IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 19, 2019, accepted August 3, 2019, date of publication August 13, 2019, date of current version August 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2935089

Heterogeneous Graph Based Similarity Measure for Categorical Data Unsupervised Learning

YANQING YE[1,2], JIANG JIANG[1], BINGFENG GE[1], (Member, IEEE), KEWEI YANG[1], AND H. EUGENE STANLEY[2]

[1]College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
[2]Center for Polymer Studies, Department of Physics, Boston University, Boston, MA 02215, USA

Corresponding author: Jiang Jiang (jiangjiangnudt@nudt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71671186, Grant 71501182, Grant 71571185, and Grant 71690233, and in part by the Research Project of National University of Defense Technology.

ABSTRACT Different from numerical attributes, measuring the similarity between categorical attributes is more complex due to their non-inherently ordered characteristic, especially in an unsupervised scheme. This work, therefore, presents a new method, Heterogeneous Graph-based Similarity measure (HGS), to measure the similarity between categorical data for unsupervised learning. In order to capture the possible complex relationships hidden among attributes, a heterogeneous weighted graph is creatively constructed by extracting the information from categorical data. Both objects and attribute values are represented as nodes and their occurrence and co-occurrence relationships are shown as edges. Based on a derived node-pair graph, three rules are used to iteratively update the similarity scores between object pairs and attribute-value pairs until the scores converge. We also analyze its complexities and validate the metric properties and convergence. In experiment validation, five state-of-the-art measures are compared with HGS based on 20 UCI datasets and 6 high-dimensional datasets in the medical domain in both k-modes and spectral clustering and similarity search experiments. The results show although no measure can outperform all other measures on all datasets, HGS can perform better in both clustering and similarity search tasks on the whole. Finally, six studies further discuss the convergence, time cost, and parameter sensitivity of the HGS, explore its application to imbalanced class distribution, and compare it with its variants by different initialization and graph construction.

INDEX TERMS Unsupervised learning, similarity measure, categorical data, heterogeneous graph-based similarity (HGS).

I. INTRODUCTION

With the continuous increase of data produced from media, medical, and social network, *etc.*, to find the relationship between objects has caught the attention of researchers. Similarity, as an important relationship, is a numeral measure of the degree to which the two data objects are alike, which is usually described as a distance with dimensions representing features of the object [1]. It's fundamental and essential for effective data analytics in various domains, such as data mining [2], recommendation [3], and information retrieval [4], [5], which can determine whether the learning process is reliable and the outcome is accurate. Normally, various attributes can be utilized to determine the similarity between objects. When they are described by numerical attributes, it's more natural to compare them by a series of mature methods, like Euclidean and Minkowski distances [6]. However, when they are described by categorical (nominal) attributes, the similarity analysis is much more complex with the values unordered and even incomparable [7]. Thus, it's very difficult and unstraightforward to quantify the difference between categorical objects.

The wide existence of categorical data makes it an urgent problem to calculate the similarity between categorical objects, especially in an unsupervised scheme, because tagging data requires a lot of manpower and resources. However, unlike supervised learning, similarity measure in the unsupervised scheme for categorical data has received much less attention until now [8], [9]. Without both the label information and the numerical attributes, it's much

The associate editor coordinating the review of this article and approving it for publication was Kuan Zhang.

TABLE 1. An example: Evaluation of applications for nursery schools.

family	parents' occupation	house condition	family form	finance
family1	usual	less_conv	foster	inconv
family2	usual	less_conv	incomplete	inconv
family3	usual	critical	incomplete	inconv
family4	pretentious	critical	complete	convenient
family5	pretentious	convenient	incomplete	convenient
family6	pretentious	convenient	complete	convenient

challengeable to distinguish different categorical values [10]. Currently, only limited efforts have been made, mainly including matching-based [11], frequency-based [12], and information theory [13] based methods. However, all of these methods fail to capture the intrinsic characteristics and recognize the complex hidden relationship between attributes. Matching-based methods can just measure the similarity roughly. When all the objects share an equal number of identical attribute values, their similarities will be the same. As for frequency-based and information theory-based methods, they will easily fail when the attribute values are uniformly distributed.

Taking a typical categorical dataset, the Nursery dataset [14], as an example, which is obtained from the forms of various families applying for nursery schools. As shown in Table 1, six families (objects) are presented with four categorical attributes: parents' occupation, family's house condition, the form of family, and the financial condition of the family. By the rule of matching, we can't tell among families 4, 5, and 6. None of them is similar to family 1 because they don't have any matching attributes with family 1. While from both frequency and information theory perspectives, family 1 is more similar to family 5 than family 4 and 6 because they share closer frequency of ''foster'' and ''incomplete'' in ''family form'' attribute. Although they try to capture the difference from the distribution of attribute values, they are still insufficient. Because the complex relationship hidden among attributes has been neglected. For instance, a family's finance and the house condition are always related to the parents' occupation. In the table, when the parents' occupation is usual, the family's finance is inconvenient, and the house condition tends to be less convenient or critical. While a pretentious occupation can always bring convenient finance to the family and convenient house condition. Besides, whether a family is complete or not can also be closely related to the family's house condition and finance, and even the parent's selection of occupation. For instance, a single mother may choose jobs that can have much flexible time to take care of her children. Thus, from the above analysis, the failure of these methods may lie in only capturing information within attribute while neglecting the genuine information hidden among attributes.

The above example shows that much more complex relationships may be embedded in categorical attributes. The occurrence of one attribute may depend on another one or even a set of other attributes. Owing to the deficient information that can be utilized in unsupervised learning, it can be very meaningful to adequately capture this information. Obviously, the complex co-occurrences and co-dependencies among attributes are not straightforwardly quantified. Thus, we consider finding an effective tool to extract relationship information. As a predominant tool to represent relationships, a graph structure is very effective to visualize the data and capture the hidden relationship among attribute values [15]. Various relationships can be represented as different types of edges and the weights on edges can represent the strengths of the relationships, *e.g.*, the number of co-occurrence [16].

Drawing on the graph operation, SimRank [17] is a traditional similarity measure on a graph which iteratively calculates the similarity scores between nodes on the graph and can converge to a stable state fastly. However, the homogeneous design can only capture the relationships between objects or attribute values, while neglecting much more complex but valuable information between object and attribute value, between attribute value and attribute value. Inspired it, in this work, we creatively introduce the graph structure to capture the information from the categorical dataset. In addition to objects being represented as nodes in SimRank, we also modeled attribute values as nodes since the complex relationships may be hidden among specific attribute values. More precisely, both objects and attribute values are represented as nodes and the occurrence relationships between objects and attribute values, and the co-occurrence relationships between attribute values, are represented as the different types of edges.

On the base of the heterogeneous graph structure, to calculate the similarity between object nodes is another challenge. As the analysis above, a set of attribute values may co-occur in the same object. For instance, in the Table 1, the usual parents' occupation, less convenient house condition, inconvenient finance, and less convenient house condition tend to co-occur in the same family. Compared with the pairwise relationship, it's more challengeable to mine complex relationships, that is, one attribute possibly depends on multiple other attributes. In order to make use of this information, we further consider deriving a node pair graph from the previously constructed heterogeneous graph. The score of each node pair can be set as the similarity value between the nodes within the pair. Through iterating the scores across the whole graph, the score can ''flow'' across the whole graph, traveling between node pairs via the edges. Thus, during this iterative process, the complex relationship information can be captured and reflected in the final score.

Our assumption is that similar objects may possess similar attribute values, while similar attribute values may also belong to similar objects. So we can infer that if the values of the attribute are similar, the values of another attribute that tend to co-occur with them are more possibly similar. Hence, the iteration process can be conducted between object

pairs and attribute-value pairs, and between attribute-value and attribute-value pairs.

Therefore, our work first constructs a heterogeneous graph composed of multiple types of nodes and edges by extracting the occurrence and co-occurrence information from the categorical dataset. Based on it, a node-pair graph is derived as the basis of the score iteration process, during which the scores flow across the whole graph until convergence. Finally, we can obtain the similarity measure for both objects and attribute values.

The rest of paper is organized as follows. Some related works in categorical learning are introduced in the following section. In Section III, the problem was formulated and a learning framework is present. In Section IV, we first illustrate the detailed graph construction process. Then, the detailed algorithm of categorical data similarity measure (HGS) is proposed. In Section V, we prove the metric properties and the convergence of solution and analyze its computational efficiency compared with other methods. In the following Section VI, we carry out clustering and similarity search experiments on 26 datasets by comparing with five common-used methods and extensive discussions on the characteristics, time and parameter sensitivity of the HGS. Finally, a conclusion is drawn in Section VII.

II. RELATED WORKS

In recent years, the increasing efforts have been contributed to addressing similarity learning for categorical data in various contexts. As known, it's much more complex than that for numerical data owing to its unorder characteristic. Three major groups are intended to solve this problem. The most simple one is matching-based (or overlap) measure, like Hamming distance (Hamming) [11], which naturally counts the number of attributes that are matched between two objects as the similarity value. It's very fast and easy to use in various fields. Another popular method group is frequency-based measure, like Inverse Occurrence Frequency (IOF), Occurrence Frequency (OF) [18]. By comparing the frequency distribution of categorical values within the attribute, we found it can work better in some conditions than the overlap measure. Furthermore, the difference between IOF and OF lies in the weight on less or more frequent values. The final one is information-theoretical similarity (Lin) [13]. However, all these methods are too rough to precisely capture more details of useful information. They only see the local difference within attributes, however, lose more possible valuable information that may hide between attributes. Besides, they do not consider the distribution of values, which is often captured for numerical attributes.

Believing the dependence between attributes, a variety of intensive data-driven methods have been proposed to capture the context information reflecting the dependency among the data samples as a supervision to improve unsupervised learning [19]. Elementary exploration introduced the Pearson or Jaccard coefficient to extract the correlation between attributes. ALGO_DISTANCE (ALGO in short) has considered the attribute-value distribution in the data set and incorporated the co-occurrence relationship between attribute-value pairs [20]. Compared with Hamming and OF, it has shown more suitability and effectiveness. However, it can't distinguish objects when the attribute values in the whole dataset are equally distributed and their co-occurrent is similar. Much more efforts have been continuously contributed to capturing the co-occurrence relationship between values for various attributes. By combining the intra-relationships and inter-relationships of attribute values, Coupled Object Similarity (COS) measure learned the similarity for categorical data [21]. Based on COS, Couple Metric Similarity (CMS) was proposed recently as a similarity metric [22]. However, in these works, only the pairwise relationship is considered, which neglected more possible complex relationships. Furthermore, they only considered the intra-coupling relationship when both categorical values co-occur with the same values of another attribute. The absolute rule may lose the relationship when both categorical values co-occur with similar categorical values.

Iterated Contextual Distances (ICD) [23] is an iterative algorithm to calculate the similarity for attributes, subrelations, and raws based on the 0-1 information table. However, it only calculates the attribute similarity instead of detailed attribute value similarity. Thus, we can't directly apply it in categorical unsupervised learning. Drawing on the graph operation, SimRank is a traditional similarity measure on the graph which iteratively calculates the similarity scores between nodes and can converge to a stable state fastly. But it's only for the homogeneous network which constructs graph composed of only one type of nodes (objects) and only focuses on object-to-object relationship, which is a structure-based method. P-Rank [25] and C-Rank [26] are variations of SimRank. Compared with SimRank, P-Rank enriches SimRank by jointly encoding both in- and out-link relationships into structural similarity computation. C-Rank is specific for measuring the similarity of two papers, which uses both in-link and out-link by disregarding the direction of references. However, in our work, we construct an undirected heterogeneous graph composed of both objects and attributes. Each object is connected to the equal number of attribute values. Thus, there is no problem with the dealing of in-link and out-link information.

Similarity computation in heterogeneous networks between objects developed recently. Several similarity measures have been proposed in heterogeneous networks recently, which can be divided into two broad categories: (1) content-based similarity measures that treat each object as a bag of items; and (2) structural-based similarity measures that considered the object-to-object relationship in terms of links. [27] proposed HeteRank similarity measure, which fully integrates the multi-type relationships into similarity computation by utilizing all the meetings between objects. [28] proposed another measure, AvgSim, to evaluate two objects through two random walk processes along

TABLE 2. The most similar families across different methods. Each entry represents the most similar family for the family in each row via the measure listed in the column.

Family	Hamming	OF	Lin	ALGO	CMS	HGS
family1	2	2	2	2	2	2
family2	1	1	1	1	1	1
family3	1,2,4	4	1	2	4	2
family4	6	3	6	6	6	6
family5	6	6	6	6	6	6
family6	4,5	5	5	5	5	5

the given meta-path and the reverse meta-path. The above methods are structural-based methods. In order to balance structure and content, [29] proposed a graph clustering method which measures the similarity between objects by balancing structural similarity and attribute similarity. In [30], SimCC method was proposed to measure the similarity for scientific papers based on both content and citations. [31] present a similarity measure NetSim for x-star network schema, which first constructs attribute graph to calculate attribute similarity, then calculated the similarity of centers. The works that combined both structural and content information in the calculation of similarity have shown their superiority to the link only based methods.

Therefore, inspired by the iterative idea and the combination of both structure and content similarity, we proposed a heterogeneous graph-based similarity measure for the unsupervised learning on categorical data. There are two types of nodes and two types of edges in the graph to extract the information from the dataset. Compared with the methods above, our method not only considers the relationship between objects and attributes, but we also take the co-occurrence relationship between attribute values into consideration. This operation can capture more subtle dependence between attribute values. Meanwhile, we use the iterative process to balance the attribute similarity and structural similarity, which can distinguish the difference between objects more effectively.

To show the difference of methods directly, we calculate the similarity scores of data in Table 1 and find their similar objects via the methods mentioned before including Hamming, OF, Lin, ALGO, CMS, and the HGS method proposed in this work. As the result shown in Table 2, Hamming can't precisely distinguish objects. For instance, concerning family 6, Hamming considers family 4 and 5 are the most similar to it, while other methods consider family 5 is more similar to it. The largest difference lies in family 3. The Hamming method considers the most similar families with it are family 1, 2, and 4 due to their equal number of matching attributes, while Lin method regards family 1 as the most similar one with it. Both OF and CMS believes family 4 is the most similar to family 3 because their attribute values share a similar frequency, while both ALGO and HGS methods consider the most similar one is family 2.

III. PROBLEM FORMULATION

In this section, we formulate the problem focused in this work and present a canonical description. Then, an overall framework of the algorithm is given as a good guidance for our work.

A. PROBLEM STATEMENT

Given a dataset composed of a number of objects observed by several categorical attributes, we can first organize it as an information table $I = \{O, A, U\}$, where $O = \{o_1, \ldots, o_n\}$ represents n objects, $A = \{a_1, \ldots, a_m\}$ represents m attributes and $U = \bigcup_{j=1}^{m} U_j$ represents all attribute values, in which $dom(U_j) = \{u_{j,1}, \ldots, u_{j,r_j}\}$, where $|dom(U_j)| = r_j$ is the number of values for attribute a_j. Obviously, the total number of attribute values is finite, which is $\sum_{j=1}^{m} r_j = |U|$. The objective of this work is to obtain the similarity matrix $S = [s_{ij}]_{n\times n}$ between corresponding object pairs, where s_{ij} is the similarity between i^{th} and j^{th} object. In unsupervised learning, the calculation of similarity between object pairs is totally based on their values difference on various attributes.

B. LEARNING FRAMEWORK

Recall the basic recursive assumption illustrated above is "two objects are similar if their corresponding attribute values are similar" and "attribute values are similar if their connected objects are similar". In this section, we propose the heterogeneous graph-based categorical data similarity measure that captures the structure context across objects and attribute values. The learning framework is shown in Fig. 1.

As the figure is shown, a heterogeneous graph structure G is constructed consisting of two types of nodes and two types of edges. Both the object and all attribute values are represented as nodes. One type of edge is connected between attribute value nodes, and the other is added between object nodes and attribute value nodes. Furthermore, a node-pair graph $\hat{G}$ was derived from G. After the graph construction, the score of each attribute-value pair is initialized and iteratively calculated based on the three rules by employing the structure information from $\hat{G}$. Until convergence, both object-pair and attribute-value pair similarity values can be obtained. In the following subsections, more detailed procedures about graph model construction and similarity calculation will be illustrated.

IV. HGS: HETEROGENEOUS GRAPH BASED CATEGORICAL DATA SIMILARITY MEASURE

In this section, we illustrate the detailed learning framework of HGS, graph model construction method and the iterative calculation process of the similarity score for object pairs and attribute-value pairs based on three basic rules.

A. HETEROGENEOUS GRAPH CONSTRUCTION

The dataset D was modeled as a heterogeneous undirected weighted graph $G = (V, E, W)$, where $V = V_1 \bigcup V_2$

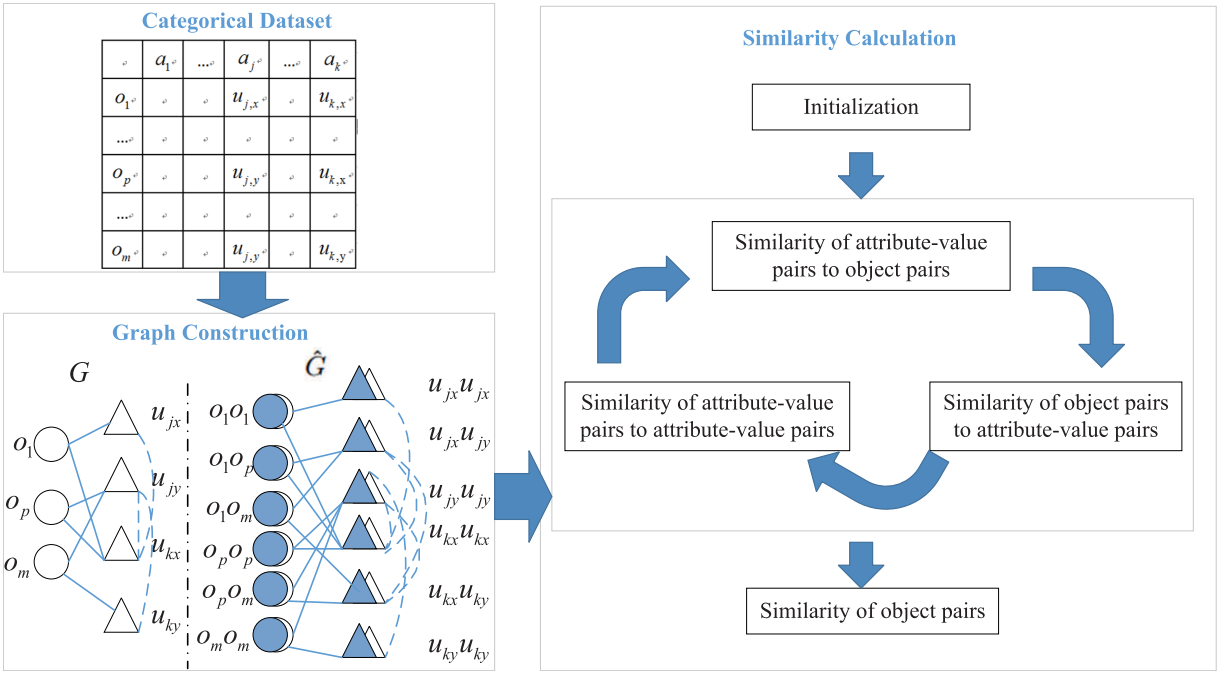

FIGURE 1. Learning framework of HGS similarity measure. A graph G and its derived node-pairs graph $\hat{G}$ were constructed from categorical dataset. Then the similarity of object pairs and attribute-value pairs were iteratively calculated based on three rules.

is the set of nodes, $E = E_1 \bigcup E_2$ represents the set of edges and $W = W_1 \bigcup W_2$ represents weights on the edges. The graph is a heterogeneous graph composed of two types of nodes and two types of edges. More precisely, nodes in $V_1 = \{v_1^1, \ldots, v_1^n\}$ and $V_2 = \bigcup_{j=1}^{m} V_2^j$ represent objects in O and attribute values in U, respectively, where $V_2^j = \{v_2^{j,1}, \ldots, v_2^{j,r_j}\}$ are the set of all values of j^{th} attribute U_j. Edges in E_1 connect nodes between V_1 and V_2 when the object in V_1 present the attribute values in V_2. While for E_2, its edges connect nodes between V_2 when both attribute values occur in the same object. In other words, if both nodes in V_2 are connected to the same node in V_1, there will be an edge between them in E_2. W_1 represents the weight of edges in E_1, but it is an adjacent matrix, where the entry equals to 1 with the existence of a relation, otherwise, 0. Because each object can be connected to an attribute value one time. On the contrary, W_2 is the set of weights on edges in E_2 representing the co-occurrence number of both attribute values in the same object. For instance, the weight between the attribute "usual" of parents' occupation and the attribute "less_conv" house condition is 2, because they co-occur in both family 1 and family 2.

Besides, we have ignored the node pair between various attributes because they are not meaningful comparison. For example, we can't compare the car's price "high" with its door number "2". So these node pairs are deleted during the graph construction step, which can save more calculating space. For any node $v_1^x, x = 1, \ldots, n$ in V_1, we denote all its neighbour set as $I(v_1^x)$ and the single node is denoted as $I_k(v_1^x)$. While for nodes in V_2, they are connected to nodes in both V_1 and V_2. Thus, for any node $v_2^{j,x}, x = 1, \ldots, r_j$ in V_2, denote its neighbour set in V_1 and V_2 as $I(v_2^{j,x})$ and $N(v_2^{j,x}) = \bigcup_{i=1}^{m} N_i(v_2^{j,x})$, in which i means this attribute value object belongs to i^{th} attribute. Similarly, the single object of $I(v_2^{j,x})$ is denoted as $I_k(v_2^{j,x})$. While for $N(v_2^{j,x})$, its single object is denoted as $N_{i,k}(v_2^{j,x})$. In order to clarify clearly, detailed notations are listed in Table 3.

TABLE 3. List of detailed notations.

Notation	Explanation
$O = o_1, ..., o_n$	The set of n objects
$A = a_1, ..., a_m$	The set of m attributes
$U = \bigcup_{j=1}^{m} U_j$	The set of all attribute values
$U_j = u_{j,1}, ..., u_{j,r_j}$	The set of all values of a_j
$r_j = \lvert U_j \rvert$	The number of values of a_j attribute
$S = [s_{ij}]_{n \times n}$	The set of similarity values between objects
$V = V_1 \bigcup V_2$	The node set derived from $O \bigcup U$
$V_1 = v_1^1, ..., v_1^n$	The node set derived from O
v_1^x, v_1^y	Specific node in V^1
$V_2 = \bigcup_{j=1}^{m} V_2^j$	The node set derived from U
$V_2^j = v_2^{j,1}, ..., v_2^{j,r_j}$	The node set derived from U_j
$v_2^{j,x}, v_2^{j,y}$	Specific node in V_2^j
$I(v_1^x)$	The neighbour node set of v_1^x
$I(v_2^{j,x})$	The neighbour node set in V_1 of $v_2^{j,x}$
$N(v_2^{j,x}) = \bigcup_{i=1}^{m} N_i(v_2^{j,x})$	The neighbor node set in V_2 of $v_2^{j,x}$
$N_i(v_2^{j,x})$	The neighbor node set of $v_2^{j,x}$ in V_2 that belongs to i^{th} attribute

In order to further formulate the calculation, a node-pair graph $\hat{G} = (\hat{V}, \hat{E}, \hat{W})$ was constructed on the base of the graph G above. Precisely, $\hat{V} = \hat{V}_1 \bigcup \hat{V}_2$, in which nodes in $\hat{V}_1 = \{(v_1^p, v_1^q), p \leq q, p, q = 1, \ldots, n\}$ represent the ordered node pairs within V_1 of G. The ordered pair means the node pair (a, b) and (b, a) are only represented as a single node (a, b). Similar as $\hat{V}_1$, $\hat{V}_2$ is derived from the node pairs in V_2 of G. The difference is, $\hat{V}_2$ represents the node pairs within the same attribute. To be specific, $\hat{V}_2 = \bigcup_{j=1}^{k} \hat{V}_2^j$, where $\hat{V}_2^j = \{(v_2^{j,p}, v_2^{j,q}), p, q = 1, \ldots, n_j\}$. As for the edges, $\hat{E} = \{\hat{E}_1 \bigcup \hat{E}_2\}$, where edges in $\hat{E}_1$ connect nodes between $\hat{V}_1$ and $\hat{V}_2$, while edges in $\hat{E}_2$ connect nodes within $\hat{V}_2$. Furthermore, $\hat{W} = \hat{W}_1 \bigcup \hat{W}_2$ represent the weights on $\hat{E}_1$ and $\hat{E}_2$, respectively. Since W_1^1 is an adjacent matrix, its derived matrix $\hat{W}_1$ is also an adjacent matrix, of which the entry is 1 when relation exists between node pairs and attribute values pairs. On the contrary, $\hat{W}_2$ is more complex. Without loss of generality, we use $< ., . >$ and $w(., .)$ represent an edge and its weight, respectively. $< (o_1, o_2), (a_1, a_2) >$ exists in $\hat{E}_1$ when both the edges $< o_1, a_1 >$ and $< o_2, a_2 >$ or $< o_2, a_1 >$ and $< o_1, a_2 >$ exist in E_1. While for $\hat{E}_2$, the edge $< (a_1, b_1), (a_2, b_2) >$ means the co-existence of both edges $< a_1, a_2 >$ and $< b_1, b_2 >$ in E_2. In reference to [22], the weight of $< (a_1, b_1), (a_2, b_2) >$ can be derived from both the weights of $< a_1, a_2 >$ and $< b_1, b_2 >$, which can be calculated by

$$\hat{w}_2((a_1, a_2), (b_1, b_2)) = \frac{max(w_2(a_1, b_1), w_2(a_2, b_2))}{2 * max(w_2(a_1, b_1), w_2(a_2, b_2)) - min(w_2(a_1, b_1), w_2(a_2, b_2))} \quad (1)$$

Obviously, when $w_2(a_1, b_1) = w_2(a_2, b_2)$, the weight $\hat{w}_2((a_1, a_2), (b_1, b_2)) = 1$, whereas, the weight is less than 1 when the weights on both edges are different. The larger difference they are, the smaller weight of their derived edges.

In order to clarify our model more clearly, a simple example is shown in Table 4. Three objects o_1, o_2 and o_3, are observed by two attributes A and B, where $dom(A) = \{a_1, a_2\}$, $dom(B) = \{b_1, b_2\}$. More precisely, the attribute values of o_1 are a_1 and b_1, respectively, while o_2 has attribute values a_2 and b_1. Then the graph G was constructed in Fig. 2. As seen in the graph, three objects and four attribute values are represented as nodes. while for edges, precisely, for o_1, the edges were inserted between o_1 and a_1, o_1 and b_1, while for o_2, edges were inserted between o_2 and a_2, o_2 and b_1. Furthermore, edges were inserted between attribute values. Because both a_1 and b_1 were synchronously connected to o_1, and a_2 and b_1 were synchronously connected to o_2, so a_1 and b_1, a_2 and b_1 were also connected, respectively.

In the bottom of Fig. 2, $\hat{G}$ is derived from G, in which six ordered object pairs and six ordered attribute value pairs are represented as nodes. While for edges, o_1o_2 was connected to a_1a_2 as the existence of both edges (o_1, a_1) and (o_2, a_2) in G. For the edge between attribute value pairs, a_1a_1 and b_1b_1 were connected because both of them were connected to node o_1o_1.

TABLE 4. A tiny example.

Object	A	B
o_1	a_1	b_1
o_2	a_2	b_2
o_3	a_2	b_2

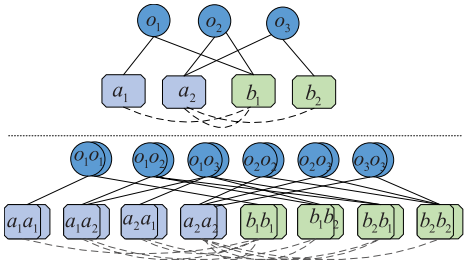

FIGURE 2. A simple example of constructed graph G and the derived node-pairs graph $\hat{G}$.

B. HGS SIMILARITY SCORE CALCULATION

Based on the derived node-pair graph $\hat{G}$, in this subsection, we can formulate the calculation process of the similarity of object pairs. First, the similarity scores of attribute-value pairs are initialized. Second, three rules are employed to iteratively calculate the scores until the scores converge or reach the iterative maximum count. Similarity can be seen as a "flowing" that propagates in $\hat{G}$ from node to node, thus the similarity scores are mutually reinforced. Finally, the scores of object node pairs are what we want. Denote the score of each node pair (a, b) at r^{th} iteration as $s(r, (a, b))$. According to SimRank, when $r \to \infty$, $s(r, (a, b))$ will converge to $s(a, b)$. The detailed proof is provided in subsection V-B. And in the practical instance, the speed of converging is very fast.

1) SIMILARITY SCORE INITIALIZATION

Before iterative calculation, the score of each attribute-value pairs should be first given. As the base case, an object or attribute value is maximally similar to itself [22]. Thus, during each r^{th} iteration, for any object pair $(v_1^x, v_1^y), x \leq y$, when $x = y$, $s(r, (v_1^x, v_1^y))$ is set as 1. Similarly, for any attribute-value pair $(v_2^{j,x}, v_2^{j,y}), x \leq y, j = 1, \ldots, r_j$, when $x = y$, $s(r, (v_2^{j,x}, v_2^{j,y})) = 1$. They will never be changed during each iteration. Besides, the similarity scores of all attribute-value pairs at first iteration should be given. For any attribute value $v_2^{j,x}$ and $v_2^{j,y}$, $x \neq y$, their Jaccard similarity coefficient of occurrence frequency in dataset was compared [32]. Denote the neighbor set of $v_2^{j,x}$ and $v_2^{j,y}$ in V_1 as $I(v_2^{j,x})$ and $I(v_2^{j,y})$, respectively. Therefore, the similarity score $s(1, (v_2^{j,x}, v_2^{j,y}))$ is initialized

as

$$s(1,(v_2^{j,x},v_2^{j,y}))=\begin{cases}\dfrac{log(p)\cdot log(q)}{log(p\cdot q)+log(p)\cdot log(q)}, & x\neq y\\ 1, & x=y\end{cases} \tag{2}$$

where log denotes the logarithm, $p=|I(v_2^{j,x})|+1$ and $q=|I(v_2^{j,y})|+1$. $I(v_2^{j,x})$ and $I(v_2^{j,y})$ denote the set of object nodes related to attribute value nodes $v_2^{j,x}$ and $v_2^{j,y}$, respectively. Additionally, the weights between the attribute-value pairs should be calculated according to Eq. 1.

2) SIMILARITY SCORE ITERATIVE CALCULATION

After initialization of attribute-value pair similarity scores, we will update the similarity scores according to three rules. The scores will flow from attribute-value pairs to object pairs, then return to attribute-value pairs again. Before they return to object pairs, the scores will flow between attribute-value pairs. In the long run, the similarity scores will flow through the whole network and finally converge. In the following, we formulate three basic rules and explain the detailed calculation of iterative calculation. It should be mentioned that to implement the experiment much faster, we only scan the database one time and save the relations in a sparse matrix. Thus the proceeding of remaining iterations doesn't necessarily access the database again.

(1) Rule 1: Attribute-Value Pair Similarity To Object Pair Similarity

As the rule 1 shows, after the initialization of attribute-value pair similarity scores, we can obtain the object pair similarity via their correlation matrix. The intuition is that similar objects tend to perform similar values on the same attribute. The similarity score of object pair can be determined by the similarity scores of all its related attribute-value pairs. Therefore, for any $x, y=1,\ldots,n$ and $x\neq y$, the object node pair (v_1^x, v_1^y), we update its similarity score in $(r+1)^{th}$ iteration by

$$s(r+1,(v_1^x,v_1^y))=\frac{C_1}{m}\sum_{k=1}^{m}s(r,(I_k(v_1^x),I_k(v_1^y))). \tag{3}$$

where $C_1\in(0,1)$ can be seen as both a memory coefficient that object pair keep the state as its previous iteration and a confidence coefficient from attribute value pair similarity to object pair similarity. $I(v_1^x)$ denotes the set of neighbours in V_2 of v_1^x and $I_k(v_1^x)$ denotes the single attribute value node. Their similarity scores in current iteration will partially derive from their own scores in previous iteration and partially from their neighbours. Because each object is related to m attributes and only the values within attribute are compared, both $|I(v_1^x)|$ and $|I(v_1^y)|$ are equal to m, so the number of neighbours of object pair (v_1^x, v_1^y) is also m.

(2) Rule 2: Object Pair Similarity To Attribute Value Pair Similarity

Similar as rule 1, similar attribute values may co-occur in similar objects. Following the above rule, for any $j=1,2\ldots,m$, $x,y=1,\ldots,r_j$ and $x\neq y$, we update the similar scores of attribute-value pair $(v_2^{j,x}, v_2^{j,y})$ by

$$s(r+1,(v_2^{j,x},v_2^{j,y}))=\frac{C_2}{|I(v_2^{j,x})||I(v_2^{j,y})|}\sum_{k_1=1}^{|I(v_2^{j,x})|}\sum_{k_2=1}^{|I(v_2^{j,y})|}s(r,(I_{k_1}(v_2^{j,x}),I_{k_2}(v_2^{j,y}))) \tag{4}$$

Similar as $C_1, C_2\in(0,1)$ can be considered both a memory coefficient that attribute value pair keeps its previous state and a confidence coefficient from object similarity to attribute value similarity, and $I_{k_1}(v_2^{j,x}), I_{k_2}(v_2^{j,y})$ represent single node in $I(v_2^{j,x})$ and $I(v_2^{j,y})$, respectively.

(3) Rule 3: Attribute-Value Pair Similarity To Attribute Value Pair Similarity

Owing to the interdependence, the occurrence of one attribute value may depend on that of another attribute value. For instance, a person's education degree may correlate with his salary. Commonly, a higher education degree may correspond to a higher salary. Compared to a bachelor degree, a philosophy degree (Ph.D.) may be more similar to a master degree. Thus, the salary similarity between persons of Ph.D. and master degree may be higher than that between persons of Ph.D. and bachelor degree. Therefore, we think the similarity score of an attribute-value pair can "flow" to the pairs of its neighbours. Intuitively, for any $j=1,2\ldots,m$, $x,y=1,\ldots,r_j$ and $x\neq y$, we re-update the similarity score of attribute-value pair $(v_2^{j,x}, v_2^{j,y})$ from all its attribute value pair neighbours:

$$\begin{aligned}&s(r+1,(v_2^{j,x},v_2^{j,y}))\\&=\frac{1}{\sum_{i=1}^{m}\sum_{k_1=1}^{|N_i(v_2^{j,x})|}\sum_{k_2=1}^{|N_i(v_2^{j,y})|}\omega_{k_1k_2}^{i}}\\&\quad\sum_{i=1}^{m}\sum_{k_1=1}^{|N_i(v_2^{j,x})|}\sum_{k_2=1}^{|N_i(v_2^{j,y})|}\omega_{k_1k_2}^{i}*s(r,(N_{i,k_1}(v_2^{j,x}),N_{i,k_2}(v_2^{j,y})))\end{aligned} \tag{5}$$

where $\omega_{k_1k_2}^{i}$ is the short of $\hat{w}_2((v_2^{j,x}, v_2^{j,y}), (N_{i,k_1}(v_2^{j,x}), N_{i,k_2}(v_2^{j,y})))$ which represents the weight of the edge from attribute-value pair $(v_2^{j,x}, v_2^{j,y})$ to its single neighbour attribute-value pair $(N_{i,k_1}(v_2^{j,x}), N_{i,k_2}(v_2^{j,y}))$ that belongs to i^{th} attribute.

The HGS algorithm can be summarized as **algorithm 1**.

V. THEORETICAL ANALYSIS

In this section, we prove the metric properties of HGS, including positivity, reflexivity, commutativity, and triangle inequality. Besides, the convergence proof of the solution is given in the second part. Finally, the computational complexity of HGS is analyzed.

Algorithm 1 HGS Similarity Measure Algorithm for the Unsupervised Learning on Categorical Data

Input: $I = \{O, A, U\}$: Information table, where
O: n objects
A: m attributes
U: the combination of all attribute values.
Output: $S = [s_{ij}]_{n\times n}$: the similarity matrix between object pairs.

Graph Construction
Heterogeneous graph $G = (V, E, W)$ construction
Node-pair graph $\hat{G} = (\hat{V}, \hat{E}, \hat{W})$ derived from G
Save graph structure in adjacent matrix W_1 and W_2
W_1: the adjacent matrix between object node and attribute-value node;
W_2: the adjacent matrix between attribute-value and attribute-value node;
Calculate the weight matrix for node-pair graph. $\hat{W}_1$: the adjacent matrix between object pair and attribute-value pair node;
$\hat{W}_2$: calculate the adjacent matrix between attribute-value and attribute-value node according to Eq.1;
Initizalze
$maxIter = 10$; $C_1 = 0.8$; $C_2 = 0.8$;
$s(r, (v_1^x, v_1^y)) = 1$ for any r when $x = y$;
$s(r, (v_2^{j,x}, v_2^{j,y})) = 1$ for any r and j when $x = y$;
$s(1, (v_2^{j,x}, v_2^{j,y})) = \frac{log(p)\cdot log(q)}{log(p\cdot q)+log(p)\cdot log(q)}$ when $x \neq y$;
for $r \in [1, MaxIter]$ **do**
  for $x \in [1, n]$ **do**
    for $y \in [i, n]$ **do**
      Calculate $s(r, (v_1^x, v_1^y))$ according to Eq. 3;
    end
  end
  for $i \in [1, m]$ **do**
    for $j \in [1, r_i]$ **do**
      for $k \in [1, r_i]$ **do**
        Update $s(r+1, (v_2^{i,j}, v_2^{i,k}))$ according to Eq. 4
      end
    end
  end
  for $i \in [1, m]$ **do**
    for $j \in [1, r_i]$ **do**
      for $k \in [1, r_i]$ **do**
        Update $s(r+1, (v_2^{i,j}, v_2^{i,k}))$ according to Eq. 5
      end
    end
  end
end

A. HGS METRIC VALIDATION

Given $s(v_1^x, v_1^y)$, which represents the similarity score between object nodes v_1^x and v_2^y obtained by HGS, in the following section, we prove it possesses the properties as follows.

1) Positivity: $0 < s(v_1^x, v_1^y) \leq 1$.

Accordingly, if $v_1^x \neq v_1^y$, $s(v_1^x, v_1^y)$ is the average of similarity scores of its related attribute-value pairs in $\hat{G}$ during each iteration, otherwise, $s(v_1^x, v_1^y) = 1$. In the first iteration, the similarity score of attribute value pairs was initialized by Eq. 2, accordingly $s(v_2^{j,x}, v_2^{j,y}) \in (0, 1]$. So after all iteration, the similarity score has flowed between attribute-value and object pairs by average operation. Therefore, as the average similarity scores, $s(v_1^x, v_1^y)$ also satisfies positivity property.

2) Reflexivity: $s(v_1^x, v_1^y) = 1 \Leftrightarrow v_1^x = v_1^y$.

First, we prove $v_1^x = v_1^y \Rightarrow s(v_1^x, v_1^y) = 1$. When $v_1^x = v_1^y$, for node pairs that consist of identical objects, their similarity in every iteration $s(r, (v_1^x, v_1^y))$ has been set as 1 and was never changed in all iterations. So $s(v_1^x, v_1^y) = 1$. Further, we prove the sufficiency $s(v_1^x, v_1^y) = 1 \Rightarrow v_1^x = v_1^y$. If $v_1^x \neq v_1^y$, they are connected to at least one attribute-value pair consisting of different values. However, according to Eq.2, the attribute value is most similar to itself. Thus, only the attribute-value pairs consisting of the same values can have a score of 1. Otherwise, the attribute-value pairs will be less than 1. According to Eq.3 and Eq.4, only when $v_1^x = v_1^y$, their connected attribute-value pair consist of the same values. Besides, as the existence of confidence coefficients $C_1 \in (0, 1)$ and $C_2 \in (0, 1)$, $s(r, (v_1^x, v_1^y)) < 1$ even all the attribute-value pairs that they are connected to are equal to 1. Thus, in the long run, $s(r, (v_1^x, v_1^y))$ can't possibly converge to $s(v_1^x, v_1^y) = 1$. So, if $s(v_1^x, v_1^y) = 1$, then $v_1^x = v_1^y$.

3) Commutativity: $s(v_1^x, v_1^y) = s(v_1^y, v_1^x)$.

In our iterative calculation, the object pairs v_1^x, v_1^y and v_1^y, v_1^x have been combined as one object pair v_1^x, v_1^y. Thus, $s(v_1^x, v_1^y) = s(v_1^y, v_1^x)$.

4) Triangle Inequality: $\frac{1}{s(v_1^x, v_1^z)} + \frac{1}{s(v_1^z, v_1^y)} \geq 1 + \frac{1}{s(v_1^x, v_1^y)}$

Denote all the attribute-value nodes that are related to v_1^x, v_1^y and v_1^z as $I(v_1^x)$, $I(v_1^y)$ and $I(v_1^z)$, respectively, in which the single node is denoted as $I_k(v_1^x)$, $I_k(v_1^y)$ and $I_k(v_1^z)$. Therefore, according to Eq. 3,

$$\begin{aligned}&\frac{1}{s(v_1^x, v_1^z)} + \frac{1}{s(v_1^z, v_1^y)} - \frac{1}{s(v_1^x, v_1^y)} - 1\\ &= \frac{m}{C_1}\Big(\frac{1}{\sum_{k=1}^{m} s((I_k(v_1^x), I_k(v_1^z)))} + \frac{1}{\sum_{k=1}^{m} s((I_k(v_1^z), I_k(v_1^y)))}\\ &\quad - \frac{1}{\sum_{k=1}^{m} s((I_k(v_1^x), I_k(v_1^y)))}\Big) - 1 \end{aligned} \tag{6}$$

Because $C_1 < 1$, hence the HGS similarity metric satisfies triangle inequality when the following equation holds.

$$\begin{aligned}&\frac{1}{\frac{1}{m}\sum_{k=1}^{m} s((I_k(v_1^x), I_k(v_1^z)))} + \frac{1}{\frac{1}{m}\sum_{k=1}^{m} s((I_k(v_1^z), I_k(v_1^y)))}\\ &\quad - \frac{1}{\frac{1}{m}\sum_{k=1}^{m} s((I_k(v_1^x), I_k(v_1^y)))} \geq C_1 \end{aligned} \tag{7}$$

Thus, if for any $k = 1, \ldots, m$, the following equation holds, so does the Eq. 7.

$$\frac{1}{s((I_k(v_1^x), I_k(v_1^z)))} + \frac{1}{s((I_k(v_1^z), I_k(v_1^y)))} - \frac{1}{s((I_k(v_1^x), I_k(v_1^y)))} \geq 1 \quad (8)$$

The similarity scores of their pairs are determined in the last iteration. According to Eq. 5, the scores could be changed by average operation, which however doesn't change the triangle inequality property.

All the possible cases are considered. We denote $|I_k(v_1^x)| + 1$, $|I_k(v_1^y)| + 1$, and $|I_k(v_1^z)| + 1$ as x, y and z, respectively. As $|I_k(v_1^x)|, |I_k(v_1^y)|, |I_k(v_1^z)| = 1$, so $x, y, z = 2$, and $logx, logy, logz \geq 0$.

(1) when $I_k(v_1^x) = I_k(v_1^y) = I_k(v_1^z)$, according to Eq. 2, $s((I_k(v_1^x), I_k(v_1^z)) = s((I_k(v_1^y), I_k(v_1^z)) = s((I_k(v_1^x), I_k(v_1^y)) = 1$, hence

$$\frac{1}{s((I_k(v_1^x), I_k(v_1^z)))} + \frac{1}{s((I_k(v_1^z), I_k(v_1^y)))} = 1 + \frac{1}{s((I_k(v_1^x), I_k(v_1^y)))} \quad (9)$$

(2) when $I_k(v_1^x) = I_k(v_1^y)$ or $I_k(v_1^y) = I_k(v_1^z)$ or $I_k(v_1^x) = I_k(v_1^z)$:

When $I_k(v_1^x) = I_k(v_1^y)$, $s((I_k(v_1^x), I_k(v_1^y) = 1$, so

$$\begin{aligned}
&\frac{1}{s((I_k(v_1^x), I_k(v_1^z)))} + \frac{1}{s((I_k(v_1^z), I_k(v_1^y)))} - \frac{1}{s((I_k(v_1^x), I_k(v_1^y)))} - 1 \\
&= \frac{log(x \cdot z) + logx \cdot logz}{logx \cdot logz} + \frac{log(z \cdot y) + logz \cdot logy}{logz \cdot logy} - 2 \\
&= \frac{log(x \cdot z)}{logx \cdot logz} + \frac{log(z \cdot y)}{logz \cdot logy} > 0
\end{aligned} \quad (10)$$

When $I_k(v_1^y) = I_k(v_1^z)$, $s((I_k(v_1^y), I_k(v_1^z)) = 1$, $y = z$, so

$$\begin{aligned}
&\frac{1}{s((I_k(v_1^x), I_k(v_1^z)))} + \frac{1}{s((I_k(v_1^z), I_k(v_1^y)))} - \frac{1}{s((I_k(v_1^x), I_k(v_1^y)))} - 1 \\
&= \frac{log(x \cdot z) + logx \cdot logz}{logx \cdot logz} - \frac{log(x \cdot y) + logx \cdot logy}{logx \cdot logy} \\
&= \frac{log(x \cdot z)}{logx \cdot logz} - \frac{log(x \cdot y)}{logx \cdot logy} \\
&= \frac{log(x \cdot z) \cdot logy - log(x \cdot y) \cdot logz}{logx \cdot logy \cdot logz} \\
&= \frac{logy - logz}{logy \cdot logz} = 0
\end{aligned} \quad (11)$$

When $I_k(v_1^x) = I_k(v_1^z)$, the case is similar as when $I_k(v_1^y) = I_k(v_1^z)$.

(3)when $I_k(v_1^x) \neq I_k(v_1^y) \neq I_k(v_1^z)$

$$\begin{aligned}
&\frac{1}{s((I_k(v_1^x), I_k(v_1^z)))} + \frac{1}{s((I_k(v_1^z), I_k(v_1^y)))} - \frac{1}{s((I_k(v_1^x), I_k(v_1^y)))} - 1 \\
&= \frac{log(x \cdot z) + logx \cdot logz}{log(x \cdot z)} + \frac{log(z \cdot y) + logz \cdot logy}{log(z \cdot y)} - \frac{log(x \cdot y) + logx \cdot logy}{log(x \cdot y)} - 1 \\
&= \frac{logx \cdot logz}{log(x \cdot z)} + \frac{logz \cdot logy}{log(z \cdot y)} - \frac{logx \cdot logy}{log(x \cdot y)} \\
&= \frac{2}{logz} > 0
\end{aligned} \quad (12)$$

Therefore, the following equation holds.

$$\frac{1}{s((I_k(v_1^x), I_k(v_1^z)))} + \frac{1}{s((I_k(v_1^z), I_k(v_1^y)))} \geq 1 + \frac{1}{s((I_k(v_1^x), I_k(v_1^y)))} \quad (13)$$

The above proof of triangle inequality is in reference to the proof provided by paper [22], [37].

B. CONVERGENCE PROOF OF HGS

In this part, we prove the uniqueness of the converged solution to the HGS similarity measure for objects and attribute-values. The similarity scores between objects are iteratively calculated from Eq. 3 and that between attribute values are iteratively calculated from Eq. 4 and 5. For every object pair $(v_1^x, v_1^y) \in \hat{V}_1$, suppose $s_1(v_1^x, v_1^y)$ and $s_2(v_1^x, v_1^y)$ are the two solutions obtained from HGS measure. For all object pairs (v_1^x, v_1^y), let $\sigma_1(v_1^x, v_1^y) = s_1(v_1^x, v_1^y) - s_2(v_1^x, v_1^y)$. Similarly, for any attribute-value pair $(v_2^{j,x}, v_2^{j,y}) \in \hat{V}_2$, let $s_1(v_2^{j,x}, v_2^{j,y})$ and $s_2(v_2^{j,x}, v_2^{j,y})$ are the two solutions and $\sigma_2(v_2^{j,x}, v_2^{j,y}) = s_1(v_1^x, v_1^y) - s_2(v_1^x, v_1^y)$ represents the difference for all attribute-value pairs. $M_1 = \max_{v_1^x, v_1^y} |\sigma_1(v_1^x, v_1^y)|$ and $M_2 = \max_{v_2^{j,x}, v_2^{j,y}} |\sigma_2(v_2^{j,x}, v_2^{j,y})|$ be the maximum absolute values of any difference, respectively. If the solution is unique, there should be $s_1(v_1^x, v_1^y) = s_2(v_1^x, v_1^y)$ for all object pairs (v_1^x, v_1^y) and $s_1(v_2^{j,x}, v_2^{j,y}) = s_2(v_2^{j,x}, v_2^{j,y})$. Thus, we want to prove $M_1 = 0$ and $M_2 = 0$.

When $v_1^x = v_1^y$, the similarity score between v_1^x and v_1^y keeps as 1, therefore, $\sigma_1(v_1^x, v_1^y) = 0$, and $M_1 = 0$;

Similarly, when $v_2^{j,x} = v_2^{j,y}$, the similarity score between $v_2^{j,x}$ and $v_2^{j,y}$ keeps as 1, therefore, $\sigma_2(v_1^{j,x}, v_2^{j,y}) = 0$, and $M_2 = 0$;

Otherwise,

$$\begin{aligned}
&\sigma_1(v_1^x, v_1^y) \\
&= s_1(v_1^x, v_1^y) - s_2(v_1^x, v_1^y) \\
&= \frac{C_1}{m} \sum_{k=1}^{m} (s_1((I_k(v_1^x), I_k(v_1^y))) - s_2((I_k(v_1^x), I_k(v_1^y))) \\
&= \frac{C_1}{m} \sum_{k=1}^{m} \sigma_2(I_k(v_1^x), I_k(v_1^y))
\end{aligned} \quad (14)$$

$\sigma_2(v_2^{j,x}, v_2^{j,y})$

$$
\begin{aligned}
&= s_1(v_2^{j,x}, v_2^{j,y}) - s_2(v_2^{j,x}, v_2^{j,y}) \\
&= \frac{C_2}{|I(v_2^{j,x})||I(v_2^{j,y})|} \sum_{k_1=1}^{|I(v_2^{j,x})|} \sum_{k_2=1}^{|I(v_2^{j,y})|} \\
&\quad \{s_1(I_{k_1}(v_2^{j,x}), I_{k_2}(v_2^{j,y})) - s_2(I_{k_1}(v_2^{j,x}), I_{k_2}(v_2^{j,y}))\} \\
&= \frac{C_2}{|I(v_2^{j,x})||I(v_2^{j,y})|} \sum_{k_1=1}^{|I(v_2^{j,x})|} \sum_{k_2=1}^{|I(v_2^{j,y})|} \sigma_1(I_{k_1}(v_2^{j,x}), I_{k_2}(v_2^{j,y}))
\end{aligned} \tag{15}
$$

Let $|\sigma_1(v_1^x, v_1^y)| = M_1$ for some object pairs $(v_1^x, v_1^y) \in \hat{V}_1$ and $\sigma_2(v_2^{j,x}, v_2^{j,y}) = M_2$ for some attribute-value pairs $(v_2^{j,x}, v_2^{j,y})$. Thus, we have

$$
\begin{aligned}
M_1 &= |\sigma_1(v_1^x, v_1^y)| \\
&= \left| \frac{C_1}{m} \sum_{k=1}^{m} \sigma_2(I_k(v_1^x), I_k(v_1^y)) \right| \\
&\le \frac{C_1}{m} \sum_{k=1}^{m} \left| \sigma_2(I_k(v_1^x), I_k(v_1^y)) \right| \\
&\le \frac{C_1}{m} \sum_{k=1}^{m} M_2 \\
&= \frac{C_1}{m} m * M_2 \\
&= C_1 M_2
\end{aligned} \tag{16}
$$

and

$$
\begin{aligned}
M_2 &= |\sigma_2(v_1^{j,x}, v_1^{j,y})| \\
&= \left| \frac{C_2}{|I(v_2^{j,x})||I(v_2^{j,y})|} \sum_{k_1=1}^{|I(v_2^{j,x})|} \sum_{k_2=1}^{|I(v_2^{j,y})|} \sigma_1(I_{k_1}(v_2^{j,x}), I_{k_2}(v_2^{j,y})) \right| \\
&\le \frac{C_2}{|I(v_2^{j,x})||I(v_2^{j,y})|} \sum_{k_1=1}^{|I(v_2^{j,x})|} \sum_{k_2=1}^{|I(v_2^{j,y})|} \left| \sigma_1(I_{k_1}(v_2^{j,x}), I_{k_2}(v_2^{j,y})) \right| \\
&\le \frac{C_2}{|I(v_2^{j,x})||I(v_2^{j,y})|} \sum_{k_1=1}^{|I(v_2^{j,x})|} \sum_{k_2=1}^{|I(v_2^{j,y})|} M_1 \\
&= \frac{C_2}{|I(v_2^{j,x})||I(v_2^{j,y})|} |I(v_2^{j,x})| * |I(v_2^{j,y})| * M_1 \\
&= C_2 M_1
\end{aligned} \tag{17}
$$

Therefore,

$$
\begin{aligned}
M_1 - C_1 M_2 &\le 0 \\
M_2 - C_2 M_1 &\le 0 \\
(1 - C_1) M_2 + (1 - C_2) M_1 &\le 0
\end{aligned} \tag{18}
$$

Since $0 < C_1 < 1$ and $0 < C_2 < 1$, $M_1 \ge 0$ and $M_2 \ge 0$, we can infer that $M_1 = 0$ and $M_2 = 0$. Therefore, both the similarity values of object pairs and attribute-value pairs can converge to unique value, respectively.

C. COMPUTATIONAL COMPLEXITY OF HGS

In HGS measure $s(r, (v_1^x, v_1^y))$ is the average of its related attribute value similarity scores $s(r, (I_k(v_1^x), I_k(v_1^y))), k = 1, \ldots, m$ during iterative process. The higher score $s(r, (v_1^x, v_1^y))$ is, the more similar two objects are.

Accordingly, in the calculation process of HGS, a node-pair graph $\hat{G}$ is constructed to formulate the calculation. The adjacent matrices between object-pairs and attribute-value pairs, between attribute-value pairs and attribute-value pairs need to be obtained. Due to the iterative process, to improve the calculating efficiency, the adjacent matrices need to be saved in calculation space. Hence, the most time and space-consuming is to calculate two adjacent matrices. Suppose the maximal number of distinct values for each attribute is $R = \max_{j=1}^{m}(r_j)$, the number of attributes is m, and the number of object pairs is $\frac{1}{2}n(n-1)$, then the maximum of attribute-value pairs is $\frac{1}{2}(mR(R+1))$. Hence, the time complexity of calculating the first adjacent matrix is $\frac{1}{4}n(n-1)mR(R+1)$, while calculating the later adjacent matrix is $\frac{1}{4}m^2R^2(R+1)^2$. Thus, the upper complexity limit of HGS is $O(n^2mR^2 + m^2R^4)$. Normally, the number of attributes m is far less than the number of objects, n. Therefore, when there are many objects in the database, during the calculation, the adjacent matrix between object pairs and attribute-value pairs can be initialized as a sparse matrix to save calculation storage and accelerate the calculation speed.

VI. EXPERIMENTS AND DISCUSSIONS

In this section, we conduct the k-modes clustering, spectral clustering, and similarity search experiments based on HGS and five comparable similarity measures on 26 datasets. The detailed dataset information and experiment design are first illustrated. Then the clustering and similarity search experiments are conducted and their results are respectively shown in the following parts. Six extensive discussions are further carried out, which analyze the convergence, parameter sensitivity of HGS, compare HGS with its variants from various graph construction and different similarity score initializations, and explore its application to imbalanced data and the comparison of its time cost with other methods.

A. DATASETS

In order to validate the effectiveness of HGS, 20 public datasets downloaded from UCI are used for the experiments. Each dataset is composed of multiple objects which are described by multiple categorical attributes and the corresponding class information that the object belongs to. Besides, 4 high-dimensional microbiome datasets, "crc_wang" [40], "ibd_morgan" [41], "ibd_papa" [42], and "Ob_ross" [43] are derived from a publicly available database(MicrobiomeHD) [44], which has collected human gut microbial raw data sets including case and control subjects. All the datasets have been processed by the same

16s processing pipeline and the relative abundance of OTUs at the genus level of each subject is obtained. In order to transform microbiome datasets into the categorical dataset, we transform the abundance value that is larger than 0 into 1 representing that the subject has or doesn't have the corresponding genus. Another high-dimensional dataset "Bats of Guyana" is derived from [45], which records the real DNA barcode of bats and the detailed species and subspecies that they belong to. Here we have used this dataset two times by using their species and subspecies as class information, respectively. Therefore, we have 26 datasets in total for experiments.

In Table 5, we presented their detailed information, including the number of objects (O), the number of attributes (A), the total number of all attribute values (V), the number of classes (C) and the imbalanced Ratio (IR) (The detailed calculation is present in Section VI-D.5. Each dataset is composed of multiple objects which are described by multiple categorical attributes and the corresponding class information that the object belongs to. Before the graph construction, the dataset was first processed. Because this work only considers categorical data, the numerical attributes were removed from the datasets. Due to the existence of missing values, the rows with missing values were removed from initial data. Furthermore, to improve the calculation efficiency, attributes with only single value were removed because they don't help distinguish different objects.

TABLE 5. The data information of 26 datasets. Each column refers to the dataset name, object number (O), attribute number (A), the number of total attribute values (C), the number of classes (C), imbalanced ratio of the dataset (IR), and the abbreviation of the set name, respectively.

Dataset	O	A	V	C	IR	Abbr.
Soybeansmall	47	21	58	4	1.70	Sos
Hayesroth	132	4	15	3	1.70	Ha
Hepatitis	143	13	27	2	1.25	He
Breastcancer	683	9	89	2	1.86	Br
Housevotes	232	16	32	2	1.15	Ho
Soybeanlarge	266	35	97	15	4.00	Sol
SPECT	267	22	44	2	3.85	SP
Zoo	101	16	36	7	10.25	Zo
DNApromoter	106	57	228	2	1.00	DN
Lymphography	148	18	59	4	40.50	Ly
Monks1	432	6	17	2	1.00	Mo
Dermatology	366	33	129	6	5.60	De
Crx	671	9	40	2	1.24	Cr
Mammographic masses	835	4	20	2	1.05	Ma
Flare	1066	9	30	6	884.00	Fl
PrimaryTumor	336	15	31	21	28.00	Pr
Tic-tac-toe	958	9	27	2	1.89	Ti
Balance	625	4	20	3	5.88	Ba
Car	1728	6	21	2	18.62	Ca
Chess	3196	36	73	2	1.09	Ch
crc_wang	98	284	568	2	1.23	Cw
ibd_morgan	126	634	1268	2	6.00	Im
ibd_papa	90	2740	5480	2	2.75	Ip
ob_ross	63	1217	2434	2	1.42	Or
Bat_species	840	657	1484	50	140.00	Bs
Bat_subs.	840	657	1484	96	74.00	Bss

B. CLUSTERING EXPERIMENT AND RESULTS

1) BASELINE CLUSTERING METHODS AND SIMILARITY MEASURES

In experiments, a typical similarity-based clustering algorithm, spectral clustering [33], and a distance-based categorical algorithm, k-modes [34], are used for the experiments. To measure the similarity between objects, HGS and other five state-of-the-art similarity/distance measures, including Hamming Distance (Hamming for short), Occurrence Frequency (OF), the information theory based method proposed by Lin. *et al.* (Lin), ALGO_DISTANCE (ALGO) [9], and the Coupled Metric Similarity (CMS) are applied, respectively. It should be noted that, for spectral clustering, we chose the non-normalized Laplacian matrix to find its Eigenvalue as the input of the k-means clustering algorithm. As the performance of spectral clustering algorithm relies on the k-means clustering algorithm, the result from spectral clustering may be varied in different runs. Thus, we repeated the spectral clustering experiments on each dataset for five times and calculate the average performance as the final result. Furthermore, in order to make the result comparable, in k-modes clustering experiment, the initial cluster centers are selected according to rule 1 defined in [35] and used for all measure-based clustering experiments. However, we found the selection of cluster centers didn't change the final result by trying different sets of initial cluster centers. As for the value of K, we decide it by the simple rule described by [data mining book], which is

$$K = \sqrt{\frac{n}{2}},$$

where n is the number of objects in the database.

It should be mentioned that during k-modes clustering process, we find the number of cluster centers may decrease when some special condition occurs. For instance, in first iteration, there are three cluster centers a, b and c. Then we allocate all the objects to these three centers and form three clusters A, B and C. Afterward, we update the cluster centers by finding the modes within each cluster according to the rule defined in [34]. Hereafter, the distance between every object and new centers are recalculated. The extreme condition may occur here when all the objects in cluster C are nearest to a or b. And no object in cluster A and B is the nearest to c. Therefore, in the next iteration, all the objects in cluster C will be re-allocated to cluster A or B, then C will become empty. In this condition, we need to update the number of cluster centers from $K = 3$ to $K = 2$.

In the experiment, HGS, OF, and CMS measure the similarity for objects. In order to apply them in k-modes algorithm, we should derive the distance or dissimilarity measure.

According to the rule provided in [2], for objects o_x and o_y, if their similarity value is $s(o_x, o_y)$, their distance value $d(o_x, o_y)$ can be

$$d(o_x, o_y) = \frac{1}{s(o_x, o_y)} - 1 \tag{19}$$

In addition, two coefficients C_1 and C_2 need to be set in HGS. C_1 in Eq. (3) reflects the memory coefficient from its previous state and the confidence extent from attribute-value pair to object pair similarity. In contrast, C_2 reflects the confidence coefficient from an object pair to attribute value pair similarity. In experiments, both C_1 and C_2 are set as 0.8. We will further explore their effects in the discussion section. As it's fast to converge, the iteration maximum time was set as 10. All the experiments are conducted in MATLAB R2018a by macOS 10.13.6 with 8GB memory.

2) CLUSTERING EVALUATION METHODS

After the experiments, two common-used methods, *F-score* and Normalized Mutual Information (*NMI*) are used to measure the clustering experiment. The larger value indicates a better clustering performance, hence, a better corresponding similarity measure. Furthermore, in order to make our result more convincing, we compute the rand index (RI) and purity index [47] to evaluate the results in the experiment. As the limit of the scope, we finally show them in the supplementary materials. Here we show the calculations of *F-score* and *NMI* as follows.

(1) F_β-score (or *F-measure*)

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}, \tag{20}$$

where $P = \frac{TP}{TP+FP}$ means precision rate, $R = \frac{TP}{TP+FN}$ means recall ratio, in which TP (true positive) denotes the number of objects belonging to the same class are also assigned into the same clusters, TN (true negative) denotes the number of objects belonging to same classes are wrongly assigned into different clusters, FP (false positive) decision assigns two objects belonging to different classes to the same clusters, and FN (false negative) decision assigns two dissimilar objects to different clusters. Here we set $\beta = 1$, i.e. *F-score* in short.

(2) Normalized mutual information (*NMI*)

$$NMI = \frac{\sum_{i=1}^{k}\sum_{j=1}^{c} n_{i,j} log(\frac{n \cdot n_{i,j}}{n_i \cdot n_j})}{\sqrt{(\sum_{i=1}^{k} n_i log\frac{n_i}{n})(\sum_{j=1}^{c} n_j log\frac{n_j}{n})}} \tag{21}$$

where c represents the true number of classes while k is the number of clusters derived from the clustering algorithm, n means the total number of objects in dataset, and $n_{i,j}$ denotes the number of objects belonging to class j which are clustered into cluster i.

TABLE 6. The *NMI* of Hamming, OF, Lin, ALGO, CMS vs. HGS-based spectral clustering.

Data	HM	OF	Lin	ALGO	CMS	HGS
Sos	0.756	0.515	**1.000**	0.814	0.756	0.814
Ha	0.106	0.200	0.065	**0.226**	0.118	0.097
He	0.090	0.075	0.100	0.083	0.084	**0.112**
Br	**0.505**	0.306	0.370	0.455	0.497	0.426
Ho	0.289	0.283	0.277	**0.395**	0.294	0.353
Sol	0.510	0.498	**0.601**	0.469	0.527	0.551
SP	0.071	0.058	0.090	0.088	**0.105**	0.086
Zo	0.870	0.769	0.784	0.849	0.782	**0.875**
DN	0.225	0.285	0.225	0.110	0.194	**0.298**
Ly	0.187	0.211	0.168	**0.216**	0.188	0.177
Mo	0.125	0.132	**0.154**	0.046	0.106	0.088
De	0.623	0.557	**0.695**	0.627	0.641	0.603
Cr	0.169	0.150	0.176	0.073	0.177	**0.192**
Ma	0.212	0.204	0.214	0.207	0.212	**0.217**
Fl	0.075	**0.106**	0.065	0.067	0.081	0.083
Pr	0.310	0.259	**0.422**	0.406	0.252	0.225
Ti	0.221	0.035	0.050	0.044	**0.232**	**0.232**
Ba	0.067	0.079	0.087	0.018	**0.094**	0.086
Ca	0.072	0.063	0.053	0.022	0.058	**0.203**
Ch	0.009	0.017	**0.021**	0.018	0.006	0.008
Cw	0.058	**0.121**	0.069	0.086	0.072	0.104
Im	**0.205**	0.130	0.120	0.071	0.168	0.175
Ip	**0.090**	**0.090**	0.051	0.082	**0.090**	**0.090**
Or	0.103	0.064	0.054	0.034	**0.126**	0.084
Bs	0.293	0.215	**0.294**	0.458	0.178	0.198
Bss	0.293	0.215	0.294	**0.458**	0.178	0.198
Average	0.251	0.217	0.250	0.247	0.239	**0.253**

3) CLUSTERING RESULTS

In this section, the experiment results from spectral clustering and k-modes clustering based on six comparative measures conducted on 26 datasets are shown and discussed.

(1) Comparison of HGS with other similarity measures derived spectral clustering.

In this part, spectral clustering experiments were conducted on 26 datasets based on Hamming, OF, Lin, ALGO, CMS, and our proposed HGS. We use *NMI* and *F-score* here to evaluate the category result and show them in Table 6 and Table 7, respectively. In both tables, the results for six measures are shown in columns and the bottom of the table shows the average performance on the whole. Besides, the best result among the six measures for each dataset is bold. Combining the performances from two tables, our measure HGS performs best on the whole, with the best average performance of 0.253 for *NMI* and 0.496 for *F-score*. According to *NMI*, Hamming distance and Lin rank the second and third place after HGS, while for *F-score*, CMS and ALGO distance tied for the second place. Lin method ranks third for *NMI* while Hamming distance achieves better for *F-score*. More precisely, from the perspective of *NMI*, HGS outperforms other measures in 8 datasets, while according to *F-score*, HGS achieves best in 7 datasets. Even though when HGS doesn't

TABLE 7. The *F-score* of Hamming, OF, Lin, ALGO, CMS vs. HGS-based spectral clustering.

Data	HM	OF	Lin	ALGO	CMS	HGS
Sos	0.786	0.598	**1.000**	0.809	0.786	0.809
Ha	0.387	0.468	0.307	**0.410**	0.351	0.361
He	0.496	0.414	0.443	0.503	**0.497**	0.565
Br	0.771	0.691	0.621	0.739	0.736	**0.782**
Ho	0.548	0.472	0.388	**0.759**	0.578	0.627
Sol	0.431	0.385	0.494	0.367	0.428	**0.491**
SP	0.474	0.522	0.328	0.344	0.470	**0.628**
Zo	**0.903**	0.803	0.789	0.854	0.787	0.899
DN	0.521	0.566	0.479	0.547	0.536	**0.623**
Ly	0.583	**0.602**	0.429	0.470	0.564	0.528
Mo	0.267	0.242	0.321	**0.533**	0.258	0.200
De	0.593	0.564	**0.694**	0.605	0.664	0.588
Cr	0.466	**0.571**	0.382	0.448	0.583	0.522
Ma	0.593	0.640	0.569	**0.662**	0.633	0.599
Fl	0.403	**0.686**	0.409	0.473	0.527	0.502
Pr	0.321	0.279	0.352	**0.369**	0.269	0.272
Ti	0.249	0.173	0.165	0.373	**0.264**	0.252
Ba	0.191	**0.241**	0.213	0.188	0.222	0.239
Ca	0.135	0.136	0.115	**0.172**	0.122	0.151
Ch	0.363	0.290	0.274	0.300	0.321	**0.401**
Cw	0.520	0.534	0.438	0.498	0.570	**0.614**
Im	0.713	0.543	0.574	0.534	**0.748**	0.622
Ip	**0.720**	**0.720**	0.695	0.567	**0.720**	**0.720**
Or	**0.664**	0.508	0.523	0.429	0.661	0.661
Bs	**0.184**	0.123	0.176	0.293	0.114	0.124
Bss	**0.184**	0.123	0.176	0.293	0.114	0.124
Average	0.479	0.458	0.437	0.482	0.482	**0.496**

TABLE 8. The *NMI* of Hamming, OF, Lin, ALGO, CMS vs. HGS-enabled k-modes clustering.

Data	HM	OF	Lin	ALGO	CMS	HGS
Sos	**1.000**	0.831	0.601	0.819	0.719	0.709
Ha	0.076	**0.194**	0.142	0.165	0.130	0.182
He	0.068	0.059	0.086	0.078	0.086	**0.114**
Br	0.331	0.267	0.295	**0.337**	0.319	0.316
Ho	**0.317**	0.303	0.300	0.308	0.289	0.300
Sol	0.581	0.555	**0.615**	0.573	0.601	0.573
SP	**0.100**	0.082	0.091	0.099	0.079	0.077
Zo	0.716	0.725	0.703	0.720	0.734	**0.753**
DN	0.109	0.090	0.130	**0.243**	0.115	0.116
Ly	0.220	0.194	0.176	0.186	0.177	**0.256**
Mo	0.025	0.043	0.029	0.041	**0.048**	0.025
De	0.624	0.363	**0.742**	0.738	0.668	0.674
Cr	0.148	0.155	0.115	0.103	0.165	**0.168**
Ma	0.197	0.182	0.164	0.188	0.195	**0.210**
Fl	**0.063**	0.060	0.059	0.062	0.059	**0.063**
Pr	0.415	0.365	0.436	**0.464**	0.431	0.447
Ti	0.033	0.024	0.031	0.022	0.042	**0.046**
Ba	0.047	0.046	0.047	0.000	0.048	**0.061**
Ca	0.081	0.066	0.081	0.000	0.068	**0.123**
Ch	0.073	0.070	**0.084**	0.062	0.076	0.076
Cw	0.040	0.033	0.037	**0.063**	0.054	0.056
Im	0.090	0.060	0.065	0.049	**0.139**	0.085
Ip	0.042	0.066	0.071	0.074	0.086	**0.110**
Or	0.034	0.018	0.017	**0.028**	0.022	0.017
Bs	**0.792**	0.771	0.788	0.785	0.772	0.785
Bss	**0.792**	0.771	0.788	0.785	0.772	0.785
Average	0.270	0.246	0.257	0.269	0.265	**0.274**

rank top, it can still obtain the comparable performance. On the contrary, OF measure gets the worst result on the average of *NMI* while Lin measure performs not good in the average evaluation of *F-score*. However, it's interesting that no measure can outperform all other measures in all datasets due to various characteristics. We can see every measure has performed best in at least one dataset. Therefore, it's essential to explore the characteristic to find a suitable measure.

(2) Comparison of HGS with other similarity measures based on k-modes clustering

Similar to before, Hamming, OF, Lin, ALGO, CMS, and our proposed HGS enabled k-modes clustering experiments were conducted on 26 datasets. The evaluation results and the average measure of *NMI* and *F-score* were shown in Table 8 and Table 9, respectively. Combining both evaluations, our HGS and ALGO perform an equally excellent performance, where HGS performs the best by *NMI* with 0.274 while ALGO is best for *F-score* with 0.462. Hamming distance ranks second according to both metrics. In the following, HGS and ALGO rank third according to *F-score* and *NMI*, respectively. More accurately, from the perspective of *NMI*, HGS performs the best in 10 datasets while ALGO is best in 5 datasets. On the contrary, ALGO obtains the best performance in 9 datasets while HGS wins in 7 datasets according to *F-score*. Subsequently, Hamming wins in 6 datasets via *NMI* and 3 datasets by *F-score*. Occurrence frequency and Lin measures perform worst in the k-modes clustering on both evaluations.

The clustering results evaluated by RI and purity indexes are shown in the supplementary materials as Tables 1-4 (See details in Supplementary materials). Similar to *NMI* and *F-score*, the results also show the effectiveness of our proposed HGS methods in both k-modes and spectral clustering experiments.

C. SIMILARITY SEARCH EXPERIMENT

In order to further validate our proposed method, in addition to the clustering experiment, we conduct the top-k similarity search experiments on the 26 datasets in Table 5 with HGS and other methods mentioned above. During the experiment, by computing the similarity matrix between all object pairs, we pick up the k objects with the top-k highest similarity score compared to the query object. According to the pre-defined label information that the dataset brings when we downloaded them, we evaluate the similarity search effectiveness by judging whether the k objects belong to the class of the query object. The similarity search accuracy (*SSA*) is

TABLE 9. The *F-score* of Hamming, OF, Lin, ALGO, CMS vs. HGS-enabled k-modes clustering.

Data	HM	OF	Lin	ALGO	CMS	HGS
Sos	**1.000**	0.873	0.703	0.851	0.731	0.727
Ha	0.314	**0.401**	0.354	0.357	0.348	0.389
He	0.380	0.375	0.412	0.376	**0.414**	0.420
Br	0.398	0.385	0.283	**0.419**	0.318	0.379
Ho	0.500	0.506	0.429	0.469	0.479	**0.512**
Sol	0.564	0.528	**0.587**	0.525	0.578	0.555
SP	0.366	0.359	0.325	**0.376**	0.354	0.325
Zo	0.657	0.686	0.657	0.665	0.685	**0.691**
DN	0.433	0.448	0.415	**0.573**	0.463	0.444
Ly	0.508	0.386	0.446	0.451	0.435	**0.538**
Mo	0.211	0.226	0.196	0.212	0.222	**0.234**
De	0.642	0.425	0.702	**0.723**	0.675	0.678
Cr	0.310	**0.358**	0.327	0.339	0.330	0.277
Ma	0.479	0.548	0.411	0.426	0.543	**0.549**
Fl	**0.235**	0.183	0.206	0.223	0.203	0.187
Pr	0.398	0.379	0.416	**0.439**	0.382	0.419
Ti	0.199	0.161	0.148	**0.240**	0.175	0.191
Ba	0.228	0.223	0.228	**0.631**	0.205	0.235
Ca	0.157	0.124	0.157	**0.824**	0.118	0.143
Ch	0.154	0.108	0.125	**0.245**	0.134	0.127
Cw	0.404	**0.442**	0.371	0.361	0.398	0.435
Im	**0.610**	0.396	0.392	0.395	0.599	0.538
Ip	0.587	0.568	0.531	0.479	0.579	**0.592**
Or	0.494	0.434	0.355	0.407	0.499	**0.600**
Bs	0.499	0.478	**0.504**	0.499	0.472	0.492
Bss	0.499	0.478	**0.504**	0.499	0.472	0.492
Average	0.432	0.403	0.392	**0.462**	0.416	0.430

calculated by

$$SSA = \frac{k_m}{k},$$

where k_m represents the number of the precisely matched objects. As the limit of the scope, here we present the similarity search experiment results when $k = 1, 5, 10$ in Fig. 3 and the average accuracy for all datasets in Table 10. As shown in Figure 3, each curve respectively represents the accuracy in similarity search experiment of each method. On the whole, HGS always run higher than other curves. Another comparative method is the CMS method with Hamming distance being the average and stable performance. Comparative speaking, occurrence frequency and Lin methods are not stable. They have arrived at the bottom of all curves for several times. The results in Table 10 show again this phenomenon. Our method HGS performed best when $k = 5$ and $k = 10$, with 0.780 and 0.754, respectively. When $k = 1$, it has obtained 0.800 for accuracy, which is only a difference of 0.001 from the first CMS with 0.801. Of course, the performance of CMS is not inferior, with 0.799 for $k = 5$ and 0.752 for $k = 10$. The subsequent method is Lin on the average, and Hamming distance performs worst when $k = 1$ and $k = 5$, while OF performs worst when $k = 10$.

TABLE 10. The average accuracy comparison of all methods for all datasets in similarity search when $k = 1, 5, 10$.

Dataset	Hamming	OF	Lin	ALGO	CMS	HGS
k=1	0.767	0.789	0.785	0.780	**0.801**	0.800
k=5	0.744	0.747	0.769	0.748	0.779	**0.780**
k=10	0.721	0.715	0.744	0.720	0.752	**0.754**

To summarize, the phenomenon shows that in the similarity search task, our proposed HGS method can perform comparably with the CMS method and better than other methods on the whole.

D. DISCUSSIONS

1) CONVERGENCE ANALYSIS

As an iterative algorithm, we need to pay attention to whether or how fast it can converge. In reference to the proof in section V-B, during calculations, similarity scores are non-decreasing and can converge fast. In order to show the convergence process more directly, we present the calculation on the Nursery dataset in Table 1 by HGS. In Fig. 4, the first one plots the changes of scores of attribute-value pairs in y-axis with iteration on the x-axis, while the bottom one plots the change of scores of family pairs during iteration. As seen, after nearly 4 iterations, the scores have approached to the stable states. Because we iterate from attribute-value pairs, at zero coordinate, the scores of all family pairs are zero while attribute-value pairs have non-zero scores. The change of scores for attribute-value pairs(occupation usual vs. pretentious) and another attribute-value pair(finance inconvenient vs. convenient) keep the same and finally converge to the same similarity score. They iterate from different scores and finally reach the highest scores among all pairs. Though sharing the same frequency in the table, less convenient, critical, and convenient house condition have been effectively distinguished. Compared with the other pairs, critical and convenient house condition are less similar, which is also in line with reality. In the bottom of the figure, we only compare family 3 with families 1,2, and 4, family 6 with families 4 and 5, which have much more divergences according to the results in Table 2. As seen, after the fourth iteration, all the scores have converged to the stable states. Although the score of family 3 vs. 2 is the highest compared with family 3 vs. 1 and family 3 vs. 4, they are very close. Besides, in the first iteration, family 3 is more similar to family 4. While after iteration, the score of family 3 vs. 2 has caught up and gone beyond its opponent. The phenomenon shows that the iteration process can capture the information from the global information structure and obtain stable and reliable results.

2) COMPARISON OF HGS AND ITS VARIANTS

In this part, we compare HGS with its two variants. As the presentation in Section IV, HGS is calculated iteratively based on a heterogeneous weighted undirected graph via three rules. Here, we are considering two other possible measures. One measure is based on a bipartite graph (called

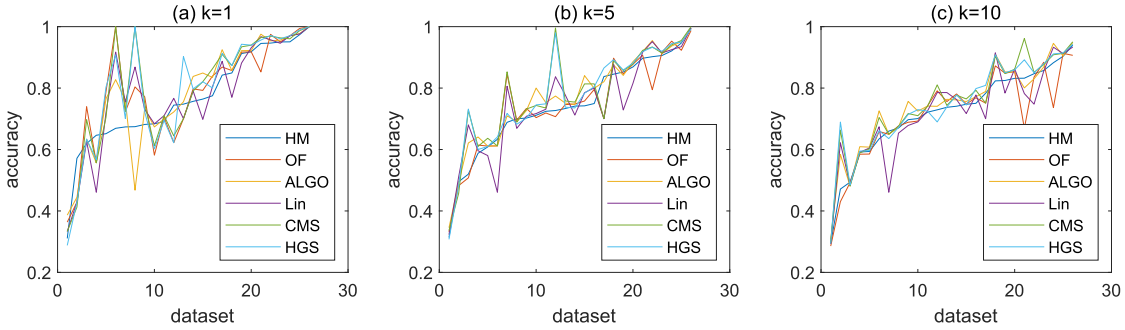

FIGURE 3. The accuracy of HGS similarity measure versus other methods based similarity search when $k = 1, 5, 10$. The dataset order is resorted ascending according to the HM results.

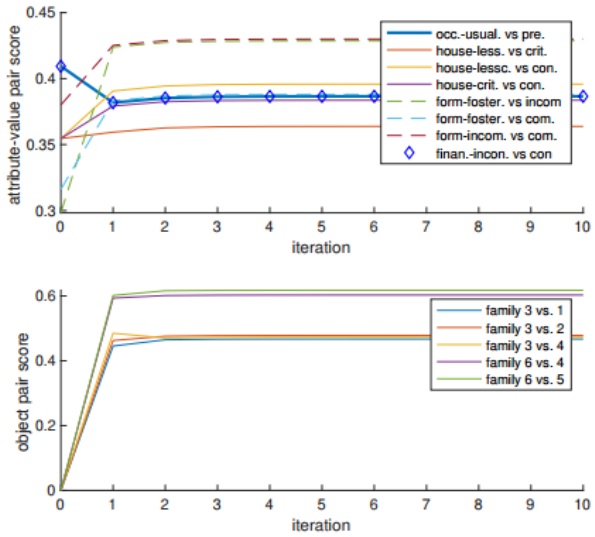

FIGURE 4. The converging process of HGS similarity measure for Car dataset in Table 1.

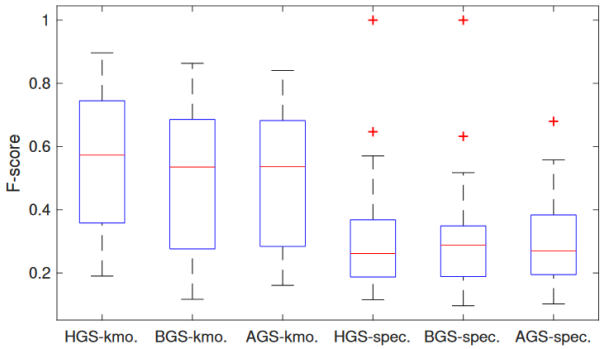

FIGURE 5. The boxplot of the *F-score* result from HGS similarity measure compared with its variants.

BGS), in which we remove the edges between attribute values and only iterate the scores between object pair and attribute-value pairs according to the rules of 1 and 2 (see Section IV-B.2). Another measure is based on an attribute graph (called AGS), for which we remove the edges between object pair and attribute-value pair, and the scores will only be iterated between attribute-value pairs according to the rule 3 until convergence. Finally, the object-pair scores will be the average score of all corresponding attribute-value pairs.

We compare their performances by *F-score* of k-modes experiments on all datasets. The result is shown in Figure 5. The figure orderly spreads out the box plot of *F-score* result derived from HGS, BGS, and AGS enabled k-modes and spectral clustering. In the box plot, the red line represents the median scores of all the results for each measure. Obviously, in k-modes clustering, the result from HGS is much better than BGS and AGS with a higher place of the whole distribution. Besides, their average performances, which are 0.555, 0.500, and 0.497, respectively, also show that HGS is much better than its two variants. With regard to spectral clustering, the results don't show an obvious difference. It is observed that BGS has a much higher median while both HGS and AGS have higher distributions. Their average performances are 0.324, 0.315, and 0.301, respectively. Thus, on average, HGS can outperform its variants by iterate the scores across the whole graph based on three rules. The subsequent is VGS which iterates the scores on a bipartite graph. The phenomenon also shows that iterating the scores between object pairs and attribute-value pairs is an effective procedure to obtain a better result.

3) COMPARISON OF VARIOUS SIMILARITY SCORE INITIALIZATION

As stated in section IV-B.1, we initialize the scores of attribute-value pairs by comparing their Jaccard occurrence frequency (JOF in short) according to the rule provided in 2. The less the frequency difference between two attribute values, the larger of their scores. An attribute value is most similar to itself. The score of an attribute-value pair consisting of different values is forever less than 1. In order to show the effectiveness of the initialization procedure, we compare it with Hamming and ALGO initialized HGS. The setting is that we only change the initialization of attribute-value pair scores according to Hamming and ALGO, respectively. Hamming simply attaches 1 to the attribute-value pair consisting of the same values, vice versa, 0. The ALGO is much more complex, which has captured the co-occurrence relationship between attribute values. The detailed procedure can in

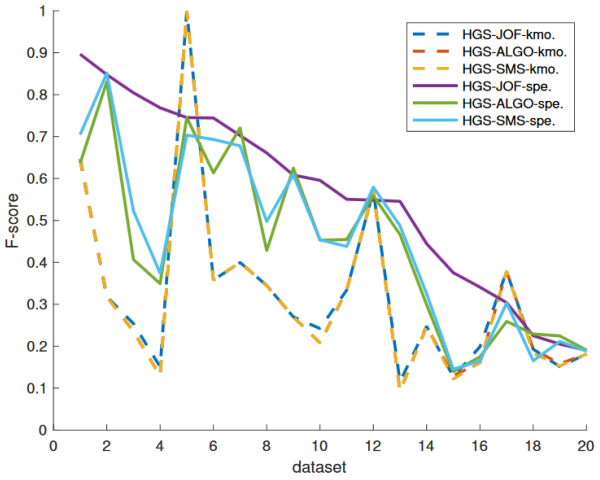

FIGURE 6. The plot of the *F-score* result from HGS Similarity Measure with different initialization methods. The x-axis represents the dataset and y-axis is the *F-score* evaluations for each clustering experiment. The dotted lines represent the results from HGS with OF, ALGO and Hamming initialization methods enabled k-modes clustering while solid lines are for spectral clustering, respectively.

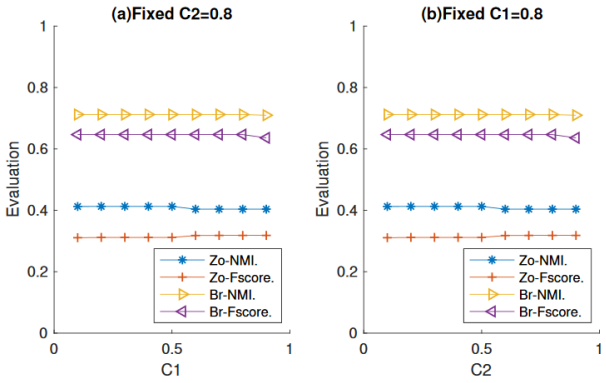

FIGURE 7. The plot of the results from HGS Similarity Measure with different C_1 and C_2. In both subplots, y-axis represents the evaluations from the experiments, however in (*a*) the x-axis represents C_1 changing from 0.1 to 0.9 while in (*b*) it represents C_2. The blue and red lines represent the *NMI* and *F-score* result derived from k-modes experiment on Zoo dataset, respectively, while the yellow and purple lines represent the *NMI* and *F-score* result on Breast cancer dataset.

reference to [20]. Then, three measures-disabled spectral and k-modes clustering algorithms were conducted on 20 datasets and their results will also be evaluated by *F-score*.

As shown in Fig. 6, in order to show the results more clearly, the *F-score* is sorted out descendingly according to the scores of HGS-OF enabled spectral clustering experiment. Therefore, the dataset order is different from that in Table 5. In the figure, by comparing the results of spectral clusterings, we can see that the purple line that represents our proposed HGS initialized with JOF is almost on top of the green and blue lines which represent HGS initialized with ALGO and Hamming methods. Thus, our measures perform better than another two methods in spectral clustering. Furthermore, the dotted lines represent the results derived from the k-modes experiment based on these three measures. We can see that three dotted lines have overlapped for the most part. But the blue dotted line representing our proposed method has gone beyond another two lines on the whole, particularly in the last five datasets. By calculating the average performance of all the results, our HGS measure has obtained the best values (0.324 for k-modes and 0.555 for spectral clustering), while HGS-ALGO is 0.318 and 0.440 and HGS-Hamming is 0.317 and 0.455 for k-modes and spectral clustering, respectively. From the analysis above, our initialization method has outperformed ALGO and Hamming initialization methods.

4) PARAMETER SENSITIVITY ANALYSIS OF C_1 AND C_2

In Eq. 3 and 4, the coefficients C_1 and C_2 not only represent the effect from its neighbours, but also a memory of its previous states. In our experiment, both of them are simply set as 0.8. In this discussion, we conducted k-modes clustering experiments on the Zoo (small scale, Zo in short) and Breast cancer (large scale, Br in short) datasets from UCI to analyze their effects on HGS measure. The experiment consists of two sub-experiments. First, the parameter C_2 was set as a fixed value of 0.8, then C_1 ranged from 0.1 to 0.9 with an interval of 0.1. In the second part, the parameter C_1 was set as a fixed value of 0.8, then C_2 was changed from 0.1 to 0.9 subsequently. In this scheme, only k-modes clustering experiment was conducted on both datasets because, in spectral clustering, there is a part of k-means clustering which depends on the initial cluster centers so that have some random change. Thus, we choose k-modes clustering of which the results are all dependent on the similarity measure.

As the result shown in Fig. 7, regardless of changing C_1 or C_2, the experiment result is very stable. Relatively speaking, the result on large-scaled datasets Br is more stable than the small-scale one Zo. With the increase of C_1 or C_2, there is only a slight drop of *NMI* for Zo dataset while a slight increase of *F-score*. The phenomenon shows that our proposed HGS measure is not sensitive to both the parameters C_1 and C_2. We think it's reasonable because the coefficients C_1 and C_2 only change the absolute values of node pairs, but don't change the relative value difference between object pairs, which are more important in distinguishing different objects in our work.

5) APPLICATION TO IMBALANCED DATA

Imbalanced data widely exists in the real-world application, which refers to the uneven distribution of the categories of the dataset. Here we analyze the sensitivity of our proposed HGS method to the imbalanced data. In Table 5, the imbalanced ratio (*IR*) of each dataset is present, which evaluates the ratio of the number of samples with the most samples to the number of classes with the fewest samples [46]. The larger the *IR* value, the more unbalanced the dataset. By reviewing the *IR* values of 26 datasets, we select the Ly and Fl datasets with the largest *IR* as the analysis basis, where the *IR* value is 40.5 and 884, respectively. Specifically, FL dataset

TABLE 11. The result comparison of the dataset with the largest IR value.

Data	Eval.	HM	OF	Lin	AlGO	CMS	HGS
Ly	S-NMI	0.202	0.188	0.168	0.201	0.183	**0.340**
	S-F_β	0.497	0.440	0.406	0.516	0.429	**0.661**
	K-NMI	0.212	0.167	**0.265**	0.198	0.253	0.254
	K-F_β	0.315	0.342	**0.361**	0.318	0.336	0.345
Fl	S-NMI	**0.228**	0.118	0.116	0.114	0.146	0.127
	S-F_β	0.622	0.478	0.208	0.420	0.482	**0.769**
	K-NMI	0.043	0.036	0.043	0.043	0.045	**0.072**
	K-F_β	0.129	0.128	0.129	0.149	0.131	**0.151**

TABLE 12. The time cost comparison of HGS. versus other methods in calculating similarity matrix. The method that has used the largest amount of time has been bold.

Data	HM	OF	Lin	ALGO	CMS	HGS
Sos	0.002	0.005	0.006	0.080	**0.101**	0.024
Ha	0.004	0.004	0.006	0.011	0.011	**0.020**
He	0.008	0.004	0.009	0.011	**0.013**	0.009
Br	0.091	0.112	0.193	0.190	0.178	**1.529**
Ho	0.012	0.017	**0.033**	0.023	0.018	0.026
Sol	0.019	0.037	0.091	**0.170**	0.162	0.111
SP	0.018	0.026	0.055	**0.051**	0.031	0.043
Zo	0.002	0.003	0.006	0.020	**0.037**	0.007
DN	0.004	0.019	0.031	0.705	**0.777**	0.085
Ly	0.005	0.008	0.016	0.058	**0.079**	0.030
Mo	0.058	0.046	0.080	0.013	0.016	**0.094**
De	0.042	0.078	0.174	0.305	0.287	**0.335**
Cr	0.085	0.079	0.158	0.067	0.071	**0.553**
Ma	0.127	0.067	0.127	0.024	0.028	**0.328**
Fl	0.210	0.185	0.403	0.074	0.070	**0.653**
Pr	0.004	0.005	0.011	**0.021**	0.017	0.010
Ti	0.178	0.210	0.369	0.051	0.052	**0.408**
Ba	0.075	0.052	0.078	0.014	0.019	**0.178**
Ca	0.573	0.359	0.593	0.104	0.115	**1.140**
Ch	2.824	3.995	**12.341**	1.506	1.387	8.543
Cw	0.011	0.044	0.114	**23.059**	3.596	1.080
Im	0.036	0.221	0.462	**260.602**	17.880	4.871
Ip	0.077	1.447	2.060	**552.184**	317.567	27.880
Or	0.021	0.312	0.412	**80.362**	56.577	5.767
Bs	1.017	4.954	14.080	**61.985**	47.887	23.575
Bss	1.017	4.954	14.080	**61.985**	47.887	23.575

is extremely unevenly distributed. There are 8 classes in Fl dataset, however, the maximum class has 884 samples, while the minimum class has only 1 sample. The F-measure and *NMI* in spectral clustering and k-modes clustering experiment of two datasets are summarized in the Table 11. As shown, compared with other methods, HGS measure obtains the best performance in Ly when applied in spectral clustering and Fl when applied in both spectral clustering and k-modes clustering. It's noted that in the k-modes clustering of Ly where the HGS is not the best, its performance is still quite competitive as the second best. Contrastly, in other cases that HGS obtains the best, other methods, such as HM, OF, *etc.*, are far from HGS. These performances indicate the effectiveness of our method HGS in application to the clustering based on imbalanced data.

6) COMPARISON OF TIME COST OF HGS AND OTHER METHODS

In performing practical data-driven tasks, with the increase of the scale of data, the time cost also increases dramatically. Here we mean to compare the time cost of different methods in computing the similarity matrix. The result is shown in Table 12. In the previous 20 datasets that are low-dimensional, our method HGS has consumed the largest amount of time in 10 datasets, while CMS, ALGO, and Lin are the highest time consumption in 5, 3, and 2 datasets. This phenomenon is reasonable as all the above methods have captured more complex relationships between attribute values. HGS has spent the most time in building the neighborhood matrix between the object-pair and attribute-value pair, attribute-value pair and attribute-value pair for the iterative process. Thus, it has spent more time in the low-dimensional dataset, however, still similar scale with other methods. In the later 6 datasets that are high-dimensional, ALGO has consumed the largest amount of time for all 6 datasets, while the second is CMS and our method HGS runs third. Specifically, in Ip dataset, ALGO has spent 552.184 seconds and CMS has cost 317.567 seconds, which are almost 20 and 11 times the cost of HGS, which is only 27.880 seconds. The similar condition exists in the other 5 datasets. This phenomenon shows that although HGS also captures the relationships between attribute values like ALGO and CMS, its increase speed is much slower than that of ALGO and CMS with the increase of the dimension of the data.

VII. CONCLUSION AND FUTURE WORKS

Measuring the similarity for categorical data in unsupervised learning has been a challenging task in data mining due to lacking the guidance from labeled results. The complex relationship hidden between attribute values and objects can provide a contribution to the measuring of similarity between objects. In order to capture this valuable information, this work creatively introduces a graph structure into the similarity measure for the unsupervised learning of categorical data.

As the most natural tool to represent a relationship, we construct a heterogeneous weighted graph to extract the information in the categorical data. Both objects and attribute values have been represented as nodes, and both the occurrence relationships between objects and attribute values and the co-occurrence relationships between attribute values have been represented as edges. In this way, the possible complex relationships can be captured into the graph structure. Besides, a node pair graph is derived from the previous constructed heterogeneous graph to formulate the calculation. By iterating the scores of both object pairs and attribute-value pairs across the node-pair graph, the complex relationships are reflected in the final converged scores. We can directly obtain both the object similarity and the attribute-value similarity in a very fast speed, which is very convenient to be applied in different domains. Finally, in experiments, we compare the performances in the k-modes and spectral clustering algorithms based on five state-of-the-art measures and our HGS measure. Although it can't outperform all other measures in every dataset, it can obtain the best average performance in 20 low-dimensional and 6 high-dimensional

datasets. Through extensive experiments, we find HGS can converge fast and is non-sensitive to the parameter C_1 and C_2. Besides, its graph setting and initialization of scores are the better choices compared with other schemes. Furthermore, it can be effectively applied to the high-dimensional dataset with the comparatively lower time cost.

This work first introduces a heterogeneous graph into the similarity measure for categorical data. There is still much work in the following. Due to the missing values in many practical conditions, how to apply HGS on this condition is what we need to focus on. Two different thoughts can be considered. The first thought is to utilize some link prediction techniques. By finding a similar object for the targeted object with missing values, their values can be copied to replace the missing values. Another thought is directly treating the missing values as a new attribute value, which is more convenient, but some disturbing information may be added. Another main direction is to apply our method in a mixed dataset that is composed of both categorical data and numerical data. For this condition, there are two solutions: 1) simply divide the dataset into categorical and numerical datasets and calculate the similarity for objects in each dataset, respectively. Then the final similarity value can be obtained by the average or weighted sum of both values; 2) discretize the numerical attribute into the categorical attribute, and then the mixed dataset has been transformed into a pure categorical dataset that the HGS can be directly applied to. Apart from the unsupervised learning, this work can also be extended to the supervised learning. With the label information in the training dataset, the object-pair within the same class can be set as a larger value than those in a different class. Besides, we can add class nodes in the graph and insert edges to connect the class node to the object nodes that belong to it. In this way, during the iterative process, the objects that are connected to the same class will get larger similarity scores. Finally, further improving the calculation efficiency is a necessary step especially when faced with big data.

ACKNOWLEDGEMENT

Many thanks to the reviewers providing valuable advice that is very helpful for improving our manuscript.

REFERENCES

[1] Y. Ye, J. Jiang, B. Ge, Y. Dou, and K. Yang, "Similarity measures for time series data classification using grid representation and matrix distance," *Knowl. Inf. Syst.*, vol. 60, no. 2, pp. 1105–1134, 2019. doi: 10.1007/s10115-018-1264-0.

[2] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques* (The Morgan Kaufmann Series in Data Management Systems), 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011, ch. 10, sec. 2, pp. 451–454.

[3] Q. Zhao, C. Wang, P. Wang, M. Zhou, and C. Jiang, "A novel method on information recommendation via hybrid similarity," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 3, pp. 448–459, Mar. 2018.

[4] G. C. Van, R. M. De, and K. Evangelos, "Neural vector spaces for unsupervised information retrieval," *ACM Trans. Inf. Syst.*, vol. 36, no. 4, pp. 1–25, 2018.

[5] X. Bai, Y. Zhang, H. Liu, and Z. Chen, "Similarity measure-based possibilistic FCM with label information for brain MRI segmentation," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2618–2630, Jul. 2019.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[7] A. Agresti, *An Introduction to Categorical Data Analysis* (The Wiley Series in Probability and Statistics), 3rd ed. Hoboken, NJ, USA: Wiley, 2018, ch. 1, sec. 1, pp. 1–5.

[8] H. Jia, Y.-M. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1065–1079, May 2016.

[9] B. McFee and G. Lanckriet, "Learning multi-modal similarity," *J. Mach. Learn. Res.*, vol. 12, pp. 491–523, Feb. 2011.

[10] I. Alabdulmohsin, M. Cisse, X. Gao, and X. Zhang, "Large margin classification with indefinite similarities," *Mach. Learn.*, vol. 103, pp. 215–237, May 2016.

[11] M. K. Ng, M. J. Li, J. Z. Huang, and Z. He, "On the impact of dissimilarity measure in K-modes clustering algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 503–507, Mar. 2007.

[12] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 243–254.

[13] D. Lin, "An information-theoretic definition of similarity," in *Proc. 15th Int. Conf. Mach. Learn.*, Madison, WI, USA, Jul. 1998, pp. 296–304.

[14] D. Dua and E. K. Taniskidou, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[15] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-K similarity search in heterogeneous information networks," in *Proc. VLDB Endowment*, 2011, pp. 992–1003.

[16] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proc. 24th ACM Inf. Knowl. Manage.*, 2015, pp. 453–462.

[17] G. Jeh and J. Widom, "SimRank: A measure of structural-context similarity," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 538–543.

[18] M. Alamuri, B. R. Surampudi, and A. Negi, "A survey of distance/similarity measures for categorical data," in *Proc. Int. Joint Conf. Neural Netw.*, 2014, pp. 1907–1914.

[19] J. Yuan and Y. Wu, "Context-aware clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[20] A. Ahmad and L. Dey, "A K-mean clustering algorithm for mixed numeric and categorical data," *Data Knowl. Eng.*, vol. 63, no. 2, pp. 503–527, 2007.

[21] C. Wang, L. Cao, M. Wang, J. Li, W. Wei, and Y. Ou, "Coupled nominal similarity in unsupervised learning," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 973–978.

[22] S. Jian, L. Cao, K. Lu, and H. Gao, "Unsupervised coupled metric similarity for non-IID categorical data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1810–1823, Sep. 2018.

[23] G. Das and H. Mannila, "Context-based similarity measures for categorical databases," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*, 2000, pp. 201–210.

[24] C. Wang, X. Dong, F. Zhou, L. Cao, and C.-H. Chi, "Coupled attribute similarity learning on categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 781–797, Apr. 2015.

[25] P. Zhao, J. Han, and Y. Sun, "P-Rank: A comprehensive structural similarity measure over information networks," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 553–562. doi: 10.1145/1645953.1646025.

[26] S.-H. Yoon, S.-W. Kim, and S. Park, "C-Rank: A link-based similarity measure for scientific literature databases," *Inf. Sci.*, vol. 326, pp. 25–40, Jan. 2016. doi: 10.1016/j.ins.2015.07.036.

[27] M. Zhang, J. Wang, and W. Wang, "HeteRank: A general similarity measure in heterogeneous information networks by integrating multi-type relationships," *Inf. Sci.*, vol. 453, pp. 389–407, Jul. 2018. doi: 10.1016/j.ins.2018.04.022.

[28] X. Meng, C. Shi, Y. Li, L. Zhang, and B. Wu, "Relevance measure in large-scale heterogeneous networks," in *Proc. Asia–Pacific Web Conf.*, 2014, pp. 636–643.

[29] Y. Zhou, H. Cheng, and J. X. Yu, "Graph clustering based on structural/attribute similarities," in *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 718–729, 2009. doi: 10.14778/1687627.1687709.

[30] M. R. Hamedani, S.-W. Kim, and D.-J. Kim, "SimCC: A novel method to consider both content and citations for computing similarity of scientific papers," *Inf. Sci.*, vol. 334, pp. 273–292, Mar. 2016. doi: 10.1016/j.ins.2015.12.001.

[31] M. Zhang, H. Hu, Z. He, and W. Wang, "Top-K similarity search in heterogeneous information networks with X-star network schema," *Expert Syst. Appl.*, vol. 42, no. 2, pp. 699–712, 2015. doi: 10.1016/j.eswa.2014.08.039.
[32] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," in *Proc. Int. Multiconf. Eng. Comput. Sci.*, 2013, vol. 1, no. 6, pp. 380–384.
[33] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[34] Z. Huang, "Extensions to the K-means algorithm for clustering large data sets with categorical values," *Data Mining Knowl. Discovery*, vol. 2, no. 3, pp. 283–304, 1998.
[35] M. K. Ng, M. J. Li, J. Z. Huang, and Z. He, "On the impact of dissimilarity measure in K-modes clustering algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 503–507, Mar. 2007.
[36] S. S. Khan and A. Ahmad, "Cluster center initialization algorithm for K-modes clustering," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7444–7456, 2013.
[37] S. Kosub, "A note on the triangle inequality for the Jaccard distance," 2016, *arXiv:1612.02696*. [Online]. Available: https://arxiv.org/abs/1612.02696
[38] D. Ienco, R. G. Pensa, and R. Meo, "From context to distance: Learning dissimilarity for categorical data clustering," *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 1, pp. 1–25, Mar. 2012.
[39] C. Zhu, L. Cao, Q. Liu, J. Yin, and V. Kumar, "Heterogeneous metric learning of categorical data with hierarchical couplings," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1254–1267, Jul. 2018.
[40] T. Wang, G. Cai, Y. Qiu, N. Fei, M. Zhang, X. Pang, W. Jia, S. Cai, and L. Zhao, "Structural segregation of gut microbiota between colorectal cancer patients and healthy volunteers," *ISME J.*, vol. 6, no. 2, pp. 320–329, 2012.
[41] X. C. Morgan, T. L. Tickle, H. Sokol, D. Gevers, K. L. Devaney, D. V. Ward, J. A. Reyes, S. A. Shah, N. LeLeiko, S. B. Snapper, A. Bousvaros, J. Korzenik, B. E. Sands, R. J. Xavier, and C. Huttenhower, "Dysfunction of the intestinal microbiome in inflammatory bowel disease and treatment," *Genome Biol.*, vol. 13, no. 9, 2012, Art. no. R79.
[42] E. Papa, M. Docktor, C. Smillie, S. Weber, S. P. Preheim, D. Gevers, G. Giannoukos, D. Ciulla, D. Tabbaa, J. Ingram, D. B. Schauer, D. V. Ward, J. R. Korzenik, R. J. Xavier, A. Bousvaros, and E. J. Alm, "Non-invasive mapping of the gastrointestinal microbiota identifies children with inflammatory bowel disease," *PLoS ONE*, vol. 7, no. 6, 2012, Art. no. e39242.
[43] M. C. Ross, D. M. Muzny, J. B. McCormick, R. A. Gibbs, S. P. Fisher-Hoch, and J. F. Petrosino, "16S gut community of the cameron county hispanic cohort," *Microbiome*, vol. 3, no. 1, 2015, Art. no. 7.
[44] C. Duvallet, S. M. Gibbons, T. Gurry, R. A. Irizarry, and E. J. Alm, "Meta-analysis of gut microbiome studies identifies disease-specific and shared responses," *Nature Commun.*, vol. 8, no. 1, 2017, Art. no. 1784.
[45] E. L. Clare, B. K. Lim, M. D. Engstrom, J. L. Eger, and P. D. N. Hebert, "DNA barcoding of Neotropical bats: Species identification and discovery within Guyana," *Mol. Ecol. Notes*, vol. 7, pp. 184–190, 2007. doi: 10.1111/j.1471-8286.2006.01657.x.
[46] C. Peng and Q. Cheng, "Discriminative regression machine: A classifier for high-dimensional data or imbalanced data," 2019, *arXiv:1904.07496*. [Online]. Available: https://arxiv.org/abs/1904.07496
[47] C. Peng, Z. Kang, S. Cai, and Q. Cheng, "Integrate and conquer: Double-sided two-dimensional k-means via integrating of projection and manifold construction," *ACM Trans. on Intel. Sys. And Tech.*, vol. 9, no. 5, 2018, Art. no. 57. doi: 10.1145/3200488.

YANQING YE received the B.E. degree in management engineering and the M.E. degree in management science and engineering from the National University of Defense Technology, Changsha, Hunan, China, in 2013 and 2015, respectively, where she is currently pursuing the Ph.D. degree in management science and engineering.

From 2017 to 2019, she has been co-cultivated with the Physics Department, Boston University, Boston, MA, USA. Her research interests include data mining, artificial intelligence, and complex networks.

JIANG JIANG received the B.E. degree in systems engineering and the M.E. and Ph.D. degrees in management science and engineering from the National University of Defense Technology, Changsha, Hunan, China, in 2004, 2006, and 2011, respectively.

He was a Visiting Scholar with the Channing Division of Network Medicine, Harvard Medical School, Boston, MA, USA. He is currently an Associate Professor of management science and engineering with the National University of Defense Technology. His research interests include evidential reasoning, uncertainty decision-making, and risk analysis.

BINGFENG GE (S'11–M'14) received the B.E. degree in systems engineering and the M.E. and Ph.D. degrees in management science and engineering from the National University of Defense Technology, Changsha, Hunan, China, in 2006, 2008, and 2014, respectively.

He was a Visiting Scholar with the Conflict Analysis Group, Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently an Associate Professor of management science and engineering with the National University of Defense Technology. His research interests include system-of-systems architecting and engineering management, portfolio decision analysis, and conflict resolution.

Dr. Ge is a Technical Committee Member of Conflict Resolution of the IEEE Systems, Man, and Cybernetics Society, and a member of the IEEE Internet of Things Technical Community and the International Council on Systems Engineering.

KEWEI YANG received the B.E. degree in systems engineering and the Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, Hunan, China, in 1999 and 2004, respectively.

He was a Visiting Scholar with the Department of Computer Science, University of York, U.K., and with the Science and Technology on Complex Systems Simulation Laboratory, Beijing, China. He is currently a Professor of management science and engineering and the Director of the Department of Management, College of Systems Engineering, National University of Defense Technology. His research interests include intelligent agent simulation, defense acquisition, and system-of-systems requirement modeling. He has been a member of Youth Working Committee in the Systems Engineering Society of China, since 2009.

H. EUGENE STANLEY received the Ph.D. degree in physics from Harvard University, in 1967. He is currently an American Physicist and a University Professor with Boston University, USA. He has made fundamental contributions to complex systems and is one of the founding fathers of econophysics. His current research interests include complexity science and econometrics. He was elected to the U.S. National Academy of Sciences, in 2004.