doi:10.11920/xnmdzk.2025.01.010

基于 MapReduce 的分类数据增量子空间聚类研究

庞　宁
(太原科技大学应用科学学院,山西 太原　030024)

摘　要:基于细粒度属性子空间构建方法提出一种适用于分类数据的并行增量聚类算法 SUC,该算法采用属性值-簇相似度度量方法,强化重要属性值对于类簇紧凑程度的正向影响力;在增量聚类阶段,更新属性权值,迭代形成增量类簇;采用 MapReduce 编程框架,实现算法 SUC 两阶段的并行化. 在人工合成数据集、UCI 数据集和真实数据集上,实验验证了算法的准确性、有效性和可扩展性.

关键词:增量子空间聚类; 细粒度属性权重; MapReduce 聚类; 分类数据
中图分类号:TP311.13　　**文献标志码**:A　　**文章编号**:2095-4271(2025)01-0071-06

Research on incremental subspace clustering of categorical data based on MapReduce

PANG Ning
(Taiyuan University of Science and Technology, School of Applied Science, Taiyuan 030024, China)

Abstract: Based on the fine-grained attribute subspace construction method, a parallel incremental clustering algorithm SUC was proposed for categorical data. The algorithm adopted the attribute value-cluster similarity measurement method to strengthen the positive influence of important attribute values on the compactness of clusters. In the incremental clustering results, the attribute weights were updated and the incremental cluster was iteratively formed. Using the MapReduce programming framework, the two-stage parallelization of algorithm SUC was realized. The accuracy, effectiveness, and scalability of the algorithm were experimentally validated on artificially synthesized datasets, UCI datasets, and real datasets.

Keywords: incremental subspace clustering; fine-grained attribute weight; MapReduce clustering; categorical data

聚类是将数据按照相似性划分为若干簇的无监督学习方法. 簇内相似、簇间远离是聚类总目标. 作为数据挖掘的重要研究内容,聚类被广泛应用于模式识别、图像处理和自然语言处理等领域.

分类数据是侧重于描述数据类别和标签的数据类型,具有有限、离散、多维的特性,无法直接进行数值计算. 随着互联网和数据采集技术的发展,分类数据呈现高维化,动态化,海量化的特征. 传统聚类算法无法直接处理海量高维分类数据. 以海量动态增长的分类数据为研究对象,已成为聚类问题新的研究热点.

1　相关工作

1.1　增量子空间聚类

针对高维数据的聚类问题,基于属性子空间的相关研究已成为重要的研究思路之一[1-5]. 文献[1]提出一种基于分布式低秩的增量式稀疏子空间聚类算法,有效利用了历史聚类结果,减少了聚类的运行时间;李凯等人[2]提出了一种新的子空间聚类算法,将

收稿日期:2024-03-12
作者简介:庞宁(1979-),女,副教授,博士,研究方向:数据挖掘与并行计算. E-mail:pn529@126.com
基金项目:山西省自然科学研究面上项目(20210302123224); 太原科技大学博士启动课题(20202066)

权向量的负结构α-熵与高斯混合模型相结合,获得了较好的聚类结果;基于属性分组技术和多目标聚类质量函数,文献[3]给出了一种子空间聚类算法,该算法利用同组属性相关性度量属性权重值,通过迭代达到最优化聚类质量目标,取得较好聚类效果;基于K-means算法,文献[4]和文献[5]分别结合优化模糊模型和最大信息熵的方法,给出了不同的属性权值度量方法,解决数值型数据的聚类问题.

近几年,增量式聚类研究成果颇丰[6-10]. 何云斌[6]等人提出了一种基于角度度量的动态增量聚类算法,解决了随机获取类簇中心而引起的聚类结果不稳定的缺陷;文明瑶[7]等人给出了一种基于模糊c-均值聚类方法,使用Relief算法度量属性权重,在增量数据下,保持稳定的聚类精度;文献[8]提出一种基于群体智能的增量软聚类算法,计算增量文本在局部区域内的群体相似性,从而快速实现增量文本聚类;曲福恒[9]等人针对Ball k-means算法缺陷,提出了一种基于多球分裂的增量式k-means聚类算法,该算法解决了初始点敏感问题,同时,有效降低算法的时间复杂度;文献[10]采用基于数据均衡的快速分裂方法产生增量聚类中心,提出一种增量式MinMax k-means聚类算法,该算法在计算效率和求解精度上均优于对比算法.

1.2 并行聚类算法

针对海量数据聚类问题,并行化聚类算法是行之有效的方法[11-13]. 基于具有层次聚类特性的RSOM树方法,文献[11]给出了一种增量分布式并行聚类算法,对高维海量数据建立聚类索引;刘仁芬[12]等人利用信息熵进行稀疏降维,提出一种基于Spark增量式聚类算法,挖掘数据特征之间的关联性,完成高维数据增量式聚类过程;文献[13]提出了一种基于MapReduce模糊C均值聚类算法,与串行传统模糊C均值和基于MapReduce的K均值聚类算法进行了实验比较,证明该算法的高效性. 上述方法均未涉及分类数据,分类数据增量聚类并行化研究的研究成果甚少.

针对分类数据,本文提出一种基于MapReduce两阶段增量子空间聚类方法SUC,该方法包括原始聚类和增量聚类两阶段,原始聚类阶段采用基于细粒度属性权重度量方法,在原始历史数据上构建属性子空间,使用一种新的属性值-类簇相似度度量方法,优化聚类目标;在增量聚类阶段,利用原始聚类结果,对增量数据,实现子空间聚类;采用MapReduce并行框架,实现聚类过程的并行化.

本文的主要贡献:1)提出一种基于属性值-类簇相似度的聚类目标;2)给出一种基于细粒度属性权重计算方法;3)给出一种增量子空间聚类的并行化方法.

2 问题描述

假设原始数据集DB包含N个数据点,每个数据点O_i均由D个分类型属性值x_{ij}($1\leqslant i\leqslant N, 1\leqslant j\leqslant D$)组成. 数据不是一次性读入存储器,而是以分批增量方式被输入. 算法SUC主要解决的问题是:以增量分类数据作为研究对象,利用原始数据的子空间聚类结果,无须重复读取和处理原始数据,直接更新子空间内的权重值,并对增量数据进行聚类. 本文主要符号表示见表1.

表1 符号表示

Table 1 Symbol description

符号表示	含义
O_i	原始数据集DB中第i个数据点
A_j	第j维属性
x_{ij}	第i个数据点在第j维属性上属性取值
N	数据总量
D	属性维度总量
W_{ij}	属性值x_{ij}的权值
ΔO_i	增量数据集Δdb中第i个数据点
ΔW_{ij}	增量更新后的权重值

3 基于MapReduce增量子空间聚类

3.1 原始聚类

对于原始数据,本文采用基于细粒度子空间聚类算法进行数据划分. 传统子空间聚类方法往往将整个属性维度作为权重度量粒度,属性对类簇的区分度不够. 算法SUC采用了一种基于细粒度的属性取值赋权方法,根据属性值频次细化属性子空间刻画粒度,构建属性子空间. 结合属性值频次,给出属性权重计算方法,见公式(1),该权重度量方法可有效过滤掉出现频繁或稀少的属性值对聚类的干扰.

$$W_{ij}=P(x_{ij})\times\lg\{N\times P(x_{ij})\times(1-P(x_{ij}))+1\}, \quad (1)$$

其中,x_{ij}是第 i 个数据点在第 j 属性上的属性取值,W_{ij}表示分类属性值 x_{ij}的权重,$P(x_{ij})$是 x_{ij}在属性 A_j上的出现概率值,N 为原始数据总数. 属性权重会根据不同属性取值分布特征的变化而动态改变. 同时,出现次数过于频繁或稀疏的属性值对于聚合类簇的作用有限,该赋权方法会为其自动赋予较低的权重.

以最大化簇集函数为聚类目标,进行子空间聚类,聚类目标是判断数据划分的依据,以数据点与簇相似度之和的最大值作为聚类目标,具体表示如式(2):

$$QS(C)=\arg\max[\sum_{j=1}^{k}\sum_{i=1}^{N}sim(O_i,C_j)], \qquad (2)$$

其中,$sim(O_i,C_j)$是数据点 O_i与簇 C_j的相似度,k 为类簇总数,N 为数据总数. 数据点 O_i与簇 C_j的相似程度可以用属性子空间下各属性值 X_{im}到簇 C_j的相似度量化,具体如式(3):

$$sim(O_i,C_j)=\sum_{m=1}^{D}W_{im}s(x_{im},C_j), \qquad (3)$$

其中,W_{im}是属性值 x_{im}的权重,由公式(1)计算而得,属性值 - 类簇相似度 $s(x_{im},C_j)$是数据点 O_i在第 m 个属性上的取值与簇 C_j的相似度,具体表示如式(4):

$$s(x_{im},C_j)=\frac{count(x_{im},A_m,C_j)}{count(C_j)}. \qquad (4)$$

$count(x_{im},A_m,C_j)$表示簇 C_j内,在第 m 个属性 A_m取值为 x_{im}的数据点数量,$count(C_j)$是簇 C_j内的数据点总数. 采用属性值 - 类簇相似度,算法 SUC 强化了重要属性值在类簇形成过程中对于簇内紧凑程度的正向影响力.

本文在所构建属性子空间的基础上,随机选择 k 个数据,作为 k 个簇的初始点,根据最大化聚类目标的原则,依次将数据点划分至各簇内,迭代数据划分过程,直到迭代前后两次的簇内数据分布保持稳定为止,具体算法过程描述见算法 1.

算法 1　子空间聚类算法

输入:原始数据集 DB

输出:原始数据聚类结果 C

①for i = 1 to N do
　　for j = 1 to D do
利用公式(1),计算数据点 O_i上各属性值的权重 W_{ij};
　　endfor
endfor

②随机选择 k 个数据点,作为 k 个簇 C 的初始元素;

③for i = 1 to N do
　　for j = 1 to k do
根据公式(2),寻找与数据点 O_i最接近的簇 C_j,并划归至该簇;
　　endfor
endfor

④if $C_{NEW}\neq C_{OLD}$ then
重复步骤 3;

⑤输出最后的聚类结果.

3.2　增量子空间聚类

利用原始数据的属性值权重和聚类结果,根据增量数据的分布特征,进行增量聚类. 增量聚类主要分为权重更新,增量数据子空间聚类,迭代调整簇集结构. 权重更新是基于原始属性频次,更新并计算增量数据各属性的权重;依次计算各增量数据与原始类簇之间的相似度,以最大化聚类目标公式(2)为聚类目标,将增量数据划分至最相似的类簇中;迭代调整各类簇内的数据分布,直至所有数据与类簇相似度之和达到最大化.

3.3　增量子空间聚类并行化

基于 MapReduce 并行子空间增量式聚类可分为两个阶段:原始数据聚类并行化阶段和增量数据聚类并行化阶段. 如图 1 所示,每个阶段由一个 Job 完成,并由 Map 和 Reduce 两部分组成.

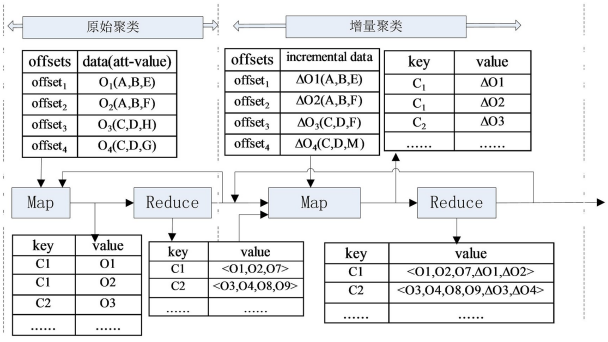


图 1　基于 MapReduce 子空间增量式聚类算法整体框架

Fig. 1 Overall framework of incremental subspace clustering algorithm based on MapReduce

3.3.1 原始聚类并行化阶段

原始聚类并行 job 采用子空间聚类算法实现原始数据 DB 的聚类过程,具体操作如下:

步骤 1　Hadoop 并行系统使用默认分区函数,将原始数据 DB 划分为若干数据分区 DB_k,DB_k使用 < key,value > 的形式保存数据;

步骤 2　Map 统计各属性值的出现频次,计算其

权值 W_{ij},选取 k 个数据点作为簇的初始元素;

步骤3 根据最大化聚类目标函数的原则,在不同的属性子空间下,对数据 O_i 实现簇集划分,输出形如 $<\text{clusterID}, O_i>$ 的初步子簇结果;

步骤4 Reduce 负责从不同计算节点获取具有相同 clusterID 值的数据点,迭代调整初步子簇,形成 $<\text{clusterID}, \text{list}\{O_i \mid O_i \in C_{\text{clusterID}}\}>$ 作为原始数据聚类结果传入 Job2 中.

3.3.2 增量聚类并行化阶段

增量聚类主要包括权重更新和增量数据聚类两部分. 利用第一阶段获取的属性值频次和权重,算法 SUC 可以计算增量数据的属性权值. 在原始数据聚类结果基础上,迭代调整增量数据的数据划分.

增量聚类并行化 job 利用原始聚类结果,对增量数据 Δdb 实现簇集划分,具体操作如下:

步骤1 根据增量数据分区 Δdb_k 中各属性值的出现频次,更新属性值的权重值 ΔW_{ij};

步骤2 Map 依次读取增量数据 ΔO_i,利用公式(2),将其划分到已有原始簇集 $C_{\text{clusterID}}$ 中或产生一个新簇,形成中间结果 $<\Delta\text{clusterID}, \Delta O_i>$;

步骤3 按照簇号 ΔclusterID,Reduce 负责将具有相同簇号 ΔclusterID 的数据点合并保存.

4 实验测试与结果分析

实验是在具有6个节点的 Hadoop 集群环境下完成的. 所有计算节点均安装了 Intel i7-10750 CPU, Centos7 操作系统,Java JDK 21.0.2 和 Hadoop3.1.3 集群平台.

4.1 实验数据

用于实验的数据可分为三类:人工合成数据DB1 ~ DB4、UCI 数据以及真实光谱数据,其中,人工合成数据主要测试算法在不同类型数据上的聚类精度和并行效率,DB1 类簇之间的属性子空间完全独立,数据集 DB2 的某些类簇属性子空间之间存在交集,DB3 属于不均衡数据集,DB4 主要测试算法的并行性能. UCI 数据和光谱数据主要负责测试算法 SUC 在真实数据上的聚类性能. 实验数据具体信息见表2.

表2 实验数据集

Table 2 Experimental Dataset

数据集	名称	数据量	属性	类簇数
人工合成	DB1	5 000	100	6
	DB2	10 000	50	6
	DB3	5 000	100	6
	DB4	$1*10^6$	50	6
UCI	Splice	3 190	61	3
	Covertype	581 012	52	7
	Mushroom	8 124	22	2
光谱数据	Spectrum	520 403	91	5

4.2 对比算法

选取三个子空间聚类算法 PROCAD[14], MWKM[15],DHCC[16] 与本文算法 SUC 做对比实验分析,其中,算法 PROCAD 和 DHCC 均为无参算法,EWKM 的主要参数:β 设为2,T_v 与 T_s 均取1,各算法性能指标均为执行50次算法的性能最优值.

4.3 评测指标

本文采用 ARI(Adjusted Rand Index),纯度 Purity 和聚类错误率 ER(the set matching error)作为外部评价指标. 其中,指标 ARI 和 Purity 值越大,聚类效果越好;指标 ER 值越小,被错误归类的数据越少.

4.4 聚类精度

该实验主要包括:1)测试算法 SUC 在三个合成数据集 DB1 ~ DB3 上的聚类精度;2)对比算法 SUC 与其他三个对比算法在四个真实数据集上的精度差异. 如图2所示,在三种不同类型的合成数据集上,算法 SUC 在 DB1 和 DB3 上的聚类性能略胜一筹,说明算法 SUC 未受到不平衡数据分布的影响. 主要原因是:属性值 – 类簇相似度可以有效强化关键属性值对类簇紧凑程度的表现能力,同时,算法 SUC 采用基于属性值 – 类簇相似度迭代聚类过程,避免了基于汉明距离度量的聚类方法在非平衡数据聚类均匀化的问题.

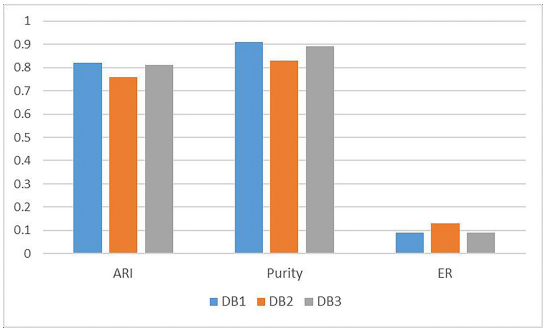


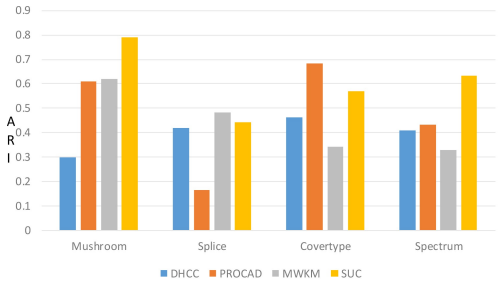
图 2　算法 SUC 在合成数据集 DB1 ~ DB3 上的效果对比

Fig. 2　The effectiveness comparison of algorithm SUC on synthetic datasets DB1 ~ DB3

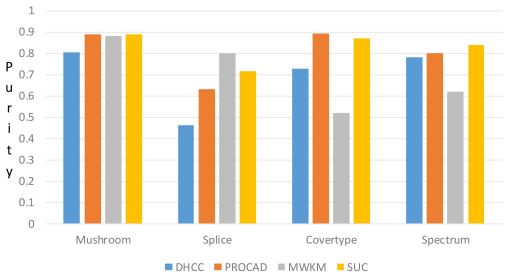
图 3 给出算法 SUC 与对比算法 DHCC、算法 PROCAD 和算法 MWKM 在真实数据 Mushroom、Splice、Covertype 和光谱数据 Spectrum 上的聚类性能差异. 由图可知,算法 SUC 在 Mushroom 和光谱数据集上的各项聚类性能指标均优于其他算法,主要原因是 SUC 采用基于细粒度的属性取值赋权方法细化了属性子空间在不同类簇上的聚类作用力,尤其是在 Mushroom 数据和 Spectrum 光谱数据上,各属性值域范围较大,细化属性值的权重区分度更有利于聚类过程. 在数据集 Covertype 上,算法 PROCAD 性能最佳,主要原因是该算法擅长处理具有二进制维度特征的数据集. 而算法 MWKM 是基于 K-MODE 的加权算法,对于 Covertype 数据和 Spectrum 光谱数据上的表现不佳,主要原因是由于这两类数据均为不均衡数据,类簇内的数据比例差距较大,极易导致大簇边缘数据中心化.

4.5　并行效率

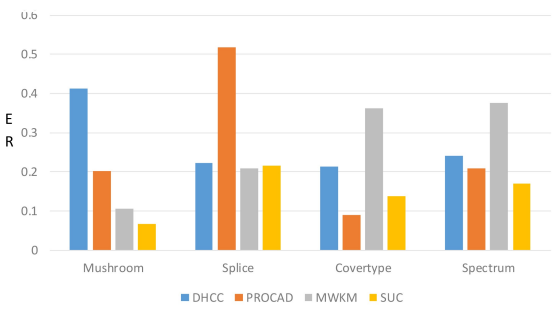
本实验主要在合成数据 DB4 上对算法 SUC 的并行效率进行测试. 图 4 显示了算法 SUC 在数据量扩展性实验上的结果. 由图可知,随着数据量的增加(从 $1 * 10^5$ 到 $8 * 10^5$),算法 SUC 在不同计算节点上的运行时间基本呈现线性增长,而计算节点的增加可以有效降低由数据量增加带来的时间成本.



(a)　ARI



(b)　Purity



(c)　ER

图 3　各算法在真实数据集的聚类性能对比图

Fig. 3　Clustering performance comparison of various algorithms on real datasets

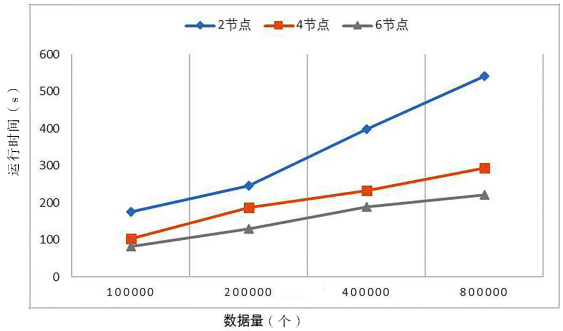


图 4 数据量对算法 SUC 效率的影响

Fig. 4 The impact of data volume on the efficiency of algorithm SUC

图 5 显示属性维度从 10 维增加到 50 维,算法 SUC 运行时间变化趋势. 如图 5 所示,SUC 运行时间的变化呈线性增长趋势. 基于细粒度属性权重计算方法会增加扫描属性取值的次数;聚类过程所采用的属性值 - 簇的相似度计算方法也会随着属性量的增加而提高聚类时间成本. 从图 5 可知,数据节点的增加,有助于降低运行时间与维度比的增长斜率,表明算法 SUC 的并行化可以有效解决海量高维数据聚类问题.

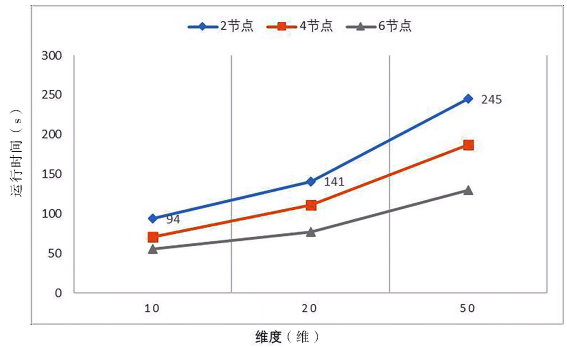


图 5 维度对算法 SUC 效率的影响

Fig. 5 The impact of dimension on the efficiency of algorithm SUC

5 总结

针对海量动态增长的分类数据,本文提出一种基于 MapReduce 增量子空间聚类算法 SUC,该算法包括原始聚类和增量聚类两个阶段,在原始聚类中,采用基于细粒度权重度量方法构建属性子空间,结合属性值 - 簇相似度度量方法,迭代调整数据分布,以达到最优类簇质量目标;增量聚类阶段无须重复读取和处理原始数据,在原始类簇的基础上,更新属性值权重,形成增量类簇;在 MapReduce 并行框架下,实现两阶段增量式聚类. 在人工合成数据集、UCI 数据集和真实数据集上,实验验证了算法的准确性、有效性和可扩展性.

参考文献

[1]许凯,吴小俊,尹贺峰. 基于分布式低秩表示的子空间聚类算法[J]. 计算机研究与发展,2016,53(7):1605-1611.

[2]李凯,张可心. 结构 α-熵的加权高斯混合模型的子空间聚类[J]. 电子学报,2022,50(3):718-725.

[3]庞宁,靳黎忠. 基于属性分组的子空间聚类算法研究[J]. 西南民族大学学报(自然科学版),2023,49(6):653-660.

[4]CHAN E Y,CHING W K,NG M K,et al. An optimization algorithm for clustering using weighted dissimilarity measures[J]. Pattern Recognition,2004,37(5):943-952.

[5]JING L P,NG M K,HUANG J Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data[J]. IEEE Transactions on Knowledge and Data Engineering,2007,19(8):1026-1041.

[6]何云斌,孙暖,万静,等. 角度度量的动态增量聚类算法[J]. 哈尔滨理工大学学报,2019,24(6):109-116.

[7]文明瑶,廖伟国. 基于机器学习的不确定数据增量式挖掘算法[J]. 计算机仿真,2021,38(11):290-294.

[8]刘艳,周斌. 增量文本软聚类速度改善算法设计及仿真[J]. 计算机仿真,2022,39(8):524-528.

[9]曲福恒,钱超越,杨勇,等. 基于多球分裂的增量式 K-MEANS 聚类算法[J]. 吉林大学学报(工学版),2022,52(6):1434-1441.

[10]胡雅婷,陈营华,宝音巴特,等. 一种增量式 MINMAX K-MEANS 聚类算法[J]. 吉林大学学报(理学版),2021,59(5):1205-1211.

[11]夏胜平,吕小军,刘建军,等. 基于集群的并行分布式聚类及其应用[J]. 郑州大学学报(理学版),2006,38(4):33-40.

[12]刘仁芬,杨凤丽,王霞. 基于改进 SPARK 技术的高维数据增量式聚类算法[J]. 计算机仿真,2022,39(12):383-386 +444.

[13]SARDAR T H,ANSARI Z. MapReduce-based fuzzy C-means algorithm for distributed document clustering[J]. Journal of the Institution of Engineers (India):Series B,2022,103(1):131-142.

[14]BOUGUESSA M. Clustering categorical data in projected spaces[J]. Data Mining and Knowledge Discovery,2015,29(1):3-38.

[15]BAI L,LIANG J Y,DANG C Y,et al. A novel attribute weighting algorithm for clustering high-dimensional categorical data[J]. Pattern Recognition,2011,44(12):2843-2861.

[16]XIONG T K,WANG S R,MAYERS A,et al. DHCC:Divisive hierarchical clustering of categorical data[J]. Data Mining and Knowledge Discovery,2012,24(1):103-135.

(责任编辑:张阳,殷锋,付强,和力新,肖丽;英文编辑:周序林,郑玉才)