基于全变分比分隔距离的时序数据异常检测

徐天慧，郭强，张彩明

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210600174

基于全变分比分隔距离的时序数据异常检测

徐天慧[1]　郭　强[1]　张彩明[2]
1 山东财经大学计算机科学与技术学院　济南 250014
2 山东大学软件学院　济南 250014
(xutianhui06@qq.com)

摘　要　时序数据异常检测是数据分析的重要研究问题之一,其主要挑战在于利用数据点上下文准确判断数据是否存在异常,若存在异常则低时延定位该异常。现有检测方法通常利用概率密度比来度量序列间的相似性,以捕捉异常,这些方法需借助交叉验证法来估计概率密度比模型中的参数。交叉验证法会提高计算复杂度,导致计算效率较低,且存在较大检测时延。针对上述问题,提出了一种基于全变分比分隔距离的检测方法。该方法采用全变分提取序列波动特征,以此为基础计算全变分比分隔距离来度量序列间的相似性,从而提高计算效率,并实现低时延定位异常。针对噪声干扰问题,将检测方法与相对全变分相结合以增强检测方法的鲁棒性,从而进一步提高该方法的检测准确度。实验结果表明,该方法在检测准确度、低时延以及计算效率3个方面均取得了较好的效果。

关键词:异常检测;概率密度比;时延;全变分;相对全变分
中图法分类号　TP391

Time Series Data Anomaly Detection Based on Total Variation Ratio Separation Distance

XU Tian-hui[1],GUO Qiang[1] and ZHANG Cai-ming[2]
1 School of Computer Science and Technology,Shandong University of Finance and Economics,Jinan 250014,China
2 School of Software,Shandong University,Jinan 250014,China

Abstract　Anomaly detection for time series data is one of the important research problems in data analysis. Its main challenge is to detect if there are any anomalies and locate anomalies with low delay according to context. Most of existing anomaly detection methods capture anomalies using the probability density ratio to measure similarity between sequences. These methods need to use the cross-validation method to estimate the parameters of probability density ratio. However,cross-validation can increase the computational complexity,resulting in low computational efficiency and a high time delay. To address these issues,this paper proposes a detection method based on total variation ratio separation distance,in which total variation is adopted to extract sequence fluctuation features. Due to the fact that the total variation ratio is better than probability density ratio,the proposed method achieves higher computational efficiency and lower time delay. To reduce noise interference and further improve the detection accuracy,the proposed method is combined with the relative total variation. Experimental results show that the proposed method performs well in terms of detection accuracy,low delay and computational efficiency.

Keywords　Anomaly detection,Probability density ratio,Time delay,Total variation,Relative total variation

1　引言

时序数据是一组按时间顺序展开的观测值的集合。当序列中某个时间点的数据偏离原有数据分布或出现不符合预期行为的数据模式时,则认为时序数据在该时间点发生异常,异常的存在会给原有正常工作进程带来一定隐患,如果时序数据中的异常不能被准确、及时地识别,很可能会引起重大损失。使用某方法检测出时序数据中的异常并确定该异常发生时间点的过程被称为时序数据异常检测,其被广泛应用于日常生活和工业生产等方面,如人类行为活动识别[1]、Web服务器稳定性监测[2]、传感数据异常实时监测[3]、时空轨迹异常检测[4]、人群异常识别[5]等。

到稿日期:2021-06-22　返修日期:2021-10-15
基金项目:国家自然科学基金(61873145,61802229);山东省自然科学省属高校优秀青年人才联合基金项目(ZR2017JL029);山东省高等学校青创科技支持计划(2019KJN045)
This work was supported by the National Natural Science Foundation of China(61873145,61802229),Natural Science Foundation of Shandong Province for Excellent Young Scholars(ZR2017JL029) and Science and Technology Innovation Program for Distinguished Young Scholars of Shandong Province Higher Education Institutions(2019KJN045).
通信作者:郭强(guoqiang@sdufe.edu.cn)

由于异常检测在数据分析中的重要性，对检测方法的研究引起了国内外学者的广泛关注[6-8]。异常检测方法大致可分为监督学习和无监督学习两类。监督学习方法通过对带有标签的数据进行训练，学习得到从输入数据到输出标签的映射关系，进而将异常检测问题转化为分类问题[9]。由于时序数据中异常数量往往远少于正常数据点的数量，从而导致标签数量严重不均衡，且在很多实际问题中数据不带有标签，这使得采用监督学习方法进行异常检测缺乏一定的实用性。

无监督异常检测方法不依赖于数据标签，其基本策略是定义正常数据模式，判断待检测数据点是否与正常数据模式一致。若两者一致，则该点为正常点，否则将该点判定为异常[10]。由于时序数据中普遍存在噪声，且时序数据动态性强，因此很难定义正常数据模式，从而导致异常容易被漏判和误判，这给异常检测带来了极大挑战[11]。为此，学者们提出了一些较为有效的异常检测方法[12-16]，最常用的检测方法主要是基于相似性度量的方法，这些方法的核心是定义相似性度量标准，基于此标准度量两个子序列间的相似性，进而判断序列内是否存在异常。若两者的相似度较小，则可认为其中某一序列含有异常；否则两者均无异常。常用的度量标准有最大均值差异(Maximum Mean Distribution，MM)[17]、KL 散度(Kullback-Leibler Divergence，KL)[18-19]、皮尔逊散度(Pearson Divergence，PE)[20]以及分隔距离(Separation Distance，SE)[21]等。此类基于相似性度量的检测方法无需定义正常数据模式，灵活性较高。然而，求解子序列间 MM 差异、KL 散度、PE 散度和 SE 距离会涉及到核函数的运算或学习参数的过程，从而导致异常检测方法的计算效率降低。为提高检测方法的计算效率，本文提出了度量子序列相似性的全变分比分隔距离指标。该指标采用全变分(Total Variation，TV)刻画子序列的波动特征[22]，并利用全变分比度量子序列间的相似性，以此捕捉异常。此外，为增强异常检测方法对噪声的抗干扰能力，我们引入相对全变分(Relative Total Variation，RTV)平滑数据，进一步提高检测方法的鲁棒性。

2 相关工作

目前，较为先进的时序数据异常检测方法主要是基于相似性度量的方法，此类方法的关键是定义某种度量子序列间相似性的标准。一种常用标准是 MM，其本质是寻找一个函数 f，将子序列映射到再生核希尔伯特空间 $\mathscr{F}$ 中，使得在该空间中子序列间的均值差异达到上界。给定参考序列 $\boldsymbol{x}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_t,\cdots,\boldsymbol{x}_n\}$ 和检测序列 $\boldsymbol{y}=\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_t,\cdots,\boldsymbol{y}_n\}$，其中样本 $\boldsymbol{x}_t,\boldsymbol{y}_t\in R^d$，两者间的 MM 定义为：

$$MM[\mathscr{F},x,y]=\sup_{f\in\mathscr{F}}\{E[f(x)]-E[f(y)]\}$$

其中，$E[\cdot]$表示期望。当序列 x 和序列 y 间的相似性较大时，$MM[\mathscr{F},x,y]$较小，则序列 y 中存在异常的可能性较小；否则序列 y 存在异常的可能性较大。利用再生核希尔伯特空间的再生性质，函数 $f(\cdot)$可由核函数求得，进而得到 MM^2 的无偏估计[17]，即：

$$MM_u^2[\mathscr{F},X,Y]=\frac{1}{n(n-1)}\sum_{i,j=1,i\neq j}^{n}h(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{y}_i,\boldsymbol{y}_j)$$

其中，$h(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{y}_i,\boldsymbol{y}_j)=g(\boldsymbol{x}_i,\boldsymbol{x}_j)+g(\boldsymbol{y}_i,\boldsymbol{y}_j)-g(\boldsymbol{x}_i,\boldsymbol{y}_j)-g(\boldsymbol{x}_j,\boldsymbol{y}_i)$，$g(\cdot)$为核函数，$n$ 为序列 x 和序列 y 的长度。MM_u^2 的大小反映序列 x 和序列 y 相似程度的高低，并以此判断序列 y 中是否存在异常。当且仅当序列 x 和序列 y 相同时，MM_u^2 为 0。

对于长度为 n 的检测序列，计算其 MM_u 的时间复杂度为 $O(n^2)$。为降低计算复杂度，Li 等[23]提出了基于 M 统计量的异常检测方法。其核心思想是给定长度均为 n 的 N 个参考序列 $x^i(i=1,2,\cdots,N)$和 1 个检测序列 y，改变序列窗口的大小 $k=2,3,\cdots,n$ 得到参考子序列$\boldsymbol{x}_k^i$ 和检测子序列$\boldsymbol{y}_k$，并计算$\boldsymbol{y}_k$ 与$\boldsymbol{x}_k^i$间的平均差异 Z_k。

$$Z_k=\frac{1}{N}\sum_{i=1}^{N}MM_u^2[\mathscr{F},\boldsymbol{x}_k^i,\boldsymbol{y}_k]$$

在此基础上，M 统计量的计算式如下：

$$M=\max_{k\in\{2,3,\cdots,n\}}\frac{Z_k}{\sqrt{Var[Z_k]}}$$

假设 $k=t$ 时，$\frac{Z_k}{\sqrt{Var[Z_k]}}$取得最大值，说明此时检测子序列$\boldsymbol{y}_t$ 与参考子序列$\boldsymbol{x}_t^i$ 的平均相似性最小，即序列 y 在点 t 发生异常的概率最高。比较 M 统计量与给定阈值的大小以进一步判断点 t 是否为异常点，若 M 统计量大于该阈值，则点 t 为异常点；否则序列 y 中不存在异常。

M 统计量方法的时间复杂度为 $O(n)$，相比 MM_u，其时间复杂度得到大幅度降低，但 M 统计量仅记录了点 t 处检测子序列与参考子序列的差异，因此适用于检测含单个异常的时序数据。针对这一问题，文献[18]提出了基于 KL 重要性估计的方法(Kullback-Leibler Importance Estimation Procedure，KLIEP)，该方法采用 KL 散度度量子序列间的相似性，进而检测多异常。该方法通过设置一个大小为 k 的窗口，分割检测序列 $y=\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_t,\cdots,\boldsymbol{y}_n\}$，样本$\boldsymbol{y}_t\in R^d$，得到多个子序列向量$\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_t,\cdots,\boldsymbol{Y}_n$，其中，$\boldsymbol{Y}_t=[\boldsymbol{y}_{t-k+1}^{\mathrm{T}},\cdots,\boldsymbol{y}_{t-1}^{\mathrm{T}},\boldsymbol{y}_t^{\mathrm{T}}]^{\mathrm{T}}\in\mathbb{R}^{kd}$，$(\cdot)^{\mathrm{T}}$ 表示矩阵的转置。图 1 给出了滑动窗口分割检测序列的过程，其中虚线矩形表示窗口，窗口中时间点 t 的集合记为 $R(t)$。

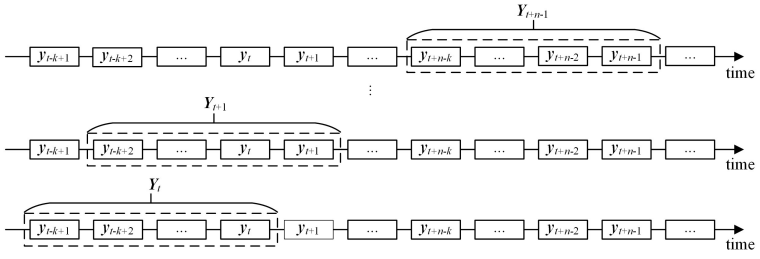


图 1 窗口分割检测序列的过程

Fig. 1 Process of window segmenting detection

令 P_{t-1} 和 P_t 分别表示两个相邻子序列 $\boldsymbol{Y}_{t-1}$ 和 $\boldsymbol{Y}_t$ 的概率分布，则 P_{t-1} 和 P_t 的 KL 散度为：

$$KL(P_{t-1} \| P_t) = \int P_{t-1}(\boldsymbol{Y}) \log_a \left(\frac{P_{t-1}(\boldsymbol{Y})}{P_t(\boldsymbol{Y})} \right) \mathrm{d}\boldsymbol{Y}$$

当 P_{t-1} 和 P_t 相似时，其概率密度比接近于 1，对其取 $\log_a(\cdot)$ 后 $KL(P_{t-1} \| P_t)$ 近似为 0；反之，$KL(P_{t-1} \| P_t)$ 的大小偏离 0。由于很难直接得到 P_{t-1} 和 P_t，因此利用核函数对概率密度比进行建模。

$$w_t(\boldsymbol{Y}, \boldsymbol{\theta}) = \sum_{i=1}^{k} \theta_i \prod_{j=1}^{k} G(\boldsymbol{Y}_t^i, \boldsymbol{Y}_{t-1}^i)$$

其中，$\boldsymbol{Y}_t^i$ 表示子序列 $\boldsymbol{Y}_t$ 中的第 i 个样本，$G(\cdot)$ 表示带宽为 σ 的高斯核函数，即：

$$G(\boldsymbol{Y}_t, \boldsymbol{Y}_{t-1}) = \exp\left(-\frac{\| \boldsymbol{Y}_t - \boldsymbol{Y}_{t-1} \|^2}{2\sigma^2} \right)$$

给定训练样本，KLIEP 利用迭代梯度法极小化以下函数来估计参数 $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_k)^{\mathrm{T}}$。

$$\min_{\boldsymbol{\theta}} \int P_{t-1}(\boldsymbol{Y}) \log_a \left(\frac{P_{t-1}(\boldsymbol{Y})}{P_t(\boldsymbol{Y}) \cdot w_t(\boldsymbol{Y}, \boldsymbol{\theta})} \right) \mathrm{d}\boldsymbol{Y} \tag{1}$$

KLIEP 不依赖参考数据，仅利用数据上下文记录每对相邻子序列间的 KL 散度，因此该方法可实现多异常检测。尽管如此，该方法中式(1)的极小化涉及迭代求解过程，降低了计算效率。为提高计算效率，Kanamori 等[20]提出了无约束最小二乘重要性拟合的方法(Unconstrained Least-Squares Importance Fitting，uLSIF)。类似于 KLIEP，该方法采用与 KLIEP 相同的概率密度比模型 $w_t(\boldsymbol{Y}, \boldsymbol{\theta})$，但不同的是，该方法通过极小化式(2)来学习参数 $\boldsymbol{\theta}$。

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \int \left(\frac{P_{t-1}(\boldsymbol{Y})}{P_t(\boldsymbol{Y})} - w_t(\boldsymbol{Y}, \boldsymbol{\theta}) \right)^2 P_t(\boldsymbol{Y}) \mathrm{d}\boldsymbol{Y} \tag{2}$$

uLSIF 采用交叉验证法求解 $w_t(\boldsymbol{Y}, \boldsymbol{\theta})$，进而计算子序列间的 PE 散度，即：

$$PE(P_{t-1} \| P_t) = \frac{1}{2} \int P_t(\boldsymbol{Y}) \left(\frac{P_{t-1}(\boldsymbol{Y})}{P_t(\boldsymbol{Y})} - 1 \right)^2 \mathrm{d}\boldsymbol{Y}$$

数据分布相似时，其概率密度比 $w_t(\boldsymbol{Y}, \boldsymbol{\theta})$ 接近于 1，$PE(P_{t-1} \| P_t)$ 近似为 0。

uLSIF 不仅具有较高的数值稳定性，且计算效率较 KLIEP 明显提高[20]，但 uLSIF 的收敛速度较慢[24]。为此，文献[1]在 uLSIF 基础上进行了改进，提出了相对 uLSIF 法(Relative uLSIF，RuLSIF)。该方法通过引入相对系数 $\alpha(0 \leqslant \alpha < 1)$ 和混合密度 $P_{t,\alpha}(\boldsymbol{Y}) = \alpha P_{t-1}(\boldsymbol{Y}) + (1-\alpha) P_t(\boldsymbol{Y})$，定义了相对 PE 散度。

$$PE_\alpha(P_{t-1} \| P_t) = PE(P_{t-1}(\boldsymbol{Y}) \| P_{t,\alpha}(\boldsymbol{Y}))$$

显然，当 $\alpha = 0$ 时，相对 PE 散度 $PE_\alpha(P_{t-1} \| P_t)$ 即为 PE 散度 $PE(P_{t-1} \| P_t)$。RuLSIF 选用同样的概率密度比模型 $w_t(\boldsymbol{Y}, \boldsymbol{\theta})$，并使用交叉验证法优化式来实现 $w_t(\boldsymbol{Y}, \boldsymbol{\theta})$ 与相对概率密度比 $\frac{P_{t-1}(\boldsymbol{Y})}{P_{t,\alpha}(\boldsymbol{Y})}$ 的逼近。

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \int \left(\frac{P_{t-1}(\boldsymbol{Y})}{P_{t,\alpha}(\boldsymbol{Y})} - w_t(\boldsymbol{Y}, \boldsymbol{\theta}) \right)^2 P_{t,\alpha}(\boldsymbol{Y}) \mathrm{d}\boldsymbol{Y} \tag{3}$$

尽管 RuLSIF 的计算效率较高，且引入 α 提高了收敛速度，但文献[1]中的实验结果显示该方法易受噪声干扰。为增强检测方法的鲁棒性，文献[21]基于分隔距离的异常检测方法(Separation Distance Change Point Detection，SEP)计算子序列间的 SE 距离以度量其相似性。

$$SE(P_{t-1} \| P_t) = \max\left(0, 1 - \frac{P_{t-1}(\boldsymbol{Y})}{P_t(\boldsymbol{Y})} \right)$$

该方法仍采用交叉验证法优化如下函数学习模型 $w_t(\boldsymbol{Y}, \boldsymbol{\theta})$ 中的参数 $\boldsymbol{\theta}$。

$$\min_{\boldsymbol{\theta}} \int \left| \frac{P_{t-1}(\boldsymbol{Y})}{P_t(\boldsymbol{Y})} - w_t(\boldsymbol{Y}, \boldsymbol{\theta}) \right| P_t(\boldsymbol{Y}) \mathrm{d}\boldsymbol{Y}$$

噪声的存在会降低子序列间的相似性，使无异常子序列间存在较小散度或距离，相比 KLIEP，uLSIF 以及 RuLSIF，SEP 通过计算 $\max(\cdot)$ 截去部分距离，增强了检测方法的鲁棒性[21]。尽管如此，SEP 仍需估计概率密度比，该估计过程存在较高的计算复杂度。概率密度比的本质是比较子序列间的分布特征，进而捕捉时序数据异常。除比较分布特征，还可基于子序列波动特征实现相似性度量，序列波动特征描述序列内部数据点梯度绝对值的波动情况。一种常用于刻画序列波动特征的指标是 TV 算子[22]，TV 比可用于度量序列间的波动特征的相似性。鉴于此，本文方法定义了全变分比分隔距离指标，该指标直接计算子序列间的 TV 比，避免了复杂的参数估计过程，从而提高了计算效率。子序列间的波动特征相似性越小，全变分比分隔距离值越大，存在异常的可能性就越大。同时，为进一步增强检测方法对噪声的抗干扰能力，我们采用 RTV 对数据进行平滑处理以增强异常检测方法的鲁棒性，从而进一步提高检测方法的准确度。在仿真数据集和真实数据集上的实验结果表明，本文方法能够快速、低时延且准确地检测异常。

3 本文方法

3.1 全变分比分隔距离

SEP 使用 SE 距离度量数据分布相似性可增强检测方法对噪声的抗干扰能力，降低异常误判率[21]。然而与 KL 散度、PE 散度以及相对 PE 散度类似，计算子序列间的 SE 距离需估计概率密度比，该估计过程复杂度较高。为此，本文使用 TV 比替代 SE 距离中的概率密度比，提出全变分比分隔距离，用于捕捉时序异常。我们将长度为 k 的子序列 $\boldsymbol{Y}_{t-1}$ 与 $\boldsymbol{Y}_t$ 间的全变分比分隔距离记为点 t 的距离得分值，其定义为：

$$SE_{TV}(\boldsymbol{Y}_{t-1} \| \boldsymbol{Y}_t) = \max\left(0, 1 - \frac{TV(\boldsymbol{Y}_{t-1})}{TV(\boldsymbol{Y}_t)} \right) \tag{4}$$

其中，$TV(\boldsymbol{Y}_t)$ 为子序列 $\boldsymbol{Y}_t$ 的全变分，即：

$$TV(\boldsymbol{Y}_t) = \sum_{i=t-k+1}^{t-1} | \boldsymbol{y}_{i+1} - \boldsymbol{y}_i |$$

TV 为序列内数据梯度绝对值的和，用于刻画序列的波动特征。若相邻子序列中均无异常，序列则会表现出相似的波动特征，此时序列间 TV 比近似 1；反之，两者波动特征相差较大，从而使得 TV 比偏离 1。因此 TV 比可比较子序列间的相似性，同时避免了复杂的参数估计过程。

基于上述分析，全变分比分隔距离不仅保留了 SEP 鲁棒性较高的优点，而且可大幅降低计算复杂度。

3.2 结合相对全变分的异常检测

尽管全变分比分隔距离提高了异常检测方法的计算效率，但计算距离得分值易受噪声干扰。为增强检测方法的抗干扰能力，本文采用 RTV 对得分序列进行处理，从而增强了

检测方法的鲁棒性。

RTV 模型最初主要用于去除图像的纹理信息[25]，其本质是含有一个保真项和一个正则项的能量泛函极小化问题。保真项可使平滑的图像保留原图像的主要特征；正则项则保证了解具有一定的光滑性和某些区域的非连续性，同时保证了极小化问题是良态的[26]。受此启发，本文将其引入到时序数据分析领域，以解决时序数据异常检测中的噪声干扰问题。

本文定义 S 为 RTV 模型的输入数据，S_t 表示 S 在 t 时间点的数据值，可利用 RTV 模型对 S 进行平滑处理，得到平滑后的数据 S^{RTV}，如式(5)所示：

$$S^{RTV}=\arg\min_{S^{RTV}}\sum_t(S_t^{RTV}-S_t)^2+\lambda\cdot\frac{D_t}{L_t+\varepsilon} \tag{5}$$

其中，λ 为权重系数；ε 为一个较小的正数，用于防止正则项趋于无穷大；D_t 和 L_t 分别表示以时间点 t 为中心的窗口 $R(t)$ 的全变分和固有变分，其定义分别为：

$$D_t=\sum_{t'\in R(t)}g_{t,t'}\cdot|(dS^{RTV})_{t'}|$$

$$L_t=\left|\sum_{t'\in R(t)}g_{t,t'}\cdot(dS^{RTV})_{t'}\right|$$

其中，$(dS^{RTV})_{t'}$ 表示窗口内数据点的梯度，$g_{t,t'}$ 为权重函数。

$$g_{t,t'}\propto\exp\left(-\frac{(S_t^{RTV}-S_t^{RTV})^2}{2\sigma^2}\right)$$

其中，σ 可控制窗口的大小。

在上述 RTV 模型中，$(S_t^{RTV}-S_t)^2$ 项保证输出和输入的偏差不会太大；而正则项 $\frac{D_t}{L_t+\varepsilon}$ 则抑制数据发生细小波动。然而，由于正则项是非凸的，无法对式(5)进行直接求解，因此可将正则项近似分解为一个非线性项和一个二次项的乘积，即：

$$\begin{aligned}\sum_t\frac{D_t}{L_t+\varepsilon}&=\sum_{t'}\sum_{t\in R(t')}\frac{g_{t,t'}}{\left|\sum_{t'\in R(t)}g_{t,t'}\cdot(dS^{RTV})_{t'}\right|+\varepsilon}|(dS^{RTV})_{t'}|\\&\approx\sum_{t'}\sum_{t\in R(t')}\frac{g_{t,t'}}{L_t+\varepsilon}\frac{1}{|(dS^{RTV})_{t'}|+\varepsilon_{S^{RTV}}}(dS^{RTV})_t^2\\&=\sum_t u_{t'}q_{t'}(dS^{RTV})_t^2\end{aligned}$$

其中，$(dS^{RTV})_t^2$ 与 $u_{t'}q_{t'}$ 分别为近似分解后的二次项部分和非线性部分。

$$\begin{aligned}u_{t'}&=\sum_{t\in R(t')}\frac{g_{t,t'}}{L_t+\varepsilon}=\left(G_\sigma*\frac{1}{|G_\sigma*dS^{RTV}|+\varepsilon}\right)_{t'}\\q_{t'}&=\frac{1}{|(dS^{RTV})_{t'}|+\varepsilon_{S^{RTV}}}\end{aligned} \tag{6}$$

其中，G_σ 是带宽为 σ 的高斯核函数，$*$ 为卷积操作。因此，式(5)可近似表示为如下矩阵形式：

$$\arg\min_{\boldsymbol{V}_{S^{RTV}}}(\boldsymbol{V}_{S^{RTV}}-\boldsymbol{V}_S)^{\mathrm{T}}(\boldsymbol{V}_{S^{RTV}}-\boldsymbol{V}_S)+\lambda(\boldsymbol{V}_{S^{RTV}}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{Q}\boldsymbol{C}\boldsymbol{V}_{S^{RTV}}) \tag{7}$$

其中，$\boldsymbol{V}_{S^{RTV}}$ 和 $\boldsymbol{V}_S$ 分别为 S^{RTV} 和 S 的列向量表示；$\boldsymbol{C}$ 为前向差分梯度算子的托普利兹矩阵；$\boldsymbol{U}$ 和 $\boldsymbol{Q}$ 为对角矩阵，且 $\boldsymbol{U}[i,i]=u_i$，$\boldsymbol{Q}[i,i]=q_i$。基于此，利用欧拉-拉格朗日方程计算式(7)，将式(7)极小化问题转换为式(8)线性方程的迭代求解问题。

$$(\mathbf{I}+\lambda\boldsymbol{L}^p)\cdot\boldsymbol{V}_{S^{RTV}}^{p+1}=\boldsymbol{V}_S \tag{8}$$

其中，$\mathbf{I}$ 为单位矩阵，$\boldsymbol{L}^p=\boldsymbol{C}^{\mathrm{T}}\boldsymbol{U}^p\boldsymbol{Q}^p\boldsymbol{C}$ 和 $\boldsymbol{V}_{S^{RTV}}^{p+1}$ 分别为第 p 次迭代的权矩阵和第 $p+1$ 次迭代 S^{RTV} 的向量表示。

根据上述讨论可知，式(5)的具体求解过程如下：

步骤 1 确定迭代次数 $p_{\max}$，令 $p=1$，$(S^{RTV})^0=S$，并将 $(S^{RTV})^0$ 和 S 向量表示为 $\boldsymbol{V}_{S^{RTV}}^0$ 和 $\boldsymbol{V}_S$；

步骤 2 根据 $p-1$ 次迭代的结果 $\boldsymbol{V}_{S^{RTV}}^{p-1}$，利用式(6)计算 $u_{t'}$ 和 $q_{t'}$，进而组成矩阵 $\boldsymbol{U}^p$ 和 $\boldsymbol{Q}^p$；

步骤 3 求解式(8)，解得 $\boldsymbol{V}_{S^{RTV}}^p$；

步骤 4 重复步骤 2 和步骤 3，至 $p=p_{\max}$；

步骤 5 基于 $\boldsymbol{V}_{S^{RTV}}^p$ 列向量转置输出平滑后的数据 S^{RTV}。

3.3 基于全变分比分隔距离的异常检测算法

由上文可知，本文采用 TV 比度量子序列波动特征的相似性。由于 TV 比不具有对称性，使得全变分比分隔距离不满足对称性的要求，即 $SE_{TV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)\neq SE_{TV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})$。为保证异常得分的对称性，我们同时求解 $SE_{TV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)$ 和 $SE_{TV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})$，在使用 RTV 对距离得分序列进行平滑处理得到 $S^{RTV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)$ 和 $S^{RTV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})$ 后，将异常得分记为：

$$score(\boldsymbol{Y}_{t-1},\boldsymbol{Y}_t)=|S^{RTV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)-S^{RTV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})|$$

通过上述方式实现异常得分的对称性，即 $score(\boldsymbol{Y}_{t-1},\boldsymbol{Y}_t)=score(\boldsymbol{Y}_t,\boldsymbol{Y}_{t-1})$，进而全面捕捉异常。此外，根据实验结果可知，可对均值明显变化的时序数据使用 RTV 进行预处理，以抑制噪声干扰。若 t 点异常引起数据均值变化，则 $\frac{TV(\boldsymbol{Y}_{t-1})}{TV(\boldsymbol{Y}_t)}\ll1$，因此对于此类数据，仅需计算 $SE_{TV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)$ 即可，以降低计算复杂度。尽管此时异常得分不满足对称性质，但仍可全面捕捉异常。

基于上述分析，所提异常检测算法可总结为以下步骤：

步骤 1 数据预处理。给定时间序列 $y=\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_t,\cdots,\boldsymbol{y}_n\}$，随机选取数据片段，求其均值并比较大小，判断随时间展开的序列 y 的均值是否发生明显变化。若变化，则采用式(5)平滑序列 y，以此为基础，选取一个大小为 k 的窗口，通过滑动该窗口得到子序列向量 $\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_t,\cdots,\boldsymbol{Y}_n$，且 $\boldsymbol{Y}_t=[\boldsymbol{y}_{t-k+1}^{\mathrm{T}},\cdots,\boldsymbol{y}_{t-1}^{\mathrm{T}},\boldsymbol{y}_t^{\mathrm{T}}]^{\mathrm{T}}$，如图 1 所示。

步骤 2 计算全变分比分隔距离。由式(4)计算相邻子序列 $\boldsymbol{Y}_{t-1}$ 和 $\boldsymbol{Y}_t$ 间的全变分比分隔距离 $SE_{TV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)$ 及 $SE_{TV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})$。对于均值明显波动的数据，令 $SE_{TV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})=0$。

步骤 3 RTV 平滑。采用式(5)平滑处理距离得分序列 $SE_{TV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)$ 和 $SE_{TV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})$，进而得到 $S^{RTV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)$ 和 $S^{RTV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})$。

步骤 4 计算异常得分。将序列 y 的异常得分记为：$score(\boldsymbol{Y}_{t-1},\boldsymbol{Y}_t)=|S^{RTV}(\boldsymbol{Y}_{t-1}\parallel\boldsymbol{Y}_t)-S^{RTV}(\boldsymbol{Y}_t\parallel\boldsymbol{Y}_{t-1})|$。

4 实验结果与分析

为了客观验证本文方法的检测性能，我们选取了 4 种先进时序数据异常检测方法与本文方法进行对比实验，并针对这 4 种方法在仿真数据集和真实数据集上的实验结果进行对比分析，以评估不同方法的性能。

4.1 评估指标

本文使用受试者工作特征曲线(ROC)及该曲线下的面积 AUC 作为评估指标，该指标被广泛用于评估异常检测方法的性能[27]。通过给检测结果设定阈值，将异常检测问题转化为二分类问题，对于每个阈值均可计算出一组假阳性率(FPR)和真阳性率(TPR)。在此基础上，以 FPR 为横坐标、TPR 为

纵坐标绘制 ROC 曲线。FPR 可反映异常检测方法中会产生的假警报数量，TPR 则反映检测方法发现异常的能力，其定义如下：

$$FPR=\frac{FP}{FP+TN}$$

$$TPR=\frac{TP}{TP+FN}$$

其中，TP 和 FN 分别表示正类实例中被预测为正类和负类的数量，FP 和 TN 分别表示负类被预测成正类和负类的数量。同时，AUC 可作为量化指标，对异常检测方法的性能进行评估，其值越大表明该方法的异常检测的性能越好。

异常检测方法的时延大小也是衡量方法检测性能高低的重要指标，但 AUC 无法反映某方法检测时延的大小。针对该问题，本文提出了一种评估指标 $t_d-AUC_{\max}$ 图。该图的横坐标 t_d 表示方法的检测时延，纵坐标 $AUC_{\max}$ 表示该方法的 AUC 值，每种检测方法对应坐标中的一个点 $(t_d, AUC_{\max})$。该点表示检测方法在时延大小为 t_d 时，能够达到的最高检测性能，此时该方法有最大的 AUC 值，即该方法的时延为 t_d。若某方法的 $(t_d, AUC_{\max})$ 点位于坐标轴的左上角，则说明该方法能够低时延实现较高的异常检测性能；反之，若某方法的 $(t_d, AUC_{\max})$ 点位于坐标轴的右下角，则说明该方法的异常检测率较低，且存在较大时延。

4.2 实验数据集

4.2.1 仿真数据集

数据集 1(D_1)的数据由如下分段函数生成：

$$h(t)=\begin{cases}5+\xi, & 0<t\leqslant 500\\ 0+\xi, & 500<t\leqslant 1\,000\end{cases}$$

其中，ξ 表示均值为 0、标准差为 0.5 的高斯噪声。该数据集在 $t=500$ 处存在一个异常点，且异常前后数据均值发生了明显变化。

数据集 2(D_2)的数据由自回归模型生成，即：

$$\begin{cases}h(1)=h(2)=0,\\ h(t)=0.6h(t-1)-0.5h(t-2)+\xi(t), 3\leqslant t\leqslant 1\,000\end{cases}$$

其中，高斯噪声 $\xi(t)$ 的标准差为 $\sigma(t)=1.5$，且每 100 个数据点噪声均值变化 1 次，即：

$$\mu(t)=\left\lfloor\frac{t-1}{100}\right\rfloor\cdot 5$$

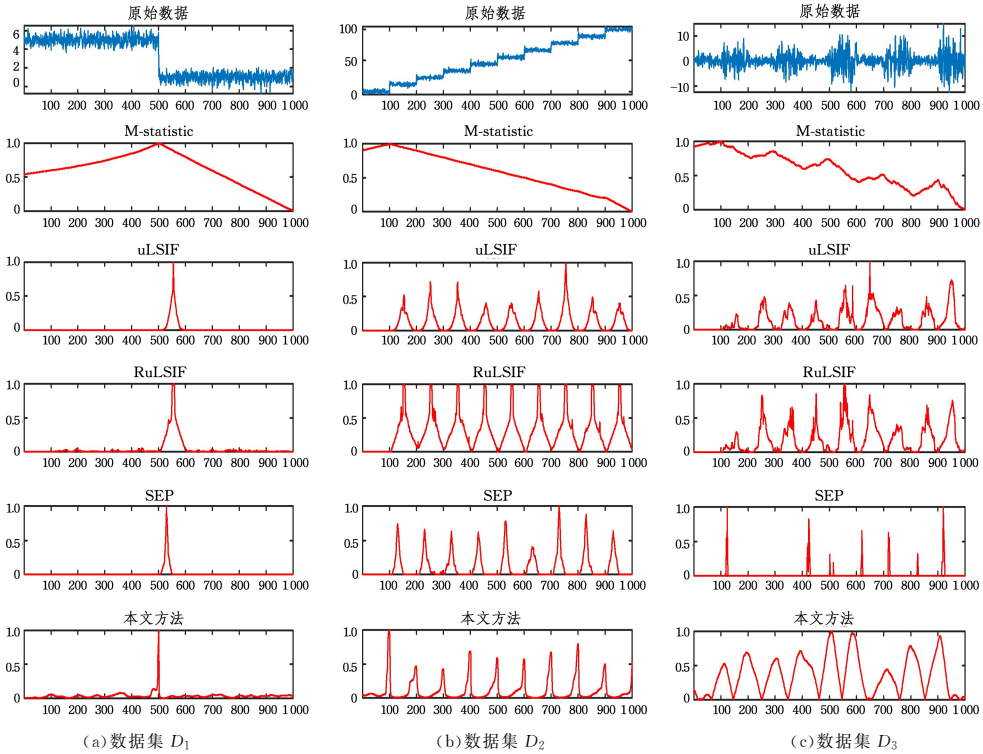
该数据集在 $t=100,200,\cdots,900$ 处存在 9 个异常，且数据均值存在明显波动。

数据集 3(D_3)采用与 D_2 相同的模型 $h(t)$ 生成数据。不同之处是，该数据集设置噪声均值 $\mu(t)=0$，标准差 $\xi(t)$ 随时间变化，即：

$$\xi(t)=\begin{cases}1, & \left\lfloor\frac{t-1}{100}\right\rfloor=0,2,4,6,8\\ \ln\left(e+\left(\left\lfloor\frac{t-1}{100}\right\rfloor+1\right)\cdot 5\right), & \left\lfloor\frac{t-1}{100}\right\rfloor=1,3,5,7,9\end{cases}$$

每隔 100 个数据点，噪声标准差改变一次，生成一异常点，共生成 9 个异常点。

图 2(a)－图 2(c)的第一行子图分别显示了以上 3 个仿真数据集的原始数据。



(a)数据集 D_1 (b)数据集 D_2 (c)数据集 D_3

图 2 不同方法在数据集 D_1，D_2，D_3 中的实验结果

Fig. 2 Experimental results of different methods in D_1, D_2 and D_3

1) http://hasc.jp/hc2011/

2) http://iops.ai/dataset_detail/?id=10

4.2.2 真实数据集

为了评估不同方法在真实数据中检测异常的能力，我们还在人类活动数据集和 KPI 数据集上对不同方法进行了对比分析。

(1)人类活动数据集[1)]。该数据集源于 HASC 挑战赛中由传感器收集的人类活动信号，其主要描述 6 种人类行为活动[1]。行为活动的转换引起信号中数据分布的改变，从而引发异常，因此数据异常位于人类改变行为活动的时间点。通过检测该数据集中的异常，在时序数据中分割出这 6 种行为的信号，以识别和理解人类活动。

(2)KPI 数据集[2)]。该数据集含有多条 KPI 曲线，这些曲线源于多个互联网公司，主要记录某产品或某应用的关键性能指标，并且该数据中记录了异常发生的时间点。运维人员通过监控 KPI 曲线是否存在异常来判断 Web 服务器是否稳定。

4.3 实验结果

本文在上述 3 个仿真数据集和两个真实数据集上将 M-statistic[23]，uLSIF[20]，RuLSIF[1]，SEP[21] 与本文方法进行对比分析。使用 MATLAB 语言实现了本文方法和 SEP，而 M-statistic，uLSIF 和 RuLSIF 则采用文献[23]和文献[1]给出的源代码。通过分析窗口大小 k 对本文方法异常检测效率的影响，选取合适的参数 k。同时，为了更直观地对实验结果进行分析，将异常得分映射到区间[0,1]。

4.3.1 仿真数据集实验结果

数据集 D_1 仅含一个异常，该异常位于 $t=500$ 处，异常的存在导致数据均值在此处明显下降，如图 2(a)第一行子图所示。由于数据中的噪声会对检测异常产生干扰，我们先使用 RTV 对数据进行平滑处理。图 2(a)给出了 M-statistic，uLSIF，RuLSIF，SEP 以及本文方法的异常检测结果。可以看出，以上 5 种方法的实验结果在区间[500,550]内出现了明显峰值，这说明这些方法能检测出该异常。M-statistic 和本文方法的结果中峰值位于 $t=500$ 处，说明这两种方法能够无时延地捕捉此异常；而其他方法的峰值出现在区间[510,600]内，说明这些方法对异常的检测存在明显时延。

图 2(b)的第一行子图给出了数据集 D_2 的原始数据，该数据在整百处存在异常点，且数据均值在异常处存在明显变化。与数据集 D_1 类似，使用 RTV 对数据进行处理以抑制噪声干扰，图 2(b)给出了不同方法的检测结果。从图 2(b)中可以看出，M-statistic 仅在 $t=100$ 处得到较大峰值，说明该方法能够捕捉此异常，而未捕捉到其他 8 个异常；其他方法均得到了多个较大峰值。其中，本文方法可无时延地检测出该数据集中的多个异常，其峰值均位于异常发生处(整百处)；而 uLSIF，RuLSIF 和 SEP 尽管检测到了多个异常，但存在一定的检测时延。同时，我们发现 RuLSIF 在多个无异常区间(如区间[660,700])内明显将噪声误判为异常。

数据集 D_3 中的异常属于现实场景中最常见的一种类型，此类异常表现为数据方差的改变。该数据集与数据集 D_1 和 D_2 不同，不存在明显的均值波动。图 2(c)分别给出了数据集 D_3 的原始数据以及 M-statistic，uLSIF，RuLSIF，SEP 和本文方法的异常检测结果。从图 2(c)中可以看出，本文方法在异常位置(整百处)有较大峰值，可较准确地捕捉全部异常；而其他方法仅在部分异常位置后出现较大峰值，意味着这些方法仅能检测出部分异常，存在漏检问题，且有明显时延。例如，SEP 的结果含有 7 处较大峰值，说明其仅检测到了 7 个异常，而未捕捉到其他 2 个异常。

图 3 给出了不同方法在 3 个仿真数据集上的 ROC 曲线。从图 3 中可以看出，在数据集 D_1 上，方法 M-statistic 和本文方法具有较高的 TPR 和较低的 FPR，其他方法则表现出了相对较低的 TPR，这说明对于含有单异常的数据集 D_1，M-statistic 和本文方法有较好的检测性能。在数据集 D_2 和 D_3 上，本文方法的 TPR 和 FPR 均优于其他方法，本文方法能够准确识别多个异常。

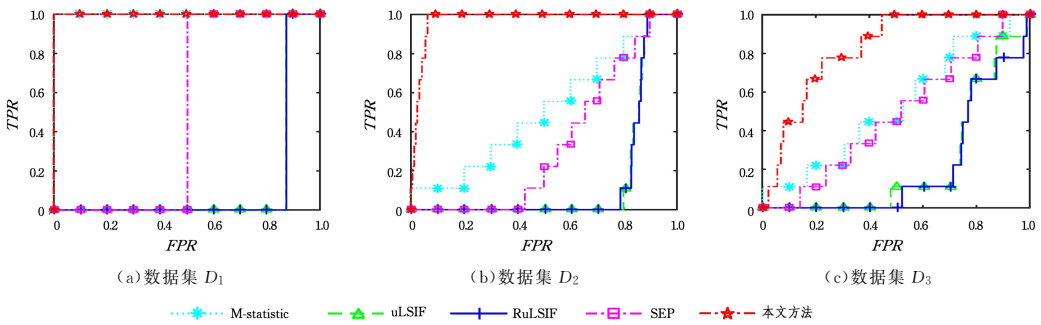


图 3 不同方法在数据集 D_1，D_2，D_3 中的 *ROC* 曲线图

Fig. 3 ROC curves of different methods in D_1, D_2 and D_3

此外，表 1 列出了不同方法的 *AUC* 值，最大的 *AUC* 值用粗体标出。由表 1 可知，本文方法的 *AUC* 值明显高于其他方法，其在数据集 D_1 上的 *AUC* 高达 1，在数据集 D_1 和 D_2 上的 *AUC* 分别为 0.9713 和 0.8236，这说明在这些方法中，本文方法的异常检测量化性能最佳。此外，M-statistic 在仅含有一个异常的数据集 D_1 上表现出了最高的检测性能，其 *AUC* 为 1，但在其他含多异常的数据集上分别为 0.5119 和 0.5263，这说明 M-statistic 更适用于检测单异常。

表 1 不同方法在数据集 D_1，D_2，D_3 上的 AUC 值

Table 1 AUC values of different methods in D_1, D_2 and D_3

DataSet	M-statistic	uLSIF	RuLSIF	SEP	本文方法
D_1	1.0000	0.1261	0.1261	0.4965	**1.0000**
D_2	0.5119	0.1502	0.1511	0.3377	**0.9713**
D_3	0.5263	0.2263	0.2092	0.4809	**0.8236**

为分析不同方法的时延大小，我们分别选取不同时延 $t=0,10,20,30,40,50$，绘制不同方法的 ROC 图(见图 4)并计算 *AUC* 值(见表 2)。由图 4 可知，本文方法在 $t=0$ 时，达到了

最高的 TPR；M-statistic 的 TPR 和 FPR 受时延影响不明显；而其他方法在不同时延下表现出了明显不同的 TPR 和 FPR。这意味着本文方法在仿真数据集上可近似无时延，实现较高的 TPR；M-statistic 性能受时延影响较小；而其他方法存在明显的检测时延。表 2 中用粗体标出了最大 AUC 值，M-statistic 和本文方法在 $t=0$ 时有最高的 AUC 值，说明这两种方法不存在检测时延；而其他方法在多数仿真数据集上，分别在 $t=50$，50 和 30 处有最高的 AUC 值，说明这些方法对这 3 个数据集的检测时延大致分别为 $t=50$，50，30。此外，本文方法在数据集 D_1 上有最高的 AUC 值，在数据集 D_2 和 D_3 上均可获得较高的 AUC 值，实现了较好的异常检测。

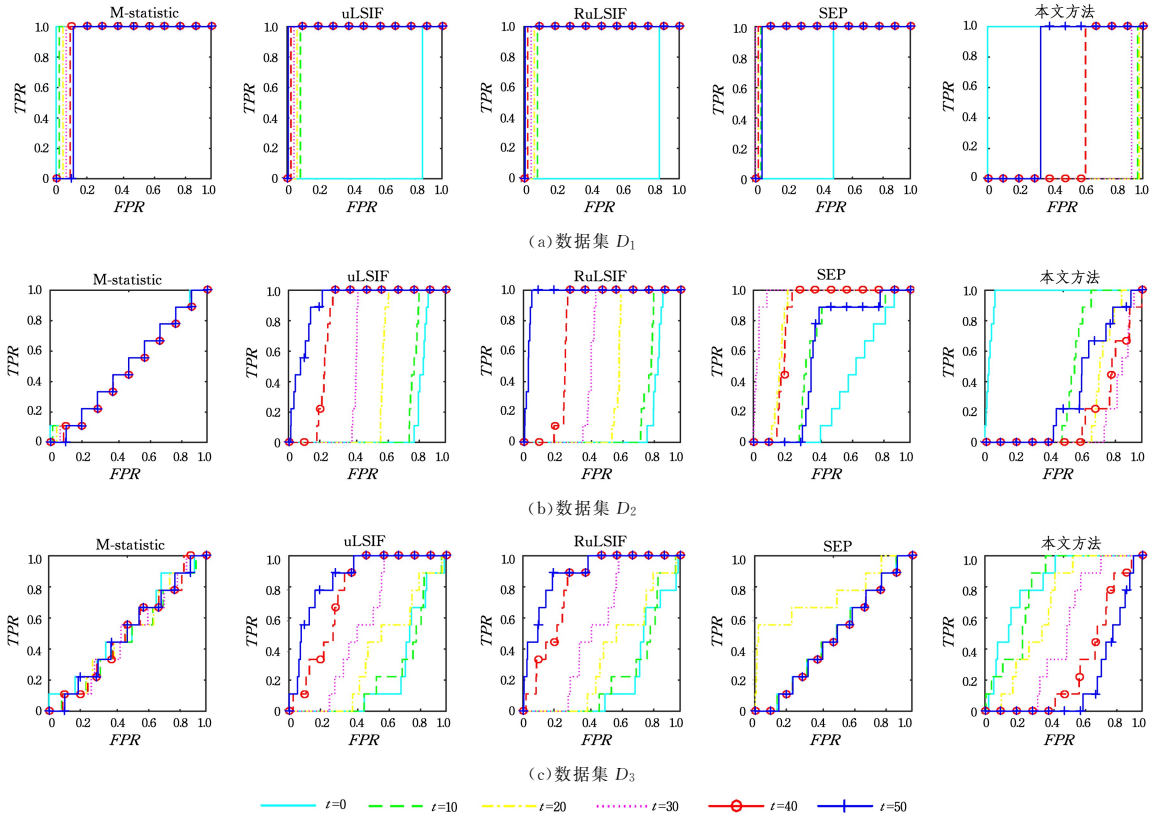


图 4 不同方法在数据集 D_1，D_2，D_3 中不同时延下的 ROC 曲线图

Fig. 4 ROC curves of different methods with different time delays in D_1，D_2 and D_3

表 2 不同方法在数据集 D_1，D_2，D_3 中不同时延下的 AUC 值

Table 2 AUC values of different methods with different time delays in D_1，D_2 and D_3

Dataset	Methods	M-statistic	uLSIF	RuLSIF	SEP	本文方法
	Time delay	AUC	AUC	AUC	AUC	AUC
D_1	0	**1.0000**	0.1261	0.1261	0.4965	**1.0000**
	10	0.9790	0.9139	0.9129	0.9630	0.0340
	20	0.9560	0.9349	0.9339	0.9800	0.0230
	30	0.9349	0.9550	0.9540	**0.9980**	0.0731
	40	0.9099	0.9750	0.9760	0.9810	0.3694
	50	0.8899	**0.9930**	**0.9920**	0.9560	0.6587
D_2	0	**0.5119**	0.1502	0.1511	0.3377	**0.9713**
	10	0.5089	0.2011	0.2025	0.5977	0.4280
	20	0.5061	0.3976	0.3976	0.8309	0.2451
	30	0.5036	0.5730	0.5715	**0.9717**	0.1348
	40	0.5012	0.7704	0.7609	0.8088	0.1843
	50	0.4995	**0.9145**	**0.9704**	0.5845	0.3435
D_3	0	**0.5263**	0.2263	0.2092	0.4809	**0.8236**
	10	0.4773	0.2352	0.2333	0.4807	0.7938
	20	0.4930	0.3564	0.3515	**0.7416**	0.6724
	30	0.4850	0.5531	0.5474	0.4751	0.4914
	40	0.4975	0.7622	0.8036	0.4751	0.2927
	50	0.5016	**0.8602**	**0.8871**	0.4751	0.1913

图 5 给出了不同方法在数据集 D_1，D_2 和 D_3 上的 t_d-AUC_{max} 图。

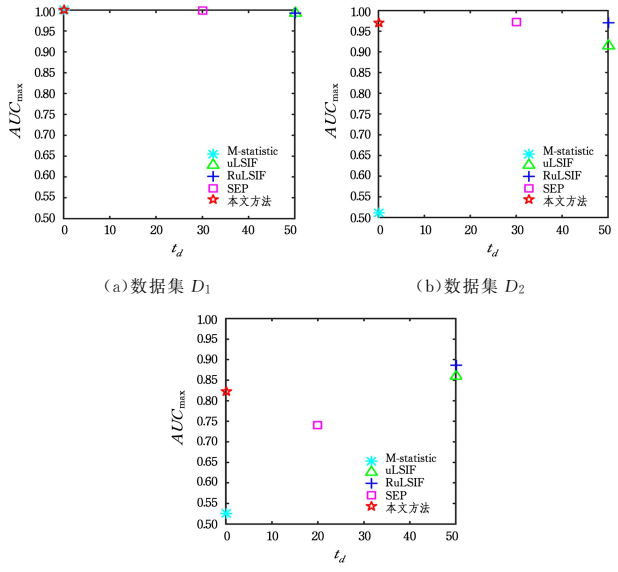


图 5 不同方法在数据集 D_1，D_2，D_3 中的 t_d-AUC_{max} 图

Fig. 5 t_d-AUC_{max} graph of different methods in D_1，D_2 and D_3

从图 5 中可以看出，M-statistic 和本文方法在 3 个数据集上的 t_d 均为 0，说明这两种方法均可低时延捕捉异常；

M-statistic 的 AUC_{max} 除了在含单异常的数据集 D_1 上与本文方法相同，在含多异常的数据集上明显低于本文方法；其他方法可获得较高的 AUC_{max} 值，同时也具有较大的 t_d，这说明这些方法有较好的检测性能，但时延较大。此外，我们观察到，本文方法位于坐标轴的左上角，而 uLSIF，RuLSIF 和 SEP 位于 3 个坐标轴的右上角，M-statistic 在数据集 D_2 和 D_3 上位于坐标左下角。这意味着本文方法捕捉异常的整体性能明显优于其他方法，而 M-statistic 更适用于捕捉单异常，uLSIF，RuLSIF 和 SEP 能够在一定时延下捕捉多异常。

表 3 列出了不同方法在仿真数据集 D_1，D_2，D_3 上的平均检测时间。我们可以看到，uLSIF 的异常检测时间最长，为 43.174 3 s；M-statistic，RuLSIF 和 SEP 的检测时间分别为 28.233 3 s，42.930 3 s 和 22.965 3 s；本文方法的检测时间最短，为 0.262 7 s。说明在上述方法中，RuLSIF 的计算效率最低，本文提出的检测方法的计算效率明显优于其他方法。

表 3 不同方法在数据集 D_1，D_2，D_3 中的平均检测时间

Table 3 Mean detection time of different methods in D_1，D_2 and D_3

Methods	M-statistic	uLSIF	RuLSIF	SEP	本文方法
Time/s	28.233 3	43.174 3	42.930 3	22.965 3	**0.262 7**

为了研究本文方法在计算全变分比分隔距离过程中滑动窗口大小与异常检测性能高低之间的关系，图 6 给出了窗口大小分别为 $k=10,30,50,70,90$ 时，本文方法在数据集 D_1，D_2 和 D_3 上 AUC 值的变化情况。可观察到，对于 D_1 和 D_2 这类异常前后数据均值发生明显波动的数据集，窗口大小的选择对本文方法 AUC 值的影响较小，说明对于此类数据，该方法的异常检测性能对窗口大小较为鲁棒；在数据集 D_3 上，本文方法的性能受窗口大小的选择影响较大，当 $k=30$ 时，本文方法的 AUC 值最大，检测性能最佳。基于此，在本文方法中选取窗口 $k=30$ 进行实验。此外，由于本文方法主要用于离线异常检测，因此尽管本文方法采用 RTV 平滑数据的过程依赖于时间点 t 之后的数据，但是我们将该模型中的窗口大小选择为 6，其值小于 $k=30$，从而使得这种选择方法是合理的。

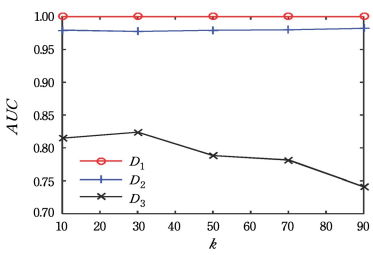


图 6 本文方法中 k 对 AUC 的影响

Fig. 6 Influence of k on AUC with the proposed method

4.3.2 真实数据集实验结果

为了进一步验证不同检测方法的检测准确度和时延大小，我们选用可捕捉多异常的 uLSIF，RuLSIF，SEP 以及本文方法对两个真实数据集中的异常进行检测，并对不同方法的性能进行对比。

选取人类活动数据集中编号为 1001 至 1010 的子集，并在这 10 个子集上对不同方法的整体性能进行对比分析。表 4 列出了 uLSIF，RuLSIF，SEP 以及本文方法在该数据集上的 t_d 和 AUC_{max} 值及其均值。从中可以看出，本文方法存在较小检测时延，其 t_d 均值为 6，uLSIF，RuLSIF 和 SEP 的 t_d 均值分别为 51，53 和 30，这意味着本文方法可以低时延捕捉异常，其他方法均存在相对较大的检测时延。此外，我们观察到本文方法在 5 个子集上获得了最高的 AUC_{max} 值，在 1010 子集上，AUC_{max} 达到最高，为 0.890 1，且本文方法的 AUC_{max} 均值为 0.748 5，明显高于其他异常检测方法。

图 7 给出了不同方法在人类活动数据集上的 t_d-AUC_{max} 图。可以看出，本文方法的 t_d 最小，AUC_{max} 最大，说明本文方法可低时延实现最佳的异常检测性能；而其他方法均位于本文方法的右下方，意味着这些方法存在相对较大的时延，且最高检测性能均低于本文方法。

表 4 不同方法在人类活动数据集中的时延和 AUC 值

Table 4 Delay time and AUC values of different methods in human activity dataset

Number	uLSIF		RuLSIF		SEP		本文方法	
	t_d	AUC_{max}	t_d	AUC_{max}	t_d	AUC_{max}	t_d	AUC_{max}
1001	50	0.702 9	40	**0.706 7**	30	0.576 3	**10**	0.660 1
1002	60	0.663 9	60	0.677 3	0	0.405 6	**10**	**0.704 0**
1003	50	0.701 8	60	0.712 2	50	0.537 6	**0**	**0.868 2**
1004	50	0.715 0	40	**0.719 2**	30	0.638 7	**20**	0.672 2
1005	50	**0.751 3**	50	0.742 1	20	0.576 9	**0**	0.682 2
1006	40	0.685 4	60	0.694 2	40	0.602 1	**0**	**0.778 9**
1007	60	0.720 5	60	0.717 6	40	0.572 1	**0**	**0.788 9**
1008	50	0.761 6	60	**0.777 0**	40	0.537 8	**10**	0.698 0
1009	50	0.730 7	50	**0.735 5**	30	0.476 2	**10**	0.666 5
1010	50	0.723 3	50	0.725 3	20	0.439 0	**0**	**0.890 1**
Average	51	0.715 6	53	0.720 7	30	0.536 2	6	0.748 5

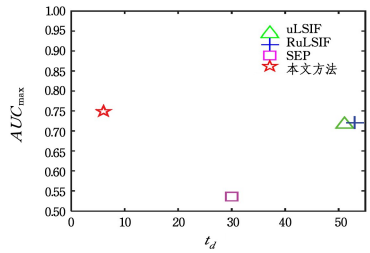


图 7 不同方法在人类活动数据集中的 t_d-AUC_{max} 图

Fig. 7 t_d-AUC_{max} graph of different methods in human activity dataset

选取 KPI 数据集中 4 条 KPI 曲线进行异常检测，将其编号为 01，02，03，04。分别使用 uLSIF，RuLSIF，SEP 以及本文方法对其进行异常检测。表 5 列出了不同检测方法在 KPI 数据集上的 t_d 和 AUC_{max} 值。我们观察到，在上述 4 种方法中，本文方法有最低的 t_d（均值为 0），而 RuLSIF 方法有最大的 t_d（均值为 53）。此外，本文方法在 02 和 04 号子集上表现出了最好的检测性能，在 KPI 数据集上取得了最高的 AUC_{max} 均值，为 0.797 7，而其他方法的均值分别为 0.656 7，0.611 1 和 0.757 4，检测性能明显低于本文方法。

表 5 不同方法在 KPI 数据集中的时延和 AUC 值

Table 5 Delay time and AUC values of different methods in KPI dataset

Number	uLSIF		RuLSIF		SEP		本文方法	
	t_d	AUC_{max}	t_d	AUC_{max}	t_d	AUC_{max}	t_d	AUC_{max}
01	60	**0.7274**	60	0.6267	40	0.7062	0	0.6503
02	60	0.5550	70	0.6251	50	0.6333	**0**	**0.8500**
03	60	0.7088	60	0.6232	50	**0.8695**	0	0.8644
04	20	0.6356	20	0.5692	40	0.8207	**0**	**0.8262**
Average	50	0.6567	53	0.6111	45	0.7574	**0**	**0.7977**

图 8 给出了不同方法在该数据集上的 t_d-AUC_{max} 图。可观察到，本文方法位于坐标轴的左上角，而其他方法位于坐标右侧。这说明与其他方法相比，本文方法能够低时延较准确地捕捉多异常。

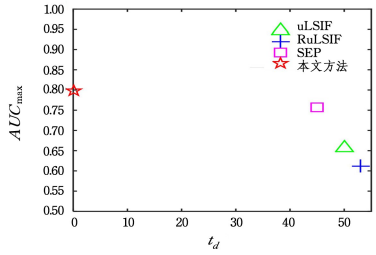


图 8 不同方法在 KPI 数据集中的 t_d-AUC_{max} 图

Fig. 8 t_d-AUC_{max} graph of different methods in KPI dataset

为了进一步分析本文方法与 uLSIF，RuLSIF，SEP 方法时间复杂度的高低，表 6 列出了不同方法在人类活动数据集和 KPI 数据集中的平均检测时间，最短检测时间用粗体标出。可以观察到，RuLSIF 方法在两个真实数据集中的检测时间最长，分别为 681.7530 s 和 484.6915 s，此方法的时间复杂度最高。本文方法的检测时间分别为 0.9323 s 和 0.8108 s，均明显低于其他 3 种异常检测方法，说明相比 uLSIF，RuLSIF 和 SEP，本文方法大幅度降低了检测异常的时间复杂度，异常检测效率明显提高。

表 6 不同方法在真实数据集中的平均检测时间

Table 6 Mean detection time of different methods in real-world dataset

（单位：s）

Dataset	uLSIF	RuLSIF	SEP	本文方法
Human Activity	566.0998	681.7530	273.1314	**0.9323**
KPI	497.6235	484.6915	214.6723	**0.8108**

结束语 针对基于概率密度比检测方法计算效率低以及时延较大的问题，本文提出了一种基于全变分比分隔距离的异常检测方法，该方法度量序列间的相似性以捕捉序列异常。本文中，全变分比分隔距离采用全变分提取序列波动特征，故使用全变分比代替概率密度比来度量序列间的相似性，从而有效避免概率密度比复杂的参数估计，提高了检测方法的计算效率，降低了检测时延。为抑制噪声干扰，本文采用 RTV 平滑来增强检测方法的鲁棒性，以进一步提高本文方法的检测性能。实验结果表明，本文方法具有较好的异常检测性能，可低时延、准确地检测异常。同时，仿真数据集和真实数据集上异常检测时间的实验结果均验证了本文方法具有较高的计算效率。

尽管本文方法可低时延、快速地检测异常，但是随着时序数据维度的增加，数据分布愈加复杂，本文方法将难以精确描述高维特征，从而导致该方法无法准确捕捉异常。如何准确检测高维数据异常将是我们的后续研究内容。

参 考 文 献

[1] LIU S, YAMADA M, COLLIER N, et al. Change-point detection in time-series data by relative density-ratio estimation[J]. Neural Networks, 2013, 43(1): 72-83.

[2] SIFFER A, FOUQUE P A, TERMIER A, et al. Anomaly detection in streams with extreme value theory[C]// Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. USA: ACM, 2017: 1067-1075.

[3] ZHANG Q, HU Y P, JI C, et al. Edge computing application: real-time anomaly detection algorithm for sensing data[J]. Journal of Computer Research and Development, 2018, 55(3): 524-536.

[4] GUO Y S, LIU M D. Anomaly detection based on spatial-temporal trajectory data[J]. Computer Science, 2021, 48(6A): 213-219.

[5] CAI R C, XIE W H, HAO Z F, et al. Abnormal crowd detection based on multi-scale recurrent neural network[J]. Journal of Software, 2015, 26(11): 2884-2896.

[6] CHANDOLA V, BANERJEE A, KUMAR V. Anomaly detection: a survey[J]. ACM Computing Surveys, 2009, 41(3): 1-58.

[7] HODGE V, AUSTIN J. A survey of outlier detection methodologies[J]. Artificial Intelligence Review, 2004, 22(2): 85-126.

[8] SU W X, ZHU Y L, LIU F, et al. Outliers and change-points detection algorithm for time series[J]. Journal of Computer Research and Development, 2014, 51(4): 781-788.

[9] STEINWART I, HUSH D, SCOVEL C. A classification framework for anomaly detection[J]. Journal of Machine Learning Research, 2005, 6(1): 211-232.

[10] ESKIN E, ARNOLD A, PRERAU M, et al. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data[M]// Applications of Data Mining in Computer Security. Boston: Kluwer Academic Publishers, 2002: 78-99.

[11] HAN D M, GUO F Z, PAN J C, et al. Visual analysis for anomaly detection in time-Series: a survey[J]. Journal of Computer Research and Development, 2018, 55(9): 1843-1852.

[12] MOSKVINA V, ZHIGLJAVSKY A. An algorithm based on singular spectrum analysis for change-point detection[J]. Communications in Statistics-Simulation and Computation, 2003, 32(2): 319-352.

[13] CHEN J, OUYANG J Y, FENG A Q, et al. DoS anomaly detection based on isolation forest algorithm under edge computing framework[J]. Computer Science, 2020, 47(2): 287-293.

[14] KEOGH E, CHU S, HART D, et al. An online algorithm for segmenting time series [C]// Proceedings 2001 IEEE International Conference on Data Mining. USA: IEEE, 2001: 289-296.

[15] DING X O, YU S J, WANG M X, et al. Anomaly detection on industrial time series based on correlation analysis[J]. Journal of Software, 2020, 31(3): 726-747.

[16] FENG A R, WANG X R, WANG Q Y, et al. Database anomaly access detection based on principal component analysis and random tree[J]. Computer Science, 2020, 47(9): 94-98.

[17] BORGWARDT K M, GRETTON A, RASCH M J, et al. Integrating structured biological data by kernel maximum mean discrepancy[J]. Bioinformatics, 2006, 22(14): e49-e57.

[18] SUGIYAMA M, SUZUKI T, NAKAJIMA S, et al. Direct importance estimation for covariate shift adaptation[J]. Annals of the Institute of Statistical Mathematics, 2008, 60(4): 699-746.

[19] JIANG H, ZHANG H F, LUO Y D, at al. Adaptive threshold network traffic anomaly detection based on KL distance[J]. Computer Engineering, 2019, 45(4): 108-113.

[20] KANAMORI T, HIDO S, SUGIYAMA M. A least-squares approach to direct importance estimation[J]. The Journal of Machine Learning Research, 2009, 10(1): 1391-1445.

[21] AMINIKHANGHAHI S, WANG T, COOK D J. Real-time change point detection with application to smart home time series data[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(5): 1010-1023.

[22] GIBBS A L, SU F E. On choosing and bounding probability metrics[J]. International Statistical Review, 2002, 70(3): 419-435.

[23] LI S, XIE Y, DAI H, et al. M-statistic for kernel change-point detection[C]// Proceedings of the 28th International Conference on Neural Information Processing Systems. USA: MIT Press, 2015: 3366-3374.

[24] KANAMORI T, SUZUKI T, SUGIYAMA M. Computational complexity of kernel-based density-ratio estimation: A condition number analysis[J]. Machine Learning, 2013, 90(3): 431-460.

[25] XU L, YAN Q, XIA Y, et al. Structure extraction from texture via relative total variation[J]. ACM Transactions on Graphics, 2012, 31(6): 1-10.

[26] DING Z P, ZHANG S W, CHEN J Z, et al. Structure-preserving image smoothing with L0 gradient minimization coupling gradient fidelity term[J]. Scientia Sinica Informationis, 2014, 44(11): 1370-1384.

[27] WANG Y Y, CHEN S C. A survey of evaluation and design for AUC based classifier[J]. Pattern Recognition and Artificial Intelligence, 2011, 24(1): 64-71.

XU Tian-hui, born in 1998, postgraduate. Her main research interests include data analysis and anomaly detection.

GUO Qiang, born in 1979, Ph.D, professor, is a member of China Computer Federation. His main research interests include computer vision and data mining.

(责任编辑:喻藜)