Pattern Recognition 128 (2022) 108694

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A categorical data clustering framework on graph representation

Liang Bai, Jiye Liang*

Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006, China

ARTICLE INFO

Article history:
Received 3 July 2021
Revised 16 March 2022
Accepted 3 April 2022
Available online 5 April 2022

Keywords:
Cluster analysis
Categorical data clustering
Data representation
Graph embedding

ABSTRACT

Clustering categorical data is an important task of machine learning, since the type of data widely exists in real world. However, the lack of an inherent order on the domains of categorical features prevents most of classical clustering algorithms from being directly applied for the type of data. Therefore, it is very key issue to learn an appropriate representation of categorical data for the clustering task. In order to address this issue, we develop a categorical data clustering framework based on graph representation. In this framework, a graph-based representation method for categorical data is proposed, which learns the representation of categorical values from their similar graph to provide similar representations for similar categorical values. We compared the proposed framework with other representation methods for categorical data clustering on benchmark data sets. The experiment results illustrate the proposed framework is very effective, compared to other methods.

1. Introduction

Data clustering is an unsupervised classification technique that aims at grouping a set of unlabeled objects into meaningful clusters, with the requirement that the objects in the same cluster have high similarity but are very dissimilar with objects in other clusters. Clustering techniques have been extensively studied in several communities (e.g., [1] and references therein). Many high-quality clustering algorithms, such as *k*-means [2], density clustering [3], graph clustering [4–6] have been designed for numerical data. Unfortunately, these algorithms cannot be directly applied for clustering of categorical data, where domain values are discrete and have no ordering defined. Since categorical data is ubiquitous in real-world applications [7,8], increasing attentions have been paid to clustering the type of data [9,10].

Currently, the wide-used methods for clustering categorical data can be categorized into the following two types.

(1) *Direct-clustering method* that designs a clustering algorithm which is suitable for categorical data. Currently, there are three types of categorical data clustering algorithms, seen in [11]. The first type is extension of the numerical data clustering algorithms, such as ROCK [12], *k*-modes [13] and its variants [14–17], which define the distance measure and the cluster centroids for categorical data. Currently, a number of dissimilarity measures for categorical data have been developed, whose related works can be found in [18]. The second type [19–21] is to employ category utility function [22] as a clustering objective function to maximize the probability that two data objects in the same cluster obtain the same attribute values. The third type [23–25] is to use the information entropy to find out groups of similar objects that have lower entropy than those of dissimilar ones. The advantage of direct-clustering methods is that their interpretability is strong. They can sufficiently reflect the characteristics of categorical data. Their disadvantage is that their adaptability is weak. For different types of categorical data, we need to design the corresponding algorithms. Besides, most of state-of-the-art clustering algorithm for numerical data can not be directly applied for categorical data.

(2) *Converting-based method* that transform categorical data into numerical data and then cluster it by one of the existing clustering algorithms. The two most popular methods are ordinal encoding and one-hot encoding. In ordinal encoding, each categorical value is converted into an integer value. However, the converted numerical data does not necessarily produce meaningful results in the case where categorical domains are not ordered. In one-hot encoding, each category is mapped with a binary variable containing either 0 or 1. Here, 0 represents the absence, and 1 represents the presence of that category. People treat the binary features as numeric in the clustering algorithms [26]. In [27,28], the researchers proposed a link-based representation method which improves the binary features and used the similarity between categorical values, instead of 0. Furthermore, Jian et al. [29,30], Zhu et al. [31] developed the coupled data embedding (CDE) technique to represent categorical data which can capture the couplings between categor-

* Corresponding author.
E-mail addresses: bailiang@sxu.edu.cn (L. Bai), ljy@sxu.edu.cn (J. Liang).

https://doi.org/10.1016/j.patcog.2022.108694
0031-3203/© 2022 Elsevier Ltd. All rights reserved.

ical values and clusters. Zheng et al. [32] make use of similarity between objects to define space structure and representation of categorical data. Besides, people make use of a similarity measure to convert the categorical data into a pairwise-similarity matrix which can be used for spectral clustering. For example, the metric learning methods for categorical data have been proposed in [33–35]. Compared to direct-clustering methods, the converting-based methods can reduce the costs of designing algorithms. However, its disadvantage is that the converted numerical data often has weak interpretability and may lead to information loss of the original categorical data.

In this paper, we focus on converting-based clustering. The effectiveness of categorical data clustering mainly depends on whether they sufficiently make use of the intrinsic similarity of categorical values. However, using numerical vectors to describe the similarity is a challenge for most of converting-based clustering algorithms. In order to solve this problem, we propose a new categorical data clustering framework based on graph representation. This framework divides a categorical data clustering problem into three subproblems: representation of categorical values, representation of objects, and numerical data clustering. Different from existing categorical data representation methods, our main work is to learn and integrate the representation of categorical values from their graph structure, which can sufficiently capture the potential similarity between categorical values and provide the similar numerical representations for similar values.

The outline of the rest of this paper is as follows. Section 2 presents a new categorical data clustering framework. Section 3 demonstrates the performance of the proposed framework. Section 4 concludes the paper with some remarks.

2. Categorical data clustering

2.1. Problem formulation

Let X be a $n \times m$ data matrix, where n is the number of objects and m is the number of categorical features, $A = \{a_j\}_{j=1}^{m}$ is a set of m categorical features, where a_j is the jth feature. $\mathbf{x}_i$ is the ith row of X which represents the ith objects with m feature values. x_{ij} is the jth feature value of $\mathbf{x}_i$. Each feature a_j describes a domain of values, denoted by $D(a_j)$, associated with a defined semantic and a data type. Here, only consider two general data types, numerical and categorical, and assume other types used in database systems can be mapped to one of the two types. The domains of features associated with the two types are called numerical and categorical, respectively. A numerical domain consists of real numbers. A domain $D(a_j)$ is defined as categorical if it is finite and unordered, i.e., $D(a_j) = \{a_{j1}, a_{j2}, \cdots, a_{jn_j}\}$ where n_j is the number of categories of feature a_j for $1 \leq j \leq m$. For any $1 \leq p \leq q \leq n_j$, either $a_{jp} = a_{jq}$ or $a_{jp} \neq a_{jq}$. If each feature in A is categorical, X is called a categorical data set.

The aim of clustering categorical data X is to find out a partition of X into k clusters. Let U be a $n \times k$ partition matrix of X, where u_{il} is membership degree of object $\mathbf{x}_i$ to the lth cluster. Compared to numerical data, the difficulty of clustering categorical data is that we can not visually observe the similarity between categorical values. In order to solve this problem, we study categorical data clustering based on data representation, which transforms the categorical data into numerical data and cluster it.

2.2. Clustering framework

In this paper, we propose a new categorical data clustering framework based on graph representation, seen in Fig. 1. Its main idea is to learn a vector for each categorical value and integrate these vectors of the categorical values which an object includes to represent it. Based on this idea, the clustering problem with categorical data representation can be seen to learn three mappings which are defined as follows.

Table 1
Set-similarity measures.

Description	Equation
Jaccard coefficient	$\frac{\lvert S(a_{jh}) \cap S(a_{rl}) \rvert}{\lvert S(a_{jh}) \cup S(a_{rl}) \rvert}$
Ochiai coefficient	$\frac{\lvert S(a_{jh}) \cap S(a_{rl}) \rvert}{\sqrt{\lvert S(a_{jh}) \rvert \lvert S(a_{rl}) \rvert}}$
Overlap coefficient	$\frac{\lvert S(a_{jh}) \cap S(a_{rl}) \rvert}{\min(\lvert S(a_{jh}) \rvert, \lvert S(a_{rl}) \rvert)}$
Dice coefficient	$\frac{\lvert S(a_{jh}) \cap S(a_{rl}) \rvert}{\lvert S(a_{jh}) \rvert + \lvert S(a_{rl}) \rvert}$

- $f(a_{jh})$ maps categorical value a_{jh} to a vector with p elements to represent it, for $1 \leq j \leq m$ and $1 \leq h \leq n_j$;
- $g(\mathbf{x}_i) = \bigodot_{j=1}^{m} f(x_{ij})$ maps $\mathbf{x}_i$ to a vector with q elements to represent it, where $\bigodot$ is an operation of integrating the vectors of the categorical values of the object;
- $\pi(g(\mathbf{x}_i)) = [u_{i1}, \cdots, u_{ik}]$ is a clustering function used to map $g(\mathbf{x}_i)$ to the membership vector of $\mathbf{x}_i$.

$f(.)$ and $g(.)$ are used to transform objects into numerical representation. $\pi(.)$ is to cluster the transformed data and get its membership matrix U. Therefore, we mainly need to discuss how to define $f(.)$ and $g(.)$.

2.3. Representation of categorical value

The one-hot encoding is the most commonly used method to define $f(a_{ij})$, i.e., $f(a_{ij})$ is a binary vector with n_j elements, where the jth element is equal to 1 and others are 0. However, the one-hot encoding does not easily reflect the similarity between different categorical values. It can only determine whether two categorical values are the same. In order to solve this problem, we propose a graph-based representation method to get $f(.)$. The main idea of this method is to construct a similarity-relational graph G of all the categorical values and use one of graph embedding methods to learn the representation of nodes in G, which is shown in Fig. 2. By a graph embedding method, we can easily capture the inherent similarity between categorical values and find similar representations for similar categorical values.

We first construct the graph G. Let $G =< V, W >$ be an undirected and weighted graph, where $V = \bigcup_{j=1}^{m} D(a_j)$ is a set of nodes, W is a $|V| \times |V|$ weight matrix and $W(a_{jh}, a_{rl})$ is a weight of the edge between nodes a_{jh} and a_{rl} used to reflect their similarity. $W(a_{jh}, a_{rl})$ is computed by measuring the similarity between two object sets $S(a_{jh}) = \{x_{ij} = a_{jh}, \mathbf{x}_i \in X\}$ and $S(a_{rl}) = \{x_{ij} = a_{rl}, \mathbf{x}_i \in X\}$. We hope $W(a_{jh}, a_{rl})$ is proportional to the number of common objects with the categorical values a_{jh} and a_{rl}, i.e.,

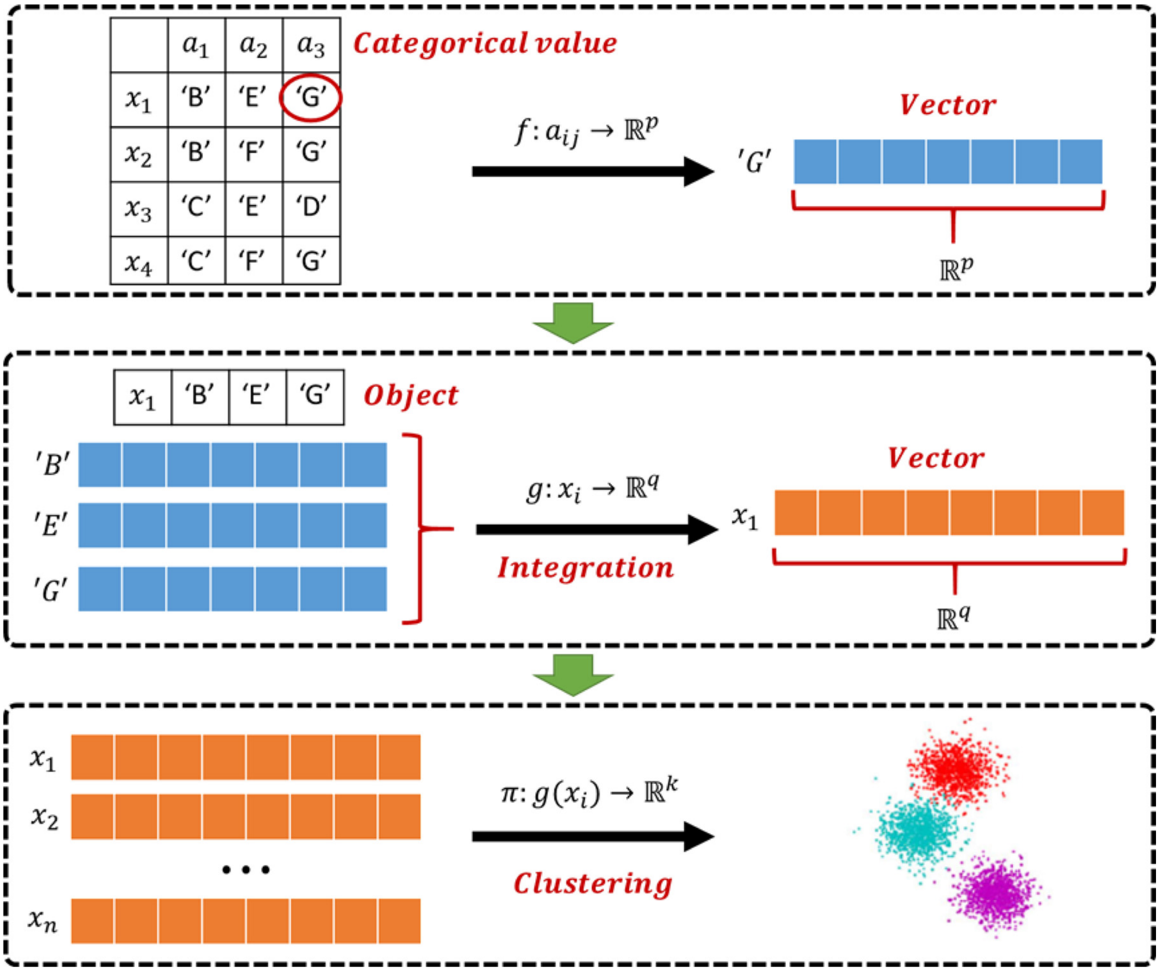
$$W(a_{jh}, a_{rl}) \propto |S(a_{jh}) \cap S(a_{rl})|.$$

Based on the assumption, we can employ one of set-similarity measures [36] to define W. The representatives of set-similarity measure are shown in Table 1.

Given the graph G of categorical values, the graph embedding is employed to find embedding of nodes to p-dimensions so that "similar" nodes in the graph have embeddings that are close together. Let $\phi(W)$ be a graph embedding function which can transform W into a $|V| \times p$ representation matrix $Z = \phi(W)$. In this case, we get the representation vector of each categorical value as follows.

$$f(a_{jh}) = \mathbf{z}_r,$$

where $r = \sum_{i=1}^{j-1} n_i + h$ and $\mathbf{z}_r$ is the rth row of Z. We can employ one of graph embedding methods to get $\phi(W)$. Currently, a num-



	a_1	a_2	a_3
x_1	'B'	'E'	'G'
x_2	'B'	'F'	'G'
x_3	'C'	'E'	'D'
x_4	'C'	'F'	'G'

x_1	'B'	'E'	'G'

Fig. 1. Clustering framework.

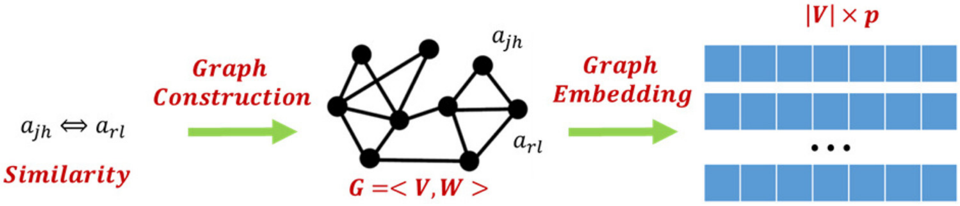


Fig. 2. Representation of categorical value.

Table 2
Graph embedding methods.

Description	Equation
NE	$Z = W$
SE	$\min_Z tr[Z^T(D - W)Z]$
NMF	$\min_Z \|\|W - ZH\|\|^2$
AE	$\min_{\phi,\psi} \|\|W - \psi(\phi(W))\|\|^2$

ber of graph embedding methods have been developed. Some classical methods, such as Non Embedding (NE) where W is directly seen as a feature data, Spectral Embedding (SE) [4], Nonnegative Matrix Factorization (NMF) [37], Autoencoder (AE) [38], are shown in Table 2. Since the graph embedding operation is implemented on the categorical values and $|V| \ll n$ in many data sets, its time complexity should be far less than directly learning the representation on a data set.

2.4. Representation of categorical data

Given $f(a_{jh})$ for each categorical value, in order to get $g(\mathbf{x}_i)$ to represent objects, we need to define the integration operations $\odot$ which uses the numerical vectors of categorical values of an object to represent it. In this paper, we provide two methods which is shown in Fig. 3 to define $\odot$ as follows.

In the first method, $\odot$ is seen as a joint operation and $g(\mathbf{x}_i)$ is defined as

$$g(\mathbf{x}_i) = [f(x_{i1}), \cdots, f(x_{im})]$$

which is a mp-dimensional vector. If $f(a_{jh})$ is computed by non embedding, i.e., $Z = W$, the squared Euclidean distance between objects is described as

$$d^2(g(\mathbf{x}_i), g(\mathbf{x}_j)) = \sum_{h=1}^{m} \sum_{a_{rl} \in V} \left[W(x_{ih}, a_{rl}) - W(x_{jh}, a_{rl})\right]^2.$$

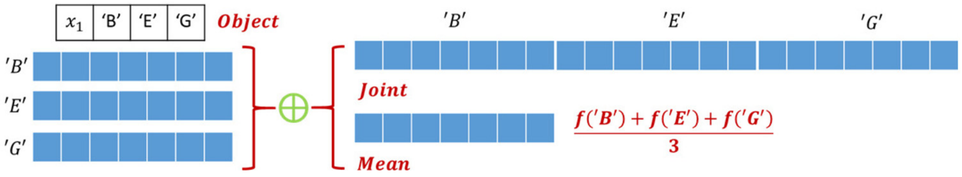


Fig. 3. Representation of categorical data.

Table 3
Description of data sets.

Data set	n	m	k	Type
Soybean	47	21	4	Categorical
Zoo	101	16	7	Categorical
Heart disease	303	8	2	Categorical
Breast cancer	699	9	2	Categorical
Dermatology	366	33	6	Categorical
Letters(E,F)	1543	16	2	Categorical
DNA	3,190	60	3	Categorical
Mushroom	8,124	22	2	Categorical
Iris	150	4	3	Numerical
Isolet	1,560	617	26	Numerical
COIL20	1,140	1024	20	Numerical
OpticalDigits	5,620	64	10	Numerical
PenDigits	10,992	36	10	Numerical

The distance does not directly measure the dissimilarity of between corresponding categorical values of two objects but evaluate the difference between the similarity of them with all the categorical values. This indicates that the inherent similarity between categorical values is sufficiently considered in the data representation.

In the second method, $\odot$ is defined as a mean operation and then $g(\mathbf{x}_i)$ becomes

$$g(\mathbf{x}_i) = \frac{1}{m}\sum_{j=1}^{m} f(x_{ij})$$

which is a p-dimensional vector. Compared to the joint operation, this representation has low dimensions and smooth feature values.

2.5. Clustering categorical data

Given $g(\mathbf{x}_i)$ for $1 \le i \le n$, we can employ one of classical clustering algorithms for numerical data to define the clustering function $\pi(.)$. The clustering algorithm includes k-means, linkage, spectral clustering and so on. Therefore, the overall clustering process in the proposed framework is described in Algorithm 1, which is called "CDC_DR". According to the description of the algorithm, the computation cost of the proposed framework is made up of constructing graph ($O(|V|^2n)$), graph embedding, integration operation ($O(nmp)$) and numerical data clustering ($O(nqk)$ if k-means is selected). For the graph embedding, different methods need different computational costs. The time complexities of NE, SE, NMF and AE (is set as the three-level network) are $O(1)$, $O(|V|^2p)$, $O(|V|^2p+|V|p^2)$ and $O(|V|^3p)$, respectively. We can see that the computational cost of NE is the lowest, since it does not represent graph. The time complexity of NMF is squared with $|V|$ and p. AE needs high training costs, compared to other methods. For spectral embedding (SE), its cost is mainly from eigen decomposition. If the number of nodes in a graph is not large, the computational cost can be acceptable.

Algorithm 1 CDC_DR.

```
Input: $X$, $k$
Parameter: 'Set-Similarity Measure', 'Graph Embedding Method', 'Integration Operation', 'Clustering Algorithm'
Output: $U$Build a graph $G =< V,W >$ of categorical values by the selected similarity measure;Get a representation matrix of categorical values $f(V)$ by the selected graph embedding method;Get a representation matrix of objects $g(X)$ by the selected integration operation;Compute $U = \pi(g(X))$ by the selected clustering algorithm;return $U$
```

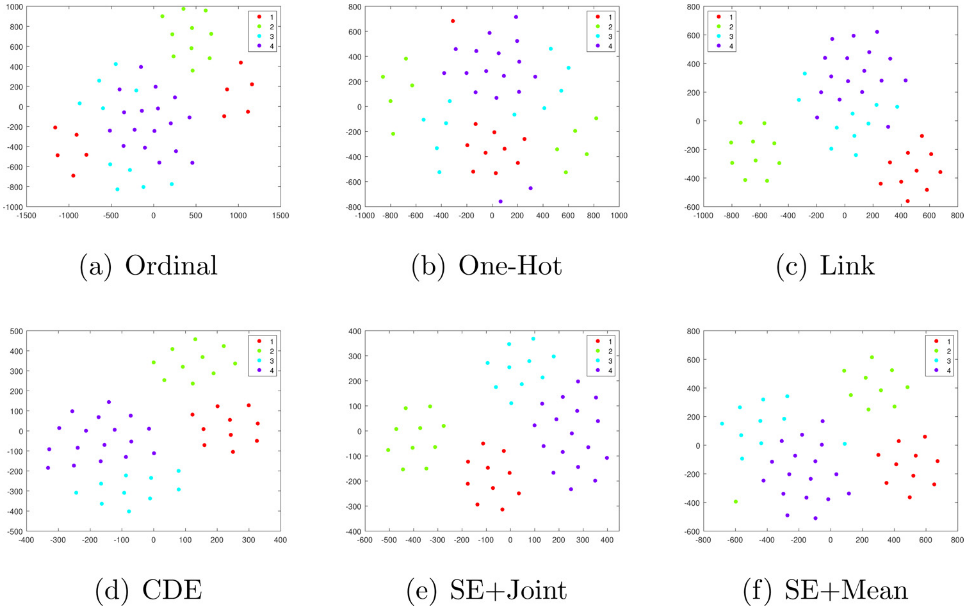
3. Experiment analysis

3.1. Experiment environment

In order to examine the performance of the proposed framework, we select Ochiai coefficient as set-similarity measure and k-

Table 4
The proposed framework for clustering categorical data.

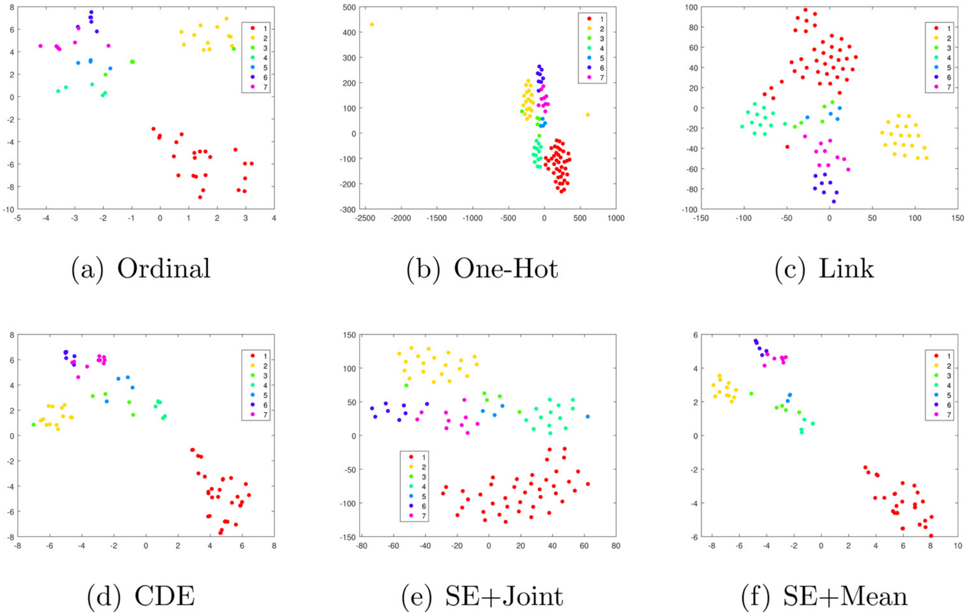
Dataset	Index	NE		SE		NMF		AE		Average	Max
		Joint	Mean	Joint	Mean	Joint	Mean	Joint	Mean		
Soybean	ARI	0.7784±0.21	0.8312±0.21	0.7498±0.2	0.8572±0.19	0.813±0.21	0.8381±0.19	0.7774±0.2	0.8159±0.21	0.8076±0.2	0.8572±0.21
	NMI	0.8787±0.12	0.9058±0.12	0.8627±0.11	0.9254±0.1	0.8974±0.12	0.9116±0.1	0.879±0.11	0.8996±0.12	0.895±0.11	0.9254±0.12
Zoo	ARI	0.6867±0.17	0.6765±0.15	0.6414±0.14	0.6623±0.14	0.666±0.16	0.6584±0.14	0.7069±0.14	0.653±0.14	0.6689±0.15	0.7069±0.17
	NMI	0.7949±0.08	0.8026±0.07	0.7777±0.07	0.7823±0.06	0.7859±0.07	0.7925±0.07	0.7999±0.06	0.7684±0.06	0.788±0.07	0.8026±0.08
Heart disease	ARI	0.4024±0.06	0.4424±0.00	0.4232±0.06	0.4208±0.00	0.4187±0.01	0.4384±0.01	0.3892±0.09	0.3169±0.06	0.4065±0.04	0.4424±0.09
	NMI	0.3144±0.05	0.3491±0.00	0.3342±0.05	0.3297±0.00	0.3278±0.01	0.3454±0.01	0.3086±0.07	0.2646±0.03	0.3217±0.03	0.3491±0.07
Dermatology	ARI	0.6273±0.16	0.589±0.12	0.6887±0.16	0.5449±0.09	0.6297±0.16	0.5742±0.11	0.6941±0.14	0.5347±0.06	0.6103±0.12	0.6941±0.16
	NMI	0.7931±0.09	0.7586±0.07	0.8183±0.08	0.7279±0.03	0.7925±0.09	0.7434±0.06	0.8247±0.07	0.6558±0.01	0.7643±0.06	0.8247±0.09
Breast cancer	ARI	0.8988±0.00	0.8988±0.00	0.8988±0.00	0.9043±0.00	0.8988±0.00	0.8988±0.00	0.8026±0.19	0.8727±0.03	0.8842±0.03	0.9043±0.19
	NMI	0.8269±0.00	0.8311±0.00	0.8269±0.00	0.8377±0.00	0.8269±0.00	0.8311±0.00	0.7252±0.18	0.7951±0.03	0.8126±0.03	0.8377±0.18
DNA	ARI	0.68±0.09	0.2442±0.03	0.5077±0.12	0.6799±0.13	0.1605±0.05	0.0618±0.00	0.5085±0.15	0.3542±0.12	0.3996±0.09	0.6800±0.15
	NMI	0.6425±0.07	0.2644±0.03	0.4739±0.09	0.6344±0.11	0.1731±0.04	0.0665±0.01	0.5225±0.11	0.3699±0.09	0.3934±0.07	0.6425±0.11
Letters	ARI	0.3741±0.26	0.4007±0.2	0.4862±0.23	0.4404±0.19	0.5247±0.00	0.3668±0.00	0.2902±0.21	0.3902±0.19	0.4092±0.16	0.5247±0.26
	NMI	0.3598±0.25	0.3778±0.19	0.4641±0.21	0.4275±0.15	0.5124±0.00	0.3084±0.00	0.2665±0.19	0.3469±0.16	0.3829±0.14	0.5124±0.25
Mushroom	ARI	0.3176±0.23	0.3909±0.23	0.6165±0.09	0.607±0.09	0.6174±0.02	0.609±0.02	0.383±0.25	0.3213±0.21	0.4828±0.14	0.6174±0.25
	NMI	0.3599±0.17	0.4010±0.18	0.5845±0.08	0.5734±0.08	0.5883±0.04	0.5658±0.04	0.3858±0.21	0.3605±0.14	0.4774±0.12	0.5883±0.21



(a) Ordinal (b) One-Hot (c) Link

(d) CDE (e) SE+Joint (f) SE+Mean

Fig. 4. Scatter plots of different methods on Soybean.



(a) Ordinal (b) One-Hot (c) Link

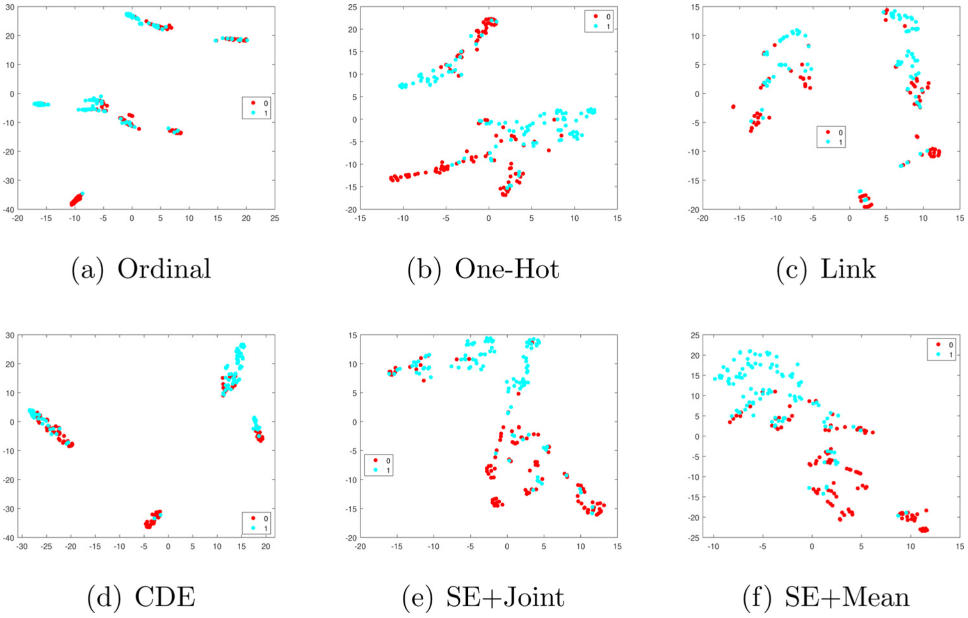
(d) CDE (e) SE+Joint (f) SE+Mean

Fig. 5. Scatter plots of different methods on Zoo.

means as clustering algorithm. Furthermore, we test four graph-embedding methods including NE, SE, NMF and AE, and two integration operations including *Joint* and *Mean* in this framework. While using NE, we know p equal to the sum of the numbers of categorical values of all the features. For SE, NMF and AE, we set p to their average number. We compare the proposed framework with categorical data clustering methods based on five different categorical data encodings: k-modes [13], k-means with ordinal encoding, one-hot encoding [26], link-graph encoding [28], and coupled data embedding [29]. Besides, we set the number of clusters on each tested data set to its "true" number of classes. The experiments are conducted on an Intel i9-7940X@3.10GHz personal computer with 128G RAM and Matlab 2016b.

In the experiments, we consider two scenarios of comparisons. The one is clustering categorical data sets. The second is clustering ensemble, where multiple clustering results of a data set are seen as its categorical features. A clustering ensemble problem can be seen as a categorical data clustering problem [39,40]. Therefore, in this paper, we also show the performance of the proposed framework on clustering ensemble. The comparisons are carried out on 13 benchmark data sets including (8 categorical data and 5 numerical data), which can been downloaded from https://archive.ics.uci.edu/ml and https://cs.nyu.edu/roweis/data.html. They are described in Table 3.

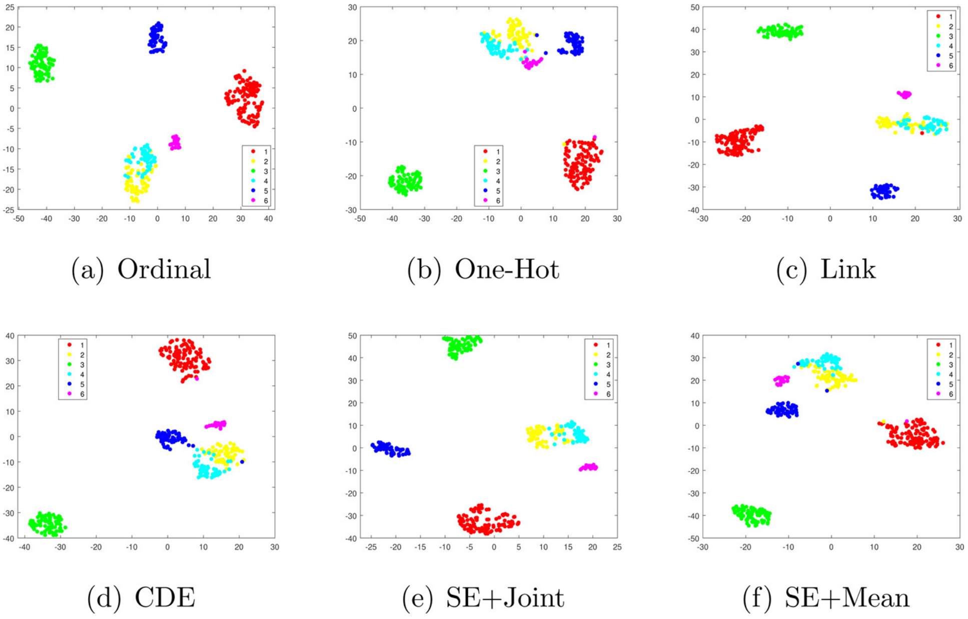
In these comparisons, we evaluate the clustering accuracy and computational cost of each method on each data set. In order to



(a) Ordinal (b) One-Hot (c) Link

(d) CDE (e) SE+Joint (f) SE+Mean

Fig. 6. Scatter plots of different methods on Heart disease.



(a) Ordinal (b) One-Hot (c) Link

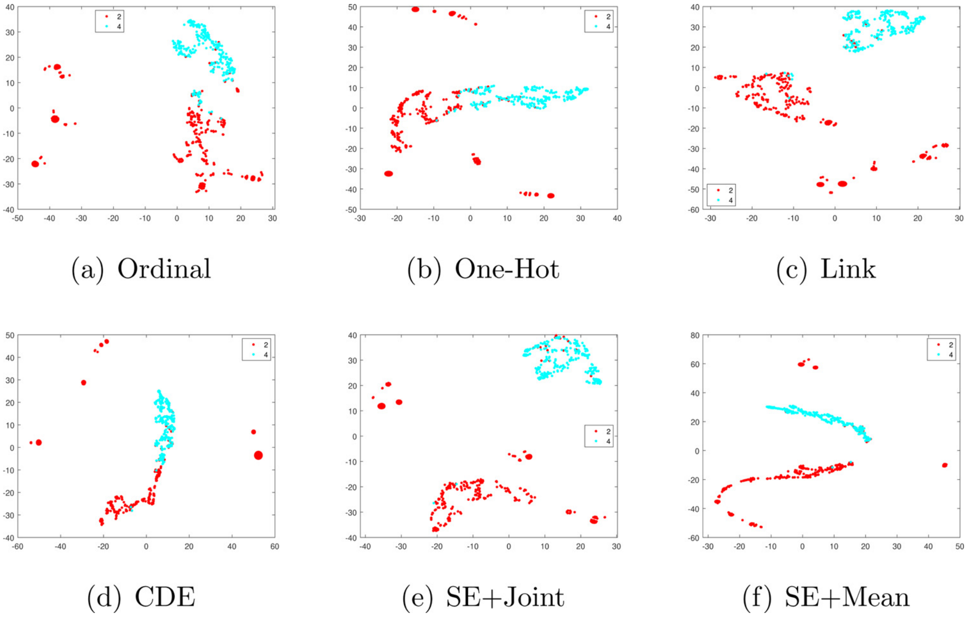
(d) CDE (e) SE+Joint (f) SE+Mean

Fig. 7. Scatter plots of different methods on Dermatology.

measure the clustering accuracy, we employ two widely-used indices, i.e., the normalized mutual information (NMI) and the adjusted rand index (ARI) [36] which measure the similarity between a clustering result and the true partition on a data set. If a clustering result is close to the true partition, then its NMI and ARI values are high. We record the mean and standard deviation of ARI and NMI of each method on each data set. Besides, we count the running time (seconds) of each compared algorithm in the task of converting categorical data.

3.2. Clustering accuracy

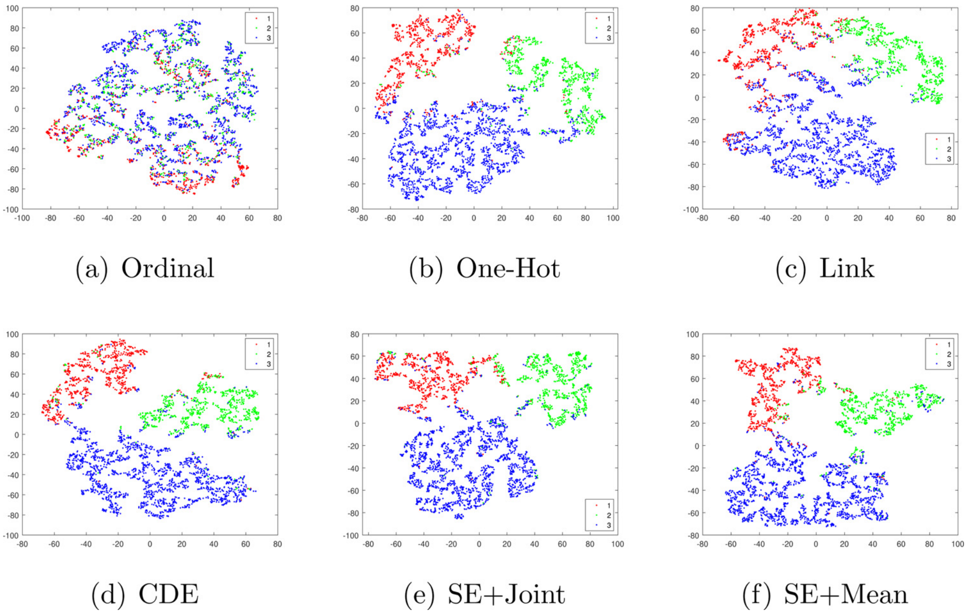
Table 4 shows the clustering performance of different graph embedding and integration methods in the proposed framework on given eight categorical data sets. In this table, we also calculate their average and maximum values on each data set. According to the table, we can see that the performance of the proposed framework based on the mean operation is similar to the joint operation. However, the computation cost of the mean operation is obviously less than the joint operation. The main reason is the Matlab environment is more suitable for the mean operation. Thus, we need to further optimize the code of the joint operation. Besides, the joint operation easily makes each of converted data become a high-dimensional data, which further adds the computation costs. In Table 4, we also can observe that the effect of the different graph embedding methods on the clustering performance. We found that the quality of the converted categorical data based on non graph embedding is not worse than that



(a) Ordinal (b) One-Hot (c) Link

(d) CDE (e) SE+Joint (f) SE+Mean

Fig. 8. Scatter plots of different methods on Breast cancer.



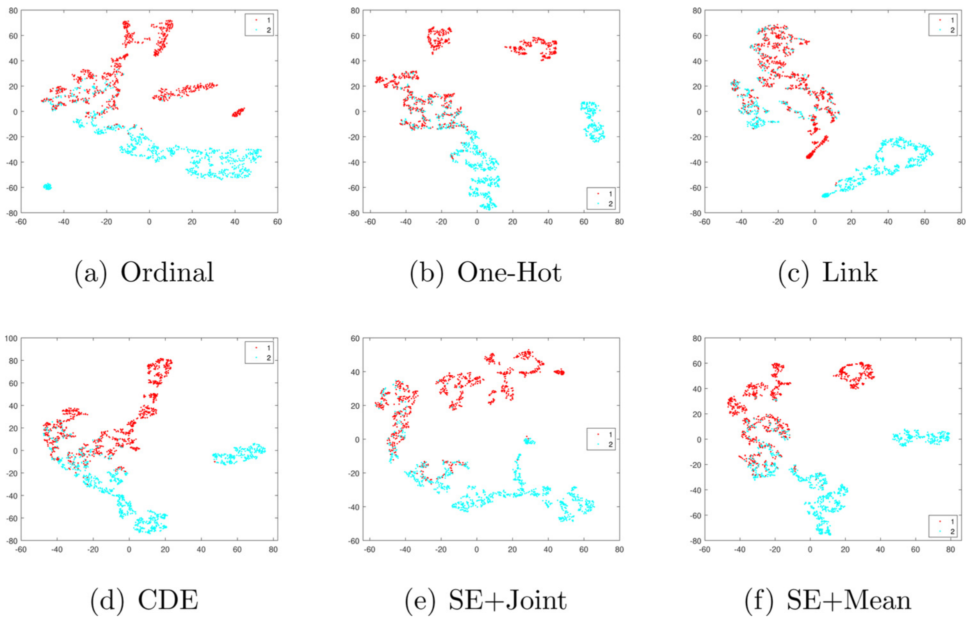
(a) Ordinal (b) One-Hot (c) Link

(d) CDE (e) SE+Joint (f) SE+Mean

Fig. 9. Scatter plots of different methods on DNA.

based on graph embedding. However, the non graph embedding method needs high computational costs, due to the high dimensionality of the converted categorical data.

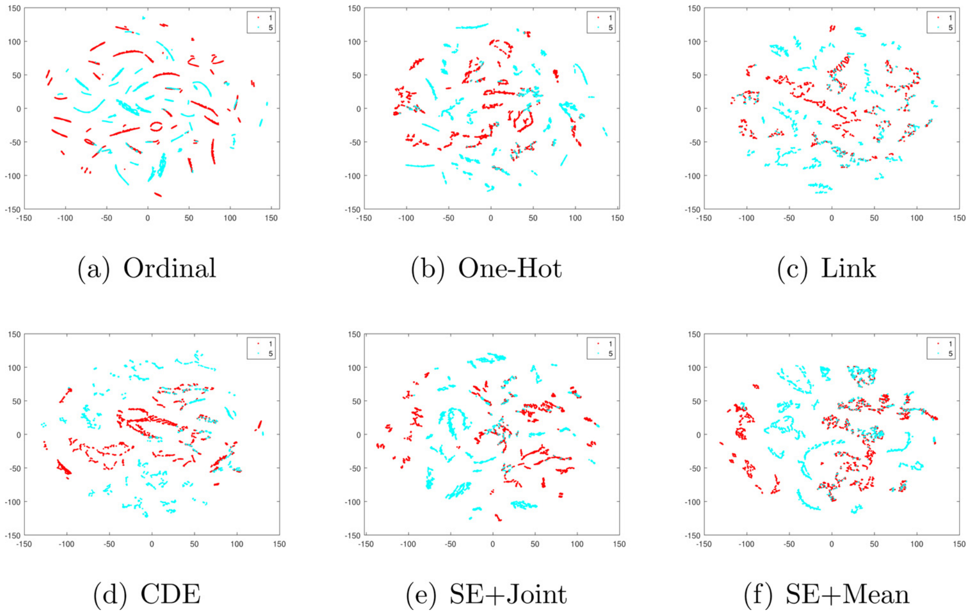
Furthermore, we compare the proposed framework based on spectral embedding (CDC_DR+SE) with other algorithms. We first show the visual results of categorical data encoding methods on different data sets, which can be seen in Figs. 4–11. These visual results were produced by using t-SNE [41] to map the encoded categorical data into 2-dimensional spaces. According to these figures, we can see that the visual results of CDC_DR+SE are better than other algorithms on most of data sets. We also can observe that the visual results of CDC_DR+SE with the joint operation are superior to the mean operation. However, since the information loss is caused by reducing dimensions, the performance of an algorithm can not be effectively evaluated in some cases. For example, on data set mushroom, we can not judge which algorithm is effective. Therefore, we need to further compare the clustering accuracy of different algorithms, which can be seen in Table 5. According to Table 5, we can observe that the clustering accuracy of the proposed framework is better than other algorithms on most of the tested data sets. The comparison results show that the proposed framework can obtain an appropriate representation for clustering categorical data in most cases. According to Tables 4 and 5, we also can observe that the maximum ARI and NMI values of the proposed framework with different graph embeddings in Table 4 are obviously superior to other algorithms in Table 5 on all the tested



(a) Ordinal (b) One-Hot (c) Link

(d) CDE (e) SE+Joint (f) SE+Mean

Fig. 10. Scatter plots of different methods on Letters.



(a) Ordinal (b) One-Hot (c) Link

(d) CDE (e) SE+Joint (f) SE+Mean

Fig. 11. Scatter plots of different methods on Mushroom.

data sets. This indicates that if we can select an appropriate graph embedding or integration method, a good representation of categorical data can be obtained.

Besides, in the experiment, we test the clustering performance of the proposed framework on clustering ensemble. We run k-means 60 times on each of given numerical data sets to get its categorical data. In order to make the categorical features as different as possible, we set different initial parameters in each time. We randomly select the number of clusters from the interval $[k/2, 2k]$ and the corresponding initial seeds from a data set. We test the proposed framework with different graph embedding and integration methods on the clustering ensemble tasks of given five data sets. The testing results are shown in Table 6. Furthermore, we select the proposed algorithm with spectral embedding to compare it with other algorithms, which is shown in Table 7. In the tables, the bold ARI and NMI values are higher than the best values of other compared algorithms. According to Tables 6 and 7, we can see that the categorical data converted by the proposed framework can further enhance the performance of clustering ensemble.

3.3. Clustering efficiency

Table 8 shows the time costs (seconds) for converting categorical data of each algorithm. We can see that the converting costs of k-modes, ordinal encoding and one-hot encoding are very low, since they need not learn the representation of categorical data. For some algorithms, e.g., Link, CDE and the proposed framework with NMF, their converting costs are very high on the tasks of clus-

Table 5
Different methods for clustering categorical data.

Dataset	Index	KModes	Ordinal	One-Hot	Link	CDE	CDC_DR+SE	
							Joint	Mean
Soybean	ARI	0.6340±0.17	0.6756±0.16	0.7263±0.2	0.6892±0.19	0.7297±0.19	0.7498±0.2	0.8572±0.19
	NMI	0.7681±0.12	0.8075±0.1	0.8460±0.11	0.8238±0.12	0.8458±0.12	0.8627±0.11	0.9254±0.1
Zoo	ARI	0.6447±0.14	0.6461±0.16	0.6393±0.14	0.654±0.13	0.6285±0.16	0.6414±0.14	0.6623±0.14
	NMI	0.7552±0.06	0.7620±0.07	0.7603±0.06	0.7554±0.07	0.7625±0.07	0.7777±0.07	0.7823±0.06
Heart disease	ARI	0.2816±0.13	0.2204±0.14	0.3784±0.11	0.0703±0.11	0.2398±0.15	0.4232±0.06	0.4208±0.00
	NMI	0.2297±0.1	0.1755±0.11	0.2988±0.09	0.0688±0.08	0.1918±0.11	0.3342±0.05	0.3297±0.00
Dermatology	ARI	0.4295±0.12	0.6568±0.13	0.652±0.17	0.5995±0.18	0.6207±0.16	0.6887±0.16	0.5449±0.09
	NMI	0.5609±0.1	0.8147±0.06	0.8007±0.09	0.7609±0.11	0.7814±0.08	0.8183±0.08	0.7279±0.03
Breast cancer	ARI	0.5655±0.3	0.8335±0.01	0.796±0.00	0.8825±0.00	0.8714±0.02	0.8988±0.00	0.9043±0.00
	NMI	0.4973±0.23	0.7293±0.01	0.7227±0.00	0.8038±0.00	0.7953±0.02	0.8269±0.00	0.8377±0.00
DNA	ARI	0.0183±0.01	0.0395±0.02	0.4759±0.11	0.3348±0.21	0.6172±0.12	0.5077±0.12	0.6799±0.13
	NMI	0.0314±0.02	0.0435±0.02	0.4529±0.08	0.3963±0.13	0.5728±0.10	0.4739±0.09	0.6344±0.11
Letters	ARI	0.1649±0.09	0.4460±0.00	0.2798±0.28	0.1160±0.18	0.3315±0.25	0.4862±0.23	0.4404±0.19
	NMI	0.1359±0.08	0.3665±0.00	0.2413±0.25	0.1122±0.17	0.3019±0.24	0.4641±0.21	0.4275±0.15
Mushroom	ARI	0.3391±0.24	0.1883±0.19	0.3895±0.25	0.1839±0.18	0.4515±0.23	0.6165±0.09	0.6070±0.09
	NMI	0.3179±0.23	0.1642±0.15	0.3871±0.21	0.2302±0.16	0.4469±0.18	0.5845±0.08	0.5734±0.08

Table 6
The proposed framework for clustering ensemble.

Dataset	Index	NE		SE		NMF		AE		Average	Max
		Joint	Mean	Joint	Mean	Joint	Mean	Joint	Mean		
Iris	ARI	0.5638±0.15	0.5907±0.12	0.663±0.11	0.6887±0.1	0.6344±0.14	0.6163±0.11	0.6753±0.12	0.7080±0.11	0.6425±0.12	0.7080±0.15
	NMI	0.6787±0.11	0.7029±0.09	0.7401±0.06	0.7658±0.07	0.7399±0.08	0.7293±0.07	0.7262±0.08	0.7757±0.08	0.7323±0.08	0.7757±0.11
Isolet	ARI	0.5384±0.04	0.5305±0.04	0.5658±0.03	0.5408±0.05	0.5527±0.03	0.5437±0.04	0.5454±0.04	0.5612±0.03	0.5473±0.04	0.5658±0.05
	NMI	0.7727±0.01	0.7694±0.01	0.7908±0.01	0.7777±0.02	0.7822±0.01	0.7765±0.02	0.7898±0.01	0.7833±0.01	0.7803±0.01	0.7908±0.02
COIL20	ARI	0.5918±0.05	0.579±0.06	0.608±0.05	0.5779±0.06	0.5995±0.04	0.5899±0.05	0.6021±0.05	0.6121±0.04	0.5950±0.05	0.6121±0.06
	NMI	0.8052±0.02	0.8052±0.02	0.8146±0.02	0.8062±0.02	0.8126±0.02	0.8124±0.02	0.8113±0.02	0.8126±0.02	0.8100±0.02	0.8146±0.02
Optical-Digits	ARI	0.6797±0.09	0.6838±0.11	0.7307±0.08	0.7285±0.07	0.7097±0.08	0.7081±0.07	0.6929±0.09	0.7253±0.06	0.7073±0.08	0.7307±0.11
	NMI	0.8024±0.04	0.8051±0.04	0.8248±0.03	0.8198±0.03	0.8212±0.03	0.8173±0.03	0.806±0.04	0.8193±0.03	0.8145±0.03	0.8248±0.04
Pen-Digits	ARI	0.5408±0.06	0.5565±0.06	0.5621±0.04	0.5481±0.05	0.5538±0.04	0.5412±0.05	0.545±0.04	0.5763±0.05	0.553±0.05	0.5763±0.06
	NMI	0.6979±0.03	0.705±0.03	0.7015±0.02	0.6923±0.03	0.6955±0.02	0.6911±0.03	0.6959±0.02	0.7098±0.02	0.6986±0.02	0.7098±0.03

Table 7
Different methods for clustering ensemble.

Dataset	Index	*KModes	Ordinal	One-Hot	Link	CDE	CDC_DR+SE	
							Joint	Mean
Iris	ARI	0.5664±0.15	0.5991±0.14	0.5464±0.17	0.6087±0.10	0.5594±0.14	0.6630±0.11	0.6887±0.10
	NMI	0.6799±0.12	0.6715±0.11	0.6547±0.14	0.7251±0.05	0.6618±0.11	0.7401±0.06	0.7658±0.07
Isolet	ARI	0.5323±0.03	0.5034±0.03	0.5277±0.04	0.5465±0.04	0.5325±0.04	0.5658±0.03	0.5408±0.05
	NMI	0.7596±0.01	0.7414±0.01	0.7623±0.01	0.7788±0.01	0.7667±0.01	0.7908±0.01	0.7777±0.02
COIL20	ARI	0.5859±0.04	0.5858±0.05	0.5826±0.05	0.5923±0.04	0.5862±0.05	0.6080±0.05	0.5779±0.06
	NMI	0.7904±0.02	0.7970±0.02	0.7962±0.02	0.8055±0.02	0.8012±0.02	0.8146±0.02	0.8062±0.02
Optical-Digits	ARI	0.6840±0.09	0.6537±0.10	0.6563±0.10	0.6859±0.10	0.6467±0.08	0.7307±0.08	0.7285±0.07
	NMI	0.7930±0.04	0.7767±0.05	0.792±0.04	0.8033±0.04	0.7852±0.03	0.8248±0.03	0.8198±0.03
Pen-Digits	ARI	0.5363±0.05	0.5439±0.06	0.5493±0.06	0.5548±0.05	0.5495±0.07	0.5621±0.04	0.5481±0.05
	NMI	0.6793±0.03	0.6898±0.03	0.7006±0.03	0.7063±0.02	0.7011±0.03	0.7015±0.02	0.6923±0.03

tering ensemble. The main reason is that their computational complexity is related to the number of the categorical values. In the clustering ensemble, if the number of clusters on a data set is not a small value, the number of the produced categorical values is very large. We can observe that the converting costs of the proposed framework based on the joint operation are higher than the mean operation, since the dimensions of the encoded objects based on the joint operation are very high, compared to the mean operation. For example, since NE does not implement the graph embedding, the number of the dimensions of the encoded data based on NE and joint is very large. Thus, we can see that its computational costs are very high, compared to the proposed framework with other graph embedding and operations. For the proposed framework with AE, its converting cost is from training the weights of the neural network. According to the above the analysis, we can see that the propose framework with spectral embedding is a good choice to convert categorical data into numerical data. It is very efficient, compared to other methods, except *k*-modes, ordinal encoding and one-hot encoding.

4. Conclusions

In this paper, we have proposed a simple categorical data clustering framework based on data representation. In this framework,

Table 8
Running time of different methods in the task of converting categorical data.

Dataset	KModes	Ordinal	One-Hot	Link	CDE	NE		SE		NMF		AE	
						Joint	Mean	Joint	Mean	Joint	Mean	Joint	Mean
Soybean	0.0000	0.0010	0.0034	0.0058	0.1166	0.0038	0.0013	0.0038	0.0026	0.0191	0.0180	4.6984	7.0965
Zoo	0.0000	0.0005	0.0029	0.0037	0.0653	0.0055	0.0017	0.0072	0.0025	0.0212	0.0203	1.6663	1.1171
Heart disease	0.0000	0.0005	0.0029	0.0048	0.0389	0.0084	0.0024	0.0056	0.0026	0.0140	0.0118	0.3443	0.3226
Dermatology	0.0000	0.0012	0.0086	0.0329	0.5235	0.1679	0.0104	0.0328	0.0195	0.1500	0.1290	6.5814	6.4826
Breast cancer	0.0000	0.0007	0.0050	0.0269	0.2378	0.0998	0.0067	0.0183	0.0101	0.0306	0.0213	1.4607	3.3474
DNA	0.0000	0.0154	0.1314	1.5587	6.3501	65.6748	0.1712	32.7012	0.2118	0.9341	0.3218	19.0252	21.8677
Letters	0.0000	0.0024	0.0181	0.2174	1.0808	2.0595	0.0316	0.1558	0.0545	0.2085	0.1608	1.3955	0.9997
Mushroom	0.0000	0.0124	0.1187	1.8680	1.6209	67.2370	0.1529	1.8635	0.1623	0.9470	0.1733	10.4491	0.9844
Iris	0.0000	0.0023	0.0083	0.0965	2.5432	0.2051	0.0146	0.0724	0.0617	0.3634	0.3418	6.2808	5.6386
Isolet	0.0000	0.0127	0.0725	21.2449	286.2975	150.1505	0.4932	2.1724	1.1184	167.7374	162.8827	56.9398	65.7873
COIL20	0.0000	0.0100	0.0653	13.5457	174.5346	105.9927	0.3413	1.4001	0.7124	89.3360	85.1458	51.5457	93.7984
OpticalDigits	0.0000	0.0346	0.2479	21.8576	49.6414	598.5708	0.5865	6.3924	0.5893	20.4497	14.4980	22.3863	20.3012
PenDigits	0.0000	0.0658	0.4844	66.0219	74.4511	2375.3709	1.2974	27.7077	1.2774	41.0335	14.8203	40.0813	21.2100

we learn the representation of each categorical value by graph embedding and integrate the represented categorical values of a data object to convert it into numerical data. Since the proposed framework fully considers the relation between categorical values, it can help existing numerical clustering algorithms to cluster categorical data and find out its potential and meaningful cluster structure. In the experimental analysis, we have compared the proposed framework with other five representation methods of categorical data on 13 benchmark data sets. The comparison results have illustrated that the proposed framework has very good performance.

In the future work, we will consider more complex or advanced similarity measures to construct the graph of categorical values. Besides, we will try more graph embedding methods to represent the categorical values. We will further analyze the effect of the similarity measures and graph embeddings on the effectiveness of the proposed framework.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work is supported by National Key Research and Development Program of China (No. 2020AAA0106100), the National Natural Science Foundation of China (No. 62022052), the Technology Research Development Projects of Shanxi (No. 201901D211192).

References

[1] A.K. Jain, Data clustering: 50 years beyond k-means, in: W. Daelemans, B. Goethals, K. Morik (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 3–4.
[2] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1967, pp. 281–297.
[3] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[4] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2001, pp. 849–856.
[5] X. Zhu, Y. Zhu, W. Zheng, Spectral rotation for deep one-step clustering, Pattern Recognit. 105 (2020) 107175.
[6] L. Guo, Q. Dai, Graph clustering via variational graph embedding, Pattern Recognit. 122 (2022) 108334.
[7] L. Romeo, E. Frontoni, A unified hierarchical XGBoost model for classifying priorities for COVID-19 vaccination campaign, Pattern Recognit. 121 (2022) 108197.
[8] A. Nazabal, P.M. Olmos, Z. Ghahramani, I. Valera, Handling incomplete heterogeneous data using VAEs, Pattern Recognit. 107 (2020) 107501.
[9] S.-K. Ng, R. Tawiah, G.J. McLachlan, Unsupervised pattern recognition of mixed data structures with numerical and categorical features using a mixture regression modelling framework, Pattern Recognit. 88 (2019) 261–271.
[10] R. Kuo, Y. Zheng, T.P.Q. Nguyen, Metaheuristic-based possibilistic fuzzy k-modes algorithms for categorical data clustering, Inf. Sci. 557 (2021) 1–15.
[11] L. Bai, J. Liang, Cluster validity functions for categorical data: a solution-space perspective, Data Min. Knowl. Discov. 29 (6) (2015) 1560–1597.
[12] S. Guha, R. Rastogi, S. Kyuseok, Rock: a robust clustering algorithm for categorical attributes, in: Proceedings of the Fifteenth International Conference on Data Engineering, 23–26, Sydney, Australia, 1999, pp. 512–521.
[13] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Min. Knowl. Discov. 2 (3) (1998) 283–304.
[14] M. Ng, M.J. Li, Z.X. Huang, Z. He, On the impact of dissimilarity measure in *k*-modes clustering algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 29 (3) (2007) 503–507.
[15] J. Huang, M. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[16] L. Bai, J. Liang, C. Dang, F. Cao, The impact of cluster representatives on the convergence of the k-modes type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 35 (6) (2013) 1509–1522.
[17] Y. Xiao, C. Huang, J. Huang, I. Kaku, Y. Xu, Optimal mathematical programming and variable neighborhood search for k-modes categorical data clustering, Pattern Recognit. 90 (2019) 183–195.
[18] S. Boriah, V. Chandola, V. Kumar, Similarity measures for categorical data: a comparative evaluation, in: Proceedings of the SIAM International Conference on Data Mining, 2008.
[19] D. Fisher, Knowledge acquisition via incremental conceptual clustering, Mach. Learn. 2 (2) (1987) 139–172.
[20] J. Wu, H. Liu, H. Xiong, J. Cao, J. Chen, K-means-based consensus clustering: a unified view, IEEE Trans. Knowl. Data Eng. 27 (1) (2015) 155–169.
[21] H. Liu, J. Wu, T. Liu, D. Tao, Y. Fu, Spectral ensemble clustering via weighted k-means: theoretical and practical evidence, IEEE Trans. Knowl. Data Eng. 29 (5) (2017) 1129–1143.
[22] M.A. Gluck, J.E. Corter, Information uncertainty and the utility of categories, in: Proceedings of the Seventh Annual Conference of Cognitive Science Society, 1985, pp. 283–287.
[23] D. Barbara, Y. Li, J. Couto, COOLCAT: an entropy-based algorithm for categorical clustering, in: Proceedings of the Eleventh International Conference on Information and Knowledge Management, 2002, pp. 582–589.
[24] K. Chen, L. Liu, HE-Tree: a framework for detecting changes in clustering structure for categorical data streams, VLDB J. 18 (5) (2009) 1241–1260.
[25] L. Bai, J. Liang, H. Du, Y. Guo, An information-theoretical framework for cluster ensemble, IEEE Trans. Knowl. Data Eng. 31 (2019) 1464–1477.
[26] H. Ralambondrainy, A conceptual version of the k-means algorithm, Pattern Recognit. Lett. 16 (11) (1995) 1147–1157.
[27] N. Iam-On, T. Boongoen, S. Garrett, C. Price, A link-based approach to the cluster ensemble problem, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 2396–2409.
[28] N. Iam-On, T. Boongeon, S. Garrett, C. Price, A link-based cluster ensemble approach for categorical data clustering, IEEE Trans. Knowl. Data Eng. 24 (3) (2012) 413–425.
[29] S. Jian, L. Cao, G. Pang, K. Lu, H. Gao, Embedding-based representation of categorical data by hierarchical value coupling learning, in: International Joint Conference on Artificial Intelligence, 2017.
[30] S. Jian, G. Pang, L. Cao, K. Lu, H. Gao, CURE: flexible categorical data representation by hierarchical coupling learning, IEEE Trans. Knowl. Data Eng. 31 (5) (2019) 853–866.
[31] C. Zhu, L. Cao, J. Yin, Unsupervised heterogeneous coupling learning for categorical representation, IEEE Trans. Pattern Anal. Mach. Intell. 44 (1) (2022) 533–549.

[32] Q. Zheng, X. Diao, J. Cao, Y. Liu, H. Li, J. Yao, C. Chang, G. Lv, From whole to part: reference-based representation for clustering categorical data, IEEE Trans. Neural Netw. Learn. Syst. 31 (3) (2020) 927–937.

[33] S. Jian, L. Cao, K. Lu, H. Gao, Unsupervised coupled metric similarity for non-IID categorical data, IEEE Trans. Knowl. Data Eng. 30 (9) (2018) 1810–1823.

[34] Y. Zhang, Y.-m. Cheung, Learnable weighting of intra-attribute distances for categorical data clustering with nominal and ordinal attributes, IEEE Trans. Pattern Anal. Mach. Intell. (2021), doi:10.1109/TPAMI.2021.3056510. 1–1

[35] E.J. Rivera Rios, M.A. Medina-Perez, M.S. Lazo-Cortes, R. Monroy, Learning-based dissimilarity for clustering categorical data, Appl. Sci. 11 (8) (2021) 3509.

[36] Data Clustering: Algorithms and Applications, C.C. Aggarwal, C.K. Reddy (Eds.), CRC Press, 2014.

[37] D.D. Lee, H.H. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556–562.

[38] G. Hinton, R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[39] A. Topchy, A. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (12) (2005) 1866–1881.

[40] A. Fred, A. Jain, Data clustering using evidence accumulation, in: 16th International Conference on Pattern Recognition, 2002.

[41] V.D.M. Laurens, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2605) (2008) 2579–2605.

Liang Bai received his PhD degree in Computer Science from Shanxi University in 2012. He is currently a Professor with the School of Computer and Information Technology, Shanxi University. His research interest is in the areas of cluster analysis. He has published several papers in his research fields, including IEEE TPAMI, IEEE TKDE, DMKD, IEEE TFS, PR, ICML and AAAI.

Jiye Liang received the PhD degree from Xi'an Jiaotong University. He is a professor in Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, the School of Computer and Information Technology, Shanxi University. His research interests include artificial intelligence, granular computing, data mining, and machine learning. He has published more than 150 papers in his research fields, including IEEE TPAMI, IEEE TKDE, IEEE TFS, DMKD and AI.