基于局部相对密度的离群点检测算法

何 旭 邓安生 葛小龙
(大连海事大学 辽宁 大连 116026)

摘 要 数据集中离群点占比很小,但大多现有的方法在检测期间需要对所有数据都进行离群度计算。针对此问题提出一种基于互近邻聚类的正常数据去除算法(EMNC),通过数据预处理最大程度消除正常点。只考虑 k 最近邻不适用分布异常的离群点,充分利用对象与其邻居的分布,同时考虑 k 最近邻、反近邻和共享近邻来进行密度估计。最后重新定义基于局部相对密度的离群度(ROF)对剩余可疑点进行离群判断。该算法在减少离群度计算量的同时提升了检测效率,在合成与真实数据集上和其他方法的对比实验结果表明了算法的有效性。

关键词 离群点检测 局部相对密度 互近邻聚类 共享近邻

中图分类号 TP391 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.12.042

A LOCAL RELATIVE DENSITY-BASED APPROACH FOR OUTLIER DETECTION

He Xu Deng Ansheng Ge Xiaolong
(*Dalian Maritime University*, *Dalian 116026*, *Liaoning*, *China*)

Abstract The proportion of outliers in the data set is very small, but the existing methods have to calculate the outliers of all the data during the outlier detection. To solve this problem, a normal data elimination algorithm based on MNN clustering (EMNC) is proposed, which preprocesses the data to eliminate normal points to the greatest extent. The density outlier detection algorithm that only considers k nearest neighbors cannot well adapt to outliers with abnormal data distribution. This algorithm made full use of the distribution of objects and their neighbors, and meanwhile considers k nearest neighbors, inverse nearest neighbors and shared nearest neighbors to estimate the density. A local relative density-based outlier factor (ROF) was redefined to evaluate the rest outlier of doubtful points. The ROF algorithm not only reduced the amount of data needed to calculate the local outlier, but also improved the detection efficiency. Experimental results on synthetic and real datasets show the effectiveness of the ROF algorithm compared with other methods.

Keywords Outlier detection Local relative density MNN clustering Shared nearest neighbors

0 引 言

近年以来,离群点检测也称为异常检测,作为数据挖掘技术的热点赢得了越来越多人的关注。Hawkins 曾给出过离群点的定义:在给定的数据集中,若某个数据对象大幅度偏离整体趋势,使人不得不怀疑这个对象与其他大多数数据对象的产生机制不同,则它是一个离群点[1]。因为一些更重要的信息可能包含在一些偏离其他数据分布的离群数据点中,所以离群点检测在识别出数据集中的异常数据的同时也是为了探索出数据集中的潜在信息。如今随着离群点检测技术的日渐成熟,离群点检测在许多应用领域拥有了十分广泛的应用前景,比如欺诈检测、视频监控、网络性能和药物研究等。离群点检测在未来的发展中将会应用在更多的行业中,并且可以更好地为人类的决策提供指导作用。现有的离群点检测研究方向可分为以下四类:基于统计[2-3]、基于距离[4-6]、基于聚类[7-10]和基于密度[11-18]的离群点检测算法。

基于统计的离群点检测算法使用参数法或非参数

收稿日期:2021-07-29。**何旭**,硕士生,主研领域:机器学习,智能信息处理。**邓安生**,教授。**葛小龙**,硕士生。

法获得数据集的生成模型,然后使用这个模型偏移标准分布的数据对象被判定为离群数据[2-3]。但是,这种方法的局限是不适用于多维数据集或者分布未知的数据集。基于距离的离群点检测算法[4-6]有效解决了多维数据集的离群点检测问题,但这些方法都是从全局的角度出发,因此如果离群点位于稠密簇或者稀疏簇时可能会出现漏检的情况。基于聚类的离群点检测算法通常把聚类结果的簇当作正常数据,主要通过数据对象与簇之间的关系来检测离群点[7-10]。

基于密度的离群点检测算法弥补了基于距离检测方法的不足,适用于密度分布不均匀的数据集。Breuing 等[11]最早提出局部离群点是指那些与其邻域分布不一致或者偏离它们的点,LOF 算法通过数据的局部可达密度计算局部离群因子(LOF),LOF 值越大,数据就有越大的可能是离群点。在 LOF 算法的基础上,Jin 等[12]提出同时考虑 k 近邻和反近邻对数据影响的 INFLO 算法,INFLO 和 LOF 相比包含了更多的邻域信息。为了简化对离群因子 LOF 的计算,Simplified-LOF[14]算法提出利用 k-距离来代替可达距离,但这种简化算法的准确性也受到了限制。Guan 等[13]提出了基于局部密度的相对离群因子很好地解决了在低密度数据集里离群点密度与其邻居密度相似不好判断的问题。Du 等[15]提出了 NSD 算法,利用截取距离有效地解决了高维数据检测精度低以及参数敏感的缺陷。Tu 等[16]使用方形邻域,并使用裁剪系数计算可达距离和局部可达密度,降低查询频率的同时减少了时间消耗。

事实上,在真实数据集中离群点往往只占很小的比例。而现有大多密度检测算法都需要计算所有数据点的局部离群因子来进行离群判断,导致了大量的时间消耗。针对这一问题本文提出了一个数据预处理步骤,利用基于互近邻的聚类方法来消除大部分正常数据算法 EMNC(Normal Data Elimination Algorithm Based on MNN Clustering)。根据离群簇和单一离群点的规模大小通常要比正常簇小得多,按顺序去除一定比例大的正常簇后留下来的只有小比例的正常点加上离群点,这大大减少了离群度的计算量。对于剩余的可疑数据充分利用它与其邻域的分布关系,使用 k 近邻、反近邻和共享近邻作为邻域空间去除一定比例的极大极小距离后进行密度估计。本文定义了一个基于局部相对密度的离群度 ROF(Local Relative Density-based Factor)来对剩余的可疑点进行离群判断。最后输出 ROF 值最大的前 n 个数据点。ROF 算法融合了基于密度和基于聚类离群点检测方法的优势,与 LOF、INFLO、SimplifiedLOF、KNN 和 RLDOF 五种方法在合成数据集和真实数据集上的对比实验结果验证了 ROF 算法的有效性。

1　相关工作

LOF(Local Outlier Factor)算法是一种经典的基于密度的局部离群点检测方法。对于一个给定的数据集,正常数据点通常位于数据分布密集的区域,而相反,离群点则位于距离正常数据偏远分布稀疏的区域。LOF 算法对离群点的判断不是通过全局计算,而是基于该点的第 k 邻域局部计算。LOF 算法可以有效地处理一些因为数据点密度分散程度的不同将正常数据点误判为离群点的情况,LOF 算法的一些主要定义如下:

定义 1　第 k 距离(K-distance)和 k 距离邻域(K-distance neighborhood):规定 a 是数据集 D 中的点,$d(a,b)$ 表示 a 和 b 之间的欧氏距离,k 为常数,数据点 a 的第 k 距离 $d_k(a)$ 定义为点 a 与距离它第 k 近的点的距离。点 a 的 k 距离邻域 $N_k(a)$ 定义为 a 的第 k 距离内所有的数据点。即:

$$N_k(a)=\{b\in D \mid d(a,b)\leqslant d_k(a)\} \tag{1}$$

定义 2　可达距离(Reach-distance):点 a 到点 b 的可达距离 $d_k(a,b)$ 定义为点 a 的第 k 距离和点 a 到点 b 的欧氏距离中的最大值,即:

$$d_k(a,b)=\max\{d_k(a),d(a,b)\} \tag{2}$$

定义 3　局部可达密度(Local Reachability Density):点 a 的局部可达密度 $\rho_k(a)$ 可以表示为点 a 到它第 k 邻域内点的平均可达距离的倒数,即:

$$\rho_k(a)=1/\frac{\sum_{b\in N_k(a)} d_k(a,b)}{|N_k(a)|} \tag{3}$$

定义 4　LOF:点 a 的局部离群因子 $LOF(a)$ 定义为点 a 的邻域点 $N_k(a)$ 的局部可达密度与点 a 的局部可达密度之比的平均值,即:

$$LOF(a)=\sum_{b\in N_k(a)}\frac{\rho_k(b)}{\rho_k(a)}/|N_k(a)| \tag{4}$$

可以发现 LOF 值越接近 1,说明点 a 与其邻域点的密度越相似,那么 a 就越不可能是离群点;如果这个比值越小于 1,则说明 a 的密度高于其邻域点的密度,a 为密集点;如果这个比值越大于 1,则说明 a 的密度低于其邻域点的密度,a 就有极大的可能是离群点。在求出每个数据点的局部离群因子之后会对这些值进行从大到小的排序,最终输出值较大的前 n 个数据对象,将它们作为此数据集的离群点集合。

2 算法设计

2.1 算法思想

现有的基于密度的离群点检测算法如 LOF 和 INFLO 等大多都需要寻找所有点的邻居来计算离群因子,然而离群点只占整个数据集很小的比例,所以这种做法的效率是很低的。本文通过一个预处理步骤去掉大部分正常数据,对于剩下的可疑点再根据它们的邻域空间(k 近邻、反近邻和共享近邻)进行密度估计,最终根据这些可疑点的密度与其邻域空间密度平均值的比例筛选出最终的离群点。基于局部相对密度的离群点检测算法 ROF 可以在减少时间消耗的同时提升离群点检测效率。

2.2 基于互近邻聚类的正常数据去除算法

定义 5 反近邻(Reverse Neighbors):一组对象 b 把 a 作为 k 近邻之一,a 的反近邻定义为:

$$RNN_k(a) = \{b \in D \mid a \in N_k(b)\} \tag{5}$$

定义 6 互近邻(Mutual Neighbors):若 b 是 a 的 k 近邻,a 也是 b 的 k 近邻,则规定 a 是 b 的互近邻,同时 b 也是 a 的互近邻。对象 a 的互近邻定义为:

$$MN_k(a) = \{b \mid b \in N_k(a) \wedge a \in N_k(b)\} \tag{6}$$

本文提出了一种新的基于互近邻的聚类来消除大部分正常数据的算法(EMNC)。在寻找 k 近邻时 k 值的确定采用文献[20]中提到的方法,首先规定 k 的初始值是 1,每一次加 1,停止条件是每个对象都被视为邻居或者不被视为邻居的对象数量不再更改,这时得到的 k 值就是算法需要的最优 k 值。通过这种 k 值的确定方法代替了手动调节参数提升了算法效率。

在数据集中随机选取一个点 x,利用最优 k 值找到它的互近邻以及它互近邻的互近邻,这些点构成第一个簇。然后再找一个点重复此步骤找第二个簇,直到所有的点都能属于某一个簇时停止。然后按照簇中数据的个数把所有簇从大到小排序,得到聚类集合 $C = \{C_1, C_2, \cdots, C_n\}$ 满足以下约束条件:

$$|C_1| \geqslant |C_2| \geqslant \cdots \geqslant |C_n| \tag{7}$$

$$|C_1| + |C_2| + \cdots + |C_i| \geqslant t \times |D| \tag{8}$$

$$|C_1| + |C_2| + \cdots + |C_{i-1}| \leqslant t \times |D| \tag{9}$$

则将 $C_{\text{normal}} = \{C_1, C_2, \cdots, C_{i-1}\}$ 称为正常数据的集合,然后将 C_{normal} 去掉不参与后面的离群判断。C 集合中去掉 C_{normal} 后剩余点的集合称为可疑点集合 $C_{\text{doubt}} = \{C_i, C_{i+1}, \cdots, C_n\}$。其中,$t$ 是一个参数值($0 < t < 1$),不同数据集中可以设置不同的 t 值。例如 $t = 0.85$,就说明打算将包含 85% 数据点的簇视为正常簇去掉,而其他簇所包含的就是可疑点。EMNC 详细执行流程如算法 1 所示。

算法 1 基于互近邻聚类的正常数据消除(EMNC)

输入:数据集 D,t。

输出:可疑点集合 C_{doubt}。

1. 初始化 C_n,n = 1
2. 确定最优 k 值
3. 在 D 中随机选取一个对象 a,label(a) = 1
4. $C_n = C_n \cup$ MNS(a)
5. while exist $b \in C_n$ and label(b) ≠1{
6. label(b) = 1
7. $C_n = C_n \cup$ MNS(b)
8. }
9. if exist $m \in D$ and label(m) ≠1{
10. n = n + 1
11. go to line 2
12. }
13. 式(7)、式(8)和式(9)的约束下得到 $C_{\text{normal}} = \{C_1, C_2, \cdots, C_{i-1}\}$
14. 输出 $C_{\text{doubt}} = \{C_i, C_{i+1}, \cdots, C_n\}$

EMNC 是一个简单的聚类算法,聚类结果不一定完全准确,不能仅通过 EMNC 算法就把所有正常数据去除。所以依然需要下文的离群度的判断来找到所有离群点。根据离群簇和单一离群点的规模大小通常要比正常簇小得多,在按顺序去除一定比例大的正常簇后留下来的只是小比例的正常点加上离群点,这大大减少了离群度的计算量。

2.3 基于局部密度的相对离群度

为了更好地估计对象在邻域内的密度分布,对它邻域的选择不仅考虑 k 近邻和反近邻,同时也考虑了共享近邻[17]。

定义 7 共享近邻(shared nearest neighbors):若对象 b 与对象 a 共享一个或多个最近邻,则对象 a 的共享近邻定义为:

$$SNN_k(a) = \{b \in D \mid N_r(a) = N_s(b), \forall r, s \leqslant k\} \tag{10}$$

如图 1 所示,当 $k = 3$ 时,点 a 和点 b 都把点 c 当作它们的邻居之一,这时就说 a 和 b 共享一个近邻 c,b 是 a 的共享近邻,同理 a 也是 b 的共享近邻。

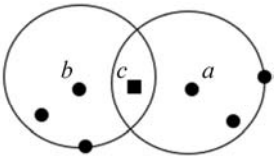

图 1 共享近邻示意图

定义 8 邻域空间(Neighborhood Space):对象 a 的邻域空间定义为 a 的 k 近邻,反近邻和共享近邻的并集,即:

$$NS(a) = N_k(a) \cup RNN_k(a) \cup SNN_k(a) \quad (11)$$

对于某些对象分布异常的情况,邻域空间如果只考虑 k 近邻区域可能会存在离群点误判的情况。有时候反近邻和共享近邻也包含判断离群点的重要信息,所以加入反近邻和共享近邻可以更好地表示对象所在区域的密度分布特性。为了进一步提升离群点检测的准确性,本文采用剔除平均算法[21]来消除对象受它邻域空间距离的极大值和极小值的影响。

定义 9　对象 a 的平均距离 $\overline{d_k(a)}$ 等于 a 到它邻域空间 $NS(a)$ 内所有点距离(剔除后)的平均值,即:

$$d_k(a) = \frac{\sum_{i=r+1, a_i \in NS(a)}^{m-r} d(a, a_i)}{m - 2r} \quad (12)$$

式中:$m = |NS(a)|$,即 m 表示 a 的邻域空间中近邻的个数。规定 $r = \rho \times |NS(a)|$,通常 $\rho \in [0.05, 0.2]$,即在计算点 a 在邻域空间的平均距离时先消除 ρ 比例的极大极小距离。通常正常点和离群点之间的距离较大,去除一定比例的极大距离可以避免离群点对正常点离群判断的影响。同时也会存在这样的离群点它距离正常数据的边缘较近,这个极小距离会影响对离群点的判断,所以也要去除一定比例的极小距离。

定义 10　对象 a 的局部密度 $den(a)$ 定义为平均距离的倒数,即:

$$den(a) = 1/\overline{d_k(a)} \quad (13)$$

定义 11　对象 a 的相对平均距离 $\overline{rd_k(a)}$ 定义为 $NS(a)$ 内所有点的平均距离的均值,即:

$$\overline{rd_k(a)} = \frac{\sum_{i=r+1, a_i \in NS(a)}^{m-r} \overline{d_k(a_i)}}{m - 2r} \quad (14)$$

定义 12　对象 a 的相对平均密度 $rden(a)$ 定义为相对平均距离的倒数,即:

$$rden(a) = 1/\overline{rd_k(a)} \quad (15)$$

定义 13　对象 a 的相对离群度 ROF 定义为相对平均密度和局部密度的比值:

$$ROF = \frac{rden(a)}{den(a)} \quad (16)$$

分析式(16)可知 ROF 值反映的是某数据点的局部密度与其邻域空间的局部密度平均值的比值。ROF 值越接近 1,说明该数据点与其邻域点分布越相似,这个数据点就越不可能是离群点。ROF 值越小于 1,说明该数据点的密度高于其邻域点的密度,则该点为分布密集的点。如果 ROF 值越大于 1,则说明该数据点的密度低于其邻域点密度,该点就有很大可能是离群点。本文采用离群点检测常用的 top-n 思想,即输出前 n 个最大的 ROF 值。算法 2 总结了本文算法的详细流程。

算法 2　基于局部相对密度的离群度(ROF)

输入:数据集 D,t,ρ,r。

输出:离群点集合 $C_{outlier}$。

1. 初始化 C_{doubt},$C_{outlier}$
2. 按算法 1 得到可疑点集合 C_{doubt}
3. for each $a \in C_{doubt}$ do{
4. 　获取 a 的 k 近邻 $N_k(a)$
5. 　获取 a 的反近邻 $RNN_k(a)$
6. 　获取 a 的共享近邻 $SNN_k(a)$
7. 　计算 a 的邻域空间 $NS(a) = N_k(a) \cup RNN_k(a) \cup SNN_k(a)$
8. }
9. for each $a \in C_{doubt}$ do{
10. 　按式(12)计算 a 的平均距离 $\overline{d_k(a)}$
11. 　按式(13)计算 a 的局部密度 den(a)
12. 　按式(14)计算 a 的相对平均距离 $\overline{rd_k(a)}$
13. 　按式(15)计算 a 的相对平均密度 rden(a)
14. 　按式(16)计算 a 的相对离群度 ROF(a)
15. }
16. 对所有对象的 ROF 值降序排序
17. 输出 ROF 最大的前 n 个点的编号及其 ROF 值,$C_{outlier} = \{a_1, a_2, \cdots, a_n\}$

3　实验与结果分析

3.1　合成数据集

为了验证 ROF 算法在分布不同的数据集上离群点检测普遍适用的有效性,本文选取三个不规则的合成数据集来进行对比实验,这三个合成数据集的说明如表 1 所示。

表 1　合成数据集

数据集	样本总数	离群点个数	正常点个数
D1	722	21	701
D2	323	11	312
D3	870	17	853

三个合成数据集的正常点和离群点分布初始图如图 2 所示,离群点是在每个数据集范围内随机生成的。其中圆形点表示正常点,三角形点表示离群点。实验参数的设置:近邻个数 k 是由算法自动得到的,不需要手动调节,三个数据集分别取 $k = 7$、$k = 3$ 和 $k = 5$。正常数据去除比例 t 和极大极小距离去除个数 r 选取了不同参数设置下 20 次实验的最优值。五种对比算法 LOF、SimplifiedLOF、INFLO、KNN、RLDOF 以及本文算法 ROF 的离群点检测结果分别如图 3 – 图 8 所示。

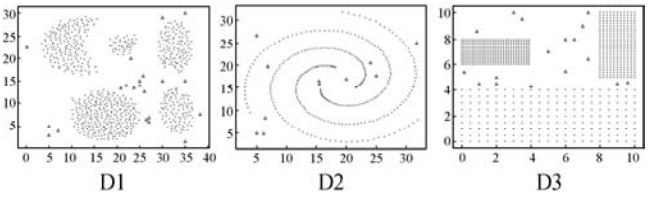

图 2 合成数据集

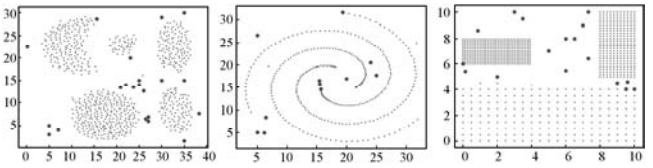

图 3 LOF 检测结果

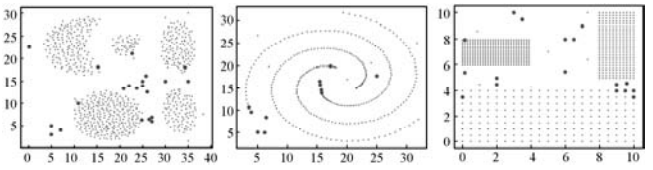

图 4 SimplifiedLOF 检测结果

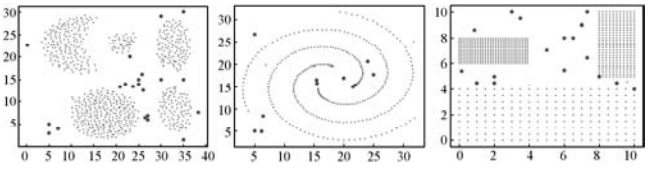

图 5 INFLO 检测结果

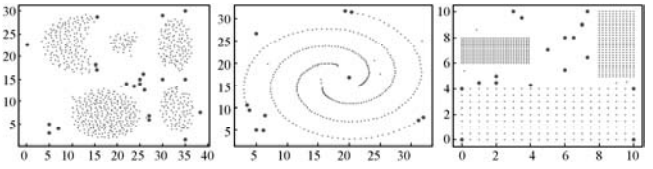

图 6 KNN 检测结果

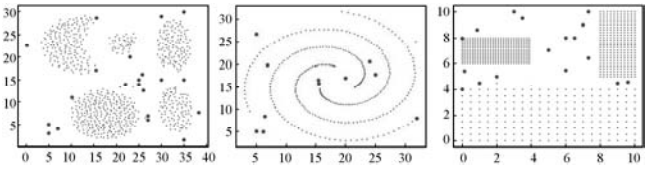

图 7 RLDOF 检测结果

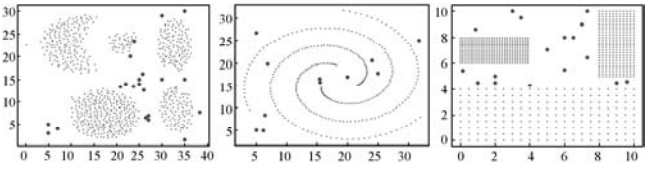

图 8 ROF 检测结果

合成数据集 D1 一共有 21 个离群点,本文的算法 ROF 最高检测到了 20 个正确的离群点,其他五种算法 LOF、SimplifiedLOF、INFLO、KNN 和 RLDOF 最高检测到的正确离群点数量分别为 20、16、21、18 和 18 个。合成数据集 D2 一共有 11 个离群点,本文的算法 ROF 最高检测到全部正确的离群点 11 个,其他五种算法 LOF、SimplifiedLOF、INFLO、KNN 和 RLDOF 最高检测到的正确离群点数量分别为 9、6、9、5 和 10 个。合成数据集 D3 一共有 17 个离群点,本文的算法 ROF 最高检测到全部正确的离群点 17 个,其他五种算法 LOF、SimplifiedLOF、INFLO、KNN 和 RLDOF 最高检测到的正确离群点数量分别为 14、14、15、13 和 15 个。通过对三种合成数据集检测结果分析,不难发现 ROF 算法准确率高的原因在于检测出了一些距离正常簇很近的离群点,而这些点也正是其他对比算法检测效率低的原因。所以 ROF 算法的好处不仅仅在于减少了离群判断的计算量,而且在预处理阶段通过聚类把正常数据聚在一起,然后把规模较大的大部分正常簇去除,有效避免了后续这些正常簇对位于正常簇边缘的离群点判断的影响。

使用准确率 p 对本文提出的算法和五种对比算法进行评价:

$$p = \frac{n_1}{n} \tag{17}$$

式中:n_1 表示算法检测出的离群点数目,n 表示检测出的正确离群点的数目。本文使用了 top-n 思想(n 表示数据集中真正离群点的个数),即每种算法都输出离群度值最大的 n 个点。所以 p 不仅反映了输出结果中离群点检测的准确率,同时也反映了检测出的正确离群点数目占真正离群点数目的比例。

为了更加直观地分析图 3 – 图 8 的实验结果,对六种不同算法在三个合成数据集的离群点检测准确率绘制了一个柱形图,如图 9 所示。可以看出,对于合成数据集 D2 和 D3,本文提出的算法 ROF 检测准确率最高都达到了 100%。对于合成数据集 D1,虽然 ROF 算法准确率(95.24%)略低于 INFLO 算法(100%),但 ROF 依然检测出了大部分离群点,准确率高于其他四种算法。对三个合成数据集综合来看,ROF 的效果是最好的。

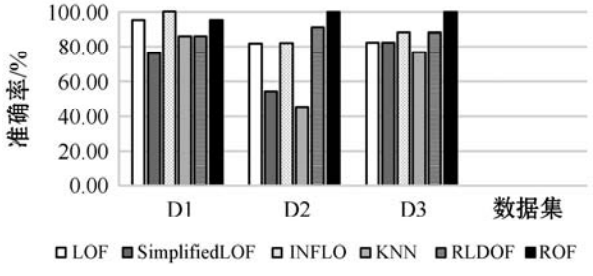

图 9 准确率变化趋势柱状图

3.2 真实数据集

UCI 数据库是一个常见的测试数据集,通常使用它来验证算法的正确性。在 UCI 数据库中由于预先知道所有数据的类别,参照文献[19]数据的处理方法,删除数目少的那个类别的大部分数据,只保留小部分的数据,保留下来的这部分数据距离其他类别的数据距离较远,可以将它们视为离群点。4 个真实数据集

的特征以及处理后的结果如表 2 所示。表 2 的第四列显示了每个数据集的处理过程删除了哪个类,第五列“[ ]”里面表示的是离群点的下标集合,“/”后面表示的是离群点的个数。

表 2　UCI 数据集

名称	属性	样本数	删除类	离群点
Iris1	3	150	Iris-0	[0,1,2,3,4,5,6,7,8,9]/10
Iris2	3	150	Iris-0	[0,1,2,3,4,5,6,7,8,9,10,11,12]/13
Glass	8	214	Glass-5,6,7	[163,164,165,166,167,168]/6
Wine	13	178	Wine-1	[0,1,2,3,4,5,6,7,8,9]/10

为了降低人为设置参数的实验误差,经过了大量的实验,四种 UCI 数据集的检测结果如表 3 – 表 6 所示。表3 – 表6 列举了不同算法输出的离群度最大的前 n 个点的下标,检测正确的离群点下标使用粗体表示。

表 3　Iris1 检测结果

算法	检测结果	比例	p/%
LOF	[**6**,**8**,**4**,**0**,**9**,**3**,**7**,**2**,**5**,**1**]	10∶10	100
SimplifiedLOF	[**4**,**1**,**6**,**8**,**5**,91,77,**3**,**0**,58]	7∶10	70
INFLO	[**6**,**8**,**4**,**0**,**2**,**9**,**7**,**5**,**1**,**3**]	10∶10	100
KNN	[**6**,**8**,**4**,**3**,**0**,**1**,**9**,91,77,78]	7∶10	70
RLDOF	[**5**,91,**9**,**4**,**0**,**7**,**8**,**2**,**1**,**6**]	9∶10	90
ROF	[**8**,**6**,**4**,**5**,**2**,**1**,**0**,**9**,**7**,**3**]	10∶10	100

表 4　Iris2 检测结果

算法	检测结果	比例	p/%
LOF	[**9**,**8**,**2**,**12**,**1**,**7**,**3**,**0**,**4**,**11**,**10**,94,**5**]	12∶13	92.31
SimplifiedLOF	[**8**,**9**,**1**,**7**,**6**,**0**,**5**,**4**,**10**,**3**,**11**,**12**,**2**]	13∶13	100
INFLO	[**9**,**8**,**2**,94,**1**,**12**,**3**,80,**4**,**11**,**5**,**10**,**0**]	11∶13	84.62
KNN	[**6**,**9**,**10**,**5**,**11**,**8**,94,80,81,**2**,85,23,**7**]	8∶13	61.54
RLDOF	[**4**,**5**,**3**,**10**,**11**,**12**,**1**,23,80,94,**2**,**8**,**9**]	10∶13	76.92
ROF	[**11**,**8**,**9**,**7**,**0**,**2**,**5**,**3**,**4**,**12**,**1**,**6**,**10**]	13∶13	100

表 5　Glass 检测结果

算法	检测结果	比例	p/%
LOF	[**163**,**168**,125,**165**,104,**166**]	4∶6	66.67
SimplifiedLOF	[**168**,**163**,115,**165**,**167**,61]	4∶6	66.67
INFLO	[**163**,**168**,**165**,113,126,**167**]	4∶6	66.67
KNN	[126,**163**,**168**,113,79,**166**]	3∶6	50
RLDOF	[**167**,126,**165**,113,**168**,**163**]	4∶6	66.67
ROF	[126,**168**,**163**,**165**,**167**,**166**]	5∶6	83.33

表 6　Wine 检测结果

算法	检测结果	比例	p/%
LOF	[**0**,**9**,**8**,**6**,**4**,**7**,22,43,**2**,**1**]	8∶10	80
SimplifiedLOF	[**7**,**4**,**6**,**8**,**9**,**0**,20,**3**,98,23]	7∶10	70
INFLO	[**0**,**9**,**8**,**6**,29,**5**,41,22,87,**4**]	6∶10	60
KNN	[**0**,**9**,**8**,**6**,**7**,**4**,22,43,**2**,29]	7∶10	70
RLDOF	[**2**,43,41,**7**,29,**4**,**6**,**8**,**9**,**0**]	7∶10	70
ROF	[**0**,**9**,**8**,**6**,**4**,**7**,43,**5**,**2**,113]	8∶10	80

通过对上述表格分析,可以发现六种不同的算法在不同数据集上的检测准确率是不一样的。本文的方法 ROF 在四个 UCI 数据集上都具有较好的效果。对于 Iris1 数据集,ROF、LOF 和 INFLO 检测结果最好都是 100% 的准确率。对于 Iris2 数据集,ROF 和 SimplifiedLOF 检测结果最好都是 100% 的准确率。对于 Glass 数据集,ROF 准确率为 83.33% 高于其他所有数据集。对于 Wine 数据集,ROF 和 LOF 准确率一样高都是 80%。对 UCI 数据库的四个数据集综合来看,实验结果检测准确率排序为 ROF > LOF > INFLO > SimplifiedLOF > RLDOF > KNN。

4　结　语

针对数据集中不需要对所有数据都进行离群判断的问题,提出了一个对数据集预处理的步骤,即使用基于互近邻的聚类算法(EMNC)先去除一定比例的正常数据。对于剩余的可疑数据利用其 k 近邻、反近邻和共享近邻来对其进行密度估计。最后提出一种简单有效的局部相对密度离群度(ROF)来进行离群点的判断。在合成数据集和真实数据集的实验结果也证明了 ROF 算法在离群点检测准确率上优于其他对比算法。在未来的研究中,希望能优化参数选择方法,并使用一种新的距离度量来代替本文使用的欧氏距离。

参　考　文　献

[1] Hawkins D M. Identification of outliers[M]//Monographs on Statistics and Applied Probability. London: Chapman and Hall,1980.

[2] Ro K, Zou C L, Wang Z J, et al. Outlier detection for high-dimensional data[J]. Biometrika,2015,102(3):589 – 599.

[3] Lei J S, Jiang T, Wu K, et al. Robust local outlier detection with statistical parameter for big data[J]. Computer Systems Science and Engineering,2015,30(5):411 – 419.

[4] Radovanović M, Nanopoulos A, Ivanović M. Reverse nea-

rest neighbors in unsupervised distance-based outlier detection[J]. IEEE Transactions on Knowledge and Data Engineering,2015,27(5):1369 - 1382.
[5] 樊瑞宣,姜高霞,王文剑,等. 一种个性化 K 近邻的离群点检测算法[J]. 小型微型计算机系统,2020,41(4):752 - 757.
[6] 王习特,申德荣,白梅,等. 一种高效的分布式离群点检测算法[J]. 计算机学报,2016,39(1):36 - 51.
[7] Jobe J M, Pokojovy M. A cluster-based outlier detection scheme for multivariate data[J]. Journal of the American Statistical Association,2015,110(512):1543 - 1551.
[8] Huang J L, Zhu Q S, Yang L J, et al. A novel outlier cluster detection algorithm without top-N parameter[J]. Knowledge-Based Systems,2017,121(4):32 - 40.
[9] Ester M, Kriegel H P, Sander J, et al. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise[C]//2nd International Conference on Knowledge Discovery & Data Mining,1996:226 - 231.
[10] Huang J L, Zhu Q S, Yang L J, et al. A novel outlier detection algorithm without top-N parameter[J]. Knowledge-Based Systems,2017,121:32 - 40.
[11] Breunig M, Kriegel H P, Ng R T, et al. LOF: Identifying density-based local outliers[J]. ACM SIGMOD Record, 2000,29(2):93 - 104.
[12] Jin W, Tung A K H, Han J W, et al. Ranking outliers using symmetric neighborhood relationship[C]//10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining,2006:577 - 593.
[13] Guan D H, Chen K, Yuan W, et al. A novel density-based outlier detection approach for low density datasets[J]. Journal of Internet Technology,2017,18(7):1639 - 1648.
[14] Terrell G R, Scott D W. Variable kernel density estimation[J]. The Annals of Statistics,1992,20(3):1236 - 1265.
[15] 杜旭升,于炯,陈嘉颖,等. 一种基于邻域系统密度差异度量的离群点检测算法[J]. 计算机应用研究,2020,37(7):1969 - 1973.
[16] 涂晓敏,石鸿雁. 基于方形邻域和裁剪因子的离群点检测方法[J]. 小型微型计算机系统,2019,40(1):188 - 191.
[17] Tang B, He H B. A local density-based approach for outlier detection[J]. Neurocomputing,2017,241:171 - 180.
[18] 谢兄,唐昱. 基于局部估计密度的局部离群点检测算法[J]. 小型微型计算机系统,2020,41(2):387 - 392.
[19] Müller E, Schiffer M, Seidl T, et al. Statistical selection of relevant subspace projections for outlier ranking[C]//27th International Conference on Data Engineering,2011:434 - 445.
[20] Dai Q Z, Xiong Z Y, Xie J, et al. A novel clustering algorithm based on the natural reverse nearest neighbor structure[J]. Journal of Information System,2019,84:1 - 16.
[21] Hu T M, Sung S Y. A trimmed mean approach to finding spatial outliers title[J]. Intelligent Data Analysis,2004,8(1):79 - 95.
[22] 王晓辉,宋学坤,王晓川. 基于邻域密度的异构数据局部离群点挖掘算法[J]. 计算机仿真,2021,38(7):281 - 285.
[23] 张倩倩,于炯,李梓杨,等. 基于近邻传播的离群点检测算法[J]. 计算机应用研究,2021,38(6):1662 - 1667.
[24] 梅林,张凤荔,高强. 离群点检测技术综述[J]. 计算机应用研究,2020,37(12):3521 - 3527.

(上接第 260 页)

[4] 回天,哈力旦·阿布都热依木,杜晗. 结合 Faster R-CNN 的多类型火焰检测[J]. 中国图象图形学报,2019,24(1):73 - 83.
[5] 任嘉锋,熊卫华,吴之昊,等. 基于改进 YOLOv3 的火灾检测与识别[J]. 计算机系统应用,2019,28(1):171 - 176.
[6] 任锴,陈俊,叶宇煌,等. 基于 SSD-MobileNet 的火情检测预警系统[J]. 电气开关,2020,58(1):34 - 38.
[7] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Computer Vision & Pattern Recognition,2016.
[8] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger[C]//IEEE Conference on Computer Vision & Pattern Recognition,2017:6517 - 6525.
[9] Redmon J, Farhadi A. YOLOv3: An incremental improvement[EB]. arXiv:1804.02767,2018.
[10] Bochkovskiy A, Wang C, Liao H. YOLOv4: Optimal speed and accuracy of object detection[EB]. arXiv:2004.10934, 2020.
[11] Zagoruyko S, Komodakis N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer[EB]. arXiv:1612.03928,2016.
[12] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,2018.
[13] 孙世华. 基于相对熵和 ESD 检测的视频关键帧抽取算法[D]. 天津:天津大学,2017.
[14] 宋勇强. VC6 中利用多线程处理多视频[J]. 山西电子技术,2019(6):64 - 66.
[15] Dollar P, Wojek C, Schiele B, et al. Pedestrian detection: A benchmark[C]//2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR),2009:304 - 311.
[16] 耿梦雅. 基于视频的复杂场景火灾检测技术研究[D]. 武汉:华中师范大学,2019.
[17] 袁梅. 基于视频图像序列的火灾烟雾检测方法研究[D]. 重庆:重庆邮电大学,2019.