一种基于谱嵌入和局部密度的离群点检测算法

李长镜[1] 赵书良[1] 池云仙[2]
(河北师范大学数学与信息科学学院 石家庄 050024)[1]
(河北师范大学资源与环境科学学院 石家庄 050024)[2]

摘 要 离群点检测问题是数据挖掘领域的研究热点之一。现有的检测算法主要应用于离群点位于初始属性子空间或底层子空间各种线性组合等情况,当离群点嵌入局部非线性子空间时,进行离群点有效检测的难度很大。为此,文中分析了典型的谱嵌入算法在离群点检测上存在的不足,然后以局部密度为基础,提出了一种基于谱嵌入和局部密度的离群点检测算法。该算法采用迭代策略对不重要的特征向量进行高效筛查,以发现有助于检测出局部非线性子空间离群点的特征向量,并利用上一次迭代获得的基于局部密度的谱嵌入结果来改进下一次迭代的相似度图,经过多次迭代可以将离群点从正常点中分离。仿真实验结果表明,所提算法的检测精度优于当前其他典型算法,且该算法对参数的设置不敏感。

关键词 离群点检测,谱嵌入,局部密度,迭代策略,相似度图,检测精度
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.03.039

Outlier Detection Algorithm Based on Spectral Embedding and Local Density

LI Chang-jing[1] ZHAO Shu-liang[1] CHI Yun-xian[2]
(College of Mathematics and Information Science, Hebei Normal University, Shijiangzhuang 050024, China)[1]
(College of Resources and Environmental Science, Hebei Normal University, Shijiangzhuang 050024, China)[2]

Abstract Outlier detection is one of the hot topics in the field of data mining. The existing detection algorithms are mainly applied to the cases where outliers lie in initial attribute subspace or various linear combinations of underlying subspace, when the outliers are embedded in local nonlinear subspace, it is very difficult to detect the outliers effectively. To solve this problem, the shortcomings of typical spectral embedding algorithm for outlier detection were firstly analyzed, and then on the basis of local density, an outlier detection algorithm based on spectral embedding and local density was proposed. The algorithm which uses iterative strategy can efficiently screen unimportant eigenvectors and discover eigenvectors that are relevant for finding outliers hidden in local non-linear subspaces, and the local density-based spectral embedding from a previous iteration is used for improving the similarity graph for the next iteration, such that outliers are gradually segregated from inliers during these iterations. The simulation results show that the detection accuracy of the proposed algorithm is better than other typical algorithms, and it is not sensitive to the parameter setting.

Keywords Outlier detection, Spectral embedding, Local density, Iterative strategy, Similarity graph, Detection accuracy

离群点检测[1]被广泛应用于信用卡欺诈检测、网络流量入侵检测、视频监控异常行为检测等领域,成为了数据挖掘领域的一项重要研究课题。离群点检测是指发现严重偏离数据总体分布范围的离群数据[2-3]。离群数据由于与数据总体分布情况不同,因此可以被看作是可疑数据。例如信用卡诈骗检测问题,数据集包括卡片主人的交易信息,交易记录记载了每个用户消费行为的卡片使用情况。如果卡片被盗,用户消费行为往往会发生变化。如果交易记录显示消费额度高、消费频率高、消费项目重复,则可认定出现异常消费模式。

目前已有多种离群点检测算法。文献[4]针对支持向量数据描述(Support Vcetor Data Description, SVDD)的训练集中同时含有正常点和离群点的问题,为减弱离群点对SVDD训练模型的不利影响,提出了一种基于单簇核可能性C-均值的SVDD离群点检测算法。文献[5]提出了一种基于数据低维子空间的离群点检测算法。文献[6]提出的LOF算法利用局部密度实现了离群点检测。文献[7]对高维数据进行多变量极值分析,提出了基于角度的离群点检测算法。然而,上述算法主要针对的是离群点位于初始属性子空间或者底层子空

到稿日期:2018-02-11 返修日期:2018-05-28 本文受国家自然科学基金(71271067),国家社科基金重大项目(13&ZD091),河北省高等学校科学技术研究项目(QN2014196)资助。

李长镜(1990—),男,硕士生,主要研究方向为数据挖掘、大数据;**赵书良**(1967—),男,博士,教授,主要研究方向为数据挖掘、智能信息处理,E-mail:zhaoshuliang@sina.com(通信作者);**池云仙**(1987—),女,博士生,主要研究方向为数据挖掘、地理信息处理。

间各种线性组合的情况，当离群点嵌入局部非线性子空间时，进行离群点有效检测的难度仍然很大。在实践中，数据往往位于具有随机形态的低维流形周围，如图 1 所示。很显然，数据点 A 和数据点 B 为离群点，且 A 和 B 的数据模式的形态有显著差异，只有采用非线性映射才能将 A 和 B 所在位置相应数据模式的形态投影到低维度上。这些离群点位于数据的低维子空间内，隐藏在局部非线性子空间中，采用当前的子空间算法难以进行有效检测。如果底层数据点的局部密度发生显著变化，进行离群点检测的难度将更大。因此，为了有效检测离群点，既需要考虑相应非线性子空间的形态，又需要考虑底层密度的变化。为此，文献[5]提出了一种谱算法 OutDST，该算法可检测出嵌入数据及具有任意形态的低维聚类。然而，虽然 OutDST 算法可在一定程度上根据发生变化的局部子空间和密度做出自适应调整，但是离群点得分的计算仍会严重失真，且 OutDST 算法的性能对数据集和参数的设置较为敏感。针对上述算法的不足，本文提出一种基于谱嵌入和局部密度的离群点检测算法，该算法将传统的谱嵌入算法与基于局部密度的检测算法相结合来提升离群点的检测性能。最后的仿真实验结果也验证了所提算法的有效性。

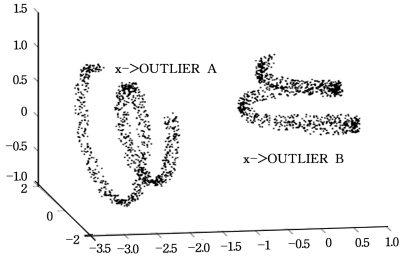


图 1　嵌入在各种流形中的离群点

Fig. 1　Outliers embedded in manifold

1　谱嵌入

迄今为止，人们对数据聚类尤其是非凸聚类条件下的谱嵌入问题进行了深入研究[9-10]，通过谱嵌入可以降低变换空间中离群点检测的难度。本文首先对典型的谱嵌入算法进行了分析，针对这些算法无法直接用于离群点检测的不足，提出了一种可用于离群点检测的谱嵌入算法。

1.1　数据点的谱嵌入

假设有由 m 个数据点构成的集合 $X=\{x_1,\cdots,x_m\}$，各个数据点为 n 维向量($x\in\Re^n$)。在变换空间中将为数据点分配新坐标的这一过程称为数据点的嵌入。目前最为典型的谱嵌入算法有 SM 算法[9]和 NJW 算法[10]。

在 SM 算法中，谱嵌入计算的第 1 步是构建由索引对(i，j)组成的集合 R。当且仅当数据点 x_i 和 x_j 互为各自的 k 近邻时，索引对(i，j)属于 R。如果 x_i 和 x_j 均在对方的 k 个最相似点集合内，则 x_i 和 x_j 均互为各自的 k 近邻。x_i 和 x_j 之间的相似度为 w_{ij}，其由热核方法[11]计算得到。集合 R 中的元素可看作加权无向图的边，边的权重即为相似度 w_{ij}，该图称为邻域图。

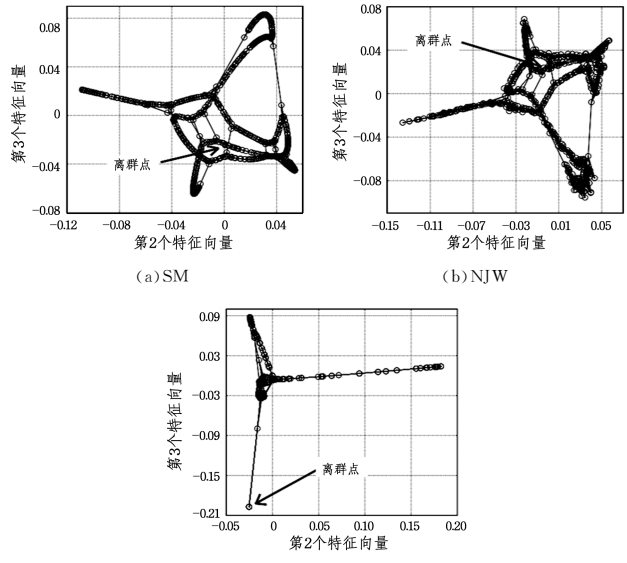
第 2 步，将数据点 $x_1,\cdots,x_m$ 映射到向量 u。如果 x_i 和 x_j 非常相似，则映射后得到的元素 u_i 和 u_j 也非常邻近。通过求解如下优化问题，即可实现上述目的。

$$\begin{aligned}&\underset{u}{\text{minimize}}\ O=\sum_{(i,j)\in R}w_{ij}(u_i-u_j)^2\\&s.t.\ \ u^{\mathrm{T}}Du=1\end{aligned}\tag{1}$$

其中，D 是一个对角矩阵，称为度矩阵。其中的每个对角元素 $d_i=\sum_{j=1}^{m}w_{ij}$ 称为点 x_i 的度。权重 w_{ij} 形成 k 最近邻(KNN)矩阵 W。式(1)中的目标函数可表示为 $O=2u^T(D-W)u$，其中矩阵 $L=D-W$ 表示邻域图的拉普拉斯算子，它是一个特征值非负的半正定矩阵。矩阵 $D^{-1}L$ 的特征向量即为式(1)的解，特征向量对应的特征值越小，该特征向量表示的解就越优。将数值最小的 h 个特征值对应的特征向量 $u_1,\cdots,u_h$ 存储在矩阵 $U\in\Re^{m\times h}$ 中，便可计算出 h 维嵌入。矩阵 U 的每一行表示将初始点 x_i 映射为点 $y_i\in\Re^h$，点 y_i 构成点 x_i 的谱嵌入。矩阵 $D^{-1}L$ 和 $D^{-1}W$ 分别称为正规化拉普拉斯算子和 KNN 矩阵。而在 NJW 嵌入方法中，正规化拉普拉斯算子和 KNN 矩阵分别表示为 $D^{-1/2}LD^{-1/2}$ 和 $D^{-1/2}WD^{-1/2}$。SM 方法的正规化矩阵非对称，NJW 方法的正规化矩阵对称。

1.2　SM 算法和 NJW 算法的不足

SM 算法和 NJW 算法存在的主要不足是它们获得的嵌入不适用于离群点检测。根据 SM 算法和 NJW 算法中谱嵌入的常规做法，分别在图 2(a)和图 2(b)中绘制出了 $k=6$ 时 SM 算法和 NJW 算法的正规化拉普拉斯算子的第 2 个和第 3 个特征向量，箭头指向离群点的嵌入位置。



(a)SM　　(b)NJW

(c)本文方法

图 2　不同谱嵌入方法的离群点检测效果

Fig. 2　Outlier detection results of different spectral embedding methods

可以看到，在这两种算法获得的嵌入中，离群点与正常点混淆在一起，即使在谱嵌入过程中获得了更多的特征向量，这一问题仍然没有得到缓解，主要原因如下：

1)设 KNN 矩阵中非对角元素为 $\bar{w}_{ij}$，根据 SM 算法和 NJW 算法，有：

$$\begin{aligned}&\text{SM}:\bar{w}_{ij}=\frac{w_{ij}}{d_i}\\&\text{NJW}:\bar{w}_{ij}=\frac{w_{ij}}{\sqrt{d_id_j}}\end{aligned}\tag{2}$$

其中，$\bar{w}_{ij}$ 的数值越大，点 x_i 和 x_j 嵌入后的距离就越小。SM 算法和 NJW 算法中的正规化 KNN 矩阵不能准确地衡量样本的局部分布，当离群点形成样本少但密度高的簇时，这些算法会错误地赋予离群点较高的置信度。

2)在 SM 算法和 NJW 算法中，将离群点与正常点连接起来的边的权重往往非常小。为了进行离群点检测，需要采用某种方法对数据点进行正规化处理，如果获得的正规化因子的数值较小，则有助于提高将离群点与正常点连接起来的边的权重。对于 SM 算法和 NJW 算法而言，当采用典型的基于度的正规化方法[12]时，即使是将离群点与正常点连接起来的边，其权重也没有显著差异，因此即使对这两种算法进行多次迭代也无益于离群点的检测。

2 基于局部密度的谱嵌入

为了解决上述问题，本文提出了一种可广泛应用于局部非线性子空间中离群点检测的谱嵌入方法(LODES)。该方法专门针对离群点检测需求设计，以局部密度概念为基础，可以经过多轮次迭代将离群点从正常点中分离。为了便于描述 LODES 方法，首先给出如下 3 个定义。

定义 1(局部密度) 数据点 x_i 的局部密度是指在其邻域内的数据点的密度。本文利用点的度 d_i 表示局部密度，数据点的度越高，局部密度就越高。

定义 2(关联密度) 已知两个数据点 x_i 和 x_j，关联密度是指这两个数据点间区域的密度。本文采用数据点相似度 w_{ij} 表示关联密度。边的权重越高，其越可能出现在密集区域。

定义 3(对称密度差值) 已知两个数据点 x_i 和 x_j，对称密度差值 q_{ij} 是指数据点 x_i 和 x_j 局部密度差值的平方。

基于上述定义，首先对 SM 算法中的 KNN 矩阵进行改进，提出一种基于关联密度与对称密度差值的正规化 KNN 矩阵 $\overline{W}^{(\ell)}$ 用于离群点检测。该矩阵的元素 $\bar{w}_{ij}$ 为：

$$\bar{w}_{ij}=\frac{\text{关联密度}}{\text{对称密度差值}}=\frac{w_{ij}}{q_{ij}}=\frac{w_{ij}}{(d_i-d_j)^2} \tag{3}$$

从式(3)可以看到，如果关联密度较小，对称密度差值较大，则表明点 x_i 和 x_j 来自局部密度差异较大的区域，映射后应该互相远离。如果关联密度较大，对称密度差值较小，则表明点 x_i 和 x_j 来自密度较大且局部密度相类似的区域，映射后应该互相靠近。

然后，利用正规化 KNN 矩阵 $\overline{W}^{(\ell)}$ 计算度矩阵 $\overline{D}^{(\ell)}$。相应的拉普拉斯算子为：

$$\overline{L}^{(\ell)}=\overline{D}^{(\ell)}-\overline{W}^{(\ell)} \tag{4}$$

此外，还需要计算 $\overline{L}^{(\ell)}$ 的前 r 个特征向量。与前文所述类似，与初始数据点 x_i 相对应的行 y_i 便是 x_i 在嵌入空间的新坐标。与初始点 x_i 相比，嵌入点 y_i 可将离群点与正常点更清晰地区分开。基于局部密度计算数据点 x_i 的嵌入被称为局部致密过程。然而，相比于普通的谱嵌入方法，基于局部密度的谱嵌入方法即使具有显著的优势，仍然没有充分利用局部密度之间的差异。下文将讨论如何利用这一重要特性对局部致密过程进行不断的精炼，进而充分发挥该方法的最大优势。在局部致密过程中进行多次迭代后，最后一次迭代获得的谱嵌入将用于计算离群点的得分。

3 LODES 算法

为实现离群点的检测并为其分配分值，一种简单的方法是将前节描述的基于局部密度的嵌入方法与 K 最近邻方法相结合。虽然这种方法可以更清晰地检测出部分离群点，但仍然有部分离群点未被检测出来。局部密度在一定程度上使谱嵌入方法更适用于离群点检测，但现有的算法只能做到局部优化[13]。实际上，离群点检测面临的主要问题是嵌入方法中的部分特征向量有可能掩盖了部分离群点的存在，于是，非常明显的离群点的存在经常会对普通离群点的检测产生干扰。因此，有必要将这些特征向量与其他特征向量区分开，以提高离群点检测的准确性。本文重点关注两类特殊的特征向量。

1)稀疏特征向量。这些特征向量的大部分为 0 元素，少部分为非 0 元素。非 0 元素对应的数据点与邻域图中其他所有数据点分离，因此往往会被标记为显著离群点。以如下稀疏特征向量为例：$u_2=(0.5,0,0,0,0)^{\mathrm{T}}$，该特征点表明数据点 x_1 为离群点。因为该特征向量包含离群点的有用信息，因此 LODES 方法特地保留这些特征向量，并将其用于离群点的打分阶段。然而，如果存在这种特征向量，将会导致我们无法获得包含所有离群点的低维嵌入，当这种特征向量的数量较多时尤其如此。

2)低基数特征向量。这些特征向量包含少量不同数值。直观来说，这些特征向量表示少量不同的数据点集合。这些特征向量将数据点嵌入集合，且集合中具有相同特征向量元素的所有点属于同一个集合。这种特征向量主要针对数据聚类进行优化，基本无法检测出底层数据中的离群点。以特征向量 $u_2=(0.5,0.5,-0.2,-0.2,-0.2)^{\mathrm{T}}$ 为例，该特征向量包含两个不同的数据 0.5 和 -0.2，因此两个集合的大小分别为 2 和 4。该特征向量没有为底层数据离群点的检测提供大量的有用信息。

相比于低基数特征向量，稀疏特征向量可以完整地表示少部分数据点集合与邻域图中其余数据点之间的关联，因此稀疏特征向量的特征值更小。下文将讨论 LODES 算法如何有效地处理这两种特征向量。

3.1 迭代方法

LODES 的基本迭代策略是在首次迭代时计算邻域图，然后在后续迭代中不断调整边的权重。根据先前迭代过程中各点之间的距离来调整边的权重，其目的是不断检测出离群点，同时将先前迭代时获得的有用特征向量存储起来。为此，在重新计算权重之前删除稀疏特征向量和低基数特征向量。需要注意的是，只有邻域图的权重在迭代期间发生改变，邻域图中的边在迭代期间始终保持不变。

3.1.1 迭代过程

LODES 算法的伪代码如算法 1 所示，首先描述 LODES 方法的首次迭代，然后讨论如何利用某次迭代获得的有用信息在后续迭代中检测出离群点。迭代轮次用变量 t 表示。首先估计计算相似度时用到的内核的宽度(见算法 1 中的第 4 行)。采用如下经验准则进行估计，随机采样 P 对数据点，并计算：

$$\sqrt{\frac{1}{|P|}\sum_{(i,j)\in P}\|x_i-x_j\|_2^2} \tag{5}$$

将该数值作为带宽 α 的估计，然后基于其计算相似度及相应的 KNN 矩阵 W_1。需要注意，迭代次数表示为矩阵的下标。例如，W_1 表示首次迭代时的 KNN 矩阵，即 $t=1$。然后，根据第 2 节所述内容计算经过局部致密和正规化的拉普拉斯算子 $\overline{L_1}^{(\ell)}$（见算法 1 中的第 12 行），计算得到 $\overline{L_1}^{(\ell)}$ 的特征向量矩阵 U_t（见算法 1 中的第 13 行），并将矩阵 U_t 各列按照特征向量升序排列。

算法 1　基于谱嵌入和局部密度的离群点检测算法(LODES)

输入：数据点 X，共同 KNN 的数量 k，稀疏阈值和基数阈值(δ,τ)，特征向量窗口尺寸 r，迭代次数 T

输出：离群点得分$\{c_1,\cdots,c_m\}$

1. for t=1 to T do
2.　if t=1 then
3.　　$\hat{X}\leftarrow X$,a←2;
4.　　利用式(5)计算$\hat{X}$的带宽 α;
5.　　用 W_t 表示带宽为 α 的$\hat{X}$的 KNN 矩阵;
6.　else
7.　　$\hat{X}\leftarrow U_{t-1}(:,a:b)$;
8.　　利用式(5)计算$\hat{X}$的带宽 σ;
9.　　用 $W_t^{(sim)}$ 表示带宽为 σ 的$\hat{X}$的所有两两相似度;
10.　$W_t\leftarrow W_t^{(sim)}\circ W_{t-1}$;
11.　end if
12.　$\overline{L_t}^{(\ell)}\leftarrow$根据第 2 节计算 W_t 的拉普拉斯算子;
13.　$U_t\leftarrow$计算$\overline{L_t}^{(\ell)}$的特征向量矩阵;
14.　(a,$\overline{\Re}$)←handlesparsity(U_t,a,δ);
15.　b←handleLowCardinality(U_t,a,τ,r);
16.　$\Re\leftarrow\Re\cup\overline{\Re}$;
17. end for
18. $\{c_1,\cdots,c_m\}\leftarrow$outlierScore($U_T(:,a:b)$,k);
19. 将 $\mathcal{R}$ 中点的得分设置为 max($\{c_1,\cdots,c_m\}$);
20. return $\{c_1,\cdots,c_m\}$;
21. function handleSparsity(U,a,δ)
22.　$\hat{a}\leftarrow$a,$\Re\leftarrow\emptyset$;
23.　for j=a to m do
24.　　if $|u_j\neq 0|\leqslant m\cdot\delta$ then
25.　　　$\hat{a}\leftarrow$j+1;
26.　　　$\mathcal{R}\leftarrow\mathcal{R}\cup\{i|u_{ij}\neq 0\}$;
27.　　end if
28.　end for
29.　return ($\hat{a}$,$\Re$);
30. end function;
31. function handleLowCardinality(U_t,a,τ,r)
32.　b←a,v←0;
33.　　while v<r and b<m do
34.　　　b←b+1;
35.　　　if $|u_b|_{\neq}>m\cdot\tau$ then
36.　　　　v←v+1;
37.　　　end if
38.　　end while
39.　　return b;
40. end function.

此外，算法 1 选择 r 个特征向量来解决稀疏特征向量和低基数特征向量问题。稀疏特征向量总是属于 U_1 中最左侧的那部分特征向量，因此只需要增加延续到下一次迭代的最左侧特征向量（矩阵 U_1 中）的列号 a，即可将这些特征向量删除。利用阈值 $\delta\in(0,1)$ 确定最左侧特征向量的数量，如果最左侧开始有连续多个特征向量包含的元素数量少于 $\delta\cdot m$，则认为这些特征向量是稀疏的。适当增大 a 的数值，以便保证 u_a 非稀疏。为此，我们设计了 *handleSparsity*()函数，在算法 1 中的第 14 行调用，具体内容见算法 1 中的第 22－31 行。然后，从左到右扫描 U_1 的列号，确定数值 b，以保证 $u_a,\cdots,u_b$ 中至少有 r 个特征向量为非低基数特征向量。如果用户指定的阈值 $\tau\in(0,1)$，特征向量中不同数值的数量低于 $\tau\cdot m$，则该特征向量为低基数特征向量。为此，我们设计了 *handleLowCardinality*()函数，在算法 1 中的第 15 行调用，具体内容见算法 1 中的第 33－42 行。实际上，稀疏阈值和基数阈值保持在 4%～6%的低值水平，可保证 LODES 算法的正常运行。另外，即使这些阈值在设计时有略微出入，LODES 算法也可有效运行。

3.1.2　相似度更新

在后续迭代中，利用先前迭代时矩阵 W_{t-1} 获得的特征向量来重新调整权重矩阵 W_t，利用先前迭代时谱嵌入确定的特征向量 $u_a,\cdots,u_b$ 来计算相似度矩阵 $W_t^{(sim)}$，其中元素 (i,j) 表示利用相同热核方法获得的嵌入点 i 和 j 之间的相似度。利用这些相似度数值来更新先前迭代时获得的初始 KNN 矩阵 W_{t-1}，从而确定新的矩阵 W_t 如下：

$$W_t=W_t^{(sim)}\circ W_{t-1} \tag{6}$$

其中，标记$\circ$表示哈达玛乘积[14]。两个矩阵的哈达玛乘积表示两个矩阵中的元素逐个相乘。计算哈达玛乘积的目的是改变 KNN 矩阵 W_{t-1} 中各个元素的相对数值，以便更准确地反映先前迭代获得的谱嵌入的相似度。同时，邻域图中的边保持不变，只改变嵌入数据点之间的相似度。上述相似度更新步骤对特征向量进行了处理，邻域图中的节点在迭代之后有可能相距更近，也有可能相距更远。该方法提高了致密过程以及局部非线性子空间中离群点的检测和分离速度。

3.2　离群点得分

LODES 算法的最后一步是计算每个数据点对应的离群点得分 c_i，本文利用最后一次迭代获得的谱嵌入 U_T 来计算得分。对于根据 U_T 进行嵌入的每个点 y_i，计算它与其第 j 个最近邻之间的距离 p_j（$j\in\{1,\cdots,k\}$）。对于 $j\in\{1,\cdots k\}$，计算 $\Delta j\leftarrow p_j-p_{j-1}$ 和 $\Delta_j^{\max}=\max_{s=1}^{j}\Delta s$。当 $j=1$ 且 $\Delta_j=p_1-p_0$ 时，p_0 为 0。x_i 的离群点得分即为 $\Delta_1^{\max},\cdots,\Delta_k^{\max}$ 的算术均值。具体过程见算法 2。因为谱嵌入方法经过改进后可根据离群点的局部密度差异及特征向量的后续处理更有为效地将离群点与正常点区分开来，所以计算得分的方法非常有效。

算法 2　outlierScore()离群点分值计算

输入：谱嵌入 U_T，最近邻数量 k

输出：离群点得分$\{c_1,\cdots,c_m\}$

1. for i=1 to m do

2. 将 $\Delta_1^{max},\cdots,\Delta_k^{max}$ 初始化为 0;
3. for j=1 to k do
4. $p_j \leftarrow U_T$ 中 y_i 的第 j 个最近邻;
5. $\Delta j \leftarrow p_j - p_{j-1}$;
6. $\Delta_j^{max} \leftarrow \max_{s=1}^{j} \Delta_s$;
7. end for
8. $c_i \leftarrow \frac{\sum_{j=1}^{k} \Delta_j^{max}}{k}$;
9. end for
10. return $\{c_1,\cdots,c_m\}$.

3.3 计算复杂度分析

本节分析 LODES 算法的计算复杂度。就 LODES 算法的一次迭代而言,计算 KNN 矩阵涉及到($k-d$)树索引,其代价为 O($nm\log m$)。构建矩阵的稀疏表示、KNN 矩阵和拉普拉斯算子需要的时间代价为 O(mk)。包含对 km 个非 0 元素的稀疏拉普拉斯算子矩阵进行特征分解时需要的时间代价 O($(m+km)g+g^2$),其中 g 表示用户定义的远小于 m 的整数。因此,LODES 算法单次迭代时所需的时间主要取决于稀疏矩阵中的元素个数 O(km)。对于 T 次迭代,LODES 算法的总计算时间代价为 O(Tkm),是可以接受的。

4 实验评估

本节对 LODES 算法进行了全面的实验分析,并将其与其他最新的离群点检测算法进行了比较。4.1 节描述数据集;4.2 节描述性能基准;4.3 节结合性能基准来评估 LODES 算法的精度;4.4 节分析算法精度对于参数 k 的敏感性。

4.1 数据集

从 UCI 机器学习数据库中获得了 11 个真实数据集[1)]。这些数据集的重要特性如表 1 所列。对于类别不均衡的数据集来说,将主流类别标记为正常点,将非主流类别标记为离群点。对于类别较为均衡的数据集来说,通过对主体类别进行均衡下采样来生成非主流类别。限于篇幅,文中只给出了 Glass,Pendigits,Ecoli 和 Vowels 4 个类别的详细结果,将这 4 个集合称为基本数据集。

表 1 数据集总体情况

Table 1 General situation of datasets

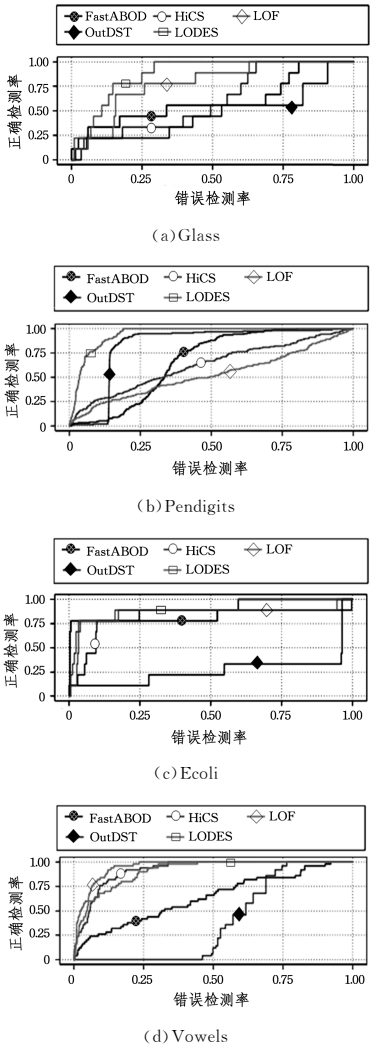
数据集	属性	离群点比例/%
Glass	9	4.2
Pendigits	16	2.2
Ecoli	7	2.6
Vowels	12	3.4
Cardio	21	9.6
Wine	13	7.7
Thyroid	6	2.4
Vertebral	6	12.5
Yeast	8	4.7
Seismic	11	6.5
Heart	14	4.4

4.2 性能基准

基准算法包括最新的基于密度的检测算法(LOF[6])、基于角度的检测算法(FastABOD[7])、子空间搜索算法(HiCS[6])以及谱算法(OutDST[8])。LODES 算法和 OutDST 算法利用带宽 α 构建 KNN 矩阵,利用式(5)估计其数值。HiCS 算法和 OutDST 算法还需要其他参数,均采用默认设置。具体来说,对于 HiCS 算法,有 $M=50$,$\alpha=0.1$,$candidate\_cutoff=400$。OutDST 算法需要预先知道数据集中离群点的比例,并将该参数作为算法的输入。OutDST 算法的参数 γ 的取值为 0.5~0.8,确定 γ 的准则是使检测出的离群点数量与数据集中离群点的实际数量之差最小化,利用可使数量之差最小的 γ 生成最终结果。最后,LODES 算法的默认设置为:$k=10$,$r=2$,$\tau=1\%$,$\gamma=2\%$,算法均迭代运行 50 次,所有算法采用相同的 k 值。除另外说明外,上述参数均采用默认设置。

4.3 精度比较

本节将 LODES 算法与其他基准算法的性能进行比较。LODES 算法的参数采用上文描述的默认设置。利用 ROC 曲线确定正确检测率与错误检测率之间的关系,精度指标即为 ROC 曲线下方的面积(AUC)。图 3 显示了多种算法在不同数据集中的 ROC 曲线。



(a)Glass

(b)Pendigits

(c)Ecoli

(d)Vowels

图 3 不同算法的 ROC 曲线比较

Fig. 3 Comparison of ROC curves of different algorithms

可以看出,LODES 算法的 ROC 远优于基准算法,其主要

1) https://archive.ics.uci.edu/ml/datasets.html

原因在于 LODES 算法综合采用了基于局部密度的谱嵌入策略及更为先进的特征空间搜索策略来对特征向量进行处理，有效地提升了离群点的检测性能。

仔细观察图 3 的曲线可以发现，虽然 OutDST 算法与本文的 LODES 算法都属于谱算法，但对种类各异的数据集的性能仍不理想，其原因在于 LODES 算法对迭代式谱方法进行了针对性改进并采用基于局部密度的谱嵌入策略进行了离群点分析。另外，有必要分析 LoF 算法、HiCS 算法和 LODES 算法在 Vowels 数据集上的表现。图 3(d)给出了不同算法在 Vowels 数据集上的 ROC 曲线。可以看到，虽然 HiCS 算法和 LoF 算法在该数据集上的 AUC 性能分别比 LODES 算法高出 1%和 3.5%，但 LODES 算法对离群点的检测时间远少于 HiCS 算法。LODES 算法的 ROC 指标在开始时增长速度较快，这是因为通过稀疏特征向量及谱嵌入方法的连续局部致密过程，离群点的检测效果得到了有效提升。LODES 算法的 ROC 指标对 Pendigits 数据集也有类似的表现，如图 3(b)所示。

表 2 列出了所有算法对所有数据集的 *AUC* 表现，排名前两位的 *AUC* 用黑体表示。对于某一个数据集，只有性能最优的两种算法用黑体标记。除了 Seismic 数据集和 Vowel 数据集，LODES 算法的 *AUC* 表现均排在前两名。对于 Glass 等数据集，LODES 算法的性能远优于排名第 2 的算法，LODES 算法与排名第 1 的算法的差异也非常小。总体来说，与其他算法相比，LODES 算法在不同数据集下的表现非常稳定，这表明本文算法能有效检测其他算法无法有效检测的高维流形离群点，同时，本文方法也能有效检测其他算法能够有效检测的离群点。将局部密度和谱嵌入技术相结合，确实提高了多种真实数据集中离群点的检测效果。

表 2　不同算法在所有数据集上的 *AUC* 比较

Table 2　*AUC* comparison of different algorithms on all datasets

数据集	LODES	HiCS	OutDST	FastABOD	LOF
Glass	87.32	60.40	46.61	60.05	78.26
Pendigits	94.40	62.42	82.89	66.39	52.55
Ecoli	89.29	81.34	26.50	87.32	86.30
Vowels	91.14	92.17	40.01	63.67	94.67
Cardio	72.08	63.02	30.78	94.52	59.67
Wine	96.60	48.50	92.43	82.50	62.18
Thyroid	**68.40**	**76.82**	51.20	55.58	67.14
Vertebral	**58.20**	56.60	48.20	34.80	**59.39**
Yeast	**81.40**	59.48	69.89	**84.55**	56.44
Seismic	63.43	60.58	**66.79**	**70.91**	57.39
Heart	**59.06**	52.10	**56.25**	40.14	30.28

此外，我们还以所有数据集中的前 10%的数据点为实验对象，采用 *F*1 得分[8]作为衡量检测精度的指标来对所有算法进行性能比较。*F*1 得分是统计学中用来衡量二分类模型精确度的一种指标，可以看作是模型准确率和召回率的一种加权平均，它的最大值为 1，最小值为 0。不同算法在所有数据集中的 *F*1 得分的比较结果如表 3 所列，排名前两名的得分用黑体表示。可以看到，LODES 算法的检测精度远优于其他所有算法。具体来看，LODES 算法在 9 个数据集下排名前 2，HiCS 算法在 4 个数据集下排名前 2，OutDST 算法在 2 个数据集下排名前 2，FastABOD 算法在 5 个数据集下排名前 2，LOF 算法在 4 个数据集下排名前 2，这也充分验证了本文算法的有效性和优越性。

表 3　前 10%数据点的 *F*1 得分比较

Table 3　Comparison of *F*1 scores of the first 10% data points

数据集	LODES	HiCS	OutDST	FastABOD	LOF
Glass	**0.263**	0.132	0.132	**0.197**	0.132
Pendigits	**0.285**	**0.097**	0.007	0.017	0.081
Ecoli	**0.328**	0.188	0.047	**0.328**	**0.328**
Vowels	**0.328**	**0.328**	0.000	0.133	**0.400**
Cardio	**0.351**	0.185	0.033	**0.597**	0.206
Wine	**0.777**	0.079	**0.341**	0.253	0.000
Thyroid	0.055	**0.166**	0.026	0.047	**0.111**
Vertebral	**0.185**	0.111	0.000	0.000	**0.111**
Yeast	**0.358**	0.159	0.153	**0.441**	0.119
Seismic	0.131	**0.140**	0.107	**0.196**	0.103
Heart	**0.161**	0.000	**0.062**	0.000	0.000

4.4　参数敏感性分析

我们还分析了各种算法性能对于参数 k 的敏感度，其中 k 表示最近邻节点的数量。图 4 给出了各种算法在基本数据集条件下 *AUC* 与参数 k 的关系。从图 4 可以看出，除了 OutDST 算法，其余算法对参数 k 的敏感度均很低，这是 LODES 算法及其他基准算法的优势。OutDST 算法对于参数 k 非常敏感，如果参数 k 设置得不合理，则会导致邻域图会多出与局部密度结构不相符的边，这些边会大幅降低谱嵌入的质量。而 LODES 算法对参数 k 的敏感度较低，这是因为 LODES 算法可有效处理这些边，通过算法的相似度更新步骤可对邻域图进行调整，每次迭代时都可大幅降低与局部密度结构不相符的边的权重，有效缓解参数设置不当带来的负面影响，增强了算法的鲁棒性，提高了离群点检测的准确性。

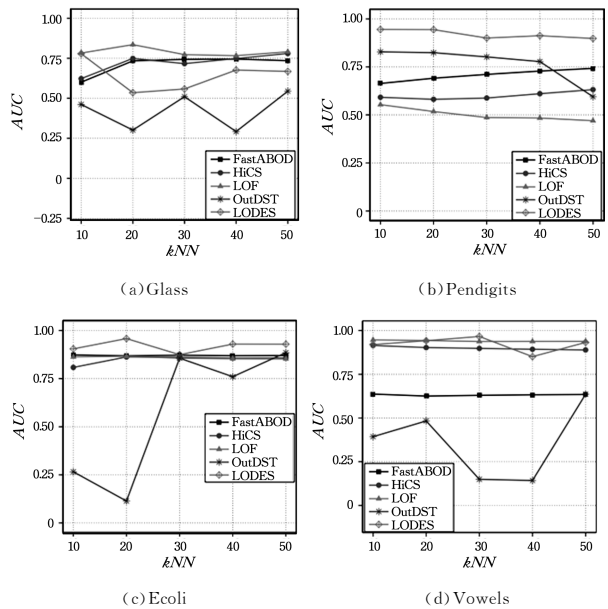


图 4　离群点检测算法对最近邻数量 k 的敏感度

Fig. 4　Sensitivity of outlier detection algorithm to nearest neighbor K

为了进一步体现 LODES 算法的优越性，以文中提及的

11 种数据集为实验对象，将 LODES 算法与目前最新的离群点检测算法[1-2] 在 *AUC* 方面进行了性能比较，实验结果如图 5 所示。从图 5 中可以看到，LODES 算法在 Glass，Pendigits，Ecoli，Vertebral，Seismic 和 Heart 6 个数据集中的 *AUC* 值都优于文献[1-2]中提出的两种离群点检测算法，而在另外 4 个数据集上的表现要与它们较接近，这都充分表明了 LODES 算法的有效性。仔细分析其原因可知，这主要是因为 LODES 算法可以处理离群点嵌入在局部非线性子空间这一情况，显著提升了离群点检测的精度，可以应用到大多数离群点检测场景中。

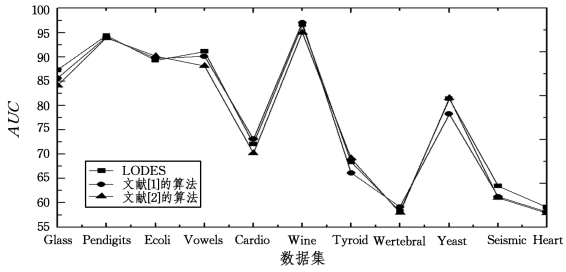


图 5 不同算法的 AUC 曲线的比较

Fig. 5 Comparison of AUC curves of different algorithms

结束语 离群点检测广泛应用于诈骗检测、医学诊断、故障检测及入侵检测等领域，人们对其进行了大量研究，并提出了多种离群点分析算法。这些算法可有效检测出线性流形周围的离群点。如果数据分布于具有随机形态的流形周围，则谱嵌入算法更为合适。但是在实践中，底层流形的密度不同，直接利用谱算法进行离群点检测的效果不佳。本文提出将谱嵌入算法与基于局部密度的检测算法相结合来提升离群点的检测性能。通过仿真实验验证了本文算法的性能优于当前其他典型算法。在未来的工作中，我们将对离群数据挖掘中的“维度灾难”问题进行研究，将基于角度的离群因子应用到高维离群数据挖掘中，拟提出一种基于随机投影算法的离群数据挖掘方法。

参 考 文 献

[1] RAHMANI M，ATIA G K. Randomized robust subspace recovery and outlier detection for high dimensional data matrices [J]. IEEE Transactions on Signal Processing，2017，65(6)：1580-1594.

[2] FAN F F，LI Z H，CHEN Q，et al. An Outlier-detection Based Approach for Automatic Entity Matching[J]. Chinese Journal of Computers，2017，40(10)：2197-2211. (in Chinese)
樊峰峰，李战怀，陈群，等. 一种基于离群点检测的自动实体匹配方法[J]. 计算机学报，2017，40(10)：2197-2211.

[3] TEMPL M，HRON K，FILZMOSER P. Exploratory tools for outlier detection in compositional data with structural zeros[J]. Journal of Applied Statistics，2017，44(4)：734-752.

[4] YANG J H，DENG T Q. A One-Cluster Kernel PCM Based SVDD Method for Outlier Detection [J]. Acta Electronica Sinica，2017，45(4)：813-819. (in Chinese)
杨金鸿，邓廷权. 一种基于单簇核 PCM 的 SVDD 离群点检测方法[J]. 电子学报，2017，45(4)：813-819.

[5] RO K，ZOU C，WANG Z，et al. Outlier detection for high-dimensional data[J]. Biometrika，2015，102(3)：589-599.

[6] BREUNIG M M，KRIEGEL H P，NG R T，et al. LOF：identifying density-based local outliers[J]. ACM Sigmod Record，2010，29(2)：93-104.

[7] KRIEGEL H P，ZIMEK A. Angle-based outlier detection in high-dimensional data [C] // Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas，Nevada，USA：ACM Press，2008：444-452.

[8] DANG X H，MICENKOVÁ B，ASSENT I，et al. Outlier detection with space transformation and spectral analysis[C] // Proceedings of the 13th SIAM International Conference on Data Mining. Austin，Texas，USA：IEEE Press，2013：225-233.

[9] NG A Y，JORDAN M I，WEISS Y. On spectral clustering：Analysis and an algorithm[C] // 26th Annual Conference on Neural Information Processing Systems 2012. Lake Tahoe，Nevada，United States：IEEE Press，2012：849-856.

[10] SHI J，MALIK J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2010，22(8)：888-905.

[11] CAMPOS G O，ZIMEK A，SANDER J，et al. On the evaluation of unsupervised outlier detection：measures，datasets，and an empirical study[J]. Data Mining and Knowledge Discovery，2016，30(4)：891-927.

[12] YANG Y，MA Z，YANG Y，et al. Multitask spectral clustering by exploring intertask correlation[J]. IEEE Transactions on Cybernetics，2015，45(5)：1083-1094.

[13] BI W，CAI M，LIU M，et al. A big data clustering algorithm for mitigating the risk of customer churn[J]. IEEE Transactions on Industrial Informatics，2016，12(3)：1270-1281.

[14] GU Y，LIU T，JIA X，et al. Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification[J]. IEEE Transactions on Geoscience and Remote Sensing，2016，54(6)：3235-3247.