第49卷第7期 中国科学技术大学学报 Vol. 49, No. 7
2019年7月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Jul. 2019

文章编号:0253-2778(2019)07-0555-09

AnomalyDetect:一种基于欧式距离的在线异常检测算法

霍文君[1],王 伟[1,2],李 文[1]

(1.同济大学电子与信息工程学院,上海 200092;2.华东师范大学数据科学与工程学院,上海 200062)

摘要:异常检测是数据挖掘中的一项关键技术,在计算机和互联网领域有广泛的应用,包括网络安全、图像识别、智能运维等,特别是智能运维,近几年取得了长足的发展.已有的异常检测算法会有低准确度、离线、无法自动更新等问题.为此对智能运维背景下的真实异常检测问题进行研究,构建高准确度、在线、通用异常检测算法,并据此在已有时间序列异常检测算法的基础上,提出了一种新的基于欧式距离的在线异常检测算法.通过实际的运维时序数据实验,发现该算法可以实时快速准确地检测流式时间序列数据中的异常数据,验证了该算法的有效性.

关键词:异常检测;时间序列;在线算法;欧式距离;智能运维

中图分类号:TP391 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2019.07.005

引用格式:霍文君,王伟,李文. AnomalyDetect:一种基于欧式距离的在线异常检测算法[J]. 中国科学技术大学学报,2019,49(7):555-563,571.

HUO Wenjun, WANG Wei, LI Wen. AnomayDetect: An online distance-based anomaly detection algorithm[J]. Journal of University of Science and Technology of China, 2019,49(7):555-563,571.

AnomayDetect: An online distance-based anomaly detection algorithm

HUO Wenjun[1], WANG Wei[1,2], LI Wen[1]

(1. *Department of Computer Science and Engineering, Tongji University, Shanghai* 200092, *China*;
2. *School of Data Science and Engineering, East China Normal University, Shanghai* 200062, *China*)

Abstract: Anomaly detection is a key challenge in data mining which has a wide range of applications in the field of the Internet, including network security, image recognition and intelligent operation. In particular, intelligent operation has made great progress in recent years. Existing anomaly detection algorithms have many problems, such as low accuracy and inability to update automatically. The problem of anomaly detection in the context of intelligent operation and a practical need for high-accuracy, online and universal anomaly detection algorithms is studied. Based on the existing algorithms, an online distance-based anomaly detection algorithm is identified. Through the experiments on Yahoo Web-scope S5 dataset it is shown that the algorithm can detect anomalies successfully. A comparative study of several anomaly detectors verifies the effectiveness of the proposed algorithm.

Key words: anomaly detection; time series; online algorithm; euclidean distance; intelligent operation

收稿日期:2018-09-25;**修回日期**:2018-12-04

基金项目:国家自然科学基金(61672384),同济大学中央高校基本科研业务费专项资金(0800219373)资助.

作者简介:霍文君,女,1994年生,硕士生.研究方向:分布式计算. E-mail:huowj@tongji.edu.cn

通讯作者:王伟,博士/副教授. E-mail:wwang@tongji.edu.cn

0 引言

时间序列是指按照时间顺序排列的一组随机变量,将某一随机事件按照时间顺序记录下来便得到一组时间序列的观察值.时间序列数据出现在生活中的方方面面,例如,每只股票的当日结算价格,银行某位客户每月的存款余额,ATM 机的每日交易金额,医学上病人每分钟的心跳数,对这些时间序列数据进行观察、分析、并发现其中的规律将会有巨大的应用前景.现阶段,有关时间序列的研究一般有预测、模式挖掘、聚类、分类、异常检测等,在金融领域,时间序列预测技术可以应用在股票价格预测中,预测和分类技术可以帮助银行判定客户的信用等级;在医学领域,模式挖掘和异常检测技术可以帮助医生快速发现病人监控数据中的异常情形并做出及时处理.

在智能运维中,通过对软硬件的各项指标进行实时检测并记录会取得一系列的时间序列数据,而一些软硬件故障、恶意攻击等异常行为会直接反映在取得的时间序列数据中,形成异常数据.运维人员的职责之一就是对这些指标数据实时监控,当异常出现时及时进行修复处理,智能运维的落地关键是利用异常检测算法自动监测实时运维数据中的异常情形.本文正是在该背景下对已有的时间序列异常检测算法进行分析,结合运维数据的特点和运维需求提出一种新的在线异常检测算法.

时间序列中的异常检测是指在时间序列数据中找到与其周围点表现异常的一个点或一段序列.真实情况下,异常点或异常序列一般代表着异常情况:非法交易、不正常心跳、网络恶意攻击等.针对时间序列异常检测的大部分算法的思想是对正常情形下的时间序列数据进行建模,当前观测点或序列与正常数据差异过多时将被视为异常点.

智能运维领域[1]中,由于服务器硬件软件产生的数据是流式数据,且由于监控的指标数目及种类较多,因此数据量较大,数据波动频繁且不规律,当异常行为出现时,数据表现出的异常形态也有所不同.因此,结合运维数据的特点以及运维场景对异常检测算法的要求,选择适合于真实运维场景下恰当的异常检测算法对智能运维技术的落地影响重大.本文提出的基于距离的在线异常检测算法运用滑动窗口机制,通过计算时间序列中当前时间子序列的最小非欧几里得距离,从而衡量当前时间子序列的异常性,当异常性超过给定阈值则判为异常.

1 问题描述与相关工作

本小节具体阐述时间序列异常检测算法的相关背景知识,对本文拟解决的具体问题进行数学描述,并对相关研究工作进行介绍.

1.1 定义与描述

时间序列一般是指某一随机变量按照时间顺序排列而得到的一段序列,可以对时间序列做如下的定义.

定义 1.1 时间序列.我们用按时间顺序排列的一组随机变量 $X_1,X_2,\cdots,X_t,\cdots$ 来表示一个随机事件的时间序列,简记为 $\{X_t,t\in T\}$,用 $x_1,x_2,\cdots,x_n$ 或 $\{x_t,t=1,2,\cdots,n\}$ 表示该随机序列的 n 个有序观察值,称为长度为 n 的观察值序列.

时间序列中的点异常是指时间序列中与其周围点差异较大的数据点,序列异常是指一段时间内的观察值与其周围序列差异较大.

判定一个点是否属于异常时,需要考虑它周围的相关点,另外,有关异常的判定,需要正常序列模型作为样本进行比较,因此,本文再引入时间子序列和滑动窗口的定义

定义 1.2 子序列.给定长度为 n 的时间序列 $X=\{x_t,t=1,2,\cdots,n\}$,定义时间序列 X 的一段子序列 $S_{i,l}$ 为 X 中以点 x_i 向前展开,长度为 $l\leqslant n$ 的一段连续点的集合,即 $S_{i,j}=\{x_{i-l+1},x_{i-l+2},\cdots,x_{i-1},x_i\}$,其中 $l\leqslant i\leqslant n$.

定义 1.3 滑动窗口.给定长度为 n 的时间序列 $X=\{x_t,t=1,2,\cdots,n\}$ 以及 X 中以点 x_i 向前展开,长度为 l 的子序列 $S_{i,l}$,定义滑动窗口 $W_{i,m}$ 为 X 中以点 x_i 向前展开,长度为 $m(l\leqslant m\leqslant n)$ 且包含 $S_{i,l}$ 的连续点的集合,即 $W_{i,m}=\{x_{i-m+1},x_{i-m+2},\cdots,x_{i-l},x_{i-l+1},x_{i-l+2},\cdots x_{i-1},x_i\}$,其中 $m\leqslant i\leqslant n$.

我们首先要解决的问题是在一段时间序列中准确监测出异常点,对当前需要判定是否异常的一段时间子序列,需要判定此段时间子序列是否与其余时间子序列较为相似,为避免与自身比较,引入非欧几里得配对的定义.

定义 1.4 非欧几里得配对.给定长度为 n 的时间序列 X 以及 X 中的时间子序列 $S_{i,l}$ 和滑动窗口 $W_{i,m}$,定义滑动窗口 $W_{i,m}$ 中以点 x_i 向前展开,长度为 l 的时间子序列 $S_{i,j}=\{x_{j-l+1},x_{j-l+2},\cdots,x_{j-1},x_j\}(i-m+l\leqslant j\leqslant i-l)$ 为 $S_{i,l}$ 的非欧几里

得配对.

在定义 1.4 中的非欧几里得性体现在,要求滑动窗口中的两个时间子序列没有交集点的存在. 在衡量滑动窗口中非欧几里得配对间的相似性时,可以使用基于距离的度量方式,在数学上,距离的定义如下.

定义 1.5　距离. 设 S 是非空集合,对 S 中任意的两个元素 x 与 y,按某一法则都对应唯一的实数 $d(x,y)$,而且满足下属 3 条公理:

(Ⅰ)非负性:$d(x,y)\geqslant 0$,且 $d(x,y)=0$ 当且仅当 $x=y$

(Ⅱ)对称性:$d(x,y)=d(y,x)$

(Ⅲ)三角不等式:对任意的 $x,y,z\in S$,恒有 $d(x,y)\leqslant d(x,z)+d(y,z)$

时间序列中基于形状的距离计算方法有欧式距离、余弦距离、皮尔逊相关系数以及对时间序列进行 SAX(symbolic time series representations)变换[18]后的 LCS(longest common subsequence)等,本文考虑到算法的使用效果以及算法特点,采用欧氏距离度量.

定义 1.6　欧式距离. 给定长度为 n 的时间序列 X 以及 X 中两个长度为 l 的时间子序列 $S_{i,l}$ 和 $S_{j,l}$,定义两者间的欧式距离为

$$\text{EuclideanDist}(S_{i,l},S_{j,l})=\sqrt{\sum_{q=0}^{l-1}(x_{i-q}-x_{j-q})^2} \tag{1}$$

式中,$l\leqslant i,j\leqslant n$.

接下来本文将给出一段时间序列中异常点的定义,首先引入最相似非欧几里得配对和异常性.

定义 1.7　最相似非欧几里得配对. 给定以点 x_i 为起始点,长度为 l 的时间子序列 $S_{i,l}$ 以及以点 x_i 展开长度为 m 的滑动窗口 $W_{i,m}$,定义 $S_{i,l}$ 的最相似非欧几里得配对 $S'_{i,l}$ 为滑动窗口 $W_{i,m}$ 中与 $S_{i,l}$ 欧氏距离最小的非欧几里得配对子序列,即

$$\begin{aligned} &S'_{i,l}=S_{j,l}\\ &\text{s.t. EuclideanDist}(S_{i,l},S_{j,l})=\\ &\qquad\min\{\text{Euclidean Dist}(S_{i,l},S_{p,l})\} \end{aligned} \tag{2}$$

式中,$S_{p,l}$ 为滑动窗口 $W_{i,m}$ 中 $S_{i,l}$ 的非欧几里得配对.

定义 1.8　异常性. 给定长度为 n 的时间序列 $X=\{x_t,t=1,2,\cdots,n\}$ 以及以点 x_i 为起始点,长度为 l 的时间子序列 $S_{i,l}$,定义点 x_i 的异常性为在以点 x_i 向前展开,长度为 m 的滑动窗口 $W_{i,m}$ 中 $S_{i,l}$ 与 $S_{i,l}$ 的最相似非欧几里得配对 $S'_{i,l}$ 之间的欧氏距离,即 $\text{Anomaly}(x_i)=\text{EuclideanDist}(S_{i,l},S'_{i,l})$,其中 $l\leqslant i\leqslant n$.

当给定滑动窗口和时间子序列长度后,可以计算时间序列中各点的异常性,并且发现,时间序列中的异常点的异常性相较于正常点取值更大. 当异常情形出现时,异常性明显增大,由此,需要引入异常点的定义.

定义 1.9　异常点. 给定长度为 n 的时间序列 $X=\{x_t,t=1,2,\cdots,n\}$,时间子序列长度 l,滑动窗口长度 m,以及阈值 K,定义 x_i 为异常点当且仅当 $\text{Anomaly}(x_i)>K$.

1.2　异常检测相关工作

时间序列异常检测技术在多领域都有所应用,如在心电图数据中寻找异常可以协助医生快速发现病人的异常情况[3],在客户信用卡记录中的异常检测可以帮助银行确定客户是否存在信用问题等. 过去较多的时间序列异常检测算法使用了传统的时间序列分析方法进行异常检测,包括 ARIMA[4]、指数平滑[5]、时间序列分解[6-7]等. 另外一些研究成果使用统计方法进行异常检测,如 PCA[8]、线性回归[9-10]、极值理论[11]、中位数理论[12]等. 近年机器学习算法的快速发展,许多机器学习算法如神经网络[13]、SVDD[14]、DBSCAN[10]等算法也应用于时间序列异常检测.

在工业界,互联网公司会根据自身业务需要进行时间序列分析平台的开发. Yahoo 的时间序列分析系统 EGADS[15]包含了时间序列建模、时间序列异常检测、预警系统三大模块,集成了多种时间序列分析及异常检测算法. Twitter 在 2014 年提出了一种基于时间序列分解和极大学生差检验的异常点检测算法[6]以及一个与之互补的对时间序列中突变点的检测算法,并提供了两种算法的 R 语言代码库[16-17]. Netflix 为保证公司所得数据的有效性开发了异常检测系统 Surus,并开源了其中的算法 RAD[18],该算法主要使用 RPCA[19]进行异常点的检测. Linkedin 的开源工具 luminol[20]是轻量级的时间序列分析库,主要支持异常检测和相关性分析.

不仅在学术界还是工业界,衡量异常检测算法的优劣性都要从以下几方面进行考虑:

(Ⅰ)准确度. 异常检测本质上是一个二分类问题,对于任意一个时间点,错误的分类都可能会导致难以弥补的后果,因此算法准确性的衡量是首要的.

在衡量异常检测算法的准确性时，通常使用的是精确率(Precision)、召回率(Recall)以及前两者的调和平均数 F_1-score.

(Ⅱ)在线算法或离线算法. 为了维持异常检测算法的准确度，需要不断地更新模型和参数，因此异常检测算法是否具有在线实时性也是需要考虑的一项关键因素.

(Ⅲ)算法参数. 通常的回归和分类算法都需要设置某些参数，如 K 近邻算法中的 K 值，参数取值对模型算法效果产生决定性的影响，异常检测算法也不例外，一个模型如果受影响的参数个数较少，且模型效果受参数大小设置影响较小，那无疑是一个更为稳定的模型算法.

(Ⅳ)时间复杂度. 异常的检测不仅需要准确，也要快速，因此算法时间复杂度也同样需要考虑[21].

异常检测算法的主要目的是能够准确快速地检测出一段时间序列中的异常点，现有的很多算法是通过对正常点进行建模，通过比较当前分析的数据与正常数据的差异性，差异过大则被认为是异常点. 基于这一思想，数据挖掘中的许多模型算法被应用在检测时间序列异常中，如线性回归[9-10]、支持向量机[14]、神经网络[13]等. 这些算法通常将得到的时间序列数据划分为训练集和测试集，在训练集中训练得到模型并在测试集上进行检测. 实际应用中，时间序列数据是流式的且不断变化的，因此，训练得到的模型如果不实时更新将会很难应用在新的数据中，另外，正常与异常数据的多变性导致很难使用单一的模型来准确检测异常点. 综上所述，在线的异常检测算法不仅能够在流式数据中运行，还可以自我不断更新模型，更加适用于实际情形的需要.

2 基于距离的在线异常检测算法

本节具体介绍本文提出的基于距离的在线异常检测算法，首先对算法的主要思想进行阐述，然后通过形式化定义给出本文提出的算法.

2.1 主要思想

分析智能运维中真实时间序列异常数据可以发现，异常点通常会相较于其周围点表现出取值的突然增大或突然减小，而异常序列通常表现出一段点的走势相较其周围点(尤其是前面的点)明显不同，如图 1 所示.

根据上述异常数据的特点，本文提出了一种基于距离的在线异常检测算法，其主要思想是对给定

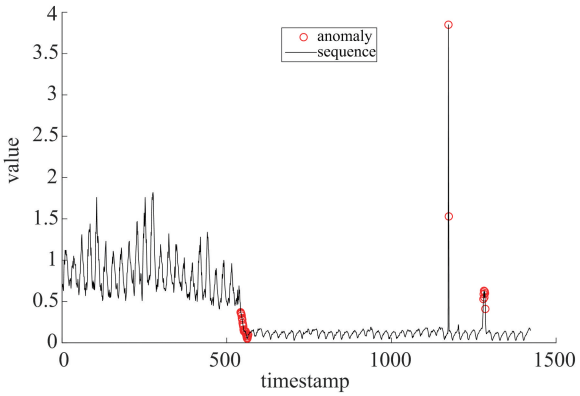

图 1 包含异常点和异常序列的真实运维数据

Fig. 1 A real operation data that contains both abnormal points and subsequences

时间序列中每一点，计算由该点向前展开的子序列在滑动窗口中的异常性. 一般来说，异常数据相较于正常数据会明显增大，算法利用某一特定的统计分布来拟合时间序列中各点向前展开的时间子序列在整条时间序列中的异常性，当异常数据偏离该分布中心较大时，该点被视为异常点.

2.2 滑动窗口中的异常性度量

算法 2.1 给出了利用滑动窗口机制计算时间序列中各点异常性的计算方法，伪代码如下：

算法 2.1 AnomalyCalculate

Input: Time Series (x_t), Sliding Window length m, Subsequence length l

Output: Anomaly(x_t)

```
1 Function Anomaly(x_t)=AnomalyCalculate(x_t,m,l)
2 Anomaly(x_t)=0⃗
3 For i=m to n
4    nearest_neighbor_dist=infinity
5      For j=i−m+1 to i−1
6          IF S_{i,l} is not anomaly & EuclideanDist(S_{i,l}, S_{j,l})<nearest_neighbor_dist
7          nearest_neighbor_dist=EuclideanDist(S_{i,l}, S_{j,l})
8      End
9 Anomaly(x_i)←nearest_neighbor_dist
10 End
```

AnomalyCalculate 算法计算时间序列中各点异常性的具体方法是：通过给定长度为 n 的时间序列 $X=\{x_t, t=1,2,\cdots,n\}$、滑动窗口长度 m、子序列长度 l，那么对时间序列中所有下标在区间 $[m, n]$ 的点，即 $\{x_i, i=m, m+1,\cdots,n\}$，有以点 x_i 向前展开长度为 l 的子序列 $S_{i,l}$ 以及以点 x_i 向前展开

长度为 m 的滑动窗口 $W_{i,m}$,其中 $m \leqslant i \leqslant n$. 在 $W_{i,m}$ 中找到与 $S_{i,l}$ 欧氏距离最小的非欧几里得配对子序列,即最相似非欧几里得配对 $S'_{i,l}$,将 $S_{i,l}$ 与 $S'_{i,l}$ 之间的欧式距离大小即 EuclideanDist($S_{i,l}$, $S'_{i,l}$) 作为点 x_i 的异常性 Anomaly(x_i),同定义 1.8 的描述,为了防止当前点与之前的异常点做比较,算法 2.1 限定最相似非欧几里得配对均为正常点.

通过算法 2.1 计算给定时间序列中各点的异常性后发现,当异常点或者异常序列出现时,异常会明显增大.

2.3　阈值选择机制

由前述可知,基于距离的异常检测算法当有异常点或者异常序列出现时,异常性会明显增大. 实际情形下得到的运维数据是流式数据,仅知道异常增大是不够的,增大到多少才算异常,或者增大到多少需要报警,这需要确定时间序列中每个点的异常阈值. 本文提出了两种阈值确定方式. 假定一段时间序列中每个点的异常服从一定分布,异常点或者异常序列的异常取值较大因此会落在该分布的 $1-\alpha(0<\alpha<1)$ 分位点之外,当异常性落在该部分时将被视为异常点或异常序列. 这可以使用正态分布和对数正态分布来分别拟合时间序列中各时间点的异常分布情况.

本文使用均值和方差的递归计算方法[22]对分布的均值和方差进行估计,计算公式为

$$\left.\begin{aligned}\mu_n &= \frac{1}{n}x_n + \frac{n-1}{n}\mu_{n-1}\\ S_n^2 &= \frac{n-1}{n}S_{n-1}^2 + \frac{n-1}{n^2}(x_n-\mu_{n-1})^2\end{aligned}\right\} \tag{3}$$

对于新元素的加入,本文在递归的均值和方差计算公式中引入遗忘因子 λ,则引入遗忘因子后,均值和方差为

$$\left.\begin{aligned}\mu_n &= \sum_{i=1}^{n}\frac{\lambda^{n-i}x_n}{\sum_{i=1}^{n}\lambda^{n-i}}\\ S_n^2 &= \sum_{i=1}^{n}\frac{\lambda^{n-i}(x_i-\mu_n)^2}{\sum_{i=1}^{n}\lambda^{n-i}}\end{aligned}\right\} \tag{4}$$

引入遗忘因子后的递归计算方法为

$$\left.\begin{aligned}\mu_n &= \frac{1-\lambda}{1-\lambda^n}x_n + \frac{\lambda(1-\lambda^{n-1})}{1-\lambda^n}\mu_{n-1}\\ S_n^2 &= \frac{\lambda(1-\lambda^{n-1})}{1-\lambda^n}S_{n-1}^2 +\\ &\quad \frac{\lambda(1-\lambda^{n-1})(1-\lambda)}{(1-\lambda^n)^2}(x_n-\mu_{n-1})^2\end{aligned}\right\} \tag{5}$$

2.4　完整算法

算法 2.1 是按照时间顺序依次计算给定时间序列中下标在区间 $[m,n]$ 内的异常点,利用 2.3 节中给出的两种拟合分布以及均值方差的递归计算方法,可以在计算时间序列中各点异常性的同时,对两种拟合分布的参数进行在线估计以及更新. 由于最先开始递归计算的均值和方差波动较大,可能导致拟合分布的不稳定性,影响检测效果,因此完整的算法引入了一段过渡期,在过渡期内不进行均值和方差的更新. 均值和方差的初始值是由在过渡期内点的异常性计算的,过渡期之后的均值和方差由公式(3)或公式(5)更新.

算法 2.2(AnomalyDetect)给出了在计算时间序列中各异常点的同时,对异常分布进行拟合并估计参数,以及划定阈值并判定给定点是否异常的具体步骤,具体下所示:

算法 2.2　AnomalyDetect

Input: Time Series (x_t), Sliding Window length m, Subsequence length l, transition t, anomaly threshhold $\alpha \in (0,1)$

Output: Anomaly flags $\vec{f}$

```
1 Function Anomaly(x_t)=AnomalyDetect(x_t,m,l)
2 Anomaly(x_t)←0⃗
3 Anomaly flags f⃗←0→
4 transition_dist←null array
5
6 For i=m to n
7    nearest_neighbor_dist=infinity
8    For j=i−m+1 to i−1
9      IF S_{j,l} is not anomaly & EuclideanDist(S_{i,l},S_{j,l})
<nearest_neighbor_dist
10       nearest_neighbor_dist=EuclideanDist(S_{i,l},S_{j,l})
11       End
12   End
13 Anomaly(x_t)←nearest_neighbor_dist
14 IF i<m+t
15   transition_dist+[nearest_neighbor_dist]
16 End
17 IF i=m+t
18   μ←mean(transition_dist)
19   σ²←var(transition_dist)
20 End
21 IF i≥m+t
22   Calculate the α−quantile z_α of N(μ,σ²) or LN(μ,
```

σ^2)

23 IF nearest_neighbor_dist$>z_\alpha$

24 Anomaly flags $\vec{f}_i \leftarrow 1$

25 Else

26 Update μ and σ^2

27 End

28 End

算法 2.2 是在算法 2.1 的基础上加入了异常性分布拟合、参数估计和划定阈值，进而进行异常检测的完整过程. 图 2 展示了算法 2.2 的具体实现过程. 如图 2 所示，对于由当前点向前展开的时间子序列，算法 2.2 在由当前点向前展开的滑动窗口中寻找它的最相似非欧几里得配对，计算两个子序列的欧式距离作为当前点的异常性，并计算该异常在拟合的异常分布中偏离中心的程度，当偏离过多时，当前点被认为是异常点. 同样，为了防止与异常点比较相似性，这里限定最相似非欧几里得配对均为正常点.

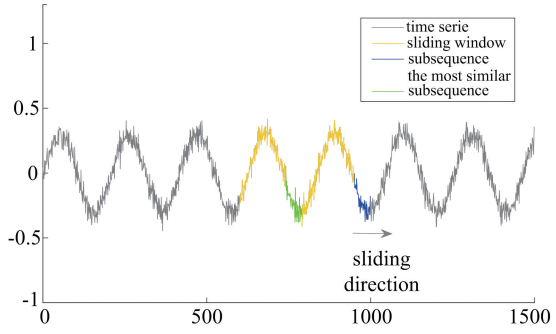

图 2 算法 2 的具体实现过程

Fig. 2 The specific implementation of AnomalyDetect algorithm

3 实验结果与分析

本节在实际运维时间序列数据集上开展异常检测算法的实验，并对结果进行分析.

3.1 数据准备

本文使用雅虎公司提供的实际运维数据集[23]，该数据集包含 4 个子文件夹，其中 A1 数据集包含 67 条真实的运维时序数据，A2、A3、A4 数据集各包含 100 条合成数据集，合成数据集由周期性、趋势性、噪声相加而成，合成的时间序列数据复杂程度逐渐加大.

研究雅虎 S5 数据中的合成数据发现，A2、A3、A4 数据集是在正常的合成数据中随机更改正常点生成异常数据点，而异常点又分为两类：anomaly 与 changePoint. A2 数据集中的正常数据是由简单的趋势和单周期的周期数据以及噪声数据相加而成，异常点类型只有 anomaly 一种；A3 数据集的正常数据由单一的趋势和 3 条不同周期、不同振幅的周期数据以及噪声数据相加而成，异常点类型只有 anomaly 一种；A4 数据集由一条趋势和 3 条变化周期、变化振幅的周期数据以及噪声数据相加而成，异常点类型包括 anomaly 和 changePoint 两种.

3.2 评价准则

时间序列中的异常检测问题可以看作一个典型的二分类问题，时间序列中的每一点，在真实情况下被标记为正常或异常，同时也可以被预测为正常或异常，若将正常标记为 0，异常标记为 1，则真实情况与预测情况可以进行频数统计为列联表，即混淆矩阵.

时间序列中的每一点根据真实情况和预测情况均可被标记为 TP、FN、FP、TN，则评价准则准确率 Precision 和召回率 Recall 计算公式为

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}};\ \text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{6}$$

F_1 score 是另一种衡量二分类模型精确度的指标，它兼顾了精确率 Precision 和召回率 Recall 是两者的调和平均数，计算公式为

$$F_1=2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{7}$$

本文在分类准确度上使用准确度 Precision、召回率 Recall 和 F_1 score 这三个指标.

3.3 实验结果与分析

本文使用算法 2.2 在雅虎 S5 数据的时间序列数据上进行实验，异常检测将在一个滑动窗口和过渡期之后进行，在检测的同时，随着均值和方差的波动，异常的分布将会有稍许变化. 检测的最后，可以得到一条时间序列中各点的异常性以及最终迭代估计出的异常性分布情况.

表 1 列出了利用算法 2.2 在雅虎 S5 数据集的检测效果，其中每条时序数据均在固定了滑动窗口大小(200)、时间子序列长度(3)、过渡期(50)以及遗忘因子(1)后进行异常值的检测，每条时序数据中实际的检测点均在滑动窗口和过渡期之后，同时表 1 还给出了 4 个数据集的准确率、召回率和 F_1 score.

从表 1 可以看出，算法 2.2 的检测准确率非常高，在 A1～A3 数据集上均达到了 0.95 以上，尤其是在真实的运维数据集 A1 上 F_1 score 也达到了 0.96，在 A4 也达到了 0.75 以上. 这样的情形与算法 2.2 的基本思想有关. AnomalyDetect 主要基于时间序列中

子序列形态相似性的比较,而不仅仅是相邻点的相关性. 不仅如此,通过对子序列长度的调整,AnomalyDetect 可以实现对异常点和异常序列的通用检测. AnomalyDetect 也有其不足性,在 A4 数据集上的检测效果不够完美,这与 A4 数据集复杂多变的形态有关.

本文使用两种开源异常检测算法作比较,表 2 列出了 Twitter 公司异常检测算法 S-H-ESD[16],Netflix 公司的异常检测算法 RPCA[18] 在 Yahoo Webscope S5 数据集上的检测准确率.

表 1　算法 2 在 Yahoo Webscope S5 数据集上的检测结果

Tab. 1　Detection results of the algorithm 2. 2 on dataset Yahoo Webscope S5

dataset	algorithm	TP	FP	FN	Precision	Recall	F_1 score
A1	norm	1 543	10	111	0. 932 9	0. 993 6	0. 962 3
	lognorm	1 543	10	111	0. 932 9	0. 993 6	0. 962 3
A2	norm	461	0	5	0. 989 3	1	0. 9946
	lognorm	461	1	5	0. 989 3	0. 997 8	0. 993 5
A3	norm	766	30	37	0. 953 9	0. 962 3	0. 958 1
	lognorm	766	29	37	0. 953 9	0. 963 5	0. 958 7
A4	norm	572	66	315	0. 644 9	0. 896 6	0. 750 2
	lognorm	577	68	281	0. 672 5	0. 894 6	0. 767 8

表 2　S-H-ESD 和 RPCA 在雅虎数据集上的实验结果

Tab. 2　Detection results of the algorithms S-H-ESD and RPCA on dataset Yahoo Webscope S5

dataset	algorithm	TP	FP	FN	Precision	Recall	F_1 score
A1	S-H-ESD	571	1 147	664	0. 462 3	0. 332 4	0. 386 7
	RPCA	636	1 033	1 655	0. 277 6	0. 381 1	0. 321 2
A2	S-H-ESD	140	172	43	0. 765	0. 448 7	0. 565 7
	RPCA	447	19	49	0. 901 2	0. 959 2	0. 929 3
A3	S-H-ESD	86	583	0	1	0. 128 6	0. 227 8
	RPCA	509	434	77	0. 868 6	0. 539 8	0. 665 8
A4	S-H-ESD	112	454	1 156	0. 088 3	0. 197 9	0. 122 1
	RPCA	432	405	411	0. 512 5	0. 516 1	0. 514 3

S-H-ESD 算法是一种离线算法,主要基于时间序列分解以及广义 ESD 检验,RPCA 同样是离线算法,主要基于 Robust PCA 算法,这两种算法都需要先得到全部的时间序列数据,再对该数据中的异常点进行检测. 将 4 种算法在雅虎 S5 数据集上的 F_1 score 绘制成条形图可以更加清晰地比较 4 种算法的检测效果,如图 3 所示.

由图 3 可以看出,AnomayDetect 在准确率上明显优于其他算法,从数据集上看,4 种算法在 A2 数据集的表现效果都是最好的,这是因为 A2 数据集的周期性和趋势性都比较明显,噪声影响较小,而异常点相较于正常点来说差异较大,因此检测比较容易. 另外,从图 3 可以看出,AnomalyDetect 在 A1

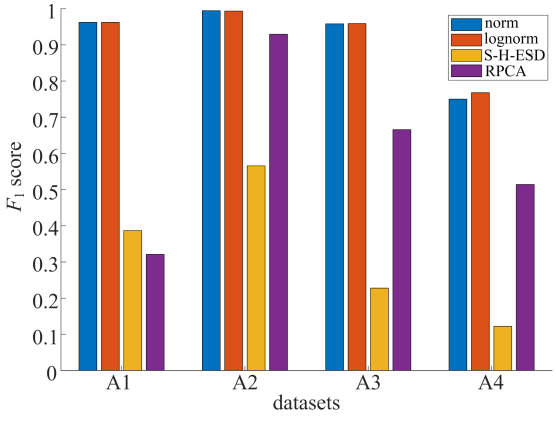

图 3　不同算法在 Yahoo Webscope S5 数据集上的 F_1 score

Fig. 3　The F_1 score of different algorithms on Yahoo webscope S5 datasets

数据集上优势显著，A1 是真实的运维数据，不同于合成数据集，A1 的时序形态复杂多变并契合真实情形，而 AnomalyDetect 这种创新的基于距离的算法更加适用于真实的时序数据.

图 4 为这两种算法的 PRC（Precision-Recall curve）曲线，因为异常序列中异常点所占比例过小，在这种不平衡数据中，PRC 可以展示算法的本质效果. 由图 4 可以看出，AnomalyDetect 算法的准确率明显高于 RPCA.

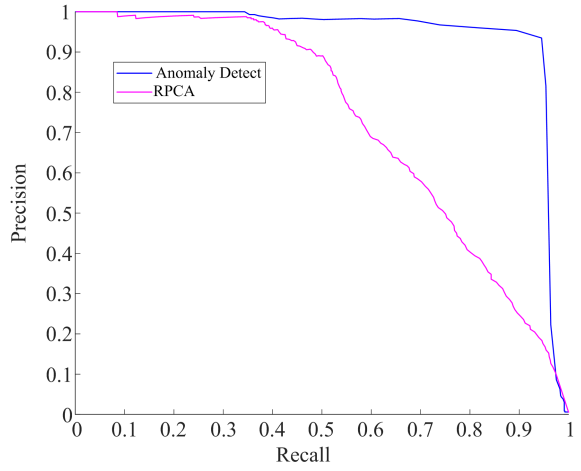

图 4 AnomalyDetect 和 RPCA 在 A3 数据集上的 PRC 曲线

Fig. 4 The PRC curve of AnomalyDetect and RPCA on the A3 dataset

为了更加细致地衡量 AnomalyDetect 算法的检测效果，本文通过多类实验来展示算法结果，图 5 是 3 种算法在同一条数据集上的检测情况. 由图 5 可以看出，S-H-ESD 和 RPCA 的检测效果比较相近，但是对于与正常数据取值相近而形态有差别的异常点，这两种算法效果均不如 AnomalyDetect.

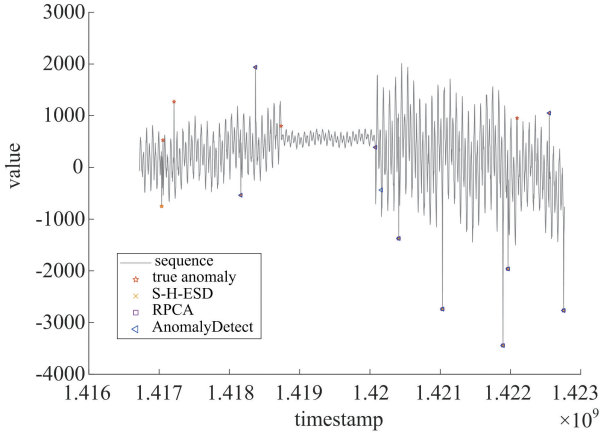

图 5 三种算法在 A3 数据集中同一条时序数据上的检测效果

Fig. 5 The detection effect of three algorithms on the same time series from A3 dataset

参数大小同样对 AnomalyDetect 的效果有影响，AnomalyDetect 的一个主要参数是在更新分布均值和方差时使用到的遗忘因子 λ，图 6 是在 A2 前 20 条时序数据的实验结果，进行比较的是不同遗忘因子下的精确率、召回率和 F_1 score，由图 6 可以看出，遗忘因子对检测效果影响较大，并且检测效果随遗忘因子的上升而变好. 总体来说，遗忘因子取值在 0.95 和 1 之间检测效果较好.

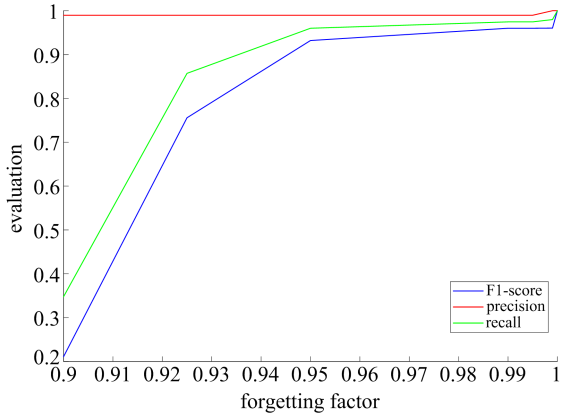

图 6 算法 2 不同遗忘因子对 A2 数据集中前 20 条时序数据的影响

Fig. 6 The effect of forgetting factor on AnomalyDetect. The experiment is performed on the first 20 datasets in A2

另外一个重要的参数是滑动窗口的长度，由上文的分析可知，滑动窗口的长度一定要大于子序列的长度，但是长度大小也要慎重选择. 图 7 是 AnomalyDetect 在 A3 中第 15 条时序数据上的实验结果，主要展示的是不同滑动窗口大小下的 FN、FP、TP，可以看出，各指标随窗口大小的增大会出

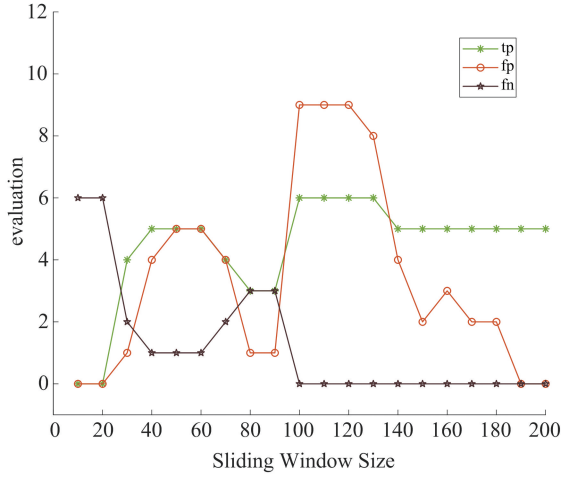

图 7 不同滑动窗口大小在 A3 15 条数据上的检测效果

Fig. 7 Detection of AnomalyDetect under different sliding window sizes on A3-15 dataset

现不同程度的波动.一般来讲,对于周期性较为明显的时序数据,滑动窗口长度应大于一个周期长度,这样可以保证算法的检测效果.

综上所述,相比于其他的异常检测算法,本文算法在以下几个方面表现出明显优势:

(Ⅰ)准确性:通过表 1、表 2 可以看出,算法在 4 个数据集上的 F_1 score 都比较高.

(Ⅱ)通用性:通常一个时间序列异常检测算法并不能适用于所有的时间序列数据,如检测异常点的算法并不能很好地检测出异常序列,但是 AnomalyDetect 基于时间序列相似性的原理却可以同时检测出异常点和异常序列,并在真实数据集上的表现毫不逊色于合成人工数据集.

(Ⅲ)在线性:由 1.2 节的分析可以看出,时间序列异常检测算法的在线性是保证算法实时性和准确性的必然要求,因此 AnomalyDetect 的在线实时更新使其明显优于其他算法.

同样 AnomalyDetect 也有其表现逊色的地方,该算法涉及参数较多,且对参数设置大小较为敏感,由于该算法需要确定阈值,同时模型需要不断更新,在这个过程中参数的大小对算法效果起到比较明显的作用.

4　结论

本文针对真实场景的需要,提出了一种新的在线异常检测算法,该算法通过计算时间序列中子序列的异常性来判断点或序列是否异常,具有在线性和实时更新等特点,通过在 Yahoo Webscope S5 数据集上进行实验,证明该算法具有较高的准确度.

参考文献(References)

[1] ZHANG S L, LIU Y, PEI D, et al. Rapid and robust impact assessment of software changes in large internet-based services[C]// 11th ACM Conference on Emerging Networking Experiments and Technologies. Heidelberg, Germany: ACM, 2015: No. 2(1-15).

[2] LIN J, KEOGH E, LONARDI S, ET AL. A symbolic representation of time series, with implications for streaming algorithms [C]//Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, USA: ACM, 2003: 2-11.

[3] KEOGH E, LIN J, LEE S H, et al. Finding the most unusual time series subsequence: Algorithms and applications[J]. Knowledge and Information Systems, 2007, 11(1): 1-27.

[4] YU Q, LYU J B, JIANG LR. An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks [J]. International Journal of Distributed Sensor Networks, 2016, (28): No. 28.

[5] YAN H, FLAVEL A, GE Z H, et al. Argus: End-to-end service anomaly detection and localization from an ISP's point of view [C]//Proceedings of IEEE INFOCOM. Orlando, USA: IEEE, 2012: 2756-2760.

[6] VALLISO, HOCHENBAUM J, KEJARIWAL A. A novel technique for long-term anomaly detection in the cloud[C]// Proceedings of the 6th USENIX Workshop on Hot Topics in Cloud Computing. Philadelphia, USA: USENIX Association, 2014: 1-6.

[7] CHEN Y Y, MAHAJAN R, SRIDHARAN B, et al. A provider-side view of web search response time[C]// Proceedings of the InternationalConference on SIGCOMM. Hong Kong, China: ACM, 2013: 243-254.

[8] HYNDMAN R J, WANG E, LAPTEV N. Large-scale unusual time series detection[C]//International Conference on Data Mining Workshop. Atlantic, USA: IEEE, 2015: 1616-1619.

[9] THILL M, KONEN W, BÄCK T. Online anomaly detection on the Webscope S5 dataset: A comparative study [C]// Evolving and Adaptive Intelligent Systems. Ljubljana, Slovenia: IEEE, 2017: 1-8.

[10] WATSON SM, TIGHT M, CLARK S, et al. Detection of outliers in time series[Z]. Working paper 362, Institute of Transport Studies, University of Leeds, 1991.

[11] SIFFER A, FOUQUE P A, TERMIER A, et al. Anomaly detection in streams with extreme value theory [C]// 23rd International Conference on Knowledge Discovery and Data Mining. Halifax, Canada: ACM, 2017: hal-01640325.

[12] SAGOOLMUANG A, SINAPIROMSARAN K. Median-difference window subseries score for contextual anomaly on time series [C]// 8th International Conference of Information and Communication Technology for Embedded Systems. Chonburi, Thailand: IEEE, 2017: 10. 1109/ICTEmSys. 2017. 7958772.

[13] SUH S, CHAE D H, KANG HG, et al. Echo-state conditional variationalautoencoder for anomaly detection [C]// International Joint Conference on Neural Networks. Vancouver, Canada: IEEE, 2016: 1015-1022.

(下转第 571 页)