Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 8690418, 10 pages
https://doi.org/10.1155/2021/8690418

Research Article

Glowworm Swarm Optimization Algorithm- and K-Prototypes Algorithm-Based Metadata Tree Clustering

Yaping Li [1,2]

[1] *Postdoctoral Workstation, Party School of Anhui Provincial Committee of the Communist Party of China (Anhui Academy of Governance), Hefei 230022, Anhui, China*
[2] *School of Management, Hefei University of Technology, Hefei 230009, Anhui, China*

Correspondence should be addressed to Yaping Li; ypliflу@126.com

Received 18 August 2020; Revised 18 November 2020; Accepted 27 January 2021; Published 10 February 2021

Academic Editor: Mukhtaj khan; mukhtaj.khan@awkum.edu.pk

The main objective of this paper is to present a new clustering algorithm for metadata trees based on K-prototypes algorithm, GSO (glowworm swarm optimization) algorithm, and maximal frequent path (MFP). Metadata tree clustering includes computing the feature vector of the metadata tree and the feature vector clustering. Therefore, traditional data clustering methods are not suitable directly for metadata trees. As the main method to calculate eigenvectors, the MFP method also faces the difficulties of high computational complexity and loss of key information. Generally, the K-prototypes algorithm is suitable for clustering of mixed-attribute data such as feature vectors, but the K-prototypes algorithm is sensitive to the initial clustering center. Compared with other swarm intelligence algorithms, the GSO algorithm has more efficient global search advantages, which are suitable for solving multimodal problems and also useful to optimize the K-prototypes algorithm. To address the clustering of metadata tree structures in terms of clustering accuracy and high data dimension, this paper combines the GSO algorithm, K-prototypes algorithm, and MFP together to study and design a new metadata structure clustering method. Firstly, MFP is used to describe metadata tree features, and the key parameter of categorical data is introduced into the feature vector of MFP to improve the accuracy of the feature vector to describe the metadata tree; secondly, GSO is combined with K-prototypes to design GSOKP for clustering the feature vector that contains numeric data and categorical data so as to improve the clustering accuracy; finally, tests are conducted with a set of metadata trees. The experimental results show that the designed metadata tree clustering method GSOKP-FP has certain advantages in respect to clustering accuracy and time complexity.

1. Introduction

As an important tool for data management, metadata play a very important role in data integration, sharing, retrieval, and the construction of data warehouses. The research and application of metadata is extensive, in which metadata-based data integration and data clustering have been widely and deeply studied, while the clustering study of the metadata itself is rare [1]. FIHC (Frequent Itemset based Hierarchical Clustering) is an agglomerative hierarchical clustering algorithm first proposed by Benjamin [2]. First, frequent word sets are mined, and then the texts are used with the same frequent word sets as an initial cluster to classify the texts. From this, Feng and Chen designed a metadata clustering method based on MFP [3]. This method measures the similarity between metadata trees by the characteristics of MFP. However, the MFP method reduces the computational complexity at the cost of losing part of metadata information, which will have a negative impact on the subsequent feature vector clustering. In addition, the test process shows that the above method is not suitable for large-scale metadata tree clustering.

In order to solve the clustering problem of mixed-attribute data more effectively, K-prototypes algorithm [4], EKP algorithm, and SBAC algorithm have been proposed [5]. Compared with other algorithms, the K-prototypes algorithm has the advantages of simple and efficient, but the K-prototypes algorithm is more sensitive to the initial clustering center, which also has a negative impact on the clustering accuracy. Since metadata usually appear in the form of JSON, XML, etc. in web engineering, metadata clustering research is mainly embodied in clustering research of XML documents [6]. Typical methods for describing the structure of a document include both trees and vectors. Therefore, research to document clustering is usually methodologically translated into trees and vectors to carry out clustering analysis. Feng et al. put forward a new clustering based on K-medoids clustering and the genetic algorithm to enhance the accuracy of the XML document clustering in 2015 [7]. Wang et al. propose a document clustering based on the CFP algorithm (Clustering with Feature Order Preference) in 2016 [8]. Based on the tree structure, Costa and Ortale projected XML documents onto the path from root nodes to leaf nodes and proposed a new clustering method combining XML features with mixing lengths [9]. Since then, with the increasing application of swarm intelligence algorithms, swarm intelligence optimization algorithms have been introduced into the application of clustering algorithms [10]. A new clustering algorithm called K-MWO has been proposed by Kang et al. in 2016. The K-MWO algorithm makes full use of global optimization ability of MWO and local search ability of K-means [11]. A novel clusterability assessment method called Density-based Cluster ability Measure (DBCM) has been proposed by Jokinen in 2019 [12]. The above methods design the corresponding clustering method for the clustering problem of mixed-attribute data and further study the clustering problem of XML documents with structural features. However, due to the structural characteristics of the metadata tree, these clustering algorithms which are mainly suitable for metadata records cannot be directly applied to the clustering of metadata trees. In addition, clustering accuracy of these clustering algorithms, such as the typical K-prototypes algorithm, is not satisfactory.

The main objective of this paper is to present a new clustering algorithm for metadata trees. At the same time, improving the accuracy of the K-prototypes clustering algorithm and the loss of critical information in the MFP method is also important contents of this research. The GSO algorithm, brought up by Krishnanand and Ghose, is a new swarm intelligence optimization algorithm [13]. Compared with other swarm intelligence algorithms, the GSO algorithm has more efficient global search advantages and simpler algorithm flow, which is suitable for solving multimodal problems [14]. The clustering accuracy of the K-prototypes algorithm can be improved by the GSO algorithm [15]. In this document, the author combines the improved GSO algorithm and K-prototypes algorithm with the idea of MFP [16] to design a new metadata tree clustering schemes to realize the clustering of metadata tree sets. This paper mainly contains four aspects: first, mine MFPs in the metadata tree set and calculate the feature vectors of these MFPs using the "term frequency and inverted document frequency (TF/IDF)" method [17]; second, combine GSO with K-prototypes to design a metadata tree-oriented clustering method GSOKP; third, employ MFP's feature vector to describe the features of the metadata tree and perform clustering on the feature vector via GSOKP to achieve the clustering of the metadata tree set; finally, establish metadata tree sets as experimental data to test the validity of text algorithms. To sum up, the contributions of this paper mainly include the design of an improved K-prototypes algorithm by the GSO algorithm, an optimization strategy of feature vector calculation of metadata tree, and finally propose a new clustering method for metadata tree sets.

2. Clustering of Metadata Trees

2.1. Description of the Metadata Tree. Metadata are special data that describe data. Metadata clustering includes the judgment about the similarity of metadata structures and metadata records. Metadata structures are often described using metadata trees. So, clustering for metadata structure similarity can be described as clustering for metadata trees.

Definition 1. Metadata tree.

To make $\mathrm{MD} = \{md_1, md_2, \ldots, md_n\}$, meaning that metadata MD are composed of n metadata elements, wherein $md_i = \{d_t, d_{t+1}, \ldots, d_{t+p}\}$ meaning that metadata elements md_i are composed of p subattributes. Each metadata element and subattribute correspond to a node of the metadata tree, and the tree structure shown in Figure 1 is an example of a metadata tree *MD*.

Definition 2. Metadata path.

In a metadata tree, a set of metadata element sequences where the nodes do not recur from the root node to the leaf node is called a metadata path. In Figure 1, for example, $\langle \mathrm{MD}, md_1, d_1 \rangle$ is a metadata path.

Definition 3. Similarity between metadata trees.

The similarity of metadata trees is primarily measured by the similarity of path sets. Assume that there is a metadata example as shown in Table 1.

The metadata example is described as metadata tree *Mete*, as shown in Figure 2.

Definition 4. Frequent item.

Frequent item is a metadata element that occurs frequently in a set of metadata trees. Its frequency is measured by the occurrence rate ρ of the elements in the metadata tree. That is, the rate of occurrences of the metadata element md_i or d_j in each metadata tree of metadata tree set $T = \{\mathrm{MD}_1, \mathrm{MD}_2, \ldots, \mathrm{MD}_s\}$. Set the frequency threshold to be θ. If $\rho \geq \theta$, md_i or d_j is a frequent term.

Based on the structure of the data example given in Table 1, the metadata tree *SURF* shown in Figure 3 can be constructed.

In the metadata set of *Mete* and *SURF*, if $\theta = 100\%$, the elements contained in both *Mete* and *SURF* are frequent items. Assume that the set of frequent items is S. Then,

$$S = \{\mathrm{mdID}, \mathrm{mdChar}, \mathrm{refData}, \mathrm{Title}, \mathrm{dsID}, \mathrm{Contact}, \mathrm{Email}, /\}. \tag{1}$$

Definition 5. Frequent path.

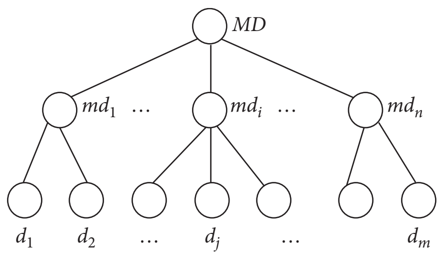

Figure 1: Metadata tree *MD*.

Table 1: Examples of core metadata for meteorological data.

Metadata identifier	Language	Character set	Creation date	Dataset name	Dataset owner
mdID	*mdLang*	*mdChar*	*refDate*	*Title*	*Contact*
—	—	—	—	*rpOrgName/dsID*	*rpIndName/city/e-mail*

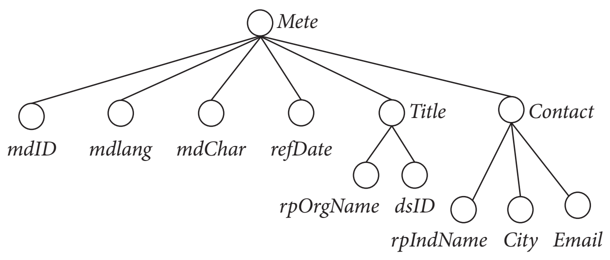

Figure 2: Example of meteorological metadata tree *Mete*.

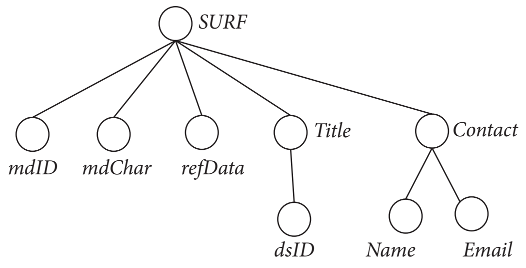

Figure 3: Example of meteorological metadata tree *SURF*.

A metadata path consisting of frequent items is called a frequent path [3]. In the metadata set of *Mete* and *SURF*, the set of frequent paths is assumed to be R. Then,

$$R = \{\langle /, \mathrm{mdID} >, </, \mathrm{mdChar}>, </, \mathrm{refData}>, </, \mathrm{Title}, / >, </, \mathrm{Title}, /, \mathrm{dsID}>, </, \mathrm{Contact}/, >, </, \mathrm{Contact}, /, \mathrm{Email}\rangle\}. \tag{2}$$

Definition 6. Maximal frequent path (MFP).

Frequent paths that are not contained by other frequent paths are called maximal frequent paths, and the set of maximal frequent paths is noted as $\max R$. Then,

$$\max R = \{\langle /, \mathrm{mdID}>, </, \mathrm{mdChar}>, </, \mathrm{refData}>, </, \mathrm{Title}, /, \mathrm{dsID}>, </, \mathrm{Contact}, /, \mathrm{Email}\rangle\}. \tag{3}$$

2.2. Feature Vectors of Metadata Trees. The similarity between metadata trees can be measured using the feature vector of the metadata tree to improve computational efficiency. Therefore, the TF/IDF method can be used to calculate the weight of each MFP and the feature vector of the metadata tree [17].

If W_{ij} denotes the weight of the jth MFP in metadata tree i, TF_{ij} and IDF_j denote the importance of the jth MFP in metadata tree i and in the whole metadata tree set, respectively. Then,

$$\begin{aligned} W_{ij} &= \mathrm{TF}_{ij} \times \mathrm{IDF}_j, \\ \mathrm{TF}_{ij} &= n_{ij} \times \frac{1}{r_{ij}}. \end{aligned} \tag{4}$$

From the perspective of metadata tree similarity, tree nodes at different levels have different weights in similarity calculation. The root nodes have significantly higher weights than the leaf nodes. So, frequent path node level parameter r_{ij} is introduced in the calculation of TF_{ij}. r_{ij} represents the level of the jth MFP in metadata tree i, i.e., the highest level of the node, and the level of the root node is 1. n_{ij} means the number of occurrences of the jth MFP in metadata tree i.

The importance of the j MFP in the metadata tree set is primarily identified by the frequency at which the path appears in the set:

$$\mathrm{IDF}_j = \log \frac{N}{\mathrm{DF}_j} + 1, \tag{5}$$

where N denotes the total number of trees in the metadata tree set whereas DF_j refers to the total number of metadata trees that contain the jth MFP.

Take the *Mete* and *SURF* metadata set for example. When $j = 1$, set $\max R_1 = </\mathrm{mdID}>$, and then $n_{11} = n_{21} = 1$, $r_{ij} = (2, 2)$, $\mathrm{DF}_1 = 2$, and $N = 2$. The result is as follows:

$$\begin{aligned} \mathrm{TF}_{i1} &= \left(\frac{1}{2}, \frac{1}{2}\right), \\ \mathrm{IDF}_1 &= 1, \\ W_{i1} &= \left(\frac{1}{2}, \frac{1}{2}\right). \end{aligned} \tag{6}$$

If there are m MFPs in set max R, metadata tree set $T = \{\mathrm{MD}_1, \mathrm{MD}_2, \ldots, \mathrm{MD}_s\}$ contains a total of s metadata trees. Then, W_{ij} is an $m \times s$ dimensional matrix. The feature vector of metadata tree T_i is denoted as $\mathrm{TW}_i = (W_{i1}, W_{i2}, \ldots, W_{im})$.

2.3. Similarity Calculation of Metadata Trees

2.3.1. Introduction of Key Feature Parameters. Using the importance of MFP in the metadata to measure the key features of the metadata tree excludes the key features such as root node and metadata tree depth from MFP and its feature vector. This means that introducing the identification information d_i of the root node and the depth h_i of the metadata tree will allow more accurate identification of the features of metadata tree T_i. That is,

$$\begin{aligned}\mathrm{TW}_i &= \left(W_{i1}, W_{i2}, \ldots, W_{im}, \ldots, W_{i(m+2)}\right)\\ &= \left(W_{i1}, W_{i2}, \ldots, W_{im}, d_i, h_i\right).\end{aligned} \quad (7)$$

2.3.2. Similarity Calculation for Heterogeneous Metadata Trees. The similarity between TW_i and TW_j is computed to identify the similarity between T_i and T_j in the metadata tree. W_{ij} has numerical characteristics in TW_i and TW_j, while d_i and h_i have categorical features. Thus, the similarity (or integrated distance) of feature vectors TW_i and TW_j is mainly a combination of $\mathrm{dis\_}d(\mathrm{TW}_i, \mathrm{TW}_j)$ and $\mathrm{dis\_}c(\mathrm{TW}_i, \mathrm{TW}_j)$.

For the calculation of numerical values, Euclidean distance is adopted:

$$\mathrm{dis\_}d\left(\mathrm{TW}_i, \mathrm{TW}_j\right) = \sqrt[2]{\sum_{p=1}^{m}\left(W_{ip} - W_{jp}\right)^2}. \quad (8)$$

For categorical values, the calculation is performed through dissimilarity:

$$\begin{aligned}\mathrm{dis\_}c\left(\mathrm{TW}_i, \mathrm{TW}_j\right) &= \sum_{d=m+1}^{m+2} \mathrm{xor}\left(W_{id}, W_{jd}\right),\\ \mathrm{xor}\left(W_{id}, W_{jd}\right) &= \begin{cases} 0, & W_{id} = W_{jd},\\ 1, & W_{id} \neq W_{jd}.\end{cases}\end{aligned} \quad (9)$$

Due to different computing methods, there are also significant differences in value ranges. As TW_i is a predominantly numerical value, a weighted proportion μ is hence introduced to calculate the integrated distance:

$$\mathrm{dis}\left(\mathrm{TW}_i, \mathrm{TW}_j\right) = \mathrm{dis\_}d\left(\mathrm{TW}_i, \mathrm{TW}_j\right) + \mu \times \mathrm{dis\_}c\left(\mathrm{TW}_i, \mathrm{TW}_j\right). \quad (10)$$

3. Design of Metadata Tree Clustering Algorithms

3.1. GSO Algorithm. The GSO algorithm is mainly used to realize optimization by simulating the characteristics of glowworms in terms of luminescence and aggregation [14], and the attraction and aggregation between individual glowworms are primarily achieved by their own brightness (fluorescein value). The steps of this algorithm are described as follows:

(i) Step 1: initialize glowworm position $Z = \{z_1, z_2, \ldots, z_n\}$, and initialize and assign a value to glowworm number n, moving step length s, fluorescein initial value l_0, and other related parameters.

(ii) Step 2: calculate the fitness of the glowworm based on the objective function, i.e., fluorescein value $l_i(t)$:

$$l_i(t) = (1-\rho)l_i(t-1) + \gamma J\left(x_i(t)\right). \quad (11)$$

In the above, $J(x_i(t))$ denotes the conversion of positions to fluorescein values, ρ means volatility of fluorescein, and γ stands for the enhancement ratio of fluorescein.

(iii) Step 3: calculate the probability $p_{ij}(t)$ that glowworm z_i moves toward the direction of glowworm z_j in the field:

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k\in N_i(t)} l_k(t) - l_i(t)}. \quad (12)$$

$N_i(t)$ represents the set of glowworms in the field.

(iv) Step 4: update the position of each glowworm z_i:

$$x_i(t+1) = x_i(t) + s \times \frac{x_j(t) - x_i(t)}{\left\|x_j(t) - x_i(t)\right\|}. \quad (13)$$

(v) Step 5: update the radius of the glowworm's decision domain, control the number of glowworms in the field, avoid excessive aggregation of glowworms, and improve the global optimization effect:

$$r_d^i(t+1) = \min\left\{r_s, \max\left\{0, r_d^i(t) + \beta\left(n_t - \left|N_i(t)\right|\right)\right\}\right\}. \quad (14)$$

$r_d^i(t)$ denotes the perceptual realm of glowworm z_i at time t; n_t means the threshold of the number of glowworms in the field; r_s refers to the radius of the perceptual realm.

(vi) Step 6: determine whether the algorithm ends and decides whether to proceed to the next round of iterations until the end of the algorithm.

3.2. Flow of the K-Prototypes Algorithm. K-means is a classic clustering algorithm based on division. The algorithm is mainly suitable for clustering numerical datasets. Based on the K-means algorithm, Huang proposed the K-modes algorithm aimed at categorical datasets. These algorithms use different methods to calculate the distance between data objects. To realize the clustering of mixture datasets, the K-prototypes algorithm combines the basic methods of the K-means and K-modes algorithm [18]. The flow of K-means, K-modes, and K-prototypes is very similar, but they are suitable for different types of datasets. The K-prototypes clustering algorithm directed at data objects with mixed attributes such as numeric and categorical data. The steps of the K-prototypes algorithm can be described as follows:

(i) Step 1: *Initialization*. Input the category number k of clustering and randomly select k data points in the dataset as the initial cluster center.

(ii) Step 2: *Clustering*. Traverse the whole dataset, calculate the distance between each data point X_i and the initial cluster center, take the minimum distance, and assign X_i to the corresponding cluster center. Thus, the dataset can be divided into k nonintersecting cluster sets $C = \{C_1, C_2, \ldots, C_k\}$.

(iii) Step 3: *Updating the Cluster Center*. For each cluster in the cluster set, calculate the new cluster center.

(iv) Step 4: *Determining Algorithm End Condition*. If there is no change after the original cluster center is updated or the maximum number of iterations is reached, the algorithm terminates; otherwise, it will go to Step 2 and enter the next round of iteration.

3.3. GSOKP-FP Design. GSO, K-prototypes, and MFP are combined to design the GSOKP-FP algorithm to realize the clustering of metadata trees. The framework of this algorithm is shown in Figure 4.

GSOKP-FP flow description is as follows:

(i) Step 1: identify the MFP in the metadata tree set. According to Section 2.1, filter out the MFPs in the metadata tree set to form an MFP set.

(ii) Step 2: calculate the feature vectors of the metadata tree. According to Section 2.2, calculate the feature vectors of the metadata tree and form a feature vector set.

(iii) Step 3: optimize the initial cluster center. Perform GSO algorithm on the feature vector set to solve multiple extreme points in the point set. In the meantime, adopt the GSO algorithm based on the optimization of the good point set to improve the global optimization effect [15].

(iv) Step 4: K-prototypes clustering. Select k extreme points as the initial cluster centers of K-prototypes to carry out the K-prototypes algorithm.

(v) Step 5: output the metadata tree clustering results. According to the K-prototypes algorithm, output k clusters, mark the category of metadata trees, and output the clustering results.

In addition, the part of GSOKP-FP flow for dataset clustering can be named the GSOKP algorithm.

4. Analysis of Experimental Data and Results

4.1. Selection of Datasets. The experiments in this paper are based on the metadata structure shown in Table 1 to construct a metadata set consisting of metadata trees. The metadata tree nodes are presented in Table 2.

A breadth traversal of all the nodes of the metadata tree with Root1 as the root node is performed to form a sequence of nodes listed as follows:

$$\text{NODE}_1 = \{\text{Root1}, \text{mdID}, \text{mdLang}, \text{mdChar}, \text{refData}, \text{Title}, \text{rpOrgName}, \text{dsID}, \text{Contact}, \text{rpIndName}, \text{City}, \text{Email}\}. \tag{15}$$

The nodes in a nonempty metadata tree are ordered, and corresponding nodes are identified via the binary system, i.e., 1 means the node included, 0 not included. The examples of metadata trees are given as follows:

The metadata tree shown in Figure 5 can be denoted as Tree_1 = $(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$.

The metadata tree shown in Figure 6 can be denoted as Tree_2 = $(1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0)$.

Since the set of leaf nodes can uniquely identify a metadata tree, it is possible to uniquely identify a metadata tree by numbering the leaf nodes. For example, a metadata tree with Root1 can have up to 9 leaf nodes. In other words, it can yield 511 metadata trees $(2^9 - 1)$. In 511 metadata trees with Root1 as the root node, 511 binary numbers can be represented as 000000001 to 111111111 according to the position of leaf nodes. When the number of leaf nodes (without individual root node) is d, and $d = 1$, there are 9 corresponding binary numbers in 511 binary numbers: 000000001,000000010,000000100, 000001000,000010000, 000100000, 001000000, 010000000, and 100000000. That is, $C_9^1 = 9$.Similarly, the number of binary numbers (metadata trees) under different d values can be calculated with C_9^d. The number of metadata trees with Root1 (or Root2 and Root3) as the root node is shown in Table 3. Also, the number of metadata trees with Root2 and Root3 as root nodes is 511, respectively. The metadata tree for the experiment is selected among a total of 1,533 metadata trees.

Because of the mutual inclusion of metadata trees with different numbers of leaf nodes, two types of datasets are experimentally designed: one is the set of metadata trees with the same number of leaf nodes (no inclusion) and the other is the set of random metadata trees (inclusion).

When $d = 7$, 108 metadata trees with Root1, Root2, and Root3 as root nodes are recorded as Tset_1, which is taken as the experimental object in the first experiment.

When the number of leaf nodes is random, 1,000 metadata trees with Root1, Root2, and Root3 as the root node are randomly selected and recorded as Tset_2, which is taken as the experimental object of the second experiment.

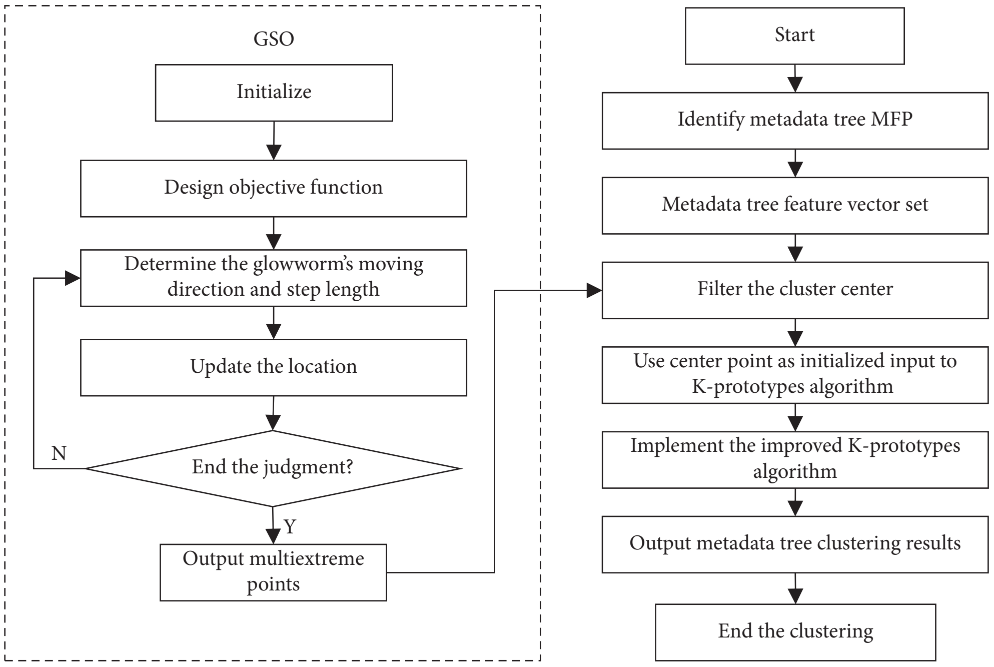
GSO
Initialize
Design objective function
Determine the glowworm's moving direction and step length
Update the location
End the judgment?
N
Y
Output multiextreme points
Start
Identify metadata tree MFP
Metadata tree feature vector set
Filter the cluster center
Use center point as initialized input to K-prototypes algorithm
Implement the improved K-prototypes algorithm
Output metadata tree clustering results
End the clustering

Figure 4: Flow of the GSOKP-FP algorithm.

Table 2: Node information of metadata tree.

Root1	*mdID* —	*mdLang* —	*mdChar* —	*refDate* —	*Title* *rpOrgName/dsID*	*Contact* *rpIndName/City/Email*
Root2	*mdID* —	*Lang* —	*Char* —	*refDate* —	*Title* *OrgName/dsID*	*Contact* *rpName/City/Email*
Root3	*ID* —	*Lang* —	*Char* —	*refDate* —	*Heading* *Name/dsID*	*Information* *IndName/City/Tel*

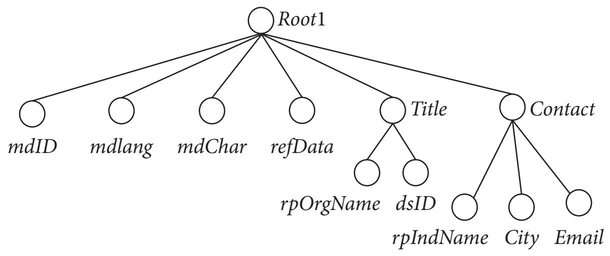

Figure 5: Example of metadata tree Tree_1.

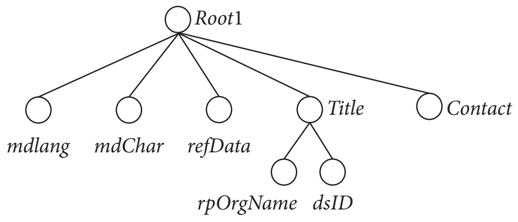

Figure 6: Example of metadata tree Tree_2.

Table 3: Correspondence between leaf node number and metadata tree number.

Number of leaf nodes (d)	1	2	3	4	5	6	7	8	9
Number of metadata trees (C_9^d)	9	36	84	126	126	84	36	9	1

4.2. Experiment Description

Experiment 1. Metadata tree set Test_1.

The value of frequency threshold θ has a significant impact on the performance and efficiency of the algorithm. The larger the value of θ, the fewer frequent paths and the shorter the execution time, but more information of the metadata tree will be lost. The smaller the value of θ, the greater the number of frequent paths, and the less information loss, the longer execution time. Therefore, 50% is a median value, which also conforms to the experience of reference [3] on the value of parameter θ.

If $\theta = 50\%$, then

$$\begin{aligned}
S &= \{\text{mdID, Lang, Char, refData, Title, dsID, Contact, City, Email/}\},\\
R &= \{\langle /\text{mdID}>, </\text{Lang}>, </\text{Char}>, </\text{refData}>, </\text{Title}>, </\text{Title/}>, </\text{Title/dsID}>,\\
&\quad </\text{dsID}>, </\text{Contact}>, </\text{Contact/}>, </\text{Contact/City}>, </\text{city}>, </\text{Contact/email}>, </\text{Email}\rangle\},\\
\max R &= \{\langle /\text{mdID}>, </\text{Lang}>, </\text{Char}>, </\text{refData}>, </\text{Title}>, </\text{Title/dsID}>, </\text{dsID}>,\\
&\quad </\text{Contact}>, </\text{Contact/city}>, </\text{City}>, </\text{Contact/email}>, </\text{Email}\rangle\}.
\end{aligned} \tag{16}$$

The individual paths in MFP are numbered as p1, p2, p3, p4, p5, ... ,p12.

In Tset_1, the feature vectors of the metadata tree are shown in Table 4.

Experiment 2. Take metadata tree set Tset_2 as the experimental object.

If $\theta = 50\%$, then

$$\begin{aligned}
S &= \{\text{refData, Title, dsID, Contact, City,/}\},\\
&= \{</\text{refData}>, </\text{Title}>, </\text{Title/}>, </\text{Title/dsID}>, </\text{dsID}>, </\text{Contact}>,\\
&\quad </\text{Contact/}>, </\text{Contact/City}>, </\text{City}>,\},\\
\max R &= \{\langle /\text{refData}>, </\text{Title}>, </\text{Title/dsID}>, </\text{dsID}>, </\text{Contact}>, </\text{Contact/City}>, </\text{City}\rangle\}.
\end{aligned} \tag{17}$$

The individual paths in MFP are numbered as p1, p2, p3, p4, p5, p6, and p7. Similarly, in Tset_2, the corresponding feature vectors of the metadata tree are shown in Table 5.

4.3. Analysis of Experimental Results. The root node information and depths of metadata trees are added to the feature vectors, and the feature vector sets are clustered. Next, the GSOKP-FP algorithm is run to obtain the corresponding experimental results, as displayed in Tables 6 and 7.

According to practical results, we focused on comparing the algorithm covered in this paper with other clustering algorithms, such as K-means, K-modes, K-prototypes, KL-FCM-GM, SBAC, EKP, DC-MDACC, and the metadata tree clustering algorithm mentioned in the literature [3] in terms of clustering accuracy, and time complexity. Clustering accuracy refers to the proportion of accurately classified samples to total samples.

First, from the perspective of classification accuracy, the comparison information of the algorithm designed in this paper with other intelligent algorithms on some UCI datasets and the K-means, K-modes, and K-prototypes algorithms on metadata tree sets in respect to clustering accuracy is indicated in Tables 8 and 9. The clustering accuracy of other algorithms can be got from the literature [19]. The test given in reference [3] is to cluster four metadata trees by pairwise comparison of their similarity. Therefore, the clustering algorithm given in reference [3] can be recorded as MCM-FP is not suitable for large-scale metadata tree set clustering, such as the metadata tree set Tset_1 and Tset_2 in this paper.

Compared with other typical clustering algorithms, the test results show that the GSOKP algorithm has higher clustering accuracy, and the algorithm is more suitable for clustering of mixed-attribute data. The advantages of GSOKP in clustering accuracy are mainly due to the following reasons: firstly, the GSO algorithm provides a better initial clustering center for the K-prototypes algorithm, which improves the clustering effect of the K-prototypes algorithm significantly; secondly, a new calculation method of the distance between mixed-attribute data has been designed in the GSOKP algorithm. The calculation method can better describe the distance or similarity between mixed-attribute data. In addition, the test results show that the DC-MDACC algorithm is also excellent. The test results of DC-MDACC are better than the GSOKP algorithm in the test of the dataset Acute and Statlog Heart. The main reason is that a swarm intelligence optimization algorithm is also used in the DC-MDACC algorithm and more preprocessing is performed on the test data. Therefore, the time complexity of the DC-MDACC algorithm is higher.

Through the GSOKP-FP algorithm, it is able to achieve better classification accuracy than the conventional K-means

Table 4: Characteristic vectors of maximal frequent path of Tset_1.

Metadata tree (Root1 as root node)	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
01111111101	0.00	0.00	0.00	0.55	0.59	0.64	0.37	0.59	0.00	0.00	0.64	0.43
01111111110	0.00	0.00	0.00	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.00	0.00
10111111011	0.64	0.00	0.00	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.64	0.43
10111111101	0.64	0.00	0.00	0.55	0.59	0.64	0.37	0.59	0.00	0.00	0.64	0.43
10111111110	0.64	0.00	0.00	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.00	0.00
11011111101	0.64	0.00	0.00	0.55	0.59	0.64	0.37	0.59	0.00	0.00	0.64	0.43
11011111110	0.64	0.00	0.00	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.00	0.00
11111011101	0.64	0.00	0.00	0.55	0.59	0.00	0.00	0.59	0.00	0.00	0.64	0.43
Metadata tree (Root2 as root node)	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
00111111111	0.00	0.00	0.64	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.64	0.43
01011111111	0.00	0.64	0.00	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.64	0.43
01111111110	0.00	0.64	0.64	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.00	0.00
10011111111	0.64	0.00	0.00	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.64	0.43
10111111110	0.64	0.00	0.64	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.00	0.00
11001111111	0.64	0.64	0.00	0.00	0.59	0.64	0.37	0.59	0.64	0.37	0.64	0.43
11111111001	0.64	0.64	0.64	0.55	0.59	0.64	0.37	0.59	0.00	0.00	0.64	0.43
11111111010	0.64	0.64	0.64	0.55	0.59	0.64	0.37	0.59	0.64	0.37	0.00	0.00
Metadata tree (Root3 as root node)	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
00111111111	0.00	0.00	0.64	0.55	0.00	0.00	0.37	0.00	0.00	0.37	0.00	0.00
01011111111	0.00	0.00	0.00	0.55	0.00	0.00	0.37	0.00	0.00	0.37	0.00	0.00
01101111111	0.00	0.64	0.64	0.00	0.00	0.00	0.37	0.00	0.00	0.37	0.00	0.00
01111111101	0.00	0.64	0.64	0.55	0.00	0.00	0.37	0.00	0.00	0.00	0.00	0.00
01111111110	0.00	0.00	0.64	0.55	0.00	0.00	0.37	0.00	0.00	0.37	0.00	0.00
10111111101	0.00	0.00	0.64	0.55	0.00	0.00	0.37	0.00	0.00	0.00	0.00	0.00
11011111101	0.00	0.00	0.00	0.55	0.00	0.00	0.37	0.00	0.00	0.00	0.00	0.00
11011111110	0.00	0.64	0.00	0.55	0.00	0.00	0.37	0.00	0.00	0.37	0.00	0.00

Table 5: Characteristic vectors of maximal frequent path of Tset_2.

Metadata tree (Root1 as root node)	p1	p2	p3	p4	p5	p6	p7
00000001010	0.00	0.00	0.00	0.00	0.61	0.74	0.43
00000001101	0.00	0.00	0.00	0.00	0.61	0.00	0.00
00001010000	0.00	0.41	0.73	0.43	0.00	0.00	0.00
00001011010	0.00	0.41	0.73	0.43	0.61	0.74	0.43
00010000000	0.65	0.00	0.00	0.00	0.00	0.00	0.00
00011011010	0.65	0.41	0.73	0.43	0.61	0.74	0.43
00011101010	0.65	0.41	0.00	0.00	0.61	0.74	0.43
00011111100	0.65	0.41	0.73	0.43	0.61	0.00	0.00
Metadata tree (Root2 as root node)	p1	p2	p3	p4	p5	p6	p7
00001011011	0.00	0.41	0.73	0.43	0.61	0.74	0.43
00001011101	0.00	0.41	0.73	0.43	0.61	0.00	0.00
00001100000	0.00	0.41	0.00	0.00	0.00	0.00	0.00
00001101010	0.00	0.41	0.00	0.00	0.61	0.74	0.43
00001110000	0.00	0.41	0.73	0.43	0.00	0.00	0.00
00001111111	0.00	0.41	0.73	0.43	0.61	0.74	0.43
00010001111	0.65	0.00	0.00	0.00	0.61	0.74	0.43
00101011010	0.00	0.41	0.73	0.43	0.61	0.74	0.43
Metadata tree (Root3 as root node)	p1	p2	p3	p4	p5	p6	p7
00010001110	0.65	0.00	0.00	0.00	0.00	0.00	0.43
00011010000	0.65	0.00	0.00	0.43	0.00	0.00	0.00
00011011110	0.65	0.00	0.00	0.43	0.00	0.00	0.43
00011101010	0.65	0.00	0.00	0.00	0.00	0.00	0.43
00101011010	0.00	0.00	0.00	0.43	0.00	0.00	0.43
00101011110	0.00	0.00	0.00	0.43	0.00	0.00	0.43
00111010000	0.65	0.00	0.00	0.43	0.00	0.00	0.00
00111011110	0.65	0.00	0.00	0.43	0.00	0.00	0.43

and K-modes methods in both the conditional and random selection of metadata trees to form metadata tree sets. The algorithm designed in this paper is more suitable for the clustering of bigger metadata tree sets, while the method described in the literature [3] is more suitable for the clustering of smaller metadata tree sets.

Table 6: Experimental results of metadata tree set Tset_1.

Metadata tree set	Number of metadata trees	Number of MFPs	Number of categories	Average classification precision (%)
Tset_1	108	12	3	93.1

Table 7: Experimental results of metadata tree set Tset_2.

Metadata tree set	Number of metadata	Number of MFPs	Number of categories	Average classification precision (%)
Tset_2	1000	7	3	84.1

Table 8: Comparison of clustering accuracy on UCI datasets (%).

UCI dataset	KL-FCM-GM	SBAC	EKP	DC-MDACC	GSOKP
Iris	33.5	42.6	45.0	96.0	97.0
Soybean	90.3	61.7	97.2	95.7	96.8
Acute	68.2	50.8	50.8	91.7	82.5
Statlog Heart	75.8	75.2	54.5	84.8	77.1

Table 9: Comparison of clustering accuracy on metadata tree sets (%).

Metadata tree set	K-means	K-modes	K-prototypes	GSOKP-FP
Tset_1	76.8	75.9	85.7	93.1
Tset_2	80.0	63.6	82.2	84.1

Next, the algorithm mentioned in this chapter mainly consists of three parts in terms of time complexity: swarm intelligence algorithm for selecting initial cluster centers, K-prototypes algorithm to solve the clustering results, and maximum frequent path mining for constructing the metadata tree dataset. Set the number of glowworms as m, the number of metadata trees as n, and the number of iterations as t. Then, the time complexity of the GSO algorithm for solving the initial cluster center can be denoted as $O(t \times m \times n)$ for convenience. Set t as the number of iterations, n as the number of metadata trees, and k as the number of clusters. Then, the time complexity of the K-prototypes algorithm for solving clusters is like this: $O(t \times k \times n)$, whereby the combined time complexity $O(t \times (m + k) \times n)$ is obtained. The comparison of time complexity is shown in Table 10, and time complexity of other algorithm can be got from the literature [3, 19].

Compared to K-means, K-modes, and K-prototypes, the algorithm covered in this chapter has higher time complexity. The time complexity of the algorithm in this chapter is lower compared to that of the agglomerative hierarchical clustering algorithm stated in the literature [3]. The time complexity of this method mainly consists of two iterative clustering algorithms: creating metadata tree similarity matrices and scanning metadata tree similarity matrices, of which the combined time complexity is about $O(n^3)$.

Table 10: Time complexity analysis of algorithms.

Algorithm	Time complexity
KL-FCM-GM	$O(t \times (d \times lk + c \times n))$
SBAC	$O(n^2 + m_c^2 \log(m_c^2))$
EKP	$O(t \times k \times n)$
K-prototypes	$O(s \times k \times n)$
MCM-FP	$O(n^3)$
DC-MDACC	$O(t \times m \times (n^2 + (n^2 - n/2) + n\log n + (n/2)))$
GSOKP	$O(t \times (m + k) \times n)$
GSOKP-FP	$O(t \times (m + k) \times n + n^2)$

5. Conclusions

The GSOKP-FP algorithm designed in this paper introduces the GSO algorithm and K-prototypes algorithm into the solution of metadata tree clustering, which enables the clustering of the metadata tree sets by clustering the feature vectors of metadata trees. MFP can better describe the key features of a metadata tree and effectively reduce the dimension of numerical computation and the time complexity. However, since MFP extracts more common information between metadata trees, through which some key information is lost while reducing the numerical dimension. In this paper, information such as root node and tree depth are added to the feature vector described by MFP to improve the computing accuracy of metadata tree similarity and the clustering precision. The experiments show that the GSOKP-FP algorithm designed in this paper is able to achieve a better metadata tree clustering effect.

Data Availability

The experimental object (metadata tree sets) used to support the findings of this study has been constructed in this paper. The datasets (Iris, Soybean, Acute, and Statlog Heart) used to support the findings of this study have been deposited in the UCI repository (http://archive.ics.uci.edu/ml/datasets.html).

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this manuscript.

Acknowledgments

The work was supported by the research project of the National Social Science Foundation of China (no. 20BTQ046).

References

[1] Y. Wang, X. Liu, C. Z. Xu et al., “A review on metadata management in large-scale distributed file systems,” *Journal of Integration Technology*, vol. 5, no. 2, pp. 57–72, 2016.

[2] B. Fung, K. Wang, and M. Ester, “Hierarchical document clustering using frequent item sets,” in *Procceedings of the SIAM International Conference on Data Mining*, pp. 1–12, San Francisco, CA, USA, May 2003.

[3] X. Z. Feng and N. Chen, “Metadata clustering method based on maximal frequent path,” *Computer Engineering*, vol. 36, no. 21, pp. 40–42, 2010.

[4] Z. X. Huang, “Clustering large data sets with mixed numeric and categorical values,” in *Proceedings of the 1st Pacific Asia Conference on Knowledge Discovery and Data Mining*, pp. 21–34, Singapore, February 1997.

[5] M. Du, S. Ding, and Y. Xue, “A novel density peaks clustering algorithm for mixed data,” *Pattern Recognition Letters*, vol. 97, pp. 46–53, 2017.

[6] V. Muyumba, “XML for catalogers and metadata librarians,” *Technical Services Quarterly*, vol. 31, no. 3, pp. 308–310, 2014.

[7] S. R. Feng, W. W. Pan, and Z. Y. Lin, “XML documents clustering based on improved k-medoids algorithm,” *Computer Engineering*, vol. 41, no. 9, pp. 56–62, 2015.

[8] C. Y. Wang, Q. W. Du, J. Sun et al., “Clustering XML documents based on feature order preference,” *Computer Engineering and Applications*, vol. 52, no. 12, pp. 64–68, 2016.

[9] G. Costa and R. Ortale, “XML clustering by structure-constrained phrases: a fully-automatic approach using contextualized N-grams,” *International Journal on Artificial Intelligence Tools*, vol. 26, no. 1, pp. 176–180, 2017.

[10] H. H. Yang, K. Wang, and L. Q. Li, “K-means clustering algorithm based on adaptive cuckoo search and its application,” *Journal of Computer Applications*, vol. 36, no. 8, pp. 2066–2070, 2016.

[11] Q. Kang, S. Liu, M. Zhou, and S. Li, “A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence,” *Knowledge-Based Systems*, vol. 104, pp. 156–164, 2016.

[12] J. Jokinen, T. Raty, and T. Lintonen, “Clustering structure analysis in time-series data with density-based clusterability measure,” *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1332–1343, 2019.

[13] K. N. Krishnanand and D. Ghose, “Glowworm swarm optimisation: a new method for optimising multi-modal functions,” *International Journal of Computational Intelligence Studies*, vol. 1, no. 1, pp. 93–119, 2009.

[14] R. Lukyanenko, V. C. Storey, and A. Castellanos, “Introducing GSO: a general systemist ontology,” in *Proceedings of the ER Forum*, pp. 1–15, Vienna, Austria, 2020.

[15] Y. Li, Z. Ni, F. Jin, J. Li, and F. Li, “Research on clustering method of improved glowworm algorithm based on good-point set,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 8724084, 8 pages, 2018.

[16] F. Wang, Y. Liu and L. Kang, “Research on scenario service of mobile government based on maximum frequent pattern mining,” *Journal of Modern Information*, vol. 40, no. 1, pp. 41–48, 2020.

[17] D. K. Kardaras, S. Kaperonis, S. Barbounaki, I. Petrounias, and K. Bithas, “An approach to modelling user interests using TF-IDF and fuzzy sets qualitative comparative analysis,” *Artificial Intelligence Applications and Innovations*, vol. 519, pp. 606–615, 2018.

[18] Z. Jia and L. Song, “Weighted k-prototypes clustering algorithm based on the hybrid dissimilarity coefficient,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 5143797, 13 pages, 2020.

[19] J. Y. Chen and H. H. He, “Research on density-based clustering algorithm for mixed data with determine cluster centers automatically,” *Acta Automatica Sinica*, vol. 41, no. 10, pp. 1798–1813, 2015.