软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(4):935−944 [doi: 10.13328/j.cnki.jos.005415]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于密度差分的自动聚类算法*

陈朝威[1,2], 常冬霞[1,2]

[1](北京交通大学 计算机与信息技术学院,北京 100044)
[2](北京交通大学 信息科学研究所,北京 100044)
通讯作者: 常冬霞, E-mail: dxchang@bjtu.edu.cn

摘 要: 聚类作为无监督学习技术,已在实际中得到了广泛的应用.但是对于带有噪声的数据集,一些主流算法仍然存在着噪声去除不彻底和聚类结果不准确等问题.提出了一种基于密度差分的自动聚类算法(clustering based on density difference,简称 CDD),实现了对含有噪声数据集的自动分类.所提算法根据噪声数据和有用数据密度的不同,实现了去噪声和数据的分类,并通过构建数据间的邻域,进一步实现了对有用数据间不同类别的划分.通过实验验证了所提算法的有效性.

关键词: 聚类;数据挖掘;离散点检测;差分;CDD
中图法分类号: TP18

中文引用格式: 陈朝威,常冬霞.基于密度差分的自动聚类算法.软件学报,2018,29(4):935−944. http://www.jos.org.cn/1000-9825/5415.htm
英文引用格式: Chen ZW, Chang DX. Automatic clustering algorithm based on density difference. Ruan Jian Xue Bao/Journal of Software, 2018,29(4):935−944 (in Chinese). http://www.jos.org.cn/1000-9825/5415.htm

Automatic Clustering Algorithm Based on Density Difference

CHEN Zhao-Wei[1,2], CHANG Dong-Xia[1,2]

[1](School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)
[2](Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China)

Abstract: As an unsupervised learning technology, clustering has been widely used in practice. However, some mainstream algorithms still have problems such as incomplete noise removal and inaccurate clustering results for the datasets with noise. In this paper, an automatic clustering algorithm based on density difference (CDD) is proposed to realize automatic classification of the datasets containing the noise. The algorithm is based on the density difference between noise data and useful data to achieve removing noise and data classification. Moreover, the useful data are classified into different classes through the neighborhood construction procedure. Experimental results demonstrate that the CDD algorithm has high performance.

Key words: clustering; data mining; outliers detection; difference; CDD

随着互联网的普及以及大数据时代的来临,聚类作为数据挖掘和机器学习领域中重要的研究课题[1],正在发挥着越来越重要的作用.聚类分析也称为点群分析,它是根据样本的属性,基于某种相似性或差异性将大量数据划分为若干类.简单地说,基于数学方法来确定样本之间的亲密关系,根据这种关系对样本进行类别划分,使

* 基金项目: 国家自然科学基金(61532005)
Foundation item: National Natural Science Foundation of China (61532005)
本文由“多媒体大数据处理与分析”专题特约编辑赵耀教授、李波教授、华先胜研究员、文继荣教授、蒋刚毅教授、常冬霞副教授推荐.
收稿时间: 2017-05-03; 修改时间: 2017-06-26; 采用时间: 2017-10-13; jos 在线出版时间: 2017-12-01
CNKI 网络优先出版: 2017-12-04 06:49:28, http://kns.cnki.net/kcms/detail/11.2560.TP.20171204.0649.015.html

得同一类数据的差异性最小,不同类数据的差异性最大[2].聚类与分类不同,分类是有监督的过程,即在类别标签已经固定的情况下,即使 1 个样本也能进行分类;而聚类是一种无监督学习过程,即在预先不知道类标志的情况下,自动地进行类别划分的一种方法.目前,聚类分析在统计学、生物学、经济学和市场分析学等领域都得到了广泛的应用[3,4].

当前研究学者已经提出了大量的聚类算法[5],从整体来说,大致可以分为基于划分[6]、基于层次、基于网格[7,8]和基于密度[9−12]这 4 种方法.基于划分的方法就是把数据分为多个不相交的点集,每个点集中相似度尽可能大,代表方法是 k-means[13].这种方法的优点是简单、快速,处理大数据时是相对可伸缩和高效率的.但该算法需要预先给定类别数,且算法对初值敏感,不适合发现非凸形或球形的簇,对“噪声”和孤立点敏感.基于层次的方法就是对数据集进行层次分解,构建一个嵌套的层次树[14−16],每个节点代表一个类别,代表方法有层次凝聚(agglomerative nesting,简称 AGNES)、层次分裂(divisive analysis,简称 DINAN).这种方法的优点是思想简单,容易实现;缺点是聚类过程不可逆,需要人为决定何时停止.基于网格的方法是将数据空间分为有限个单元的网格结构,以单元为对象进行数据处理,代表算法是 CLIQUE(clustering in QUEst).这种方法的优点是处理速度快,速度与划分单元的数目有关;同时缺点也和划分单元的数目有关,精确度和时间都受此影响.基于密度的方法是通过对每个点密度进行定义,根据密度大小进行分类,代表方法有 DBSCAN(density-based spatial clustering of applications with noise)、OPTICS(ordering points to identify the clustering structure).这种方法的优点是可以发现任意形状的簇,数据输入顺序不敏感;缺点是算法性能对参数比较敏感,受密度变化影响较大.

DBSCAN 作为典型的基于密度的聚类算法由于其性能不受数据分布形状的影响,因此在实际中得到了广泛的应用.然而,该算法的性能对参数的选择比较敏感,为了改进其性能,研究者们提出了一些改进算法,比较流行的有基于影响空间的聚类算法(influence space-DBSCAN,简称 IS-DBSCAN)[17]和基于自然近邻的算法(edited natural neighbor algorithm,简称 ENaN)[18].IS-DBSCAN 算法提出了影响空间的概念,充分利用了最近邻和反向最近邻,取两者的交集即为影响空间,并根据影响空间的秩将数据点划分为核心点和噪声点,从而实现了对含有噪声数据的有效聚类.这种方法虽然只使用一个参数实现了去除离散点并进行聚类,但是由于离散点去除得不彻底,会把一些离散点当作一个类别,无法实现数据的有效聚类.ENaN 算法提出了自然最近邻的概念[19],根据样本点近邻,寻找各数据点的反向最近邻,当拥有反向近邻点的个数不增加时,则说明达到稳定状态.样本点的反向最近邻即为其自然最近邻,而一些没有反向近邻的点则被视为噪声点,通过自然最近邻构建邻域图实现聚类.ENaN 虽然通过无参数的方法实现了聚类效果,但对于带有均匀噪声的数据集往往不能实现有效的划分.

为了进一步改善含有噪声数据的聚类性能,实现含噪数据的自动类别划分,本文提出了一种基于密度差分的自动聚类算法,实现对含噪数据集类别数的自动划分.第 1 节首先提出一种基于密度的噪声检测算法以及相关的基础知识.第 2 节详细介绍本文所提出的基于密度差分的自动聚类算法(clustering based on density difference,简称 CDD).第 3 节为实验结果和分析.第 4 节是对我们提出的算法进行总结,并对未来工作加以展望.

1 基于密度差分的噪声检测算法

首先,我们给出一些算法所需要的定义.

定义 1(近邻距离). 样本点 p 的近邻距离 $k_{dist}(p)$ 定义为:

(1) 仅存在 k 个样本 q 使得 $d(p,q)\leqslant k_{dist}(p)$;

(2) 仅存在 $k-1$ 个样本 q 使得 $d(p,q)<k_{dist}(p)$,

其中, $d(p,q)$ 表示样本 p 与 q 之间的距离,这里我们均采用欧式距离.

定义 2(邻域). 样本点 x 的邻域 $NN_k(x)$ 定义为到样本点 x 的距离小于 $k_{dist}(x)$ 的所有样本点的集合,即:

$$NN_k(p)=\{X\in D\setminus\{p\}\mid d(p,X)\leqslant k_{dist}(p)\} \tag{1}$$

其中,D 代表数据集.

定义 3(样本密度). 给定数据集,则其样本点密度可以定义为

$$F(x)=1\Big/\sum\nolimits_{y\in NN_k(x)}d(x,y) \tag{2}$$

由于噪声点和有用数据点往往具有不同的密度,有用数据拥有较大的密度值,噪声数则具有较小的密度值,从而在边界点有一个跳跃.因此,如果我们对此阈值进行有效的估计,即可将噪声和数据进行有效的划分,从而实现去除噪声的目的.下面,我们将以图 1 所示的二维数据集 $D1$ 为例,通过式(2)对数据集中所有的样本密度进行计算并排序,排序后的密度曲线如图 2 所示.

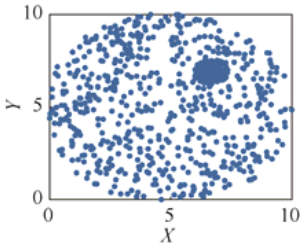


Fig.1 Dataset $D1$

图 1 数据集 $D1$

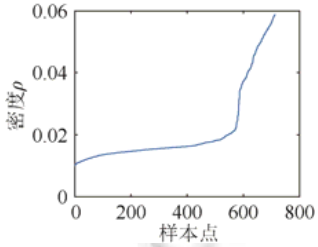


Fig.2 Density ranking curve

图 2 密度排序曲线

从图 2 可以看出,需要找到中间跳跃比较大的密度值来确定我们的阈值.这里,我们提出了一种基于密度差分的噪声检测算法,通过对阈值的有效估计,从而实现对噪声和有效数据的划分,其具体步骤如下所示.

(1) 根据式(2)计算样本密度并进行排序,使得 $F(x_i) < F(x_{i+1})$;

(2) 对排序后密度曲线进行差分计算,即计算 $F(x_{i+1}) - F(x_i)$;

(3) 找到差分最大值对应的密度阈值 $T = \max(F(x_{i+1}) - F(x_i))$;

(4) 根据密度阈值检测噪声点,输出有用数据点.

根据上述所提算法,我们对图 2 所示的数据集进行噪声检测.首先对排序后的密度进行差分运算,结果如图 3 所示,横坐标为样本点,纵坐标为每个点的密度差分值,从而估计出密度阈值点所在的位置.通过这个密度阈值把数据分为两部分,密度小于阈值的点为噪声,密度大于阈值的点为有效数据,数据分类结果如图 4 所示,三角形区域所示为噪声点,椭圆区域所示为有用数据.从图 4 所示结果可以看出,该阈值对数据噪声和数据进行了有效的划分.

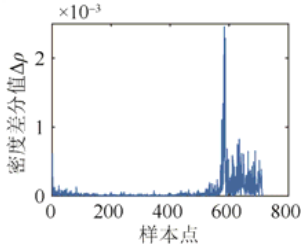


Fig.3 Density difference curve

图 3 密度差分曲线

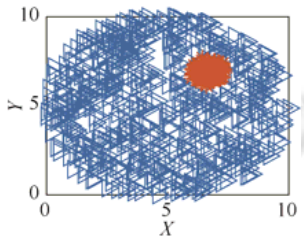


Fig.4 Classification result

图 4 分类结果

然而,对于一些数据集,由于数据集中存在一些特别稀疏的点会产生非常小的密度值,而一些特别密集的区域会产生非常大的密度值,因此,对数据密度进行排序差分后,往往会在差分曲线开始和结束的地方出现比较大的跳跃.下面我们以图 5 所示的数据集 $D2$ 为例来说明这一情况.对于图 5 所示的数据采用上述步骤进行差分后所得差分曲线如图 6 所示,由图 6 可见,差分曲线开始和结束的地方会出现比较大的跳跃.如果我们依然采用上述估计最大值的方法则无法实现对噪声和数据的有效分离.为了克服这一问题,我们选择数据集密度排序 5%~95%之间的部分来寻找最大的差分值以估计密度阈值.

因此,本节所提基于密度差分的噪声检测算法的具体步骤即可修正为如算法 1 所示.

算法 1. 基于密度差分的噪声检测算法.

(1) 根据式(2)计算样本密度并进行排序,使得 $F(x_i) < F(x_{i+1})$;

(2) 对排序后密度曲线进行差分计算,即计算 $F(x_{i+1}) - F(x_i)$;

(3) 对于密度位于 5%~95%的样本找到差分最大值对应的密度阈值 $T = \max(F(x_{i+1}) - F(x_i))$;

(4) 根据密度阈值检测噪声点,输出有用数据点.

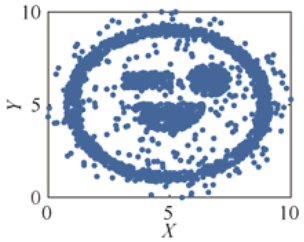


Fig.5 Dataset $D2$

图 5 数据集 $D2$

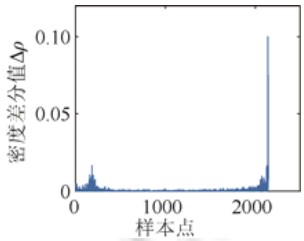


Fig.6 Density difference curve

图 6 密度差分曲线

2 基于密度差分的自动聚类算法

本节提出了一种基于密度差分的自动聚类算法,实现了对含噪数据集的聚类,算法流程如图 7 所示.首先对输入数据集去除噪声,然后构建每个点的邻域,反向近邻点的个数代表每个点的密集程度,最后,从最密集的点开始出发寻找近邻构建类簇,从而实现对有用数据的分类.

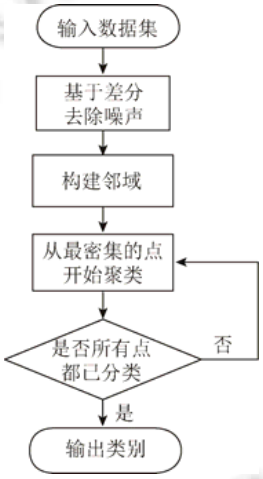


Fig.7 The overall flow chart of CDD algorithm

图 7 CDD 算法整体流程图

定义 4(反向近邻数). 对于给定数据集, $Nb(i)$ 表示算法进行过程中数据点 i 作为其他点近邻的次数,即如果存在: $i \in NN_k(p), p \in D$ 且 $p \neq i$, 则 $Nb(i) = Nb(i) + 1, k = 1,2,3,...,n$, 这里,$k$ 有截止条件.

由反向近邻数的定义可见,对于一个数据集中的数据点,其 Nb 值越大,则说明该点被更多的点包围,直观上它看起来像数据集的质心.因此,我们选择从反向近邻数最大的数据点开始来构建连通区域,从而保证没有移除干净的 $Nb(i) = 0$ 的离散点最后被计算.CDD 算法的具体步骤如算法 2 所示.首先,初始化反向近邻数 $Nb(i)$ 和邻域 $NN_r(i)$. 从 $r = 1$ 开始,依次计算每个点的第 r 近邻,假设计算点 i 的第 r 近邻为 j,则 $Nb(j)$ 加 1,把点 j 合并到 $NN_r(i)$. 当所有点计算完第 r 近邻后,统计此时没有反向近邻数的点,即 $Nb(i) = 0$ 的点的个数 Num.如果从 $r-1$ 近邻到 r 近邻 Num 值不变,则进一步计算每个点的 $r+1$ 近邻.如果 Num 值仍然不变,则保存 $Nb(i)$ 和 $NN_r(i)$ 退出循

环,否则,r 加 1 继续进行计算.在循环结束后,由于我们得到的 $NN_r(i)$ 个数可能太大,以至于不好分开两个靠近的类别簇,这里,我们取前 2/3 个邻域点作为新的邻域进行扩张.在对每个点分类之前,首先判定 $NN_r(i)$ 邻域内的点是否已被分类,如果存在已经被分类的点,那么选取邻域内最近的被分类的点的标签赋给点 i;如果不存在,则把当前标签 Lay 赋值给点 i.对点 i 分类完成后,将点 $Nb(i)$ 赋值为–1,因为每次要寻找 $Nb(i)$ 最大的点,所以保证一些 $Nb(i)=0$ 的离散点也能被找到.之后把点 i 的 $NN_r(i)$ 邻域放在一个队列,依次计算队列内未被分类的点,如果第 1 个点未被分类,则按照上面给点 i 分类的方法进行分类,并将第 1 个点的未被分类的邻域内的点放在队列中,移除第 1 个点.直到队列内的点全部分类完毕,即队列清空,表示一个类簇形成,类别数 Lay 加 1.

算法 2. CDD 算法.

输入:数据集 X;

输出:C:类别标签.

主要步骤:

1. 初始化: $r=1, Nb(i)=0, NN_r(i)=\varnothing$;
2. 分别对 X 中的每个点 i 计算它的第 r 近邻 j;
   a. $Nb(j)=Nb(j)+1$
   b. $NN_r(i)=NN_r(i)\cup\{j\}$
3. 计算 $Nb(i)=0$ 的点的数量 Num;
4. 如果 Num 的值不变:
   $r=r+1$, 重新计算第 2 步和第 3 步;
   如果 Num 的值仍然不变:
   进入步骤 5;
   否则:
   进入步骤 2;
   否则:
   $r=r+1$, 进入步骤 2;
5. 保存并输出 $NN_r(i), Nb(i)$;
6. 取前 $2\times|NN_r(i)|/3$ 个点作为新的 $NN_r(i)$;
7. **While**(**1**):
8. 找到 $Nb(i)$ 最大的点 x;
9. 如果 $y\in NN_r(x)$ 且 $Class(y)\neq 0$:
10. 那么把距 x 最近的 y 的类标签赋给 x;
11. 否则:
12. $Class(x)=Lay$;
13. 把 $NN_r(i)$ 内的点放进一个空队列;
14. **While**(队列非空):
15. 选择队列第 1 个点 x_p;
16. 如果 $Nb(x_p)\neq -1$ (未分类):
17. 如果 $y'\in NN_r(x_p)$ 且 $Class(y')\neq 0$:
18. 那么把距 x_p 最近的 y'的类标签赋给 x_p;
19. 否则:
20. $Class(x_p)=Lay$;
21. $Nb(x_p)=-1$;
22. 把 $NN_r(x_p)$ 内未分类的点插入到队列中;

```
        移除队列内的第 1 个点;
    end While
    Lay = Lay + 1;
    如果所有点 Nb(i) = −1 成立:
        那么 Break;
end While
```

3 实验结果及分析

为了验证本文所提 CDD 算法的有效性,我们进行了仿真实验.实验中,采用 4 组带噪声的人工数据集,对算法的性能进行比较.实验中,我们将 IS-DBSCAN、DBSCAN 和本文所提算法 CDD 进行了对比.实验结果表明,本文所提 CDD 算法可以有效地确定数据集的类别数,实现对含有噪声数据集的有效分类.实验中所采用的数据集如图 8 所示,分别为 *D*2、*D*3、*D*4 和 *D*5.实验对比结果如图 9 所示.

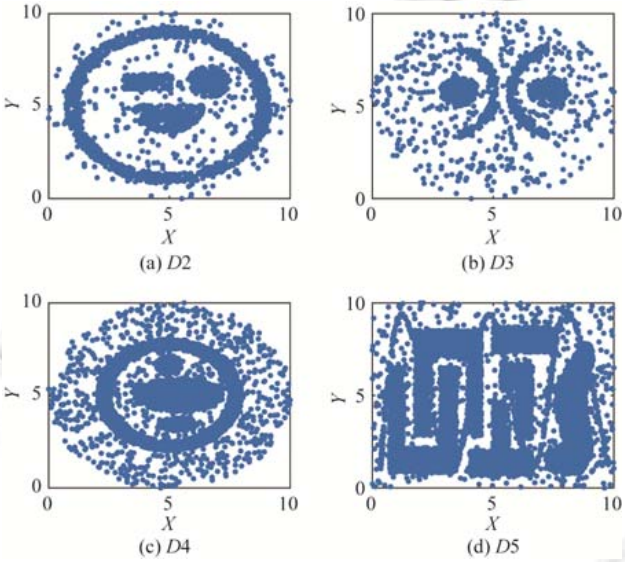


Fig.8 Four groups of datasets used in the experiments

图 8 实验所使用的 4 组数据集

由图 9 所示结果可以看出:对于数据集 *D*2,3 种算法都能检测出离散点正确地分为 5 类.对于数据集 *D*3,由于 IS-DBSCAN 算法去除噪声的效果不好,很多噪声没有去除,最终把数据分为 9 类,原因是把其中一些离散点当作了有用点;DBSCAN 算法和 CDD 算法能够检测离散点,正确地分为了 5 类.对于数据集 *D*4,IS-DBSCAN 分成了 9 类,除了正确地分出 5 类数据点外,还包括 4 个错把噪声当成有用点的类簇;DBSCAN 算法和 CDD 算法正确地分成了 5 类.对于数据集 *D*5,IS-DBSCAN 算法最终分成了 11 类,DBSCAN 算法结果分成了 8 类,只有 CDD 算法正确地分为了 7 类,它不会把靠近有用点的噪声点单独分为一类.综上分析,对比使用两个参数的 DBSCAN 算法和使用一个参数的 IS-DBSCAN 算法,CDD 算法在只使用一个参数的情况下,能够实现噪声的检测和得出正确的类别数,而不会出现把一些未被检测到的离散点单独归为一类的情况.

为了进一步验证所提算法的有效性,我们对算法所得聚类质量作进一步对比.这里,我们采用平均聚类纯度[19]对算法性能进行定量分析,其定义如下:

$$pur = \frac{\sum_{i=1}^{K} \frac{|C_i^d|}{|C_i|}}{K} \times 100\% \tag{3}$$

其中, K 表示聚类的数量, $|C_i^d|$ 表示在类别 i 中占主导地位类别标签的点的数量. $|C_i|$ 表示类别 i 中点的数量.这里对于少数离散点单独分为一类的情况,将 $|C_i^d|$ 视为 0.直观来看,纯度值越大,则表示类别个数分类准确且聚类质量越高;纯度值越小,则表示类别个数越多且聚类质量越差.

3 种算法对 4 组数据集纯度计算结果见表 1.由表 1 所示结果可以看出,对于数据集 $D2$,3 种算法的 pur 相同;对于数据集 $D3$、$D4$ 和 $D5$,可以看出本文算法的优势,CDD 算法的 pur 值高于其他算法.相对于 DBSCAN 算法使用两个参数,CDD 算法仅使用一个参数.

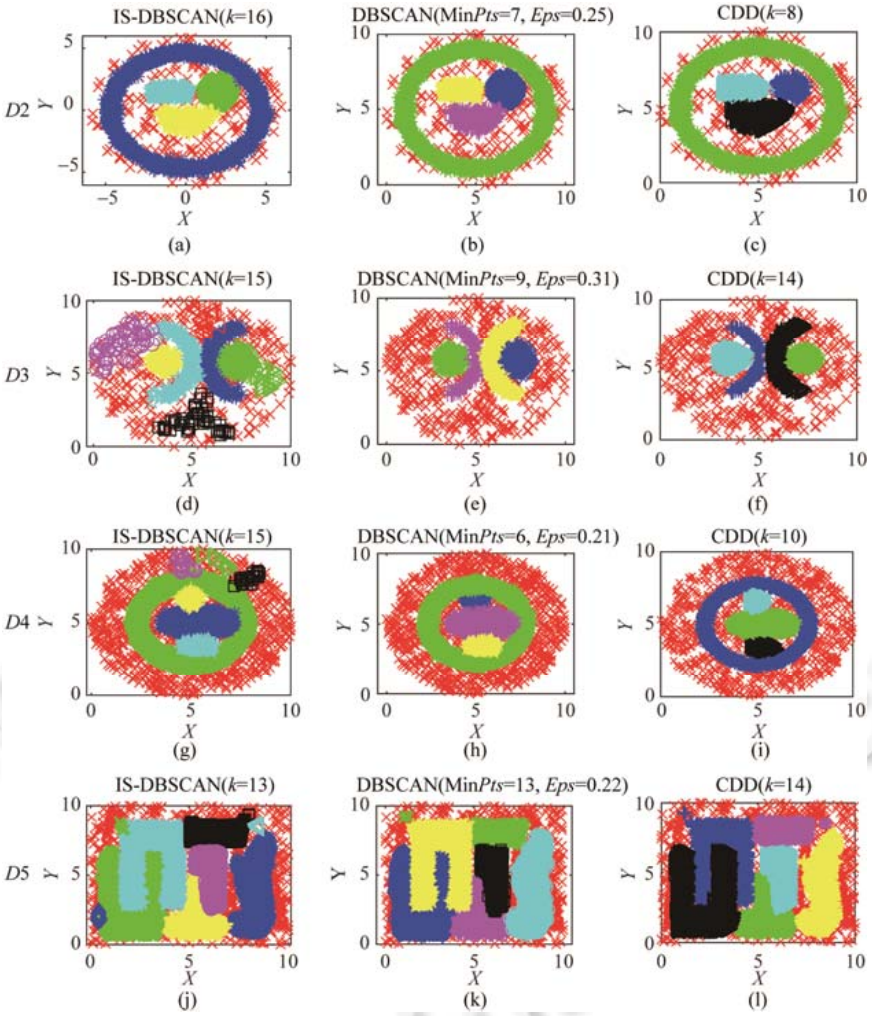


Fig.9 Clustering comparison results by IS-DBSCAN, DBSCAN and CDD for four datasets. Clusters arerepresented by different colors

图 9 IS-DBSCAN、DBSCAN 和 CDD 3 种算法在 4 组数据集上的对比结果,不同颜色代表不同类别

Table 1 Purity calculation results

表 1 纯度计算结果

数据集	IS-DBSCAN	DBSCAN	CDD
$D2$	96%(5 类)	96%(5 类)	96%(5 类)
$D3$	49%(9 类)	87%(5 类)	91%(5 类)
$D4$	51%(9 类)	92%(5 类)	96%(5 类)
$D5$	64%(11 类)	85%(8 类)	99%(7 类)

为了说明算法中一些参数的经验取值问题和 k 值敏感性问题,我们进行了实验对比分析.

(1) 针对噪声检测部分的取值 5%~95%,分别取全部点、5%~95%及 10%~90%这 3 种情况对 $D2$、$D3$、$D5$ 进行噪声检测,实验对比结果如图 10 所示,三角形区域所示为检测的噪声点,*符号区域所示为有用数据,这里,k 默认为 10.从对比结果可以看出,如果我们不设定范围,从全部点中寻找阈值,那么效果非常糟糕,几乎所有的点都判定成了噪声;从设定 5%~95%、10%~90%两个范围来看,可以很好地分离出噪声和有用点,对噪声的检测具有鲁棒性.

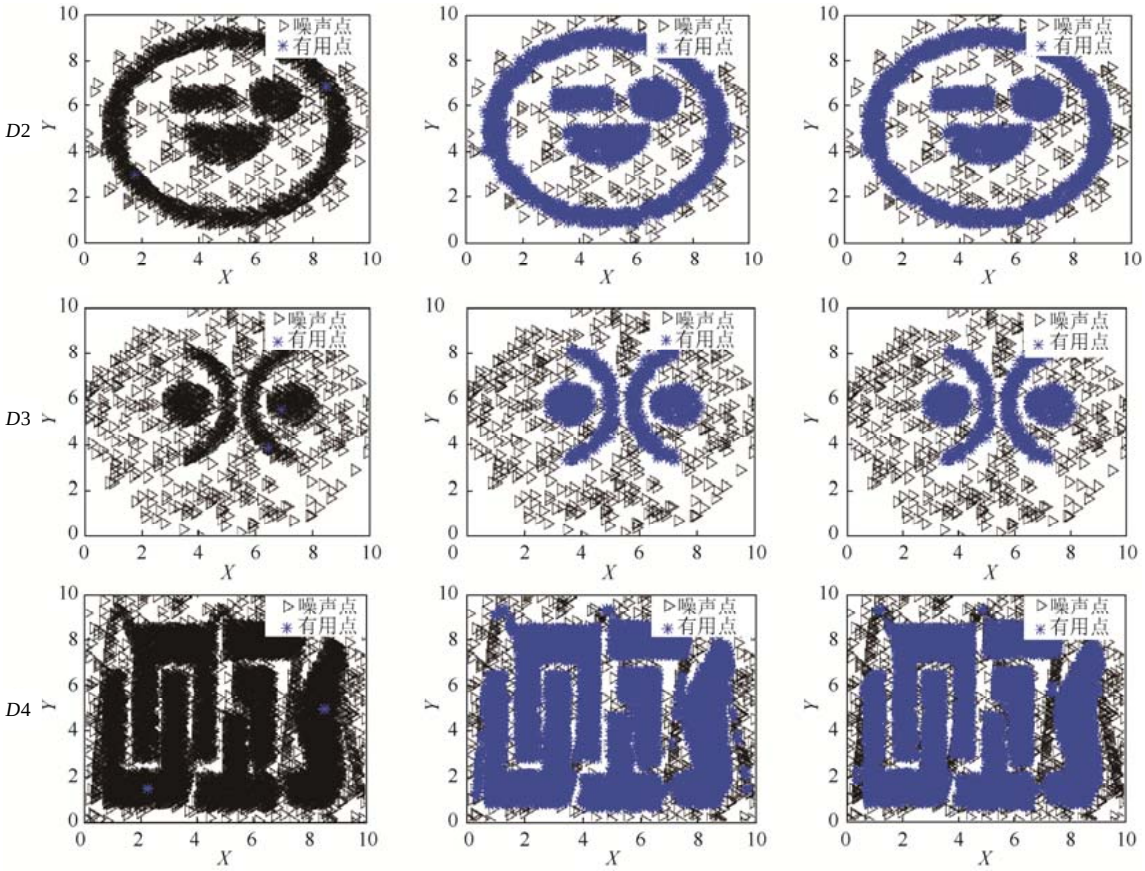


Fig.10 Noise detection results of different data range.

(a) column: All points, (b) column: 5%~95%, (c) column: 10%~90%

图 10 不同数据点范围检测噪声结果,(a)列:全部点,(b)列:5%~95%,(c)列:10%~90%

(2) 针对自动分类过程中,取前 2/3 个邻域点作为每个点新邻域问题,我们取 $D2$、$D3$、$D4$、$D5$ 进行实验对比分析,分别设置邻域系数为 1/2、2/3、5/6 和 1.不同邻域系数下得到的最终聚类数目见表 2,从表中可以看出,在不同邻域系数下,聚类数目变化不够明显.聚类结果的 pur 值如图 11 所示.从结果可以看出:当邻域系数为 1/2 时,$D3$ 和 $D4$ 数据集分类效果不好;当邻域系数为 1 时,相对比较密集的 $D5$ 数据集分类效果不好;邻域系数为 2/3 和 5/6 时都能够得出正确的类别数.这里,我们选取 2/3 作为系数是因为对于一些比较密集的数据集效果会更好.

(3) 对参数 k 敏感性的分析,我们对数据集 $D2$、$D3$、$D4$ 和 $D5$ 分别从 k=6 到 k=20 之间间隔为 2 取值进行对比实验.表 3 是聚类数目随着 k 值变化的结果,可以看出,$D2$、$D3$ 和 $D4$ 数据集在一定区间内是稳定的,$D5$ 数据集则波动较大.pur 值的计算结果如图 12 所示.从结果可以看出,虽然 k 变化很大,但对于数据集 $D2$ 没有影响,数据集 $D3$ 和 $D4$ 相对波动较小,而相对比较密集的 $D5$ 数据集波动较大,但只要设置正确的 k 值,也能分类出

准确的结果.

Table 2 Comparison of clustering numbers of different neighborhood coefficients

表 2 不同邻域系数聚类数目对比

数据集	1/2	2/3	5/6	1
*D*2	5	5	5	5
*D*3	5	5	5	5
*D*4	4	5	5	5
*D*5	7	7	7	6

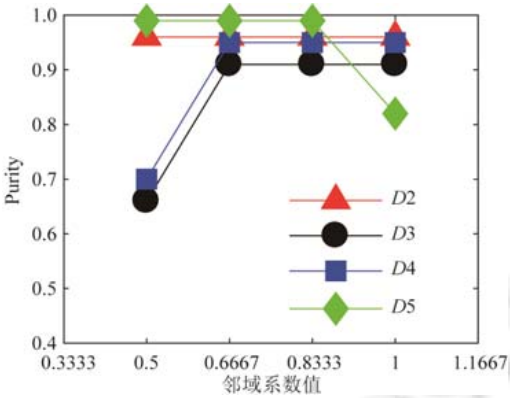


Fig.11 The clustering results obtained with different neighborhood coefficients

图 11 不同邻域系数聚类变化情况

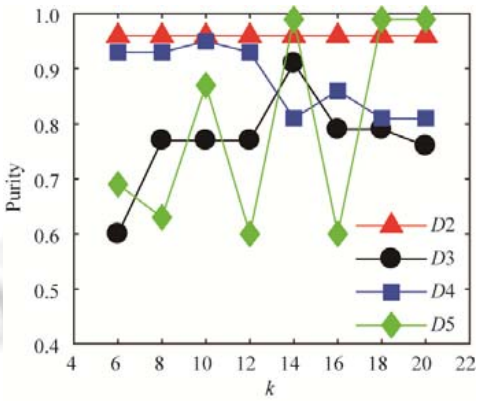


Fig.12 Clustering results obtained with different *k* values

图 12 不同 *k* 值聚类变化情况

Table 3 Comparison of clustering numbers of different *k* values

表 3 不同 *k* 值聚类数目对比

数据集	6	8	10	12	14	16	18	20
*D*2	5	5	5	5	5	5	5	5
*D*3	8	6	6	6	5	4	4	6
*D*4	5	5	5	5	2	3	2	2
*D*5	11	7	8	5	7	5	7	7

4 总结与展望

本文提出了一种基于密度差分的自动聚类算法,算法根据噪声数据与有效数据密度分布的差异,提出了基于密度差分的噪声点检测,并通过构建邻域利用近邻的方法,进一步实现了对有用数据类别的自动划分.所提 CDD 算法在实现聚类的过程中仅需要输入近邻参数 *k*,无需提前设定类别数,即可实现对数据集类别数和类别的自动划分,因此,在实际中具有更广泛的应用前景.实验验证了所提算法的有效性.未来工作将致力构造更合理的密度函数,实现自动确定近邻参数.

References:

[1] Kaufman L, Rousseeuw PJ. Finding groups in data: An introduction to cluster analysis. DBLP, 1990. http://dblp.org/rec/books/wi/KaufmanR90.html [doi: 10.1002/9780470316801]

[2] Han J, Kamber M. Data Mining: Concepts and Techniques. Burlington: Morgan Kaufmann Publishers, 2001.

[3] Jain AK, Murty MN, Flynn PJ. Data clustering: A review. ACM Computing Surveys, 1999,31:264–323. [doi: 10.1145/331499.331504]

[4] Kriegel HP, Zimek A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans. on Knowledge Discovery from Data, 2009,3(1):1–58. [doi: 10.1145/1497577.1497578]

[5] Sun JG, Liu J, Zhao LY. Clustering algorithms research. Ruan Jian Xue Bao/Journal of Software, 2008,19(1):48–61 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048]

[6] Ng RT, Han J. CLARANS: A method for clustering objects for spatial data mining. IEEE Trans. on Knowledge & Data Engineering, 2002,14(5):1003–1016. [doi: 10.1109/TKDE.2002.1033770]

[7] Liang J, Bai L, Cao F. *K*-Modes clustering algorithm based on a new distance measure. Journal of Computer Research and Development, 2010,47(10):1749–1755 (in Chinese with English abstract). http://crad.ict.ac.cn/EN/Y2010/V47/I10/1749.html

[8] Wang W, Yang J, Muntz R. Sting: A statistical information grid approach to spatial data mining. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA, eds. Proc. of the 23rd Int'l Conf. on Very Large Data Bases. San Francisco: Morgan Kaufman Publishers, 1997. 186–195. http://www.vldb.org/conf/1997/P186.PDF

[9] Ankerst M, Breunig MM, Kriegel HP, Sander J. OPTICS: Ordering points to identify the clustering structure. ACM SIGMOD Record, 1999,28(2):49–60. [doi: 10.1145/304182.304187]

[10] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis E, Han JW, Fayyad U, eds. Proc. of the KDD, Vol. 96. AAAI Press, 1996. 226–231. https://aaai.org/Library/KDD/1996/kdd96-037.php

[11] Hinneburg A, Keim DA. An efficient approach to clustering in large multimedia databases with noise. In: Agrawal R, Stolorz P, eds. Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. New York: AAAI Press, 1998. 58–65. http://www.ub.uni-konstanz.de/kops/volltexte/2008/7049/

[12] Ester M, Kriegel HP, Xu X. Density-Based clustering in spatial databases: The algorithm GDBSCAN and its applications. Data Mining & Knowledge Discovery, 1998,2(2):169–194. [doi: 10.1023/A:1009745219419]

[13] MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability, Vol. 1. Berkeley: University of California Press, 1967. 281–297. https://projecteuclid.org/euclid.bsmsp/1200512992

[14] Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. ACM, 1996. 103–114. [doi: 10.1145/235968.233324]

[15] Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases. ACM SIGMOD Record, 1998,26(1):35–58. [doi: 10.1145/276305.276312]

[16] Karypis G, Han EH, Kumar V. Chameleon: Hierarchical clustering using dynamic modeling. Computer, 2002,32(8):68–75. [doi: 10.1109/2.781637]

[17] Cassisi C, Ferro A, Giugno R, Pigola G, Pulvirenti A. Enhancing density-based clustering: Parameter reduction and outlier detection. Information Systems, 2013,38(3):317–330. [doi: 10.1016/j.is.2012.09.001]

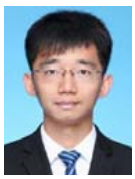
[18] Yang L, Zhu Q, Huang J, Cheng D. Adaptive edited natural neighbor algorithm. Neurocomputing, 2017,230:427–433. [doi: 10.1016/j.neucom.2016.12.040]

[19] Cao F, Ester M, Qian W, Zhou A. Density-Based clustering over an evolving data stream with noise. In: Proc. of the SIAM Int'l Conf. on Data Mining. Bethesda, 2006. 328–339. [doi: 10.1137/1.9781611972764.29]

附中文参考文献:

[5] 孙吉贵,刘杰,赵连宇.聚类算法研究.软件学报,2008,19(1):48–61. http://www.jos.org.cn/1000-9825/19/48.htm [doi: 10.3724/SP.J.1001.2008.00048].

[7] 梁吉业,白亮,曹付元.基于新的距离度量的 *K*-Modes 聚类算法.计算机研究与发展,2010,47(10):1749–1755.



陈朝威(1994－),男,山东菏泽人,硕士生,主要研究领域为聚类算法,深度学习.

常冬霞(1977－),女,博士,副教授,CCF 专业会员,主要研究领域为模式识别,图像处理.