文章编号: 1001-0920 (2011) 12-1846-05

# 一种网格聚类的边缘检测算法

张鸿雁[1], 刘希玉[1], 付 萍[2]

(1. 山东师范大学 管理科学与工程学院，济南 250014；2.山东省济南市公安消防分局 防水处，济南 250000)

**摘 要:** 提出一种应用在基于密度的网格聚类算法中的边缘检测方法. 通过对密集单元格周围的稀疏单元格进行检测, 将符合条件的稀疏单元格归并到密集单元格中. 该方法不仅可以修正由于网格划分失去的数据边缘部分, 而且可以修正由于网格间隔和密度阈值设置不合理所导致的误差, 同时所消耗的时间对整个聚类过程影响不大, 是一种效果明显的网格聚类的改进方法.

**关键词:** 网格聚类；数据挖掘；CLIQUE 算法

**中图分类号:** TP391 **文献标识码:** A

# Boundaries detection algorithm for grid-based clustering

*ZHANG Hong-yan*[1], *LIU Xi-yu*[1], *FU Ping*[2]

(1. School of Management Science and Engeering，Shandong Normal University，Ji'nan 250014，China；2．Fire Department，Ji'nan Municipal Public Security Bureau，Ji'nan 250000，China. Correspondent：ZHANG Hong-yan, E-mail：zswanz@yahoo.com.cn)

**Abstract:** An boundaries detection algorithm for grid-based clustering is proposed in this paper, which can find lost edges near by the density units. It is good for the special dataset with the irregular, and it can amend the errors because of the unreasonable threshold value or grids. At the same time, the cost time has little impact on clustering process, so the improved method is effective.

**Key words:** grid-based clustering；data mining；CLIQUE algorithm

## 1 引 言

基于网格的聚类方法采用一种多分辨率的网格数据结构, 将空间量化为有限数目的单元, 从而所有的聚类操作都能在由这些单元组成的网格上进行. 在目前所有的聚类方法中, 网格聚类方法不考虑每一个点, 把包含许多点的单元作为处理的对象, 使得处理时间独立于数据对象的数目, 仅依赖于量化空间中每一维上的单元数目, 因此对数据输入顺序没有要求. 目前典型的网格聚类方法有: STING 算法、WaveCluster 算法以及 CLIQUE[1]算法. 虽然网格聚类具有效率高、聚类结果与输入数据顺序无关、可扩展性好的优点, 但也存在不足之处, 影响最大的有以下两方面:

1) 聚类参数的选定, 例如每一维被划分的个数和密度阈值的选择都会影响聚类的效果;

2) 网格一般被划分为矩形, 使得聚类的结果边缘都是直线, 从而对于平滑曲线边缘或者较稀疏的边缘将会被忽略, 导致聚类精度和质量下降.

对于 CLIQUE 算法, 人们提出了一些改进算法, 如 MAFIA[2], ENCLUS[3], CON-CLIQUE[4], CGDCP[5] 等, 但改进算法大多是针对网格的分割以及空间"剪枝"而提出的, 只有少数文章提出了针对边缘精度的处理方法[6-8].

本文提出一种针对 CLIQUE 算法的边缘检测方法 —— BDCLIQUE 算法. 该算法利用单元重心偏移的思路将密集单元边缘的点归为所偏向的密集单元, 从而能够较为精确地检测出聚类的边缘, 并且对于网格分割和密度阈值设置的不合理具有修正作用.

## 2 网格聚类

CLIQUE 算法的基本思路是将数据集划分为不相交的网格, 根据每一个单元格中点的个数定义其密度. 给定一个密度阈值 $\tau$, 若单元格的密度大于 $\tau$, 则定义该单元格是密集的; 否则, 定义该单元格是稀疏的. CLIQUE 算法的单元格聚类操作就是搜索连通密

**收稿日期:** 2010-07-26；**修回日期:** 2010-11-19.

**基金项目:** 国家自然科学基金项目(60873058)；山东师范大学研究生科研创新基金项目(BCX1005).

**作者简介:** 张鸿雁(1981–), 女, 博士生, 从事信息管理与电子商务、计算智能等研究；刘希玉(1964–), 男, 教授, 博士生导师, 从事信息管理与电子商务、计算智能等研究.

集单元的最小覆盖.

### 2.1　问题描述

高维数据集的定义: 设 $A = D_1, D_2, \cdots, D_m$ 是 $m$ 个有界定义域, 则 $m$ 维数据集可表示为 $S = D_1 \times D_2 \times \cdots \times D_m$.

设输入 $m$ 维数据集是一个点集, 数量为 $n$, 则可表示为 $V = v_1, v_2, \cdots, v_m$, 其中第 $i\,(0 < i < n)$ 个点由 $v_i = v_{i1}, v_{i2}, \cdots, v_{im}$ 表示, $v_i$ 的第 $j$ 个分量为 $v_{ij} \in D_j$.

单元格的定义: 把空间的每一维都划分为等长的区间(也可以是不等长区间), 则整个空间分成有限个相切的矩形单元, 每一个单元可表示为 $u = u_1, u_2, \cdots, u_m$, 其中 $u_i = [l_i, h_i)$, 为了区间的连续性设定左闭右开.

数据归属的定义: 如果对于一个点 $v_i$, 当且仅当对于单元 $u$ 的每一个 $u_i$ 都有 $l_i \leqslant v_i \leqslant h_i$ 成立, 则这个点 $v_i = v_{i1}, v_{i2}, \cdots, v_{im}$ 属于单元 $u = u_1, u_2, \cdots, u_m$.

密集单元的定义: 单元 $u$ 的密度为: denisty = 单元格中点的个数/数据空间中总的点个数. 对于输入的密度阈值 $\tau$, 当且仅当 denisty $(u) > \tau$, 单元 $u$ 是密集的, 设为 dense$(u)$; 否则, 单元 $u$ 是稀疏单元, 并设为 spare$(u)$.

在 $S$ 的任何子空间上, 例如子空间 sub$= D_{t1} \times D_{t2} \times \cdots \times D_{tk}, k < n$, 并且当 $i < j$ 时有 $t_i < t_j$ 成立, 该定义依然适用.

聚类的概念: 在 $k$ 维空间中由一些连通的密集单元格组成的连通分支. 两个 $k$ 维中的单元格 $u_1$,$u_2$ 称为连通的, 当且仅当:

1) 这两个单元格有一个公共的面;

2) $u_1$ 和 $u_2$ 都与另一个单元格 $u_3$ 连通.

两个单元格 $u_1 = R_{t1}, R_{t2}, \cdots, R_{tk}$, $u_2 = R'_{t1}, R'_{t2}, \cdots, R'_{tk}$ 有一个公共的面是指: 存在一个 $k-1$ 个维度, 有 $R_{tl} = R'_{tl}\,(l = 1, 2, \cdots, k)$ 成立, 并且对于第 $t_k$ 维有 $h_k = l_{tk}$ 或者 $h_{tk} = l'_{tk}$ 成立.

### 2.2　边缘检测方法

在二维子空间中, 采用 CLIQUE 算法聚类将会得到图 1 所示的结果, 而边缘部分由于密度没有达到阈值将被忽略. 一般而言, 只要分割网格的单位和密度阈值 $\tau$ 选得适中, 聚类的结果是比较令人满意的, 但如果类本身的结构不规整(如环形、曲线形的类), 或者密度分布不均匀, 将会导致聚类结果不精确. 不论哪种网格分割, 静态或者动态, 都有可能把弧形的边缘或者密度较稀疏的边缘分割出去, 从而在密度统计过程中成为稀疏单元格的一员, 不会再被聚类操作. 假设一段弧形二维数据集, 设定分割的单元格初始状态可以完全包括数据集, 由于数据恰好被包括在内, 密度较大, 可视为密集单元, 继续缩小单元格, 将会发现常见的被分割出去的边缘有两种情况, 如图 2 所示. 弧形可以进一步推广到带有角度的形状. 从不同的边缘情况, 可分析得到它们应该所属的密集单元格的位置; 第 1 类边缘所属密集单元格是与它有公共边的单元格; 第 2 类边缘所属的密集单元格是与它有公共角的单元格. 共有 3 种选择, 具体的选择规则下面会详细描述.

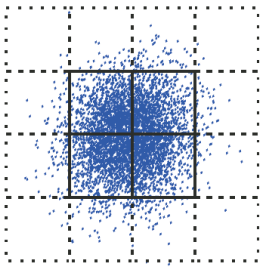
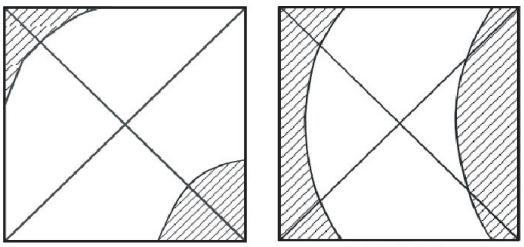


**图 1　网格聚类图**



(a)　第1种边缘情况　　(b)　第2种边缘情况

**图 2　不规则数据集边缘的两种情况**

除了公共边还有公共角, 所以可以按照对角线来划分单元格. 本文所采用的边缘检测方法是针对密集单元周围的稀疏单元进行检测, 首先把单元格 $u$ 用对角线划分为 4 个区间, 设单元格的中心为质心 $A_u$, 则质心 $A$ 由 4 个方向的力来保持平衡. 若某一区域的点较多, 且离质心较远, 则表示该方向向外的力较大, 质心偏向于该区域, 即该区域的点归并为相邻密集单元格. 如图 3 所示, 单元格 $u$ 为密度稀疏单元, 它的质心 $A_u$ 在两条对角线焦点上. 点 $b(x_b, y_b)$ 距离质心的

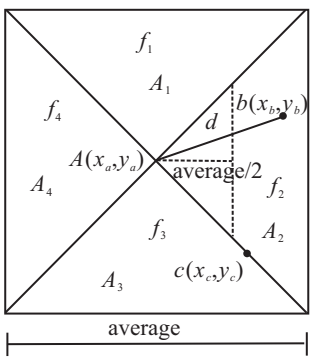



**图 3　密集单元格周围的稀疏单元格的分割方法**

距离为 $d$, 则偏向该方向的力为 $f$. 设数据点向外方向的力与离质心的距离有关, 若区域 $A_j$ 中有 $m$ 个数据点, 则可通过以下 3 个公式计算出区域 $A_j$ 总的向有公共边或公共角的密集单元的力 $F_j$:

$$d_i = (x_a - x_b)^2 + (y_a + y_b)^2,\ 1 \leqslant i \leqslant m; \quad (1)$$

$$f_i = \begin{cases} \left(\dfrac{2}{\text{average}} d_i\right)^2,\ 0 \leqslant d_i < \dfrac{\text{average}}{2}; \\ (d_i)^3 + 1 - \left(\dfrac{\text{average}}{2}\right)^3, \\ \qquad \dfrac{\text{average}}{2} \leqslant d_i \leqslant \text{average}; \end{cases} \quad (2)$$

$$F_j = \sum_{i=1}^{m} \frac{1}{f_i}. \quad (3)$$

其中: $f_i$是每个点对质心向外的力, average 是每个单元格的边长, 当距离质心 $d_i$ 小于 average/2 时, 数据点靠近质心, 向外的力就越小; 数据点距离质心越远时, 向外的力越大, 因此越偏向于有公共边或公共角的密集单元格. 这样, 可以保证距离较远的个别点对全局的影响不会太大, 否则只有极少数的质心周围的数据点被归并到密集单元格中. 设定 4 个区域分别为 ($A_1$,$A_2$,$A_3$,$A_4$), 点 $b$ 在该单元格内满足

$$\begin{cases} x_a - \text{average} \leqslant x_b < x_a + \text{average}, \\ y_a - \text{average} \leqslant y_b < y_a + \text{average}. \end{cases} \quad (4)$$

判断点 $b(x_b,y_b)$ 所在的区域由下式获得:

$$\begin{cases} \alpha = y_b + x_b - (x_a + y_a), \\ \beta = y_b + x_b. \end{cases} \quad (5)$$

当 $\alpha > 0$, $\beta > 0$时, $b \in A_1$; 当 $\alpha > 0$, $\beta < 0$ 时, $b \in A_2$; 当 $\alpha < 0$, $\beta < 0$ 时, $b \in A_3$; 当 $\alpha < 0$, $\beta > 0$ 时, $b \in A_4$.

点 $c$ 在对角线上, 有可能属于 3 个与该单元格有公共角的任何一个密集单元格, 则对于稀疏单元格中点的所属密集单元有如下规则:

1) 若对角线两边的力相差较远, 力较大的区域有公共边的单元格是密集的, 则将其归为该密集单元格; 否则为孤立点.

2) 若对角线两边的力相近, 且公共对角线方向的相邻单元格是密集的, 则两方都归为对角线所对应的单元格; 若公共对角线方向的相邻单元格是稀疏的, 则为孤立点.

3) 对角线上的点分为两种情况: ① 若对角线两边的区域力相近, 则可归为任何一方区域; ② 若对角线两边区域力相差较远, 则归为力较大的一方.

设每个区域的力为 $F_j(j = 1,2,3,4)$, 相邻区域的相似度为 $D = (A_i - A_j)/A_i(i,j = 1,2,3,4)$, 则按照上述规则有:

1) 若 $D \to \varepsilon$($\varepsilon$ 为一个趋于 0 的实数), 即两区域的力相近, 且两区域公共对角线所对应的单元格为密集单元格, 则两区域都归为该密集单元格.

2) 若 $D \to N$($N$ 为一个趋于 1 或者 $-1$ 的实数), 即两区域的力相差太远, 且与力大的区域有公共边的单元格是密集的, 则力大的区域归并为该密集单元格.

### 2.3 BDCLIQUE 算法

对于给定的 $n$ 维数据集, 首先划分为网格; 然后按照 CLIQUE 算法进行聚类; 再通过边缘检测, 得到较为精确、完整的聚类结果. 具体算法如下:

Step 1: 将高维数据空间进行划分. 将数据空间划分为互不相交的矩形单元, 并统计每个单元格中的点数为该单元格的密度 density($u$).

Step 2: 子空间的搜索. CLIQUE 算法符合 Aprior 算法的先验性质, 所以可以采用自底向上的方法进行子空间的搜索.

先验性质是: 如果一个 $k$ 维单元是密集的, 则它在 $k-1$ 空间上的投影也是密集的. 即给定一个 $k$ 维的候选密集单元, 如果检查它的 $k-1$ 维投影空间, 发现任何一个不是密集的, 则可知道第 $k$ 维的单元也不可能是密集的.

Step 3: 识别簇. 在候选子空间中找出相通密集单元的最大集合.

Step 4: 边缘检测. 扫描密集单元周围的稀疏单元格, 按照2.2节给出的规则进行边缘检测, 最终得到较为精确的聚类结果.

Step 5: 生成一个聚类的描述. 为每个簇生成最小化的描述, 对每个簇确定覆盖相连的密集单元的最大区域, 然后确定最小的覆盖.

由于前 3 步都是普通的基于密度的网格聚类算法, 在此只给出 Step 5 边缘检测的具体过程:

for dense($u_{ij}$); //扫描未被检查的密集单元格, $i$ 和 $j$ 为单元格的坐标号

if spare($u_{ij}$) //如果是稀疏单元格且未被检查

//求质心 $a$ ($x_a$,$y_a$)

$x_a = \text{average} \times i - \dfrac{\text{average}}{2}$;

$y_a = \text{average} \times j - \dfrac{\text{average}}{2}$;

for 扫描 spare($u_{ij}$) 中的点

统计 4 个区域的力 ($F_1$,$F_2$,$F_3$,$F_4$);

$D = F_i - F_j$; //判断相邻两个区的相近度 $D$,

if ($D \to \varepsilon$ 其公共对角线所对应的是 dense($u$)),

then 两个区域都归并到该单元格上;

if ($D \to N$ and 与力大的区域有公共边的单元格是 dense($u$)),

then 力大的区域归并到该单元格上.

## 3 实验结果

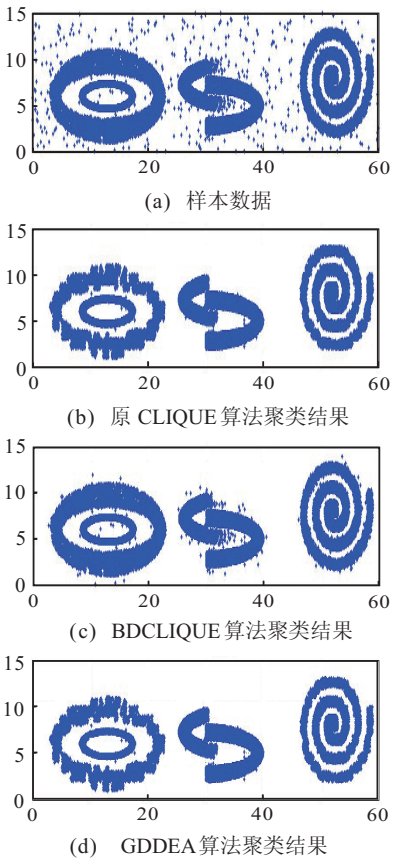
通过实验发现加入边缘检测有以下优点:

1) 精确聚类边缘. 网格聚类的特点导致了聚类边缘由于网格的划分而不平滑, 过于垂直或水平. 通过边缘检测, 可以把不规则的边缘加入到聚类中, 使得聚类结果更加精确.

2) 对于阈值和网格参数不合适的设置, 可通过边缘检测得到修正. 如果阈值取的较大, 则会缺失部分密度相近的单元格, 但由数据集聚类特征, 可以从密集单元格的周围找到, 因此通过边缘检测可以修正由于阈值设置较大而造成的聚类精确误差. 如果网格间隔设置不合理, 例如为了增加聚类的准确性, 设置较小的网格间隔, 会导致不规则边缘被分割到相邻的单元格, 因此通过边缘检测可以找到在稀疏单元格中的边缘部分, 增强聚类的伸缩性.

### 3.1 实验对比

图 4 为采用 CLIQUE 聚类加入边缘检测前后的对比. 样本数据包含 3 种常见的测试数据集, 共有 40 043 个样本数据点, 其中双环数据集约 20 000 点, 双半圆数据点约 10 000 点, 螺旋线数据点约 10 000 点, 噪声约 1 000 点. 数据范围 $x$ 在 [0,60] 区间, $y$ 在 [0,15] 区间, 利用 40×17 的网格进行分割, 阈值设为 30, 即每个网格如果点的密度大于 30 则为密集单元格.




(a) 样本数据

(b) 原 CLIQUE 算法聚类结果

(c) BDCLIQUE 算法聚类结果

(d) GDDEA 算法聚类结果

**图 4　原 CLIQUE 聚类与改进后的算法聚类结果对比**

从图 4(b) 可以明显看出, 当设置密度阈值不合适时, 相近密度的单元格将会缺失. 图 4(c) 为 BDCLIQUE 算法, 缺失的单元格基本都被找回, 增强了聚类的精确性. 由于网格的切割, 导致弧形边缘被分割到稀疏的单元格中, 使得没有加入边缘检测的聚类结果边缘是水平或者垂直的, 当加入边缘检测后弧形的边缘都被检测出来, 因而边缘检测修正了网格间隔设置不合理的误差. 图 4(a) 是原数据, 图 4(b) 是没有加入边缘检测的聚类结果. 图 4(d) 是文献 [8] 中的边缘精度算法, 可以看出该算法只提高了簇之间的聚类效果, 但仍旧只有密集单元格的聚类, 由网格分割而引起的边缘缺失并没有找回.

### 3.2 复杂度分析

通过算法可以看出, 因为基本的聚类算法没有改变, 所以影响整个聚类的算法复杂度都在边缘检测上. 影响边缘检测的复杂度有以下两个因素:

1) 密集单元格的数量. 在边缘检测时, 为了寻找边缘, 只扫描一遍密集单元格, 所以密集单元格的数量会直接影响到边缘检测的算法.

2) 边缘稀疏单元格中的数据点集的数量. 因为需要计算在边缘稀疏单元格中 4 个区域的力, 所以需要扫描单元格中所有点. 边缘稀疏单元格中的点越多, 计算花费就越大. 当然, 检测的边缘也越准确.

由此可见, 影响边缘检测的复杂度的最大因素是聚类数据集的边缘范围, 如果数据集聚类不紧凑, 比较分散, 将会导致聚类的边缘较大, 涉及的单元格较多, 从而大大影响边缘检测的速度. 设密集单元格有 $k$ 个, 则边缘检测的复杂度为 $O(k)$. 本文采用螺旋线形数据讨论边缘单元格与运行时间的关系. 检测程序在 Pentium(R)4, CPU2.8, 1 GB 内存的 Windows XP 系统运行, 所生成的螺旋线形数据集半径为 10, 整个背景的范围是 $[-40, 40]$, 用 120×120 的网格进行分割, 每次通过边缘检测后的结果是完全的, 所有数据集点都被聚类. 实验数据见表 1.

**表 1　BDCLIQUE 算法复杂度实验数据**

| 数据集数目 | 密集单元格数 | 检测到边缘单元格数 | 边缘检测运行时间/ns |
|---|---|---|---|
| 10 000 | 26 | 19 | 17 620 984 |
| 20 000 | 105 | 40 | 32 682 349 |
| 30 000 | 223 | 77 | 46 562 129 |
| 40 000 | 345 | 150 | 61 657 596 |
| 50 000 | 498 | 257 | 74 972 192 |
| 60 000 | 829 | 602 | 105 322 280 |
| 70 000 | 2 316 | 632 | 146 253 998 |
| 80 000 | 3 137 | 1 077 | 175 540 312 |
| 100 000 | 4 531 | 1 977 | 221 257 896 |
| 150 000 | 6 049 | 3 258 | 267 508 157 |
| 200 000 | 6 828 | 3 584 | 294 506 031 |

从图5中可以看出, 密集单元格和边缘单元格的数量与边缘检测时间基本上是线性关系, 但边缘单元格所影响的检测时间比密集单元格快, 因此不论数据集的数据量大小, 如果数据集较为分散, 所涉及到的范围大, 则边缘检测的时间将会增加. 但一般情况下, 数据集聚类较为紧密, 对边缘检测的时间不会有太大的影响. 从表1中的运行时间也可以看出, 在万级数据集的聚类中, 而且是螺旋形数据, 边缘较多, 边缘检测运行的时间是在毫秒级, 因此对于一般CLIQUE算法的聚类, 增加边缘检测将不会在时间上有太大影响.

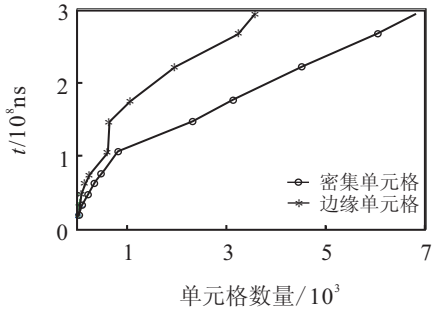



**图5 BDCLIQUE算法密集单元格、边缘单元格与运行时间的关系图**

## 4 结 论

网格聚类是大型空间数据聚类的主要方法之一, 具有效率高、聚类结果与输入数据顺序无关、可扩展性好的优点. 但由于其特殊的聚类方法, 也具有聚类边缘太规整以及参数设置要求高的缺点. 本文提出一种针对CLIQUE算法的边缘检测方法(BDCLIQUE), 利用单元重心偏移的思路把密集单元边缘的点归为所偏向的密集单元, 从而较为精确地检测出聚类的边缘, 并且对于网格分割和密度阈值设置不合理的情况具有修正作用. 该方法既可以检测到应该归并到类当中的边缘, 也可以分辨出孤立点, 是一种对网格聚类具有有效改进的方法.

### 参考文献(References)

[1] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, et al. Automatic subspace clustering of high dimensional data for data mining applications[C]. Proc of ACM SIGMOD Int Conf on Management of Data. New York, 1998: 94-105.

[2] Goil Sanjay, Harasha Nagesh, Alok Choudhary. MAFIA: Effrcient and scalable subspace clustering for very large data sets[R]. Sheridan Road: Center for Parallel and Distributed Computing, Northwestern University, 1999.

[3] Cheng C, Fu A, Zhang Y. Entropy-based subspace clustering for mining numerical data[C]. Proc of the 5th ACM SIGKDD. San Diego, 1999: 84-93.

[4] Hinncburg A, Keim D. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering[C]. Proc of the 25th Very Large Databases Conf. Edinburgh, Scotland, 1999: 506-517.

[5] 冯兴杰, 黄亚楼. 带约束条件的聚类算法研究[J]. 计算机工程与应用, 2005, 41(7): 12-15.
(Feng X J, Huang Y L. Research on the algorithm of the constrained clusteirng[J]. Computer Engineering and Applications, 2005, 41(7): 12-15.)

[6] 余灿玲, 王丽珍, 张元武. 基于网格密度方向的聚类簇边缘精度加强算法[J]. 计算机研究与发展, 2010, 47(5): 815-823.
(Yu C L, Wang L Z, Zhang Y W. An enhancement algorihtm of cluster boundaries precision based on grid's density direction[J]. J of Computer Research and Development, 2010, 47(5): 815-823.)

[7] 邱保志, 沈钧毅. 网格聚类中的边界处理技术[J]. 模式识别与人工智能, 2006, 19(2): 277-280.
(Qiu B Z, Shen J Y. Border-processing technique in grid-Based clustering[J]. PR and AI, 2006, 19(2): 277-280.)

[8] 王生生, 刘大有, 曹斌, 等.一种高维空间数据的子空间聚类算法[J]. 计算机应用, 2005, 25(11): 2615-2617.
(Wang S S, Liu D Y, Cao B, et al. A subspace clustering algorithm for high dinensional spatial data[J]. Computer Applications, 2005, 32(3): 216-218.)