基于重心点转移的 St-DBSCAN 改进算法

刘 勇,何 婧,姚绍文,向 毅,张 浩
(云南大学 国家示范性软件学院,云南 昆明 650500)

摘 要:在目前已提出的聚类算法中,St-DBSCAN 算法是一种基于密度且性能优越的时空聚类算法。但是当时空点分布出现密度倾斜时,St-DBSCAN 算法会出现聚类时间过长和聚类效果不好的问题。基于此,通过对空间点分布存在的三种数据倾斜,采用数据重心点转移策略,提出了对应的解决方案,以此实现了改进后的 St-DBSCAN 算法。为了验证改进后算法的性能,以昆明市出租车 GPS 数据为实验数据,进行了算法性能对比实验。实验结果表明,改进 St-DBSCAN 算法的时间性能和聚类效果有了一定程度的提升。

关键词:时空聚类算法;St-DBSCAN 算法;转移策略;密度倾斜;重心点
中图分类号:TP302 **文献标识码**:A **文章编号**:1673-629X(2018)11-0006-06
doi:10.3969/j.issn.1673-629X.2018.11.002

An Improved St-DBSCAN Algorithm Based on Center of Gravity Shifting

LIU Yong, HE Jing, YAO Shao-wen, XIANG Yi, ZHANG Hao
(National Pilot School of Software, Yunnan University, Kunming 650500, China)

Abstract: In the presented clustering algorithms, the St-DBSCAN is a spatio-temporal clustering algorithm based on density with superior performance. However, when the spatial distribution is tilted, the St-DBSCAN algorithm may produce too long clustering time and poor clustering effect. Based on the problem, we propose the corresponding solution by using the data center point transfer strategy for three kinds of data skew in spatial point distribution, and then implement the improved St-DBSCAN algorithm. In order to verify the proposed algorithm, the GPS data of taxi of Kunming is used as experimental data for performance comparison, which shows that the improved St-DBSCAN algorithm is improved in time performance and clustering effect.

Key words: spatio-temporal clustering algorithm; St-DBSCAN algorithm; transfer strategy; density dip; center of gravity shifting

0 引 言

聚类(clustering)是数据统计分析的重要组成部分。对实际数据的聚类需要聚类算法根据不同的方法进行聚类分析。迄今为止,已经提出了大量的聚类算法,包括 OPTICS[1]、DENCLUE[2]、DBSCAN[3]、GCHL[4]、BIRCH[5]、K-MEDOIDS[6]和 Clara[7]等,其中 DBSCAN 算法是一种常用的基于密度的聚类算法。

DBSCAN 在生产中的应用十分广泛。例如,利用 DBSCAN 算法对某段时间的居民出行数据进行聚类,能有效预测出当地可能出现的交通堵塞的地点,为公民出行提供良好的建议;利用 DBSAN 算法对台风出现地点进行聚类,能有效地预测出台风的走向。从 DBSACN 算法提出以来,众多学者对其提出了改进,如 IDBSCAN[8]、VDBSCAN[9]、OVDBSCAN[10]、DP-DBSCAN[11]、SA-DBSCAN[12]、Greedy DBSCAN[13]等,这些算法概括来讲主要是从聚类方式、聚类参数、空间点区域划分等方面进行的改进。

Derya Birant[14]在 DBSCAN 算法的基础上,通过增加非空间距离值参数,使 DBSCAN 算法能够对含有非空间距离值的空间数据进行聚类。但是自 St-DBSCAN 算法提出以来,至今尚未有相关文献对其进行改进。但 St-DBSCAN 算法在空间数据分布式出现数据倾斜时,会出现聚类时间过长和聚类效果差的问题。基于上述不足,文中通过求邻近点重心转移的方法对

收稿日期:2017-09-04 **修回日期**:2018-01-10 **网络出版时间**:2018-05-25
基金项目:国家自然科学基金(61363021);云南省教育科学研究基金(2014Y013)
作者简介:刘 勇(1990-),男,硕士研究生,研究方向为数据分析、人工智能;何 婧,讲师,博士,CCF 会员(29926M),通信作者,研究方向为云数据管理、数据挖掘、知识发现;姚绍文,教授,博士,研究方向为网络协议工程、物联网与云计算、语义网技术与应用。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20180525.1550.002.html

其进行改进。改进 St-DBSCAN 算法对空点分布出现密度倾斜时,通过寻找邻近点的重心能更快地找到邻近点分布最多的区域,然后再以重心点为核心点进行聚类,使形成的簇更加紧密,从而提高算法的聚类效果。

1　St-DBSCAN 算法概述

St-DBSCAN 算法中有三个参数: Minpts 、Eps 、Δt 。其中 Minpts 表示形成簇的最小点个数, Eps 表示形成簇的时空点距离, Δt 表示形成簇的最大时间。St-DBSCAN算法的基本思想是:通过循环判断时空核心对象 c 以 Eps 为半径, Δt 时间差内点的个数是否大于等于 Minpts ,如果大于则形成簇,反之则对下一个时空对象进行聚类,直到所有的时空对象都归在某个簇中,或被标记为时空孤立点,则聚类结束。

但是,如果选取的核心点偏离空间点分布的密度集中区域,St-DBSCAN 算法不能很好地解决密度倾斜问题,同时由于 St-DBSCAN 算法需要对所有的空间点进行遍历,导致运行时间较长。

2　改进的 St-DBSCAN 算法设计

改进的 St-DBSCAN 算法通过对空间点分布出现的三种密度倾斜情形,采用求取空间点密度重心的方法获得周围空间点密度分布的重心,然后根据不同的密度倾斜,给出不同的改进方案,对空间点进行聚类。

2.1　相关概念

定义 1(邻近重心点):在以核心点为圆心,半径为 Eps 的圆域内所有邻近点的重心点。

邻近重心点 $P(x,y)$ 的坐标与所有邻近点的分布有关,具体的坐标计算如下:

$$p_x = \frac{\sum_{i=1}^{k} p_{i_x}}{k} \tag{1}$$

$$p_y = \frac{\sum_{i=1}^{k} p_{i_y}}{k} \tag{2}$$

其中, k 表示邻近点个数; p_{i_x} 表示任一邻近点的 x 轴坐标; p_{i_y} 表示任一邻近点的 y 轴坐标。

定义 2(重心点偏离):重心点相对于核心点的偏离程度。如图 1 所示,在点 L_1 为圆心, Eps 为半径的圆域内,由圆心 L_1 与重心点构成的向量 $\overrightarrow{L_1O_1}$ 的长度 $|\overrightarrow{L_1O_1}|$ 与半径所在向量 $\overrightarrow{L_1D_3}$ 的长度 $|\overrightarrow{L_1D_3}|$ 的比值 ∂ ,即为该圆域的重心点偏离。

算法中采用的是欧氏距离,因此重心点 O_1 与核心点 L_1 的距离表示为:

$$d = \sqrt{(p_x - p_{x_0})^2 - (p_y - p_{y_0})^2} \tag{3}$$

其中, p_x 表示邻近点的 x 轴坐标; p_y 表示邻近点的 y 轴坐标。

通过式 4 可以计算重心点偏离 ∂ 。

$$\partial_p = \frac{\sqrt{(p_x - p_{x_0})^2 - (p_y - p_{y_0})^2}}{\text{Eps}} \tag{4}$$

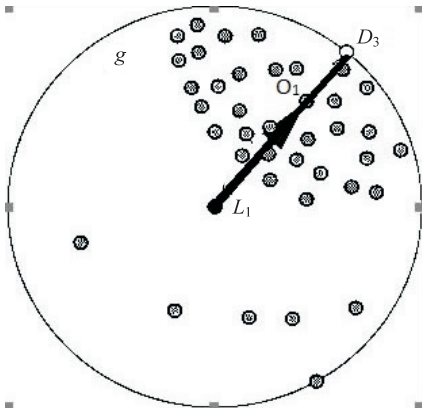

图 1　重心点偏离

定义 3(密度距离差, ΔDensity_distance):在任意的簇中,任意一点 p 的簇内最小距离 desity_distance_min(p) 与该点的簇内最大距离 desity_distance_max(p) 的差值,称为该点的密度距离差,即 ΔDensity _ distance (p),其数学定义为: desity_distance_max(p) = MAX{dist(p,q) | $q \in D \wedge$ dist(p,q) ≤ Eps} ,desity_distance_min(p) = MIN{dist(p,q) | $q \in D \wedge$ dist(p,q) ≤ Eps} 。

对于任意点 p 的 ΔDensity_distance 计算如下:

$$\Delta\text{Density\_distance} = \text{density\_distance\_max}(p) - \text{density\_distance\_min}(p) \tag{5}$$

定义 4(簇聚合度):对于簇 C 密度因子的计算公式如下:

$$\text{density\_factor}(C) = 1/\left[\frac{\Delta\text{Density\_distance}}{|C|}\right] \tag{6}$$

簇聚合度反映了簇的稠密程度。由其定义可知,任意簇 C 的簇聚合度在 (0, + ∞) 的区间内,当簇 C 的簇聚合度越接近 0 时,簇 C 越松散,反之当簇 C 的簇聚合度越大,则簇 C 越紧密。

2.2　算法改进

2.2.1　重心点转移的理论推导

引理 1:邻近重心点与核心点的距离一定在以核心点为圆心, Eps 为半径的圆域内。

假设核心点为 $P(x_0,y_0)$,核心点 p 有 n 个邻近点,邻近点 $\{p_1,p_2,\cdots,p_n\}$ 坐标分别为 (x_0,y_0), (x_1,y_1), $\cdots$, (x_n,y_n) ,点 h 为点 p 的邻近重心点,设点 h 的坐标为 (p_x,p_y) 。

证明:

点 h 为核心点 p 的邻近重心点,故 p 点的 x 轴坐标

为 $p_x = \frac{1}{n}(x_1, x_2, \cdots, x_n)$，$p$ 点的 y 轴坐标为 $p_y = \frac{1}{n}(y_1, y_2, \cdots, y_n)$。

$p_1, p_2, \cdots, p_n$ 是核心点 p 的邻近点，有：

$\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \leqslant \text{Eps}$,

$\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} \leqslant \text{Eps}, \cdots,$

$\sqrt{(x_n - x_0)^2 + (y_n - y_0)^2} \leqslant \text{Eps}$

对上式求和可得：

$\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} + \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} + \cdots + \sqrt{(x_n - x_0)^2 + (y_n - y_0)^2} \leqslant n * \text{Eps}$

邻近重心点与核心点 p 的距离为：

$$d = \sqrt{(p_x - p_0)^2 + (p_y - p_0)^2} = \sqrt{\left[\frac{1}{n}(x_1 + x_2 + \cdots + x_n) - x_0\right]^2 + \left[\frac{1}{n}(y_1 + y_2 + \cdots + y_n) - y_0\right]^2}$$

$$n * d = \sqrt{n^2 * \left[\frac{1}{n}(x_1 + x_2 + \cdots + x_n) - x_0\right]^2 + n^2 * \left[\frac{1}{n}(y_1 + y_2 + \cdots + y_n) - y_0\right]^2} =$$

$$\sqrt{[(x_1 + x_2 + \cdots + x_n) - n * x_0]^2 + [(y_1 + y_2 + \cdots + y_n) - n * y_0]^2} =$$

$$\sqrt{[(x_1 - x_0) + (x_2 - x_0) + \cdots + (x_n - x_0)]^2 + [(y_1 - y_0) + (y_2 - y_0) + \cdots + (y_n - y_0)]^2}$$

因为

$$[\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} + \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} + \cdots + \sqrt{(x_n - x_0)^2 + (y_n - y_0)^2}]^2 - (n * d)^2 =$$

$$2 * \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} * \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} + \cdots + 2 * \sqrt{(x_{n-1} - x_0)^2 + (y_{n-1} - y_0)^2} *$$

$$\sqrt{(x_n - x_0)^2 + (y_n - y_0)^2} = 2 * (x_1 - x_0) * (x_2 - x_0) + \cdots + 2 * (x_{n-1} - x_0) * (x_n - x_0) + 2 * (y_1 - y_0) *$$

$$(y_1 - y_0) + \cdots + 2 * (y_{n-1} - y_0) * (y_n - y_0) = 2 * [\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} * \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} -$$

$$((x_1 - x_0)(x_2 - x_0) + (y_1 - y_0)(y_2 - y_0)) + \cdots + \sqrt{(x_{n-1} - x_0)^2 + (y_{n-1} - y_0)^2} *$$

$$\sqrt{(x_n - x_0)^2 + (y_n - y_0)^2} - ((x_{n-1} - x_0)(x_n - x_0) + (y_{n-1} - y_0)(y_n - y_0))]$$

令：

$p = \sqrt{(x_k - x_0)^2 + (y_k - y_0)^2} * \sqrt{(x_l - x_0)^2 + (y_l - y_0)^2}$，$q = (x_k - x_0)(x_l - x_0) + (y_k - y_0)(y_l - y_0)$，其中，$k, l \in [1, n]$。

$$p^2 - q^2 = (x_k - x_0)^2 (y_l - y_0)^2 + (y_l - y_0)^2 (x_l - x_0)^2 - 2 * (x_k - x_0)(x_l - x_0)(y_k - y_0)(y_l - y_0) =$$

$$[(x_k - x_0)(y_l - y_0) + (y_l - y_0)(x_l - x_0)]^2 \geqslant 0$$

故：

$$(\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} + \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2} + \cdots + \sqrt{(x_n - x_0)^2 + (y_n - y_0)^2})^2 - (n * d)^2 \geqslant 0$$

$$\sqrt{\left[\frac{1}{n}(x_1 + x_2 + \cdots + x_n) - x_0\right]^2 + \left[\frac{1}{n}(y_1 + y_2 + \cdots + y_n) - y_0\right]^2} < \text{Eps}$$

由此得证。

引理 2：当邻近点出现密度倾斜时，邻近重心点一定会偏离聚类点。

证明：由式 1 和式 2 可得，重心点 $G(p_x, p_y)$ 的坐标点的表达式为：

$$p_x = \frac{1}{n}p_{1_x} + \frac{1}{n}p_{2_x} + \cdots + \frac{1}{n}p_{n_x} \tag{7}$$

$$p_y = \frac{1}{n}p_{1_y} + \frac{1}{n}p_{2_y} + \cdots + \frac{1}{n}p_{n_y} \tag{8}$$

当邻近点某个区域的点比较集中时，设该区域的点为 $\{p_1, p_2, \cdots, p_k\}$，区域点的个数为 k，则该区域的重心点表达式为：

$$p_x = \frac{1}{k}p_{1_x} + \frac{1}{k}p_{2_x} + \cdots + \frac{1}{k}p_{k_x}, k \leqslant n \tag{9}$$

$$p_y = \frac{1}{k}p_{1_y} + \frac{1}{k}p_{2_y} + \cdots + \frac{1}{k}p_{k_y}, k \leqslant n \tag{10}$$

由此可得：

$$k * p_x^k = p_{1_x} + p_{2_x} + \cdots + p_{k_x}, k \leqslant n \tag{11}$$

$$k * p_y^k = p_{1_y} + p_{2_y} + \cdots + p_{k_y}, k \leqslant n \tag{12}$$

对于重心点坐标公式，可以整理成：

$$p_x = \frac{1}{n}(p_{1_x} + p_{2_x} + \cdots + p_{k_x}) + \frac{1}{n}p_{k+1_x} + \cdots + \frac{1}{n}p_{n_x}(\{p_1, p_2, \cdots, p_k\} \in P, \{p_{k+1}, p_{k+2}, \cdots, p_k\} \notin P) \tag{13}$$

$$p_y = \frac{1}{n}(p_{1_y} + p_{2_y} + \cdots + p_{k_y}) + \frac{1}{n}p_{k+1_y} + \cdots +$$

$$\frac{1}{n}p_{n_y}(\{p_1,p_2,\cdots,p_k\}\in P,\{p_{k+1},p_{k+2},\cdots,p_k\}\notin P) \quad (14)$$

将式 11、式 12 带入式 13、式 14 可得:

$$p_x=\frac{k}{n}p_x^k+\frac{1}{n}p_{k+1_x}+\cdots+\frac{1}{n}p_{n_x}(\{p_1,p_2,\cdots,p_k\}\in P,\{p_{k+1},p_{k+2},\cdots,p_k\}\notin P) \quad (15)$$

$$p_y=\frac{k}{n}p_y^k+\frac{1}{n}p_{k+1_y}+\cdots+\frac{1}{n}p_{n_y}(\{p_1,p_2,\cdots,p_k\}\in P,\{p_{k+1},p_{k+2},\cdots,p_k\}\notin P) \quad (16)$$

由式 15 和式 16 可知,如果 k 无限接近于 n ,重心点坐标将接近于该区内重心点的坐标,如果该区域内的点都偏离于聚类点时,则邻近点偏离聚类点。因此在密度倾斜时,邻近点的重心将偏离聚类点。

2.2.2　核心点重心密度转移

当空间对象在某一个区域比较集中时,会出现密度倾斜的问题。通过对大量的空间对象分布的分析和总结,得出空间对象出现数据倾斜有核心点密度倾斜、边界点密度倾斜和噪声点密度倾斜三种情形。

1. 密度倾斜出现的情形。

(1)某个空间数据点符合核心点条件,但是该核心点的邻近点普遍集中于偏离核心点的某一个区域(是否普遍集中可以通过邻近点的重心点偏移核心点的程度决定),称这样的密度倾斜为核心点密度倾斜。

(2)某个空间数据点与邻近点符合时间属性条件,但是邻近点的个数小于 Eps,称这样的密度倾斜为边界点密度倾斜。

核心点密度倾斜和边界点密度倾斜统称为同类密度倾斜,因为核心点密度倾斜和边界点密度倾斜中的核心点在时间属性上符合聚类条件,只是在邻近点个数上有所不同。

(3)某个空间数据点与邻近点不符合时间属性条件,称这样的密度倾斜为噪声点密度倾斜。

2. 密度倾斜的解决方案。

(1)密度倾斜的判定。

根据式 4 可知,重心点偏离能反映邻近重心点偏离核心点的程度。如果重心点偏离 ∂ 在区间[0,0.5]内,表示重心点靠拢于核心点。反之如果重心点离心率 ∂_p 在[0.5,1]内,则表示重心点偏离于核心点,出现了密度倾斜的问题。根据引理 1 和引理 2 可知,邻近点重心转移是解决密度倾斜的有效方法。

(2)解决方案。

·核心点密度倾斜解决方案。

图 2 是核心点密度倾斜情形。图中原本的核心点是 A_4,但是核心点 A_4 的邻近点明显向图中的右上角倾斜。对于该问题,可以求邻近点的重心点(图 2 中的 G_3 点),然后再以重心点为聚类点,对空间数据进行聚类。

·边界点密度倾斜解决方案。

图 3 是边界点密度倾斜情形。图中原本的聚类点是 A_5,但是 A_5 的邻近点明显向图中的右下角倾斜。对于该问题,可以求邻近点的重心点(图 3 中的 E_4 点),然后再以重心点 E_4 为聚类点,如果重心点 E_4 的邻近点小于参数 minpt,则将所有的点标记为噪声点。反之,则按照核心点密度倾斜方案对空间数据进行聚类。

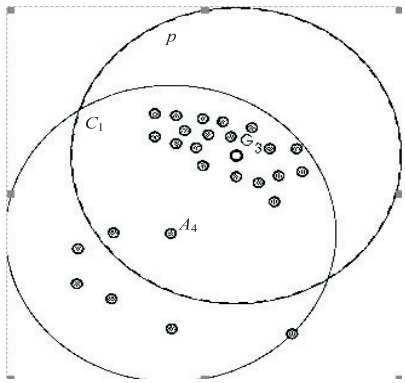

图 2　核心点密度倾斜解决方案

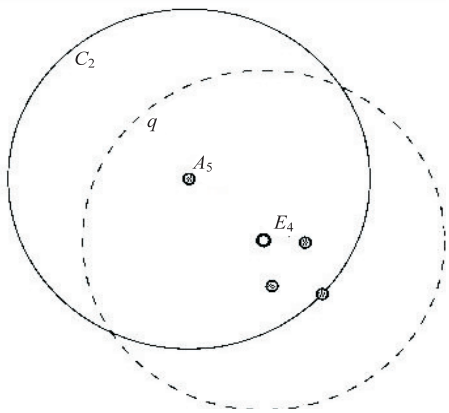

图 3　边界点密度倾斜解决方案

·噪声点密度倾斜解决方案。

图 4 是噪声点倾斜情形。图中原本的核心点是 A_4,但是 A_4 与邻近点明显不属于同一时间维度,不是同一类。对于该问题,将点 A_4 标记为噪声点。

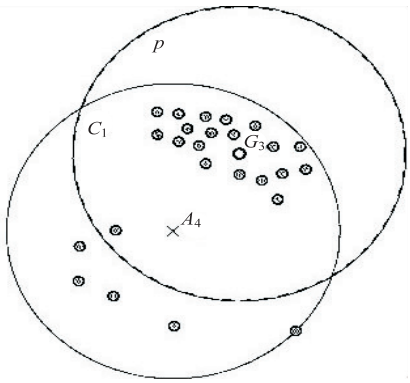

图 4　噪声点密度倾斜解决方案

2.2.3 重心点密度转移的 St-DBSCAN 算法描述

基于上述理论推导和改进方案,改进 St-DBSCAN 算法的具体执行步骤如下:

步骤 1:从任意的空间对象集合 $D = \{p_1, p_2, \cdots, p_n\}$ 取出一点 p,把时空对象当作一个核心对象 o。

步骤 2:在空间数据集中搜寻满足式 17 的空间对象,并计算邻近点的个数 num 是否大于阈值 Minpts。如果大于,执行步骤 4;反之执行步骤 3。

$$\text{timeCondition}(p_i, p_{i+1}) \leqslant \Delta t, \forall p_i, \forall p_{i+1} \in L_m \tag{17}$$

步骤 3:通过式 3 和式 4 计算出重心点位置和重心点偏离空间对象 p 的偏离程度 l,计算出空间对象 p 的邻近点个数 num,根据 num 与 Minpts 和 l 的大小来判断空间对象分布是否出现密度倾斜和出现的是什么类型的密度倾斜,根据不同的数据类型采用不同的密度倾斜解决方案。

步骤 4:如果没有出现密度倾斜且邻近点的个数大于等于 Minpts,则在邻近点中搜寻每个没有被访问的点的邻近点,如果邻近点的邻近点个数小于 Minpts 则标记该点为 border 点。如果该点的邻近点的邻近点个数大于等于 Minpts,则标记该点为核心点,并把邻近点的邻近点以及邻近点本身添加到核心点所在簇中。

步骤 5:如果没有出现密度倾斜且邻近点的个数小于 Minpts,则将该点标记为噪声点。

步骤 6:循环执行步骤 2 ~ 6,直到对于 $\forall L_m, L_n \subset \text{STD}, m \neq n$ 满足式 18 和式 19,则聚类结束。

$$L_m \cap L_n = \varnothing \tag{18}$$

$$\underset{\forall p_i, p_j \in L_m}{\text{MAX}}(\text{Att}(p_i, p_j)) > \underset{\forall p_x \in L_m, \forall p_y \in L_n}{\text{MIN}}(\text{Att}(p_x, p_y)) \tag{19}$$

基于重心点转移的 St-DBSCAN 算法伪代码如下:

改进后 ST_DBSCAN(D, Eps_1, Eps_2, Minpts, Δt)

输入:数据对象集 $D = \{o_1, o_2, \cdots, o_n\}$;最大的地理空间距离值 Eps_1;最大的非空间距离值 Eps_2; Eps_1 和 Eps_2 距离内的空间点个数 Minpts;形成非空间阈值的阈值 Δt,这里是指时间维度

输出:空间点对象聚类簇 $C = \{C_1, C_2, \cdots, C_k\}$

(1) for each point p in dataset D

(2) if p is visited

(3) continue next point

(4) mark p as visited

(5) NeighborPts = regionQuerybyTime(p, Δt, Eps_2)

(6) / * 如果邻近点的个数小于 Minpts,将 p 和 p 点的邻近标记为噪声点 * /

(7) if size of(NeighborPts) <Minpts

(8) mark p as NOISE

(9) mark NeighborPts as Noise

(10) / * 求得 p 点的重心点 * /

(11) gravityPoint = getgravity(p, NeighborPts)

(12) / * 如果重心点偏离聚类点,则以重心点为聚类点对空间数据进行聚类 * /

(13) else if distRate(p, gravityPoint) ≥0.5

(14) C = next cluster

(15) NeighborPts = regionQuery(gravityPoint, Eps_1)

(16) If size of(NeighborPts) <Minpts

(17) Add NeighborPts to C

(18) Make NeighborPts as border

(19) Continue next cycle

(20) else

(21) gravityPoint = getgravity(p, NeighborPts)

(22) Continue next cycle

(23) expandCluster (gravityPoint, NeighborPts, C, Eps_1, Minpts)

(24) / * 如果重心点不偏离聚类点,则以点 p 为聚类点,对空间数据进行聚类 * /

(25) else if distRate(p, gravityPoint) <0.5

(26) C = next cluster

(27) expandCluster(p, NeighborPts, C, Eps_1, Minpts)

(28) End for

End Algorithm

3 实验结果与分析

3.1 实验数据

以昆明市出租车 GPS 数据为实验数据,在 Intel Core Duo T9900 3.06 GHz,4 核心 8 线程 CPU,内存 8 GB 的 HP 台式电脑上运行了改进 St-DBSCAN 算法与 St-DBSCAN 算法,并统计了两种算法的运行时间和聚类簇的密度因子。具体的实验数据例样及字段说明如表 1 所示。

表 1 交通信息字段及字段样例

字段名	字段解释	字段样例
name	出租车牌号	云 AT0001
jd	经度	102.709 47
wd	纬度	25.004 537
time	时间	2012-08-11-12.01.06
egid	路段 ID	28705
roadname	道路名称	“希望路”
kind	公路等级	6

3.2 算法时间性能的对比

在参数 Minpt = 50, Maxtime = 500 000 ms, Eps = 70 m 下,选取的数据集条数从 500 到 50 000 的运行耗时情况如图 5 所示。

从图 5 中可以看出,随着数据集的增大,改进后算法的运行时间与改进前算法的运行时间不断增大,但

是改进后算法的运行时间总是少于改进前的算法,其主要原因是改进后算法首先通过时间维度,将不同时间的空间点划分成不同的区域,通过外层循环减少了对空间对象的聚类操作,从而减少了算法的运行时间,提高了算法的时间性能。

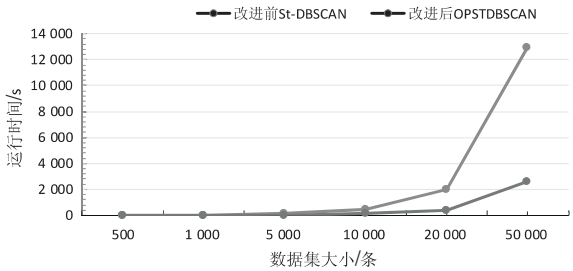

图 5　算法执行时间对比

3.3　算法聚类效果的对比

为了验证改进后 St-DBSCAN 与改进前 St-DBSCAN 的聚类效果,在参数为 Minpt = 100, Maxtime = 50 000 000 ms, Eps = 1 000 m 下,分别计算空间对象数量在 10 000 和 20 000 条件下,6 个不同数据集下簇的平均聚合度。具体的实验结果如图 6 和 7 所示。

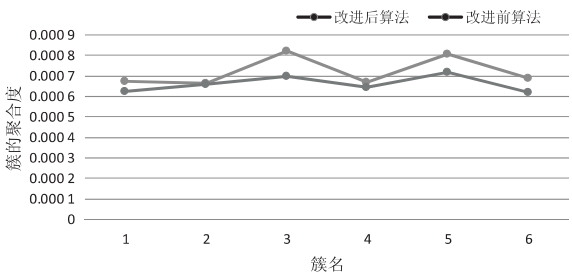

图 6　数据集为 20 000 时算法的簇的聚拢因子

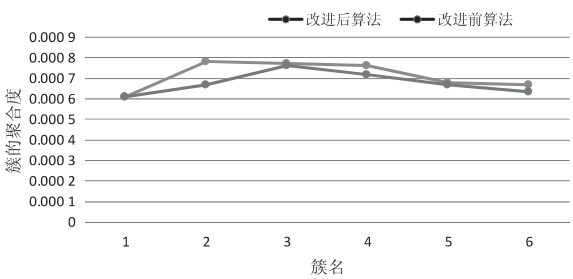

图 7　数据集为 10 000 时算法的聚合度

如图 6 和图 7 所示,在数据集数据条数为 10 000 和 20 000 时,改进后算法的聚合度大于改进前算法的聚合度,说明改进后算法提高了簇的聚类效果,其原因是改进后算法能通过重心点转移的方式解决空间点分布密度倾斜的问题。

综上,改进后 St-DBSCAN 算法在算法执行时间和聚类效果上都优于改进前 St-DBSCAN 算法,因此通过重心转移的方式改进 St-DBSCAN 算法切实可行。

4　结束语

采用核心点密度转移的方法解决了空间点分布密度倾斜的问题。实验结果表明,改进的 St-DBSCAN 算法在时间性能和聚类结果中的簇的密度因子都比 St-DBSCAN 算法有一定程度的提高。但是,对 St-DBSAN 算法的改进空间依然很大,例如如何通过选取更好的参数进一步提高 St-DBSCAN 算法的聚类结果,非空间属性与空间属性如何采取合适的聚类策略以提高 St-DBSCAN 的聚类效果,等等,这些都有待进一步研究。

参考文献:

[1] 安建瑞,张龙波,王　雷,等. 一种基于网格与加权信息熵的 OPTICS 改进算法[J]. 计算机工程,2017,43(2):206-209.

[2] 朱　亮,李东波,何　非,等. 采用改进型 DENCLUE 和 SVM 的电子皮带秤故障诊断[J]. 哈尔滨工业大学学报,2015,47(7):122-128.

[3] 韩利钊,钱雪忠,罗　靖,等. 基于区域划分的 DBSCAN 多密度聚类算法[J]. 计算机应用研究,2018,35(6):1668-1671.

[4] PILEVAR A H,SUKUMAR M. GCHL:a grid-clustering algorithm for high-dimensional very large spatial data bases[J]. Pattern Recognition Letters,2005,26(7):999-1010.

[5] 李　帅,吴　斌,杜修明,等. 基于 Spark 的 BIRCH 算法并行化的设计与实现[J]. 计算机工程与科学,2017,39(1):35-41.

[6] 赖向阳,宫秀军,韩来明. 一种 MapReduce 架构下基于遗传算法的 K-Medoids 聚类[J]. 计算机科学,2017,44(3):23-26.

[7] 白晓清,赵　瞻,鲍海波. 基于 CLARA 算法的考虑时序特性分布式电源规划[J]. 电力自动化设备,2016,36(5):14-22.

[8] GOYAL N. An efficient density based incremental clustering algorithm in data warehousing environment[C]//Proceedings of international conference on computer engineering and applications. [s. l.]:[s. n.],2009.

[9] 周　董,刘　鹏. VDBSCAN:变密度聚类算法[J]. 计算机工程与应用,2009,45(11):137-141.

[10] 赵文冲,蔡江辉,张继福. 改进 k 值自动获取 VDBSCAN 聚类算法[J]. 计算机系统应用,2016,25(9):131-136.

[11] 吴伟民,黄焕坤. 基于差分隐私保护的 DP-DBScan 聚类算法研究[J]. 计算机工程与科学,2015,37(4):830-834.

[12] 夏鲁宁,荆继武. SA-DBSCAN:一种自适应基于密度聚类算法[J]. 中国科学院研究生院学报,2009,26(4):530-538.

[13] 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.

[14] 戴　露. 基于手机 GPS 数据的出行端点识别方法研究[D]. 成都:西南交通大学,2017.