第 34 卷 第 11 期 计 算 机 工 程 2008 年 6 月
Vol.34 No.11 Computer Engineering June 2008
·软件技术与数据库· 文章编号：1000—3428(2008)11—0072—02 文献标识码：A 中图分类号：TP391

基于邻接图的离群数据聚类算法

金义富[1,2]，朱庆生[2]，邹咸林[2]
(1. 湛江师范学院信息学院，湛江 524048；2. 重庆大学计算机学院，重庆 400044)

摘 要：离群数据是数据中的小模式，因其固有的少数据与稀疏性等特征，使得基于距离或基于统计等常规聚类方式不适用于对离群数据的分类。该文根据离群对象关键域子空间的重合度，定义了离群共享属性集与离群相似度等概念，提出β-离群簇分析技术。通过构建离群邻接图并将其稀疏化，将β-离群簇搜索与相应的离群邻接图的最大完全子图搜索一一对应，给出一种基于邻接图的离群数据聚类算法。算例及实验结果表明，该方法具有较高的效率及良好的直观性。

关键词：离群数据；关键域子空间；离群邻接图；聚类算法

Clustering Algorithm of Outliers Based on Adjacency Graph

JIN Yi-fu[1,2], ZHU Qing-sheng[2], ZOU Xian-lin[2]
(1. School of Information, Zhanjiang Normal College, Zhanjiang 524048; 2. College of Computer, Chongqing University, Chongqing 400044)

【Abstract】 Outliers are small pattern in data space. General clustering approaches, such as distance-based and statistics-based, are not adapted to classification of outliers because of their characteristic of fewness data and sparseness. This paper defines concepts of outlying shared attribute and outlying similarity based on the key attribute subspace of an outlier and proposes an analysis technique on β-cluster of outliers. An algorithm for clustering of outliers based on adjacency graph is put forward in this paper. Its main idea includes establishment and simplification of outlying adjacency graph in which a maximum complete subgraph is corresponding with a β-cluster of outliers. Examples and experimental results show that the algorithm is intuitionistic and well efficient.

【Key words】 outliers; key attribute subspace; outlying adjacency graph; clustering algorithm

1 概述

离群数据(outliers)是远离常规数据对象的数据[1]，它们与多数常规对象有明显差异。现有对离群数据的研究主要集中在离群数据挖掘，研究人员已经提出了大量的离群挖掘算法[2-4]，其挖掘的目的一般是为了通过去除被发现的离群对象获得更好的数据质量，力图为常规数据挖掘与分析提供更稳定可靠的结果，而涉及对挖掘出的离群数据进行进一步研究的文献较少。离群数据分析是针对已挖掘出的离群数据研究其离群特征或出现原因、离群对象分类与聚类以及对时序数据的离群趋势进行预测等。文献[5]以属性子空间为背景，按对象与属性统一的观点对离群数据对象特性进行了分析，文中离群点被分为平凡(trivial)和非平凡(non-trivial)离群点两类。文献[6]从更一般的角度提出了离群对象关键域子空间(Key Attribute Subspace, KAS)概念，以及离群数据聚类(Clustering of Outliers based on KAS, COKAS)算法，本文在此基础上提出了一种基于离群邻接图的聚类方法。

2 β-离群簇

设数据集$X=(x_{ij})_{n\times d}$, $x_i=(x_{i1},x_{i2},\cdots,x_{id})$为第$i$个数据对象；属性集$A=\{A_1,A_2,\cdots,A_d\}$，其中，$n$为数据规模；$d$为维数。设$L^d=A_1\times A_2\times\cdots\times A_d$为全部属性集$A$构成的$d$维数字空间，$XO_d$为$X$在整个属性空间$L^d$上挖掘的离群数据集。离群数据聚类是指对$XO_d$中的全部离群对象按一定方式进行分类，从而探索各类离群数据的特征及数据离群的原因。

定义 1(离群共享属性集) 设$o_i, o_j\in XO_d$，其关键域子空间分别用$kas(o_i)$, $kas(o_j)$表示，如果$\exists H\subseteq A$, $H\neq\varnothing$且$H\subseteq kas(o_i)\cap kas(o_j)$，则称属性子集$H$为对象$o_i, o_j$的离群共享属性集(Outlying Shared Attributes, OSA)，如图 1 所示，同时如果$kas(o_i)\cap kas(o_j)\neq\varnothing$，则$kas(o_i)\cap kas(o_j)$是对象$o_i, o_j$的元素个数最多的离群共享属性集，称$kas(o_i)\cap kas(o_j)$为最大离群共享属性集，记为$losa(o_i, o_j)$。

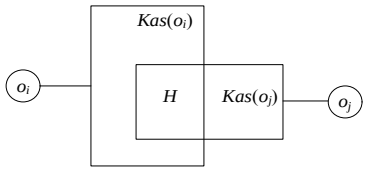

图 1 两个离群对象的关联

定义 2(离群共享属性相似度) 设离群对象$o_i, o_j\in XO_d$，其关键域子空间分别为$kas(o_i)$, $kas(o_j)$；$losa(o_i, o_j)$为其最大离群共享属性集，则称：

$$osim(o_i,o_j)=|losa(o_i,o_j)|/|kas(o_i)\cup kas(o_j)| \quad (1)$$

为对象o_i, o_j的离群共享属性相似度(OSA-based Similarity, OSAS)，简称为离群相似度。

可以看出，$0\leq osim(o_i,o_j)\leq 1$，而$osim(o_i,o_j)=0$ 当且仅当$losa(o_i,o_j)=\varnothing$, $osim(o_i,o_j)=1$ 当且仅当$kas(o_i)=kas(o_j)$。

定义 3(β-离群簇) 设离群数据子集$CO\subseteq XO_d$，给定参数β，$0<\beta\leq 1$，如果$osim(CO)\geq\beta$，称CO为一个β-离群簇。

基金项目：重庆市自然科学基金资助项目(2005BB2224)
作者简介：金义富(1969－)，男，副教授、博士研究生，主研方向：数据挖掘，软件工程；朱庆生，教授、博士生导师、博士；邹咸林，副教授、博士研究生
收稿日期：2007-07-23 **E-mail：**constudy@21cn.com

离群聚类搜索离群集XO_d中满足给定最低离群共享属性相似度β的离群簇，从而可以把关键域子空间应用于离群簇分析，在更细粒度的离群簇范围内讨论离群数据特征。

3 离群邻接图

以离群集XO_d中m个数据点两两之间的离群共享属性相似度形成$m\times m$相似度矩阵，从而构建一个相应的m个结点的稠密图，所有结点对之间都有一条边相连。

定义 4(离群邻接图) 离群邻接图(Outlying Adjacency Graph, OAG)是一个简单无向图，它的结点为离群对象，边的权值为其两端结点所代表对象的离群共享属性相似度。

一个β-离群簇对应于离群邻接图的一个完全子图，且其中边的最小权值不低于β。如果离群邻接图有两个结点无边相连或边的权值小于β，则这两个结点不可能在同一个β-离群簇内。这些结论提供了基于离群邻接图的离群数据聚类算法(outlying Adjacency Graph-based Outliers Clustering algorithm, AGOC)的基本策略，即首先获得离群相似度矩阵，构建离群邻接图，然后对离群邻接图进行稀疏化处理：断开权值低于给定最小离群相似度β的边；从稀疏化后的邻接图中寻找完全子图，判断是否为β-离群簇。

4 AGOC 算法

AGOC 算法根据最低离群相似度阈值β对初始离群邻接图进行稀疏化，最后的连通分支即是β-离群簇。算法根据边的权值启发式搜索完全子图，即使在离群集规模较大时也能较快地获得最终聚类结果。AGOC 算法描述如下：

输入 数据集X，离群数据集XO_d，参数β

输出 β-离群簇CO_k及CO_k的最大离群共享属性集$losa(CO_k)$

Step1 获取离群集XO_d中每个对象的关键域子空间$kas(o_i)$;

Step2 $CO[0]=\varnothing$; $losa[0]=\varnothing$; $m=|XO_d|$;

Step3 For i=1 To m Do If $kas(o_i)=\varnothing$ Then $CO[0]=CO[0]\cup\{o_i\}$; $XO=XO_d-CO[0]$; /*去噪*/

Step4 For each pair $o_i, o_j\in XO$ Do {

$osim[i, j]\leftarrow$ 离群相似度$osim(o_i, o_j)$;

If $osim[i, j]$=1 Then {

合并对象o_i, o_j为同一离群簇;

$XO= XO-\{o_j\}$;}

/* 设本步完成后获得c个离群簇$CO_1[1], CO_1[2],\cdots, CO_1[c]$ */

Step5 If β=1 Then {

For k=1 to c Do {

$CO[k]=CO_1[k]$;

$losa[k]\leftarrow$簇 $CO[k]$的 KAS;}

$m\leftarrow XO$中未入CO_1的单个对象数; For $k=c$+1 To $c+m$ Do {

$CO[k]\leftarrow XO$中未入CO_1的对象;

$losa[k]\leftarrow$簇 $CO[k]$的 KAS;}

转 Step12;}

/* 以下各步处理β<1 的情形 */

Step6 设k=0;对处理后的XO中的元素下标重新编号; $m=|XO|$，仍设$XO=\{o_1, o_2,\cdots, o_m\}$; 利用$XO$及所有$osim[i, j]$构建离群邻接图$ORG(V, E)$，其中，$V$为结点集；$E$为边集；$e(i, j)\in E$为连接结点$o_i$与$o_j$的边;

Step7 For each pair $o_i, o_j\in V$ Do IF $osim[i, j]<\beta$ Then 从ORG中删除$e(i, j)$;

Step8 $e(i, j)\leftarrow e(i, j)$：$max\{osim[i, j]: e(i, j)\in E\}$;

If $e(i, j)$为空 Then 转 Step11;

k++;

Step9 $CG(V', E')\leftarrow ORG$ 中包含 $e(i, j)$的团;

Step10 If $osim(V')\geq\beta$ Then {$CO[k]=V'$; $losa[k]= losa(V')$; 从ORG 中去除子图 $CG(V', E')$及相连的所有边; 转 Step8;}

Else {断开 $CG(V', E')$中权值最小的边; 转 Step9;}

Step11 For each $p\in[1, k]$ Do If $\exists q$, $1\leq q\leq c$，$CO[p]$中的某一个对象存在于$CO_1[q]$中 Then

$CO[p]=CO[p]\cup CO_1[q]$;

Step12 输出每一组 k, $CO[k]$, $losa[k]$;

算法结束。

AGOC算法总的时间复杂度为$O(mT_1+m^2)$。 由于通过去除噪声与合并关键域子空间相等的类降低了结点数，同时该算法在找到一个β-离群簇后即从当前的离群邻接图中删除掉相应的完全子图，使得下一次搜索范围大大缩小，从而进一步提高了算法效率。

5 实验分析

利用实际的数据集对AGOC算法与文献[6]提出的COKAS算法进行了对比实验。测试数据集来自广东某市移动通信业务数据库，选其中一个子集 18 个属性 1×10^4条记录组成数据对象集X，离群挖掘算法采用文献[2]提出的基于分区的方法，测试中共发现 20 个离群点，用o_1~o_{20}代表离群对象，表 1 列出了COKAS和AGOC两种算法的结果离群簇及相应的最大离群共享属性集。

表 1 COKAS 和 AGOC 算法的聚类结果(β=0.7)

	COKAS 算法		AGOC 算法	
	$CO[k]$	$losa(CO[k])$	$CO[k]$	$losa(CO[k])$
簇 1	o_4,o_{13}	$\varnothing$	o_4,o_{13}	$\varnothing$
簇 2	$o_1,o_5,o_6,$ $o_{12},o_{14},$ o_{16},o_{20}	*call_duration* *call_count* *gn_charge*	$o_1,o_5,o_6,$ $o_{12},o_{14},$ o_{15},o_{20}	*call_duration* *call_count* *gn_charge*
簇 3	$o_2,o_9,$ $o_{10},o_{15},$ o_{18}	*call_duration* *call_count* *sn_charge*	$o_2,o_9,$ $o_{10},o_{16},$ o_{18}	*call_duration* *call_count* *sn_charge*
簇 4	$o_3,o_8,$ o_{17},o_{19}	*call_duration* *gj_charge*	$o_3,o_8,$ o_{17},o_{19}	*call_duration* *gj_charge*
簇 5	o_7,o_{11}	*newfun_fee*	o_7,o_{11}	*newfun_fee*

在最低离群相似度β=0.7 时，聚类结果为 5 个离群簇，第 1 个为没有关键域子空间的噪声簇$CO[0]=\{o_4,o_{13}\}$。两种算法的结果中各离群簇的最大离群共享属性集完全一样，离群簇也都几乎一致，仅对象o_{15}和o_{16}在两种算法结果中所属离群簇相互交换了，这是由于这两种算法过程中对加入离群簇的对象选择顺序略有差异所致。

本文选取可能含有不同离群对象数量m的数据集对两种算法进行测试，图 2 显示了$n=5\times10^4$条记录而维数d=15 时算法运行时间与发现的离群点数m的总体增长趋势(图中运行时间包含执行离群挖掘算法所需要的时间)。

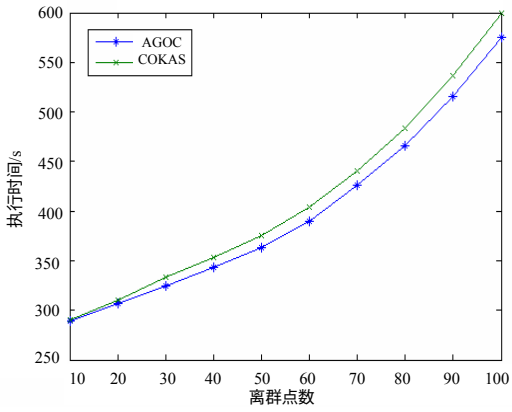

图 2 离群点数不同时算法执行时间比较

(下转第 76 页)