中图法分类号：TP301.6；TP18　　**文献标志码**：A　　**文章编号**：1006-8961(2010)10-1471-07

索引信息：胡彩平，秦小麟，任韧. 局部空间离群点算法的改进及其实现[J]. 中国图象图形学报,2010,15(10):1471-1477

局部空间离群点算法的改进及其实现

胡彩平，秦小麟，任　韧

（南京航空航天大学信息科学与技术学院，南京　210016）

摘　要：LOF 算法是一个著名的局部离群点查找方法，该方法赋予了表征每一个空间点偏离程度的数值。但 LOF 算法存在效率低和性能差的问题，为此对该算法进行了以下两个方面的改进：第一，提出了降低该算法时间复杂度的两步改进方法，并对这两步改进方法的时间复杂度也进行详细分析，第二，使得该算法在查找局部离群点时，不仅考虑了空间属性，也考虑了非空间属性。另外还通过实验测试了 LOF 算法及其改进方法的时间效率，以及在模拟数据和真实数据情况下的查找离群点的效果。实验结果表明，改进方法具有更好的时间效率和性能。

关键词：数据挖掘；空间离群点；可达距离；局部离群因子

The improvements and experiments of local spatial outlier detecting algorithm

HU Caiping, QIN Xiaolin, REN Ren

(*College of Information Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing* 210016)

Abstract: The LOF (local outlier factor) algorithm is a very distinguished local outlier detecting method, which assigns each object an outlier-degree value. In this paper, we present the two improvements of this algorithm. First, the two step improvements was introduced and their time complexity was analysed. Second, when the algorithm identified local outliers, it can consider spatial attributes and non-spatial attribute. The experiments have tested the executing time of the LOF algorithm and its improvements, the performance of computing synthetic and real data set. The experimental results show that is its improvements outperform the LOF algorithm in efficiency and performance.

Keywords: data mining; spatial outliers; reachability distance; local outlier factor (LOF)

0　引　言

如今遥感遥测、地理信息系统、多媒体系统、医学影像和卫星图像等多种应用中产生了大量空间数据，空间数据的数量、规模和复杂性都在飞快地增长，由于这些数据远远超过了人脑的分析能力，终端用户也不可能详细地分析所有的这些数据，并提取感兴趣的空间知识，致使“空间数据爆炸，但知识贫乏”，如何有效地从空间数据中获取有用的知识，对用户来说至关重要，因此空间数据挖掘的研究工作日益受到人们关注。利用空间数据挖掘从空间数据库中自动或半自动地挖掘事先未知，却潜在有用的空间模式，以及抽取没有清楚表现出来的隐含的知识和空间关系十分必要。所谓空间数据挖掘指的是从空间数据库中抽取隐含的知识、空间关系或非显

基金项目：国家高技术研究发展计划(863)项目(NO2007AA01Z404)；国家自然科学基金(60673127)；南京航空航天大学科研启动基金项目(S0848-042)和南京航空航天大学基本科研业务费专项科研项目(NS2010094)。

收稿日期：2009-09-28；**改回日期**：2010-04-14

第一作者简介：胡彩平(1977—　)，男。讲师，2007 年获南京航空航天大学计算机应用技术专业博士学位，研究方向为空间数据挖掘等。E-mail：hucaiping@ nuaa. edu. cn。

式地存储在空间数据库中有意义的特征或模式[1-2]的技术。

一般地,空间数据挖掘可以分成空间分类和预测、空间聚类、空间离群点和空间关联规则[3]4 类挖掘方法。其中离群点可分为全局离群点和局部离群点。对于全局离群点,Hawkins 将其定义为“一个离群点是一个观察点,它偏离其他观察点如此之大,以至引起怀疑是由不同机制生成的”[4]。Shekhar 等人最早提出了空间离群点的定义[5]。随着研究的不断深入,又相继提出了一些空间离群点的查找方法[6-8]。但这些方法都是针对全局空间离群点。而现实世界中的很多复杂结构的数据集中,某些对象仅仅是在一个局部的观测范围内,相对于某一个数据群组分离,而从宏观和整体上看,这样的分离并不那么明显。而这种类型的点,往往可能是人们需要寻找的具有特殊意义的点,如果从全局的角度来考察,则无法对它们进行合适的界定。

局部离群点是指在数据集中与其邻域表现不一致或大大地偏离其邻域的观测点[9]。Breuning 等学者提出了查找局部离群点的 LOF 算法[10],该算法不是去界定哪些点是离群点,哪些点不是离群点,而是转向量化地描述点的离群程度,通过赋予每一个点一个表征它离群程度的量化指标来进行离群点的搜索。自从 LOF 算法出现后,已出现了许多离群度的度量方法,其中比较典型的有基于连接的离群系数[11]、多粒度偏差因子[12]和局部空间离群测度[13]等方法。

1 LOF 算法的基本概念

LOF 算法的核心思想,就是通过赋予每一个空间点一个表征该点偏离程度的因子,而不是明确地来界定哪些点是离群点,哪些点不是离群点,而这个表征一个空间点偏离程度的数值,实质上反映了该点是否分布在空间点较为集中的局部区域之中。关于 LOF 算法的详细介绍可参考文献[10],以下仅给出反映空间点偏离程度的局部离群因子的定义。

定义 1 局部离群因子(LOF)

对象 p 的局部离群因子定义为

$$LOF_M(p)=\frac{\sum_{o\in N_M(p)}\frac{D_M^{LR}(o)}{D_M^{LR}(p)}}{|N_M(p)|}$$

其中 $N_M(p)$ 表示对象 p 的第 M 个距离领域,$D_M^{LR}(p)$ 表示局部可达密度(local reachability density),它反映了空间对象 p 所处局部空间区域的密度情况。分析该式可知,$\frac{\sum_{o\in N_M(p)}D_M^{LR}(o)}{|N_M(p)|}$ 是计算出对象 p 的第 k 距离邻域内的所有对象的局部可达密度的平均值,该数值反映了在对象 p 的第 k 距离范围之内的空间点的平均分布密度,若对象 p 是偏离程度较大的点,则它的第 k 距离邻域内的点则大多数是距离对象 p 较远并且处于某一个群组中的点,那么这些点的局部可达密度的数值则较大,该式的计算结果也较大,而对象 p 本身由于是偏离程度较大的点,因此其局部可达密度数值较小,最终导致 LOF 数值较大。反之,若点 p 处于某一个群组之中,则其第 k 距离邻域内的点与其属于同一个群组的可能性较大,所得出的平均分布密度与该点的局部可达密度数值差异不会很大,其 LOF 数值接近于 1,这已经成功剥离了不同群组之间的密度差异所带来的影响。

2 LOF 算法的复杂度分析及改进

2.1 LOF 算法的复杂度分析

设空间点的总数为 N。

考察公式

$$LOF_M(p)=\frac{\sum_{o\in N_M(p)}\frac{D_M^{LR}(o)}{D_M^{LR}(p)}}{|N_M(p)|}$$

可以改写为

$$LOF_M(p)=\frac{\sum_{o\in N_M(p)}D_M^{LR}(o)}{|N_M(p)|\times D_M^{LR}(p)}$$

由此可以看出,LOF 的复杂度分为 3 个部分之和:第 1 部分为局部可达密度的复杂度,即计算对象 p 的局部可达密度的时间复杂度;第 2 部分为计算 p 的第 M 个距离邻域的复杂度,第 3 部分即遍历 p 的第 M 个距离邻域的复杂度(基本算术运算和集合大小的计算均为常数时间)。

$$O(LOF_M)=O(D_M^{LR})+O(N_M)+O(|N_M|\times O(D_M^{LR}))$$

$\tilde{d}_k(p)$ 表示对象 p 的第 k 个距离,由距离 $\tilde{d}_k(p)$ 的定义[10]可知,对于每一对象,$\tilde{d}_k(p)$ 的计算均等价,计算复杂度与所涉及的对象无关,所以设 $\tilde{d}_k(p)$ 的计算复杂度为 $O(\tilde{d}_k)$。

$\hat{d}_k(p,o)$表示对象 p 相对于对象 o 的可达距离[10],而可达距离$\hat{d}_k(p,o)=\max\{\tilde{d}_k(0),d(p,o)\}$,与所涉及的对象无关,所以有

$$O(\hat{d})=O(\tilde{d}_k)$$

由 $D_M^{\mathrm{LR}}(p)=1\Big/\left(\dfrac{\sum_{o\in N_M(p)}\hat{d}_M(p,o)}{|\boldsymbol{N}_M(p)|}\right)$可知[10],局部可达密度的复杂度分为以下两个部分之和:第 1 部分为计算对象 p 的第 M 个距离邻域的复杂度,第 2 部分为遍历对象 p 的第 M 个距离领域的复杂度,即

$$O(D_M^{\mathrm{LR}})=O(\boldsymbol{N}_M)+O(|\boldsymbol{N}_M|\times O(\hat{d}))$$

由 $\boldsymbol{N}_M(p)$的定义可知,该计算过程是要找出到对象 p 的距离小于第 M 个距离的所有对象的集合,通过遍历集合中的所有对象并找出满足条件的对象,该计算过程的复杂度为

$$O(\boldsymbol{N}_M)=O(\tilde{d}_M)+N$$

假定$\tilde{d}_k(p)$的计算采用以下算法:

1) 遍历整个数据集,计算出每一个对象到对象 p 的距离;

2) 对距离进行排序(假设采用 $N\log N$ 级的排序算法)并返回第 k 大的值。

若采用该计算方法,则$\tilde{d}_k(p)$的计算复杂性与 k 的取值没有任何关系,即

$$O(\tilde{d}_k)=O(N\log N)$$

综上,在假定了距离$\tilde{d}_k(p)$的计算方法之后,考虑最坏的情况:参数 M 取 $N-1$,使得$|\boldsymbol{N}_M(p)|=N-1$,该种下情况有:

$$O(\hat{d})=O(N\log N)$$
$$O(N_M)=O(N\log N)$$
$$O(D_M^{\mathrm{LR}})=O(N^2\log N)$$
$$O(LOF_M)=O(N^3\log N)$$

计算每一个点的 LOF 值算法循环 N 次。所以,在最坏情况下 LOF 算法计算出所有对象的离群因子的时间复杂度为 $O(N^4\log N)$,这是 LOF 算法的极限上界,因为对于任意的对象 p 属于 D 均有$|\boldsymbol{N}_M(p)|=N-1$。

考虑该算法的下界,当参数 M 取 1,即当对于任意的对象 p 属于 D 均有$|N_M(p)|=1$ 时,则有

$$O(D_M^{\mathrm{LR}})=O(N\log N)$$
$$O(LOF_M)=O(N\log N)$$

LOF 算法的复杂度降为 $O(N^2\log N)$,这是算法的最好情况。所以该算法的复杂度在假定了计算$\tilde{d}_k(p)$的时间复杂度为 $O(N\log N)$后,其下界为$\Omega(N^2\log N)$,上界为 $O(N^4\log N)$。

引理　在一个由 N 个元素组成的线性集合中提取第 k 大元素,SELECT 算法的时间复杂度是$\Theta(N)$[14]。

根据该条引理,$\tilde{d}_k(p)$的算法可改写为:

1) 遍历整个数据集,计算出每一个对象到对象 p 的距离;

2) 通过 SELECT 算法找出该序列中的第 k 小值。

则 $O(\tilde{d}_M)$的时间复杂度降为 $O(N)$,由于$O(|\boldsymbol{N}_M|\times O(N))$是 $O(N)$的高阶,所以有

$$O(\boldsymbol{N}_M)=O(N)$$
$$O(\hat{d})=O(N)$$
$$O(D_M^{\mathrm{LR}})=O(|\boldsymbol{N}_M|\times N)$$

而 $O(D_M^{\mathrm{LR}})$是 $O(N_M)$的高阶,而 $O(|\boldsymbol{N}_M|\times O(D_M^{\mathrm{LR}}))$是 $O(D_M^{\mathrm{LR}})$的高阶,所以 $O(LOF_M)$可化简为 $O(LOF_M)=O(|\boldsymbol{N}_M|\times O(D_M^{\mathrm{LR}}))$,代入可得

$$O(LOF_M)=O(|\boldsymbol{N}_M|^2\times N)$$

所以,按照先前的分析,最坏的情况$|\boldsymbol{N}_M(p)|=N-1$,最好的情况$|\boldsymbol{N}_M(p)|=1$。在计算$\tilde{d}_k(p)$采取最优化的算法的情况下,算法的复杂度下界为$\Omega(N^2)$,上界为 $O(N^4)$。

2.2　LOF 算法的改进

任何空间对象不仅反映了其空间方位的空间属性,也反映了其他特征的非空间属性。因此查找空间离群点,需要综合考虑空间与非空间属性。本文定义空间离群点是在相邻空间区域内,其非空间属性与其他空间对象区别十分显著的空间对象,其相关的定义如下:

定义 2　加权距离

设 $p,q\in\boldsymbol{D}$,p 和 q 的 t 维非空间属性是 $f(p)$和 $f(q)$,其中 $f_k(p)$和 $f_k(q)$是第 $k(k=1,2,\cdots,t)$维规一化属性,且 $0\leqslant f_k(p),f_k(q)\leqslant 1$, w_k是第 k 维的权值,且 $0\leqslant w_k\leqslant 1$,则数据对象 p 和 q 之间的加权距离为

$$d(p,q,w)=\sqrt{\sum_{k=1}^{t}w_k\,(f_k(p)-f_k(q))^2}$$

式中,$\sum_{k=1}^{t}w_k=1$

定义 3　空间对象 p 的邻域 $\boldsymbol{N}(p)$是指一个对

象集 $\boldsymbol{P}=\{p_1,\cdots,p_k\}$，这里每个对象 p_i 都与 p 相邻。

定义 4 ε-匹配邻域

给定 ε 和对象 $p\in\boldsymbol{D}$，对象 p 的匹配邻域用 $\tilde{\boldsymbol{N}}_\varepsilon(p)$ 来表示，它被定义为

$$\tilde{\boldsymbol{N}}_\varepsilon(p)=\{x\in\boldsymbol{D}\mid d(p,x,w)\leqslant\varepsilon \text{ and } x\in\boldsymbol{N}(p)\}$$

定义 5 对象 p 的第 k 个距离

对于一个正整数 k，对象 p 的第 k 个距离记作 $\tilde{d}_k(p)$。在数据库 $\boldsymbol{D}$ 中，在一个相邻的空间区域内，存在一个对象 o，则将该对象与对象 p 之间的距离记作 $d(p,o,w)$。

若满足以下条件，则取 $\tilde{d}_k(p)$ 等于 $d(p,o,w)$：

1）至少存在 k 个对象 $\tilde{o}\in\boldsymbol{D}/\{p\}$ 满足 $d(p,\tilde{o},w)\leqslant d(p,o,w)$；

2）至多存在 $k-1$ 个对象 $\tilde{o}\in\boldsymbol{D}/\{p\}$ 满足 $d(p,\tilde{o},w)<d(p,o,w)$。

定义 6 对象 p 的第 k 个距离邻域

如果已知对象 p 的第 k 个距离，那么对象 p 的第 k 个距离邻域就是所有到对象 p 的距离小于等于对象 p 的第 k 个距离的所有对象集合，记作

$$\tilde{\boldsymbol{N}}_{\tilde{d}_{k(p)}}(p)=\{q\in\boldsymbol{D}\backslash\{p\}\mid d(p,q,w)\leqslant\tilde{d}_k(p)\}$$

本文将 $\tilde{\boldsymbol{N}}_{\tilde{d}_{k(p)}}(p)$ 简写为 $\tilde{\boldsymbol{N}}_k(p)$。

定义 7 对象 p 相对于 o 的可达距离

设 k 为一自然数，则对象 p 相对于对象 o 的可达距离记作

$$\hat{d}_k(p,o)=\max\{\tilde{d}_k(o),d(p,o,w)\}$$

定义 8 局部可达密度

设 M 为一正整数，则局部可达密度记作

$$D_M^{\mathrm{LR}}(p)=1\Big/\left(\frac{\sum_{o\in\tilde{N}_M(p)}\hat{d}_M(p,o)}{|\tilde{\boldsymbol{N}}_M(p)|}\right)$$

定义 9 新局部离群因子

新的局部离群因子定义为

$$LOF_M(p)=\frac{\sum_{o\in\tilde{N}_M(p)}\frac{D_M^{\mathrm{LR}}(o)}{D_M^{\mathrm{LR}}(p)}}{|\tilde{\boldsymbol{N}}_M(p)|}$$

在计算对象 p 的 LOF 数值时，需要首先计算出对象 p 的若干个性化的参数，包括 $\tilde{\boldsymbol{N}}_M(p)$，$\tilde{d}_M(p)$，$D_M^{\mathrm{LR}}(p)$。而在 $D_M^{\mathrm{LR}}(p)$ 的计算过程中，则需要计算 $\hat{d}_k(p,o)$。由定义可知，$\hat{d}_k(p,o)=\max\{\tilde{d}_k(o),d(p,o,w)\}$。那么在对邻域 $\tilde{\boldsymbol{N}}_M(p)$ 中的某一个对象 o 进行其 LOF 数值的计算时，若对象 p 也在 o 的邻域 $\tilde{\boldsymbol{N}}_M(o)$ 之中，则会重复计算 $\tilde{d}_M(p)$ 以及 $d(p,o,w)$，由此可知，若存在对象 p 和 q，满足 $p\in\tilde{\boldsymbol{N}}_M(q)$ 以及 $q\in\tilde{\boldsymbol{N}}_M(p)$，则这两个对象的可达距离以及 $d(p,q,w)$ 均被重复计算了一次，而对于一个密度较大的群组中的点来说，其每一个点 o 的邻域 $\tilde{\boldsymbol{N}}_M(o)$ 中的点 p 的邻域 $\tilde{\boldsymbol{N}}_M(p)$ 包含点 o 的可能性非常之大，若点 o 的邻域 $\tilde{\boldsymbol{N}}_M(o)$ 之中存在 m 个满足这样条件的点，则 $\tilde{d}_M(o)$ 和两点之间的距离会被重复计算 m 次。

同样的问题出现在第 k 个距离邻域的计算过程中，在 LOF 值的计算过程之中，需要计算 $\tilde{\boldsymbol{N}}_M(p)$，而对于邻域 $\tilde{\boldsymbol{N}}_M(p)$ 中的每一个对象 o，在它的局部可达密度值的计算过程中，由于需要计算邻域 $\tilde{\boldsymbol{N}}_M(o)$，因此可以考虑通过适当增大空间开销来获取时间效率的提高。其优化步骤如下：

1）在算法执行之前，先计算所有对象的 $\tilde{d}_k(p)$ 值并保存起来，以作备用。在算法的执行过程之中凡是需要用到 $\tilde{d}_k(p)$ 的地方，皆直接使用预先计算好的资源；然后，计算所有对象 p 的邻域 $\tilde{\boldsymbol{N}}_M(p)$，并保存起来，以备用，这一步的优化可降低算法时间复杂度的阶。

2）在第 1 步优化的基础上，在算法执行之前，先计算所有对象两两之间的距离，并保存起来，以作备用。在算法的执行过程之中凡是需要用到两点之间距离的地方皆直接使用预先计算好的资源。这一步的优化可以降低算法时间复杂度的隐含常数。

第 1 步优化之后的时间复杂度分析，在该优化中，$\tilde{d}_k(p)$ 在计算过程中只需要常数时间就可以获得，而由公式 $\hat{d}_k(p,o)=\max\{\tilde{d}_k(o),d(p,o,w)\}$ 可知，$\hat{d}_k(p,o)$ 在常数时间可获得，而对于任意一对象 p，其邻域 $\tilde{\boldsymbol{N}}_M(p)$ 也可以在常数时间内获得，因此计算局部可达密度参数的时间复杂度可简化为

$$O(D_M^{\mathrm{LR}})=O(|\tilde{\boldsymbol{N}}_M(p)|)$$

同理，因为 $O(|\tilde{\boldsymbol{N}}_M(p)|\times O(D_M^{\mathrm{LR}}))$ 是最高阶，则有：

$$O(LOF_M)=O(|\tilde{\boldsymbol{N}}_M(p)|\times O(D_M^{\mathrm{LR}}))=O(|\tilde{\boldsymbol{N}}_M(p)|^2)$$

而在计算 LOF 值之前，需要计算所有对象的离群距离 $\tilde{d}_k(p)$ 数值以及所有对象的邻域集合，计算 $\tilde{d}_k(p)$ 的时间复杂度为 $N\times O(\tilde{d}_k)$，而计算单个对象的邻域的复杂度，由于第 k 个距离可以在常数时间

获得,其时间复杂度为 $O(\widetilde{N}_M(p)) = O(N)$,所以算法总的复杂度分为 3 部分之和,第 1 部分为计算所有点的$\widetilde{d}_k$,第 2 部分为计算所有点的第 k 个距离邻域,第 3 部分为 LOF 数值的计算,即

$$N \times O(\widetilde{d}_k) + N \times O(N) + N \times O(|\widetilde{N}_M(p)|^2)$$

1) 第一个距离$\widetilde{d}_k$ 采用 $O(N)$级算法,同理可知,最坏情况下,LOF 算法总的时间复杂度为 $O(N^3)$,最好情况下,单个 LOF 数值的计算的时间复杂度可降低到$O(N)$级,总的复杂度为 $O(N^2)$,因此,现在的复杂度下界为 $\Omega(N^2)$,上界为 $O(N^3)$,相对于未进行此优化之前的复杂度下界 $\Omega(N^2)$,复杂度上界 $O(N^4)$有了降低。

2) 如果计算第 k 个距离$\widetilde{d}_k$ 采用 $O(N\log N)$级算法,则算法最坏情况的复杂度为 $O(N^3)$,最好情况复杂度为 $O(N^2\log N)$。相对于下界为 $\Omega(N^2\log N)$,上界为 $O(N^4\log N)$的未优化情况的复杂度也有了降低。

这样,算法中就需要增加 N 个规模的空间开销来存储第 k 个距离$\widetilde{d}_k$ 的数值,而对于空间的第 k 个距离邻域,算法则需要增加规模为 N^2 级的空间开销,以一定的空间来换取时间的方式有效地降低了算法执行时间的上界,实验表明,该改进将至少 10 倍以上的提高算法执行的时间效率,效率的提升随着计算规模的扩大而增大。

第 2 步优化之后的时间复杂度分析,第 2 步的优化是在第 1 步优化的基础之上,将两两空间点之间的距离预先计算并存储,该步计算会进行$\frac{1}{2}n(n-1)$次循环,由于两点之间的距离计算可在常数时间内完成,因此增加的时间开销为 $O(N^2)$,这种改进不会增加算法的时间复杂度的阶。由此可见该步改进的最终效果是降低了隐含常数因子。实验表明,该步改进可以将算法的执行速度在第 1 步的基础之上再加快一倍左右。而付出的代价就是需要增加至少$\frac{1}{2}n(n-1)$,即 N^2 规模的空间开销。

这两种改进的实际实验效果将在下文中进行展示,而对于第 k 个距离$\widetilde{d}_k$ 的计算,则不采用 $O(N)$级的 SELECT 算法,因为该算法时间复杂度的隐含常数因子非常之大,而在实现 LOF 演示的程序之中的样本规模较小,使得 SELECT 算法的实际复杂度会接近于 $O(N^2)$级别,所以在演示程序中,第 k 个距离$\widetilde{d}_k$ 的计算采用复杂度为 $O(N\log N)$级的算法。

3　实验结果及分析

3.1　实验概述

该实验程序使用 C#编写,采用的开发环境为 Microsoft Visual Studio 2005,数据库采用 Microsoft Access 2003。数据的计算结果将以可视化的方式展示给用户,在完成分析之后,将赋予每一个点一个 LOF 数值,程序中以最大的 LOF 数值为标准,在视图中显示出所有点的 LOF 数值相对于最大数值高度的柱状图,以展示出每一个点的离群程度(如图 1 所示)。

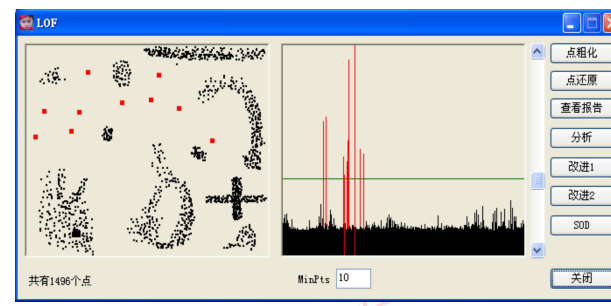

图 1　LOF 算法分析窗体

Fig. 1　The analyzing forms of the LOF algorithm

3.2　算法的效率比较

通过不同规模的数据和阈值对 3 种方法进行测试,以比较其执行时间效率,通过上一节的分析可知,$\widetilde{d}_k(p)$若采用线性时间复杂度的算法,能使得 LOF 算法的复杂度达到 $O(N^4)$,但是查找线性序列中第 k 个元素的线性时间复杂度,由于算法在阶数 N 之前的隐含常数非常之大,对于较小规模的数据量,若常数过大,则其效能反而不如 $O(N^2)$级的算法。所以,在实现中,计算 $\widetilde{d}_k(p)$ 采用复杂度为 $O(N\log N)$的方法,即先计算再排序。图 2 是在参数

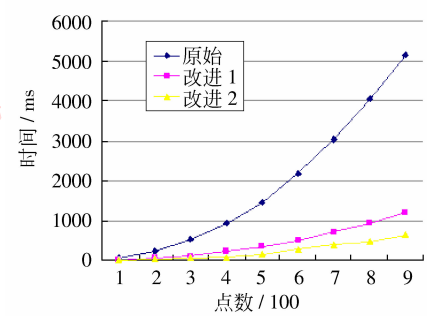

图 2　3 种方法的运行时间比较

Fig. 2　Runtime comparison of three methods

M 为 1 情况下的测试结果,虽然 3 种方法的时间复杂度的下界均相同,但是改进 2 的实际运行时间基本上是改进 1 方法的一半,而改进 1 方法和改进 2 方法都比原始方法有了实质性的提高。

3.3 算法性能测试

本实验分别用人工合成的数据和真实数据来测试该算法的性能。测试时,首先用人工合成的数据,测试 LOF 算法对离群点进行搜索的准确性,一般情况下数据样本包括以下几种典型的类型:

1)点在数据空间中较为均匀地分布,不存在明显的离群现象。

由图 3 可以看出,在一个密度较为均匀的群组内部,其每一个点的 LOF 值差异很小并且均围绕 1 上下波动,LOF 的最大值 1.141,最小值 0.954,方差 0.181,平均值为 1.024。

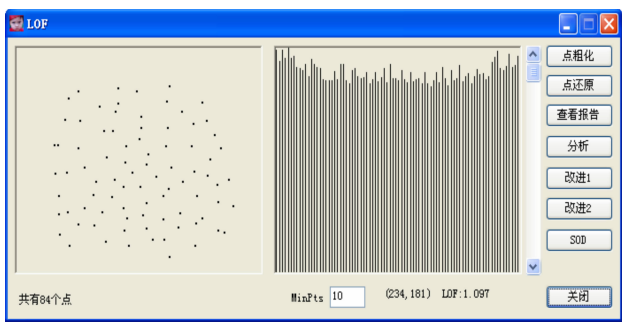

图 3 均匀分布的计算结果

Fig. 3 The result for uniform densities

2)数据空间中存在一个群组和周围的离群点。

图 4 展示了 LOF 算法对于离群程度的检测能力,同一群组之中的点的 LOF 数值差异很小,并且均在 1 上下,而偏离点的 LOF 数值则表现出了非常大的差异。离群点的 LOF 的最大值 7.455,最小值为 2.390,群组中的点 LOF 的最大值为 1.164,最小值为 0.915。

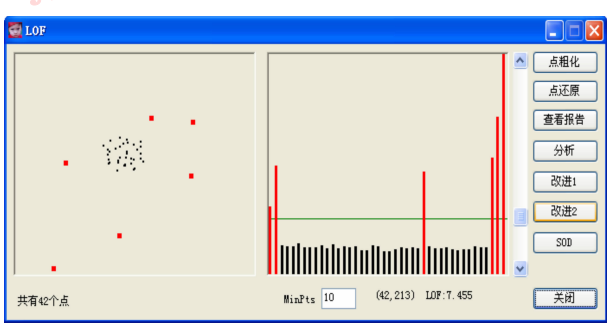

图 4 存在群组和周边离群点的计算结果

Fig. 4 The result for a cluster and outliers around it

3)数据空间中存在多个不同密度的群组和离群点。

LOF 算法的一个最为重要的性能,那就是面对存在多个不同密度的群组的情况,LOF 算法剥离了由不同群组的密度差异带来的影响,如图 5 所示,表现出了离群点和聚集点之间的数值差异。离群点的 LOF 的最大值为 6.916,最小值为 2.000,群组中的点 LOF 最大值为 1.353,最小值为 0.919。

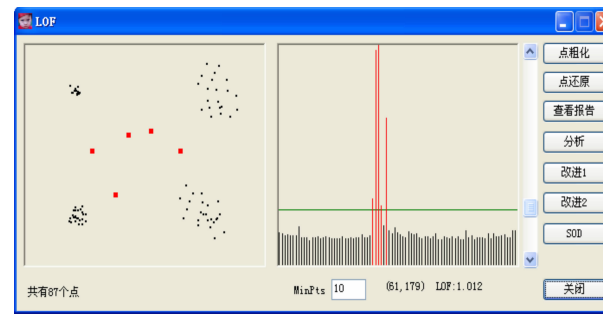

图 5 存在不同密度群组的计算结果

Fig. 5 The result for different densities

4)数据空间中存在分级群组,即在一个密度较小的大的群组之中又包含了密度较大的子群组。离群点的 LOF 的最大值为 4.820,最小值为 2.688 ,群组中的点 LOF 最大值为 1.856,最小值为 0.924。

图 6 为一个大的密度较低的群组之中存在一个密度较大的子群组的情形,在这种较为复杂的情形下,LOF 算法仍然能够准确地给 3 个偏离的点以较大的 LOF 数值。

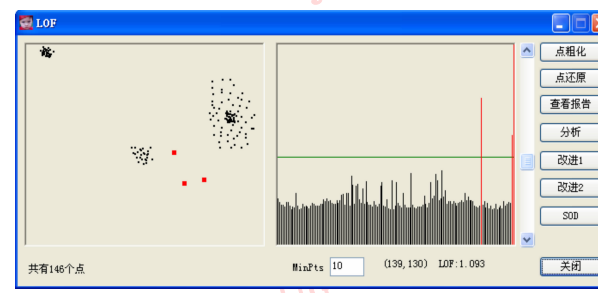

图 6 存在分级群组的计算结果

Fig. 6 The result for Leveled clusters

由上述几种情况的测试可以发现,LOF 算法不仅能较好地符合理想的偏离系数曲线,而且能够充分地突出离群点和有效地剥离不同规模不同密度的群组对离群点检测的影响,这是 LOF 算法的一个非常好的性能。

在第 2 个实验中,以某地 2005 年水稻产量数据

库为例进行测试,并与文献[5]方法(SLZ 算法)进行了比较。为减小计算量,本文选取了 40×40 网格单元组成一个平面空间区域,每个单元实际代表 1000 m×1000 m 大小的耕地,从每个网格中选取 5 个属性(F_1,F_2,F_3,F_4 和 F_5),F_1 代表空间位置,F_2 代表水稻产量,F_3,F_4 和 F_5 依次表示降水、日照和温度等气象要素。用这两种方法查找空间离群点的实验结果如表 1、表 2 所示,上述所有属性都经过规一化处理。实验结果表明,改进算法发现的 5 个离群点都是局部离群点。由于空间数据具有空间自相关性,即每一个事物都与其他事物相关,但邻近事物间的相关性比距离较远的事物间的相关性要大得多,因此局部离群点的查找对评价水稻产量有实际意义。但用 SLZ 算法查找的坐标为(15,36)的离群点是一个全局离群点,可见改进算法的性能比 SLZ 算法的好。

表 1　用改进 LOF 算法查找的离群点

Tab.1　Outliers identified by the new LOF algorithm

坐标	产量	降水	日照	温度
(2,3)	0.25	0.12	0.79	0.52
(10,12)	0.34	0.21	0.31	0.13
(13,20)	0.26	0.32	0.44	0.24
(25,30)	0.27	0.58	0.56	0.98
(30,10)	0.91	0.82	0.83	0.28

表 2　用 SLZ 算法查找的离群点

Tab.2　Outliers identified by SLZ

坐标	产量	降水	日照	温度
(2,3)	0.25	0.12	0.79	0.52
(10,12)	0.34	0.21	0.31	0.13
(15,36)	0.11	0.42	0.23	0.78
(25,30)	0.27	0.58	0.56	0.98
(30,10)	0.91	0.82	0.83	0.28

4　结　论

随着空间数据库应用的不断深入,查找空间离群点已成为数据挖掘和知识发现的一项重要任务。基于密度的 LOF 算法为空间离群点分析和搜索提供了一个非常好的手段。本文对该算法进行了改进,即在查找空间离群点时,综合考虑了空间与非空间属性。而 LOF 算法的不足之处在于,其对于每一个点都是同等地进行分析,算法复杂度仍然较高。实验结果表明,改进方法具有更好的时间效率和性能。

参考文献(References)

[1] Han J W, Kamber M. Data Mining Concepts and Techniques [M]. Beijing China Machine Press,2001. [Han J W, Kamber M 著. 数据挖掘概念与技术[M]. 范 明,孟小峰等译. 北京:机械工业出版社,2001.]

[2] Lu W,Han J W. Discovery of general knowledge in large spatial databases [C]// Proceedings of Far East Workshop on Geographic Information Systems. Singapore: World Scientific, 1993: 275-289.

[3] Shekhar S, Chawla S. (Xie Kunqing, Ma Xiujun, Yang Dongqing, et al. Translate.) Spatial Databases: A Tour[M]. Beijing: China Machine Press,2004. [Shekhar S,Chawla S 著. 空间数据库[M]. 谢昆青,马修军,杨冬青等译. 北京:机械工业出版社,2004.]

[4] Hawkins D. Identification of Outliers [M]. London: UK: Chapman and Hall,1980.

[5] Shekhar S,Lu Changtien,Zhang Pusheng. A unified approach to detecting spatial outliers [J]. GeoInformatica,2003,7(2): 139-166.

[6] Lu Changtien,Chen Dechang,Kou Yufeng. Algorithms for spatial outlier detection[C]//Proceedings of the 3rd IEEE International Conference on Data Mining, Melbourne, Florida, USA: IEEE Computer Society,2003: 597-600.

[7] Lu Changtien, Chen Dechang, Kou Yufeng. Detecting spatial outliers with multiple attributes [C]//Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento,California,USA:IEEE Computer Society,2003: 122-128.

[8] Hu Tianming,Sung S Y. A trimmed mean approach to finding spatial outliers[J]. Intelligent Data Analysis,2004(8): 79-95.

[9] Xue Anrong, Ju Shiguang, He Weihua, et al. Study on algorithms for local outlier detection [J]. Chinese Journal of Computers. 2007,30(8): 1455-1463. [薛安荣,鞠时光,何伟华,等. 局部离群点挖掘算法研究[J]. 计算机学报. 2007,30(8): 1455-1463.]

[10] Breunig M,Kriegel H P,Ng R, et al. LOF: Identifying density-based local outliers [C]//Proceedings of ACM SIGMOD Conference. New York, USA: ACM, Press 2000: 93-104.

[11] Tang J,Chen Z,Fu A, et al. Enhancing effectiveness of outlier detections for low-density patterns[C]//Proceeding of Advances in Knowledge Discovery and Data Mining 6th Pacific Asia Conference. Berlin, German: Springer, 2002: 535-548.

[12] Papadimitirou S, Kitagawa H, Gibbons P B, et al. LOCI: Fast outlier detection using the local correlation integral [C]// Proceedings of the 19th International Conference on Data Engineering. Bangalore, Los Alamitos, CA, USA: IEEE Computer Society,2003: 315-326.

[13] Chawla Sanjay,Sun Pei. SLOM: A new measure for local spatial outliers[J]. Knowledge and Information Systems,2006,9(4): 412-429.

[14] Alsuwaiyel M H. Algorithms Design Techniques and Analysis [M]. Beijing: Electronics Industry Press,2001. [Alsuwaiyel M H. 算法设计技巧与分析[M],北京:电子工业出版社,2001.]