带障碍的空间分级聚类算法*)

周丽华　王丽珍　陈克平

(云南大学信息学院计算机科学与工程系　昆明 650091)

摘　要　带障碍的聚类问题是一个具有实际应用价值的问题，因为现实世界中确实存在河流、山脉等之类的物理障碍，它们的存在会影响聚类结果的合理性。传统的聚类算法在进行空间数据的聚类时，往往忽略了障碍对于聚类结果的影响。本文讨论了不同障碍对数据点间连通性的不同影响，提出了带障碍的分级聚类算法 OBHIEC。分级聚类方法使得需要计算障碍距离的点对数目减少，并能处理数据分布密度不同的情况。实验结果表明，OBHIEC 算法能有效完成带障碍的聚类，并具有较好的增量特性。

关键词　数据挖掘，障碍，聚类，分级

Spatial Hierarchical Clustering in the Presence of Obstacle

ZHOU Li-Hua　WANG Li-Zhen　CHEN Ke-Ping

(Department of Computer Science and Engineering, School of Information, Yunnan University, Kunming 650091)

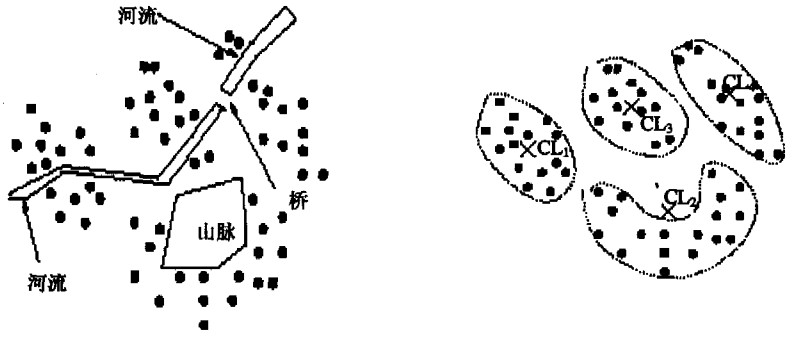
Abstract　The problem of spatial clustering in the presence of obstacles has many practical applications. Many traditional clustering algorithms are performed without the presence of obstacles that exist in the real world, such as rivers, lakes and hills, but their presence may affect the result of clustering substantially. In this paper, a hierarchical clustering algorithm, called OBHIEC, is proposed, which can reduce the calculation of obstructed distance and is suitable for data set with varied distributing density. The experiment results show that OBHIEC is both effective and efficient.

Keywords　Data mining, Obstacle, Clustering, Hierarchical

1　引言

聚类是根据数据对象在特征空间的距离、相连性、密度等特性将相似的数据点聚集成簇，从而发现数据集中数据的分布模式，因此聚类分析在模式识别、空间数据分析、图像处理、地理信息系统、生物学、医学图像分析等领域有广泛的应用。多年来，聚类分析的研究一直非常活跃，至今已提出了许多聚类算法。可以将这些方法大致分类为：划分方法、层次方法、基于密度的方法、基于格的方法和基于模型的方法[3~7]。这些方法在大规模数据集上均有较好的有效性和效率，但它们不能直接用于带障碍的聚类。文[1]指出：在现实世界中，存在许多河流、山脉之类的物理障碍，这些障碍的存在会极大地影响聚类结果。因为目前的聚类算法在聚类时常常忽略数据点之间的障碍，认为任意两个数据对象之间都是直接可达的，所以在计算两个数据点之间的距离时，常常是计算两点之间的直接欧氏距离。实际上，当两个数据点之间存在障碍时，它们之间的距离往往比直接欧氏距离大得多，有时甚至是无穷大(两个数据点不可达)。因此，在聚类过程中，若不考虑实际物理障碍，得到的聚类结果往往不切合实际或没有什么实际意义。文[1]给出了如下实例作为说明。

一家银行希望在图 1(a)所示的区域设置 4 台 ATM，为用点表示的消费者服务，区域内存在河流和山脉物理障碍。若在聚类过程中不考虑这些障碍，则聚类结果如图 1(b)所示。很显然，该聚类结果不切合实际，比如簇 CL1 被河流分割成两部分，分配给该簇的 ATM 只能设置在河的一岸，这样河对岸的消费者只能走很远的路才能得到服务，因此区域内的障碍不容忽视。



(a) 消费者的位置和障碍　　(b) 忽略障碍时形成的聚类簇

图 1

*)云南省教育厅科学研究基金项目(03Y173D)、国家自然科学基金项目(60463004)。**周丽华**　副教授，主要研究方向：数据挖掘与数据仓库；**王丽珍**　教授，主要研究方向：数据挖掘与数据仓库；**陈克平**　硕士研究生，主要研究方向：数据挖掘与数据仓库。

文[1]给出了带障碍的聚类问题的具体定义。

定义 1.1 给定(1)包含 n 个数据点的数据集 $P=\{p_1,p_2,\cdots,p_n\}$

(2)二维区域 R 中 m 个不相交的障碍的集合 $O=\{O_1,O_2,\cdots,O_m\}$。每个障碍 O_i 由一个简单多边形表示。不考虑障碍时，p_k 和 p_j 两点之间的直接欧氏距离记为 $d(p_k,p_j)$；考虑障碍存在时，两点之间的障碍距离是指不被任何障碍切断的从 p_k 到 p_j 的最短路径，记为 $d'(p_k,p_j)$。带障碍距离的聚类问题是将 P 划分为使各点距离各个簇中心最近的 k 个簇 $CL_1,CL_2,\cdots,CL_k$。

各点距离各个簇中心最近等价于使 $E=\sum_{i=1}^{k}\sum_{p\in CL_i}(d'(p,c_i))^2$ 最小，式中 c_i 是簇 CL_i 的中心。

为了解决带障碍的聚类问题，文[1]基于划分的方法提出了 COD-CLARANS 算法，文[2]基于网格和密度思想提出了 DCellO 算法。这两个算法在聚类过程中都考虑了障碍的存在，且在整个数据集上聚类算法执行一次。两个算法在聚类过程中都需要计算大量的障碍距离，从而影响了算法效率。

物各有性。不同的障碍有不同的形状、特性，比如河流呈长条形，在不考虑桥梁、船舶等介质的情况下，障碍两边的数据对象是互不可达的，它们不可能聚集成一个簇，它们之间的距离根本不用计算。因此，充分利用障碍的特性可以减少聚类过程中的计算量，提高算法效率。本文在考虑障碍存在的情况下，充分利用数据对象的整体信息、局部信息及障碍的特性对数据对象进行分级聚类。文章第 2 节为 OBHIEC 算法思想，第 3 节为 OBHIEC 算法描述及其分析，第 4 节是实验验证，最后是总结。

2 OBHIEC 算法思想

本文所提算法(OBHIEC)主要分 3 步：首先以障碍为主体，根据障碍的形状和特性将整个区域划分成多个子区域，在每个子区域内不包含障碍，任意两个数据对象之间是直接可达的，因此数据对象之间的距离用直接欧氏距离计算；然后在各个子区域内，使用不考虑障碍的聚类算法，对子区域内的数据对象进行一级聚类，聚类后每个簇用其中心来代表，所有簇的代表点构成一个集合；最后在代表点集合上应用带障碍的聚类算法对代表点进行二级聚类，聚集为同一个簇的所有代表点所代表的数据对象同属一个聚类。代表点集合中代表点之间可能直接可达，也可能被障碍分隔。在计算障碍距离时，桥梁、船舶等因素应考虑在内。

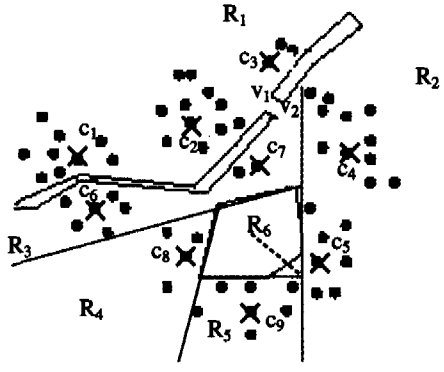


图 2 区域的划分及各簇代表

例如，根据障碍可以将图 1(a)所示区域划分成如图 2 所示的 $R_1,R_2,\cdots,R_6$ 等 6 个子区域，每个子区域内都不包含障碍；然后分别对 $R_1\sim R_6$ 等 6 子区域内的数据对象进行聚类，聚类后得代表点集合 $C=\{c_1,c_2,\cdots,c_9\}$；最后再对 C 进行带障碍的聚类。计算 c_i 与 c_j 之间的距离时，若 c_i 与 c_j 之间无障碍，则它们之间的距离为 $d(c_i,c_j)$，否则为 $d'(c_i,c_j)$。由于河上有桥，因此 c_1,c_2,c_3 与 c_4 等点之间是可达的，它们之间的距离可以计算出来。比如 $d'(c_3,c_4)=d(c_3,v_1)+d(v_1,v_2)+d(v_2,c_4)$，其中 v_1,v_2 是表示河流的多边形的两个顶点。

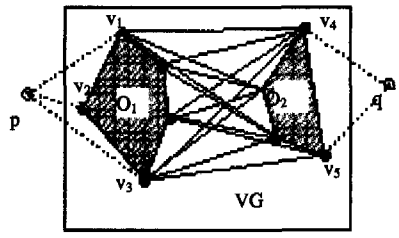


图 3 障碍距离

当然，在区域划分的过程中，可能会将本应是同一簇的数据对象划分到不同的子区域，在不同子区域参与聚类，形成了不同的簇，如图 2 中 c_5,c_9 所代表的簇。但在带障碍的聚类过程中，这些本应属于同一簇却被聚成不同簇的代表点之间没有障碍分隔，它们之间的距离显然最小，因此它们又会被归并到同一簇中。

该方法有如下几个特点：

(1)适用于大规模数据集的聚类。因为第一级聚类是在各子区域进行，子区域中数据对象的数目总是小于数据对象的总数；而第二级带障碍的聚类虽然在整个区域进行，但由于子区域中每个簇都用代表点表示，聚类是针对这些代表点进行的，其数量已大大减小，因此不会出现由于数据量大而算法无法运行的情况。

(2)需计算障碍距离的点对数量减少。p,q 两点之间的障碍距离是指不被任何障碍切断的从 p 到 q 的最短路径。如图 3 所示，p,q 两点之间的障碍距离是始于 p 到 v_1,v_2 或 v_3 之间的一条边，穿越 VG 中某条路径，止于 v_4 或 v_5 到 q 之间的一条边的最短路径。可见，障碍距离的计算比直接欧氏距离的计算复杂得多。在本文中，由于第一级聚类在各子区域上进行，子区域内不包含障碍，因此不涉及障碍距离的计算。第二级聚类虽然涉及障碍距离，但由于聚类对象是各个簇的代表点，其数目远远小于原始数据点的数目，因此需计算障碍距离的点对也减少。

(3)适用于数据分布密度不同的聚类。数据对象在整个区域内的分布密度不一定一致，比如一条护城河是一个物理障碍，河的一边是市区，另一边是郊区。市区经济发达，人口密度高，而郊区人口密度低，如果用同样的标准对市区和郊区的人口数据进行统一聚类，结果不尽如人意。如果将市区和郊区以不同的标准分开处理，聚类结果将更符合常理。

(4)适宜增量处理。

①障碍不变，增加了数据对象。障碍不变，则区域的划分不改变。若某一子区域的数据对象发生剧烈变化，引起该区域簇的改变，如河流东岸原经济较落后，现在政府调整政策，要开发河流东岸，造成河流东岸涌入大量人口，城市面积扩大。这种情况的聚类只需对河流东岸子区域的数据重新进行第一级无障碍聚类，然后进行第二级的全局带障碍聚类即可，其它子区域(如河流西岸)的局部聚类不受影响。

②新增障碍。若某区域内出现了新的障碍，比如河流东岸新建了一条高速公路，则只需将涉及这个障碍的区域再进行划分、聚类，不影响与此障碍完全无关的区域。

③跨越障碍。若原来不可逾越的障碍变为可逾越，如河流上架桥、山脉凿隧洞等，则只需在带障碍聚类时重新计算簇代表点间的距离，子区域的聚类不受影响。

(5)本文的聚类是一个两级聚类，经历了整体-局部-整体的过程。在整个聚类过程中充分利用了数据对象的整体信息和局部信息，这一过程与人的直观聚类过程极为相似。分析一下人的直观聚类过程可以发现，人主要是根据数据对象整体和局部分布情况所形成的反差对比来进行聚类判断的。人所依据的基本规则是：同一聚类中对象应相距较近，而各聚类间应相距较远。至少从对象整体分布情况看，其聚类结果应该符合这一规则要求。

3 OBHIEC 算法描述

如前所述，OBHIEC 算法分 3 个步骤：(1)划分区域；(2)子区域聚类；(3)全区域聚类。下面将对它们所涉及的算法进行描述及分析。

3.1 划分区域

障碍的存在影响了区域的连通性，不同的障碍对区域连通性的影响是不同的。

定义 3.1 障碍划分数：若根据障碍物 O_i 可以将区域划分成 r_i 个子区域，各个子区域的连通性不再受 O_i 的影响，则称障碍物 O_i 的划分数 $\mathrm{DIVM}(O_i)=r_i$。

河流、高速公路之类的障碍呈长条形，在不考虑桥梁、船舶等因素的情况下，障碍两边的数据对象不可达。这类障碍将区域划分为两个子区域，它们的划分数为 2。空间数据库中常常存有数据对象相对于障碍的方向关系，如 B west O_j，A east O_j 等。若 O_j 代表一条河流，则 B west O_j 表示数据对象 B 在河流的西岸，A east O_j 代表数据对象 A 在河流的东岸。利用数据库中的这些空间方向关系，可以很容易地将障碍两边的数据对象分开，形成 P_1、P_2 两个集合，其中 $P_1=\{A \mid \forall$ 满足 A east $O_j, A\in P\}$，$P_2=\{B \mid \forall$ 满足 B west $O_j, B\in P\}$。

山脉、湖泊之类的障碍，四周都可能有数据对象，绕道后，这些数据对象之间都是可达的。为了保证每个子区域内任意两个数据对象之间不被障碍分隔，划分区域时可以根据代表障碍的多边形的边进行。多边形的每条边对应一条直线方程 $g_i(x)=0$，每条直线将区域一分为二。对于点 p，若满足 $g_i(p)>0$，则表明点 p 在直线 $g_i(p)=0$ 的正侧；若满足 $g_i(p)<0$，则表明点 p 在直线 $g_i(p)=0$ 的负侧；$g_i(p)=0$ 表明点 p 在直线上。本文规定：数据对象位于障碍的正侧。有 r 条边的多边形对应 r 条直线，可以将区域划分为 r 个子区域。因此，由有 r 条边的多边形表示的障碍的划分数为 r。

障碍物 O_i 对区域的划分只是使各个子区域的连通性不再受 O_i 的影响。若某个子区域内还包含障碍 O_j，则根据 O_j 再对该子区域进行划分。

区域划分算法 DivArea 描述如下：

```
输入：包含 n 个数据点的集合 P={p1,p2,…,pn}
m 个不相交的障碍集合 O={o1,o2,…,om}
输出：区域划分后，各子区域内的数据集合 D1,D2,…
  1)Rsub=P,Rremain=P,t=1 //Rsub是当前需要进行划分的区域，
    Rremain是需要进行划分的区域集合
  2)Rtemp=Division(Rsub) // Rtemp是Rsub被划分后的子区域集合
  3)若 Rtemp=Φ,则 Dt=Rsub,t=t+1
  4)Rremain=Rremain∪Rtemp−Rsub
  5)若 Rremain≠Φ,则 Rsub=Rremain中任一元素,转 2)
  6)输出 Dt
Division(Rsub) //区域划分函数
{1)Otemp=Rsub中数据对象所占据的区域中包含的障碍；
 2)如果 Otemp=Φ,则 Rtemp=Φ；
   否则{Ott=Otemp中具有最小障碍划分数的障碍；
        Ott将 Rsub划分为 w 个子区域的集合 Rtemp={S1,S2,…,
        Sw}
   }
 3)Return(Rtemp)
}
Ott对 Rsub的划分：
  1)如果Ott的障碍划分数 DIVM(Ott)=2
        则 S1={p| ∀p east Ott,p∈Rsub},S2={p| ∀p west Ott,
        p∈Rsub}
  2)如果DIVM(Ott)>2
        for (i=1;i<r;i++) //r为Ott的多边形边的边的数目
            {对于 Rsub中的每一个数据对象 p,如果 gi(p)>0,则
            将 p 从 Rsub中移到 Si 中；}
        //gi(x)=0 是 Ott的第 i 条边所对应的直线方程
```

区域划分算法 DivArea 需要扫描一遍数据库，对所有数据点的划分需要进行 $e=\sum_{i=1}^{m}\mathrm{DIVM}(O_i)-m$ 次判定，最坏情况下复杂度为 $O(e\times n)$。当障碍数目不太多，障碍形状不是太复杂时，e 较小，此时 DivArea 的复杂度只与数据点数目 n 有关。

3.2 子区域聚类

进行区域划分之后，每个子区域内任意两个数据对象之间都不被障碍分隔，对子区域内的数据对象聚类可以不考虑数据对象之间的障碍距离。

CLARANS[7]算法是基于随机搜索的不考虑障碍的聚类方法，具有很好的性能。COD-CLARANS[1]算法就是基于 CLARANS 的思想发展起来的。但 CLARANS 算法不适宜本文的子区域聚类，因为 CLARANS 算法要求输入聚类簇的数目，而本文中每个子区域内究竟包含多少个簇在聚类前是未知的，有时一个子区域内可能只包含一个簇，甚至根本没有簇。文[2]中提出的 DCellO 算法是将基于密度与基于网格的聚类算法相结合，并利用图形学中邻接网格的概念而提出的。算法首先划分网格，然后以一个随机抽取的核心网格为基础，不断地寻找邻接网格，向四周扩散；然后以寻找到的包含对象的网格作为核心网格，继续向四周弥散。该算法能够进行任意形状的带障碍的聚类，并且不需要指定聚类簇的数目。但该算法有两个缺点：(1)待计算的网格集合中，每一个包含对象的网格虽然已经指定了簇标号，但都要作为核心网格进行处理，造成大量的重复计算；(2)在整个区域内，网格的划分及聚类半径是固定的，这在数据对象的分布密度不同时会造成聚类结果不合理。本文基于 DCellO 算法的思想提出了不考虑障碍的网格弥散聚类算法 PCell。

定义 3.2 网格的连通与间断：设当前正在处理的网格为 Cell_i，若 Cell_i 的 8 个邻接网格中有网格包含对象，则称这些包含对象的网格与 Cell_i 相连通。若 Cell_i 的 8 个邻接网格中没有网格包含对象，则称 Cell_i 发生间断。

定义 3.3 扩散宽度 r_{maxdist}：网格在扩散过程中发生间断时允许间断的最大网格数。

定义 3.4 扩展范围：与 Cell_i 的距离为 r_{maxdist} 的网格区域称为 Cell_i 扩散范围。

定义 3.5 候选网格：Cell_i 扩散范围内包含数据对象的网格称为 Cell_i 的候选网格。

如图 4 所示，扩散宽度 $r_{\mathrm{maxdist}}=2$，网格 A、B、C 都是 Cell_i 的候选网格。网格 A 与 Cell_i 相连通，网格 B、C 与 Cell_i 不连通。

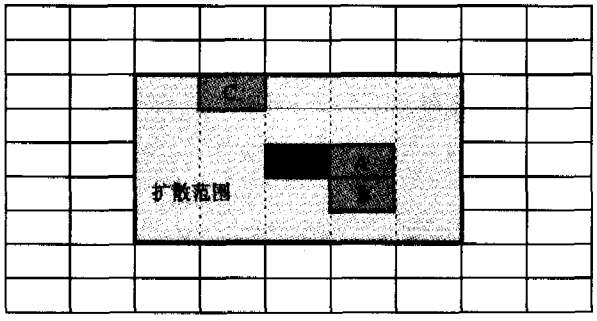


图 4　扩散宽度、扩散范围、候选网格

使用扩散范围的目的是为了避免两个相距很近的网格由于不连通而聚类到不同簇的情况发生。扩散宽度 $r_{maxdist}$ 是一个容易定义的参数，数据点分布密集时，$r_{maxdist}$ 可以较小；反之，$r_{maxdist}$ 可以较大。

PCell 算法描述如下：

输入：扩散宽度 $r_{maxdist}$、第 i 个子区域的数据集合 D_i
输出：第 i 个子区域内各个聚类簇的代表点 c_{i1}、c_{i2}、…
1)划分网格；
2)将数据对象映射到网格中，所有网格中包含对象的集合记为 S，聚类数 $j=1$；
3)第 j 个簇的网格集合 $Cluster_j=\Phi$；
4)从 S 中随机选取一个网格为 $Cell_{current}$；
5)$Cell_{candidate}=Cell_{current}$；
6)以 $Cell_{current}$ 为起始网格，在扩展范围内寻找其未计算过的候选网格，所有候选网格的集合记为 $Cell_{temp}$；
7)$Cell_{candidate}=Cell_{candidate}\cup Cell_{temp}-Cell_{current}$；
8)$Cluster_j=Cluster_j\cup Cell_{current}$；
9)若 $Cell_{candidate}$ 非空，则从中任选一个网格作为 $Cell_{current}$，转 6)；
10)否则 $j=j+1$，$S=S-Cluster_j$；
11)若 S 非空，则转 3)；//第 $j+1$ 个簇开始弥散
12)属于同一个簇的所有网格内的数据对象属于一个簇，计算各个簇的中心 c_{i1}，c_{i2}，…，然后输出。

PCell 算法处理的是不带障碍的聚类，因此不需要计算网格间的障碍距离，并且算法对不包含对象的网格不处理，每个包含对象的网格只处理一次。若所有网格的数目为 d，扩散范围内的网格数目为 w，则在最坏情况下 PCell 算法的时间复杂度为 $O(d\times w)$。通常 w 很小，因此时间复杂度只与网格数有关。同时，由于各个子区域单独调用 PCell 算法，不同区域可以使用不同的网格划分和不同的扩散宽度，适宜数据对象在整个区域内分布密度不一致的情况。

3.3　全区域聚类

全区域聚类是二级聚类，是针对一级聚类的各个簇的代表点进行。代表点之间可能直接可达，也可能被障碍分隔。该级聚类可以使用 COD-CLARANS 算法实现，也可以使用 DCellO 算法实现。由于代表点的数目较少，障碍距离的计算量较小。本文二级聚类使用 DCellO 算法实现。

4　实验分析

本节用实验验证 OBHIEC 算法的有效性和效率。

为了与 DCellO 算法进行比较，本文在相同的数据集上，进行 DCellO 算法与 OBHIEC 算法的比较，并且使用与文[2]相同的实验环境。实验环境为 Windows2000，Server sp4，256MB 内存，PIV 1.8GHz。开发语言为 Visual C^{++} 6.0 sp5。

本文采用一个字母 E 和一组并排的竖线作为障碍，在图上有三个呈正态分布的数据簇，用于比较 DCellO 算法与 OBHIEC 算法的聚类质量。DCellO 算法与 OBHIEC 算法的聚类结果如图 5、图 6 所示。图中的黑色线条表示障碍，位于相同聚类中的点用相同的数字或字母代表。DCellO 算法共将数据聚成了 7 个聚类，分别用 0～6 代表。而 OBHIEC 算法将数据分成了 7 个聚类，分别用 1～7 代表。由图可见，障碍对聚类结果的影响是非常显著的，但 OBHIEC 算法和 DCellO 算法都能进行有效的带障碍的聚类。

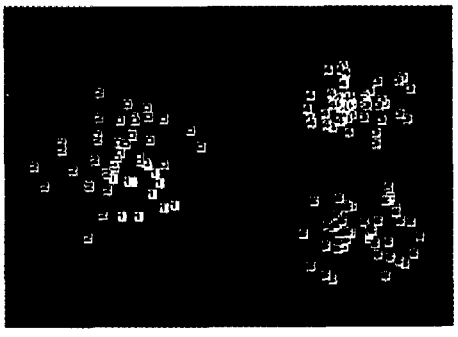


图 5　DCellO 算法的聚类结果

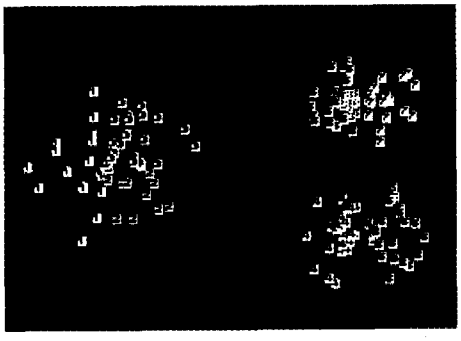


图 6　OBHIEC 算法的聚类结果

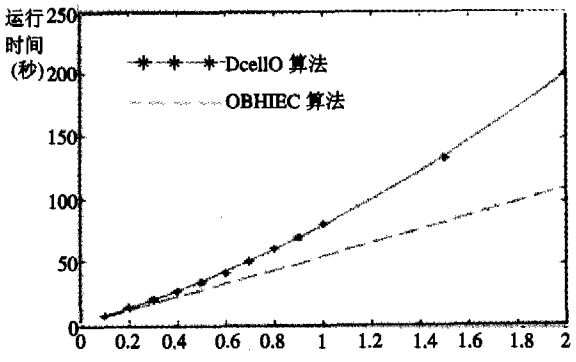


图 7　对象数量对执行时间的影响

图 7 是 DCellO 算法和 OBHIEC 算法的运行时间随数据点数目的变化情况。

图 7 表明，当聚类半径固定时，DCellO 算法与 OBHIEC 算法的执行时间都与对象数量基本呈正线性相关。在大数据量的时候，OBHIEC 算法的效率要明显好于 DCellO 算法，适用于大规模数据集。

结论　在聚类过程中考虑障碍与不考虑障碍往往会得到不同的聚类结果，原因是障碍的存在会影响空间数据对象之间的连通性，使得数据点之间的距离不再是单纯的直接欧氏距离。由于障碍距离的计算比直接欧氏距离的计算复杂得多，因此障碍距离的计算成为影响算法效率的关键。本文讨

（下转第 204 页）

(9):917～922
4 Liu H, Motoda H. Feature selection for knowledge discovery and data mining [M]. Boston:Kluwer Acadcemic Press, 1998
5 苗多谦,王钰. 粗糙集理论中概念与运算的信息表示[J]. 软件学报, 1999, 10(2):113～116
6 王国胤,于洪,杨大春. 基于条件信息熵的决策表约简[J]. 计算机学报, 2002, 25(7):759～766
7 Aha D, Bankert R. A comparative evaluation of sequential feature selection algorithms [A]. In:Fisher D, Lenz H,eds. 5th International Workshop on Artificial Intelligence and Statistics [C]. New York: Springer, 1991
8 Cover T M. Elements of Information Theory [M]. New York: Wiley, 1991
9 苗夺谦,胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展, 1999, 36(6):681～684
10 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社,2001

表6 BEMMIMAMI 算法搜索空间大小

Dataset	All	BEMMIMAMI(M=2p)		BEMMIMAMI(M=p)		BEMMIMAMI(M=1)	
		Ev	*Ratio*	*Ev*	*Ratio*	*Ev*	*Ratio*
Corral	2^6	8	0.125	8	0.125	8	0.125
Monk1	2^6	8	0.125	8	0.125	8	0.125
Monk3	2^6	8	0.125	8	0.125	8	0.125
Parity5+5	2^{10}	12	0.012	12	0.012	12	0.012
Parity5+2	2^{10}	61	0.060	61	0.060	25	0.024
Vote	2^{16}	99	0.002	99	0.002	35	0.001
Lenses	2^4	6	0.375	6	0.375	6	0.375
Zoo	2^{16}	1751	0.027	406	0.006	57	0.001
Mushroom	2^{22}	4173	0.001	1366	0.000	106	0.000
Sonar	2^{60}	224583	0.000	10562	0.000	296	0.000

All 为整个搜索空间的大小，*Ratio* 为搜索空间压缩率，等于 *Ev* 除以 *All*。

表7 BEMMIMAMI 算法属性约简结果的启发式粗糙集值约简分类错误率

Dataset	BEMMIMAMI (M=2p)	BEMMIMAMI (M=p)	BEMMIMAMI (M=1)
Corral	0.00	0.00	0.00
Monk1	3.21	3.21	3.21
Monk3	1.45	1.45	1.45
Parity5+5	0.00	0.00	0.00
Parity5+2	$0.00^{(1)}$ $0.00^{(2)}$ $0.00^{(3)}$ $0.00^{(4)}$	$0.00^{(1)}$ $0.00^{(2)}$ $0.00^{(3)}$ $0.00^{(4)}$	0.00
Vote	5.23	5.23	5.23
Lenses	25.33	25.33	25.33
Zoo	$5.58^{(1)}$ $5.02^{(2)}$ $4.38^{(3)}$ $6.81^{(4)}$ $5.08^{(5)}$ $3.44^{(6)}$ $2.69^{(7)}$	$5.58^{(1)}$ $5.02^{(2)}$ $4.38^{(3)}$ $6.81^{(4)}$ $5.08^{(5)}$ $3.44^{(6)}$ $2.69^{(7)}$	5.58
Mushroom	$2.64^{(1)}$ $1.83^{(2)}$	$2.64^{(1)}$ $1.83^{(2)}$	2.64
Sonar	-	-	24.68

(上接第 185 页)

论了不同障碍对连通性的不同影响,提出了带障碍的分级聚类算法 OBHIEC。该算法首先依据障碍将整个区域划分成多个子区域,在每个子区域内不包含障碍,任意两个数据对象之间直接可达;然后在各个子区域内,使用不考虑障碍的聚类算法对子区域内的数据对象进行一级聚类,聚类后每个簇用其中心来代表,所有簇的代表点构成一个集合;最后在代表点集合上应用带障碍的聚类算法对代表点进行二级聚类,聚集为同一个簇的所有代表点所代表的数据对象同属一个聚类。分级聚类使得需要计算障碍距离的点对数目减少,并能处理数据分布密度不同的情况。本文实验结果表明,OBHIEC 算法能有效完成带障碍的聚类,并具有较好的增量特性,适用于大规模数据集。

参 考 文 献

1 Tung A K H, Hou Jean, Han Jiawei. Spatial Clustering in the Presence of Obstacles. Int conf on Data Engineering(ICDE'01), Heidelberg, Germany, April 2001
2 陈克平,等. 一种带障碍的网格弥散聚类算法 DCellO. 计算机研究与发展(增刊),2004,41(Suppl)
3 Han Jiawei, Kamber M. 数据挖掘——概念与技术(影印版). 北京: 机械工业出版社,2001
4 Ng R, Han J. Clarans: A method for clustering objects for spatial data mining. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(5):1003～1016
5 Wang W, Yang J, Muntz R. STING: A statistical information grid approach to spatial data mining. In:Proc 1997 Int Conf Very Large Data Bases (VLDB'97), Athens, Greece, Aug 1997. 186～195
6 Karypis G, Han E-H, Kumar V. Chameleon: a hierarchical clustering algorithm using dynamic modeling. Computer, 1999, 32:32～68
7 Ng R, Han J. Efficient and effective clustering methods for spatial data mining. In. Bocca J, Jarke M, Zaniolo C, eds. Twentieth International Conference on Very Large Databases Santiago, Chile, 1994. 144～145