文章编号:1671-6833(2023)06-0033-07

两阶段的近邻密度投票模拟离群点检测算法

郑忠龙[1], 曾 心[1], 刘华文[2]

(1. 浙江师范大学 数学与计算机科学学院,浙江 金华 321004;2. 绍兴文理学院 计算机系,浙江 绍兴 312000)

摘 要: 基于近邻的离群点检测算法对近邻选择较为敏感,邻域范围过小会增加模型复杂度,导致过拟合;邻域范围过大会使模型过于简单,忽略大量可用信息。为了降低邻域范围选择对离群点识别的影响,达到更高的精确度,基于近邻关系设计了一种投票决策的算法。该算法包括密度估计和模拟投票 2 个步骤:密度估计用于加速收敛数据点的密度得到稳态密度,从而根据稳态密度进行不同策略的模拟投票;模拟投票策略是基于社区发现算法改进得到的离群点检测核心算法,同时考虑数据点的重要性与其近邻的相似性进行投票。数据点的重要性与其稳态密度呈正相关,重要性越大的数据点将优先进行主动投票,把自身信息传递给邻域内与其相似度最大的数据点,并累计被投票数据点的投票排名。当每个数据点都进行主动投票后,算法停止迭代,得到各数据点的投票排名,将投票排名越靠后的数据点视为离群点。在 11 个真实数据集上的实验结果表明:基于近邻的投票模拟检测算法平均精确度为 79%,证明了所提算法的有效性。

关键词:近邻关系; 密度估计; 投票; 相似性; 离群点检测

中图分类号:TP301.6 **文献标志码**:A **doi**:10.13705/j.issn.1671-6833.2023.03.022

离群点检测是指从给定数据中找出或发现那些与其他数据存在明显差异的数据的技术[1]。由于能够带来诸多潜在价值,离群点检测在现实各领域有着广泛应用,如信用卡欺诈检测[2]、网络安全入侵检测[3]、安全系统故障检测[4]以及对敌人活动的军事监视[5]等。

近年来,学者们提出了许多离群点检测算法,其中以基于近邻的检测方法应用较广。该方法主要依据数据的近邻信息,判断其是否属于离群点。由于具有较强的可解释性和直观性,基于近邻的离群点检测是目前应用最为广泛的离群点检测方法之一。根据近邻信息的表示方式不同,基于近邻的离群点检测可细分为基于距离的方法和基于密度的方法。基于近邻距离的方法更关注全局离群点,而基于近邻密度的方法既能关注到全局离群点,也能有效发现局部离群点[6]。

为更好地表达现实生活中个体之间的复杂关系,并发现其中的结构团体,通常将社会关系描述成复杂网络或图的形式,其中每个节点代表一个实体,而节点之间的边则表示为实体之间的社会连接关系。社区发现是指社会网络从中发现那些紧密关系的节点的集合[7]。社区发现在现实生活中有着广泛应用,如在社交网络中挖掘具有共同兴趣或相似社会背景的群体进行内容推送[8];建立传染病网络动态模型,预测疫情发展趋势。

本文旨在利用社区发现中的消息传递机制,度量数据的密度信息,进而发掘数据中的离群点。传统的社区发现算法通过信息传递在复杂网络中交互信息,根据信息量优先识别最具影响力的节点。这与离群点检测原理一致,事实上,离群数据通常与大部分正常数据存在显著差异。因此,位于网络边缘、且对周边节点的影响力越小的节点,被认为是离群点的可能性越大。基于这种思想,本文提出了一种两阶段的近邻密度投票离群点检测算法。

收稿日期:2023-05-09;**修订日期**:2023-06-12

基金项目:国家自然科学基金资助项目(62272419, 61976195);浙江省自然科学基金资助项目(LZ22F020010, LZ23F020003)

作者简介:郑忠龙(1976—),男,河北沧州人,浙江师范大学教授,博士,博士生导师,主要从事加强学习与视觉研究,E-mail: zhonglong@zjnu.edu.cn。

通信作者:刘华文(1977—),男,江西乐安人,绍兴文理学院教授,博士,博士生导师,主要从事人工智能、大数据分析、机器学习研究,E-mail: liu@usx.edu.cn。

引用本文:郑忠龙,曾心,刘华文. 两阶段的近邻密度投票模拟离群点检测算法[J]. 郑州大学学报(工学版),2023,44(6):33-39. (ZHENG Z L, ZENG X, LIU H W. A two-stage outlier detection method based on neighbor denstity using voting[J]. Journal of Zhengzhou University (Engineering Science),2023,44(6):33-39.)

本文主要创新点如下:①利用社区发现中节点之间的消息传递机制产生的影响力,提出了一种新的基于近邻关系的投票离群点检测算法;②采用随机游走技术,计算节点及其近邻的密度,以反映节点的重要程度;③节点之间的信息交互只在近邻内部发生,从而降低了计算量,使得投票决策更具有可解释性,相比于全局范围内的投票决策节省了运算时间和空间。

1 相关工作

基于密度的离群点检测算法核心思想是根据数据对象给出的某种合理阈值,形成每个数据点的邻域范围,若数据点远离各邻域,则将该数据点视为离群点。近邻密度方法对数据整体分布没有要求,在局部离群点检测上表现优异,从而得到飞速发展,其中 Chen 等[9]提出的 K 近邻算法理论成熟且逻辑简单,是目前使用最为广泛的检测算法之一。但是基于近邻密度的方法对参数选取较为敏感,此类方法的离群点检测精确度容易受到参数的影响而产生较大波动。

影响力最大化(influence maximization, IM)[10]作为消息传递机制的主要研究内容之一,广泛应用于病毒营销、谣言控制、社交计算等实际应用场景中。投票作为一种有效识别节点影响力的算法,能够快速识别核心节点以进行信息传播。VoteRank 算法[11]首次使用二元组表示数据节点的投票能力与投票得分,按照一定的策略记录投票得分,发现节点影响力与投票得分呈正相关。WVoteRank 算法[12]考虑到权重对节点投票能力的影响,构建加权的一阶近邻矩阵来衡量不同相似度对节点投票能力的影响程度。在此基础上,VoteRank^{++}算法[13]同时引入一阶、二阶近邻迭代用于表示节点的投票能力。由于投票进行消息传递具有鲁棒性强、容错率高、可解释性强等优势,广泛应用于社区发现和集群检测中。然而,目前并没有直接将投票算法应用于离群点检测的相关研究。

本文受到节点影响力算法的启发,将投票模拟算法与传统离群点检测方法相结合,提出一种新颖的离群点检测算法。离群点通常位于网络边缘,且对周边节点的影响力较小,故本文选取影响力最小的数据为离群点。实验发现,本文将投票机制嵌入到离群点检测领域是一种新颖而有效的检测算法。

2 近邻投票的离群点检测

基于近邻投票的离群点检测算法包括密度估计和投票模拟 2 个步骤:密度估计通过随机游走进行密度迭代,得到估计密度和节点重要性;投票模拟通过重要节点传递信息,得到信息平衡时节点的信息量,将信息量最少的点视为异常点。

2.1 密度估计

设 $X=\{x_i\}_{i=1}^{N}, x_i \in \mathbf{R}^c$, X 是由 N 个 c 维数据对象构成的数据集。对数据集 X 中任意一个数据对象 x_i,计算出该数据对象到其他数据对象的欧氏距离,距离矩阵 $\boldsymbol{Dist}$ 为

$$\boldsymbol{Dist}(i,j)=\sqrt{\sum_{m=1}^{c}(x_i^m-x_j^m)^2}, i,j\in[1,N]。\quad(1)$$

距离矩阵是半正定的,为减小量纲影响,标准化处理得到 $\boldsymbol{Dist}^*$,使每个元素都在[0,1]内,$\boldsymbol{Dist}^*(i,j)$ 为节点 i 与 j 之间的距离。模型认为,两点间距离越大,相似性越低。基于此,令相似性矩阵 $\boldsymbol{SM}=\boldsymbol{I}-\boldsymbol{Dist}^*$。将相似性矩阵 $\boldsymbol{SM}$ 的行作为节点的初密度 $\boldsymbol{D}^0=[d_1^0,d_2^0,\cdots,d_N^0]^{\mathrm{T}}$,即 $d_i^0=\sum_{j=1}^{N}\boldsymbol{SM}_{ij}$。由 $\boldsymbol{SM}$ 可得到与节点 x_i 相似度最高的 K 个近邻集 $Q(x_i)$。

根据 $Q(x_i)$,设每个数据对象与其 K 近邻间存在一条无向边,由此构成了以近邻关系为基础的无权图 G。由图 G 得到邻接矩阵 $\boldsymbol{A}$:

$$\boldsymbol{A}(i,j)=\begin{cases}1, x_j\in Q(x_i);\\ 0, 其他。\end{cases}\quad(2)$$

基于以上假定,在图 G 上随机游走[14]得到各个节点的估计密度 $\boldsymbol{D}^t=[d_1^t,d_2^t,\cdots,d_N^t]^{\mathrm{T}}$,其迭代公式为

$$d_i^{t+1}=\alpha\sum p_{ij}d_j^t+(1-\alpha)d_i^0;\quad(3)$$

$$\boldsymbol{P}=\boldsymbol{M}^{-1}\boldsymbol{A}。\quad(4)$$

式中:d_j^t 为节点 j 在 t 时刻的密度,$j\in Q(x_i)$;p_{ij} 为马尔科夫转移概率矩阵 $\boldsymbol{P}$ 中的元素;$\boldsymbol{M}=\mathrm{diag}\left(\sum_j\boldsymbol{A}_{1j},\sum_j\boldsymbol{A}_{2j},\cdots,\sum_j\boldsymbol{A}_{Nj}\right)$ 为揭示每个节点近邻数量关系的对角矩阵。

基于贡献均等和无后效性假设,随机游走可达平稳状态。具体来说,贡献均等指节点 i 的 K 个近邻对 i 的贡献均等,无后效性指节点 i 在随机游走中具有无后效性,即 i 在 t+1 时刻状态只与 t 时刻状态有关。任意节点 i 做随机游走,如图 1 所示,箭头方向表示节点间的有向关系。近邻节点为当前节点可直达的节点,如图 1 中节点 2 的近邻节点为节点 1 和节点 3,节点 4 的近邻节点为节点 2 和节点 3。参数 α 表示由任意节点 i 根据箭头所指的有向关系游走到其近邻 j 的转移概率,认为每个节点 i 都有均等

的概率转移到其近邻节点,取值在[0,1]中;节点 i 直接回到自身会形成一个自环,如图 1 中 0 节点,其概率为 $1-\alpha$,又称回退概率。

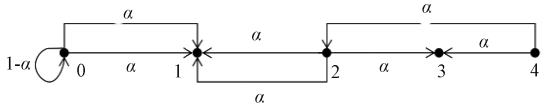


图 1　随机游走示意图

Figure 1　Illustration of random walk

将式(3)写作向量形式得

$$\boldsymbol{D}^{t+1} = \alpha \boldsymbol{P}\boldsymbol{D}^{t} + (1-\alpha)\boldsymbol{D}^{0}。\tag{5}$$

由 $\boldsymbol{P}$ 的定义可知,$|\boldsymbol{P}| \neq 0$,$|\alpha \boldsymbol{P}| \neq 0$,$\boldsymbol{P}$ 为标准化的矩阵,则 $\boldsymbol{I}-\alpha \boldsymbol{P}$ 非负定。由于 $\boldsymbol{P}$ 每一行中非 0 元的个数等于 K,当 $K>1$ 且 $\alpha \in [0,1]$ 时,$\alpha \boldsymbol{P}$ 至少有一个非主对角元不为 0、主对角元不等于 1,则 $|\boldsymbol{I}-\alpha \boldsymbol{P}| \neq 0$,$\boldsymbol{I}-\alpha \boldsymbol{P}$ 可逆。

该随机游走满足马尔科夫条件,故当概率 $\alpha<1$,$\boldsymbol{D}^{t+1}$ 将收敛到某值 Den,且有 $Den=(\boldsymbol{I}-\alpha \boldsymbol{P})^{-1}(1-\alpha)\cdot \boldsymbol{D}^{0}$,证明如下:

$$\boldsymbol{D}^{t+1} = \alpha \boldsymbol{P}\boldsymbol{D}^{t} + (1-\alpha)\boldsymbol{D}^{0};$$
$$\boldsymbol{D}^{t} = \alpha \boldsymbol{P}\boldsymbol{D}^{t-1} + (1-\alpha)\boldsymbol{D}^{0};$$
$$\vdots$$
$$\boldsymbol{D}^{2} = \alpha \boldsymbol{P}\boldsymbol{D}^{1} + (1-\alpha)\boldsymbol{D}^{0};$$
$$\boldsymbol{D}^{1} = \alpha \boldsymbol{P}\boldsymbol{D}^{0} + (1-\alpha)\boldsymbol{D}^{0}。$$

通过迭代可得到 $\boldsymbol{D}^{t+1}$ 关于 $\boldsymbol{D}^{t}$,$\boldsymbol{D}^{t-1}$,…,$\boldsymbol{D}^{0}$ 的表达式:

$$\begin{aligned}\boldsymbol{D}^{t+1} &= \alpha \boldsymbol{P}\boldsymbol{D}^{t} + (1-\alpha)\boldsymbol{D}^{0}\\ &= \alpha \boldsymbol{P}(\alpha \boldsymbol{P}\boldsymbol{D}^{t-1} + (1-\alpha)\boldsymbol{D}^{0}) + (1-\alpha)\boldsymbol{D}^{0}\\ &= \alpha^{2}\boldsymbol{P}^{2}\boldsymbol{D}^{t-1} + \alpha \boldsymbol{P}(1-\alpha)\boldsymbol{D}^{0} + (1-\alpha)\boldsymbol{D}^{0}\\ &\vdots\\ &= \alpha^{t+1}\boldsymbol{P}^{t+1}\boldsymbol{D}^{0} + (1+\alpha \boldsymbol{P} + \alpha^{2}\boldsymbol{P}^{2} + \cdots + \alpha^{t}\boldsymbol{P}^{t})(1-\alpha)\boldsymbol{D}^{0}。\end{aligned}$$

级数展开式:

$$1 + \alpha \boldsymbol{P} + \alpha^{2}\boldsymbol{P}^{2} + \cdots + \alpha^{t}\boldsymbol{P}^{t} + \cdots = (\boldsymbol{I}-\alpha \boldsymbol{P})^{-1}\boldsymbol{D}^{t+1} = \alpha^{t+1}\boldsymbol{P}^{t+1}\boldsymbol{D}^{0} + (\boldsymbol{I}-\alpha \boldsymbol{P})^{-1}(1-\alpha)\boldsymbol{D}^{0}。\tag{6}$$

当 $\alpha<1$ 且 $t\to\infty$ 时,$\alpha^{t+1}\to 0$,因此,$Den=(\boldsymbol{I}-\alpha \boldsymbol{P})^{-1}(1-\alpha)\boldsymbol{D}^{0}$ 得证。

利用式(6)可以得到估计密度值 Den,则近邻密度估计算法的简要描述如下。

算法 1　近邻密度估计。

输入:数据集 X,近邻参数 K,转移概率 α;

输出:每个数据点的估计密度值 Den。

Step 1　计算数据对象间的欧氏距离,标准化后得到距离矩阵 $\boldsymbol{Dist}^{*}$;

Step 2　获取数据的相似性 $\boldsymbol{SM}=\boldsymbol{I}-\boldsymbol{Dist}^{*}$;

Step 3　根据相似性矩阵 $\boldsymbol{SM}$,计算每个数据对象的初始密度 $\boldsymbol{D}^{0}$,及其 K 近邻集 $Q(x_i)$;

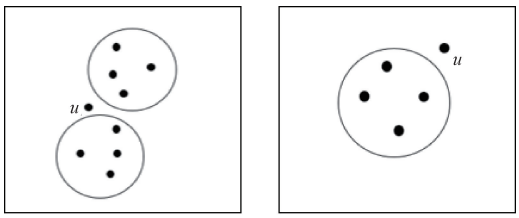
Step 4　利用 K 近邻 $Q(x_i)$,根据式(2),构造邻接矩阵 $\boldsymbol{A}$,并获取概率转移矩阵 $\boldsymbol{P}$;

Step 5　根据式(6),估计数据的密度值 Den。

算法 1 的时间主要花费在计算数据对象间的距离矩阵 $\boldsymbol{Dist}^{*}$ 和计算估计密度 Den,其最坏情况下的时间复杂度为 $O(NK+\log N)$,其中 N 为数据集中数据对象的数量;K 为每个数据对象的近邻个数。节点的估计密度和近邻关系是进行投票模拟的前提。

2.2　投票模拟

由于传统的异常检测算法对离群点的判别过于苛刻,易将小簇间节点视作异常,导致误判。如图 2 所示,图 2(a)中数据对象 u 处于两簇间,且簇间距较小,易聚合成更大的簇,图 2(b)中数据对象 u 位于单个簇外,远离簇中心。事实上是只有图 2(b)中 u 点为离群点,而传统的离群点检测算法会把这 2 种情况下的 u 均视为异常。投票算法通过 N 轮节点信息传递,宽松对待“可能异常”数据对象,将图 2(a)中 u 检测为正常节点,是一种容错率高的算法。



(a) 簇间离群点　　(b) 簇外离群点

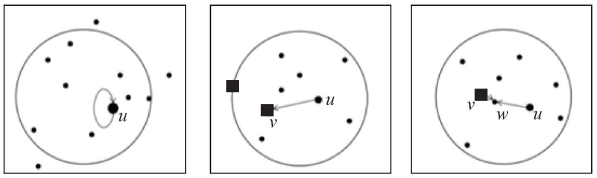
图 2　2 种不同离群点情况

Figure 2　Two different conditions

高密度节点周围往往环绕更多节点,成为中心节点的可能性大,而中心节点对近邻影响更大,故给予估计密度最大的节点优先投票权,每一轮投票均以当前密度最大的节点为候选投票节点,辐射更多的对象。每个数据点根据相似性和密度自发投票,直到每个数据点都完成投票,统计每一轮迭代的投票结果得到投票排名。在上述背景下,模拟投票算法根据以下 3 种原则进行迭代:①当 u 为未投票节点中估计密度最大的点,且 u 的 K 近邻集 $Q(u)$ 中不存在比 u 估计密度更大的节点,u 投票给自己;②当 u 的 K 近邻 $Q(u)$ 中存在比 u 估计密度更大的点,记与 u 相似性最大的点为 v;③若 v 未投票,则 u 投票给 v,若 v 投票给 w,则 u 也投票给 w。

图 3 为上述 3 种模拟投票原则。箭头方向表示投票方向,箭尾为投票者,箭头记录被投票者;点的大小为该点密度大小,正方形点是比 u 密度更大的候选节点;圆圈范围为 u 的 K 近邻集 $Q(u)$。

如图 3(a)所示,当 u 是所有未投票的节点中密



(a) 自我投票 (b) 近邻投票 (c) 信赖投票

图 3 模拟投票原则图解

Figure 3 Illustration of the principle of simulated voting

度最大的点时,u 的影响力最大,能够传递更多的信息给近邻节点,则 u 投票给自己。如图 3(b)所示,v 是 u 的邻域内比 u 密度更大且最近的节点,若 v 还未投票给其他节点,则 u 投票给 v。如图 3(c)所示,v 的密度在 u 的 K 近邻集 $Q(u)$ 中最大,且其影响力大于 u,若 v 投票给了其近邻点 w,则 u 投票给 w。算法 2 为投票模拟(voting simulation outlier detection, VSOD)算法。

算法 2 投票模拟算法。

输入:各数据点的估计密度值 *Den*,近邻点个数 K,异常个数 *nns*;

输出:离群点的索引。

Step 1 利用算法 1 计算密度值 *Den*,并对其降序排列,得到投票候选节点索引 *voteWho*;

Step 2 根据相似性 ***SM*** 计算每个候选点的 K 近邻集 $Q(x_i)$;

Step 3 以 *voteWho* 索引为序,利用 3 种不同模拟投票原则在候选点的近邻集 $Q(x_i)$ 中投票,记录投票得分 *votescore*;

Step 4 完成投票的节点跳出 *voteWho*,直到 *voteWho* 为空;

Step 5 取 *votescore* 降序排列对应的节点索引 *nodeOrder*;

Step 6 取 *nodeOrder* 后 *nns* 个索引。

算法 2 为本文的核心算法,这个阶段的时间复杂度为 $O(K^2+N\log N)$,其中 K 为每个数据点的近邻个数,N 为数据集中数据对象的数量。本文算法 $K\in[2,\sqrt{N}]$,其时间复杂度为 $O(N\log N)$。

通过算法 1 密度估计可以得到数据点之间的相似性与密度信息,为算法 2 投票消息传递奠定了基础。算法 2 中节点排序 *nodeOrder* 反映了节点重要性,当每个数据点都迭代投票后,得到 *nodeOrder* 越靠前的数据点,成为中心节点的可能性越大,而离群点往往远离中心,故选取 *nodeOrder* 靠后的节点为离群点。

3 实验

本节对所提出的投票离群点检测算法进行了实验分析,在 11 个不同类型和大小的公共数据集上与其他 9 种离群点检测算法进行比较。这些数据集从实际应用中收集得到,广泛用于评价异常检测方法的性能,实验数据来自于 UCI Machine Learning Repository 网站。

3.1 数据集

表 1 为各数据集的基本介绍。这些数据集最初应用在分类任务中,而离群点检测是无监督的,在实验中往往将更少类标签对象视为异常,其余对象视为正常[15]。本文在实验中也遵循该原则,将较少一类作为异常类。例如 WPBC 数据集,有 198 条维数为 33 的数据记录,其中有 47 条标签为 R 的异常数据记录,WPBC 数据集中异常数据量占总数据记录的 23.74%。

表 1 实验数据简要描述

Table 1 Brief description of experimental data

数据集	数据记录		异常数据记录		异常数据
	维数	数据量	标签	异常值	占比/%
WPBC	33	198	R	47	23.74
ALLAML	7 129	38	AML	11	28.95
DLBCL	7 130	59	1	27	45.76
HeartDisease	13	270	1	150	45.55
Ovarian	151 551	251	M	92	36.65
Parkinson	22	195	patients	48	24.62
Wine	13	129	1	10	7.75
Vertebral	6	310	NO	100	32.26
Flame	2	240	1	87	36.25
Speech	401	3 685	1	60	1.63
Lung	12 534	181	3	30	16.57

为验证本文算法的有效性,选取 9 种不同的离群点检测方法进行比较,实验在 AMD A9-9425 RADEON R5 处理器(3.10 GHz)、4 GB RAM 的计算机上进行。

3.2 评价指标

不同的检测算法参数不同,算法参数的选取可能会对最终结果产生影响,因此,合适的参数对算法至关重要。为保证严谨性,实验中的每种异常检测算法的参数都被设置为原文献中建议的值或工具箱推荐的值。为便于比较各方法的检测效果,实验采用常用的 3 个指标:*Precision*、max_*F*1 和 *AUC*。上述 3 个指标值均在[0,1]中,值越大算法检测性能越好。

3.3 实验结果及分析

根据随机游走模型[14]的参数设定,实验将算法 1 近邻密度估计中的参数 α 设置为 0.9,以加速迭代过程。在$[2,\sqrt{N}]$中取 VSOD 算法下精确度趋于平

稳时的 K 值,例 WPBC 数据集在 $K=6$ 以后精确度波动平稳,则在该数据集上 K 取值为 6,篇幅原因不做图表展示。根据表 1 中真实的异常数量,设置算法 2 投票模拟中的预测异常参数 *nns*,例 WPBC 数据集中异常数量值为 47,则算法 2 投票模拟中预测异常参数 *nns* 为 47[15]。

表 2 为异常检测算法的 *Precision* 比较。可以看出,本文投票模拟算法(VSOD)的平均精确度为 79%,在大部分数据集上表现较好。在 DLBCL 数据集上,其他检测方法的精确度最高为 62%,而所提的 VSOD 算法精确度为 81%。此外,VSOD 算法在少数数据上表现不足,但与其他传统检测算法差异不大,如在 Speech 数据集上,VSOD 算法的精确度为 90%,仍高于传统检测方法的均值 89%。而 Parkinson 数据集更适合采用线性和降维方法检测离群点。

表 3 为 9 种检测方法与 VSOD 算法在 11 个特点、规模不同的数据集上的 max_*F*1 得分。可以看出,max_*F*1 得分与表 2 中 *Precision* 具有类似的结论。当使用 Speech 数据集进行检测时,多数算法的 *Precision* 较高,但 max_*F*1 非常低,这可能是算法出现了欠拟合现象。而与其他算法相比,VSOD 算法包容性强,通过与最近邻的信息交互,其 *Precision* 和 max_*F*1 得分均较高。*Precision* 与 max_*F*1 均接近 1,验证了本文算法的有效性。在这 11 个数据集上,VSOD 算法的 max_*F*1 得分表现均较好,在 ALLAML、DLBCL、HeartDisease 和 Lung 数据集上,指标值均在 80%以上。

表 4 给出了不同算法的 *AUC*。在 Speech 数据集上,VSOD 算法与 ABOD 算法表现相当,又明显优于其他算法。这可能是由于 Speech 数据集的分布不可区分,导致密度相近,容易聚集成为更大的簇类,需要严格的检测手段加以区分。在其他数据集上,VSOD 算法的性能明显优于其他比较检测算法。综上,本文提出的投票检测算法是有效的。

表 2　VSOD 算法在不同数据集上与其他算法的 *Precision* 对比

Table 2　*Precision* of VSOD algorithm and other algorithms on different data sets

数据集	*Precision*/%									
	LOF	KNN	ABOD	SOD	COF	OCSVM	OutRank	ECOD	COPOD	VSOD
WPBC	69	68	69	70	69	70	65	54	74	**87**
ALLAML	66	66	61	66	71	66	59	70	68	**84**
DLBCL	59	59	62	55	59	59	52	61	55	**81**
HeartDisease	57	59	60	57	58	59	56	66	67	**78**
Ovarian	62	62	61	63	63	61	60	50	68	**70**
Parkinson	66	67	**69**	67	63	66	58	55	51	53
Wine	68	68	67	66	66	67	66	74	75	**83**
Vertebral	65	64	69	63	64	66	65	63	70	**71**
Flame	65	67	65	64	67	62		67	71	**82**
Speech	88	89	**91**	89		87				90
Lung	87	88	84	86	88	86	88	85	87	**91**

表 3　VSOD 算法在不同数据集上与其他算法的 max_*F*1 对比

Table 3　Max_*F*1 of VSOD algorithm and other algorithms on different data sets

数据集	max_*F*1/%									
	LOF	KNN	ABOD	SOD	COF	OCSVM	OutRank	ECOD	COPOD	VSOD
WPBC	48	49	42	40	43	45	39	41	44	67
ALLAML	59	63	53	60	67	52	50	60	61	88
DLBCL	62	62	62	62	62	62	66	62	66	89
HeartDisease	62	67	66	64	62	63	65	64	69	85
Ovarian	54	53	53	63	53	53	53	76	76	76
Parkinson	49	40	40	40	40	51	51	45	41	55
Wine	44	51	43	42	44	49	56	59	59	69
Vertebral	49	49	49	50	49	52	51	50	51	62
Flame	53	53	53	72	53	54	66	71	54	73
Speech	08	09	17	06		11	11			67
Lung	51	54	47	53	57	58	58	57	57	86

表4 VSOD算法在不同数据集上与其他算法的*AUC*对比

Table 4 *AUC* of VSOD algorithm and other algorithms on different data sets

数据集	*AUC*/%									
	LOF	KNN	ABOD	SOD	COF	OCSVM	OutRank	ECOD	COPOD	VSOD
WPBC	52	50	47	47	47	49	48	51	56	66
ALLAML	62	70	55	71	73	55	58	65	66	82
DLBCL	45	45	45	52	46	46	46	46	43	81
HeartDisease	56	63	64	59	55	58	52	64	75	88
Ovarian	46	47	47	49	43	47	47	41	42	78
Parkinson	51	34	36	28	39	57	66	64	52	77
Wine	41	57	46	42	40	56	90	69	68	91
Vertebral	42	26	29	33	42	35	45	48	34	62
Flame	53	56	54	53	56	51	47	66	48	89
Speech	53	51	73	52		51	57			71
Lung	68	69	64	65	69	67	67	81	79	85

4 结论

本文提出了一种有效的异常检测方法,该方法将密度估计方法与投票算法结合,在每一轮投票中宽容算法与严格算法结合,连贯进行,相互平衡。通过在11个真实数据集上进行对比实验得到如下结论。

(1)本文结合近邻密度与模拟投票的离群点检测算法可以在一定程度上提高检测的精确度。在11个真实数据集上的实验结果表明,基于近邻的投票模拟检测算法平均精确度为79%。

(2)通过邻域内投票迭代网络信息,提高了单纯采用近邻方法进行离群点检测的算法有效性,如LOF算法和KNN算法。本文算法是一种动态信息传递过程,减少了对数据点初密度的依赖性,通过投票不断修正中心节点和离群点。

本文提出的离群点检测算法在小规模数据集上检测效果较好,但需要根据数据集在一定区间内调整K的取值才能使其达到较优效果,故下阶段将主要研究新颖的消息传递方式,从而降低算法对K值的敏感度。

参考文献:

[1] XU X D, LIU H W, YAO M H. Recent progress of anomaly detection[J]. Complexity, 2019, 2019: 1-11.

[2] JIANG J F, HAN G J, LIU L, et al. Outlier detection approaches based on machine learning in the internet-of-things[J]. IEEE Wireless Communications, 2020, 27(3): 53-59.

[3] 汪祖民, 王冬昊, 梁霞, 等. 基于DBSCAN_GAN_XGBoost的网络入侵检测方法[J]. 郑州大学学报(工学版), 2022, 43(3): 44-51.
WANG Z M, WANG D H, LIANG X, et al. Network intrusion detection method based on DBSCAN_GAN_XGBoost[J]. Journal of Zhengzhou University (Engineering Science), 2022, 43(3): 44-51.

[4] 陈梦婷, 王兴刚, 刘文予. 基于密集深度插值的3D人体姿态估计方法[J]. 郑州大学学报(工学版), 2021, 42(3): 26-32.
CHEN M T, WANG X G, LIU W Y. Dense depth interpolation for 3D human pose estimation[J]. Journal of Zhengzhou University (Engineering Science), 2021, 42(3): 26-32.

[5] 吴小燕, 刘强, 朱成璋. 社交网络中协同舆论欺诈检测方法应用研究[J]. 郑州大学学报(工学版), 2022, 43(2): 7-14.
WU X Y, LIU Q, ZHU C Z. Research on application of collaborative public opinion fraud detection method in social network[J]. Journal of Zhengzhou University (Engineering Science), 2022, 43(2): 7-14.

[6] TANG B, HE H B. A local density-based approach for outlier detection[J]. Neurocomputing, 2017, 241: 171-180.

[7] YANG J W, RAHARDJA S, FRÄNTI P. Mean-shift outlier detection and filtering[J]. Pattern Recognition, 2021, 115: 107874.

[8] AFRASSA K W, COSGUN G, GURSOY U F, et al. On the community discovery methods for complex networks: a case study[C]//2020 15th Conference on Computer Science and Information Systems (FedCSIS). Piscataway: IEEE, 2020: 473-477.

[9] CHEN Y W, ZHOU L D, PEI S W, et al. KNN-BLOCK DBSCAN: fast clustering for large-scale data[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 51(6): 3939-3953.

[10] KEMPE D K, KLEINBERG J M, TARDOS É. Maximi-

zing the spread of influence through a social network[J]. Theory of Computing, 2015, 11: 105-147.
[11] ZHANG J X, CHEN D B, DONG Q, et al. Identifying a set of influential spreaders in complex networks[J]. Scientific Reports, 2016, 6(1): 1-10.
[12] SUN H L, CHEN D B, HE J L, et al. A voting approach to uncover multiple influential spreaders on weighted networks[J]. Physica A: Statistical Mechanics and Its Applications, 2019, 519: 303-312.
[13] LIU P F, LI L J, FANG S Y, et al. Identifying influential nodes in social networks: a voting approach[J]. Chaos, Solitons & Fractals, 2021, 152: 111309.
[14] DING J R, SHAH S, CONDON A. DensityCut: an efficient and versatile topological approach for automatic clustering of biological data[J]. Bioinformatics, 2016, 32(17): 2567-2576.
[15] DOMINGUES R, FILIPPONE M, MICHIARDI P, et al. A comparative evaluation of outlier detection algorithms: experiments and analyses[J]. Pattern Recognition, 2018, 74: 406-421.

A Two-stage Outlier Detection Method Based on Neighbor Density Using Voting

ZHENG Zhonglong [1], ZENG Xin [1], LIU Huawen[2]

(1. Institute for Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China; 2. Department of Computer Science, Shaoxing University, Shaoxing 312000, China)

Abstract: The outlier detection algorithm based on the nearest neighbor is sensitive to the selection of the nearest neighbor. Too small neighborhood range will increase the complexity of the model, resulting in over-fitting; Too much neighborhood will make the model too simple and ignore a lot of available information. In order to reduce the influence and achieve higher accuracy, a voting decision algorithm was designed based on the neighbor relationship. This algorithm consisted of two steps: density estimation and simulated voting. The density estimation was used to accelerate the density of convergent data to obtain the steady-state density, so that the simulated voting of different strategies could be carried out according to the steady-state density. Simulated voting strategy was the core algorithm of outlier detection based on the improvement of community discovery algorithm, and the importance of data points and the similarity of their neighbors to vote were taken into account. The importance of data points was positively correlated with their steady-state density. The data points with greater importance would have priority to vote actively, transmit their own information to the data with the greatest similarity in the neighborhood, and accumulate the voting ranking of the voted data. After each data has took the initiative to vote, the algorithm stopped iteration and obtained the voting ranking of each data point. The data with lower voting ranking was regarded as outlier. The experimental results on 11 real data sets showed that the average accuracy of the voting simulation detection algorithm based on the nearest neighbor was 79%, which could prove the effectiveness of the algorithm.

Keywords: neighbor relationship; density estimation; vote; similarity; outlier detection