软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(10):2625−2639 [doi: 10.13328/j.cnki.jos.005134]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于密度偏倚抽样的局部距离异常检测方法*

付培国[1,2], 胡晓惠[2]

[1](中国科学院大学,北京 100049)

[2](天基综合信息系统重点实验室(中国科学院 软件研究所),北京 100190)

通讯作者: 付培国, E-mail: peiguo12@iscas.ac.cn

摘 要: 异常检测是数据挖掘的重要研究领域,当前基于距离或者最近邻概念的异常数据检测方法,在进行海量高维数据异常检测时,存在运算时间过长的问题.许多改进的异常检测方法虽然提高了算法运算效率,然而检测效果欠佳.基于此,提出一种基于密度偏倚抽样的局部距离异常检测算法,首先利用基于密度偏倚的概率抽样方法对所需检测的数据集合进行概率抽样,之后对抽样数据利用基于局部距离的局部异常检测方法,对抽样集合进行局部异常系数计算,得到的异常系数既是抽样数据的局部异常系数,又是数据集的近似全局异常系数.然后对得到的每个数据点的局部异常系数进行排序,异常系数值越大的数据点越可能是异常点.实验结果表明,与已有的算法相比,该算法具有更高的检测精确度和更少的运算时间,并且该算法对各种维度和数据规模的数据都具有很好的检测效果,可扩展性强.

关键词: 异常检测;局部异常系数;局部距离;密度偏倚抽样;SLDOF 算法

中图法分类号: TP181

中文引用格式: 付培国,胡晓惠.基于密度偏倚抽样的局部距离异常检测方法.软件学报,2017,28(10):2625−2639. http://www.jos.org.cn/1000-9825/5134.htm

英文引用格式: Fu PG, Hu XH. Anomaly detection algorithm based on the local distance of density-based sampling data. Ruan Jian Xue Bao/Journal of Software, 2017,28(10):2625−2639 (in Chinese). http://www.jos.org.cn/1000-9825/5134.htm

Anomaly Detection Algorithm Based on the Local Distance of Density-Based Sampling Data

FU Pei-Guo[1,2], HU Xiao-Hui[2]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)

[2](Science and Technology on Integrated Information System Laboratory (Institute of Software, The Chinese Academy of Sciences), Beijing 100190, China)

Abstract: Anomaly detection is an important research area of data mining. Current outlier mining approaches based on the distance or the nearest neighbor can result in unmanageable long operation time when applied to massive high-dimensional data. Many improvements have been proposed to improve the algorithms, but the detection is ineffective. This paper presents a new anomaly detection algorithm based on the local distance of density-based sampling data. First, the density-based of probability sampling method is used to find a subset of the data in detection. Then, the method based on the local distance of local outlier detection is used to calculate the abnormal factor of each object in the subset. In using the density-based of sample data, the abnormal factor is obtained both as local outlier factor of the subset and as the approximate value of global outlier factor of the hole data. Having the abnormal factor of each object in the subset, data points with higher factor score indicate higher degree of outliers. Experimental results show that, compared with the existing

* 基金项目: 国家自然科学基金(U1435220); 国家高技术研究发展计划(863)(2012AA011206)

Foundation item: National Natural Science Foundation of China (U1435220); National High-Tech Research and Development Plan of China (863) (2012AA011206)

收稿时间: 2015-07-15; 修改时间: 2016-03-18, 2016-06-12, 2016-09-07; 采用时间: 2016-09-12; jos 在线出版时间: 2016-10-11

CNKI 网络优先出版: 2016-10-12 16:26:38, http://www.cnki.net/kcms/detail/11.2560.TP.20161012.1626.009.html

algorithms, this algorithm has higher detection accuracy and less computation time. The algorithm has higher efficiency and stronger scalability for various dimensions and size of data points.

Key words: anomaly detection; outlier factor of local set; local distance; density-based sampling; SLDOF algorithm

异常检测是数据挖掘领域一个重要的研究方向,用于发现数据集中的异常数据.Hawkins[1]给出了异常点的一个经典定义:一个异常点是一个数据点,它严重偏离其他数据点以至怀疑是由不同机制生成的.异常检测的目的是在大量的、复杂的数据集合中消除噪音数据或发现潜在的、有意义的知识.异常检测的典型应用包括电子商务犯罪、电信和信用卡欺诈、网络入侵检测、气象预报、生态系统失调、公共卫生、医疗以及天文学上稀有未知种类的天体发现等许多领域.

已有的异常检测算法有很多,包括有基于分布的、基于聚类的、基于分类的、基于深度的、基于距离的和基于密度的异常检测方法等.随着机器学习、人工智能、模式识别等领域的发展和进步,越来越多新颖、有效的异常检测技术和方法不断被提出.例如:交通数据流的时空移动异常检测[2]、基于人群移动性和兴趣点发现城市区域不同功能的方法[3]、基于分区的异常检测方法[4]、基于模糊粗糙集的异常检测方法[5]、利用自组织映射技术进行异常点检测[6]以及基于角度分布的异常检测方法[7]等.

随着数据收集设备性能的提高和数量的增加,收集的数据维度和数量均呈上升趋势,有些数据的维度高达数百维,数据点的数量高达 TB 级,这对已有的异常检测算法是一个挑战[8].传统的基于距离和基于密度的异常检测算法,需要进行高维数据邻近搜索.而在高维情况下数据十分稀疏,数据点之间的距离及区域密度不再具有直观的意义,并且数据维度越高,最近邻和最远邻数据就越难以区分.现有的挖掘算法大多具有 $O(n^2)$(n 为数据对象数目)的计算复杂度,计算开销太大.所以,传统的异常检测方法在高维数据上的应用有限,并且由于“维度灾难”问题,当前,大多数算法都或多或少地需要在全维空间对欧几里德距离进行考察,效果欠佳.因此,提高度量的有效性及计算的高效性是当前研究的热点.

Zhang 等人[9]提出基于局部距离的异常检测方法,但是该算法在进行复杂数据集合的异常检测时效果却不尽如人意;Wu 等人[10]提出了一种完整的基于抽样的异常检测算法;Sugiyama 等人[11]提出了对基于抽样异常检测算法的改进,可以大大缩短运行时间.基于以上情况,本文提出了一种基于密度偏倚抽样的局部距离异常检测算法.首先使用基于密度偏倚的概率抽样方法,对数据集合进行基于密度偏倚的概率抽样,得到抽样数据子集合;之后利用基于局部距离的局部异常检测方法,对抽样集合进行局部异常系数的计算;最后,对得到的每个数据点的局部异常系数进行排序,得分越高的数据点越可能是异常点.通过对合成数据和实际数据进行实验,其结果表明,与已有算法相比,本算法具有更高的检测精确度和更少的运算时间,对各种维度和数据规模的数据集合都具有很好的检测效果,算法可扩展性强.

本文第 1 节介绍相关工作.第 2 节介绍本文的一些预备知识.第 3 节具体介绍基于密度偏倚抽样的局部距离异常检测方法.第 4 节介绍具体的实验过程并给出结论.最后总结全文,并对未来值得关注的研究方向进行初步探讨.

1 相关工作

1.1 基于距离的异常检测方法

Zhang等人[9]最早提出了基于局部距离的异常检测方法.在基于局部距离的异常检测算法中,局部距离异常系数(local distance-based outlier factor,简称 LDOF)是基于局部距离的异常度量.考察点 p 与邻域点的平均距离,平均距离越大,则 p 是异常点的可能性越大.局部距离异常系数是两个平均距离的比值,反映的是局部统计特征的差异性.该度量比 k 最近邻算法(k-nearest neighbor,简称 KNN)有了较大改进,能在一定程度上识别局部异常点.LDOF 根据局部统计特征的差异性识别异常点,非常适合于局部异常点的检测.

基于密度的局部异常检测方法不是将异常点看作一种二元性质,而是转向量化地描述数据对象的异常程度,其可在数据分布不均匀的情况下准确地发现异常点.Breuning 等人[12]最先提出基于密度的异常定义,是在基

于距离定义[13]的基础上建立起来的.将给定范围内点之间的距离和点的个数这两个参数结合起来得到密度的概念.引入一个专门的度量单位:异常系数,用局部异常系数来表征一个对象的局部异常程度,局部异常程度是指对象与其局部邻域的偏离程度.通过数据空间的所有维度来计算对象的距离,进而计算对象的可达密度,最后通过局部的异常系数来判断异常点.该算法量化地描述数据对象的异常程度,通过赋予每一个数据对象一个表征其异常程度的量化指标来进行异常点的搜索.在局部异常检测(local outlier factor,简称 LOF)算法[12]中,根据给定的最少邻居数 k 和最近邻距离来确定邻域,通过计算对象的 k-距离、可达距离和可达密度,用数据对象邻域的平均可达密度与数据对象自身的可达密度之比表示局部异常因子.LOF 算法可以很好地解决局部异常点的挖掘问题,但该算法存在计算量大、计算结果受指定参数 k 影响等问题.

自从 LOF 算法被提出之后,出现了许多计算异常度的度量方法,比较典型的有基于局部信息熵的加权子空间异常检测算法[14]、基于连接的异常系数(connectivity-based outlier factor,简称 COF)[15]、多粒度偏差系数(multi-granularity deviation factor,简称 MDEF)[16]和局部空间异常测度(spatial local outlier measure,简称 SLOM)[17,18]等方法.

在 COF 算法中,根据给定的最少邻居数 k 和数据对象的连接性来确定邻域,计算与其邻域的平均连接距离,用平均连接距离比作为基于连接的异常系数.COF 算法虽可克服 LOF 算法中对于序列数据和低密度数据对象不能有效度量的缺陷,但仍存在计算复杂度高、计算结果受指定参数 k 影响等问题.而且,COF 增加了连接路径,因此时间复杂度比 LOF 算法还要高.

在 MDEF 算法中,有两个邻域概念,即 r-邻域和 αr-邻域,其中,$r>0$,$0<\alpha<1$.MDEF 算法的优点是可以根据应用要求设置多级邻域,并用邻域中包含的对象数目替代距离计算,降低了计算复杂度.但 r 和 α 很难确定,为了获得满意结果,需要反复修改参数.因此,MDEF 算法的检测结果和计算复杂度取决于用户的经验.

在 SLOM 算法中,将数据对象的属性分为空间属性和非空间属性,利用空间属性及空间邻接关系确定对象的邻域,以邻域距离 d 和波动系数 β 的乘积为空间局部异常程度,即 $SLOM=d\times\beta$.SLOM 算法与上述其他算法相比,在邻域的确定上不再依赖用户输入的参数,可从数据自身特点出发,利用空间数据的空间属性和空间关系确定空间邻域,解决了邻域的确定依赖于用户输入的参数等问题.利用空间索引技术,可极大地缩小数据搜索范围,减少对数据的访问次数,从而提高算法的效率.但是由于波动系数 β 仅由对称分布状况来决定,在空间邻居较少或波动幅度较小的情况下难以准确表现波动情况.因此会出现较高的漏检和误检现象,甚至得到的不是局部异常点,而是全局异常点.SLOM 算法存在计算复杂度高($O(kn^2)$)、检测结果的精度和重复计算的次数依赖于用户给定的参数等问题.

Kriegel 等人[7]提出运用基于角度分布的方法来计算高维数据的潜在异常点,但基于角度的异常检测算法存在运算量太大的问题($O(n^3)$).在此基础上,Pham 等人[19]提出基于随机投影的算法来近似估计各数据点的角度方差,该算法具有近似线性运行时间.但该算法在非圆数据中无法保证检测精度,并且当存在若干异常点聚集成一个簇时,易被误判为正常点,以致影响检测结果.

现有的基于异常度的局部异常检测算法主要区别在于邻域的确定方法和异常度的计算方法有所不同[20].但是,上述算法都存在以下问题:对最近邻、索引数据结构等方法均具有依赖性,检测结果的精度和重复计算的次数依赖于用户给定的参数,计算复杂度较高.

1.2 基于抽样的异常检测方法

Wu 等人[10]提出一种完整的基于抽样的异常检测算法,有效地解决了近似估计问题.算法在检测每一个数据对象时都要进行一次均匀抽样,并考察被测对象与每个样本间的距离,以 k 距离度量每个对象的异常度.由于不用在整个数据空间搜索 k 近邻,算法能获得线性的时间复杂度,但这没有包括抽样本身的时间复杂度.并且,用随机均匀抽样方法所抽出的样本,很难代表整个数据集的分布.由于所有数据点以相同的概率被抽出,使得异常点本身亦有同等的机会出现在样本集中.Sugiyama 等人[11]提出了对基于抽样异常检测算法的改进,把对每一个数据点的迭代随机抽样改为只进行一次随机抽样,实验效果很好,可以大大缩短运行时间.

基于以上情况,本文提出一种基于密度偏倚抽样的局部距离异常检测算法.首先使用基于密度偏倚的概率

抽样方法,对数据点集合进行基于密度偏倚的概率抽样.在进行密度偏倚的概率抽样时,每个点被抽中的概率与其所在区域的数据密度密切相关,密度大(小)的区域以较大(较小)的概率被抽中.由于在概率密度函数中设置了概率控制参数,用户可以自己决定不同数据簇中数据点被选中的概率.之后利用基于局部距离的异常检测方法,对抽样集合进行局部异常系数的计算.通过之前的抽样步骤,极大地减小了局部异常系数的计算量,有效缩短了算法运算时间.最后对得到的每个数据点的局部异常系数进行排序,异常系数值越大的数据点越可能是异常点.实验结果表明,与已有算法相比,本算法具有更高的检测精确度和更少的运算时间,并且对各种维度和数据规模的数据点都具有很好的检测效果,算法可扩展性强.

2 预备知识

2.1 基于概率的数据抽样

Wu 等人[10]提出一种完整的基于迭代抽样的异常检测方法,对数据中每一个点的 k_{th}-NN 距离给出近似值 q_{kthNN}.Sugiyama 等人[11]提出了对基于抽样异常检测算法的改进,把对每一个数据点的迭代随机抽样改为只进行一次随机抽样,重新定义数据点 o 的异常系数: $q_{S_p}(o)=\min_{o'\in S'} d(o,o')$, 比文献[10]中的抽样方法具有更好的异常检测效果.

设数据集 $S=\{o_1,o_2,...,o_i,\ldots,o_N\}$ 中的数据点分属于 k 个簇 $\{C_1,C_2,...,C_k\}$. $S=\bigcup_{i=1}^{k} C_i$, $C_i\cap C_j=\varnothing$,$i,j=1,\ldots,k$.$S'=\{o_1,o_2,\ldots,o_j,\ldots,o_M\}$ 为 S 的一个样本,是数据点 o 的随机迭代抽样,样本容量为 $M<N$.定义数据点 o 的异常系数: $q_{kthS_p}(o)=d^k(o,S')$.

定义 1(保持密度抽样). 一个抽样是保持密度的,如果对于任意的 C_j,$j=1,\ldots,k$,有

$$\sum_{o\in C_j}\Pr(o\in S'\mid o\in C_j)=\alpha_j\mid C_j\mid \tag{1}$$

其中,α_j 为某个常数.

定义 2(均匀抽样). 一个抽样是均匀的,如果对于任意的 $o\in C_i,o'\in C_j,i,j\in\{1,\ldots,k\}$,都有

$$\Pr(o\in S'\mid o\in C_i)=\Pr(o'\in S'\mid o'\in C_j) \tag{2}$$

显然,对于均匀抽样,有

$$\Pr(o\in S'\mid o\in S)=M/N \tag{3}$$

这个概率就是式子中的常数α.

均匀抽样存在一定的弊端,Guha 等人[21]指出:要保证数据集 S 中的一个大小为 u 的簇$\phi(0\leqslant\phi\leqslant 1)$部分的数据以不小于 $1-\delta(0\leqslant\delta\leqslant 1)$的概率被采样,样本的大小 M 必须满足:

$$M\geqslant\phi\times N+\frac{N}{u}\log\left(\frac{1}{\delta}\right)+\frac{N}{u}\sqrt{\left(\log\left(\frac{1}{\delta}\right)\right)^2+2\times\phi\times u\times\log\left(\frac{1}{\delta}\right)} \tag{4}$$

基于抽样的异常检测算法基于以下两个条件.

(1) 异常点的个数远远小于正常簇的数据点个数.设 O 为异常点的集合,则有

$$\mid O\mid\ll\mid\min\{\mid C_1\mid,...,\mid C_k\mid\}\mid \tag{5}$$

(2) 抽样是基于随机抽样进行的,包括两层含义:一是簇内均匀抽样,即对任意的 $o,o'\in C_i,i\in\{1,\ldots,k\}$,有

$$\Pr(o\in S')=\Pr(o'\in S') \tag{6}$$

二是抽样概率是簇密度的一个函数,即簇间抽样概率与每个簇内数据点个数相关.

在实际应用中,一方面,欲使符合某类特征的簇能有较大的概率被采样,不得不以增加样本容量为代价,但过大的样本将失去抽样的意义.另一方面,当需要符合另一类特征的簇以较小的概率被采样时,又不得不将样本容量降到很小,从而失去代表性.因此,如何选择满足条件(1)和条件(2)的抽样函数是关键问题.第 3.1 节给出了基于密度偏倚的数据抽样方法以解决这一问题.

2.2 局部距离的异常检测方法

Zhang 等人[9]提出了基于局部距离的异常检测方法.在基于局部距离的异常检测算法中,LDOF 是基于距离的异常度量.考察点 p 与邻域点的平均距离,平均距离越大,则 p 是异常点的可能性越大.LDOF 是两个平均距离的比值,反映的是局部统计特征的差异性.该度量比 KNN 方法有了较大改进,能在一定程度上识别局部异常点.LOF 是基于局部密度的异常度量,而 LDOF 是基于局部距离的异常度量.LDOF 根据局部统计特征的差异性确定异常点,非常适合于局部异常检测.

首先给定一个数据集合 S,点 $o\in S$,其 k-距离记为 $dist_k(o)$,是点 o 与另一个数据点 $p\in S$ 之间的距离 $dist(o,p)$,使得:

(1) 至少有 k 个数据点 $o'\in S\backslash\{o\}$,使得 $dist(o,o')\leqslant dist(o,p)$.

(2) 至少有 $k-1$ 个数据点 $o''\in S\backslash\{o\}$,使得 $dist(o,o'')<dist(o,p)$.

即 $dist_k(o)$是 o 与其第 k 个最近邻之间的距离.o 的 k-距离邻域包含其到 o 的距离不大于 $dist_k(o)$的所有对象,记为

$$N_k(o)=\{o'\mid o'\in S,dist(o,o')\leqslant dist_k(o)\} \tag{7}$$

在集合 $N_k(o)$中分别计算数据点 o 的平均距离 $dist_k(o)$以及集合 $N_k(o)$中所有点的平均距离 $\overline{D_k(o)}$,之后再计算点 o 的局部距离异常系数 $LDOF_k(o)$,具体计算方法在第 3.3 节中给出.

3 基于密度偏倚的局部距离异常检测方法

基于密度偏倚抽样的局部距离异常检测算法的具体过程如下.首先使用基于密度偏倚的概率抽样方法对所需检测的数据集合进行概率抽样;之后对抽样数据利用基于局部距离的局部异常检测方法,在进行局部异常系数计算时,使用抽样数据进行局部异常选择,对抽样集合进行局部异常系数计算,得到的异常系数既是抽样数据的局部异常系数,又是数据集的近似全局异常系数;最后对得到的每个数据点的局部异常系数进行排序,异常系数值越大的数据点越可能是异常点.仿真实验结果表明,本算法具有更高的检测精确度和更少的运算时间,并且该算法对各种维度和数据规模的数据点都具有很好的检测效果,算法效率高,可扩展性强.

3.1 基于密度偏倚的数据抽样

在实际应用中,欲使符合某类特征的簇能有较大的概率被采样,不得不以增加样本容量为代价,但过多的样本将失去抽样的意义.另一方面,当需要符合另一类特征的数据簇以较小的概率被采样时,又不得不将样本容量降到很小,从而失去代表性.所以提出了一个与密度相关的概率函数,使得抽样满足上述条件.本文采用如下的抽样概率函数.

$$\Pr(o\in S'\mid o\in C_j)=\frac{\alpha}{|C_j|^{\lambda}},\ j\in\{1,...,k\} \tag{8}$$

其中,α为大于 0 的常数,λ 为调节参数,$-1\leqslant\lambda\leqslant 1$.在这里使用参数 λ 来调节不同簇被抽中的概率,在第 3.2 节具体讨论参数λ的函数设定.这样的抽样函数满足如下条件.

(1) 簇内均匀抽样,即对任意的 $o,o'\in C_i,i\in\{1,2,\ldots,k\}$,有

$$\Pr(o\in C_i)=\Pr(o'\in C_i) \tag{9}$$

(2) 抽样概率的簇密度是一个函数;

(3) 总的样本数学期望为 M.

定理 1. 为使得总样本的数学期望为 M,按照式(8)定义的抽样概率函数中的参数α需满足:

$$\alpha=\frac{M}{\sum_{i=1}^{k}|C_i|^{1-\lambda}} \tag{10}$$

定理 2. 以式(8)和式(10)定义的概率抽样,为保证一个大小为 u 的数据簇 C 至少有$\phi\times u(0\leqslant\phi\leqslant 1)$个数据点以不低于 $1-\delta(0\leqslant\delta\leqslant 1)$的概率被采样,所需的最小样本数比均匀抽样的样本数少,只需要数据簇 C 的大小 u

满足:

$$u \leqslant \mathrm{e}^{\frac{\log M+\log N-\log\sum_{j=1}^{k}u_j}{1-\lambda}} \tag{11}$$

其中,$u_j=|C_j|,j=1,2,\ldots,k$.

定理 1 和定理 2 的证明在附录中给出.

3.2 基于哈希的密度近似估计

采用网格划分方法将数据集合划分为网格,每一个单元格视为一个子簇,以单元格内的对象个数为近似密度.为减少簇划分的复杂性,采用哈希表的方法,将网格映射成哈希表中的单元,同时累计数据点的个数.这样只需对数据集合进行一次扫描,即可估计出所有对象的近似密度.

哈希方法的一个潜在的问题是碰撞,碰撞的结果会影响抽样的概率.假如由于碰撞的原因,使得簇 C_i 和簇 C_j 合并为一个簇,则两个簇原来的抽样概率 $P_{C_i}=\frac{\alpha}{|C_i|^{\lambda}},P_{C_j}=\frac{\alpha}{|C_j|^{\lambda}}$ 实际被如下的抽样概率所代替.

$$P_{C_i\cup C_j}=\frac{\alpha}{(|C_i|+|C_j|)^{\lambda}} \tag{12}$$

注意到两个簇发生冲突后,使得抽样概率要大于原来的值.但从上式得知,抽样概率却变小了,这部分抵消了冲突带来的影响,使得冲突后从合并的簇中抽出的样本总数并没有发生很大的变化.因此,本文所采用的抽样概率函数对冲突不敏感.

因此,为了进一步减少簇划分的复杂性,对局部具有相似密度的相近单元格进行上卷合并,组合其为更大的数据簇,并计算上卷之后的数据簇的近似密度.通过对单元格的合并操作,上卷为接近实际的聚类分析的数据簇,从而克服了在数据复杂度未知时进行数据聚合的簇个数选择问题.最终得到数据集合的簇聚类,之后对聚类的簇按照抽样概率函数进行抽样.

通过上卷操作,可以对之前定义的变量λ引入函数定义.将λ定义为关于聚类簇中的网格个数 p 的函数 $\lambda=f(p)$.f 是减函数,即 p 越大,λ值越小,聚类簇中的数据以更大的概率被采样.这就使得空间范围大的数据簇以更大的概率被采样,而空间范围小的数据簇以更小的概率被采样.

经过对数据的抽样之后,得到数据集合 S 的一个子集合 S',子集合 S'即为被检测点 o 的新异常系数计算空间.之后,在子集合 S'上利用局部异常算法进行数据点异常系数计算,下一节将讨论基于局部距离的异常系数计算的具体步骤.

3.3 基于局部距离的异常系数计算并识别异常点

Zhang 等人[9]最先提出基于局部距离的异常定义,是在基于距离定义[13]的基础上建立起来的.将给定点和给定范围内的点之间的平均距离与给定范围内的所有点的平均距离这两个参数结合起来,得到局部距离异常系数的概念.引入一个专门的度量单位:LDOF,用局部距离异常系数来表征一个对象的局部异常程度.

通过第 2.2 节的介绍,对于给定数据点 o,$dist_k(o)$是点 o 与 $N_k(o)$中的所有数据点之间的距离的平均.即设点 $o'\in N_k(o)$且 $dist(o,o'\geqslant 0)$,则点 o 的 $dist_k(o)$定义为

$$dist_k(o)=\frac{1}{k}\sum_{o'\in N_k(o)}dist(o,o') \tag{13}$$

数据点 o 在 $N_k(o)$的 k-最邻近内部距离定义为邻域 $N_k(o)$中所有数据点的平均距离:

$$\overline{D_k(o)}=\frac{1}{k(k-1)}\sum_{o,o'\in N_k(o)}dist(o,o') \tag{14}$$

则点 o 的局部距离异常系数定义为

$$LDOF_k(o)=\frac{dist_k(o)}{\overline{D_k(o)}} \tag{15}$$

如果把点 o 的 k-距离邻域作为数据集合的子集合,$LDOF_k(o)$表示点 o 与邻域 $N_k(o)$中的其他数据点的偏离程度.可以很直观地看出,$LDOF_k(o)$是邻域 $N_k(o)$中点 o 与其 k-距离邻居之间的距离比值,表示检测点与局部数据点的偏离程度.$LDOF_k(o)<<1$ 表示点 o 属于邻域 $N_k(o)$的内部.相反地,$LDOF_k(o)>>1$ 表示点 o 属于邻域 $N_k(o)$的外部.$LDOF_k(o)$值越高,表示点 o 越远离其 k-距离邻域 $N_k(o)$,对于一个深藏在一致簇内部的对象,局部距离异常系数越小.这一性质确保了无论簇是稠密的还是稀疏的,簇内部的对象不会被错误地标记为异常点.

通过对集合中所有数据点的异常系数计算之后,对得到的所有数据点的局部异常系数进行排序,局部异常系数值越大的数据点越有可能是异常点.

3.4 SLDOF 算法流程

通过上述对包括基于密度偏倚概率的数据抽样、基于哈希的近似密度估计以及基于局部距离的异常系数计算这 3 部分的描述,SLDOF 算法的具体操作步骤如下.

算法 1. Hash(把数据对象映射为 hash 表中的单元).

输入:数据对象 o_i,

维度 D,

哈希表长度 H,

网格划分刻度 l;

输出:哈希表的索引 ID.

1. 初始化 h=0
2. **for** j=0:D **do**
3. $h=h*65599+(o_{i,j}/l)$;
4. **end for**
5. **return** (h%H);

算法 2. 基于密度偏倚抽样的局部距离异常检测算法.

输入:数据集 S,

考察的近邻个数 k,

样本数 M,

哈希表大小 H,

偏倚度调节参数 λ;

输出:Top n 个异常点 O.

1. 初始化 $HASH\_Table[H]$,$S'=\varnothing$
2. **for** each $o_i\in S$ **do**
3. $HASH\_Table[Hash(o_i)]=HASH\_Table[Hash(o_i)]+1$;
4. **end for**
5. $\alpha = M\Big/\sum_{i=1}^{H}(HASH\_Table[i])^{1-\lambda}$;
6. **for** each $o_i\in S$ **do**
7. $r=random()$
8. **if** $r<\alpha/HASH\_Table[Hash(o_i)]^{\lambda}$ **then**
9. $S'\leftarrow o_i$
10. **end if**
11. **end for**
12. **for** each $o_i\in S$ **do**
13. $dist_k(o_i)$ //在 S'中计算 o_i 的平均距离

14. $\overline{D_k(o_i)}$ //在 S'中计算所有数据点的平均距离

15. $LDOF_k(o_i)=\frac{dist_k(o_i)}{\overline{D_k(o_i)}}$

16. $O\leftarrow(o_i,LDOF_k(o_i))$

17. **end for**

18. **return** O;

通过使用基于密度偏倚抽样的局部距离异常检测算法,对数据集合进行基于概率密度偏倚的抽样.在抽样集合的基础上使用基于局部距离的异常系数计算,之后对得到的每个数据点的局部异常系数进行排序,局部异常系数值越大的数据点越可能是异常点.通过下节的具体实验结果表明,与已有的算法相比,本算法具有更高的检测精确度和更少的运算时间,并且具有更好的数据适应性.

4 实验仿真

为了验证本文 SLDOF 算法的性能,在本节中通过具体的实验结果和其他几个经典的异常检测算法如 FastABOD 算法[7]、LDOF 算法[9]、LOF 算法[12]、FastVOA 算法[21]以及利用本文的基于密度偏倚抽样的 SLOF 算法,通过合成数据集和实际数据集的运算结果来检测算法的有效性和可用性.其中,SLOF 算法是对 LOF 算法增加基于密度偏倚的概率抽样步骤的改进算法.所有算法用 Java 实现,运行环境为 JDK8,利用合成数据集和真实实验数据集,在 12 核心的 Intel Xeon 3.5GHz CPU,32G RAM 的 64 位的 Windows 10 平台上进行实验.

4.1 性能度量

评价异常检测方法的好坏,一个常用的度量标准是算法的准确率,即通过计算该算法检测出来的异常点中真实异常点所占据的比例加以判别.同时,对于大规模数据的异常检测,除了算法的准确率之外,还要考虑算法的运算时间和泛化能力,以避免可能存在的过拟合问题.已有的分类器性能测量包括准确率、敏感度、特效性、精度、F_1 和 F_β等.注意到,尽管准确率是一个特定的度量,但是准确率也经常用于讨论分类器的预测能力.在实际的数据检测中,将考虑各类元组大致均匀分布的情况,也考虑类不平衡的情况.在实际的数据集中经常会出现类不平衡现象,即负样本比正样本多很多(或者相反),而且测试数据中的正负样本的分布也可能随着时间发生变化.特别是在异常数据的检测中主要是考虑类不平衡的情况,其中感兴趣的异常类是稀少的,即数据集的分布负类显著地占据多数,而正类占据少数.因此,在本文中选取受试者工作特征曲线(receiver operating characteristic curve,简称 ROC 曲线)来作为异常检测算法性能度量的标准,这是因为 ROC 曲线具有很好的特性:当测试集中的正负样本的分布发生变化时,ROC 曲线能够保持不变.

4.2 数据描述和参数设定

对异常检测算法进行性能检测.在选取测试数据集合时,由于在实际数据集合中很难确切地知道哪些数据点是由不同的机制生成的,这也是评估异常检测算法性能时存在的一个问题.另外,虽然存在很多的异常检测方法,但是确定数据点是否为异常点更多地是由主观角度来决定的.存在的另一个问题是,在算法检测出来的异常列表中,很难去评估算法检测出来的异常值是否是数据集合中真实的异常值.鉴于此,在本文中,使用人工合成的数据对异常检测算法进行性能评估.使用异常点的初始定义,分别用不同的机制生成数据集合的正常点和异常点.为了生成具有良好定义但又不明显异常的异常数据集合,采用如下步骤进行.

本文使用合成混合数据生成方法[21],生成基于高斯分布和泊松分布的数据簇.这些数据簇中正常点的均值和方差均随机生成,利用全维空间均匀分布作为异常点.同时生成与数据簇相独立的基于均匀分布的 100 个异常点,并利用不同规模和维度的合成数据集对各种算法进行性能评估.在进行数据规模和数据维度的性能比较时,分别生成 100,500 和 1 000 维度的、数据规模分别为 5 000 和 50 000 等不同规模的数据集合;在进行算法的效率比较时,分别生成数据规模为 5 000,10 000,20 000,30 000,40 000 和 50 000 这 6 个 500 维度的混合数据,分别对算法运行 10 次求平均值,然后从有效性和运行时间两个方面对几种算法进行比较.另外,由于这几种算法

都需要进行参数 k 的设置,Zhang 等人[9]指出,在大数据集合中 k 值的合理取值范围为[20,50],因此在本实验中,设置 k 的范围为[20,25].另外,由于 SLDOF 算法和 SLOF 算法还需要设置抽样概率,在这里设置为 $p=(\lambda\times k)/N$,其中,k 为之前设置的参数,N 为所有数据点的个数,λ为大于等于 1 的参数,用作抽样数据集合和参数 k 的比率.λ越大,其异常系数值越接近全局异常系数,但相应的算法运算时间会有所增加.因此,需要选取合适的λ值来得到最优的结果,在本文中,设置λ=1.5.

选用了两个实际数据集合 Isolet 和 Multiple Features.其中,Isolet 是 UCI 机器学习库为分类和机器学习任务设计的[22],Isolet 包括字母表 26 个字母的发音数据.Multiple Features 也是 UCI 机器学习库为分类和机器学习任务设计的[22],Multiple Features 包括手写体的阿拉伯数字(0~9).对每个数据集,选择具有共同行为的某种类别的所有数据点作为正常点,从另一类选择 10 个数据点作为异常点.例如,选择 Isolet 数据集 C,D,E 类别都有 e 声的点作为正常点,选择 Y 类别 10 个点作为异常点.类似地,在 Multiple Features 数据集中,选择类 6 和 9 作为正常点,选取 10 个 0 作为异常点.需要指出的是,很有可能部分异常点位于内点覆盖区域,因此,无法准确分离所有异常点.但是,希望本文算法能够将异常点划入异常值范围.

本文算法将对异常点检测方法的有效性和效率等以算法的准确率和运算时间等作为评测指标,并在同等实验条件下与 FastABOD 算法、LDOF 算法、LOF 算法、SLOF 算法和 FastVOA 算法分别对合成实验数据和真实数据进行了性能对比.

4.3 实验结果分析

(1) 有效性

为了比较不同算法在不同数据规模和数据维度的检测性能,使用人工合成的包括 5 000 和 50 000 个数据规模的 5 个聚簇的 100,500 和 1 000 维等不同维度的 6 个数据集合进行测试,分别对 SLDOF 算法、SLOF 算法、LDOF 算法、LOF 算法和 FastABOD 算法的运算情况进行分析.

图 1 显示了不同算法在不同数据规模和不同数据维度的 ROC 曲线,包括数据规模为 5 000 和 50 000 个点的数据集合.对于不同规模的数据集合,分别选取 100,500 和 1 000 维度的数据集合进行性能比较(包括 5 000 个数据点的 ROC 曲线和 50 000 个数据点的 ROC 曲线).

从维度为 100 的相对低维数据集合来看(如图 1(a)和图 1(d)所示),FastABOD 算法、SLDOF 算法和 SLOF 算法这 3 种算法都具有很好的检测效果.从图 1(a)可以看出,FastABOD 算法、SLDOF 算法和 SLOF 算法检测的 top n 个点全部为异常点,具有比其他算法更好的 AUC 值(area under the ROC curve,简称 AUC).LDOF 算法和 LOF 算法的结果却差强人意.

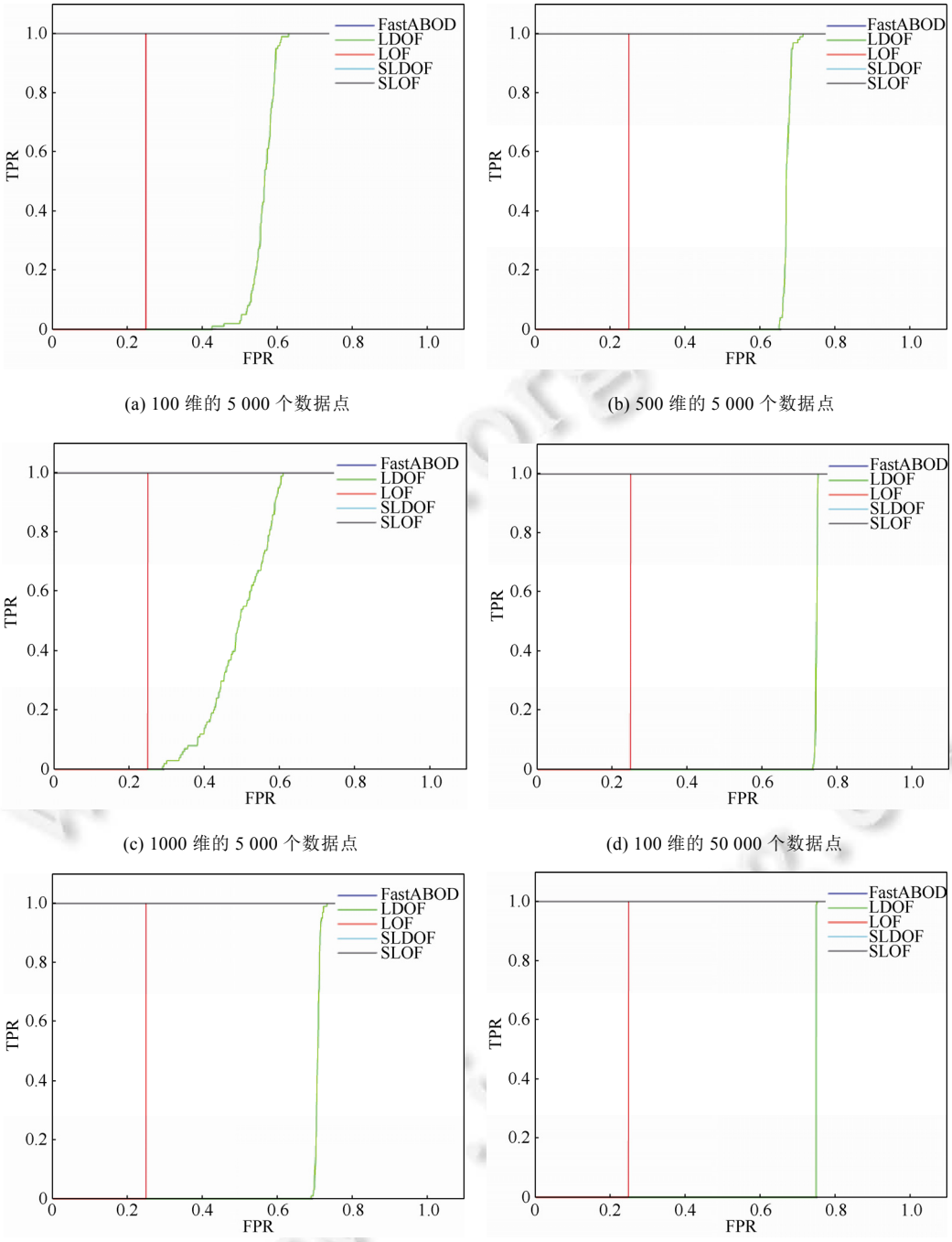
为了比较 SLDOF 算法在所有数据维度的数据集合中都有很好的表现,又设计了 5 种算法在 500 维和 1 000 维的数据集合上进行测试(如图 1(b)、图 1(e)和图 1(c)、图 1(f)所示).从中等规模数据维度的集合上的检测结果可以看出,FastABOD 算法、SLDOF 算法和 SLOF 算法仍然具有很好的检测效果,其 AUC 值仍为 1.0.LDOF 算法和 LOF 算法的检测结果仍然较差.最后,又测试了 5 种算法在高维数据集合上(如图 1(c)、图 1(f)所示)的检测效果,FastABOD 算法、SLDOF 算法和 SLOF 算法也具有同样优秀的检测效果,其 AUC 值均为 1.0.相反地,LDOF 算法和 LOF 算法无论是在低维数据、中维数据还是在高维数据上,其检测结果都不如前 3 种算法好.在本实验中,LDOF 算法和 LOF 算法的检测效果不如前 3 种算法,可能是因为选取的是人工合成的数据,异常点和正常点的分布混杂在一起,比较难以区分.

通过这 6 个包括不同数据规模和不同维度的数据集合的测试,SLDOF 算法的检测效果都非常出色,超过了 LDOF 算法和 LOF 算法的检测效果,得到的 top n 个数据点全部为异常点,所有数据集合的检测结果其 AUC 值都为 1.0.但是,在前面的实验中发现,FastABOD 算法和 SLOF 算法具有与 SLDOF 算法同样的检测效果.接下来通过另外的实验,可以得到 SLDOF 算法具有比 FastABOD 算法和 SLOF 算法更好的性能.

(2) 效率

在本节中,比较所有算法在选取 top n 个异常点的运算时间上的不同.因此,选取了在 5 000、10 000、20 000、30 000、40 000 和 50 000 个数据规模的 6 个不同的数据集合上进行算法运算时间的比较.在本实验中,选取数

据维度为 500,按照第 4.2 节的设定选取算法所需参数,分别对 SLDOF 算法、SLOF 算法、LDOF 算法、LOF 算法和 FastABOD 算法的运算情况进行分析.



(a) 100 维的 5 000 个数据点 (b) 500 维的 5 000 个数据点

(c) 1000 维的 5 000 个数据点 (d) 100 维的 50 000 个数据点

(e) 500 维的 50 000 个数据点 (f) 1 000 维的 50 000 个数据点

Fig.1 The ROC curve of synthetic data sets of different dimensions as 100, 500 and 1 000

图 1 100、500 和 1 000 维度的人工合成数据集合的 ROC 曲线图

实验结果如图2所示.由于FastABOD算法的运算时间过长,为了使比较结果更加易于观察,在图2中只显示了FastABOD算法的部分运算时间.从图 2 中可以看出,随着数据规模的扩大,所有算法的运算时间都随之增加,其中,增加最快的是FastABOD算法.LDOF算法和LOF算法具有相近的运算时间,并且随着数据规模的扩大,运算时间的增长也比较相近.但在不同数据规模下,LDOF算法的运算时间都稍微低于 LOF 算法的运算时间.类似地,SLDOF 算法的运算时间也都稍稍小于SLOF算法的运算时间.另外,通过图2可以明显看出,由于采用了基于密度偏倚的概率抽样,在相同数据规模下,SLDOF 算法和 SLOF 算法的运算时间要远远小于其他算法的运算时间.

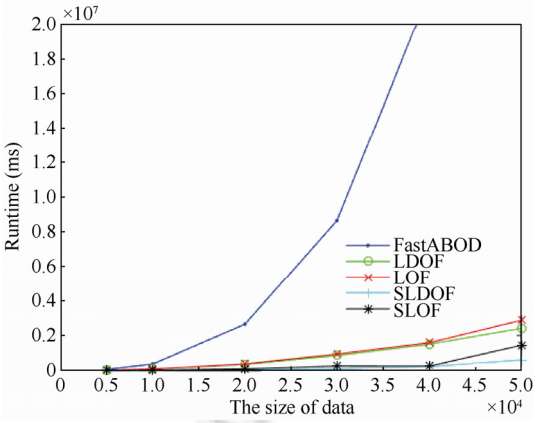


Fig.2 The runtime of algorithms under synthetic dataset

图2 在合成数据集各算法的运算时间

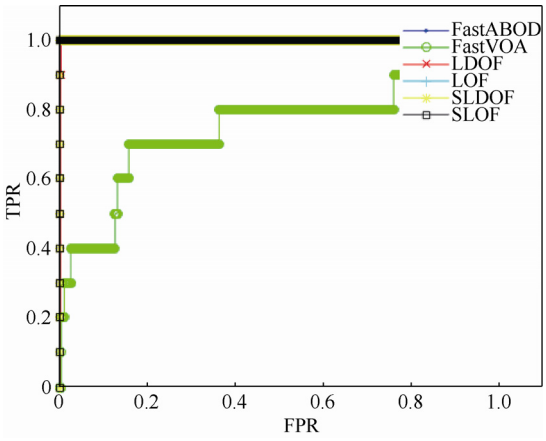
(3) 算法在真实数据集合的性能测试

在之前的实验中,所有的算法都是在人工合成的数据集合上进行性能检测,在本节中选择实际数据集合Isolet和Multiple Features来测试算法.在实际数据集合中,某些异常点不可避免地会和正常点的距离很近.因此,在实验中不是一定要找到所有的异常点,但是要使得异常点的排序足够高.

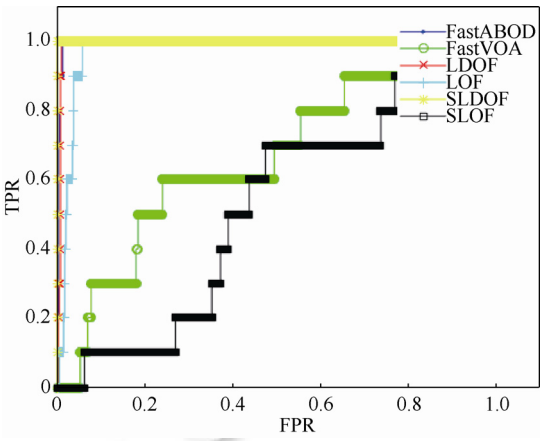
在第1个实验中,选取Isolet数据集.Isolet包括字母表26个字母的发音数据,选择Isolet数据集C,D,E类别都有e声的点作为正常点,选择Y类别10个点作为异常点.从图3(a)可以看出,FastABOD算法、LOF算法、SLDOF算法和SLOF算法表现得都非常好,都能够得到AUC值为1.0的好结果.其次为LDOF算法,随着FPR值的增加,TPR也迅速地增长为1.各种算法在Isolet集合上的运算时间如图3(c)中的第1部分直方图所示,顺序依次为FastABOD算法、FastVOA算法、LDOF算法、LOF算法、SLDOF算法和SLOF算法的运算时间表示.由于FastVOA算法的运算时间值太大,在图3(c)中只显示了部分值.通过图3(c)中的第1部分直方图可以明显地看出,由于采用了基于密度偏倚的概率抽样,SLDOF算法和SLOF算法的运算时间要远远小于其他算法的运算时间,其中SLDOF算法的运算时间最少.

在第2个实验中,选取Multiple Features数据集.Multiple Features数据集合包括手写体的阿拉伯数字(0~9),在Multiple Features数据集中,选择类6和9作为正常点,选取10个0作为异常点.类似Isolet数据集的结果,从图3(b)可以看出,只有SLDOF算法取得了最完美的结果,能够得到AUC值为1.0的最好结果.其次为FastABOD算法和LDOF算法,这两种算法具有近似的结果,没有明显的区别,都稍优于LOF算法.在Isolet数据集中表现很好的SLOF算法在本次实验中结果最差,明显不如SLDOF算法的泛化性能好.各种算法在Multiple Features数据集合上的运算时间如图3(c)中的第2部分直方图所示,可以明显地看出,由于采用了基于密度偏倚的概率抽样,SLDOF算法和SLOF算法的运算时间要远远小于其他算法的运算时间.虽然SLOF算法的运算时间也很少,但是检测结果却最差.表现最好的是SLDOF算法,不仅具有最少的运算时间,还具有最好的检测效果.

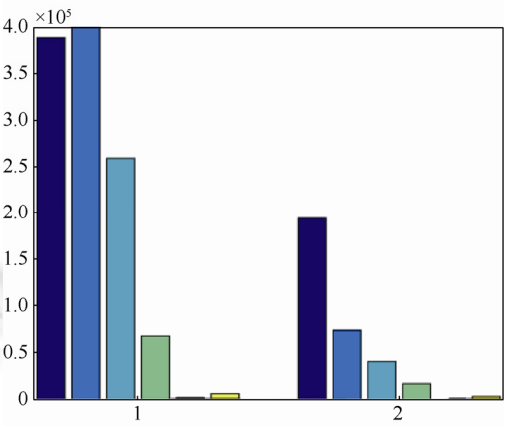
综上,对各种异常检测方法在对数据维度和数据规模的可扩展性、算法的运算时间以及算法的泛化性等几个方面的测试,通过对合成实验数据的性能对比结果表明,在同等实验条件下,SLDOF算法与SLOF算法、LDOF算法、LOF算法、FastABOD算法相比,在任何情况下,SLDOF算法都具有更好的运算结果和更少的运算时间.由于合成实验数据集是人为生成的数据集,其生成的数据点具有一定的规律性,所以不完全代表算法性能.之后,我们又通过对真实数据集合Isolet和Multiple Features数据集的异常检测的实验结果表明,SLDOF算法仍然具有更好的运算结果和更少的运算时间.但在参数选取上,由于在进行数据抽样时,设置了抽样数据集为 $p=(\lambda\times k)/N$,其中,k 为最近邻参数,N 为所有数据点的个数,λ为大于等于 1 的参数,用作抽样数据集合和参数 k 的比率,SLDOF算法相比其他算法多了抽样概率调节参数λ.



(a) Isolet 数据集 ROC 曲线



(b) Multiple Features 数据集 ROC 曲线



(c) 算法在实际数据集合的运算时间

Fig.3

图 3

5 结 论

当前已有的异常数据检测算法都存在运算时间过长的问题,因此本文提出了一种基于密度偏倚抽样的局部距离异常检测算法.首先利用基于密度偏倚的概率抽样对所需检测的数据集合进行抽样,之后对抽样数据利用基于局部距离的局部异常检测方法,在进行局部异常系数计算时,对抽样集合进行局部异常系数计算,得到的异常系数既是抽样数据的局部异常系数,又是数据集的近似全局异常系数.最后对得到的每个数据点的局部异常系数进行排序,得分越高的数据点越可能是异常点.通过对数据维度和数据规模的可扩展性、算法的运算时间以及算法的泛化性等几个方面的测试,在合成数据集合和真实数据集合的实验结果表明,本算法具有更高的检测精确度和更少的运算时间,对各种维度和数据规模的数据点都具有很好的检测效果,并且效率高,可扩展性强.

但是由于需要进行数据抽样,相比其他算法需要更多的参数设置,这增加了算法的不确定性.因此,下一步的工作是研究参数与数据分布、运行结果之间的关系,尽量找到一种通用的参数设置机制,弥补现存算法的不足.未来将考虑进行不确定性数据和复杂数据,如基于时空线索的数据的异常检测工作.

致 谢 向给予支持和提出宝贵建议的评审专家深表感谢.

References:

[1] Hawkins DM. Identification of Outliers. London: Chapman and Hall, 1980.

[2] Liu W, Zheng Y, Chawla S, Yuan J, Xing X. Discovering spatio-temporal causal interactions in traffic data streams. In: Proc. of the 17th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM, 2011. 1010–1018. [doi: 10.1145/2020408.2020571]

[3] Yuan J, Zheng Y, Xie X. Discovering regions of different functions in a city using human mobility and POIs. In: Proc. of the 18th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM, 2012. 186–194. [doi: 10.1145/2339530.2339561]

[4] Stone-Gross B, Cova M, Vigna G. Your botnetis my Botnet: Analysis of a botnet takeover. In: Proc. of the ACM Conf. on Computer and Communication Security (CCS). 2009. 635–647. [doi: 10.1145/1653662.1653738]

[5] Yaday S, Reddy A, Ranjan S. Detecting algorithmically generated malicious domain names. In: Proc. of the 10th Annual ACM Conf. on Internet Measurement. New York, 2010. 48-61. [doi: 10.1145/1879141.1879148]

[6] Stalmans E, Irwin B. A framework for DNS based detection and mitigation of malware infections on a network. In: Proc. of the Information Security South Africa (ISSA). 2011. 1–8. [doi: 10.1109/ISSA.2011.6027531]

[7] Kriegel HP, Schubert M, Zimek A. Angle-Based outlier detection in high dimensional data. In: Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2008. [doi: 10.1145/1401890.1401946]

[8] Han JW, Micheline K. Data Mining: Concepts and Techniques. 2nd ed., Elsevier, 2006.

[9] Zhang K, Hutter M, Jin H. A new local distance-based outlier detection approach for scattered real-world data. In: Advances in Knowledge Discovery and Data Mining. Berlin, Heidelberg: Springer-Verlag, 2009. 813–822. [doi: 10.1007/978-3-642-01307-2_84]

[10] Wu M, Jermaine C. Outlier detection by sampling with accuracy guarantees. In: Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. ACM, 2006. 767–772. [doi: 10.1145/1150402.1150501]

[11] Sugiyama M, Borgwardt K. Rapid distance-based outlier detection via sampling. In: Advances in Neural Information Processing Systems. 2013. 467–475.

[12] Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: Identifying density-based local outliers. ACM SIGMOD Record, 2000,29(2): 93–104. [doi: 10.1145/335191.335388]

[13] Knorr EM, Ng RT. Algorithms for mining distance-based outliers in large datasets. In: Proc. of the 24th Int'l Conf. on Very Large Data Bases. New York: Morgan Kaufmann Publishers Inc., 1998. 392–403.

[14] Ni W, Chen G, Lu J, Wu Y, Sun Z. Local entropy based weighted subspace outlier mining algorithm. Journal of Computer Research and Development, 2008,45(7):1189–1192 (in Chinese with English abstract).

[15] Tang J, Chen Z, Fu AWC, Cheung DW. Enhancing effectiveness of outlier detections for low density patterns. In: Advances in Knowledge Discovery and Data Mining. Berlin, Heidelberg: Springer-Verlag, 2002. 535–548. [doi: 10.1007/3-540-47887-6_53]

[16] Papadimitriou S, Kitagawa H, Gibbons PB, Faloutsos C. LOCI: Fast outlier detection using the local correlation integral. In: Proc. of the 19th Int'l Conf. on Data Engineering. IEEE, 2003. 315–326. [doi: 10.1109/ICDE.2003.1260802]

[17] Chawla S, Sun P. SLOM: A new measure for local spatial outliers. Knowledge and Information Systems, 2006,9(4):412–429. [doi: 10.1007/s10115-005-0200-2]

[18] Xue AR, Ju SG, He WH, Chen WH. Study on algorithms for local outlier detection. Chinese Journal of Computers, 2007,30(8): 1454–1463 (in Chinese with English abstract).

[19] Pham N, Pagh R. A near-linear time approximation algorithm for angle-based outliner detection in high dimensional data. In: Proc. of the 18th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2012. 877–885. [doi: 10.1145/2339530.2339669]

[20] Ng RT, Han JW. Efficient and effective clustering methods for spatial data mining. In: Proc. of the 20th Int'l Conf. on Very Large Data Bases. Morgan Kaufmann Publishers Inc., 1994. 144–155.

[21] Guha S, Rastogi R. Cure: An efficient clustering algorithm for large databases. Information Systems, 1998,27(1):35–58.

[22] Bache K, Lichman M. UCI Machine Learning Repository. 2013. https://www.researchgate.net/publication/272825857_UCI_Machine_Learning_Repository

[23] Buot M. Probability and computing: Randomized algorithms and probabilistic analysis. Journal of the American Statistical Association, 2005,75(473):395−396.

附中文参考文献:

[14] 倪巍伟,陈耿,陆介平,吴英杰,孙志挥.基于局部信息熵的加权子空间离群点检测算法.计算机研究与发展,2008,45(7):1189−1192.

[18] 薛安荣,鞠时光,何伟华,陈伟鹤.局部离群点挖掘算法研究.计算机学报,2007,30(8):1454−1463.

付培国(1986－),男,河南鹤壁人,博士,主要研究领域为 Web 服务计算,数据挖掘.

胡晓惠(1960－),男,博士,研究员,博士生导师,主要研究领域为计算机应用技术,信息系统集成,数据挖掘.

附 录

定理 1. 为了使总样本的数学期望为 M,按照式(8)定义的抽样概率函数中的参数α需满足:

$$\alpha=\frac{M}{\sum_{i=1}^{k}|C_i|^{1-\lambda}} \tag{10}$$

证明:设 O_i,i=1,…,N,为随机变量,且

$$O_i=\begin{cases}1, & \text{若}o_i\text{被抽中}\\ 0, & o_i\text{未被抽中}\end{cases}'$$

则有:

$$\begin{aligned}M&=\sum_{i}^{N}O_i\\&=\sum_{o\in O}\Pr(o\in O'\mid o\in O)\\&=\sum_{i=1}^{k}\sum_{o\in C_i}\Pr(o\in O'\mid o\in C_i)\\&=\sum_{i=1}^{k}|C_i|\frac{\alpha}{|C_i|^{\lambda}}\\&=\frac{\alpha}{\sum_{i=1}^{k}|C_i|^{1-\alpha}}.\end{aligned}$$

即 $\alpha=\dfrac{M}{\sum_{i=1}^{k}|C_i|^{1-\alpha}}$. □

定理 2. 以式(8)和式(10)定义的概率抽样,为保证一个大小为 u 的数据簇 C 至少有$\phi\times u(0\leqslant\phi\leqslant 1)$个数据点以不低于 $1-\delta(0\leqslant\delta\leqslant 1)$的概率被采样,所需的最小样本数比均匀抽样的样本数少,只需数据簇 C 的大小 u 满足:

$$u\leqslant \mathrm{e}^{\frac{\log M+\log N-\log\sum_{j=1}^{k}u_j}{1-\lambda}} \tag{11}$$

其中,$u_j=|C_j|$,j=1,2,…,k.

证明:设 O_i,m=1,…,M,为独立同分布的随机变量,当样本 O'中的第 i 个点属于簇 C 时,取值为 1,否则为 0,则样本中属于簇 C 的样本数 O 可由下式计算:

$$O=\sum_{j=1}^{M}O_j \tag{16}$$

C 中至少有 $\phi\times u$ 个数据点以不低于 1–δ的概率被抽中,即:

$$\Pr(O<\phi\times u)<\delta \tag{17}$$

记 $\mathrm{E}[O]=\mathrm{E}\left[\sum_{j=1}^{M}O_j\right]=\sum_{j=1}^{M}\mathrm{E}[O_j]$.根据 Chernoff Bounds[23],对任意的 $0<\varepsilon\leqslant 1$,有:

$$\Pr(O<(1-\varepsilon)\mu)<\mathrm{e}^{\frac{-\mu\varepsilon^2}{2}} \tag{18}$$

式(17)可以写成:

$$\Pr\left(X<\left(1-\left(1-\frac{\phi\times u}{\mu}\right)\right)\times\mu\right)<\delta \tag{19}$$

比较式(18)和式(19),只要下式成立,则式(18)成立:

$$\mathrm{e}^{\frac{-\mu\left(1-\frac{\phi\times u}{\mu}\right)^2}{2}}\leqslant\delta \tag{20}$$

对式(20)两边取对数,并进行代数计算,有:

$$\mu^2-2\mu\left(\phi\times u+\log\left(\frac{1}{\delta}\right)\right)+(\phi\times u)^2\geqslant 0 \tag{21}$$

求解不等式方程(21),可得:

$$\mu\geqslant\phi\times u+\log\left(\frac{1}{\delta}\right)+\sqrt{\log^2\left(\frac{1}{\delta}\right)+2\times\phi\times u\times\log\left(\frac{1}{\delta}\right)} \tag{22}$$

记抽样概率为 p,则 O 的数学期望为

$$\mu=\sum_{i=1}^{M}O_i=M\times p \tag{23}$$

由式(22)和式(23),有:

$$M\geqslant\phi\times u\times\frac{1}{p}+\frac{1}{p}\times\log\left(\frac{1}{\delta}\right)+\frac{1}{p}\times\sqrt{\log^2\left(\frac{1}{\delta}\right)+2\times\phi\times u\times\log\left(\frac{1}{\delta}\right)} \tag{24}$$

式(24)是样本 M 的下界.接下来证明:只要式(11)满足,则这个下界(不等式右侧)比式(4)的下界要小,即所需的样本数更少.

对式(11)两边取对数,有:

$$\log u\leqslant\frac{\log M+\log N-\log\sum_{j=1}^{k}u_j}{1+\lambda} \tag{25}$$

进一步对式(25)作代数运算,可得:

$$\frac{M}{u^{\lambda}\sum_{j=1}^{k}u_j^{1-\lambda}}\geqslant\frac{u}{N} \tag{26}$$

注意到式(26)的左侧即 p,故有:

$$p\geqslant\frac{u}{N} \tag{27}$$

将式(27)代入式(24),再与式(4)比较,定理得证. □