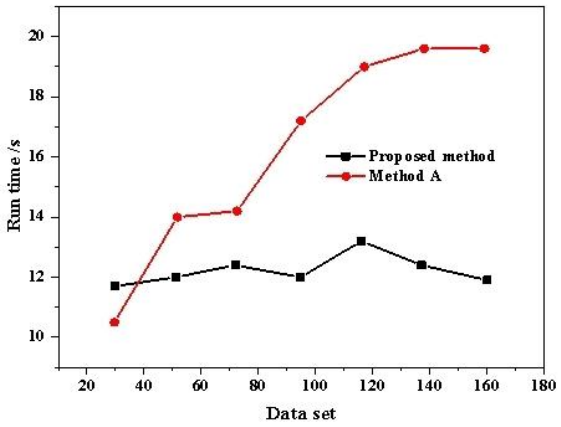

Fig. 3.1: Comparison of runtime (s) of different methods

Table 3.1: Clustering variation of different methods (%)

SJ/s	W7	W8	ST
2	59	66	43
4	63	69	47
6	57	70	39
8	62	73	32
10	60	83	30

3. Results and Analysis. In order to verify the comprehensive effectiveness of the proposed machine learning based cloud storage data segmentation clustering method, a simulation is required, and the simulation environment is: Intel (R) Core (TM) 2i5-3210M, CPU main frequency 2.3GHz, 2GB of memory, operating system Windows 7, software environment Anaconda3 (64 bit), simulation program using Python language. Cluster the proposed method with method A for cloud storage data, and compare the operational efficiency (%) of the two methods. The comparison results are shown in Figure 3.1. (Method A: Segmented clustering method for cloud storage data based on trend function space).

As shown in Figure 3.1, as the number of samples increases, the running time of the two methods also changes. Overall, the proposed method has a shorter running time than method A [10]; Specific analysis shows that when the number of data in the dataset is 30, the clustering time of method A is 11 seconds, which is 1 second shorter than the proposed method. However, when the number of data in the dataset is higher than 30, the running time of method A increases linearly, which is 8 seconds longer than the proposed method. The main reason is that method A needs to calculate the integrated spatial distance between a large number of data points, the proposed method effectively reduces the calculation of integrated spatial distance and shortens the clustering running time by dividing the data in the cloud storage database into multiple small datasets. Compare the clustering changes (%) at different times on the same dataset using the proposed clustering method, A clustering method, and B clustering method, respectively. The comparison results are shown in Table 3.1: In Table 3.1, SJ represents different time points in seconds, represented by (s); JL represents the degree of clustering change, in%; ST represents the method proposed; W7 represents method A; W8 represents method B. (Method A: Segmented clustering method for cloud storage data based on trend function space, Method B: Segmented clustering method for cloud storage data based on principal component analysis).

Compare the experimental data in Table 3.1 in the form of experimental graphs, as shown in Figure 3.2.

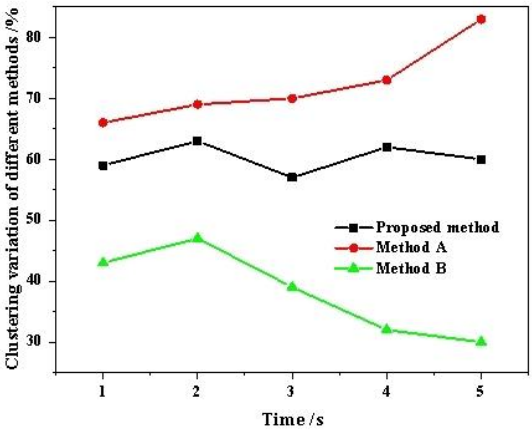

Fig. 3.2: Clustering variation of different methods

From Table 3.1 and Figure 3.2, it can be seen that the clustering degree of the three methods is not the same at different times. At 2 seconds, the clustering degree of the B clustering method is 65%, which is 23% higher than the proposed clustering method and 7% higher than the A clustering method; As the clustering time increases, the clustering degree of A and B clustering methods also increases. At 10 seconds, the clustering degree of A method is 30% higher than that of the proposed method, and the clustering degree of B method is 53% higher than that of the proposed method; Throughout the clustering process, only the proposed method showed a decreasing degree of clustering variation with increasing clustering time, decreasing from 42% to 29%. Overall, the proposed method effectively improved the smoothness of clustering results.

4. Conclusion. In response to the problems of low efficiency and insufficiently smooth clustering results of traditional cloud storage data segmentation clustering methods, the author proposes a machine learning based cloud storage data segmentation clustering method. The experimental results indicate that, the method proposed by the author has significantly improved the clustering variation compared to traditional methods. The proposed method reduces the clustering variation from 42% to 29%, effectively improving the smoothness of the clustering results. There are still many problems in using the proposed cloud storage data segmentation and clustering method in practical applications, and the network environment is becoming increasingly complex, with the types of data becoming more diverse, after the emergence of complex and diverse data, the proposed method cannot effectively cluster all types of data. In the face of a wide variety of data, it is necessary to improve the ability of segmented clustering.

5. Acknowledgement.

Source: University Level Scientific Research Projects;

Name: Research on the Application of Artificial Intelligence Algorithm in UAV Detection System; Number XJKY202209.

REFERENCES

[1] Sabur, A., Chowdhary, A., Huang, D., & Alshamrani, A. (2022). Toward scalable graph-based security analysis for cloud networks. Computer Networks, 206(468), 108795-.

[2] Zhao, H. (2023). Research on the recognition and localization of momordica grosvenori based on binocular vision and a convolutional neural network. 2023 IEEE International Conference on Control, Electronics and Computer Technology 223(ICCECT), 404-408.

[3] Wang, P., Nie, S., Wang, J., Wang, C., Xi, X., & Du, M. (2022). Segmentation of the communication tower and its accessory

equipment based on geometrical shape context from 3d point cloud. International Journal of Digital Earth, 15(1), 1547-1566.
[4] Anna E. Sikorska-Senoner. (2022). Clustering model responses in the frequency space for improved simulation-based flood risk studies: the role of a cluster number. Journal of Flood Risk Management, 15(1), n/a-n/a.
[5] Saurabh, & Dhanaraj, R. K. (2023). Enhance qos with fog computing based on sigmoid nn clustering and entropy-based scheduling. Multimedia Tools and Applications, 83(1), 305-326.
[6] Gao, W., & Zhang, L. (2022). Semantic segmentation of substation site cloud based on seg-pointnet. J. Adv. Comput. Intell. Intell. Informatics, 26(546), 1004-1012.
[7] Zhang, Y., Gao, X., Bai, Y., Wang, M., & Tian, Q. (2022). Multi-condition identification of thermal process data based on mixed constraints semi-supervised clustering. SN Applied Sciences, 4(7), 1-19.
[8] Li, J., Xing, Y., & Zhang, D. (2022). Planning method and principles of the cloud energy storage applied in the power grid based on charging and discharging load model for distributed energy storage devices. Processes, 10(2), 194-.
[9] Chen, G., Bai, B., Mao, Z., & Dai, J. (2022). Real-time road object segmentation using improved light-weight convolutional neural network based on 3d lidar point cloud. International Journal of Ad Hoc and Ubiquitous Computing, 39(3), 113-.
[10] Singh, A., & Kumar, M. (2023). Bayesian fuzzy clustering and deep cnn-based automatic video summarization. Multimedia Tools and Applications, 83(1), 963-1000.

Edited by: Zhigao Zheng
Special issue on: Graph Powered Big Aerospace Data Processing
Received: Jan 16, 2024
Accepted: Mar 5, 2024