Neurocomputing 341 (2019) 118–134

Contents lists available at ScienceDirect

Neurocomputing

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

Hierarchical division clustering framework for categorical data

Wei Wei[a], Jiye Liang[a], Xinyao Guo[a], Peng Song[a,b,*], Yijun Sun[c,d]

[a] 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
[b] School of Management, Shanxi University, Taiyuan, Shanxi 030006, China
[c] Department of Microbiology and Immunology, State University of New York at Buffalo, Buffalo, NY 14201, USA
[d] Department of Computer Science and Engineering, Department of Biostatistics, State University of New York at Buffalo, Buffalo, NY 14201, USA

ARTICLE INFO

Article history:
Received 8 July 2018
Revised 3 January 2019
Accepted 22 February 2019
Available online 4 March 2019

Communicated by Dr Weiguo Sheng

Keywords:
Rough set
Categorical data
Hierarchical clustering
Divisive clustering

ABSTRACT

Although many divisive hierarchical clustering methods for processing categorical data have been presented in the literature, none have been systematically or comprehensively investigated. This paper presents a systematic analysis of existing methods, with respective advantages and disadvantages summarized to develop a unified divisive hierarchical clustering framework that follows three general steps: (1) select attributes for splitting a selected cluster; (2) based on these attributes, generate bipartitions of the cluster; and (3) determine which of the resulting clusters should be further split. Using the proposed framework, representative existing algorithms are compared, and better-performing algorithms are produced through improvements relevant to each step of the unified framework. Experimental results on fifteen UCI benchmark datasets reveal that application of the proposed framework significantly improves the clustering performance of a number of algorithms relative to baseline.

1. Introduction

Clustering is an important technique in data mining and machine learning, in which a dataset can be divided into multiple clusters without the use of prior knowledge. The goal of clustering is to group a dataset into clusters so that the objects in the same cluster have high similarity but are very dissimilar with objects in other clusters [1]. Clustering is therefore often used to uncover the inherent structure of a dataset [2] by, for example, grouping objects into clusters in which the objects are similar to each other or to a centroid [3]. A large number of clustering algorithms have been proposed, and these can be classified as partitional, hierarchical, density-based, grid-based, and model-based clustering [4,5]. Among these, partitional and hierarchical clustering are the most popular. Briefly, partitional clustering algorithms, in general, preset a given number of clusters and, essentially, seek a single partition of a dataset. Hierarchical clustering algorithms, essentially heuristic procedures, produce a hierarchy of partitions of a dataset, according to an agglomerative strategy or to a divisive one, and then a dendrogram (a tree-like nested structure) is built.

A number of clustering algorithms were originally developed for processing numerical data [4] and that is represented by K-means algorithm [6]. However, a large number of categorical data exist in many real applications, such as market basket data analysis, DNA or protein sequence analysis and text mining [7]. As there is no inherent distance measure between categorical objects, clustering categorical data is more challenging than clustering numerical data [8–10] and clustering algorithms developed for managing numerical data cannot directly be used to cluster categorical data. To address this deficiency, several clustering algorithms have been developed to deal with categorical data, in which k-modes algorithm [11,12], its several modified versions [13–15] and COOLCAT [16] belong to partitional clustering, and ROCK [17], COBWEB [18], Chameleon [19] and LIMBO [20] belong to hierarchical clustering.

In this study, we focus solely on hierarchical clustering for categorical data. We thus review some references on hierarchical clustering, which can be divided into two main types: agglomerative and divisive methods. Agglomerative hierarchical clustering follows a “bottom-up” style in which pairs of individual clusters are successively merged in a stepwise manner. Many agglomerative clustering algorithms, including CURE [21], ROCK [17], Chameleon [19], and PHA [22], have been successfully applied in the fields of information retrieval [23], image processing [24], recommendation systems [25], and bio-informatics [26–28]. The most representative agglomerative clustering algorithms are single-link [29], complete-link [30], and average-link [31] algorithms, which primarily

* Corresponding author at: School of Management, Shanxi University, Taiyuan, Shanxi 030006, China.
E-mail addresses: weiwei@sxu.edu.cn (W. Wei), ljy@sxu.edu.cn (J. Liang), 1303590343@qq.com (X. Guo), songpeng@sxu.edu.cn (P. Song), yijunsun@uffalo.edu (Y. Sun).

https://doi.org/10.1016/j.neucom.2019.02.043
0925-2312/

Table 1
Eight *UCI* datasets with two classes.

Dataset	Vote	Cancer	Mushroom	Chess	Balloon	Lymphography	Promoters	Space
Number of objects	435	699	5644	3196	20	142	106	15
Number of attributes	16	9	22	36	4	18	57	6
Number of classes	2	2	2	2	2	2	2	2

Table 2
Seven *UCI* datasets with more than two classes.

Dataset	Soybean	Zoo	Car	Nursery	Balance	Hayes-Roth	Lenses
Number of objects	47	101	1728	12960	625	132	24
Number of attributes	21	16	6	8	4	4	4
Number of classes	4	7	4	5	3	3	3

Table 3
Comparative analyses of existing algorithms using the proposed framework.

Algorithm	Algorithm TR	Algorithm MMR	Algorithm MDA	Algorithm MGR
Step 1	Total Roughness (*TR*)	Min-Min-Roughness(*MMR*)	Max Dependency of Attributes (*MDA*)	Mean information Gain Ratio (*MGR*)
Step 2	N/A	Overall Roughness(*OR*)	N/A	Information Entropy (*IE*)
Step 3	N/A	Number of Objects (*NO*)	N/A	Information Entropy (*IE*)

⋆ "N/A" indicates that there is no the step in an algorithm.

differ in their definitions of the distance between pairs of clusters [2]. A number of agglomerative hierarchical clustering algorithms represent variants of these three algorithm types obtained by using different metrics to evaluate the distance between pairs of clusters [32,33]. Divisive method algorithms, by contrast, employ a top-down style in which the data objects are initially treated as a unified cluster that is gradually split until the desired number of clusters is obtained [34–36]. Although it is natural to attempt to analyze all possible bipartitions by which a cluster can be divided into two sub-clusters, as complete enumeration of these would obviously enable the determination of a global optimum; however, this is generally computationally prohibitive. Thus, a number of divisive clustering methods that do not consider all bipartitions have been proposed. One representative divisive clustering algorithm is the bisecting k-means method, which can produce results more accurately than either the k-means or agglomerative clustering approaches [37]. Zhong et al. [5] proposed a novel reference-point-based dissimilarity measure (DIVFRP) and incorporated it into a divisive hierarchical clustering algorithm for splitting datasets. Feng et al. [38] introduced an improved particle swarm optimizer (IDPSO) to find a near-optimal partitioning hyperplane for splitting selected clusters into two smaller clusters. Divisive hierarchical cluster methods that use this splitting technique are both efficient and effective. However, although divisive clustering is attractive in terms of computational time, clustering quality is generally worse than that of partitional clustering.

The problem of clustering categorical data has also received much attention recently, leading to the proposal of a number of divisive hierarchical clustering algorithms. Mazlack et al. [39] proposed a bi-clustering method for selecting multi-valued attributes based on two-valued attribution and a Total Roughness (TR) technique and suggested that attributes with higher TR are better suited to successfully splitting clusters. In [40], the Min-Min-roughness (MMR) measure was developed to address uncertainty in the process of clustering categorical data. However, MMR simply represents the opposite of TR and does not produce clustering algorithms with any relative advantage with respect accuracy or complexity [41,42]. Xiong et al. [43] proposed a divisive hierarchical clustering algorithm for categorical data based on Multiple Correspondence Analysis (MCA). Herawan et al. [42] proposed Maximum Dependency of Attributes (MDA) to select attributes used for divisive hierarchical clustering. MDA is constructed based on the dependency of attributes in rough set theory and is used to evaluate the dependency of one attribute on the other attributes in a dataset. Qin et al. [44] proposed an information-theory-based hierarchical divisive clustering algorithm for categorical data that is implemented by selecting a clustering attribute using the Mean Gain Ratio (MGR) and then selecting an equivalence class on the clustering attribute using cluster entropy. Although algorithms such as MDA and MGR achieve good clustering performance, there is still no unified framework for hierarchical division clustering. This gap has, to a certain degree, limited the growth of divisive hierarchical clustering algorithm performance, and designing a framework for the steps in divisive clustering has become an issue of some urgency.

To solve this problem, in this study, we use existing divisive clustering algorithms as an inspiration to introduce a uniform framework of hierarchical division clustering for use in the general analysis of existing algorithms and the design of new, better-performing algorithms. Our framework comprises three main steps: (1) selecting several attributes for splitting a cluster; (2) splitting the cluster based on the selected attributes; and (3) determining which of the resulting clusters should be further split. In the first step, several informative attributes are selected to generate candidate partitions of a selected cluster. In the second step, several appropriate partitions are selected from the candidate partitions using an evaluation method. Application of the first and second steps produces a bipartition of a cluster, since any given number of clusters can be reached by recursively running a divisive bisecting clustering procedure. In the third step, one of the two clusters resulting from the bipartition is selected for application of the next iteration of splitting. The contributions of this study include the following: (1) We propose a general divisive hierarchical clustering framework that is used to systematically analyze a number of existing algorithms and to construct new,

Table 4
Ranking attributes based on their *HCA* in datasets with two-classes.

Dataset																	
Vote	*Attribute*	4	3	5	12	8	9	14	7	13	15	16	1	6	11	2	10
	HCA	0.956	0.874	0.848	0.841	0.832	0.812	0.770	0.761	0.759	0.731	0.701	0.687	0.674	0.641	0.614	0.614
Cancer	*Attribute*	2	3	6	7	5	8	4	1	9							
	HCA	0.927	0.923	0.911	0.907	0.896	0.896	0.864	0.861	0.790							
Mushroom	*Attribute*	5	20	12	19	13	10	14	15	9	4	22	3	8	21	18	6
	HCA	0.984	0.911	0.854	0.854	0.848	0.790	0.776	0.768	0.748	0.725	0.702	0.699	0.658	0.630	0.629	0.621
	Attribute	17	1	2	7	11	16										
	HCA	0.619	0.619	0.619	0.618	0.618	0.618										
Chess	*Attribute*	10	33	21	8	7	35	6	18	32	15	13	16	27	9	23	3
	HCA	0.683	0.682	0.661	0.609	0.585	0.575	0.568	0.568	0.565	0.556	0.554	0.554	0.549	0.547	0.540	0.538
	Attribute	29	31	11	22	14	5	25	28	1	2	4	12	17	19	20	24
	HCA	0.537	0.534	0.531	0.527	0.527	0.526	0.524	0.523	0.522	0.522	0.522	0.522	0.522	0.522	0.522	0.522
	Attribute	26	30	34	36												
	HCA	0.522	0.522	0.522	0.522												
Shuttle	*Attribute*	5	2	1	3	4	6										
	HCA	0.867	0.800	0.667	0.667	0.600	0.600										
Lymphography	*Attribute*	13	2	18	15	10	14	8	7	12	16	1	4	11	3	5	6
	HCA	0.789	0.739	0.739	0.711	0.690	0.648	0.641	0.599	0.599	0.585	0.578	0.578	0.578	0.570	0.570	0.570
	Attribute	9	17														
	HCA	0.570	0.570														
Promoters	*Attribute*	15	16	17	39	6	18	20	41	49	38	40	8	10	19	42	31
	HCA	0.802	0.802	0.793	0.755	0.708	0.689	0.679	0.679	0.670	0.660	0.660	0.651	0.651	0.651	0.651	0.642
	Attribute	7	9	14	30	46	11	33	43	48	51	2	26	27	5	24	32
	HCA	0.632	0.623	0.623	0.623	0.623	0.604	0.604	0.604	0.604	0.604	0.594	0.594	0.594	0.585	0.585	0.585
	Attribute	45	12	13	21	35	37	52	53	54	57	23	47	50	22	28	34
	HCA	0.585	0.578	0.578	0.578	0.578	0.578	0.578	0.578	0.578	0.578	0.566	0.566	0.557	0.547	0.547	0.547
	Attribute	44	55	56	25	29	36	1	3	4							
	HCA	0.547	0.547	0.547	0.538	0.538	0.538	0.528	0.528	0.519							
Balloon	*Attribute*	1	2	3	4												
	HCA	0.800	0.800	0.600	0.600												

Table 5
Ranking attributes based on their HARI in datasets with two-classes.

Dataset																	
Vote	*Attribute*	4	3	5	12	8	9	14	7	13	15	1	16	6	11	10	2
	HARI	0.832	0.556	0.484	0.464	0.440	0.387	0.290	0.270	0.266	0.212	0.138	0.116	0.076	0.614	0.004	0.004
Cancer	*Attribute*	2	3	6	7	5	8	4	1	9							
	HARI	0.725	0.711	0.674	0.657	0.623	0.620	0.518	0.511	0.306							
Mushroom	*Attribute*	5	20	12	19	13	10	14	15	9	4	3	22	8	21	11	18
	HARI	0.938	0.674	0.493	0.493	0.475	0.337	0.287	0.269	0.236	0.202	0.144	0.141	0.070	0.065	0.039	0.017
	Attribute	2	6	1	17	16	7										
	HARI	0.015	0.004	0.003	0.002	0.000	−0.021										
Chess	*Attribute*	10	33	21	8	7	35	6	18	32	15	13	16	9	27	23	3
	HARI	0.134	0.133	0.102	0.046	0.028	0.022	0.018	0.018	0.015	0.011	0.011	0.010	0.008	0.008	0.005	0.004
	Attribute	29	31	11	5	22	14	24	4	25	34	28	1	2	36	26	17
	HARI	0.004	0.004	0.003	0.002	0.002	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Attribute	19	20	12	30												
	HARI	−0.001	−0.001	−0.001	−0.001												
Shuttle	*Attribute*	5	2	1	3	4	6										
	HARI	0.504	0.315	0.067	0.067	−0.031	−0.045										
Lymphography	*Attribute*	13	2	18	15	10	14	8	12	7	16	11	1	4	9	6	17
	HARI	0.329	0.224	0.223	0.173	0.139	0.080	0.071	0.031	0.022	0.020	0.017	0.014	0.006	0.001	−0.004	−0.005
	Attribute	5	3														
	HARI	−0.010	−0.012														
Promoters	*Attribute*	15	16	17	39	6	18	41	20	49	40	38	10	8	19	42	31
	HARI	0.358	0.358	0.336	0.253	0.165	0.134	0.121	0.120	0.107	0.095	0.094	0.083	0.083	0.083	0.082	0.072
	Attribute	7	46	9	14	30	11	43	33	51	48	2	26	27	32	5	45
	HARI	0.062	0.052	0.052	0.051	0.051	0.035	0.035	0.034	0.034	0.034	0.029	0.026	0.026	0.022	0.021	0.020
	Attribute	24	52	37	21	54	13	57	12	53	35	23	47	50	28	55	22
	HARI	0.020	0.019	0.017	0.017	0.017	0.015	0.015	0.014	0.014	0.013	0.010	0.008	0.004	0.004	0.002	0.001
	Attribute	56	29	44	34	25	36	1	4	3							
	HARI	0.001	0.000	0.000	0.000	−0.002	−0.004	−0.004	−0.006	−0.006							
Balloon	*Attribute*	1	2	3	4												
	HARI	0.326	0.326	−0.053	−0.053												

Table 6
Top attribute by measure (TR, MMR, MDA, and MGR) and respective HCAs and HARIs.

Measures		*Vote*	*Cancer*	*Mushroom*	*Chess*	*Shuttle*	*Lymphography*	*Promoters*	*Balloon*
	Top 1 *attribute*	1	9	17	28	6	9	1	1
TR	*HCA*	0.687	0.790	0.619	0.523	0.600	0.570	0.528	**0.800**
	HARI	0.138	0.306	0.002	0.000	−0.045	0.001	−0.004	**0.326**
	Top 1 *attribute*	1	9	17	28	3	9	1	1
MMR	*HCA*	0.687	0.790	0.619	0.523	**0.667**	0.570	0.528	**0.800**
	HARI	0.138	0.306	0.002	0.000	**0.067**	0.001	−0.004	**0.326**
	Top 1 *attribute*	1	9	17	28	6	9	1	1
MDA	*HCA*	0.687	0.790	0.619	0.523	0.600	0.570	0.528	**0.800**
	HARI	0.138	0.306	0.002	0.000	−0.045	0.001	−0.004	**0.326**
	Top 1 *attribute*	5	2	6	14	3	4	16	1
MGR	*HCA*	**0.848**	**0.927**	**0.621**	**0.527**	**0.667**	**0.578**	**0.802**	**0.800**
	HARI	**0.484**	**0.725**	**0.004**	**0.001**	**0.067**	**0.006**	**0.358**	**0.326**

Table 7
CAs and ARIs obtained by using different measures in Step 2 on datasets in Table 1.

Index	Step 1	Step 2	*Vote*	*Cancer*	*Mushroom*	*Chess*	*Shuttle*	*Lymphography*	*Promoters*	*Balloon*
AC	*MMR*	*OR*	0.614	0.790	0.619	0.523	0.600	0.570	0.519	**0.800**
		IE	0.674	0.660	0.619	0.523	0.667	0.570	0.509	**0.800**
	MGR	*OR*	0.614	**0.883**	**0.621**	**0.527**	0.600	**0.577**	**0.604**	**0.800**
		IE	**0.828**	**0.883**	**0.621**	**0.527**	**0.667**	**0.577**	0.566	**0.800**
ARI	*MMR*	*OR*	−0.007	0.306	0.002	0.000	−0.031	0.001	−0.006	**0.326**
		IE	0.118	0.008	0.002	0.000	**0.067**	0.001	−0.004	**0.326**
	MGR	*OR*	−0.011	**0.585**	**0.004**	**0.001**	−0.031	**0.006**	**0.038**	**0.326**
		IE	**0.428**	**0.585**	**0.004**	**0.001**	**0.067**	**0.006**	0.014	**0.326**

Table 8
CAs and ARIs obtained using different measures in Step 3 on datasets in Table 2.

Index	Step 1	Step 2	Step 3	*Soybean*	*Zoo*	*Car*	*Nursery*	*Balance*	*Hayes-Roth*	*Lenses*
AC	*MMR*	*OR*	*NO*	**0.830**	0.911	**0.700**	**0.486**	**0.635**	0.386	**0.625**
		OR	*IE*	**0.830**	0.911	**0.700**	**0.486**	**0.635**	0.386	**0.625**
	MGR	*OR*	*NO*	**0.830**	0.832	**0.700**	**0.486**	**0.635**	**0.500**	**0.625**
		OR	*IE*	**0.830**	**0.921**	**0.700**	**0.486**	**0.635**	**0.500**	**0.625**
ARI	*MMR*	*OR*	*NO*	**0.674**	0.913	**0.013**	**0.069**	**0.101**	−0.015	**−0.043**
		OR	*IE*	**0.674**	0.913	**0.013**	**0.069**	**0.101**	−0.015	**−0.043**
	MGR	*OR*	*NO*	**0.674**	0.716	**0.013**	**0.069**	**0.101**	**0.066**	**−0.043**
		OR	*IE*	**0.674**	**0.956**	**0.013**	0.061	**0.101**	**0.066**	**−0.043**

better-performing algorithms. (2) We recommend some good measures in the steps of the proposed framework to construct high-performing algorithms. (3) Based on an assessment of the shortcomings of MGR, we develop the mean normalized information gain (MNIG) algorithm to sort attributes and thereby enhance the quality of selected attributes. (4) We propose a cluster-splitting method that, instead of the top attribute used in existing divisive hierarchical clustering algorithms, chooses the top k attributes in each iteration to split a cluster.

The rest of this paper is organized as follows. Section 2 presents some preliminaries on the rough set concept and some representative divisive hierarchical clustering algorithms for use with categorical data. In Section 3, a unified framework of divisive hierarchical clustering is introduced. In Section 4, our proposed framework is used to analyze the advantages and disadvantages of the representative algorithms discussed in Section 2. In Section 5, a new measure is presented for application in Step 1 of the proposed framework, and new methods are proposed for Steps 2 and 3, respectively. In Section 6, we discuss the results of extensive experiments carried out to validate the performance of algorithms constructed based on the proposed framework. Section 7 concludes the paper.

2. Preliminary

2.1. Rough set basics

A categorical dataset can be described as an information table in rough set theory comprising the 4-tuple $S = (U, A, V, f)$ (for short $S = (U, A)$). In this information table, U is a non-empty and finite set of objects called a universe, A is a non-empty and finite set of attributes, V_a is the domain of attributes a, where $V = \bigcup_{a \in A} V_a$m, and $f : U \times A = V$ is a function $f(x, a) \in V_a$ for each $a \in A$ [45–47].

An indiscernibility relation $R_B = \{(x, y) \in U \times U \mid f(x, a) = f(y, a), \forall a \in B\}$ can be determined by a non-empty subset $B \subseteq A$. $U/R_B = \{[x]_B \mid x \in U\}$ (just as U/B) indicates the partition of U resulting from R_B, where $[x]_B$ denotes the equivalence class determined by x with respect to B, i.e., $[x]_B = \{y \in U \mid (x, y) \in R_B\}$.

Table 9
MGR values of all attributes in Dataset "*Lymphography*".

Attributes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	*MGR*	*Rank*
1	1.000	0.001	0.001	0.006	0.019	0.010	0.002	0.008	0.009	0.051	0.020	0.013	0.022	0.111	0.010	0.031	0.003	0.065	0.022	18
2	0.002	1.000	0.161	0.030	0.117	0.095	0.001	0.000	0.017	0.015	0.065	0.055	0.120	0.034	0.049	0.006	0.000	0.019	0.046	13
3	0.003	0.242	1.000	0.145	0.299	0.144	0.061	0.001	0.006	0.009	0.011	0.021	0.082	0.032	0.007	0.006	0.022	0.086	0.069	5
4	0.040	0.136	0.433	1.000	0.359	0.168	0.158	0.004	0.003	0.190	0.060	0.101	0.150	0.228	0.062	0.092	0.052	0.442	0.157	1
5	0.038	0.150	0.255	0.103	1.000	0.174	0.070	0.006	0.007	0.043	0.048	0.048	0.053	0.048	0.007	0.039	0.021	0.092	0.071	4
6	0.015	0.094	0.094	0.037	0.134	1.000	0.004	0.002	0.014	0.044	0.057	0.036	0.010	0.089	0.030	0.009	0.031	0.084	0.046	14
7	0.011	0.006	0.157	0.137	0.213	0.015	1.000	0.009	0.003	0.124	0.004	0.023	0.157	0.142	0.073	0.016	0.054	0.346	0.088	3
8	0.014	0.000	0.001	0.001	0.005	0.002	0.003	1.000	0.003	0.161	0.033	0.017	0.130	0.072	0.114	0.045	0.032	0.177	0.048	11
9	0.124	0.159	0.036	0.007	0.049	0.131	0.008	0.020	1.000	0.546	0.003	0.077	0.250	0.226	0.097	0.013	0.042	0.391	0.128	2
10	0.046	0.009	0.004	0.025	0.020	0.026	0.019	0.082	0.035	1.000	0.078	0.056	0.087	0.193	0.121	0.062	0.039	0.222	0.066	6
11	0.027	0.059	0.007	0.012	0.034	0.052	0.001	0.026	0.000	0.120	1.000	0.131	0.062	0.071	0.098	0.014	0.005	0.040	0.045	15
12	0.012	0.033	0.008	0.014	0.023	0.022	0.004	0.009	0.005	0.057	0.087	1.000	0.186	0.097	0.077	0.049	0.036	0.095	0.048	12
13	0.022	0.076	0.035	0.021	0.026	0.007	0.025	0.072	0.017	0.093	0.043	0.196	1.000	0.123	0.097	0.022	0.010	0.108	0.059	7
14	0.068	0.013	0.008	0.020	0.015	0.036	0.014	0.025	0.010	0.129	0.031	0.063	0.077	1.000	0.034	0.030	0.031	0.126	0.043	16
15	0.010	0.033	0.003	0.009	0.004	0.020	0.013	0.067	0.007	0.140	0.073	0.087	0.104	0.059	1.000	0.007	0.031	0.171	0.049	10
16	0.051	0.007	0.004	0.022	0.033	0.010	0.004	0.043	0.002	0.113	0.017	0.088	0.038	0.082	0.012	1.000	0.085	0.117	0.043	17
17	0.006	0.000	0.020	0.016	0.023	0.044	0.019	0.038	0.006	0.092	0.008	0.082	0.023	0.109	0.062	0.108	1.000	0.279	0.055	9
18	0.041	0.008	0.024	0.041	0.030	0.035	0.037	0.064	0.017	0.156	0.018	0.065	0.071	0.132	0.104	0.045	0.084	1.000	0.057	8

Table 10
NMIG values of all attributes in Dataset "*Lymphography*".

Attributes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	*NMIG*	*Rank*
1	1.000	0.002	0.002	0.010	0.025	0.012	0.003	0.010	0.016	0.048	0.023	0.012	0.022	0.084	0.010	0.038	0.004	0.051	1.372	17
2	0.002	1.000	0.193	0.049	0.131	0.095	0.002	0.000	0.031	0.011	0.062	0.041	0.093	0.019	0.039	0.007	0.000	0.011	1.786	10
3	0.002	0.193	1.000	0.217	0.275	0.114	0.087	0.001	0.010	0.005	0.008	0.012	0.049	0.013	0.005	0.005	0.021	0.037	2.054	5
4	0.010	0.049	0.217	1.000	0.160	0.060	0.147	0.002	0.004	0.044	0.020	0.024	0.037	0.037	0.016	0.036	0.024	0.074	1.961	6
5	0.025	0.131	0.275	0.160	1.000	0.152	0.105	0.005	0.012	0.027	0.039	0.031	0.035	0.023	0.005	0.035	0.022	0.045	2.127	3
6	0.012	0.095	0.114	0.060	0.152	1.000	0.006	0.002	0.025	0.033	0.054	0.027	0.008	0.051	0.024	0.009	0.036	0.049	1.757	11
7	0.003	0.002	0.087	0.147	0.105	0.006	1.000	0.004	0.005	0.032	0.002	0.006	0.044	0.026	0.022	0.007	0.028	0.066	1.592	14
8	0.010	0.000	0.001	0.002	0.005	0.002	0.004	1.000	0.005	0.109	0.029	0.012	0.092	0.037	0.084	0.044	0.035	0.094	1.565	16
9	0.016	0.031	0.010	0.004	0.012	0.025	0.005	0.005	1.000	0.065	0.001	0.009	0.032	0.019	0.013	0.003	0.011	0.033	1.294	18
10	0.048	0.011	0.005	0.044	0.027	0.033	0.032	0.109	0.065	1.000	0.094	0.056	0.090	0.155	0.130	0.080	0.055	0.183	2.217	2
11	0.023	0.062	0.008	0.020	0.039	0.054	0.002	0.029	0.001	0.094	1.000	0.105	0.051	0.043	0.084	0.016	0.006	0.025	1.662	12
12	0.012	0.041	0.012	0.024	0.031	0.027	0.006	0.012	0.009	0.056	0.105	1.000	0.191	0.077	0.082	0.063	0.050	0.077	1.875	8
13	0.022	0.093	0.049	0.037	0.035	0.008	0.044	0.092	0.032	0.090	0.051	0.191	1.000	0.095	0.101	0.028	0.014	0.085	2.067	4
14	0.084	0.019	0.013	0.037	0.023	0.051	0.026	0.037	0.019	0.155	0.043	0.077	0.095	1.000	0.043	0.044	0.049	0.129	1.944	7
15	0.010	0.039	0.005	0.016	0.005	0.024	0.022	0.084	0.013	0.130	0.084	0.082	0.101	0.043	1.000	0.009	0.041	0.129	1.837	9
16	0.038	0.007	0.005	0.036	0.035	0.009	0.007	0.044	0.003	0.080	0.016	0.063	0.028	0.044	0.009	1.000	0.095	0.065	1.584	15
17	0.004	0.000	0.021	0.024	0.022	0.036	0.028	0.035	0.011	0.055	0.006	0.050	0.014	0.049	0.041	0.095	1.000	0.129	1.620	13
18	0.051	0.011	0.037	0.074	0.045	0.049	0.066	0.094	0.033	0.183	0.025	0.077	0.085	0.129	0.129	0.065	0.129	1.000	2.282	1

Table 11
Sorting all attributes in Dataset "*Lymphography*" using *MGR* and *MNIG*.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Attributes sorted by *MGR*	4	9	7	5	3	10	13	18	17	15	8	12	2	6	11	14	16	1
HCA	0.578	0.570	0.599	0.570	0.570	0.690	0.789	0.739	0.570	0.711	0.641	0.599	0.739	0.570	0.578	0.648	0.585	0.578
HARI	0.006	0.001	0.022	−0.010	−0.012	0.139	0.329	0.223	−0.005	0.173	0.071	0.031	0.224	−0.004	0.017	0.080	0.020	0.014
Attributes sorted by *NMIG*	18	10	5	13	3	4	14	12	15	2	6	11	17	7	16	8	1	9
HCA	0.739	0.690	0.570	0.799	0.570	0.578	0.648	0.599	0.711	0.739	0.570	0.578	0.570	0.599	0.585	0.641	0.578	0.570
HARI	0.223	0.139	−0.010	0.329	−0.012	0.006	0.080	0.031	0.173	0.224	−0.004	0.017	−0.005	0.022	0.020	0.071	0.014	0.001

Furthermore, given an information table $S = (U, A)$ and an object subset $X \subseteq U$, $B \subseteq A$, it is possible to construct a rough set of X on the universe by elemental information granules using the following definition:

$$\underline{B}X = \cup\{[x]_B | [x]_B \subseteq X\}, \text{ and } \overline{B}X = \cup\{[x]_B | [x]_B \cap X \neq \emptyset\},$$

where $\underline{B}X$ and $\overline{B}X$ are called the B-lower and B-upper approximations with respect to B, respectively. The order pair $\langle \underline{B}X, \overline{B}X \rangle$ is called a rough set of X. Based the definitions of upper and lower approximations, the roughness of X with respect to B is defined as follows:

Definition 2.1 [45,48]**.** Given information systems $S = (U, A)$, and $B \subseteq A$, the roughness of X with respect to B is defined as

$$R_B(X) = 1 - \frac{|\underline{B}(X)|}{|\overline{B}(X)|}.$$

2.2. Total roughness

Mazlack et al. [39] used TR to determine important attributes for clustering:

Definition 2.2. Given an information system $S = (U, A)$ and $a_i \in A$, the TR of a_i is defined as

$$TR(a_i) = \frac{\sum_{j=1, j\neq i}^{|A|} CRough_{a_j}(a_i)}{|A| - 1},$$

where $U/\{a_i\} = \{X_1^{a_i}, X_2^{a_i}, \ldots, X_h^{a_i}\}$, and $CRough_{a_j}(a_i) = \frac{\sum_{k=1}^{h}(1 - R_{a_j}(X_k^{a_i}))}{h}$.

Mazlack et al. [39] pointed out that better clustering attribute selection occurs at high values of *TR*.

2.3. MMR algorithm

The MMR algorithm was proposed in [40] to incorporate the consideration of uncertainty into the process of clustering categorical data:

Definition 2.3. Given an information system $S = (U, A)$ and $a_i \in A$, $U/\{a_i\} = \{X_1^{a_i}, X_2^{a_i}, \ldots, X_h^{a_i}\}$, the roughness on attribute a_i with respect to a_j is defined as the mean of the roughness values of $X_1^{a_i}, X_2^{a_i}, \ldots, X_h^{a_i}$ with respect to a_j and is denoted as

$$Rough_{a_j}(a_i) = \frac{\sum_{k=1}^{h} R_{a_j}(X_k^{a_i})}{h}$$

where $a_i, a_j \in A$ and $a_i \neq a_j$.

The min-roughness (MR) of attribute a_i refers to the minimum of the mean roughness, or $MR(a_i) = Min_{a_j \in A, a_j \neq a_i}\{Rough_{a_j}(a_i)\}$. Based on the MR, MMR is then defined as follows:

Definition 2.4. Given an information system $S = (U, A)$ and $a_i \in A$, the MMR of a_i is defined as

$$MMR(a_i) = Min_{a_i \in A}\{MR(a_i)\}.$$

In [40], MMR was used to determine the degree of roughness of sub-cluster splitting by a given attribute. This provided a means of selecting the proper attribute for splitting a cluster. The MMR algorithm iteratively divides a set of objects with the goal of achieving better clustering crispness, which is described as follows:

Algorithm MMR

Input: $S=\{U,A\}$, k;
Output: Clustering result of U.
Step 1: Set current dataset $C=U$ and set current number of clusters $CNC=1$;
Step 2: For each attribute $a_i\in A$, calculate $MR(a_i)$;
Step 3: Determining clustering attribute a, $a=argmin(MR(a_i))$, for $a_i\in A$;
Step 4: Assume $C/\{a\}=\{X_1,X_2,\ldots,X_h\}$ and calculate $\sum_{a_i\in A,a_i\neq a}R_{a_i}(X_i)$;
Step 5: Selecting splitting equivalence class X_0,
$X_0=argmin(\sum_{a_i\in A,a_i\neq a}R_{a_i}(X_i))$, for $X_i\in C/a$ and $X_0'=C-X_0$;
Step 6: If $|X_0'|\geq|X_0|$,
$C=X_0'$, $C_{CNC}=X_0$
Else
$C=X_0$, $C_{CNC}=X_0'$
Endif
Step 7: $CNC=CNC+1$
If $CNC<k$ and $C\neq\emptyset$
Goto Step 2
Else
If $CNC=k$ and $C\neq\emptyset$
output $\{C_1,C_2,\ldots,C_{CNC}\}$ as the last cluster
Endif
Endif
End

2.4. MDA algorithm

Herawan et al. [41] proposed MDA, another technique based on the dependency of attributes in an information system, which is defined as follows:

Definition 2.5. Let $S=(U,A)$ be an information system, D, $C\subseteq A$. Dependency of attribute D on C in degree k $(0\leq k\leq 1)$ is denoted by $C\Rightarrow_k D$, where the degree k is defined as

$$k=\frac{\sum_{X\in U/D}|\underline{C}(X)|}{|U|}.$$

Based on MDA, in [42], Herawan et al. proposed the following algorithm:

Algorithm MDA

Input: Dataset U;
Output: Cluster attribute.
Step 1: Compute the equivalence classes using indiscernibility relation on each attribute;
Step 2: Determine the dependency degree of attribute a_i with respect to all a_j, where $i\neq j$;
Step 3: Select the maximum dependency degree of each attribute;
Step 4: Select the clustering attribute based on the maximum degree of dependency of the attributes;
End

It is evident that, although the MDA algorithm can be used to select attributes for spitting a cluster, it is not a complete divisive hierarchical clustering algorithm.

2.5. MGR algorithm

In [44], the MGR algorithm was proposed for selecting attributes for splitting clusters in divisive hierarchical clustering. Here, we review MGR, first providing some related definitions as follows:

Definition 2.6. Given an information system $S=(U,A)$ and a_i, $a_j\in A$, the information gain of a_i with respect to a_j denoted by $IG_{a_j}(a_i)$ is defined as

$$IG_{a_j}(a_i)=E(a_i)-CE_{a_i}(a_j).$$

where $E(a_i)=-\sum_{i=1}^{n}\frac{|X_i|}{|U|}log_2\frac{|X_i|}{|U|}$ is the information entropy, $CE_{a_i}(a_j)=-\sum_{j=1}^{m}\frac{|Y_j|}{|U|}\sum_{i=1}^{n}\frac{|X_i\cap Y_j|}{|X_i|}log_2\frac{|X_i\cap Y_j|}{|X_i|}$ is the conditional information entropy, and $U/\{a_i\}=\{X_1,X_2,\ldots,X_n\}$ and $U/\{a_j\}=\{Y_1,Y_2,\ldots,Y_m\}$, respectively.

Definition 2.7. Given an information system $S=(U,A)$ and a_i, $a_j\in A$, the information gain ratio(GR) of a_i with respect to a_j denoted by $GRa_j(a_i)$ is defined as

$$GR_{a_j}(a_i)=\frac{IG_{a_j}(a_i)}{E(a_i)}.$$

Definition 2.8. Given an information system $S=(U,A)$ and a_i, $a_j\in A$, the mean information gain ratio (MGR) of a_i is defined as

$$MGR(a_i)=\frac{\sum_{j=1,j\neq i}^{|A|}GRa_j(a_i)}{|A|-1}.$$

In the MGR algorithm, the definition of the GR is extended to the MGR, which is used to measure the similarity of all partitions defined by a given attribute as well as the similarities of the partitions defined by all other attributes. A partition defined by an attribute with a high MGR will be closer to partitions defined by the other attributes. A divisive hierarchical clustering method based on MGR was designed in [44] as follows:

Algorithm MGR

Input: $S=(U,A)$, k;
Output: Clustering result of U.
Step 1: Set current dataset $C=U$.
Set current number of clusters $CNC=1$.
Step 2: For each attribute $a_i\in A$,
Calculate $MGR(a_i)$ using Eq. (5)
EndFor
Step 3: Determining clustering attribute a,
$a=argmax_{a_i}(MGR(a_i))$, for $a_i\in A$.
Step 4: Assume a defined partition: $C/a=\{X_1,X_2,\ldots,X_h\}$,
For each equivalence class $X_i\in C/a$ for $i=1,2,\ldots,h$
Calculate $Entropy(X_i)$ using Eq. (6)
EndFor
Step 5: Selecting splitting equivalence class X,
$X=argmin(Entropy(X_i))$, for $X_i\in C/a$ where $i=1,2,\ldots,h$
Step 6: Output X as one cluster and set $C=C-X$.
Step 7: $CNC=CNC+1$
If $CNC<k$ and $C\neq\emptyset$
Goto Step 2
Else
If $CNC=k$ and $C\neq\emptyset$
output C as the last cluster
EndIf
EndIf
End

3. Divisive hierarchical clustering framework for categorical data

Although several high-performing divisive hierarchical clustering algorithms for categorical data have been proposed, these algorithms have not been systematically or comprehensively investigated, and no unified framework for this type of algorithm has

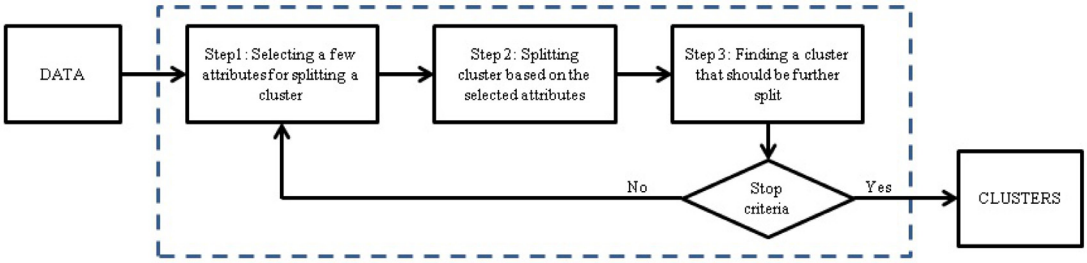

Fig. 1. Framework of divisive hierarchical clustering for categorical data.

Table 12
Top 20% attributes selected in the first iteration by different measures and corresponding *CA*s and *HARI*s.

		Vote	*Cancer*	*Mushroom*	*Chess*	*Shuttle*	*Lymphography*	*Promoters*	*Balloon*
TR	Top 20% attributes	1, 2, 3, **4**	9, **4**	17, 6, 18, **19**, 7	28,14,19,25 29,**32**, 26, 36	6, **1**	9, 4, **7**, 3	1, 2, 3, 4, 5, **6** 7,8,9,10,11,12	**1**
	HCA of Top 1 attribute	0.687	0.790	0.619	0.523	0.600	0.570	0.528	**0.800**
	HCA of Top 20% attributes	**0.956**	0.864	**0.854**	0.565	0.667	0.599	0.708	**0.800**
	HARI of Top 1 attribute	0.138	0.306	0.002	0.000	−0.045	0.001	−0.004	**0.326**
	HARI of Top 20% attributes	**0.832**	0.518	**0.493**	0.015	0.067	0.022	0.165	**0.326**
MMR	Top 20% attributes	1, 2, 3, **4**	9, **4**	17, 18, 6, **19**, 14	28,29,14,25 32, 19, **8**, 22	**3**, 6	9, 4, **7**, 3	1, 2, 3, 4, 5, **6** 7,8,9,10,11,12	**1**
	HCA of Top 1 attribute	0.687	0.790	0.619	0.523	0.667	0.570	0.528	**0.800**
	HCA of Top 20% attributes	**0.956**	0.864	**0.854**	**0.609**	0.667	0.599	0.708	**0.800**
	HARI of Top 1 attribute	0.138	0.306	0.002	0.000	0.067	0.001	−0.004	**0.326**
	HARI of 20% attributes	**0.832**	0.518	**0.493**	**0.046**	0.067	0.022	0.165	**0.326**
MDA	Top 20% attributes	1, 2, 3, **4**	9, **6**	17, 6, 18, **19**, 11	28,29,14,25 32, 19, **8**, 22	6, **1**	9, 4, **7**, 3	1, 2, 3, 4, 5, **6** 7,8,9,10,11,12	**1**
	HCA of Top 1 attribute	0.687	0.790	0.619	0.523	0.600	0.570	0.528	**0.800**
	HCA of Top 20% attributes	**0.956**	0.911	**0.854**	**0.609**	0.667	0.599	0.708	**0.800**
	HARI of Top 1 attribute	0.138	0.306	0.002	0.000	−0.045	0.001	−0.004	**0.326**
	HARI of Top 20% attributes	**0.832**	0.674	**0.493**	**0.046**	0.067	0.022	0.165	**0.326**
MGR	Top 20% of attributes	5, 8, **4**, 3	**2**, 3	6, 17, **19**, 12, 13	14,28,29,19 31, 11, **32**, 25	**3**, 1	4, 9, **7**, 5	**16**, 15, 42, 38, 17, 43 41,6,19,31,29,13	**1**
	HCA of Top 1 attribute	0.848	**0.927**	0.621	0.527	0.667	0.578	**0.802**	**0.800**
	HCA of Top 20% attributes	**0.956**	**0.927**	**0.854**	0.565	0.667	0.599	**0.802**	**0.800**
	HARI of Top 1 attribute	0.484	0.725	0.004	0.001	0.067	0.006	**0.358**	**0.326**
	HARI of Top 20% attributes	**0.832**	0.725	**0.493**	0.015	0.067	0.022	**0.358**	**0.326**
MNIG	Top of 20% attributes	5, 8, **4**, 3	**2**, 3	**19**, 13, 12, **5**, 14	11,31,26,35 36, **8**, 9, 15	**2**, 3	18, 10, 5, **13**	**16**, 42, 38, 15, 43, 41 6,31,17,51,29,19	**1**
	HCA of Top 1 attribute	0.848	**0.927**	**0.854**	0.531	**0.800**	0.739	**0.802**	**0.800**
	HCA of Top 20% attributes	**0.956**	**0.927**	**0.854**	**0.609**	**0.800**	**0.789**	**0.802**	**0.800**
	HARI of Top 1 attribute	0.484	**0.725**	**0.493**	0.003	**0.315**	0.223	**0.358**	**0.326**
	HARI of Top 20% attributes	**0.832**	**0.725**	**0.493**	**0.046**	**0.315**	**0.329**	**0.358**	**0.326**

been developed to date. To solve this issue, we use these existing algorithms as inspiration for proposing a uniform framework of hierarchical division clustering for overall analysis of existing algorithms. In addition, we propose some new, better-performing algorithms.

The uniform framework, which is illustrated in Fig. 1, comprises three steps: (1) selecting a few attributes for splitting a cluster; (2) using the selected attribute to split the cluster into smaller clusters; and (3) determining which of the resulting clusters should be further split. Note that for a given dataset, any given number of clusters can be obtained by recursively using a divisive bisecting clustering procedure. Thus the bisecting divisive approach, which is very popular in many applications (e.g. in document-retrieval/indexing problems), is chosen to split clusters in Step 2 [37,49]. These steps are described in detail as follows.

3.1. Step 1: Selecting attributes to use for splitting cluster

An object in a categorical dataset will generally have only a few values that can be captured by each attribute; this makes it easy to select an attribute for splitting a cluster. However, the attributes that can most effectively split a cluster must still be determined. To solve this problem, it is necessary to find suitable measures for assessing attributes. Algorithms for selecting attributes include TR [39], MMR [40], MDA [42], and MGR [44]. It must be noted that in these existing algorithms, only the Top 1 attribute based the order of attributes sorted by certain measure is picked out for splitting a cluster, but there could be some better attributes than the Top 1 in the sense of clustering performance, which will be detailed discussed in Section 5.1. Therefore, in our proposed framework, we select more than one attributes in Step 1.

3.2. Step 2: Splitting a cluster into two sub-clusters

If a cluster has an attribute with only two values, it is easy to split the cluster into two sub-clusters using this attribute. However, it is very common that a cluster will take more than two values for an attribute, leading to the production of more than two sub-clusters if the attribute is used for splitting the cluster. In this case, it is necessary to derive a method to merge some of the resulting

Table 13
Values of K-modes object function of all partitions and values of two clusters derived from these partitions on Dataset Vote.

	Bipartition 1	Bipartition 2	Bipartition 3	Bipartition 4	Bipartition 5	Bipartition 6	Bipartition 7	Bipartition 8	Bipartition 9	Bipartition 10
KOF	**1772**	1789	1808	1881	1897	1897	1995	2011	3136	3161
AC	0.848	0.828	0.832	0.816	**0.956**	0.945	0.867	0.874	0.630	0.614
ARI	0.484	0.428	0.440	0.397	**0.832**	0.791	0.536	0.556	0.023	−0.011

Table 14
*CA*s on datasets with two classes using 30 algorithms constructed based on the proposed framework.

Algorithms			Datasets								K-modes	MGR
Step 1	Step 2	Step 3	*Vote*	*Cancer*	*Mushroom*	*Chess*	*Shuttle*	*Lymphography*	*Promoters*	*Balloon*	W/T/L	W/T/L
TR	*KOF*	*MO*	**0.874**	0.811	**0.848**	0.522	**0.667**	0.570	**0.651**	**0.800**	**5/0/3**	**3/2/3**
	SOR	*MO*	0.614	0.691	**0.848**	**0.564**	0.600	0.577	0.509	**0.800**	3/0/5	2/2/4
	SIE	*MO*	**0.874**	*0.831*	**0.848**	0.522	0.600	0.577	0.509	**0.800**	4/0/4	2/2/4
	KOF	*IE*	**0.956**	*0.864*	**0.848**	0.522	**0.667**	0.570	**0.651**	**0.800**	**6/0/2**	**3/2/3**
	SOR	*IE*	**0.956**	0.691	**0.848**	**0.564**	0.600	0.577	0.509	**0.800**	4/0/4	**3/2/3**
	SIE	*IE*	**0.956**	*0.835*	**0.848**	0.522	0.600	0.577	0.509	**0.800**	4/0/4	2/2/4
MMR	*KOF*	*MO*	**0.874**	0.811	**0.848**	**0.564**	**0.66**7	0.570	**0.651**	**0.800**	**6/0/2**	**4/2/2**
	SOR	*MO*	0.614	0.691	**0.848**	**0.564**	0.600	0.577	0.509	**0.800**	3/0/5	2/2/4
	SIE	*MO*	**0.874**	*0.831*	**0.848**	**0.609**	0.600	0.577	0.509	**0.800**	5/0/3	3/2/3
	KOF	*IE*	**0.956**	*0.864*	**0.848**	**0.609**	**0.667**	0.570	**0.651**	**0.800**	**7/0/1**	**4/2/2**
	SOR	*IE*	**0.956**	0.691	**0.848**	**0.564**	0.600	0.577	0.509	**0.800**	4/0/4	3/2/3
	SIE	*IE*	**0.956**	*0.835*	**0.848**	**0.609**	0.600	0.577	0.509	**0.800**	5/0/3	3/2/3
MDA	*KOF*	*MO*	**0.874**	*0.881*	**0.848**	**0.564**	**0.667**	0.570	**0.651**	**0.800**	**7/0/1**	**4/2/2**
	SOR	*MO*	0.614	0.692	0.618	**0.564**	0.600	0.577	0.509	**0.800**	2/0/6	1/2/5
	SIE	*MO*	**0.874**	*0.877*	0.618	**0.609**	0.600	0.577	0.509	**0.800**	4/0/4	2/2/4
	KOF	*IE*	**0.956**	**0.897**	**0.848**	**0.609**	**0.667**	0.570	**0.651**	**0.800**	**7/0/1**	**5/2/1**
	SOR	*IE*	**0.956**	0.697	0.618	**0.564**	0.600	0.577	0.509	**0.800**	3/0/5	2/2/4
	SIE	*IE*	**0.956**	*0.877*	0.618	**0.609**	0.600	0.577	0.509	**0.800**	4/0/4	2/2/4
MGR	*KOF*	*MO*	**0.874**	**0.927**	**0.848**	*0.534*	**0.667**	0.570	0.528	**0.800**	**5/0/3**	**4/2/2**
	SOR	*MO*	0.630	0.685	0.618	**0.564**	**0.667**	0.577	0.538	**0.800**	3/0/5	1/3/4
	SIE	*MO*	**0.874**	*0.844*	0.618	*0.534*	**0.667**	0.577	0.538	**0.800**	4/0/4	2/3/3
	KOF	*IE*	0.816	**0.927**	**0.848**	*0.534*	**0.667**	0.570	0.528	**0.800**	**4/0/4**	**3/2/3**
	SOR	*IE*	0.630	0.685	**0.848**	**0.564**	**0.667**	0.577	0.538	**0.800**	**4/0/4**	2/3/3
	SIE	*IE*	0.816	*0.844*	**0.848**	*0.534*	**0.667**	0.577	0.538	**0.800**	**4/0/4**	2/3/3
MNIG	*KOF*	*MO*	**0.874**	**0.927**	**0.848**	**0.556**	**0.733**	0.570	0.528	**0.800**	**6/0/2**	**5/1/2**
	SOR	*MO*	0.630	0.685	**0.848**	0.522	**0.667**	*0.599*	0.538	**0.800**	3/0/5	2/2/4
	SIE	*MO*	**0.874**	*0.844*	*0.652*	0.522	**0.667**	**0.627**	0.538	**0.800**	5/0/3	3/2/3
	KOF	*IE*	0.816	**0.927**	**0.848**	0.522	**0.667**	**0.690**	0.528	**0.800**	**5/0/3**	**3/2/3**
	SOR	*IE*	0.630	0.685	**0.848**	0.522	**0.667**	**0.627**	0.538	**0.800**	4/0/4	2/2/4
	SIE	*IE*	0.816	*0.844*	*0.652*	0.522	**0.667**	**0.627**	0.538	**0.800**	4/0/4	2/2/4
K-modes	N/A	N/A	0.863	0.820	0.800	0.553	0.625	0.624	0.600	0.669	0/8/0	5/0/3
MGR in [44]	N/A	N/A	0.828	0.883	0.621	0.527	0.667	0.577	0.566	0.800	3/0/5	0/8/0

sub-clusters to acquire bipartition. Because the number of attribute values per cluster is generally not too large (usually less than 10), in this method all possible bipartitions in which all sub-clusters are divided into two groups are evaluated to pick out which are the most suitable for use in the next step. It is possible to implement Step 2 by leveraging or modifying existing methods such as overall minimum roughness [40], minimum information entropy [44], etc.

3.3. Step 3: Determining which cluster should be split

Divisive hierarchical clustering starts with placing all objects into one cluster and successively dividing it into smaller sub-clusters. Except for the first iteration, it is necessary to select or derive a method to determine which cluster should be split in each iteration. In addition to new methods, existing methods such as choosing the cluster with the largest number of objects [40] or the maximum entropy [44] can also be used.

4. Analyses of existing algorithms based on the proposed framework

Using the proposed framework, several representative hierarchical divisive clustering methods for categorical data were reviewed and comprehensively analyzed to produce an overall comparison of the algorithms with respect to the three generalized steps discussed in Section 3. The overall results are discussed in detail in the following sections and summarized in Table 3.

4.1. Analyses on Step 1

To determine which attribute in a dataset is optimal in terms of good clustering results, it is necessary to find a criterion for evaluating all attributes in the dataset. Clustering Accuracy(CA) and Adaptive Rand Index(ARI) are two widely-used indexes to evaluate clustering results [44], which are defined as follows:

Table 15
ARIs on datasets with two classes using 30 algorithms constructed based on the proposed framework.

Algorithms			Datasets								K-modes	MGR
Step 1	Step 2	Step 3	*Vote*	*Cancer*	*Mushroom*	*Chess*	*Shuttle*	*Lymphography*	*Promoters*	*Balloon*	W/T/L	W/T/L
TR	*KOF*	*MO*	**0.556**	0.360	**0.475**	0.000	*0.056*	−0.012	**0.083**	**0.326**	**5/0/3**	**3/1/4**
	SOR	*MO*	0.004	0.068	**0.475**	**0.015**	−0.045	*0.006*	−0.004	**0.326**	2/1/5	2/2/4
	SIE	*MO*	**0.556**	0.436	**0.475**	0.000	−0.045	*0.006*	−0.004	**0.326**	3/0/5	2/2/4
	KOF	*IE*	**0.832**	**0.518**	**0.475**	0.000	*0.056*	−0.012	**0.083**	**0.326**	**6/0/2**	3/1/4
	SOR	*IE*	**0.832**	0.068	**0.475**	**0.015**	−0.045	*0.006*	−0.004	**0.326**	3/1/4	**3/2/3**
	SIE	*IE*	**0.832**	0.448	**0.475**	0.000	−0.045	*0.006*	−0.004	**0.326**	3/0/5	2/2/4
MMR	*KOF*	*MO*	**0.556**	0.360	**0.475**	**0.015**	*0.047*	−0.012	**0.083**	**0.326**	**5/1/2**	**4/1/3**
	SOR	*MO*	0.004	0.068	**0.475**	**0.015**	−0.031	*0.006*	−0.004	**0.326**	2/1/5	2/2/4
	SIE	*MO*	**0.556**	0.436	**0.475**	**0.046**	−0.045	*0.006*	−0.004	**0.326**	4/0/4	3/2/3
	KOF	*IE*	**0.832**	**0.518**	**0.475**	**0.046**	*0.047*	−0.012	**0.083**	**0.326**	**7/0/1**	**4/1/3**
	SOR	*IE*	**0.832**	0.068	**0.475**	**0.015**	−0.031	*0.006*	−0.004	**0.326**	3/2/3	3/1/4
	SIE	*IE*	**0.832**	0.448	**0.475**	**0.046**	−0.045	*0.006*	−0.004	**0.326**	4/0/4	3/2/3
MDA	*KOF*	*MO*	**0.556**	**0.570**	**0.475**	**0.015**	*0.056*	−0.012	**0.083**	**0.326**	**6/1/1**	**4/1/3**
	SOR	*MO*	0.004	0.071	−0.021	**0.015**	−0.045	*0.006*	−0.004	**0.326**	1/1/6	1/2/5
	SIE	*MO*	**0.556**	**0.567**	−0.021	**0.046**	−0.045	*0.006*	−0.004	**0.326**	4/0/4	2/2/4
	KOF	*IE*	**0.832**	**0.623**	**0.475**	**0.046**	*0.056*	−0.012	**0.083**	**0.326**	**7/0/1**	**5/1/2**
	SOR	*IE*	**0.832**	0.080	−0.021	**0.015**	−0.045	*0.006*	−0.004	**0.326**	2/1/5	2/2/4
	SIE	*IE*	**0.832**	**0.567**	−0.021	**0.046**	−0.045	*0.006*	−0.004	**0.326**	4/0/4	2/2/4
MGR	*KOF*	*MO*	**0.556**	**0.725**	**0.475**	*0.004*	*0.047*	−0.010	−0.004	**0.326**	**5/0/3**	**4/1/3**
	SOR	*MO*	0.023	0.059	−0.015	**0.015**	*0.056*	*0.006*	0.002	**0.326**	2/1/5	1/2/5
	SIE	*MO*	**0.556**	0.473	−0.016	*0.004*	**0.067**	*0.006*	0.002	**0.326**	3/0/5	2/3/3
	KOF	*IE*	0.397	**0.725**	**0.475**	*0.004*	*0.047*	−0.010	−0.004	**0.326**	**4/0/4**	**3/1/4**
	SOR	*IE*	0.023	0.059	**0.475**	**0.015**	*0.056*	*0.006*	0.002	**0.326**	3/1/4	2/2/4
	SIE	*IE*	0.397	0.473	**0.475**	*0.004*	**0.067**	*0.006*	0.002	**0.326**	3/0/5	2/3/3
MNIG	*KOF*	*MO*	**0.556**	**0.725**	**0.475**	*0.011*	**0.166**	−0.010	−0.004	**0.326**	**5/0/3**	**5/1/2**
	SOR	*MO*	0.023	0.059	**0.475**	0.000	*0.056*	*0.022*	0.002	**0.326**	3/0/5	2/1/5
	SIE	*MO*	**0.556**	0.473	*0.046*	0.000	**0.067**	*0.048*	0.002	**0.326**	3/0/5	3/2/3
	KOF	*IE*	0.397	**0.725**	**0.475**	0.000	*0.047*	**0.138**	−0.004	**0.326**	**5/0/3**	**3/1/4**
	SOR	*IE*	0.023	0.059	**0.475**	0.000	*0.056*	*0.048*	0.002	**0.326**	3/0/5	2/1/5
	SIE	*IE*	0.397	0.473	*0.046*	0.000	**0.067**	*0.048*	0.002	**0.326**	2/0/6	2/2/4
K-modes	N/A	N/A	0.521	0.515	0.247	0.015	0.005	0.053	0.055	0.095	0/8/0	5/0/3
MGR in [44]	N/A	N/A	0.428	0.585	0.004	0.001	0.067	0.006	0.014	0.326	3/0/5	0/8/0

Given the true class labels and the required number of clusters, k, clustering accuracy is defined as

$$CA = \frac{\sum_{i=1}^{k} a_i}{n},$$

where k are the true class labels and the required number of clusters, n is the number of objects in the dataset, and a_i is the number of objects with the class label that dominates cluster i. According to this measure, if *CA* of a clustering results reaches to 1, it indicates that all objects in one cluster possess the same class label.

Given a dataset of n objects, suppose $P = \{p_1, p_2, \ldots, p_s\}$ and $Q = \{q_1, q_2, \ldots, q_t\}$ represent the original classes and the clustering result. Let n_{ij} be the number of objects that are in both class p_i and cluster q_j. Let c_i and d_j be the number of objects in class p_i and cluster q_j respectively. The adjusted rand index is defined as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{c_i}{2}\sum_j \binom{d_j}{2}\right]\Big/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{c_i}{2} + \sum_j \binom{d_j}{2}\right] - \left[\sum_i \binom{c_i}{2}\sum_j \binom{d_j}{2}\right]\Big/\binom{n}{2}},$$

If the clustering result is in line with the true class distribution, then the value of *ARI* is high.

In general, applying an attribute can generate one or more bipartitions. If more than one bipartition is present, the bipartition that corresponds to the highest clustering performance should be the best possible clustering result that the attribute can result in. Therefore, it is important to find a criterion for seek the bipartition. If *CA* can *ARI* are employed to evaluate the bipartitions generated by each attribute, the best attribute could be picked up. Furthermore, it is easy to know that the bipartition with the Highest Clustering Accuracy (*HCA*) and Highest Adaptive Rand Index (*HARI*) correspond to the best possible *CA* and best possible *ARI* of two-class dataset clustering, respectively, and we can therefore use the *HCA* and *HARI* as the attribute selection criterion. The attributes of each dataset in Table 1 are sorted in descending order of *HCA* and *HARI* in Tables 4 and 5, respectively.

To assess existing measures for selecting attributes (*TR, MMR, MDA*, and *MGR*), Table 6 lists the *HCA* and *HARI* of the top attribute for each measure, in which the highest *HCA* and *HARI* in each column are marked bold. From Table 6, it is seen that *MGR* produces the best *HCA* among the measures because *MGR* has the maximum number of bold values of *HCA* and *HARI*. From the preceding discussion, it would be reasonable to assume that *MGR* would be the best choice among the measures for use in Step 1.

4.2. Analyses on Step 2

Using an attribute selected in Step 1, it is easy to acquire a partition of a cluster (or a dataset). However, in cases in which the partition comprises more than two sub-clusters, it is challenging to select an appropriate bipartition based on such an attribute. To address this issue, the *MMR* algorithm [40] can be used to select the cluster with the overall minimum roughness—i.e., the cluster X for which Overall Roughness (*OR*) ($\sum_{a_k \in A-\{a_j\}} R_{a_k}(X)$ is minimum—as one part of a bipartition and then designate the remaining clusters as the other part of the bipartition. Alternatively, as smaller cluster

Table 16
*CA*s of on datasets with multi-classes using 30 algorithms constructed based on the proposed framework.

Algorithms			Datasets							K-modes	MGR
Step 1	Step 2	Step 3	*Soybean*	*Zoo*	*Car*	*Nursery*	*Balance*	*Haysroth*	*Lenses*	W/T/L	W/T/L
TR	*KOF*	*MO*	**0.979**	*0.891*	**0.700**	0.402	*0.590*	*0.439*	*0.625*	**4/1/2**	**1/2/4**
	SOR	*MO*	*0.830*	0.851	**0.700**	*0.430*	*0.590*	0.394	*0.625*	1/1/5	1/3/3
	SIE	*MO*	*0.830*	0.832	**0.700**	*0.430*	*0.590*	0.394	*0.625*	1/1/5	1/3/3
	KOF	*IE*	**0.979**	0.851	**0.700**	*0.420*	**0.635**	0.386	*0.625*	2/1/4	**2/3/2**
	SOR	*IE*	*0.830*	0.851	**0.700**	**0.492**	**0.635**	0.386	*0.625*	2/1/4	1/4/2
	SIE	*IE*	*0.830*	*0.871*	**0.700**	**0.520**	**0.635**	0.386	*0.625*	**3/1/3**	1/4/2
MMR	*KOF*	*MO*	**0.979**	*0.891*	**0.700**	0.402	*0.590*	0.417	*0.625*	**3/1/3**	**1/2/4**
	SOR	*MO*	0.787	0.812	**0.700**	*0.430*	*0.590*	0.417	*0.625*	1/1/5	**1/2/4**
	SIE	*MO*	0.787	0.832	**0.700**	*0.430*	*0.590*	0.386	*0.625*	1/1/5	**1/2/4**
	KOF	*IE*	**0.979**	0.851	**0.700**	*0.420*	**0.635**	0.386	*0.625*	**2/1/4**	**2/3/2**
	SOR	*IE*	0.787	0.802	**0.700**	**0.492**	**0.635**	0.386	*0.625*	**2/1/4**	1/3/3
	SIE	*IE*	0.787	0.851	**0.700**	**0.520**	**0.635**	0.386	*0.625*	**2/1/4**	1/3/3
MDA	*KOF*	*MO*	**0.979**	*0.891*	**0.700**	0.402	*0.590*	*0.439*	*0.625*	**3/1/3**	1/2/4
	SOR	*MO*	*0.830*	0.851	**0.700**	*0.430*	*0.590*	0.394	*0.625*	1/1/5	**1/3/3**
	SIE	*MO*	*0.830*	0.832	**0.700**	*0.430*	*0.590*	0.394	*0.625*	1/1/5	**1/3/3**
	KOF	*IE*	**0.979**	0.851	**0.700**	*0.420*	**0.635**	0.386	*0.625*	2/1/4	**2/3/2**
	SOR	*IE*	*0.830*	0.851	**0.700**	**0.492**	**0.635**	0.386	*0.625*	2/1/4	**2/3/2**
	SIE	*IE*	*0.830*	*0.861*	**0.700**	**0.520**	**0.635**	0.386	*0.625*	**3/1/3**	1/4/2
MGR	*KOF*	*MO*	**0.979**	*0.911*	**0.700**	0.402	*0.590*	**0.485**	*0.625*	**4/1/2**	1/3/3
	SOR	*MO*	0.766	*0.881*	**0.700**	*0.430*	*0.590*	*0.477*	*0.625*	3/1/3	1/2/4
	SIE	*MO*	0.766	**0.931**	**0.700**	*0.430*	*0.590*	0.386	*0.625*	2/1/4	**2/2/3**
	KOF	*IE*	**0.979**	**0.921**	**0.700**	*0.420*	**0.635**	**0.500**	*0.625*	**4/1/2**	**3/4/0**
	SOR	*IE*	0.766	*0.881*	**0.700**	**0.492**	**0.635**	**0.485**	*0.625*	**4/1/2**	1/4/2
	SIE	*IE*	0.766	*0.891*	**0.700**	**0.520**	**0.635**	**0.485**	*0.625*	**4/1/2**	1/4/2
MNIG	*KOF*	*MO*	**0.979**	**0.931**	**0.700**	*0.436*	*0.590*	**0.485**	*0.625*	**4/1/2**	**3/3/1**
	SOR	*MO*	*0.894*	0.673	**0.700**	*0.430*	*0.590*	*0.477*	*0.625*	3/1/3	2/2/3
	SIE	*MO*	0.660	*0.891*	**0.700**	*0.430*	*0.590*	0.386	*0.625*	2/1/4	1/2/4
	KOF	*IE*	**0.979**	0.832	**0.700**	**0.486**	**0.635**	**0.500**	*0.625*	**4/1/2**	**3/3/1**
	SOR	*IE*	**0.979**	0.851	**0.700**	**0.492**	**0.635**	0.386	*0.625*	3/1/3	2/3/2
	SIE	*IE*	0.766	*0.871*	**0.700**	**0.520**	**0.635**	0.386	*0.625*	3/1/3	1/3/3
K-modes	N/A	N/A	0.877	0.854	0.700	0.461	0.542	0.420	0.663	0/7/0	3/1/3
MGR in [44]	N/A	N/A	0.830	0.921	0.700	0.419	0.635	0.485	0.625	3/1/3	0/7/0

information entropies (*IE*s) correspond to greater object similarity within a cluster, the sub-cluster with the minimum *IE* can be selected as one part of a bipartition and output as the ultima clustering result cluster using the *MGR* algorithm [44]. To compare the performance of these two methods, we designed an experiment in which attributes for a cluster (or dataset) were selected using the *MMR* and *MGR* algorithms, with *OR* and *IE* used to perform partitioning (as the *TR* and *MDA* methods do not specify how to split a cluster into two disjoint parts, we did not analyze them here). As the produced datasets were all two-class, it was impossible to run Step 3 in these assessments. The results of applying Steps 1 and 2 using the two algorithms on the eight datasets in Table 1 are shown in Table 7. In the table, the highest value in each column is marked in bold, with the values in a column all marked in bold if they are all equal. It is seen from the results that the *MGR* algorithm outperformed the *MMR* algorithm. The algorithms using *OR* and *IE* in Step 2 have the similar performances of clustering.

4.3. Analyses on Step 3

Following Step 2, a bipartition comprising two disjoint subclusters of a cluster is obtained. The important problem of determining which sub-cluster is more suitable for splitting in next iteration must be solved in Step 3. The *MMR* algorithm presented in [40] chooses the sub-cluster with the Maximum number of Objects (*MO*) for further splitting in the next iteration. From the discussion in the preceding Section, we know that the *MGR* algorithm [44] selects the sub-cluster with minimal (*IE*) in each iteration as the eventual clustering result. The minimal *IE* can thus also be employed to determine the sub-cluster most suitable for the next iteration. To illustrate the performance in Step 3 of clustering algorithms using *MO* and *IE* (as neither the *TR* nor the *MDA* algorithm have explicit methods for selecting which cluster should be split in the next iteration, we do not consider them here), experiments were carried out on seven of the datasets in Table 2, with the results shown in Table 8, in which the largest value in each column is marked in bold (with values in columns with all equal values all marked in bold). It is seen from the table that the applications of the two methods in Step 3 result in the same *CA* and *ARI* on nearly all datasets in Table 3 except "*Zoo*"; this indicates that there are no significant differences between the performances of algorithms using *MO* and *IE* in Step 3.

5. New algorithms proposed based on the unified framework

Based on the hierarchical divisive clustering framework proposed in Section 3 and the analyses of several existing algorithms in Section 4, this section presents some feasible improvements for each step of the proposed framework. Briefly, mean normalized information gain (*MNIG*) is proposed to overcome the shortcomings of *MGR* in Step 1, and in Step 2, top 10 bipartitions from all possible ones are picked out according to certain measures instead of outputting one bipartition, the Maximum number of Objects (*MO*) is used to select one from some qualified bipartitions as the input to the next iteration in Step 3. All these improvements are employed as the basis for designing new algorithms. the number of objects in each cluster produced by the *MMR* algorithm can be used for evaluation

Table 17
*ARI*s of on datasets with multi-classes using 30 algorithms constructed based on the proposed framework.

Algorithms			Datasets							K-modes	MGR
Step 1	Step 2	Step 3	*Soybean*	*Zoo*	*Car*	*Nursery*	*Balance*	*Haysroth*	*Lenses*	W/T/L	W/T/L
TR	*KOF*	*MO*	**0.937**	*0.761*	−0.093	*0.028*	*0.050*	*0.002*	*0.046*	**4/0/3**	**3/0/4**
	SOR	*MO*	*0.674*	*0.691*	−0.093	*0.028*	*0.050*	−0.009	*0.046*	2/0/5	2/1/4
	SIE	*MO*	*0.674*	0.607	−0.093	*0.028*	*0.050*	−0.009	*0.046*	1/0/6	2/1/4
	KOF	*IE*	**0.937**	*0.688*	*0.013*	*0.041*	**0.101**	−0.015	*−0.043*	**3/0/4**	**3/2/2**
	SOR	*IE*	*0.674*	*0.690*	*0.013*	**0.073**	**0.101**	−0.015	*−0.043*	**3/0/4**	2/3/2
	SIE	*IE*	*0.674*	*0.831*	−0.012	**0.093**	**0.101**	−0.015	*−0.043*	**3/0/4**	1/3/3
MMR	*KOF*	*MO*	**0.937**	*0.764*	−0.093	*0.028*	*0.050*	−0.005	*0.046*	**4/0/3**	**3/0/4**
	SOR	*MO*	0.632	*0.687*	−0.093	*0.028*	*0.050*	*0.005*	*0.046*	3/0/4	2/0/5
	SIE	*MO*	0.632	0.607	−0.093	*0.028*	*0.050*	−0.010	*0.046*	1/0/6	2/0/5
	KOF	*IE*	**0.937**	*0.691*	*0.013*	*0.041*	**0.101**	−0.015	*−0.043*	**3/0/4**	**2/3/2**
	SOR	*IE*	0.632	*0.677*	*0.013*	**0.073**	**0.101**	−0.015	*−0.043*	**3/0/4**	1/3/3
	SIE	*IE*	0.632	*0.699*	−0.012	**0.093**	**0.101**	−0.015	*−0.043*	**3/0/4**	1/2/4
MDA	*KOF*	*MO*	**0.937**	*0.761*	−0.093	*0.028*	*0.050*	*0.002*	*0.046*	**4/0/3**	**3/0/4**
	SOR	*MO*	*0.674*	*0.691*	−0.093	*0.028*	*0.050*	−0.009	*0.046*	2/0/5	2/1/4
	SIE	*MO*	*0.674*	0.607	−0.093	*0.028*	*0.050*	−0.009	*0.046*	1/0/6	2/1/4
	KOF	*IE*	**0.937**	*0.688*	*0.013*	*0.041*	**0.101**	−0.015	*−0.043*	**3/0/4**	**2/3/2**
	SOR	*IE*	*0.674*	*0.690*	*0.013*	**0.073**	**0.101**	−0.015	*−0.043*	**3/0/4**	1/4/2
	SIE	*IE*	*0.674*	*0.822*	−0.012	**0.093**	**0.101**	−0.015	*−0.043*	**3/0/4**	1/3/3
MGR	*KOF*	*MO*	**0.937**	*0.947*	−0.093	*0.028*	*0.050*	**0.043**	*0.046*	**4/0/3**	**4/1/2**
	SOR	*MO*	0.554	*0.901*	−0.093	*0.028*	*0.050*	**0.100**	*0.046*	3/0/4	3/1/3
	SIE	*MO*	0.554	*0.916*	−0.093	*0.028*	*0.050*	−0.008	*0.046*	2/0/5	2/1/4
	KOF	*IE*	**0.937**	**0.958**	*0.013*	*0.041*	**0.101**	**0.066**	*−0.043*	**4/0/3**	**4/1/2**
	SOR	*IE*	0.554	*0.930*	*0.013*	**0.073**	**0.101**	**0.039**	*−0.043*	**4/0/3**	**4/1/2**
	SIE	*IE*	0.554	*0.916*	−0.012	**0.093**	**0.101**	**0.039**	*−0.043*	**4/0/3**	**4/1/2**
MNIG	*KOF*	*MO*	**0.937**	*0.945*	−0.093	*0.048*	*0.050*	**0.043**	*0.046*	**4/0/3**	**4/0/3**
	SOR	*MO*	**0.748**	0.406	−0.093	*0.027*	*0.050*	**0.100**	*0.046*	3/0/4	**4/0/3**
	SIE	*MO*	0.304	*0.845*	−0.093	*0.027*	*0.050*	−0.008	*0.046*	2/0/5	2/0/5
	KOF	*IE*	**0.937**	*0.849*	*0.013*	**0.061**	**0.101**	**0.066**	*−0.043*	**4/0/3**	**3/3/1**
	SOR	*IE*	**0.937**	0.604	*0.013*	**0.073**	**0.101**	−0.012	*−0.043*	3/0/4	2/3/2
	SIE	*IE*	0.539	*0.837*	−0.012	**0.093**	**0.101**	−0.012	*−0.043*	3/0/4	1/2/4
K-modes	N/A	N/A	0.737	0.656	0.019	0.053	0.027	−0.006	0.054	0/7/0	3/1/3
MGR in [44]	N/A	N/A	0.674	0.956	0.013	0.023	0.101	0.039	−0.043	3/1/3	0/7/0

Table 18
The progress of clustering Dataset "*Balloon*".

Iteration	Step 1		Step 2	Step 3		AC	ARI
	Measures	Attribute	Measure	Measure	Bipartition		
1	TR	1	KOF/SOR/SIE	MO/IE	(1–12),(13–20)	0.800	0.326
1	MMR	1	KOF/SOR/SIE	MO/IE	(1–12),(13–20)	0.800	0.326
1	MDA	1	KOF/SOR/SIE	MO/IE	(1–12),(13–20)	0.800	0.326
1	MGR	1	KOF/SOR/SIE	MO/IE	(1–12),(13–20)	0.800	0.326
1	MNIG	1	KOF/SOR/SIE	MO/IE	(1–12),(13–20)	0.800	0.326

Table 19
The progress of clustering Dataset "*Car*".

Iteration	Step 1		Step 2	Step 3			AC	ARI
	Measures	Attribute	Measure	Measure	Bipartition	Cluster		
1	TR/MMR/MDA/MGR/MNIR	1	KOF	MO	(1–432),(433–1728)	1–432		
2	TR/MMR/MDA/MGR/MNIR	2	KOF	MO	(1–108),(109–432)	1–108		
3	TR/MMR/MDA/MGR/MNIR	3	KOF		**(1–9,28–36,55–63,82–90), (10–27,37–54,64–81,91–108)**		0.700	−0.093
1	TR/MMR/MDA/MGR/MNIR	1	SOR/SIE	MO	(1–432),(433–1728)	41–432		
2	TR/MMR/MDA/MGR/MNIR	2	SOR/SIE	MO	(1–108),(109–432)	1–108		
3	TR/MMR/MDA/MGR/MNIR	3	SOR/SIE		**(1–27),(28–108)**		0.700	−0.093
1	TR/MMR/MDA/MGR/MNIR	1	KOF/SOR	IE	(1–432),(433–1728)	433–1728		
2	TR/MMR/MDA/MGR/MNIR	1	KOF/SOR	IE	(433–864),(865–1728)	865–1728		
3	TR/MMR/MDA/MGR/MNIR	1	KOF/SOR		**(865–1296),(1297–1728)**		0.700	0.013
1	TR/MMR/MDA/MGR/MNIR	1	SIE	IE	(1–432),(433–1728)	433–1728		
2	TR/MMR/MDA/MGR/MNIR	1	SIE	IE	(433–540,865–972,1297–1404), (541–864,973–1296,1405–1728)	(541–864,973–1296,1405–1728)		
3	TR/MMR/MDA/MGR/MNIR	1	SIE		**(541–864),(973–1296,1405–1728)**		0.700	−0.012

The input of some qualified bipartitions into Step 3 requires solving the problem of selecting a proper cluster. To facilitate this, the number of objects in each cluster produced by the *MMR* algorithm can be used for evaluation, with the cluster with the

5.1. Improvement for Step 1

Quinlan [50] pointed out that an attribute with a high number of values can lead to a larger *IG* if it shares the same information with a specific attribute that is shared by other attributes. To reduce the bias towards multi-valued attributes, the information *GR* was introduced in [50] to enhance the performance of decision tree algorithms. Based on the results, Qin et al. [44] leveraged the information *GR* to measure the similarities between partitions derived from various attributes and designed the *MGR* as a measure of the similarity between a given attribute and all other attributes.

However, it is easy to see that information *GR* of one attribute with respect to another can be very large if the first attribute's entropy and *IG* with respect to the other attribute are both low; in other words, the similarity between two attributes can be low even if the *GR* is large. In such cases, it would be inappropriate to use *MGR* to select representative attributes for splitting a cluster. To address this problem, we introduce a new measure—mean normalized information gain (*MNIG*):

Definition 5.1. Given an information system $S = (U, A)$ and a_i, $a_j \in A$, the *MNIG* of a_i—denoted by $MNIG(a_i)$—is defined as

$$MNIG(a_i) = \frac{\sum_{j=1, j\neq i}^{|A|} NIGa_j(a_i)}{|A| - 1},$$

where $NIGa_j(a_i) = \frac{IGa_j(a_i)}{E(a_i)+E(a_j)}$.

To better illustrate this problem, we use Tables 9–12. Tables 9–10 list the values of *MGR* and *MNIG* of all attributes in Dataset "*Lymphography*", respectively. In Table 9, the entries in columns 1–18 give the values of the information *GR*, with Column *MGR* listing the *MGR*s of the attributes and Column *Rank* showing the attributes sorted by *MGR*. In Table 10, the entries in columns 1–18 give the *NIG* values of the attributes, with entries in Column *MNIG* giving the *MNIG*s of the attributes and Column *Rank* showing the attributes sorted by *MNIG*. From Tables 9 and 10, we see that there is a significant difference between the order of attributes sorted by *MGR* and those sorted by *MNIG*. To indicate which of these two orders of attributes is more useful in clustering, Table 11 lists the *HCA* and *HARI* of each attribute in Database "*Lymphography*". From Table 11 it is easy to see that, except for the third attribute, the *HCA*s and *HARI*s of the first five attributes under *MNIG* are all at least as large as those obtained using *MGR*. Table 12 lists the top 20% attributes sorted by Measures *TR, MMR, MDA, MGR*, and *MNIG*, the *HCA*s and *HARI*s derived from the first attributes of each dataset, and the *HCA*s and *HARI*s derived from the top 20% attributes, with the highest *HCA* and *HARI* values in each column bolded. It is seen from Table 12 that the *HCA*s and *HARI* obtained from the top 20% attributes are preferable to those of the first attributes and that using *MNIG* results in the maximum *HCA* and *HARI*, which indicates using *MNIG* could produce the best clustering results.

5.2. Improvements for Step 2

In Step 1, we obtain the most appropriate attribute for splitting a cluster; using this attribute, dataset partitions can be obtained. If the attribute a_i of a cluster selected in Step 1 is binary, then the cluster will obviously be divided into two sub-clusters by the attribute. Similarly, selecting an attribute a_j whose domain includes more than two values will result in cluster division into more than two sub-clusters. It is therefore necessary to determine how to construct bipartitions using these equivalence classes.

In the *MMR* algorithm, the sub-cluster with minimum *IE* is regarded as one part of the bipartition and the other sub-clusters are merged into the other part. In the *MGR* algorithm, the sub-cluster with minimum *OR* is regarded as one part of the bipartition and the others are combined into the other part. However, the *OR* and *IE* measures assess each sub-cluster individually, and therefore not all possible bipartitions derived for a categorical attribute with more than three values can be covered. Taking an attribute with four values as an example, bipartitions comprising two sub-clusters and two sub-clusters will be overlooked by *MMR* and *MGR*, which can lead to situations in which the attribute *HCA* cannot be determined. To solve this problem, all of the bipartitions derived from an attribute can be considered candidate partitions to which an appropriate criterion can be applied to choose one bipartition for the next step.

In addition, the *MMR* and *MGR* algorithms select a bipartition by evaluating each of its two parts, a method that cannot be used to assess the overall bipartition. In this section, we explore the assessment of bipartitions as a whole by applying either the sum of overall roughness (*SOR*) or the sum of information entropy (*SIE*) to each bipartition half or the K-modes object function (*KOF*) to the whole bipartition. We determine that many partitions have very similar values of *SOR, SIE*, and *KOF*, making it difficult to identify which bipartition is best. This phenomenon can be observed in Table 13, in which the values of the *KOF* of the bipartitions of Datasets "*vote*" and the *CA*s and *ARI*s derived for each bipartition are listed. It is seen that bipartitions 1–8 possess very similar *KOF* values, while bipartition 5 has the maximum clustering accuracy but the fifth-smallest *KOF* value. In this case, using the top bipartition sorted by some measures (*SOR, SIE*, or *KOF*) might not be a good choice. To address this problem, we sort all the bipartitions by their values for a specific measure and choose several bipartitions according to a specified threshold of the measure as inputs to Step 3. Here, we use a method in which, assuming the bipartitions are sorted by *SOR, SIE*, or *KOF* as $BP_1, BP_2, \ldots, BP_s$, we extract the top 10 ten bipartitions $BP_1, BP_2, \ldots, BP_{10}$ and choose those prior to bipartition BP_i that satisfy $\frac{BP_i - BP_{i-1}}{BP_{i+1} - BP_i} > 2$. To obtain the top BP_1, we apply a method in which we only choose BP_1 if $\frac{BP_2 - BP_1}{BP_3 - BP_2} > 2$.

It should be noted that in Step 2, many other methods can be used to select one bipartition or build a better one for Step 3. For example, the criterions in [51–53] can also be used to assess all possible bipartitions, and all the bipartitions can be regarded as an ensemble and incorporated to built a better bipartition. Because of the limitation of space, the alternative methods will be tested in future investigation.

5.3. Improvement for Step 3

The input of some qualified bipartitions into Step 3 requires solving the problem of selecting a proper cluster. To facilitate this, the number of objects in each cluster produced by the *MMR* algorithm can be used for evaluation, with the cluster with the Maximum number of Objects (*MO*) selected as the input to the next iteration. For two-classes datasets, the step is used to select a bipartition including the cluster with the minimum *MO* and the selected bipartition will be final clustering result. Alternatively, clusters can be evaluated by using Measure *IE* in *MGR* algorithm to assess a cluster.

6. Experimental analysis

We tested a number of algorithms constructed based on our proposed divisive hierarchical clustering framework on the

performance of clustering categorical data and report the results in this Section. Fifteen benchmark datasets were divided into two groups: two-class datasets (shown in Table 1), and datasets with more than two classes (shown in Table 2). Using these dataset groupings, we compared the performance of various algorithms under our proposed framework with the performance of two baseline algorithms: *MGR* ([44]) and *K-modes*.

We first performed several experiments on the datasets to demonstrate the performance of algorithms constructed based on our proposed divisive hierarchical clustering framework. The experimental results for the datasets in Table 1 are listed in Tables 14 and 15, respectively. The *AC*s of clustering the eight datasets in Table 1 by applying thirty algorithms generated using five measures (*TR, MMR, MDA, MGR*, and *MNIR*) in Step 1, three measures (*KOF, SOR*, and *SIE*) in Step 2, and two measures (*MO* and *IE*) in Step 3 are shown in Table 14, in which the last two columns give the *Win-Tie-Loss*(*W*|*T*|*L*) of all the algorithms based on our proposed framework compared with two baselines in each row. The *AC*s—none of which are less than those obtained by the two baselines—are given in boldface, and the *AC*s which are more than one of those obtained by the two baselines are marked in italic. In last two columns, the entries with the highest values of the number of *Win* minus the one of *Loss*" in each group of algorithms—algorithms are grouped by the measures (*TR, MMR, MDA, MGR*, and *MNIR*) used in Step 1 and then by the measures (*MO* and *IE*) used in Step 3—are marked in bold.

From Table 14, it is easy to see that algorithms of using *KOR* in Step 2 get the highest values of *AC* in Columns *K-Modes W*|*T*|*L* and *MGR W*|*T*|*L* in their respective groups regardless of the measures in Step 1 and the measures in Step 3. The results suggest that *KOF* is the best measure for use in Step 2 of our proposed framework. And it must be noted that the algorithms based on *MMR*(Step 1)+*KOR*(Step 2)+*IE*(Step 3), *MDA*(Step 1)+*KOF*(Step 2)+*MO*(Step 3) and *MDA*(Step 1)+*KOR*(Step 2)+*MO*(Step 3) perform a little better than those based on *MNIG*(Step 1)+*KOF*(Step 2)+*MO*(Step 3) and *MNIG*(Step 1)+*KOF*(Step 2)+*IE*(Step 3). In terms of the results mentioned above, we can find that although, as described in Section 5.1, measure *MNIR* can produce "*better*" attributes, the performance of algorithms based on *MNIR* is not absolutely better than those of other algorithms. This can be explained as follows: although there are many partitions generated from the attributes selected by *MNIR*, the highest accuracy of clustering will be achieved only if the best of these partitions are selected in Steps 2–3. In other words, for two-classes datasets, the high-quality bipartitions generated by the "*better*" attributes fail to be picked out by the measures in Steps 2 and 3. Similar with Table 14, *ARI*s obtained by running the thirty new algorithms to cluster the datasets from Table 1 are shown in Table 15. From the table, the results consistent with those in Table 14 are obtained, suggesting that *KOF* is the best measure in Step 2 of our proposed framework.

For the datasets with more than two classes in Table 2, the experimental results are listed in Tables 16 and 17, respectively. The *AC*s of clustering the seven datasets in Table 2 by applying thirty algorithms generated using five measures (*TR, MMR, MDA, MGR*, and *MNIR*) in Step 1, three measures (*KOF, SOR*, and *SIE*) in Step 2, and two measures (*MO* and *IE*) in Step 3 are shown in Table 16, whose last two columns give the *Win-Tie-Loss*(*W*|*T*|*L*) values of all the algorithms based on our proposed framework compared with two baselines in each row. The *AC*s—none of which are less than those obtained by the two baselines—are given in boldface, and the *AC*s which are more than one of those obtained by the two baselines are marked in italic. In last two columns, the entries with the highest values of "*Win-Loss*" in each group of algorithms—algorithms are grouped by the measures (*TR, MMR, MDA, MGR*, and *MNIR*) used in Step 1 and then by the measure used in Step 3—are marked in bold. Similarly, Table 17 shows the *ARI*s obtained by running the thirty new algorithms to cluster the seven datasets from Table 1.

From Table 16, we can find out that all the algorithms of using *KOR* in Step 2 get the highest values of *AC* in Columns *K-Modes W*|*T*|*L* and *MGR W*|*T*|*L* in their respective groups regardless of the measures in Step 1 and the measures in Step 3. The results indicate that *KOF* could also be the best measure for use in Step 2 of our proposed framework for multi-classes datasets. And it must be pointed out that the algorithms based on *MGR*(Step 1)+*KOR*(Step 2)+*IE*(Step 3), *MNIG*(Step 1)+*KOF*(Step 2)+*MO*(Step 3) and *MNIG*(Step 1)+*KOR*(Step 2)+*MO*(Step 3) perform better than both baselines. In terms of the results mentioned above, we can find that the performance of algorithms based on *MNIR* is not absolutely better than those of using *MGR*, it seem to conflict with results described in Section 5.1 (measure *MNIR* can produce "*better*" attributes). The reason for this contradiction is that although there are many partitions generated from the attributes selected by *MNIR*, the highest accuracy of clustering will be achieved only if the best of these partitions are selected in Steps 2 and 3. In other words, for multi-classes datasets, the high-quality bipartitions generated by the "*better*" attributes fail to been picked out by the measures in Step 2 and Step 3. Similar with Table 16, *ARI*s got from running the thirty algorithms to cluster the datasets in Table 2 are listed in Table 17. From the table, the results consistent with those in Table 16 are obtained, suggesting that for datasets with more than one classes, *KOF* is also the best measure in Step 2 of our proposed framework.

In addition, there is an interesting phenomenon should be investigated further, which is that these values of Column "*Balloon*" in Table 14 and Columns "*Car*" and "*lenses*" in Table 16 are the same regardless of measures in Steps 1–3. To explain the reason for this phenomenon, Tables 18 and 19 were employed.

Table 18 shows the detailed process of clustering Dataset "*Balloon*", in which iteration number is listed in Column "*Iteration*", the measure used in Step 1 and the attribute selected in Step 1 are shown in Column "*Step 1 Measure*" and "*Step 1 Attribute*" respectively, and the measure used in Step 1 and the attribute selected in Step 1 are shown in Column "*Step 1 Measure*" and "*Step 1 Attribute*" respectively, and the measure used in Step 2—because there are more than one partitions are selected in Step 2 and the limitation of the table, the bipartitions selected in Step 2 are not listed in Table 18—are shown in Column "*Step 2 Measure*", and the measure used and the bipartition selected in Step 3 are put into Column "*Step 3 Measure*" and "*Step 3 Bipartition*", and then *AC* and *ARI* of the selected bipartition is shown in the last two columns, respectively. In an entry of Column "*Bipartition*", two clusters are in two sets of parentheses, respectively. Taking the first row in Column "*Bipartition*" as an example, (1–12) is one cluster, and (13–20) is the other cluster. Because Dataset "*Balloon*" consists of two-classes, an expected partition including two clusters can be obtained by running Steps 1–3 once, and thus all values in Column "*Iteration*" are "1". From Table 18, we can see that running algorithms of using all fives measures in Step 1, all three measures in Step 2 and two measures in Step 3 produce the same bipartition, and thus the same *AC* and *ARI* can be obtained for all these algorithms.

The process of clustering Dataset "*Car*" is detailed in Table 19, in which the number of iteration is shown in Column "*Iteration*", the measure used in Step 1 and the attribute selected in Step 1 are shown in Column "*Step 1 Measure*" and "*Step 1 Attribute*", respectively; the measure used in Step 1 and the attribute selected in Step 1 are shown in Column "*Step 1 Measure*" and "*Step 1 Attribute*", respectively, and the measure used in Step 2 is shown in Column "*Step 2 Measure*", and the measure used in Step 3, the bipartition selected in Step 3 and the cluster selected in Step 3 are shown in Column "*Step 3 Measure*", "*Step 3 Bipartition*", and "*Step 3 Cluster*", respectively; and *AC* and *ARI* of the selected par-

tition are shown in the last two columns, respectively. Because Dataset *Car* consists of four classes, an expected partition including four clusters can be got by running Steps 1–3 for three times, and thus the values in column *Iteration* is "1", "2" or "3". From Table 19, we can see that there are four different final partitions (marked in boldface) to be acquired, which results in the same *AC* ("0.007") and three different *ARIs* ("−0.093, 0.013 and −0.012"). Because the reason why there are the same value in Column "*Lenses*" of Table 16 regardless of measures in Steps 1–3 is the same as the one in Column "*Car*", the progress of clustering Dataset "Lenses" weren't shown in this paper.

In summary, for our proposed framework, measure *MNIR* in Step 1 can produce "*better*" attributes, but the performance of algorithms based on *MNIR* is not absolutely better than those of other algorithms. The reason is that the highest accuracy of clustering will be achieved only if the best of these partitions which are generated from the the attributes selected by *MNIR* are selected in Steps 2 and 3; Measure *KOF* is the best one for use in Step 2 of our proposed framework; And for measures *MO* and *IE* in Step 3, it is hard to say which is better than the other.

7. Conclusions

This paper presented a uniform framework of divisive hierarchical clustering comprising three main steps: (1) select some attributes for splitting a selected cluster; (2) generate bipartitions of the cluster using these selected attributes; and finally (3) determine which cluster should be split. Using the proposed framework, several common divisive hierarchical algorithms for categorical data were analyzed in a stepwise manner and their shortfalls were explored. To address these shortfalls, we introduced two new measures–*Mean Normalized Information Gain* (*MNIG*) and *K-mode Object Function* (*KOF*)–for use in Steps 1 and 2, respectively. Through experimental analysis, we determined that, although a better *HCA* and *HARI* can be achieved using measure *MNIG*, obtaining a good measure in Step 1 is merely a precondition for obtaining good clustering results, for which proper selection of the methods used in Steps 2 and 3 is also vital. We found that *KOF* is the best measure for use in Step 2 of our proposed framework, although for Step 3 none of the measures *MO* and *IE* examined in this study was obviously superior. Thus, obtaining a truly optimized measure remains a difficult problem that can yield valuable future research results.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Nos. 61772323, 61303008, 71301090, 61402272 and U1435212), the National Key Basic Research and Development Program of China (973) (No. 2013CB329404), and the Research Project Supported by Shanxi Scholarship Council of China (No. 2017-005).

References

[1] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Mateo, CA, USA, 2001.
[2] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[3] B. Fischer, J.M. Buhmann, Bagging for path-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1411–1415.
[4] R. Xu, I.I. D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.
[5] C.M. Zhong, D.Q. Miao, R.Z. Wang, X.M. Zhou, DIVFRP: an automatic divisive hierarchical clustering method based on the furthest reference points, Pattern Recognit. Lett. 29 (16) (2008) 2067–2077.
[6] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
[7] L. Bai, J.Y. Liang, C.Y. Dang, F.Y. Cao, A cluster centers initialization method for clustering categorical data, Expert Syst. Appl. 39 (2012) 8022–8029.
[8] X.W. Zhao, J.Y. Liang, C.Y. Dang, Clustering ensemble selection for categorical data based on internal validity indices, Pattern Recognit. 69 (2017) 150–168.
[9] L. Bai, J.Y. Liang, C.Y. Dang, F.Y. Cao, novel attribute weighting algorithm for clustering high-dimensional categorical data, Pattern Recognit. 44 (2011) 2843–2861.
[10] L. Bai, J.Y. Liang, Cluster validity functions for categorical data:a solution-space perspective, Data Min. Knowl. Discov. 29 (6) (2015) 1560–1597.
[11] Z.X. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Min. Knowl. Discov. 2 (3) (1998) 283–304.
[12] Z.X. Huang, M.K. Ng, A fuzzy k-modes algorithm for clustering categorical data, IEEE Trans. Fuzzy Syst. 7 (4) (1999) 446–452.
[13] J.Z. Huang, M.K. Ng, H.Q. Rong, Z.C. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 657–668.
[14] M.K. Ng, M.J. Li, J.Z. Huang, Z.Y. He, On the impact of dissimilarity measure in k-modes clustering algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 29 (3) (2007) 503–507.
[15] O.M. San, V.N. Huynh, Y. Nakamori, An alternative extension of the k-means algorithm for clustering categorical data, Pattern Recognit. 14 (2) (2004) 241–247.
[16] D. Barbara, J. Couto, Y. Li, COOLCAT: an entropy-based algorithm for categorical clustering, in: Proceedings of the 11th International Conference of Information Knowledge Management, USA, 2002, pp. 582–589.
[17] S. Guha, R. Rastogi, K. Shim, ROCK: a robust clustering algorithm for categorical attributes, in: Proceeding of 15th ICDE, 1999, pp. 512–521.
[18] D. Fisher, Knowledge acquisition via incremental conceptual clustering, Mach. Learn. 2 (1987) 139–172.
[19] G. Karypis, E.H. Han, V. Kumar, CHAMELEON: hierarchical clustering using dynamic modeling, Computer 32 (8) (1999) 68–75.
[20] P. Andritsos, V. Tzerpos, Information-theoretic software clustering, IEEE Trans. Softw. Eng. 31 (2) (2005) 150–165.
[21] S. Guha, R. Rastogi, K. Shim, CURE: an efficient clustering algorithm for large databases, in: Proceeding of ACM SIGMOD International Conference on Management of Data, 1998, pp. 73–88.
[22] Y.G. Lu, Y. Wan, PHA: a fast potential-based hierarchical agglomerative clustering method, Pattern Recognit. 46 (2013) 1227–1239.
[23] G. Bordogna, G. Pasi, A quality driven hierarchical data divisive soft clustering for information retrieval, Knowl. Based Syst. 26 (2012) 9–19.
[24] W. Yao, C.O. Dumitru, O. Loffeld, M. Datcu, Semi-supervised hierarchical clustering for semantic SAR image annotation, IEEE J. Sel. Top. Appl. Earth Observations Remote Sens. 9 (5) (2016) 1993–2008.
[25] J.D. West, I. Wesley-Smith, C.T. Bergstrom, A recommendation system based on hierarchical clustering of an article-level citation network, IEEE Trans. Big Data 2 (2) (2016) 113–123.
[26] I. Cattinelli, G. Valentini, E. Paulesu, N.A. Borghese, A novel approach to the problem of non-uniqueness of the solution in hierarchical clustering, IEEE Trans. Neural Netw. 24 (7) (2013) 1166–1173.
[27] Q. Mao, W. Zheng, L. Wang, Y.P. Cai, V. Mai, Y.J. Sun, Parallel hierarchical clustering in linearithmic time for large-scale sequence analysis, in: Proceeding of the IEEE International Conference on Data Mining (ICDM), Atlantic City, NJ, USA, 14–17.
[28] P.A. Vijaya, M.N. Murty, D.K. Subramanian, Efficient bottom-up hybrid hierarchical clustering techniques for protein sequence classification, Pattern Recognit. 39 (2006) 2344–2355.
[29] R. Sibson, SLINK: an optimally efficient algorithm for the single link cluster method, Comput. J. 16 (1) (1973) 30–34.
[30] D. Defays, An efficient algorithm for a complete link method, Comput. J. 20 (4) (1977) 364–366.
[31] E.M. Voorhees, Implementing agglomerative hierarchic clustering algorithms for use in document retrieval, Inf. Process. Manag. 22 (6) (1986) 465–476.
[32] I. Gurrutxaga, I. Albisua, O. Arbelaitz, J.I. Martń, J. Muguerza, J. M. Pérez, I. Perona, SEP/COP: an efficient method to find the best partition in hierarchical clustering based on a new cluster validity index, Pattern Recognit. 43 (2010) 3364–3373.
[33] J.M. Leski, M. Kotas, Hierarchical clustering with planar segments as prototypes, Pattern Recognit. Lett. 54 (2015) 1–10.
[34] D.L. Boley, Principal direction divisive partitioning, Data Min. Knowl. Discov. 2 (4) (1998) 325–344.
[35] M. Chavent, Y. Lechevallier, O. Briant, DIVCLUS-t: a monothetic divisive hierarchical clustering method, Comput. Stat. Data Anal. 52 (2) (2007) 687–701.
[36] S.M. Savaresi, D.L. Boley, S. Bittanti, G. Gazzaniga, Cluster selection in divisive clustering algorithms, in: Proceedings of the 2nd SIAM ICDM, Arlington, VA, 2002, pp. 299–314.
[37] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques, in: Proceedings of the KDD Workshop on Text Mining, 2000. Boston
[38] L. Feng, M.H. Qiu, Y.X. Wang, Q.L. Xiang, Y.F. Yang, K. Liu, A fast divisive clustering algorithm using an improved discrete particle swarm optimizer, Pattern Recognit. Lett. 31 (11) (2010) 1216–1225.
[39] L.J. Mazlack, A. He, Y. Zhu, S. Coppock, A rough set approach in choosing clustering attributes, in: Proceedings of the ISCA 13th International Conference (CAINE), 2000, pp. 1–6.
[40] D. Parmar, T. Wu, J. Blackhurst, MMR: an algorithm for clustering categorical data using rough set theory, Data Knowl. Eng. 63 (3) (2007) 879–893.
[41] T. Herawan, M.M. Deris, A framework on rough set-based partitioning attribute selection, Lect. Notes Comput. Sci. 5755 (2009) 91–100.

[42] T. Herawan, M.M. Deris, J.H. Abawajy, A rough set approach for selecting clustering attribute, Knowl. Based Syst. 23 (3) (2010) 220–231.
[43] T.K. Xiong, S.R. Wang, A. Mayers, E. Monga, A new MCA-based divisive hierarchical algorithm for clustering categorical data, in: Proceedings of the Ninth IEEE International Conference on Data Mining, 2009, pp. 1058–1063.
[44] H.W. Qin, X.Q. Ma, T. Herawan, J.M. Zain, MGR: an information theory based hierarchical divisive clustering algorithm for categorical data, Knowl. Based Syst. 67 (2014) 401–411.
[45] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Boston, 1991.
[46] Z. Pawlak, A. Skowron, Rudiments of rough sets, Inf. Sci. 177 (1) (2007) 3–27.
[47] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, Inf. Sci. 177 (1) (2007) 41–73.
[48] W. Wei, J.Y. Liang, Y.H. Qian, C.Y. Dang, Can fuzzy entropies be effective measures for evaluating the roughness of a rough set? Inf. Sci. 232 (2013) 143–166.
[49] S.M. Savaresi, D. Boley, On the performance of bisecting k-means and PDDP, in: Proceeding on SIAM Data Mining Conference, 2001.
[50] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann publishers, 1993.
[51] D. Huang, J.H. Lai, C.D. Wang, Ensemble clustering using factor graph, Pattern Recognit. 50 (2016) 131–142.
[52] D. Huang, J.H. Lai, C.D. Wang, Robust ensemble clustering using probability trajectories, IEEE Tans. Knowl. Data Eng. 28 (5) (2016) 1312–1326.
[53] D. Huang, C.D. Wang, J.H. Lai, Locally weighted ensemble clustering, IEEE Tans. Cybern. 48 (5) (2018) 1460–1473.

Wei Wei received his Ph.D. degree in Computer Science from Shanxi University in 2012. He is currently an Associate Professor with the School of Computer and Information Technology, Shanxi University. His research interest is in the areas of cluster analysis and granular computing. He has published more than 20 journal papers in his research fields.

Jiye Liang received the M.S. and Ph.D. degrees from Xian Jiaotong University, Xian, China, in 1990 and 2001, respectively. He is currently a Full Professor of School of Computer and Information Technology and Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education at Shanxi University. His current research interests include computational intelligence, granular computing, data mining and knowledge discovery. He has published more than 180 journal paper in his research fields.

Xinyao Guo is currently a Ph.D. candidate in the School of Computer and Information Technology in Shanxi University. He received the B.S. degree from Shanxi University, in 2016. He His research interests are focused on clustering and rough sets.

Peng Song received the M.S. degree from Central University of Finance and Economics, Beijing, China, in 2006, and received the Ph.D. degree from Shanxi University, Taiyuan, China, in 2012. He is currently an Associate Professor of School of Economics and Management and Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education at Shanxi University. His current research interests include intelligent decision, granular computing and data mining. He has published more than 20 journal papers in his research fields.

Yijun Sun received a dual B.S. degree in electrical and mechanical engineering from Shanghai Jiao Tong University in 1995, and obtained M.S. and Ph.D. degrees in electrical engineering from the University of Florida, in 2003 and 2004, respectively. He is Associate Professor in bioinformatics at the Department of Microbiology and Immunology and the New York State Center of Excellence in Bioinformatics and Life Sciences. His research interests are primarily on machine learning, data mining, bioinformatics and their applications to microbial ecology and cancer informatics.