Journal of Computer Applications
计算机应用, 2020, 40(5): 1329 - 1334
ISSN 1001-9081
CODEN JYIIDU
2020-05-10
http://www.joca.cn

文章编号:1001-9081(2020)05-1329-06
DOI:10.11772/j.issn.1001-9081.2019091631

基于自编码器和隐马尔可夫模型的时间序列异常检测方法

霍纬纲*,王慧芳
(中国民航大学 计算机科学与技术学院,天津 300300)
(*通信作者电子邮箱 wghuo@cauc.edu.cn)

摘　要:针对已有基于隐马尔可夫模型(HMM)的时间序列异常检测模型的符号化方法不能很好地表征原始时间序列的问题,提出了一种基于自编码器和HMM的时间序列异常检测方法(AHMM-AD)。首先,通过滑动窗口对时间序列样本进行分段,按照分段位置形成若干时间序列分段样本集,由正常时间序列上不同位置的分段样本集训练各个分段的自编码器;然后,利用自编码器得到每个分段时间序列样本的低维特征表示,通过对低维特征表示向量集的*K*-means聚类处理,实现时间序列样本集的符号化;最后,由正常时间序列的符号序列集生成HMM,根据待测样本在已建HMM上的输出概率值进行异常检测。在多个公共基准数据集上的实验结果显示,AHMM-AD比已有的基于HMM的时间序列异常检测模型在精确度、召回率和F1值分别平均提高了0.172、0.477、0.313,比基于autoencoder的时间序列异常检测模型,在这三方面分别平均提高了0.108、0.450、0.319。实验结果表明,AHMM-AD方法能够提取时间序列中的非线性特征,解决已有HMM建模时间序列符号化过程中不能很好表征时间序列的问题,并在时间序列异常检测性能上也有显著提升。

关键词:自编码器;符号化序列;隐马尔可夫模型;异常检测;时间序列
中图分类号:TP391.4　　**文献标志码**:A

Time series anomaly detection method based on autoencoder and HMM

HUO Weigang*, WANG Huifang
(*School of Computer Science and Technology, Civil Aviation University of China, Tianjin* 300300, *China*)

Abstract: To solve the issue that the existing symbolic methods of anomaly detection based on Hidden Markov Model (HMM) cannot well represent the original time series, an Autoencoder and HMM-based Anomaly Detection (AHMM-AD) method was proposed. Firstly, the time series samples were segmented by sliding window, and several time series segmented sample sets were formed according to the positions of the segmentations, and the autoencoder of each segmentation was trained by the segmented sample set of different positions on the normal time series. Then, the low-dimensional feature representation of each segmented time series sample was obtained by using the autoencoder, and through *K*-means clustering of low-dimensional feature representation vector sets, the symbolization of time series sample sets was realized. Finally, the HMM was generated based on the symbol sequence set of the normal time series, and the abnormal detection was carried out according to the output probability values of the test samples on the established HMM. The experimental results on multiple common benchmark datasets show that AHMM-AD improves the accuracy, recall rate, and F1 value by 0.172, 0.477 and 0.313 respectively compared to those of the HMM-based time series anomaly detection model, and has 0.108, 0.450 and 0.319 increasement in these three aspects respectively compared with the autoencoder-based time series anomaly detection model. The experimental results illustrate that AHMM-AD method can extract the nonlinear features in time series, solve the problem that the time series cannot be well represented during the symbolization process of existing HMM-based time series modeling, and also improve the performance of time series anomaly detection.

Key words: autoencoder; symbol sequence; Hidden Markov Model (HMM); anomaly detection; time series

0　引言

时间序列异常检测是时序数据挖掘的重要研究问题之一,已被广泛地应用于航天[1]、金融[2]、医疗[3]等领域。隐马尔可夫模型(Hidden Markov Model, HMM)作为一种动态时间序列统计分析模型,最早在20世纪60年代末引入和研究。由于其强大的数学基础理论和完善的算法,HMM成为语音识别[4]、经济分析、机械工程等领域的主流技术。Rabiner[4]验证了HMM对具有时序性信息的识别能力较强。近年来,HMM方法已被成功引入到时间序列的故障诊断[5]和异常检测领域[1-3,6]。

基于HMM的时间序列的异常检测方法一般主要包含两个重要步骤:1)符号化可观测时间序列;2)参数学习与概率估

收稿日期:2019-09-24;**修回日期**:2019-10-19;**录用日期**:2019-10-21。　**基金项目**:国家自然科学基金委员会-中国民用航空局民航联合研究基金资助项目(U1633110);中央高校基本科研业务费项目中国民航大学专项(3122019190)。

作者简介:霍纬纲(1978—),男,山西洪洞人,副教授,博士,CCF会员,主要研究方向:数据挖掘、模糊分类;　王慧芳(1993—),女,山西大同人,硕士研究生,主要研究方向:数据挖掘。

计。符号化时间序列作为时间序列表示方法之一，旨在以字符串序列表示原始时间序列。符号化时间序列不但可以达到以低维数据表示高维数据的目的，而且符合HMM算法对观测序列的要求。目前符号化时间序列的方法主要有分段符号化线性表示法[7]和聚类符号化表示方法[8-10]。分段符号化线性表示方法是将原始时间序列分段，并以时域和频域信息表示为符号序列。文献[7]将HMM应用在心电图(ElectroCardioGram，ECG)医疗数据的阻塞性睡眠呼吸暂停(Obstructive Sleep Apnea，OSA)病情检测中，该方法选取均值、方差和偏度等信息构成符号化序列，但是在转化的过程中需要大量的先验知识和丰富的实验经验，这不满足现实对于时间序列异常检测的需要。聚类符号化表示方法采用聚类的方法将时间序列的连续实数值映射到有限的符号表上，使得时间序列转化为有限符号的有序集合[11]。文献[8]将HMM应用在飞机着陆操作数据的异常检测中，采用*K*-means聚类算法将原始时间序列转化成由*K*个簇标记表示的符号序列；文献[9]将聚类HMM应用在快速存取记录器(Quick Access Recorder，QAR)数据分析的研究中，采用二次广度优先邻居搜索聚类算法将原始序列转化成由聚类的符号表示的符号序列；文献[10]将HMM应用在多维时间序列上的异常检测中，分别采用模糊C均值(Fuzzy C-Means，FCM)聚类和模糊积分技术将多维时间序列转换成单维的符号序列，且提高了HMM的异常检测能力。

符号化序列分析的结果主要由符号化算法和时间序列的相空间轨迹决定。值得注意的是，如果使用原始时间序列中的所有数据进行分析，在符号化过程之前不进行特征提取，符号序列可能过长，噪声的干扰对计算效率会造成影响[12]。上述文献[7,8-10]在符号化步骤之前均无特征提取过程。文献[12]结合感知重要点(Perceptually Important Point，PIP)[13]和HMM实现对液压泵的故障诊断，该方法寻找对原始时间序列整体运动形状影响较大的数据点，减少了时间序列压缩过程中的信息损失，再根据重要点在符号空间位置划分区域实现符号化。文献[14]采用奇异值分解代替主成分分析中的特征分解对含有高维度的流量数据包进行降维，再应用*K*-means方法为HMM创建有意义的观测序列。以上两种方法都是在符号化序列之前，对分段后的原始序列进行特征提取和数据降维，并在故障分类和入侵检测的准确度中取得了较好的效果。但是基于感知重要点技术的符号化方法需要计算每个点对于时间序列的影响力，复杂的计算过程降低了异常检测的效率，而基于改进主成分分析的符号化方法在特征提取时会将数据方差较小的数据信息忽略，使得模型对非线性特征提取效果较差。

自编码网络是Hinton在2006年提出的一种由编码器和解码器构成的无监督学习算法，旨在从大量无标记的数据中学习数据的有效信息，并实现对输入数据的非线性压缩和重构[15]。为克服目前已有HMM建模时间序列符号化过程中特征提取方法的不足，本文通过自编码器提取时间序列片段中的非线性特征。具体方法为：通过滑动窗口对时间序列样本进行分段，据此生成若干时间序列分段样本集，由正常时间序列上的不同位置上分段样本集训练自编码器。利用训练后的自编码器得到每个分段时间序列样本的低维特征表示。低维特征表示向量集采用*K*-means算法进行聚类处理，从而实现时间序列样本集的符号化。正常时间序列的符号序列集构建HMM，并通过待测样本在生成的HMM上的输出概率值进行异常检测。单变量和多变量时间序列数据集上的实验表明了文中所提方法具有较好的异常检测效果。

1 AHMM-AD方法

本文提出的基于自编码器和HMM的时间序列异常检测(Autoencoder and HMM-based Anomaly Detection，AHMM-AD)方法的具体流程如图1所示。该方法首先把样本集划分为带有正常时间序列样本的训练集、带有正常和异常时间序列样本的验证集和测试集；其次采用滑动窗口将正常时间序列样本集进行等长有重叠分段，由相同位置的正常时间序列片段样本集训练自编码器，训练集样本通过训练后的自编码器进行特征表示，并得到低维特征表示向量集；然后采用*K*-means聚类算法对向量集进行聚类处理，并实现对训练集样本的符号化；最后采用Baum-Welch算法对样本符号化序列集进行HMM建模，得到HMM的模型参数。验证集和测试集的样本经过与训练集相同的符号化处理过程得到符号化序列。验证集的每个样本通过训练完成的HMM计算输出概率，并在输出概率的最大值和最小值之间均匀划分1 000个值，根据F1值最大原则来确定阈值。根据测试集样本符号化序列的输出概率和验证集样本计算所得阈值对测试集进行异常检测。

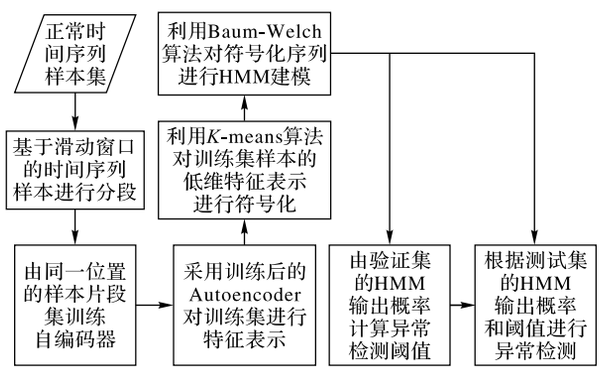

图1 本文时间序列异常检测方法流程

Fig. 1 Flowchart of the time series anomaly detection method in this paper

1.1 基于滑动窗口的时间序列分段

时间序列的分段方法通常有两种：序列关键点(间断点和突变点)分段[12,16]和等长有重叠时间窗口分段[11,17-18]。第一种方法通过序列中的间断点和突变点来分段描述时间序列，然而当数据在连续上升或下降的幅度较小时，序列存在的转折点或突变点无法识别[9]，这将会影响时间序列的分段表示。为此，大部分的方法研究采用第二种方法将原始时间序列创建为固定大小的段。这样的方法不仅能够保证将持续时间较长的数据模式被完整地分割出来[17]，还保持了原有时间序列数据在时序上的依赖性[11]。文献[11]采用固定滑动窗口的方法分段并分析分段中存在的形态模式特征，较好地实现时间序列局部特征的符号化。文献[17]采用有重叠滑动窗口分段视频数据，并建立行为识别模型。本文采用等长有重叠时间窗口分段方法的具体实现为：设包含N个多维时间序列的样本集记为$\boldsymbol{X}=\{\boldsymbol{X}^1,\boldsymbol{X}^2,\cdots,\boldsymbol{X}^n,\cdots,\boldsymbol{X}^N\}$，其中单个长度为$T$的样本可表示为$\boldsymbol{X}^n=(\boldsymbol{x}_0^n,\boldsymbol{x}_1^n,\cdots,\boldsymbol{x}_t^n,\cdots,\boldsymbol{x}_T^n),\boldsymbol{x}_t^n\in\mathbf{R}^m$表示第$n$个样本$t$时刻$m$维的向量。样本集中每个样本$\boldsymbol{X}^n$应用滑动窗口以窗口大小$w$和步长$s$滑动截取，样本的划分表示如下：

$$\boldsymbol{X}^{sn}=f(\boldsymbol{X}^n)=\begin{bmatrix}\boldsymbol{X}_1^n\\ \boldsymbol{X}_2^n\\ \vdots\\ \boldsymbol{X}_i^n\\ \vdots\\ \boldsymbol{X}_d^n\end{bmatrix}=\begin{bmatrix}\left[\boldsymbol{x}_1^n:\boldsymbol{x}_{1+w}^n\right]\\ \left[\boldsymbol{x}_{1+s}^n:\boldsymbol{x}_{1+s+w}^n\right]\\ \vdots\\ \left[\boldsymbol{x}_{1+(i-1)s}^n:\boldsymbol{x}_{1+(i-1)s+w}^n\right]\\ \vdots\\ \left[\boldsymbol{x}_{1+(d-1)s}^n:\boldsymbol{x}_T^n\right]\end{bmatrix}$$

其中:$\boldsymbol{X}^{sn}$表示第n个样本片段集,$\boldsymbol{X}_i^n$表示分段后第n个样本的第i个时间序列片段,$f(x)$表示滑动窗口函数,d为切片后的时间序列的片段总数,$d=(T-w)/s+1$。

1.2　基于autoencoder的时间序列符号化

自编码器能够在压缩高维数据的同时保留数据中重要特征,本文采用多个自编器提取不同时间序列片段中的特征。图2为本文的自编码器训练过程示意图。

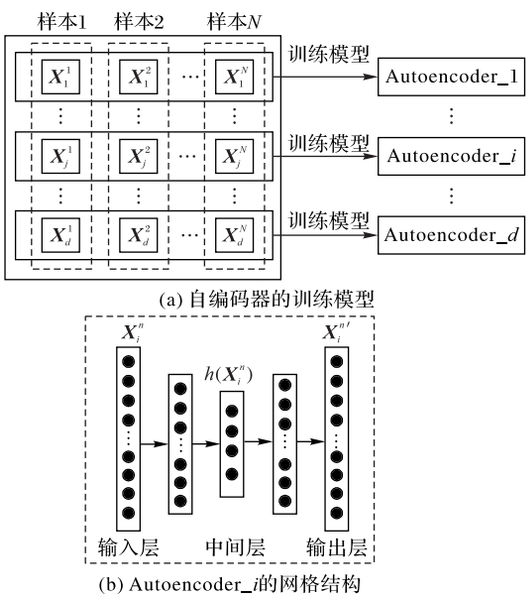

图2　自编码器训练过程示意图

Fig. 2　Schematic diagram of training process of autoencoder

由1.1节描述的方法对正常时间序列样本分段处理,且所有样本位于同一分段位置的数据片段为$\boldsymbol{X}_i=\{\boldsymbol{X}_i^1,\boldsymbol{X}_i^2,\cdots,\boldsymbol{X}_i^N,i=1,2,\cdots,d\}$,将其用于训练分段位置$i$的自编码器Autoencoder_$i$。图2(b)是第$i$个自编码器的网络结构,输入层到隐含层的转换表示编码过程(Encoder),隐含层到输出层的转换表示解码过程(Decoder)。Autoencoder_i的输入为$\boldsymbol{X}_i^n$,输入样本集的第i个片段经编码器得到的低维表示为$\boldsymbol{h}_i^n$。解码过程是把低维表示$\boldsymbol{h}_i^n$映射到输出层,对输入$\boldsymbol{X}_i^n$重构为$\boldsymbol{X}_i^{n\prime}$。自编码器的训练目标是最小化$\boldsymbol{X}_i^n$与$\boldsymbol{X}_i^{n\prime}$之间的重构误差,重构误差函数为$J(\boldsymbol{W},\boldsymbol{b},\boldsymbol{X}_i^n)$。编码和解码过程可由式(1)、(2)表示,重构误差函数由式(3)表示:

$$\boldsymbol{h}_i^n=\sigma_e\left(\boldsymbol{W}^{(1)}\boldsymbol{X}_i^n+\boldsymbol{b}^{(1)}\right)\tag{1}$$

$$\boldsymbol{X}_i^{n\prime}=\sigma_d\left(\boldsymbol{W}^{(2)}\boldsymbol{h}_i^n+\boldsymbol{b}^{(2)}\right)\tag{2}$$

$$J\left(\boldsymbol{W},\boldsymbol{b},\boldsymbol{X}_i^n\right)=\left\|\boldsymbol{X}_i^{n\prime}-\boldsymbol{X}_i^n\right\|^2\tag{3}$$

其中:参数集合为$\theta=\{\boldsymbol{W}^{(1)},\boldsymbol{b}^{(1)},\boldsymbol{W}^{(2)},\boldsymbol{b}^{(2)}\}$;$\sigma_e$和$\sigma_d$是激活函数,本文使用的是激活函数sigmoid。

第n个样本$\boldsymbol{X}^{sn}$经自编码器特征表示后得到的特征序列为$\boldsymbol{H}^n=(\boldsymbol{h}_1^n,\boldsymbol{h}_2^n,\cdots,\boldsymbol{h}_i^n,\cdots,\boldsymbol{h}_d^n)$。样本集$\boldsymbol{X}^s$的低维特征表示向量集为$\boldsymbol{H}=\{\boldsymbol{H}^1,\boldsymbol{H}^2,\cdots,\boldsymbol{H}^n,\cdots,\boldsymbol{H}^N\}$,其符号化是由$K$-means聚类算法实现。符号化样本序列可表示为$\boldsymbol{O}=\{\boldsymbol{O}^1,\boldsymbol{O}^2,\cdots,\boldsymbol{O}^n,\cdots,\boldsymbol{O}^N\}$,其中单个样本符号化序列为$\boldsymbol{O}^n=(O_1^n,O_2^n,\cdots,O_i^n,\cdots,O_d^n)$。在使用该算法之前,本文实验使用肘部方法(Elbow Method)验证参数——符号个数K的设置[14]。文中基于自编码器的时间序列符号化描述如下所示。

输入　样本片段集$\boldsymbol{X}^s=\{\boldsymbol{X}^{s1},\boldsymbol{X}^{s2},\cdots,\boldsymbol{X}^{sn},\cdots,\boldsymbol{X}^{sN}\}$,其中每个样本$\boldsymbol{X}^{sn}=(\boldsymbol{x}_1^n,\boldsymbol{x}_2^n,\cdots,\boldsymbol{x}_i^n,\cdots,\boldsymbol{x}_d^n)$;

输出　符号化样本序列$\boldsymbol{O}=\{\boldsymbol{O}^1,\boldsymbol{O}^2,\cdots,\boldsymbol{O}^n,\cdots,\boldsymbol{O}^N\}$,其中$\boldsymbol{O}^n=(O_1^n,O_2^n,\cdots,O_i^n,\cdots,O_d^n)$。

开始

For 每一个时间序列样本

$\boldsymbol{X}^{sn}=(\boldsymbol{x}_1^n,\boldsymbol{x}_2^n,\cdots,\boldsymbol{x}_i^n,\cdots,\boldsymbol{x}_d^n)$; $1\leqslant n\leqslant N$

For 每一个样本片段$\boldsymbol{X}_i^n(1\leqslant i\leqslant d)$

$\boldsymbol{h}_i^n=\sigma_e\left(\boldsymbol{W}^{(1)}\boldsymbol{X}_i^n+\boldsymbol{b}^{(1)}\right)$

$\boldsymbol{H}^n=\boldsymbol{H}^n\cup\boldsymbol{h}_i^n$

End for

$\boldsymbol{H}=\boldsymbol{H}\cup\boldsymbol{H}^n$

End for

由肘部方法确定聚类中心的个数K

采用K-means算法确定K个聚类中心$Core=\{\mu_1,\mu_2,\cdots,\mu_K\}$;

For 特征表示样本$\boldsymbol{H}^n(1\leqslant n\leqslant N)$

For 每个样本的特征片段$\boldsymbol{h}_i^n(1\leqslant i\leqslant d)$

根据特征片段与各聚类中心的距离,将距离最近的聚类中心确定数据片段的簇标记:

$\lambda_j=\underset{j\in\{1,2,\cdots,j,\cdots,k\}}{\arg\min}\left\|\boldsymbol{h}_i^n-\mu_j\right\|_2$

$O_i^n=\lambda_j$

$\boldsymbol{O}^n=\boldsymbol{O}^n\cup O_i^n$

End for

$\boldsymbol{O}=\boldsymbol{O}\cup\boldsymbol{O}^n$

End for

返回符号化序列$\boldsymbol{O}=\{\boldsymbol{O}^1,\boldsymbol{O}^2,\cdots,\boldsymbol{O}^n,\cdots,\boldsymbol{O}^N\}$

结束

1.3　本文的时间序列异常检测算法

本文时间序列异常检测的算法具体为:HMM是由隐藏状态序列和观测序列构成的双重随机过程,序列的每一个位置对应一个时刻的隐藏状态和观测状态。HMM由$\boldsymbol{G},\boldsymbol{M},\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B}$来确定,其中,$\boldsymbol{G}$表示HMM的隐藏状态集合;$\boldsymbol{M}$表示时间序列符号化后有限的观测状态集合,且$\boldsymbol{M}$中的观测状态个数由肘部方法确定;$\boldsymbol{\pi}$表示初始状态概率向量,是由各个隐藏状态的初始概率组成;$\boldsymbol{A}$表示状态转移概率矩阵,是由隐藏状态之间的转换概率组成;$\boldsymbol{B}$表示输出概率矩阵,是由隐藏状态下输出观测值的概率组成。

文中将样本集按比例划分为训练集$\boldsymbol{X}^{\text{train}}$、验证集$\boldsymbol{X}^{\text{val}}$和测试集$\boldsymbol{X}^{\text{test}}$。$\boldsymbol{X}^{\text{train}}$用于训练HMM,目的在于对训练集的正常时序行为建模。由1.2节描述的基于autoencoder模型的符号化方法处理$\boldsymbol{X}^{\text{train}}$,得到训练集的样本符号化序列$\boldsymbol{O}^{\text{train}}$。由于在HMM的训练数据只包含观测序列没有对应的隐藏状态序列,所以HMM的训练过程运用非监督学习算法——Baum-Whelm算法,学习HMM的参数$\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$。验证集$\boldsymbol{X}^{\text{val}}$用来计算本文异常检测方法的阈值$\tau$,测试集$\boldsymbol{X}^{\text{test}}$用来测试本文方法的异常检测效果。$\boldsymbol{X}^{\text{val}}$和$\boldsymbol{X}^{\text{test}}$同样经符号化处理后得到$\boldsymbol{O}^{\text{val}}$

和$\boldsymbol{O}^{\text{test}}$，并依次输入到训练后的HMM中，计算每个样本的输出概率。在验证集样本的输出概率的最大值与最小值之间均匀划分1 000个值作为异常检测的阈值候选值，F1值最高时所取的τ值即为本文方法的阈值，F1值的计算如式(4)~(6)所示。测试集的符号化样本序列$\boldsymbol{O}^{\text{test}}$计算的输出概率表示为$\boldsymbol{P}^{\text{test}}=\left\{p^{test1},p^{test2},\cdots,p^{testn},\cdots,p^{testN}\right\}$，其中$p^{testn}$表示测试集的第$n$个样本的输出概率。结合阈值$\tau$，当$p^{testn}<\tau$时，判定当前样本为异常样本；否则，判定其为正常样本。由于本文是采用正常时间序列建模，所以p^{testn}的值越小表示发生异常的可能性越大。

$$Precision=\frac{TP}{TP+FP} \tag{4}$$

$$Recall=\frac{TP}{TP+FN} \tag{5}$$

$$F1=\frac{2Recall*Precision}{Recall+Precision} \tag{6}$$

其中：TP(True Positives)表示检测为正常的正常样本数；FP(False Positives)表示检测为正常的异常样本数；FN(False Negatives)表示检测为异常的正常样本数；TN(True Negatives)表示检测为异常的异常样本数。

算法描述如下所示：

输入 验证集$\boldsymbol{X}^{\text{val}}=\left\{\boldsymbol{X}^{val1},\boldsymbol{X}^{val2},\cdots,\boldsymbol{X}^{valn},\cdots,\boldsymbol{X}^{valN}\right\}$，测试集$\boldsymbol{X}^{\text{test}}=\left\{\boldsymbol{X}^{test1},\boldsymbol{X}^{test2},\cdots,\boldsymbol{X}^{testn},\cdots,\boldsymbol{X}^{testN}\right\}$，正常时间序列集上训练完成的HMM的模型$\lambda=(\boldsymbol{\pi},\boldsymbol{A},\boldsymbol{B})$；

输出 测试集样本的输出概率$\boldsymbol{P}^{\text{test}}=\{p^{test1},p^{test2},\cdots,p^{testn},\cdots,p^{testN}\}$。

开始

通过基于1.2节的符号化算法，将验证集$\boldsymbol{X}^{\text{val}}$转化为$\boldsymbol{O}^{\text{val}}=\left\{\boldsymbol{O}^{val1},\boldsymbol{O}^{val2},\cdots,\boldsymbol{O}^{valn},\cdots,\boldsymbol{O}^{valN}\right\}$，其中$\boldsymbol{O}^{valn}=\left(O_1^n,O_2^n,\cdots,O_i^n,\cdots,O_d^n\right)$；

For $n=val1$ to $valN$ do

$p^{valn}=P\left(\boldsymbol{O}^{valn}|\lambda\right)$；

$\boldsymbol{P}^{\text{val}}=\boldsymbol{P}^{\text{val}}\cup p^{valn}$；

End for

在$\left(\min\left(\boldsymbol{P}^{\text{val}}\right),\max\left(\boldsymbol{P}^{\text{val}}\right)\right)$范围内均匀划分1 000个值，将每个值作为异常检测阈值，根据式(4)、(5)、(6)分别计算$\boldsymbol{X}^{\text{val}}$上的F1值；

将最大F1值对应的概率值P^{valn}作为文中的阈值τ；

通过基于autoencoder的符号化算法，将测试集$\boldsymbol{X}^{\text{test}}$转化为$\boldsymbol{O}^{\text{test}}=\left\{\boldsymbol{O}^{test1},\boldsymbol{O}^{test2},\cdots,\boldsymbol{O}^{testn},\cdots,\boldsymbol{O}^{testN}\right\}$，其中$\boldsymbol{O}^{testn}=\left(O_1^n,O_2^n,\cdots,O_i^n,\cdots,O_d^n\right)$；

For $testn=test1$ to $testN$ do

$p^{testn}=P\left(\boldsymbol{O}^{testn}|\lambda\right)$；

$\boldsymbol{P}^{\text{test}}=\boldsymbol{P}^{\text{test}}\cup p^{testn}$；

End for

在$\left(\min\left(\boldsymbol{P}^{\text{val}}\right),\max\left(\boldsymbol{P}^{\text{val}}\right)\right)$范围内均匀划分1 000个值，将每个值作为异常检测阈值，根据式(4)、(5)、(6)分别计算$\boldsymbol{X}^{\text{val}}$上的F1值；

将最大F1值对应的概率值P^{valn}作为文中的阈值τ；

通过基于autoencoder的符号化算法，将测试集$\boldsymbol{X}^{\text{test}}$转化为$\boldsymbol{O}^{\text{test}}=\left\{\boldsymbol{O}^{test1},\boldsymbol{O}^{test2},\cdots,\boldsymbol{O}^{testn},\cdots,\boldsymbol{O}^{testN}\right\}$，其中$\boldsymbol{O}^{testn}=\left(O_1^n,O_2^n,\cdots,O_i^n,\cdots,O_d^n\right)$；

For $testn=test1$ to $testN$ do

$p^{testn}=P\left(\boldsymbol{O}^{testn}|\lambda\right)$；

$\boldsymbol{P}^{\text{test}}=\boldsymbol{P}^{\text{test}}\cup p^{testn}$；

End for

For $testn=test1$ to $testN$ do

if $p^{testn}<\tau$ then

判定p^{testn}所对应的样本$\boldsymbol{X}^{testn}$为异常样本；

Else

判定p^{testn}所对应的样本$\boldsymbol{X}^{testn}$为正常样本；

End if

End for

结束

2 实验与分析

2.1 实验设置

为验证本文方法的有效性，选用了6个数据集进行实验。数据集描述如下：

1)雅虎(Yahoo)数据集是记录Yahoo会员的登录状态的单维时间序列数据集。样本数目300，每个样本长1 420，样本异常率为33.33%(https://webscope.sandbox.yahoo.com/)。

2)电力需求(Power Demand)数据集是记录一个家庭一年的电力需求的单维时间序列。实验中将长为35 000的数据集等长截为100个样本，每个样本长350，样本异常率为9%(https://www.cs.ucr.edu/~eamonn/discords/)。

3)呼吸(Respiration)数据集是记录病人醒来时胸腔扩张频率的单维时间序列。实验将长为42 000数据集等长截为140个样本，每个样本长300，样本异常率为7.86%(https://www.cs.ucr.edu/~eamonn/discords/)。

4)简易电子传输协议(Simple Mail Transfer Protocol，SMTP)数据集是记录网络流量和攻击手段的三维时间序列。实验将长为95 040数据集等长截为264个样本，每个样本长360，异常样本占有率7.86%(https://www.openml.org/)。

5)基于多传感器数据融合的活动识别系统(Activity Recognition system based on Multisensor data fusion，AReM)数据集记录了一个人七种动作的六维时间序列数据集，样本数目78，每个样本长480，实验使用动作类型作为标签，其中将走路动作的15个样本作为异常样本，样本异常率为19.23%(https://archive.ics.uci.edu/ml/datasets/)。

6)美国国家航空航天局飞行(NASA Flight)数据集是美国宇航局在多核异常检测(Multiple Kernel Anomaly Detection，MKAD)项目中记录B_777飞机飞行过程的五维时间序列数据集。样本数目300，每个样本长1 000，样本异常率为4%(https://c3.nasa.gov/dashlink/resources/136/)。

实验的软件环境为：tensorflow1.10.0，Python3.6.2，Windows10 64位操作系统；硬件环境为：Intel Core i7-3770处理器，4 GB内存。

2.2 参数选取

影响本文异常检测方法的性能主要有两个参数：一个是时间序列分段个数d；另一个是符号化序列所需符号的个数K。下面以Yahoo数据集的实验为例，讨论上述两个参数的选择过程。

实验选择10~50长度范围的重构序列来测试计算效率、平均重构误差和F1值，结果如表1。从表1中可看出，当样本序列随着时间序列分段个数的增多，训练时间将会变长，但时间持续较短对模型整个训练影响较小；平均重构误差也相差较小；F1值在序列长度为23时最高为0.63，综合分析Yahoo数据集的重构长度可设置在20~30范围内较为合适。

表1　时间序列不同分段个数的实验结果对比

Tab. 1　Experimental results comparison of different time series segmentation number

时间序列分段数(d)	训练时间/s	平均重构误差	F1值
13	15.7174	0.5501	0.463
23	17.9311	0.5310	0.630
33	28.9500	0.5000	0.409
43	31.9050	0.5624	0.579
53	39.1536	0.5840	0.457

实验使用K-means聚类方法对经过特征表示的样本序列进行符号化,K值由肘部方法确定。图3为Yahoo数据集在对比实验基于HMM异常检测方法中符号化的肘部曲线,当K=8时,平均畸变程度变化较高,即符号个数取为8。图4是该数据集在本文异常检测方法中符号化的肘部曲线,由图4可看出K值被确定为40。

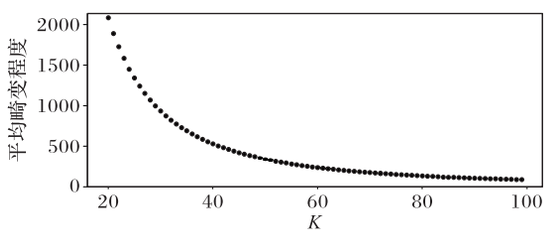

图3　肘部方法确定基于HMM异常检测方法的K值实验图

Fig. 3　Experimental diagram of K value of HMM-based anomaly detection method determined by elbow method

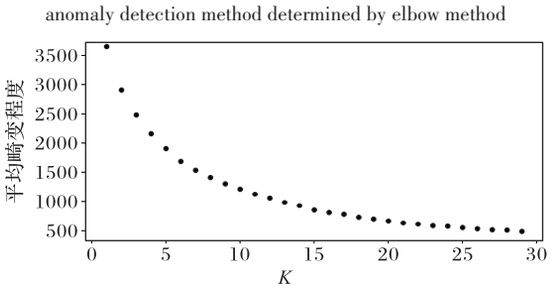

图4　肘部方法确定本文方法的K值实验图

FIG. 4　Experimental diagram of K value of the method in this paper determined by elbow method

按此原理,所有数据集上的时间序列分段个数d和符号化序列所需符号个数K值设置如表2所示。

2.3　实验结果与分析

实验中对比方法有:基于HMM的时间序列异常检测的方法(Hidden Markov Model-based Anomaly Detection, HMM-AD)[8],该方法中采用聚类方法对原始时间序列聚类处理并进行符号化;基于自编码器模型的时间序列异常检测方法(Autoencoder-based Anomaly Detection, AE-AD)[19],该方法通过学习重构正常的时序行为,并利用重构误差检测异常。采用精确率(Precision)、召回率(Recall)和F1值作为异常检测的评判标准。实验结果如下:

1)在Yahoo、Power demand和Respiration三个单变量时间序列上的实验结果如表3所示。从3个数据集的实验结果可以看出,本文提出的AHMM-AD方法均取得了较好的结果。与HMM-AD模型相比,本文方法在3个数据集的检测中,Precision、Recall和F1值均有所改进,对比实验结果说明仅仅在原始数据进行符号化分析,不能避免噪声的负面干扰,但是在符号化之前的特征表示有利于符号化过程对原始时间序列的表征。然而,与AE-AD方法相比,本文将autoencoder融合于HMM的方法在3个数据集的F1值平均提高0.339,表明先采用自编码器对样本片段分别进行局部提取表示,再用HMM对低维的样本片段建模的方法在时间序列异常检测中是有效的。

表2　实验参数设置

Tab. 2　Setting of experimental parameters

数据集	样本划分片段	符号化个数(K值)	HMM状态数
Yahoo	23	8	5
Power demand	26	5	3
Respiration	21	7	5
NASA Flight	26	10	5
AReM	26	8	5
SMTP	21	8	5

2)在NASA Flight、SMTP和AReM三个多变量时间序列的实验结果如表3所示。从表中可看出,AReM数据集的检测效果在多变量时间序列中表现最好,其原因是该数据集的异常样本是不同于正常样本的行为动作,异常样本模式突出。值得注意的是,AE-AD模型在多变量时间序列的检测效果没有和单变量时间序列的样本的一样优于HMM-AD,因此,HMM-AD模型在多变量时间序列的异常检测中有一定的优越性。而本文所提出的模型与HMM-AD模型相比,在Recall值上具有较高的检测得分,分别为0.916,0.700和0.890,且在F1值也有明显的提高。实验表明了文中的AHMM-AD能在HMM-AD的基础上能进一步提高在多变量时间序列集上异常检测效果。

表3　单变量和多变量时间序列实验结果

Tab. 3　Experimental results of univariate and multivariate time series

时间序列类别	数据集	HMM-AD			AE-AD			AHMM-AD		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
单变量时间序列	Yahoo	0.231	0.150	0.182	0.506	0.233	0.315	0.750	0.540	0.630
	Respiration	0.317	0.146	0.185	0.257	0.182	0.206	0.490	0.682	0.569
	Power Demand	0.294	0.220	0.253	0.355	0.183	0.239	0.679	0.554	0.581
多变量时间序列	NASA Flight	0.170	0.430	0.265	0.160	0.500	0.250	0.275	0.916	0.423
	AReM	1.000	0.330	0.463	1.000	0.125	0.220	0.607	0.700	0.636
	SMTP	0.165	0.445	0.239	0.275	0.360	0.248	0.400	0.890	0.552

3　结语

针对已有基于隐马尔可夫模型的时间序列异常检测模型的符号化方法不能很好地表征原始时间序列的问题,本文提出了基于自编码器和HMM的时间序列异常检测方法。该方法利用自编码器对时间序列片段的特征表示优化符号化方法,进而提高了HMM表达正常时间序列的建模能力。在单变量和多变量时间序列上的实验表明了文中方法的有效性。由

于本文方法在对时间序列片段的特征表示过程中没有考虑片段内的时序关系，下一步将采用长短时间记忆网络学习时间序列片段的特征表示，以进一步提高文中所提方法的异常检测效果。

参考文献（References）

[1] LI L, HANSMAN R J, PALACIOS R, et al. Anomaly detection via a Gaussian mixture model for flight operation and safety monitoring [J]. Transportation Research Part C: Emerging Technologies, 2016, 64: 45-57.

[2] CAO Y, LI Y, COLEMAN S, et al. Adaptive hidden Markov model with anomaly states for price manipulation detection [J]. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(2): 318-330.

[3] ZHANG C, CHEN Y, YIN A, et al. Anomaly detection in ECG based on trend symbolic aggregate approximation [J]. Mathematical Biosciences and Engineering, 2019, 16(4): 2154-2167.

[4] RABINER L R. A tutorial on hidden Markov models and selected applications in speech recognition [J]. Proceedings of the IEEE, 1989, 77(2): 257-286.

[5] 孙群丽，刘长良，甄成刚. 隐马尔科夫模型在滚动轴承故障诊断中的应用[J]. 热能动力工程, 2018, 33(10):95-100. (SUN Q L, LIU C L, ZHEN C G. Application of hidden Markov model in fault diagnosis of rolling bearing [J]. Journal of Engineering for Thermal Energy and Power, 2018, 33(10):95-100.)

[6] 刘子豪，李凌，叶枫. 基于SparkR的水文传感器数据的异常检测方法[J]. 计算机应用, 2019, 39(2): 436-440. (LIU Z H, LI L, YE F. Anomaly detection method for hydrologic sensor data based on SparkR [J]. Journal of Computer Applications, 2019, 39(2): 436-440.)

[7] SONG C, LIU K, ZHANG X, et al. An obstructive sleep apnea detection approach using a discriminative hidden Markov model from ECG signals [J]. IEEE Transactions on Biomedical Engineering, 2016, 63(7): 1532-1542.

[8] KEYGHOBADI H, SEYEDIN A. Abnormality detection in a landing operation using hidden Markov model [J]. Journal of Computer and Robotics, 2017, 10(1): 31-37.

[9] 杨慧，毛好好，霍纬纲. 聚类HMM模型在QAR数据分析中的应用研究[J]. 计算机应用与软件, 2018, 35(1): 85-91. (YANG H, MAO H H, HUO W G. The application of clustering HMM model in QAR data analysis [J]. Computer Applications and Software, 2018, 35(1): 85-91.)

[10] LI J, PEDRYCZ W, JAMAL I. Multivariate time series anomaly detection: a framework of hidden Markov models [J]. Applied Soft Computing, 2017, 60: 229-240.

[11] 谭宏强，牛强. 基于滑动窗口及局部特征的时间序列符号化方法[J]. 计算机应用研究, 2013, 30(3): 796-798. (TAN H Q, NIU Q. Symbolic representation algorithm for time series based on sliding window and local features [J]. Application Research of Computers, 2013, 30(3): 796-798.)

[12] JIA Y, XU M, WANG R. Symbolic important point perceptually and hidden Markov model based hydraulic pump fault diagnosis method [J]. Sensors, 2018, 18(12): No. 4460.

[13] FU T C, HUNG Y K, CHUNG F L. Improvement algorithms of perceptually important point identification for time series data mining [C]// Proceedings of the IEEE 4th International Conference on Soft Computing and Machine Intelligence. Piscataway: IEEE, 2017: 11-15.

[14] ZEGEYE W K, DEAN R A, MOAZZAMI F. Multi-layer hidden Markov model based intrusion detection system [J]. Machine Learning and Knowledge Extraction, 2019, 1(1): 265-286.

[15] FARSAD N, RAO M, GOLDSMITH A. Deep learning for joint source-channel coding of text [C]// Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE, 2018: 2326-2330.

[16] 廖俊，周中良，寇英信，等. 一种基于重要点的时间序列分割方法[J]. 计算机工程与应用, 2011, 47(24): 166-170. (LIAO J, ZHOU Z L, KOU Y X, et al. Method for time series segment based on important point [J]. Computer Engineering and Applications, 2011, 47(24): 166-170.)

[17] 吴晓婕，胡占义，吴毅红. 基于Segmental-DTW的无监督行为序列分割[J]. 软件学报, 2008, 19(9):2285-2292. (WU X J, HU Z Y, WU Y H. Unsupervised behavior sequence segmentation based on Segmental-DTW [J]. Journal of Software, 2008, 19(9): 2285-2292.)

[18] LEKHA S, SUCHETHA M. Real-time non-invasive detection and classification of diabetes using modified convolution neural network [J]. IEEE Journal of Biomedical and Health Informatics, 2018, 22(5): 1630-1636.

[19] SAKURADA M, YAIRI T. Anomaly detection using autoencoders with nonlinear dimensionality reduction [C]// Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis. New York: ACM, 2014: 4-11.

This work is partially supported by the Civil Aviation Joint Research Fund of Committee of National Natural Science Foundation of China and Civil Aviation Administration of China (U1633110), the Special Fund for Civil Aviation University of China of Fundamental Research Funds for the Central Universities (3122019190).

HUO Weigang, born in 1978, Ph. D., associate professor. His research interests include data mining, fuzzy classification.

WANG Huifang, born in 1993, M. S. candidate. Her research interests include data mining.