基于区域划分的 DBSCAN 多密度聚类算法*

韩利钊，钱雪忠，罗　靖，宋　威
（江南大学 物联网技术应用教育部工程研究中心，江苏 无锡 214122）

摘　要：DBSCAN 聚类算法使用固定的 Eps 和 minPts，处理多密度的数据效果不理想，并且算法的时间复杂度为 $O(N^2)$。针对以上问题，提出一种基于区域划分的 DBSCAN 多密度聚类算法。算法利用网格相对密度差把数据空间划分成密度不同的区域，每个区域的 Eps 根据该区域的密度计算自动获得，并利用 DBSCAN 算法进行聚类，提升了 DBSCAN 的精度；避免了 DBSCAN 在查找密度相连时需要遍历所有数据的不足，从而改善了算法效率。实验表明算法能有效地对多密度数据进行聚类，对各种数据的适应力较强，效率较优。

关键词：区域划分；多密度；相对密度差；DBSCAN 聚类
中图分类号：TP301.6　　**文献标志码**：A　　**文章编号**：1001-3695(2018)06-1668-04
doi:10.3969/j.issn.1001-3695.2018.06.015

Multi-density clustering algorithm DBSCAN based on region division

Han Lizhao, Qian Xuezhong, Luo Jing, Song Wei
(*Engineering Research Center of Internet of Things Technology Applications for Ministry of Education, Jiangnan University, Wuxi Jiangsu* 214122, *China*)

Abstract: Because of the fixed Eps and minPts, DBSCAN clustering algorithm is not ideal for multi-density data, and its time complexity is $O(N^2)$. Aiming at the above problems, this paper proposed a multi-density clustering algorithm DBSCAN based on region division. This algorithm used the relative grid density difference to divide the spatial data into different density regions, then generated different Eps automatically according to the different density of each region, and used DBSCAN algorithm to improve the accuracy. This idea kept DBSCAN from traversing of all data when it searched for density connected region. So it also improved the algorithm efficiency. Experiments show that the algorithm can effectively cluster the multi-density data. It has a better adaptability to various kinds of data and better efficiency.

Key words: region division; multi-density; relative density difference; DBSCAN clustering

0　引言

DBSCAN(density-based spatial clustering of application with noise)[1]是经典的基于密度的聚类算法，该算法能在含有噪声的数据中识别出任意形状的簇，具有较好的聚类效果。由此引起了国内外大量学者对其进行研究和改进。根据笔者的调研发现，对其研究主要分为两方面的内容。

一方面，DBSCAN 聚类算法需要用户在没有先验知识的情况下输入 Eps 和 minPts 两个参数，参数对聚类结果影响很大，特别是在密度分布不均匀的多密度数据中，聚类效果不理想。首先针对参数敏感提出的算法有 OPTICS(ordering points to identify the clustering structure)[2]，该算法通过产生簇排序和记录簇连接信息屏蔽参数敏感，但对多密度聚类仍无能为力。针对多密度数据改进的算法如文献[3]所述，该算法的核心思想为：任意选取一数据 p 和最初的较小半径 ε，如果 $N_\varepsilon(p) <$ minPts($N_\varepsilon(p)$为数据 p 的 ε 邻域内数据个数)，则 $\varepsilon = \varepsilon + \Delta\varepsilon$，直到 $N_\varepsilon(p) \geqslant$minPts，这样就可以根据不同数据区域的密度得到不同的 ε 值，能够有效地处理多密度数据，但 ε 每增加一个 $\Delta\varepsilon$ 就要遍历数据集，大大影响了算法的效率。受文献[3]的启发，文献[4]提出了一种贪心的 DBSCAN 改进算法，贪心策略的实质是通过问题的一系列局部最优解来逼近整体最优解。该算法虽然用邻域查询方法减少盲目随机寻找核对象的耗时操作，但算法效率仍然高于 DBSCAN 算法，且对于类中心密、逐渐向边缘变稀的数据处理不理想。

另一方面，DBSCAN 时间复杂度为 $O(N^2)$，限制其处理大规模数据的能力。针对此问题，周水庚等人[5]提出了一种基于密度的快速聚类算法，该算法通过选用核心对象邻域里的个别代表对象作为种子点进行类扩展，减少了区域查询的次数，从而降低了聚类时间。文献[6]对文献[5]进行了改进，减少丢失点的情况和代表点不是核心点带来的问题。传统的网格聚类算法，算法效率高，现有的算法有 STING、CLIQUE、SNN、ECRGDD[7~9]等。所以有学者提出把网格划分的思想用到 DBSCAN 聚类中，如刘淑芬等人[10]提出的基于网格单元的 DBSCAN 算法，冯玲等人[11]提出的一种基于网格查询的改进 DBSCAN 算法等。

本文提出基于区域划分的 DBSCAN 多密度聚类算法。通过把数据空间划分成密度不同的区域，避免在密度相连查找时对所有数据的遍历，以此来提高效率；然后根据不同区域自适

收稿日期：2017-01-16；**修回日期**：2017-02-28　　**基金项目**：中央高校基础研究资助项目(JUSRP51510,JUSRP51635B)

作者简介：韩利钊(1990-)，男，河北邯郸人，硕士，主要研究方向为数据挖掘(han_lizhao@163.com)；钱雪忠(1967-)，男，江苏无锡人，副教授，硕导，主要研究方向为数据挖掘、数据库技术、网络安全等；罗靖(1991-)，男，湖北荆州人，硕士，主要研究方向为数据挖掘；宋威(1981-)，男，湖北恩施人，副教授，博士，主要研究方向为数据挖掘、人工智能和模式.

应地生成适合本区域的 Eps 参数进行 DBSCAN 聚类,更好地处理多密度数据。

1　DBSCAN 算法

DBSCAN 算法是一种经典的基于密度的聚类算法,该算法计算每个数据对象的 Eps 邻域,通过密度可达数据对象聚成一个类簇来得到聚类结果。DBSCAN 算法可以自动确定类簇的个数,发现任意形状的类簇,且对噪声数据不敏感。给定一个 d 维数据集 $D(i=1,2,\cdots,d)$,DBSCAN 中的定义如下:

定义 1　数据对象 p 的 Eps 邻域。数据对象 $\forall p \in D$ 的 Eps 邻域 $N_{\text{Eps}}(p)$ 定义为以 p 为核心,Eps 为半径的 d 维超球体区域内包含的点的集合,即 $N_{\text{Eps}}(p)=\{q \in D \mid \text{dist}(p,q) \leqslant \text{Eps}\}$,其中 D 为 d 维空间上的数据集,$\text{dis}(p,q)$ 表示 D 中点 p 和 q 之间的距离。

定义 2　核心数据对象。给定参数 Eps 和 minPts,对于数据对象 p,如果 p 的 Eps 邻域包含的对象个数满足 $|N_{\text{Eps}}(p)| \geqslant \text{minPts}$,则称 p 为核心对象。

定义 3　直接密度可达。给定 Eps 和 minPts,对于数据对象 $p,q \in D$,如果 p 满足 $p \in N_{\text{Eps}}(q)$ 且 $|N_{\text{Eps}}(q)| \geqslant \text{minPts}$ 这两个条件,则称 p 是从 q 关于 Eps、minPts 直接密度可达的。此外,直接密度可达不满足对称性。

定义 4　密度可达。给定 Eps 和 minPts,对于数据对象 q,$p \in D$,如果存在对象序列 $p_1,p_2,\cdots,p_n \in D$,其中 $p_1=q,p_n=p$,并且 p_{i+1} 是从 p_i 直接密度可达的,则称 p 是从 q 关于 Eps、minPts 密度可达的。此外,密度可达也不满足对称性。

定义 5　密度相连。给定 Eps 和 minPts,对于数据对象 p,$q \in D$,如果存在一个数据 o 使得 p 和 q 都是从 o 密度可达的,则称 p 和 q 是关于 Eps、minPts 密度相连的。因此,密度相连满足对称性。

当给定 Eps 和 minPts 时,DBSCAN 算法的简要流程如下:选择任一未划分的数据对象,判断其是否为核心数据对象,若是,则寻找所有与其密度可达的数据对象,将这些数据对象标记为一类;若不是,则进行噪声数据判断,若是噪声点,则对其进行标记,若不是噪声,则不对该对象进行处理。如此重复,直至所有的数据对象都被划分。

2　基于区域划分的 DBSCAN 多密度聚类算法

2.1　网格划分

给定一个 d 维数据集 $D(i=1,2,\cdots,d)$,数据个数为 N,D 的任意维属性 A_i 是有界的,设第 i 维上的值在区间 $Rg_i=[l_i,h_i]$ 中,则 $S=Rg_1 \times Rg_2 \times \cdots \times Rg_d$ 就是 d 维数据空间[12]。将数据空间的每一维分成长度相等、互不相交的区间,从而形成网格单元。这些网格在每维区间上都是左闭右开的。这样将数据空间分割成 $\prod \text{num}_i$ 个等体积的超矩形网格单元(num_i 为数据空间第 i 维上的区间数目)。设定网格边长为

$$\text{length}=a \times \sqrt[d]{\prod_{i=1}^{d}\frac{(h_i-l_i)}{N}} \tag{1}$$

其中:a 是网格控制因子[13],用来控制网格的大小。本文的所有实验均使用 $a=1.5$。依据网格边长,可以计算每一维上的区间数目为

$$\text{num}_i=\lceil (h_i-l_i)/\text{length} \rceil \tag{2}$$

2.2　数据分箱

把数据集中的每个对象映射到相对应的网格中,对于每一个数据对象 $X(x_1,x_2,\cdots,x_d)$,其所对应的网格在每一维上的下标为

$$\text{ind}_i=\lceil (x_i-l_i)/\text{length} \rceil \tag{3}$$

对数据集中的每个对象 X,根据式(3)将其映射到相应的网格 g 中,网格 g 中的对象个数为 $\text{den}(g)$。

2.3　网格合并形成区域

相邻网格定义为:如果网格 g_1 和 g_2 相邻,则 $|\text{ind}_i(g_1)-\text{ind}_i(g_2)| \leqslant 1(i=1,2,\cdots,d)$,且一个网格的相邻网格最多有 3^d-1 个。

利用网格相对密度差[9],即两个网格单元 g_1 和 g_2,它们的密度分别为 $\text{den}(g_1)$ 和 $\text{den}(g_2)$,则 g_2 相对于 g_1 的网格相对密度差定义为

$$\text{rgdd}(g_1,g_2)=\frac{|\text{den}(g_1)-\text{den}(g_2)|}{\text{den}(g_1)} \tag{4}$$

以此作为网格合并的条件。首先选取网格密度最大的网格为初始单元网格 g_0,并根据式(4)依次计算相邻网格 g 与 g_0 的网格相对密度差 $\text{rgdd}(g_0,g)$,若 $\text{rgdd}(g_0,g)<\varepsilon$($\varepsilon$ 为给定参数),则 g_0,g 合并,此时初始单元网格 g_0 就变为合并后的大网格区域,初始单元网格密度 $\text{den}(g_0)=\dfrac{\text{den}(g_0)+\text{den}(g)}{G_{\text{num}}}$,其中 G_{num} 表示已经合并的网格数,也就是说初始网格单元密度是动态的。以此方法继续向外扩展合并网格,直到所有边界网格不满足公式 $\text{rgdd}(g_0,g)<\varepsilon$ 为止,由于边界网格中的某些点可能是该区域的边界点,如图 1 所示。数据点 A、B 为簇 C 的边界点,并不能将其视为噪声丢掉,所以把边界网格也合并到该区域(边界网格不再继续向外扩展),合并后的网格形成一个区域,记此区域的数据集为 D_1。

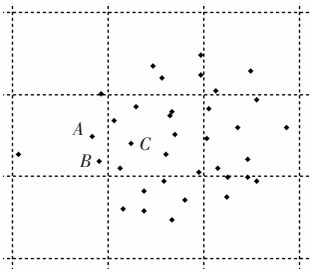


图 1　边界处理

然后在剩余未被处理的网格中取密度最大者作为初始单元网格,重复以上步骤,直到剩下的数据点不能再聚类为止。这样就把数据集 D 分成 $D_1,D_2,\cdots,D_K$ 和初步的噪声点。

2.4　基于区域划分的 DBSCAN 聚类

以上的区域划分方法已经把数据集 D 粗略地分割成了 K 个不同密度的数据区域和噪声点,接下来就是对每一个不同密度区域进行 DBSCAN 聚类。

根据网格的划分方法,每次划分时都是把剩余网格密度最大的网格作为初始网格单元,因此从第一个区域到第 K 个区域密度基本上是递减的。聚类时本文针对第一个区域 D_1 输入 Eps 和 minPts 参数进行 DBSCAN 聚类。D_2 到 D_K 的 Eps 参数通过以下公式自动获得:

$$\text{Eps}_i=\frac{\text{Eps} \times \text{num}_1}{G_1}\Big/\frac{\text{num}_i}{G_i} \qquad i=2,3,\cdots,K \tag{5}$$

其中：num_i 代表 D_i 中数据个数；G_i 代表第 i 个区域合并的网格数。

2.5 算法具体描述

本文首先通过网格划分把数据空间粗略地划分成不同的数据区域，然后对每一区域根据不同密度自动获取适合本区域的 Eps 参数，进行 DBSCAN 聚类。区域划分减少了 DBSCAN 算法中不必要的密度相连的查询操作，提高了效率；根据各自区域的密度不同自动获取 DBSCAN 算法的 Eps 参数，使其对数据的适应力更强，特别是处理多密度数据，效果更优。

算法的主要步骤如下：

输入：数据集 D，Eps，minPts，ε_3。

根据式(1)～(3)进行网格划分和数据装箱，把数据集 D 映射到划分的网格内，并统计每个网格的密度。

根据式(4)和网格密度差参数 ε_3，把密度相近、距离相邻的网格合并。这样就把数据空间划分成不同区域，初步生成数据块 $D_1, D_2, \cdots, D_K$ 和初步的噪声点。

根据参数 Eps、minPts 和式(5)对不同密度的数据区域进行 DBSCAN 聚类。

输出聚类结果。

2.6 时间复杂度分析

输入数据个数为 N，数据空间被划分成 G 个网格，合并网格共形成了 K 个区域，各个区域 $D_i(i=1,2,\cdots,K)$ 的数据个数为 n_i，未被划分到任意区域的数据点数为 n，则有 $n+\sum_{i=1}^{K} n_i = N$。扫描数据集 D，把数据映射到各自网格的时间复杂度为 $O(N)$；网格合并形成区域的时间复杂度为 $O(G^2+N_1)$，其中 N_1 为所有区域数据个数。在各个区域进行 DBSCAN 聚类时间复杂度为 $O(\sum_{i=1}^{K} n_i^2)$。综上，基于区域划分的 DBSCAN 聚类算法的时间复杂度为 $O(N+G^2+N_1+\sum_{i=1}^{K} n_i^2)$，其中 $G << N$，所以最终本文算法的时间复杂度为 $O(N+N_1+\sum_{i=1}^{K} n_i^2)$。大量实验证明当 K 比较大时(如 $K>3$)，本文算法的时间复杂度小于传统的 DBSCAN 时间复杂度 $O(N^2)$，且 K 越大，效率优势越明显。接下来证明当 $K>3$ 时

$$N+N_1+\sum_{i=1}^{K} n_i^2 \leqslant N^2 \tag{6}$$

成立。

证明 令 n_i 为区域 D_i 的数据个数，且 $n_i \geqslant n_{i+1}$，由 $N_1+n=N$ 可知

$$N+N_1+\sum_{i=1}^{K} n_i^2 \leqslant 2N+\sum_{i=1}^{K} n_i^2 \tag{7}$$

由

$$N=n+\sum_{i=1}^{K} n_i \tag{8}$$

可知

$$N^2 \geqslant (\sum_{i=1}^{K} n_i)^2 \tag{9}$$

$$(\sum_{i=1}^{K} n_i)^2 = \sum_{i=1}^{K} n_i^2 + 2(n_1n_2+n_2n_3+\cdots+n_{K-1}n_K) \tag{10}$$

由式(7)～(10)可知，要证明式(6)，可证

$$2N+\sum_{i=1}^{K} n_i^2 \leqslant \sum_{i=1}^{K} n_i^2 + 2(n_1n_2+n_2n_3+\cdots+n_{K-1}n_K)$$

化简可得

$$N \leqslant n_1n_2+n_2n_3+\cdots+n_{K-1}n_K$$

代入式(8)继续化简

$$n_1n_2+n_2n_3+\cdots+n_{K-1}n_K-\sum_{i=1}^{K} n_i - n \geqslant 0 \tag{11}$$

$$n_1n_2+n_2n_3+\cdots+n_{K-1}n_K-\sum_{i=1}^{K} n_i - n =$$
$$n_1(n_2-1)+n_2(n_3-1)+\cdots+n_{K-1}(n_K-1)-n_K-n =$$
$$\sum_{i=1}^{K-2} n_i(n_{i+1}-1)+n_{K-1}(n_K-1)-n_K-n$$

由于 n_i 代表第 i 个区域的数据个数，所以 $\forall n_i \geqslant 4$，故

$$\sum_{i=1}^{K-2} n_i(n_{i+1}-1) \geqslant 3\sum_{i=1}^{K-2} n_i$$

由于 $\forall n_i \geqslant n_{i+1}$，那么

$$n_{K-1}(n_K-1)-n_K \geqslant 0$$

此外由于 n 为噪声点个数，当 $K>3$ 时，$3\sum_{i=1}^{K-2} n_i - n \geqslant 0$，一般成立。

综上所述，式(11)成立，即式(6)成立。证毕。

3 实验结果及分析

本文中所有的算法通过 MATLAB 工具实现并处理实验结果，实验环境是：CPU 为 Intel i3 3.7 GHz；内存为 4 GB；OS 为 Windows 7。

说明：所有实验结果中的“×”为噪声点；为了避免大量的边界点丢失，大量实验表明 ε_3 取 0.65～0.73 较为合理，这样既可以基本上把本区域的边缘数据(可能是边界点，也可能是噪声点)全部合并到本区域，又可避免造成区域连通，本文算法中 $\varepsilon_3=0.7$。

实验 1 数据集 DS1[14] 存在四个类有噪声点，且存在多密度。DBSCAN 算法中 Eps = 1.5，minPts = 10；Greedy DBSCAN 算法中 minPts = 10；ECRGDD 中的两个参数分别为 $\varepsilon_3=0.53$，$u_\lambda=0.1$；本文算法 DBSCAN 同样使用 Eps = 1.5，minPts = 10。聚类结果如图 2 所示。

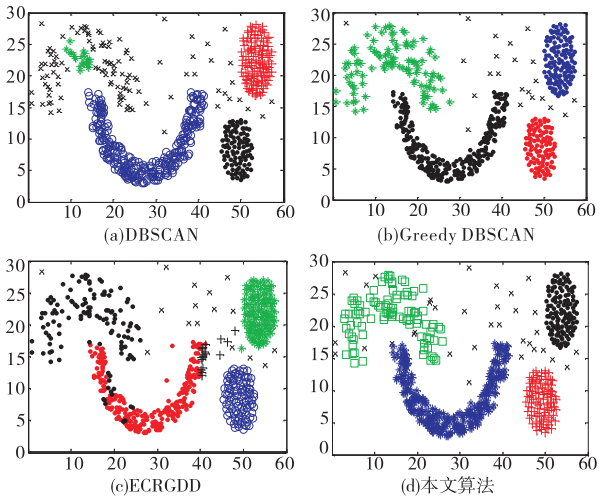


图2 DS1聚类结果

从图 2(a)中可以看出，在保证高密度区域不吸收噪声点时，低密度区域数据丢失严重；反之，保证低密度区域不丢失数据，高密度区域就要吸收噪声点或者左边两个 U 型数据将被合并成一个类。图 2(b)中聚类结果比较理想，能够在多密度数据中识别还有噪声的类，但算法效率有待提高。图 2(c)中对于不规则的图形边界点处理效果欠佳。图 2(d)中本文的聚类算法采用网格划分数据区域的方法，针对不同密度使用不同 Eps 参数进行 DBSCAN 聚类，不仅识别出了四个类，且几乎不吸收噪声点。

实验 2 本实验使用有类边界干扰的数据集 DS2[15]，DBSCAN 算法中半径 Eps = 1，密度阈值 minPts = 5；Greedy DB-

SCAN 算法中 minPts = 9;ECRGDD 算法中的两个参数为 s = 0.53,u_λ = 0.1;本文算法 Eps = 1,minPts = 5。聚类结果如图 3 所示。

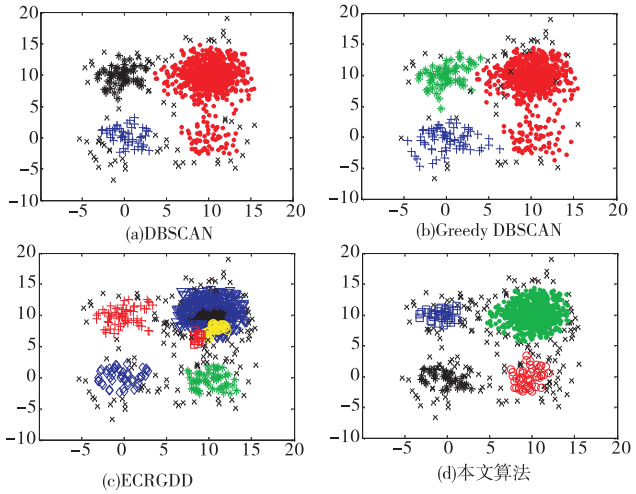


图3　DS2聚类结果

由图 3(a)可以看出,DBSCAN 识别出了三个类,其中密度较小且靠近高密度的类被吸附到高密度区域中,且吸收了较多的噪声点;图 3(b)中右上角的数据中心密度大,沿边缘方向密度逐渐变小,导致 Eps = $\varepsilon + k\Delta\varepsilon$ 较大,适合边缘数据,从而导致把距离右上角较近且密度相近的右下角数据合并成了一个类;图 3(c)中 ECRGDD 算法把高密度区域分成了多个类,对于密度中心到边缘逐渐变小的数据处理也存在欠缺;本文算法采用网格划分,对于不同密度块采取不同参数的 DBSCAN 聚类算法,很好地识别了不同密度的簇,且吸收了较少的噪声点。

实验 3　本实验采用数据量较大、簇较多、形状丰富的数据集 DS3[16],针对 DBSCAN 与本文算法作专项对比。本组实验密度阈值统一为 minPts = 5。数据 3 含有 9 个簇。实验结果如图 4 所示。

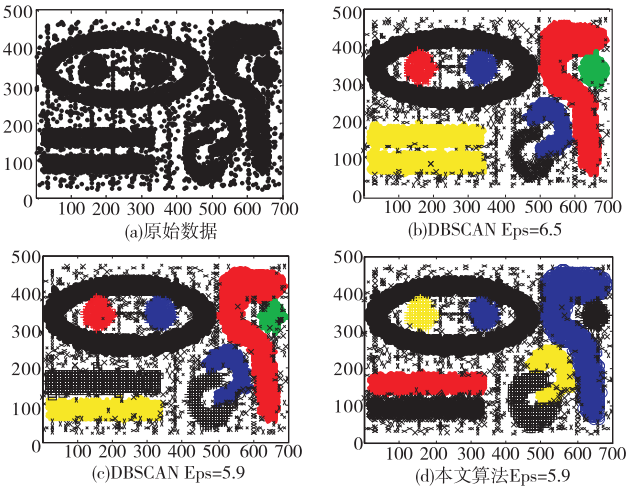


图4　DS3聚类结果

在图 4(b)中,Eps 较大,密度低的簇处理比较理想,但不可避免地比较接近的两个类被合并成一个类。图 4(c)中 Eps 相对较小,密度小的边缘部分被当做噪声点,且类中相对孤立的点被当做噪声点处理。图 4(d)中,最初 Eps = 5.9,但 Eps 不是固定的,会随着区域密度大小根据式(5)自动调整,密度大的 Eps 小,密度小的 Eps 大。这样密度大的区域不至于吸收太多噪声点,密度小的区域不至于被分割或者被当做噪声,很好地处理了多密度数据。

以上实验主要通过实验结果图像显示了本文提出的算法对多密度、任意形状数据聚类的适应性,接下来对上述对比作定量分析。由于 DS2 存在聚类类数不正确的算法,且有临界点干扰,故不作比较。DS1、DS3 原始数据为三维数据,第三维是类标签,聚类结果中的类标签与原始数据的类标签相同,即聚类正确。聚类正确的样本个数记为 num_R,数据总个数记为 num_T,准确度计算公式为

$$\text{准确度} = \frac{\text{num}_R}{\text{num}_T} \times 100\% \tag{12}$$

以下表格中的数据都是通过重复运行 20 次取的平均值。DS1 实验结果数据如表 1 所示。

表 1　DS1 实验结果

算法	时间/ms	准确度/%
DBSCAN	69.2	83.62
Greedy DBSCAN	238.6	100
ECRGDD	15.4	94.75
本文算法	38.6	98.41

DS1 共 628 个数据,四个类,由于类数较小,划分的区域个数 K 较小,本文算法较 DBSCAN 算法效率的提高并不明显。DS3 实验结果数据如表 2 所示。

表 2　DS3 实验结果

算法	时间/ms	准确度/%
DBSCAN	3 613.7	98.23
Greedy DBSCAN	50 892.6	98.85
ECRGDD	260.9	86.5
本文算法	751.7	99.37

DS3 共 10 000 个数据,九个类,本文算法把 DBSCAN 的执行效率提升了近 5 倍。Greedy DBSCAN 算法随着数据的增大,耗时更多。而 ECRGDD 算法虽然执行效率较高,准确率有所欠缺。

4　结束语

以上实验可以验证,基于网格的 DBSCAN 多密度聚类算法,对于多密度数据有较好的聚类效果;且在给定半径 Eps 查找密度相连对象时,只在本区域内查询,避免了遍历所有数据,在一定程度上提高了 DBSCAN 算法效率,且数据集簇越多,效率提升得越明显。当然如果簇较少,甚至只有一个簇,效率并不会有明显提升。本文聚类算法比较适合中低维数据的聚类,随着维数的增加,划分的网格数将以指数级增长,如何避免对于空网格或者密度极小网格的处理将是下一步研究的工作。

参考文献:

[1] Ester M,Kriegel H P,Sander J,*et al*. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Proc of International Conference on Knowledge Discovery and Data Mining. Palo Alto,CA:AAAI Press,1996:226-231.

[2] Ankerst M,Breunig M M,Kriegel H P,*et al*. OPTICS:ordering points to identify the clustering structure[C]//Proc of ACM SIGMOD International Conference on Management of Data. New York:ACM Press,1999:49-60.

[3] Dharni C,Bnasal M. An improvement of DBSCAN algorithm to analyze cluster for large datasets[M]//Innovation and Technology in Education. Piscataway,NJ:IEEE Press,2013:42-46.

[4] 冯振华,钱雪忠,赵娜娜. Greedy DBSCAN:一种针对多密度聚类的 DBSCAN 改进算法[J]. 计算机应用研究,2016,33(9):2693-2696,2700.

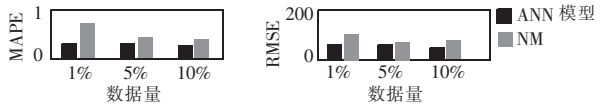
(下转第 1685 页)

同数据分别预测缺失时段的行程时间,并计算两性能指标。表 8 表示在不同数据量的情况下,两模型的 MAPE 值与 RMSE 值的对比结果。表中结果显示,当选取 10% 的数据量作为预测数据时,ANN 模型的 MAPE 误差约为 0.3,而 NM 明显高于 ANN 模型,约为 0.4。同样,ANN 模型的 RMSE 误差约为 54.7,NM 为 75.7,精度大幅低于 ANN 模型。

表 8　ANN 模型与 NM 的行程时间评估性能指标

数据量(百分比)		53(1%)	264(5%)	528(10%)
MAPE	ANN 模型	0.318 5	0.309 3	0.302 9
	NM	0.721 8	0.448 7	0.429 6
RMSE	ANN 模型	58.839 8	57.244 6	54.682 1
	NM	108.313 0	76.707 7	75.744 9

在图 5 中用直方图更加直观地反映了两模型在评估精度上的差异。由图可见,随着数据量的增大,ANN 模型和 NM 的误差值都有降低趋势,而且 ANN 模型的两项误差值均低于 NM,具有更好的预测精度。由于现有数据质量问题,目前只能达到现有精度,但结果显示,ANN 模型的平均绝对百分比误差和均方根误差仍优于 NM。



(a)MAPE 比较　　(b)RMSE 比较

图 5　ANN 模型与 NM 的行程时间评估性能指标直方图

5　结束语

浮动车的运行状态在一定程度上可以反映道路交通拥堵状况,因此,可以利用浮动车的交通信息对路段行程时间进行推断。以往的研究大多基于数据充足的条件下推断行程时间,但是在实时数据缺失的情况下,已有方法很难对数据缺失路段的行程时间进行有效估计,并且多数估计方法通过仿真数据实现,应用价值不高。本研究首次提出利用属性信息、空间信息衡量路段间的相似性,并利用历史交通大数据提取相似路段间的时空关联关系,通过设计的三层神经网络模型对路段行程时间进行推断的方法。实验结果表明,在数据缺失情况下,用该模型对目标路段行程时间进行推断,能达到较好的实验效果,验证了本文方法的有效性。本文现有的模型方法仍受限于交通设施、车道占用等特殊情况的影响,因此后续研究需要综合考虑这些因素,以提高路段行程时间推断的准确度。同时,由于现有数据质量问题,目前只能达到现有精度。随着数据质量的改善,相信未来的研究能达到更高的估计精度,并实现对区域性路网行程时间的估计,拓宽应用范围。

参考文献:

[1] 杨兆升. 关于智能运输系统的关键理论——综合路段行程时间预测的研究[J]. 交通运输工程学报,2001,1(1):65-67,89.

[2] 张发明,朱欣焰,禹维,等. 利用浮动车大数据进行稀疏路段行程时间推断[J]. 武汉大学学报:信息科学版,2017,42(1):56-62.

[3] Zheng Yu, Liu Furui, Hsieh H P, *et al*. U-Air: when urban air quality inference meets big data[C]//Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2013: 1436-1444.

[4] Jula H, Dessouky M, Ioannou P A. Real-time estimation of travel times along the arcs and arrival times at the nodes of dynamic stochastic networks[J]. IEEE Trans on Intelligent Transportation Systems, 2008, 9(1): 97-110.

[5] Jenelius E, Koutsopoulos H N. Travel time estimation for urban road networks using low frequency probe vehicle data[J]. Transportation Research, Part B: Methodological, 2013, 53(4): 64-81.

[6] Zhang Fuzheng, Wilkie D, Zheng Yu, *et al*. Sensing the pulse of urban refueling behavior[C]//Proc of ACM International Joint Conference on Pervasive and Ubiquitous Computing. New York: ACM Press, 2013: 13-22.

[7] 刘春,黄美娴,杨超. 浮动车数据缺失道路的速度推估模型与实现[J]. 同济大学学报:自然科学版,2010,38(8):1255-1260.

[8] 王晓蒙,彭玲,池天河. 基于稀疏浮动车数据的城市路网交通流速度估计[J]. 测绘学报,2016,45(7):866-873.

[9] 袁晶. 大规模轨迹数据的检索、挖掘和应用[D]. 合肥:中国科学技术大学,2012.

[10] 袁冠. 移动对象轨迹数据挖掘方法研究[D]. 徐州:中国矿业大学,2012.

[11] 塔娜,柴彦威,关美宝. 建成环境对北京市郊区居民工作日汽车出行的影响[J]. 地理学报,2015,70(10):1675-1685.

[12] 张玲. POI 的分类标准研究[J]. 测绘通报,2012(10):82-84.

[13] Zhang Faming, Zhu Xinyan, Guo Wei, *et al*. Analyzing urban human mobility patterns through a thematic model at a finer scale [J]. ISPRS International Journal of Geo-Information, 2016, 5(6): DIO: 10.3390/ijgi5060078.

[14] Deza M M, Deza E. Encyclopedia of distances[M]//Encyclopedia of Distances. Berlin: Springer, 2009: 1-583.

[15] Kiremire A R. The application of the Pareto principle in software engineering[EB/OL]. (2011-10-19). http://www2. latech. edu/ ~ box/ase/papers2011/Ankunda_termpaper. PDF.

[16] Zhang Faming, Zhu Xinyan, Hu Tao, *et al*. Urban link travel time prediction based on a gradient boosting method considering spatiotemporal correlations [J]. International Journal of Geo-Information, 2016, 5(11): DOI: 10.3390/ijgi5110201.

[17] Cheng Tao, Wang Jiaqiu, Haworth J, *et al*. A dynamic spatial weight matrix and localized space-time autoregressive integrated moving average for network modeling[J]. Geographical Analysis, 2014, 46(1): 75-97.

(上接第 1671 页)

[5] 周水庚,周傲英,曹晶,等. 一种基于密度的快速聚类算法[J]. 计算机研究与发展,2000,37(11):1287-1292.

[6] 王桂芝,王广亮. 改进的快速 DBSCAN 算法[J]. 计算机应用,2009,29(9):2505-2508.

[7] Zhao Yanchang, Cao Jie, Zhang Chengqi, *et al*. Enhancing grid-density based clustering for high dimensional data[J]. Journal of Systems and Software, 2011, 84(9): 1524-1539.

[8] Karypis G, Han E H, Kumar V. Chameleon: hierarchical clustering using dynamic modeling[J]. Computer, 1999, 32(8): 68-75.

[9] 黄红伟,黄天民. 基于网格相对密度差的扩展聚类算法[J]. 计算机应用研究,2014,31(6):1702-1705.

[10] 刘淑芬,孟冬雪,王晓燕. 基于网格单元的 DBSCAN 算法[J]. 吉林大学学报:工学版,2014,44(4):1135-1139.

[11] 冯玲,刘克剑,唐福喜,等. 一种基于网格查询的改进 DBSCAN 算法[J]. 西华大学学报:自然科学版,2016,35(5):25-29.

[12] 张枫,邱保志. 基于网格的高效 DBSCAN 算法[J]. 计算机工程与应用,2007,43(17):167-169,184.

[13] Piegl L A, Tiller W. Algorithm for finding all k, nearest neighbors[J]. Computer-Aided Design, 2002, 34(2): 167-172.

[14] http://cs. joensuu. fi/sipu/datasets/[EB/OL].

[15] http://www. pudn. com/downloads219/sourcecode/math/detail1030717. html[EB/OL].

[16] Khalid S, Razzaq S. TOBAE: a density-based agglomerative clustering algorithm[J]. Journal of Classification, 2015, 32(2): 241-267.