旋转网格:一种新的聚类融合方法

曹巧玲 郭华平 范 明
(郑州大学信息工程学院 郑州 450052)

摘 要 网格聚类以网格为单位学习聚簇,速度快、效率高。但它过于依赖密度阈值的选择,并且构造的每个聚簇边界呈锯齿状,不能很好地识别平滑边界曲面。针对该问题,提出一种新的面向网格问题的聚类融合算法(RG)。RG不是通过随机抽样数据集或随机初始化相关参数来创建有差异的划分,而是随机地将特征划分为K个子集,使用特征变换得到K个不同的旋转变换基,形成新的特征空间,并将网格聚类算法应用于该特征空间,从而构建有差异的划分。实验表明,RG能够有效地划分任意形状、大小的数据集,并能有效地解决网格聚类过分依赖于密度阈值选择以及边界处理过于粗糙的问题,其精度明显高于单个网格聚类。

关键词 网格聚类,聚类算法,聚类融合

Rotation Grid: A New Cluster Ensemble Method

CAO Qiao-ling GUO Hua-ping FAN Ming
(Department of Computer Science, Zhengzhou University, Zhengzhou 450052, China)

Abstract Although it is rapid and efficient to use the grid-based clustering approach to learn the partition of a data set, grid clustering is excessively dependent on the initialization of density threshold, and the margin of each cluster constructed by the approach presents zigzag manner, which prohibits the recognition of smooth boundary surface. Thus, this paper proposed a new grid-oriented cluster ensemble approach (RG) to solve this problem. Instead of constructing the partitions with diversity on a given data set by random sampling or initializing parameters of corresponding algorithm, RG randomly splits the features set into K subsets, uses feature transformation method on the subsets to learn K different rotation basis, and applies grid cluster algorithm to the new feature space formed by the K axis rotations to learn the partitions with diversities. Experimental results show that, compared with single grid clustering, RG not only partitions the data set with arbitrary shape or size efficiently, but also alleviates its dependence on the density threshold initialization and smoothes the rough boundary.

Keywords Grid clustering, Clustering algorithm, Clustering ensemble

1 引言

聚类是一种无监督学习分类的过程,其结果是将数据对象分组成为多个簇(类),使得在同一个簇中的对象具有较高的相似度,而不同簇中的对象之间尽量相异。迄今为止,人们已经提出了许多聚类算法[1]。其中,基于网格的聚类算法易于实现且能很好地处理高维数据,因此被广泛应用于空间数据离散化等问题中。然而,网格聚类方法过度依赖于密度阈值MinPts的选择。如果MinPts太高,则簇可能丢失;反之,则本应分开的两个簇可能被合并。如果存在不同密度的簇和噪声,也许不可能找到适用于数据空间所有部分的单个MinPts值。另外,矩形网格单元不能准确地捕获圆形边界区域的密度,使得聚类边界成锯齿状,不能很好地识别平滑边界曲面。

聚类分析的主要目的是获得目标数据的真实分布情况。为了实现这一目的,传统的作法是对于目标数据,分别采用单一聚类算法实现,然后选择其中一个最好的聚类算法作为最终的解决方法。但是研究发现,尽管其中一个聚类算法有着较好的性能,但是不同的聚类算法产生的聚类集合是不重叠的,这表明不同的聚类算法对聚类的模式有着互补信息,可以利用这种互补信息来提高聚类性能。组合分类方法如bagging[2]和boosting[3]已被证明是非常普遍和有效的、能够改进学习精确度的监督方法。依据同样的原理,聚类融合的目的是融合来自多个划分的结果以得到更高质量和鲁棒性的聚类结果。与数据挖掘和机器学习领域中较成熟的组合分类方法相比,虽然至今为止已有大量聚类算法被提出,但聚类融合方法的研究近几年才开始出现。聚类融合可以得到比单一聚类算法更好的结果,主要体现在[4]:

(1)鲁棒性:在各领域和数据集中有更好的平均性能;

(2)新颖性:能找到单一聚类算法难以得到的组合聚类结果;

(3)稳定性和信任度估计:聚类结果对噪声、孤立点和抽

到稿日期:2010-08-19 返修日期:2010-11-30 本文受国家自然科学基金项目(60773048)资助。

曹巧玲(1983-),女,硕士生,主要研究方向为数据挖掘、机器学习等,E-mail:caoqiaolingshangqiu@126.com;郭华平(1982-),男,博士生;范 明(1948-),男,教授,博士生导师。

样变化不敏感，聚类的不确定性能从组合分布中得到弥补；

(4)并行性和可扩展性：能对数据子集进行并行聚类并组合子序列结果，能对多个分布式数据源或特征属性的聚类结果进行合并。

结合网格聚类的优缺点，本文提出一种新的聚类融合算法(RG)。给定数据集 D，RG 采用迭代方法学习有差异的聚类器。对于第 i 次迭代，RG 随机地将特征划分为 K 个子集，使用特征变换学习 K 个不同的旋转矩阵，形成新的特征空间 M_i，结合 D 与 M_i 构造新的训练数据集合 D_i，将网格聚类算法应用于 D_i 得到新的划分 p_i，并加入到 $P=\{p_j|j=1,2,\cdots,i-1\}$。然后设计融合函数融合 P 中成员，得到一个准确率较高的聚类结果。该算法保持了网格聚类算法速度快、效率高的特点，并解决了单个网格聚类对密度阈值 T 的选择依赖问题以及对边界处理粗糙的问题。

本文第 2 节介绍相关研究的背景知识；第 3 节给出面向网格的聚类融合算法的形式化描述和伪代码；第 4 节是实验结果和本文的方法与单个网格聚类方法的分析比较。最后总结全文。

2 背景介绍

2.1 网格背景介绍

现有的聚类算法如 DBSCAN[5]，Chameleon[6] 等，都是致力于如何发现任意形状和大小的类，但在聚类的过程中需要用户输入一些参数，如密度阈值 MinPts，来判断网格是否为高密度单元，以便有效地去除孤立点或噪声。这就将选择有效参数的任务留给了用户，而算法对参数值又是非常敏感的：参数设置的细微不同可能会导致差别很大的聚类结果[7]。在文献[8]可知，DBSCAN 算法由于选取的参数不同导致了聚类结果的巨大差异性。同时表明 DBSCAN 算法聚类结果的好坏在很大程度上取决于参数的选择，并且很难有效地处理好聚类结果对参数的依赖性。

为了解决参数依赖问题，本文使用参数自动化技术来处理网格某一维划分数目(k)以及网格的密度阈值(T)。该密度阈值定义了网格稀疏或稠密的边界值。这种方法很好地解决了聚类结果对参数的依赖性问题。令$|D|$为数据集 D 中数据点的个数，则 k 定义为

$$k=\sqrt{|D|} \tag{1}$$

关于密度阈值，用一种贪心的方法搜索近似最优密度阈值，其处理过程如下：根据 k 的值划分网格，并将数据集 D 映射到网格单元中，计算出网格单元中点数的最大值 Max 和非空的网格数 G_n。初始化 $W=\mathrm{Max}^{1/2}$，$A_1=\mathrm{Max}$，$A_m=A_{m-1}-W$，$B_m=(A_m+A_{m+1})/2$，$C=N/G_n$，其中 $W>1$，$m\geqslant 1$ 且 $m\leqslant W$；则密度阈值 T 的取值为

$$T=\frac{C\sum_{m=1}^{W-1}\frac{B_m}{W-1}}{\sum_{m=1}^{W}\frac{A_m}{H}}=\frac{HC\sum_{m=1}^{W-1}B_m}{(W-1)\sum_{m=1}^{W-1}A_m} \tag{2}$$

实验表明，采用参数自动化聚类方法，可以有效地解决聚类过程中对参数的选择依赖。聚类结果中的簇和孤立点或噪声能够被有效地分离开来。然而，把该方法用于数据集 D 产生的不同聚类结果的误聚类集合仍是不重叠的。为了进一步提高聚类结果的准确率，本文引入了聚类融合思想，下面简单介绍聚类融合的背景。

2.2 聚类融合背景介绍

给定数据集 $X=\{x_1,x_2,x_3\cdots,x_n\}$，对 X 采用空间变换，产生 h 个不同的数据集$\{X_1,X_2\cdots,X_h\}$，并在这些数据集上用相同或不同的算法学习聚类，得到 h 个聚类成员$\{P_1,P_2,P_3,\cdots,P_h\}$，其中 $P_i(i=1,2,3,\cdots,h)$为运行第 i 次算法得到的聚类成员。设计一种融合函数 Γ，对这 h 个聚类成员进行合并，得到一个最终的聚类成员 P'，聚类融合过程如图 1 所示。

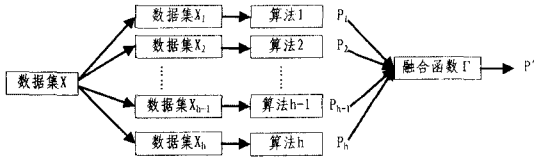


图 1 聚类融合示意图

近年来，聚类融合的研究主要集中在以下两个方面：如何产生有效的聚类成员，不同的聚类成员之间需要有什么样的差异度；如何设计融合函数以便对聚类成员进行合并，包括对聚类成员结果标志的匹配(在不同聚类结果中，对实质相同的类给予相同的标志)等。

产生不同聚类成员最常用的方法是采用 K-means 方法：随机选择不同的初始点运行多次，从而产生所需要的聚类成员。该方法算法复杂度低，运行方便，但是对于分布非球形的数据以及在处理高维数据时，效果并不理想。Bidgoli[9] 将随机抽样的方法运用于数据集，产生不同的数据子集，然后对每个数据子集使用 K-means 算法生成不同的聚类成员。Fern[10] 等采用随机投影法将高维数据投影到低维空间，通过多次投影得到若干个数据子集，然后用 EM 聚类算法对每次的投影子集聚类，从而得到聚类成员。

在设计聚类融合函数方面，Fred[11] 提出了 Co-Association 矩阵，用于衡量数据点之间的相似度，其中第 i 个数据点与 j 个数据点之间的相似度为：

$$A_{ij}=\frac{i\text{与}j\text{属于同一聚类的次数}}{\text{聚类算法总次数}} \tag{3}$$

文中提出的 Voting-Kmeans 方法首先随机产生初始点进行 K-means 算法，运行 H 次得到聚类成员，然后计算出聚类成员的 Co-Association 矩阵。该方法选取 0.5 为阈值，Co-Association 矩阵中大于 0.5 的点即属于最终聚类结果中的同一类。该方法空间复杂度为 $O(n^2)$，时间复杂度为 $O(kn^2)$，由此看出它对于海量数据是不太适合的。阳琳赟等[12] 提出了一种对聚类成员进行加权的融合方法。该方法引入粗糙集理论中的属性重要性度量，根据聚类成员对融合的重要性赋予其权重，生成加权共生矩阵，进而产生融合结果。Strehl 和 Ghosh[13] 提出了 3 种基于超图的方法：CSPA，HGPA 和 MCLA。CSPA (*Cluster-based Similarity Partitioning Algorithm*)是利用共识矩阵值作为边的权值进行图分割的算法。假设有 N 个数据对象 $D_i(i=1\cdots N)$，由不同的聚类过程产生 H 个聚类成员(clusterings)，通过表决法产生共识矩阵 CA (co-association matrix)，其计算方法为：$CA(i,j)=\mathrm{stime}(i,j)/H$。其中，$\mathrm{stime}(i,j)$为表决数据对象 D_i 和 D_j 在同一个聚类中的聚类成员个数。由数据对象集中的所有对象作为顶点、矩阵 CA 所对应的值作为边权值构成图，然后运用图分割算法实现最终的聚类融合。HGPA(Hypergraph Partitioning Algorithm)首先生成一个超图，超图由超边、顶点组成。其中顶点为数据点。在一般图中，一条边连接两个顶点；在超图

中，一条超边可以连接任意的顶点。每一个聚类成员中的每个类均为一条封闭的边，包含了属于该类的所有顶点，所有的顶点和边的权重都设置为相同的值。然后 HGPA 将聚类融合问题转化成通过切割最少的超边来划分超图的问题。在这里，HGPA 考虑多个数据对象之间的关系信息，而 CSPA 仅考虑两两之间的关系。MCLA(Meta-Clustering Algorithm)是一种对聚类成员再聚类的超图算法，MCLA 的思想为：将聚类成员中的每个簇写成单位向量形式：检查数据集的每个点，如果点在簇中，则标记为 1；否则标记为 0。将每个向量看作顶点，构造一个带权超图。然后对超图边进行断裂操作，再由断裂操作来决定每个数据点最终属于的类。

本文沿用了 MCLA 算法的思想，并提出一种新的聚类融合方法，该方法对 H 个有一定互补信息的网格聚类成员$\{P_1, P_2, P_3, \cdots, P_h\}$进行融合，得到一个最终准确率比较高的划分结果。

3 一种面向网格的聚类融合算法

由前面的讨论可知，网格聚类不能很好地处理噪声问题，并且不能很好地识别平滑边界曲面。为了解决该问题，本文提出一种新的面向网格问题的聚类融合算法 RG(Rotation Grid)。聚类融合算法成功的关键是如何学习有差异聚类成员以及如何设计合理的划分融合函数。

对于前一个问题，论文引入基于特征抽取的方法学习不同的聚类成员。将特征集 F 划分成 k 个子集并把相应的特征变换(如 PCA[14])应用到训练数据集合 D 上，学习得到不同的特征变换矩阵 M；然后在 $D_i=MD$ 上学习不同聚类成员 P_i。在此选择网格聚类算法作为聚类融合的基聚类器。该方法的核心问题是选择或构建特征变换矩阵。例如，当$|F|=2$时，特征划分对创建有差异的训练数据集并无很大益处。因此，本文引入随机变换方法处理该问题。具体过程如下：假定有二维数据集 $D=\{(x_1^i, x_2^i)^T \mid i=1,2,3,\cdots,n\}$，本文采用迭代的方法为聚类器 f 创建不同的训练数据集；对于第 j 次迭代，令：

$$D_{ji}=M_jD_i+M_j'=\left[\begin{pmatrix}\sin\alpha_j & \cos\alpha_j\\ \cos\alpha_j & -\sin\alpha_j\end{pmatrix}^T\begin{pmatrix}x_i\\ y_i\end{pmatrix}\right]^T+\begin{pmatrix}a_j\\ b_j\end{pmatrix} \tag{4}$$

式中，D_{ji}是 D_j 的第 i 个成员，$a_j\in(0,15)$表示旋转的角度，M_j'为平移矩阵。通过随机化 α 以及 M_j'，构建不同的数据集 D_j，并在 D_j 上学习划分 $P_j(j=1,2,\cdots,h)$，进而得到 H 个具有差异性的聚类成员$\{P_1, P_2, P_3, \cdots, P_h\}$。对于$|F|=3$，使用

$$D_{ji}=M_jD_i+M_j'=(T_{rx}*T_{ry}*T_{rz})\begin{pmatrix}x_i\\ y_i\\ z_i\end{pmatrix}+\begin{pmatrix}a_j\\ b_j\\ c_j\end{pmatrix} \tag{5}$$

对数据集做旋转变换，其中

$$T_{rx}=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\alpha & \sin\alpha\\ 0 & -\sin\alpha & \cos\alpha\end{bmatrix}\quad T_{ry}=\begin{bmatrix}\cos\beta & 0 & -\sin\beta\\ 0 & 1 & 0\\ \sin\beta & 0 & \cos\beta\end{bmatrix}$$

$$T_{rz}=\begin{bmatrix}\cos\gamma & \sin\gamma & 0\\ -\sin\gamma & \cos\gamma & 0\\ 0 & 0 & 1\end{bmatrix} \tag{6}$$

通过随机化 $\alpha,\beta,\gamma(\alpha,\beta,\gamma\in(0,360))$，构建不同的训练数据集。当$|F|\geqslant 3$时，划分 F 为$|F|/3$个子集，并在每个子集上应用式(6)。

对于设计融合函数，将聚类成员 $P_i(i=1,2\cdots,h)$中的每个簇写成单位向量形式：检查数据集的每个点，如果点在簇中，则标记为 1；否则标记为 0。将每个向量看作顶点，构造一个带权超图。假定 P_a 和 P_b 是两个不同的聚类成员，分别将样本集聚成 k 个簇，即 $P_a=\{C_{a1}, C_{a2}, \cdots, C_{ak}\}$和 $P_b=\{C_{b1}, C_{b2}, \cdots, C_{bk}\}$，对应超图的顶点分别为 $V_a=\{V_{a1}, V_{a2}, \cdots, V_{ak}\}$和 $V_b=\{V_{b1}, V_{b2}, \cdots, V_{bk}\}$。对于属于聚类成员 P_a 和 P_b 的任意一对簇 C_{ai}和 C_{bj}，在这里用 W_{ij}表示对应于簇对 C_{ai}和 C_{bj}的顶点对 V_{ai}和 V_{bj}之间的边权重。边权重用式(7)计算：

$$w_{ij}=\frac{h_a{}^Th_b}{\|h_a\|_2^2+\|h_b\|_2^2-h_a{}^Th_b} \tag{7}$$

依次计算出所有顶点对之间的边权重，并找出边权重最大的两个簇，用相同的标签来标示。然后用投票的方法决定数据集中的每个点所在的簇。表 1 给出 RG 算法的伪代码。

表 1 RG 算法伪代码

```
算法名称：RG
输入：一个具有 n 个样本的数据集 X={x1,x2,x3…,xn}；
    H：聚类成员的个数；
输出：样本的聚类结果；
方法：
    初始化矩阵 matrix[i][j](i=1,2,3,…,K;j=1,2,3,…,K);//存储第 i 个样本被归入第 j 类的次数；
for(i=1;i<=H;i++){
    (1)  统计相关数据，采用参数自动化方法计算网格划分值 K，密度阈值 T；
    (2)  由式(4)构建数据集 Di，并在相应的数据集上学习不同的划分，生成聚类成员 Pi,i=1,2,…,h；
    }
    (3) 选择其中一个聚类成员作为基准来进行标签转换；
    (4) for i=1,2,…,k{
    (5)     for j=1,2,…,k{
    (6)         计算分别属于聚类成员 Pa 和 Pb 中两个簇(Cai,Cbj)对应顶点对之间的边权重 Wij
        }
    (7)     查找出最大的 Wij；即聚类成员 Pa 中第 i 类聚类成员 Pb 中第 j 类之间的边权重最大
    (8)     将聚类成员 Pa 中第 i 类与聚类成员 Pb 中第 j 类用相同的类标签标示
    }
    (9) 依次扫描聚类成员{P1,P2,P3,…,Ph}，计算每个样本被分别归入 k 个类中的次数，建立矩阵 matrix[i][j]
    (10) 扫描矩阵 matrix[i][j]，依次比较样本 i 被分别归入 k 个类中的次数，选择最大的 matrix[ i][ j]值，然后将样本 i 归入第 j 类
```

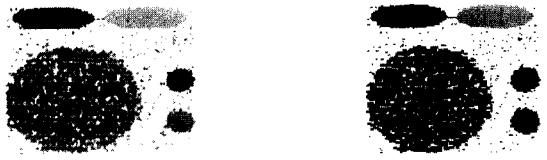
4 实验与结果分析

4.1 综合数据集

实验环境是 Intel(R)Pentium(R)D 2.80 GHz CPU，512 MB 内存，Microsoft Windows XP。算法的编写和编译是在 eclipse 环境下实现的。为了验证算法的正确性和精度，本文做了大量的实验，仅取 3 个具有代表性的进行具体的说明。

图 2 是一个多密度的、具有较平滑边界曲面的数据集。它含有 5034 个记录，包含 5 个不同形状、不同大小的类以及一些随机分散的孤立点。图 2(a)是采用参数自动化方法生成的单个网格聚类的结果。图 2(b)是 RG 算法的聚类结果。使用的参数为 $H=20$。由图 2 可以看出，单个网格聚类基本上能够识别出各个类，但却吸收了一些孤立点到类中，并且有很多边界点被误判，聚类精度比较低。而 RG 算法不仅能够

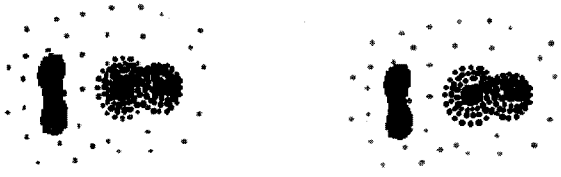
有效地识别出各个类，孤立点以及边界点也能够被有效地识别出来，聚类结果较好。



(a)单个网格聚类　　(b)RG

图 2　单个网格聚类与 RG 聚类结果比较 1

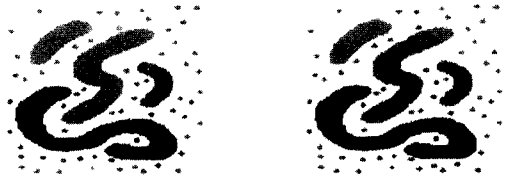
图 3 是一个带噪声的多密度的数据集，含有 2353 条记录，分为 4 类，其中 3 类属于高密度数据集，另一类为形状类似“8”字型的低密度数据集。图 3(a) 是采用参数自动化方法生成的单个网格聚类的结果。可以看出，单个网格算法只能识别出高密度类，却不能识别出“8”字型的低密度类。图 3(b)是 RG 算法的聚类结果。使用的参数为 $H=21$。该数据集是比较特殊的多密度数据集，高密度被低密度包围，不同密度的聚类没有分离开来。由图 3 可以看出，RG 算法不仅可以正确地识别出高密度类，还可以识别出“8”字，有很高的聚类精度。



(a)单个网格聚类　　(b)RG

图 3　单个网格聚类与 RG 聚类结果比较 2

图 4 是一个密度均匀的具有较平滑边界曲面的数据集。它含有 9993 个记录，包含 4 个不同形状、不同大小的类，并带有噪声。图 4(a)是采用参数自动化方法生成的单个网格聚类的结果。图 4(b)是 RG 算法的聚类结果。使用的参数为 $H=25$。由图 4 可以看出，单个网格聚类基本上能够识别出各个类，但却不能识别噪声，使得一些噪声点被聚到簇内。而 RG 算法不仅能够能有效地识别出各个类，噪声以及边界也能够被有效地识别出来，聚类结果较好。



(a)单个网格聚类　　(b)RG

图 4　单个网格聚类与 RG 聚类结果比较 3

4.2　真实数据集

为了进一步地验证 RG 算法，从 UCI 机器学习数据存储库中选择 2 个数据集 Iris 和 Wine 进行实验。Iris 数据集为 4 维，分为 3 类，有 150 条记录，每一类有 50 个记录；Wine 数据集为 13 维，分为 3 类，有 178 条记录，其一类分别有 59，71，48 个记录。

在本次实验中，用 $|A|$ 表示第 i 类实际包含的元素个数，用 $|B|$ 表示实验所检测到的第 i 类包含的元素个数，$|A\cap B|/|B|$ 为聚类的精度(Precision)，$|A\cap B|/|A|$ 为聚类的召回率(Recall)。采用参数自动化方法生成的单个网格聚类的结果与 RG 算法的聚类结果如表 2 和表 3 所列。

表 2　单个网格算法在 Iris 和 Wine 数据集上的实验结果

比较值	Iris			Wine		
	i=1	i=2	i=3	i=1	i=2	i=3
$\|A\|$	50	50	50	59	71	18
$\|B\|$	49	57	23	56	51	47
$\|A\cap B\|$	47	43	19	48	42	42
Precision(%)	95	75	82	86	82	89
Recall(%)	94	86	38	81	59	88

表 3　RG 算法在 Iris 和 Wine 数据集上的实验结果

比较值	Iris			Wine		
	i=1	i=2	i=3	i=1	i=2	i=3
$\|A\|$	50	50	50	59	71	48
$\|B\|$	49	53	33	58	63	43
$\|A\cap B\|$	49	47	31	53	51	41
Precision(%)	100	89	93	91	80	95
Recall(%)	98	94	62	90	71	86

从表 2 和表 3 可以看出，在真实数据集上，RG 算法的融合结果要高于采用参数自动化方法生成的单个网格聚类的结果。但是在聚类过程中，对于密度不均匀的数据集，如果各个类是交叉分布的，即各个簇没有分离开来，则实验结果不是很理想。

结束语　本文提出了基于旋转网格的聚类融合算法，即通过旋转网格得到具有差异的数据集，在这些数据集上学习聚类得到不同的划分。然后设计融合函数对聚类成员进行融合，得到高质量和鲁棒性的聚类结果。实验表明，该方法能对含有不同形状、不同密度、不同大小聚类及噪声的数据集进行聚类，能够较好地解决单个网格聚类对密度阈值 T 的选择依赖问题以及对边界处理粗糙的问题。

参 考 文 献

[1] Han Jia-wei, Kamber M. Data Mining: Concepts and Techniques(2nd ed)[M]. Morgan Kaufmann Publishers, 2006: 398-401

[2] Tulyakov S, et al. Review of Classifier Combination Methods [M]. Studies in Computational Intelligence(SCI), 2008: 361-386

[3] Dudoit S, Fridlyand J. Bagging to improve the accuracy of a clustering procedure[J]. Bioinformatics, 2003, 19(9): 1090-1099

[4] Topchy A, Jain A K, Punch W. Clustering for grouping of smooth curves and texture segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(4): 513-518

[5] Chen Min, Gao Xue-dong, Li Hui-fei. Parallel DBSCAN with Priority R-tree[C]// The 2nd IEEE International Conference on Information Management and Engineering(ICIME). Chengdu, April 2010: 508-511

[6] Shacham O, Vechev M, Yahav E. Chameleon: adaptive selection of collections[C]// Proceedings of the 2009 ACM SIGPLAN conference gramming language design and implementation. June 2009

[7] Chen Ling, Tu Li, Chen Hong-jian. Data clustering by ant colony on a digraph[C]// Proceedings of the 4th International Conference on Machine Learning and Cybernetics. Guangzhou, August 2005: 1686-1692

[8] Ertoz L, Steinbach M, Kumar V. Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data

[C]// Proceedings of the 2003 SIAM International Conference on Data Mining(SDM'03). San Francisco:CA,2003

[9] Minaei-bidgli B, Topchy A, Punch W F. Ensembles of Partitions via Data Resampling[C]// Proceedings International Conference on Information Technology, Coding and Computing (ITCC 2004). 2004:188-192

[10] Fern X Z, Brodley C E. Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach[C]// Proceedings of the 20th International Conference on Machine Learning. 2003:186-193

[11] Fred A L. Finding Consistent Clusters in Data Partitions[C]// Proceedings of the 2nd International Workshop on Multiple Classifier Systems. Volume 2096 of Lecture Notes in Computer Science. Springer, 2001:309-318

[12] 阳琳赟,周海京,卓晴,等. 基于属性重要性的加权聚类融合[J]. 计算机科学,2009,36(4):243-249

[13] Strehl A, Ghosh J. Cluster Ensembles: A Knowledge Reuse Framework for Combining Multiple Partitions[J]. Journal of Machine Learning Research, 2003, 3(3):583-617

[14] Rodriguez J J, Kuncheva L I, Alonso C J. Rotation forest: A new classifier ensemble method[J]. IEEE Transactions Pattern Anal Mach Intell, 2006, 28:1619-1630

(上接第 151 页)

系通过预处理时属性频度和数据对象频度计算相联系,避免了相互之间大量的 LOF 计算,同时在挖掘过程中使用熵值作为度量标准,避免了数据集中离群点挖掘时高维空间距离度量的弊端。SPOD 算法在利用信息熵选取子空间后还要进行 LOF 计算,数据量较大时,耗费的时间较多。利用信息论作为度量标准得出的离群点易于解释,熵值比较大,不确定性比较大,成为异常也就很正常了。

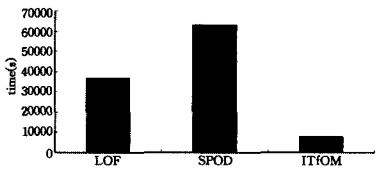


图 2 不同算法的执行时间对比

5.4 算法对数据集维数的伸缩性

为了测试算法对数据集维度的伸缩性,分别在计算熵权后,将参数设置为 35,28,18,9 和 7,按序提取属性,去除冗余属性,对 LOF 算法、SPOD 算法和 ITfOM 算法运行情况进行了分析。

从图 3 可以看出,由于 ITfOM 算法和 SPOD 算法对数据进行了降维处理,因此相应算法对高维有更好的伸缩性,而 LOF 算法没有对高维数据进行降维处理,算法对高维数据的挖掘精确性受到明显的影响。

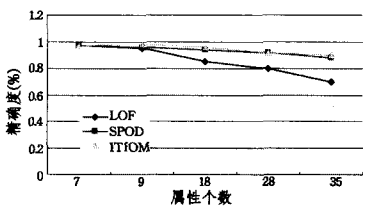


图 3 不同算法对数据集属性个数的伸缩性

结束语 本文研究了高维海量数据集的离群点挖掘问题,提出了基于信息论的高维海量数据的离群点挖掘算法。该算法采用了信息论中信息熵和互信息的概念,利用估算的互信息数值客观地依据熵权排序后进行属性选择,去除冗余属性降维。利用信息熵作为离群点判断的度量标准,可消除距离和密度量纲的弊端。实验结果表明,所提出的算法针对高维海量数据是切实有效的,效率和精度得到了明显提高。进一步提高算法的实现效率和精度以及实时挖掘高速数据并将其扩展到分布式环境,是下一步的研究内容。

参 考 文 献

[1] Knorr E M, Ng R T. Algorithms for mining distance-based outliers in large datasets[C]// A Gupta, O Shmueli, J Widom, eds. Proc of the 24th int'l conf on Very Large Databases. New York: ACM, 1998:392-403

[2] Yu D, Sheikholeslami S, Zhang A. FindOut: Finding Outliers in very large datasets[R]. 99-03. Univ. of New York, Buffalo, 1999:1-19

[3] Breunig M M, Kriegel H, Ng R T, et al. LOF: Identifying density-based local outliers[C]// Chen WD, Naughton JF, Bernstein PA, eds. Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. Dallas: ACM Press, 2000:93-104

[4] Papadimitirou S, Kitagawa H, Gibbons P B, et al. LOCI: Fast outlier detection using the local correlation integral[C]// Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the 19th Int'l Conf. on Data Engineering. IEEE Computer Society Press, 2003:315-326

[5] Aggarwal C, Yu P. An effective and efficient algorithm for high-dimension outlier detection[J]. The VLDB Journal, 2005, 14(2):211-221

[6] 倪巍伟,陈耿,陆介平,等. 基于局部信息熵的加权子空间离群点检测算法[J]. 计算机研究与发展,2008,45(7):1189-1194

[7] Fleuret F. Fast binary feature selection with condition mutual information[J]. Journal of machine learning research, 2004, 5:1531-1555

[8] Wang G, Loehovsky F. Feature seleetion with conditional mutual information maximin in text Categorization[C]// Proeeeding of the Thirteenth ACM Conference on Information and know-ledge management. 2004:342-349

[9] Mao Ye, Xue Li, Orlowska M E. Projected outlier detection in high-dimensional mixed-attributes data set[J]. Expert Systems with Applications, 2009, 36(3)PART 2:7104-7113

[10] He Zeng-you, Xu Xiao-fei, Deng Sheng-chun. A fast Greedy Algorithm for Outlier Mining[C]// Proceedings of PAKDD'2006 (LNAI3918). 2006:567-576

[11] 于绍越,商琳. ENBROD:基于信息熵的相对离群点的检测方法[J]. 南京大学学报:自然科学,2008,44(2):1189-1194

[12] Jiang Feng, Sui Yue-fei, Cao Cun-gen. An information entropy-based approach to outlier detection in rough sets[J]. Expert Systems with Applications, 2010, 37(10):6338-6344

[13] Kraskov A, Stogbauer H, Grassberger P. Estimating Mutual information[J]. Physical Review, 2004, E69(7):69-84

[14] 周晓云,张净,孙志挥. 高维 Turnstile 型数据流聚类算法[J]. 计算机科学,2006,33(11):14-17

[15] 张净,孙志挥. GDLOF:基于网格和稠密单元的快速局部离群点检测算法[J]. 东南大学学报:自然版,2005,42(5):863-866

[16] 李存华,孙志挥. GridOF:面向大规模数据集的高效离群点检测算法[J]. 计算机研究与发展,2003,40(11):1586-1592