基于层级实时记忆算法的时间序列异常检测算法

曾惟如[1],吴 佳[1],闫 飞[2]
(1. 电子科技大学信息与软件工程学院,四川成都 610054;
2. 西南交通大学信息科学与技术学院,四川成都 611756)

摘 要: 时间序列异常检测是数据分析中一个重要的研究领域. 传统的时间序列的异常检测方法主要通过比较检测数据和历史数据的差异程度,以判断被检测数据是否为奇异点(Surprise)、离群(Outlier)点等. 然而序列和窗口的划分,状态的划分或者异常的定义和判定等问题,使得这类方法存在一定的局限性. 本文针对传统时间序列检测算法不足,提出一种基于层级实时记忆算法的时间序列异常检测算法. 该方法对时间序列内在模式关系进行学习,建立预测模型,通过比较预测值和真实值的偏离程度来判断数据是否异常. 首先使用稀疏离散表征在保证保留数据相关性的同时又将数据离散化;然后输入到模型网络,预测下一时刻的数据值;最终根据预测值和真实值的差异为数据异常程度进行定量评分. 在人造数据和真实数据上的实验表明,该方法能够准确、快速地发掘时间序列中的异常.

关键词: 异常检测; 神经网络; 层级实时记忆; 稀疏离散表征
中图分类号: TP311.1 **文献标识码:** A **文章编号:** 0372-2112 (2018)02-0325-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.02.010

Time Series Anomaly Detection Model Based on Hierarchical Temporal Memory

ZENG Wei-ru[1], WU Jia[1], YAN Fei[2]
(1. *School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan* 610054, *China*;
2. *School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan* 611756, *China*)

Abstract: Time series anomaly detection is an important area of data mining. Traditional methods of time series anomaly detection usually find the surprise, outlier, etc., by comparing the data with the historical data. However, there are some limits with these methods, such as the inaccurate separation of the sequence, the false decision of the state and the window size or the incorrect definition and judgement of the anomaly. This paper proposes a time series anomaly detection model based on hierarchical temporal memory (HTM) to overcome the shortages of the traditional methods. This method can recognize and learn the intrinsic patterns in the time series and build a prediction model to determine an anomaly by comparing the real value with the predicted one. First, sparse distributed representation (SDR) is used to represent the raw data; then, the SDR is entered into the HTM model to make prediction; lastly, the proposed model evaluates the data by computing the difference of the actual value and the predicted one. The experiments on the artificial data and the real data show that HTM can detect anomalies accurately and quickly.

Key words: anomaly detection; neuron network; hierarchical temporal memory; sparse distributed representation

1 引言

随着信息产业的不断发展,我们已经步入了一个数据时代. 在实际生产生活过程中产生了大量结构化和非结构化的数据,这些数据为我们带来效益的同时也给我们提出了不小的挑战,即如何从中发掘有效的信息. 对数据中异常模式的检测(即异常检测)是数据分析领域一个非常重要的研究方向,尤其对时间序列的异常检测是其中的一个难点.

时间序列即按照一定顺序(时间顺序)记录得到的一系列的值,其广泛地存在于各种金融、医疗、工程和社会学数据库中. 时间序列具有两个特性:一是时间属性,

收稿日期:2016-09-02;修回日期:2017-04-25;责任编辑:李勇锋
基金项目:国家自然科学基金(No. 61503059, No. 61403316)

即每个变量的记录必须有时间维,并按照先后顺序进行排列;二是序列性,变量在某一时间段内以一定的规律发生着变化. 时间序列的异常检测包括两个难点:首先,对时间序列的异常检测要保持信息的时效性,即检测机制要能够不断地学习数据新的模式,实时分析并检测出异常;其次,能够发现数据在时间维度上的规律,而这些规律往往具有一定的时间跨度,增加了模式挖掘的难度. 另外,时间序列中的异常为小概率事件,导致了异常样本少,也增大异常检测难度[1].

传统的异常检测方法如高斯异常检测[2],支持向量机[3]等,并不考虑时间因素,因此不适用于时间序列的异常检测. 后续研究在传统的异常检测方法上引入滑动窗口技术[4],如 Etsy 公司开发的 skyline 系统[5],将最近的一段数据作为研究对象,同时用多种方法分析序列的统计特性,然后以投票的方式为数据异常评分,然而此方法依然无法有效地发掘时间序列的内部规律,特别是无法识别数据周期性规律. 也有研究者利用小波分析的方法分析时间序列在频率域上的信息[14],从而判断序列是否异常,但是这类方法计算量通常较大,以至于不能进行实时预测. 文献[6]指出利用历史数据建立时间序列的预测模型,通过比较预测数据和实际数据的差异,以判断是否出现异常. 该方法是解决时间序列异常检测问题的常见方法,将异常检测问题转化为预测问题,这样能够兼顾时间序列的两个特性. Stan Salvador 等人[7]提出的通过学习时间序列的状态和规则以检测时间序列异常的方法就符合这一思路. 该方法使用聚类算法将时间序列划分为不同状态,根据历史数据之间的状态转化建立简单的状态转化逻辑,以此实现对高阶时间序列数据预测. 但是该方法使用聚类方法对数据进行状态划分,将使数据缺失大量信息,如不同状态之间在数值上的联系和同一状态中不同数据的差异;对于高维数据,聚类算法也表现不佳. 另外,该方法所使用的状态转换逻辑也过于简单,不能对数据进行准确预测. 新兴的长短期记忆(Long Short-Term Memory,LSTM)模型[8]是一种专门处理时间序列的神经网络模型,能够较好的捕捉数据在时间维度上的关联,但是需要进行大量数据进行离线训练,使得该算法不具备实时性.

针对目前时间序列异常检测算法中的诸多不足,本文提出了一种基于层级实时记忆算法(Hierarchical Temporal Memory,HTM)的时间序列异常检测方法. HTM 算法是一种仿生学算法,该算法对脑皮质层神经元结构和组织方式进行模拟,形成“记忆 - 预测”的运作模式,使得该算法能够有效捕捉数据的时间维度上的信息,对时间序列进行准确、快速地预测. 本方法的优点体现在以下几方面:

(1) 算法自适应性强,能够根据数据的变化自动在线学习,及时发现数据的模式和规律;

(2) 实现数据的稀疏离散表征,对数据进行离散化,保留了原数据的信息,提高算法鲁棒性;

(3) 算法可以对数据进行实时检测,而无需采用滑动窗口法批处理数据;

(4) 能够对数据异常进行提前预警.

2 方法描述与整体架构

本文利用层级实时记忆算法实现时间序列的异常检测. HTM 模型对历史数据模式进行学习,即输入数据在空间分布上的模式,并且建立空间模式之间的时间联系,即模式序列. 接下来,依赖于已学习存储的大量模式序列,对下一时刻的数据模式进行预测和识别. HTM 网络模型以二值数据作为输入,因此需要对数据进行预处理,获得符合输入要求的数据.

整体系统的结构如图 1 所示,分成三个部分:编码模块、HTM 网络模型和评分模块. 编码模块负责建立输入数据的离散稀疏表征,将数值离散化,以符合后续模型的输入要求,同时尽量保留原数据信息;HTM 网络模型对输入数据中隐含的模式进行识别、存储和学习,预测下一时刻数据的模式;评分模块根据预测值和真实值的相异程度对被检测对象进行评定.

图1 系统整体架构

3 层级实时记忆算法

本节将介绍系统的主模块 HTM 网络模型的构建方法及学习规则. 首先介绍 HTM 神经元(细胞)模型,HTM 神经元是 HTM 网络最基本的组成单位,整个 HTM 网络的功能和特性都是以神经元的结构和学习方式为基础得以实现的;接下来描述高阶时间序列数据学习过程,介绍了 HTM 网络如何有机地将神经元组织成为一个网络,实现了对时间序列的识别、记忆和学习;最后本节将详细讲解 HTM 网络模型的学习规则.

3.1 HTM 神经元模型

HTM 神经元(如图 2 所示)是由文献[9]提出的新型人工神经元模型. HTM 神经元结构和学习规则与传统的人工神经元模型[16]的不同在于:传统神经元模型使用 sigmoid 等函数计算激励值;HTM 神经元则是以树突分支为基本单元来决定神经元是否被激活,树突分支的状态又由该树状区域中活跃突触的数量决定. HTM 神经元与传统的人工神经元最大的区别在于 HTM 构建了一个更复杂的神经元模型,模拟了生物神经元

的树突分支和大量的突触,每个突触就像一个“阈值符合探测器”,处理接收到的信息.

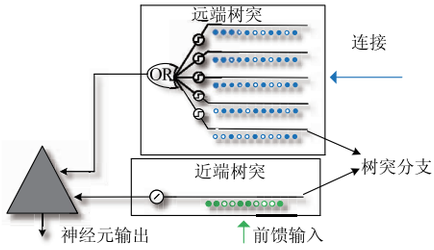


图2　HTM神经元模型

HTM 神经元包含两个输入区域:近端树突和远端树突. 近端树突包含树突分支数量较少,与前馈输入(输入数据)构成突触(图 2 中的树突分支上的小圆圈代表突触);远端树突则包含多个独立的树突分支,与同区域的其他细胞构成突触. 根据突触所处的区域,突触所传递的信息不相同. 近端突触传递前馈输入的信息;而远端突触传递的是来自于 HTM 网络中的其他细胞的信息(神经元输出). 当近端树突分支上同时活跃的突触数量达到一定阈值,该树突分支将被激活,进而激活细胞体,使得细胞体处于“活跃(激活)状态”. 同样,当某远端树突分支上同时活跃的突触的数量达到某阈值时,该树突分支也将被激活. 然而,活跃的树突分支不会激活细胞,而是使细胞进入极化状态,即“预测状态”. 即表示在当前输入下,该细胞在下一时刻可能被激活. 通过该结构 HTM 网络实现了模式的预测. 综上所述,神经元的状态分为:活跃状态、非活跃状态以及预测状态.

3.2　高阶时间序列数据预测

本节将对 HTM 如何组织各个神经元以实现对数据模式的预测进行简单介绍. HTM 网络模型是以细胞柱为基本结构单元. 如图 3 所示中的蓝色虚线框中所示,图中圆代表神经元细胞,黑色圆代表处于激活状态的神经元. 同一细胞柱内所有神经元细胞共享同一前馈输入,即细胞柱中的所有神经元共享同一近端树突. 整个网络中,细胞之间又通过远端突触相互连接.

HTM 网络模型对于时间序列的模式学习分为两步. 首先,学习输入数据的空间分布模式. HTM 通过少量活跃的细胞柱(存在活跃神经元的细胞柱)来表征网络当前输入数据的空间模式. 该模式表征处于细胞柱层面上. 接下来,HTM 对空间模式在时间维度上的前后关联进行学习,实现对高阶时间序列的预测. 该部分学习则体现在细胞层面上. 当一个神经元细胞变得活跃后,它会与前一时刻活跃的细胞建立连接(即形成突触);细胞通过关注他们的连接(突触)来预测其何时会被再次激活. 具体而言,由于网络中活跃细胞的存在,与这些活跃细胞建立连接的突触将被激活. 当某些细胞远端树突分支上活跃突触数量超过阈值时,这些细胞进入“预测状态”. 处于“预测状态”的细胞就表示对下一时刻数据模式的预测. 如果细胞柱中存在处于“预测状态”的细胞,且稍后这些细胞又有相应的前馈输入,那么预测状态细胞将进入活跃状态,并抑制细胞柱中的其他细胞;如果收到前馈输入的细胞柱中没有处于预测状态的细胞,那么表示由这些细胞柱表征的模式为新的模式,该细胞柱中的所有细胞将进入活跃状态.

为了更加直观的描述数据预测的过程,下面以 A – B – C – D 和 X – B – C – Y 这两个抽象的模式序列(如图 3)为例进行说明. 开始,输入数据的时间关联未被 HTM 网络学习到,输入数据的空间模式由稀疏的细胞柱表征,如图 3(*a*)所示. 在完成模式时间关联学习之后,HTM 网络采用不同方式表征在不同环境中的相同输入. 虽然同一输入仍采用同样的细胞柱表征,但是细胞柱中只有一个细胞变得活跃. 如图 3(*b*)所示, B’ 和 B” ,C’ 和 C”. 由于不同环境中的输入表征方式(被激活细胞)完全不同,因此能够进行高阶时间序列模式的预测.

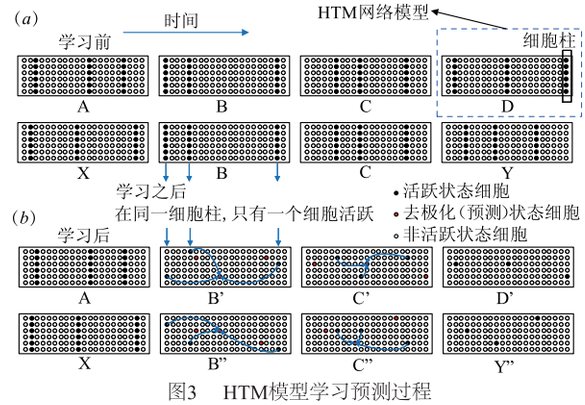


图3　HTM模型学习预测过程

3.3　HTM 神经元学习规则

本节将对 HTM 神经网络的学习规则进行详细地介绍. HTM 学习规则类似 Hebbian 学习规则[10],即加强做出了准确预测的突触的连接强度;削弱未做出准确预测的突触的连接强度. 在介绍其学习规则前,我们先定义如下符号:

假设 HTM 网络中有 N 个细胞柱,每个细胞柱中神经元细胞个数为 M. 另外,还包括如下参数:

(1)$\boldsymbol{A}^t=\{a_{ij}^t\}$:$t$ 时刻 HTM 网络中神经元活跃状态矩阵. a_{ij}^t表示在 t 时刻,网络中的第 j 个细胞柱中的第 i 个细胞是否处于活跃状态. $a_{ij}^t=1$ 表示处于活跃状态,$a_{ij}^t=0$ 表示处于非活跃状态. $\boldsymbol{A}^t$ 为 $M\times N$ 的二值矩阵.

(2)$\boldsymbol{\pi}^t=\{\pi_{ij}^t\}$:$t$ 时刻 HTM 网络中神经元预测状态矩阵. π_{ij}^t表示 t 时刻,网络中的第 j 个细胞柱中的第 i 个

细胞是否处于预测状态. $\pi_{ij}^{t}=1$ 即处于预测状态;反之,$\pi_{ij}^{t}=0$. $\boldsymbol{\pi}^{t}$ 为 $M\times N$ 的二值矩阵.

(3)$\boldsymbol{D}_{ij}^{d}$:HTM 网络连通值矩阵,表示细胞柱 j 中,神经元 i 的第 d 个树突分支上突触的连通性. $\boldsymbol{D}_{ij}^{d}$大小为 $M\times N$. 突触的连通值是指树突与轴突之间连接的程度. 当一个突触连接加强时,它的连通值增加;反之,减少. 连通值是一个标量,值域为[0.0,1.0].

(4)$\tilde{\boldsymbol{D}}_{ij}^{d}$:表示细胞柱 j 中,细胞 i 的第 d 个树突分支上已连通的突触. 当一个突触的连通值超过阈值时,那么该突触就是连通的,赋予其权值 1;反之,表示不连通并且权值为 0. $\tilde{\boldsymbol{D}}_{ij}^{d}$为二值矩阵,大小为 $M\times N$.

HTM 神经元学习过程如下:

(1) 初始化:首先初始化 HTM 网络,选定网络中的一个神经元子集,令其中的每一个神经元的树突分支包含一些潜在突触(连通值非 0);且随机设定潜在突触中的 50% 为连通的. 潜在突触指某些细胞的轴突足够接近其他细胞的树突,具有较大可能连通形成突触.

(2) 计算细胞状态:为了实现输入数据的稀疏表征,只选取一定较低比例的细胞柱被前馈输入激活,记为 $\boldsymbol{W}^{t}$. 当前时刻 t 神经元细胞的活跃状态计算过程如下:

$$a_{ij}^{t}=\begin{cases}1, & \text{if } j\in \boldsymbol{W}^{t} \text{ and } \pi_{ij}^{t-1}=1\\ 1, & \text{if } j\in \boldsymbol{W}^{t} \text{ and } \sum_{i}\pi_{ij}^{t-1}=0\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

上式第一行表示如果位于活跃细胞柱的某神经元在 $t-1$ 时刻已处于预测状态,那么该神经元在 t 时刻将被激活. 第二行表示如果位于活跃细胞柱的所有神经元在 $t-1$ 时刻都不处于预测状态,那么将该细胞柱内所有神经元将在 t 时刻被激活.

接下来,计算 t 时刻神经元的预测状态. 如果某细胞的树突分支连接的活跃突触数大于阈值 θ,则该神经元进入极化状态,即预测状态. 计算公式如下:

$$\pi_{ij}^{t}=\begin{cases}1, & \text{if } \exists_{d}\left\|\tilde{\boldsymbol{D}}_{ij}^{d}\circ\boldsymbol{A}^{t}\right\|_{1}>\theta\\ 0, & \text{otherwise}\end{cases} \tag{2}$$

其中,θ 表示树突分支被激活的阈值;$\circ$表示矩阵点乘.

(3) 修改突触连通值: HTM 神经元学习过程在于修改树突分支上潜在突触的连通值. 突触连通值的学习规则与 Hebbian 规则相似. 如果一个细胞树突分支连接的活跃突触数超过阈值,该细胞被激活,那么这个树突分支中突触的连通值将被修改. 活跃的突触有助于神经元细胞被激活,其连通性会增强;反之,不活跃的突触对神经元细胞激活没有贡献,其连通性将会下降. 因此,首先找到需要修改的神经元及其树突分支,其计算过程如下:

$$\forall_{j\in \boldsymbol{W}^{t}}(\pi_{ij}^{t-1}>0) \text{ and } \left\|\tilde{\boldsymbol{D}}_{ij}^{d}\circ\boldsymbol{A}^{t-1}\right\|_{1}>\theta \tag{3}$$

上式前半部分表示在 $t-1$ 时刻预测到该前馈输入的神经元,后半部分表示选择该神经元中在 $t-1$ 时刻被激活的树突分支 d.

如果在被激活的细胞柱中没有进入预测状态的神经元,则必须在该细胞柱里选择一个神经元来表示当前未被预测到的输入模式. 被选择的神经元的远端树突分支接近激活状态. 其计算过程如下:

$$\begin{aligned}&\forall_{j\in \boldsymbol{W}^{t}}\left(\sum_{t}\pi_{ij}^{t-1}=0\right) \text{ and}\\ &\left\|\dot{\boldsymbol{D}}_{ij}^{d}\circ\boldsymbol{A}^{t-1}\right\|_{1}=\max_{i}\left(\left\|\dot{\boldsymbol{D}}_{ij}^{d}\circ\boldsymbol{A}^{t-1}\right\|_{1}\right)\end{aligned} \tag{4}$$

其中,$\dot{\boldsymbol{D}}_{ij}^{d}$表示一个大小同 $\boldsymbol{D}_{ij}^{d}$的二值矩阵,$\dot{\boldsymbol{D}}_{ij}^{d}$中为 1 的元素对应 $\boldsymbol{D}_{ij}^{d}$中非零元素. 找到需要修改连通值的树突分支 d 后,根据如下规则修改该分支上所有突触的连通值:

$$\Delta\boldsymbol{D}_{ij}^{d}=p^{+}(\dot{\boldsymbol{D}}_{ij}^{d}\circ\boldsymbol{A}^{t-1})-p^{-}\dot{\boldsymbol{D}}_{ij}^{d} \tag{5}$$

上式对活跃的神经元细胞进行修改. 通过一个较大的值 p^{+}提高那些使细胞活跃的树突分支上突触的连通性,同时通过一个较小的值 p^{-}降低其他树突分支上突触的连通性. 另外,为了模拟细胞的长时程抑制[11],算法针对树突分支活跃而细胞体未被激活的神经元施加一个很小的衰退效果 p^{--}:

$$\begin{aligned}&\Delta\boldsymbol{D}_{ij}^{d}=p^{--}\boldsymbol{D}_{ij}^{d}\\ &\text{where } a_{ij}^{t}=0 \text{ and } \left\|\tilde{\boldsymbol{D}}_{ij}^{d}\circ\boldsymbol{A}^{t-1}\right\|_{1}>\theta\end{aligned} \tag{6}$$

其中,$p^{--}<<p^{-}$.

4 数据稀疏离散表征

在数据输入 HTM 网络模型之前,将输入数据转换成稀疏离散表征(Sparse Distributed Representation, SDR)[12]. 稀疏离散表征相对于传统信息表征方式具有众多优势,如高容量和极强的容错性,其证明已在相关文献[12]中给出.

SDR 作为算法接受信息的途径,应该尽量保留原数据的信息,体现数据的相关性. SDR 编码应该遵循如下准则:

(1)SDR 是一组由‘0’和‘1’组成的序列,每一位代表一个神经元细胞的状态. 仅用少量的活跃位(值为‘1’)来表征信息,以使编码具有稀疏性,且对于同一类型数据,活跃位比例保持不变;

(2)实际意义越相近的数据,其对应的 SDR 也相似,即相同位置活跃位的位数较多.

具体的实现过程可参考文献[17].

5 异常评分机制

本节将介绍评判数据异常的机制. 该方法通过比较 HTM 网络预测结果和输入数据的相异程度来进行评分. 异常分值的计算公式如下:

$$S=\frac{|\boldsymbol{A}_{t}-(\boldsymbol{P}_{t-1}\cap\boldsymbol{A}_{t})|}{|\boldsymbol{A}_{t}|} \tag{7}$$

其中,$\boldsymbol{P}_{t-1}$ 为 $t-1$ 时刻处于预测状态的细胞柱,$\boldsymbol{A}_t$ 为 t 时刻处于活跃状态的细胞柱. $\boldsymbol{P}_{t-1} \cap \boldsymbol{A}_t$ 表示预测到在 t 时刻处于激活状态的细胞柱. 上式分母表示激活细胞柱的个数;分子计算了激活的细胞柱中,未被正确预测的个数. 异常分值 S 越接近 1,则数据异常程度越高.

至此,本方法的基本步骤已介绍完毕. 该方法将对实时的每一条数据按时间顺序执行以上步骤,计算异常分值,对异常程度进行度量.

6　实验分析

本节将对基于层级实时记忆算法的时间序列异常检测算法性能进行测试分析. 测试数据集分为人造数据和真实数据. 实验对比方法为自回归积分活动平均模型(Autoregressive Integrated Moving Average, ARIMA)[13]和基于统计分析的异常检测商业开源软件 Skyline 以及 twitter 开源项目 AnomalyDetection[15]. 实验环境为 CPU Intel ® Core™ i5 – 4460 CPU @ 3.20GHz × 4、内存 8GB、硬盘 20GB、Ubuntu Kylin 14.04 操作系统.

6.1　人造数据集

本方法是以预测为基础,通过比较预测值和真实值的偏差程度判断数据是否异常,预测的准确性和高效性直接影响异常检测的效果. 因此在预测能力上,本文将 HTM 和常用 ARIMA 预测方法在人造数据进行预测准确性和时间开销的比较,以验证 HTM 能够为本异常方法提供可靠而高效的预测基础. 人造数据集由单一模式的无噪声数据、单一模式含噪声数据、多模式的无噪声数据构成. 本节将在这三个数据集上分别对本算法预测的准确性进行测试.

6.1.1　数据生成和评价指标

人造数据序列 $S=[s_1,\cdots,s_i,\cdots,s_N]$,$N=5000$,$s_i$ 计算方法如表 1 所示.

表 1　测试数据集

单一模式无噪声数据集 S_1		$s_i=0.5\times\sin(i\times\pi/10)+0.5\times\sin(i\times\pi/5)$	
单一模式含噪声数据	S_2	$s_i=0.5\times\sin(i\times\pi/10)+0.5\times\sin(i\times\pi/5)+N(\mu,\sigma)$	$\mu=0,\sigma=0.01$
	S_3		$\mu=0,\sigma=0.03$
	S_4		$\mu=0,\sigma=0.06$
	S_5		$\mu=0,\sigma=0.09$
多模式无噪声数据 S_6 $\mu=0,\sigma=0$		$s_i=0.5\times\sin(i\times\pi/10)+0.5\times\sin(i\times\pi/5)$	$i\in[1,2500]$
		$s_i=\sin(i\times\pi/10)+N(\mu,\sigma)$	$i\in(2500,5000]$

为了评估模型对数据预测的准确性,本文采用相对百分比误差(Mean Absolute Percentage Error, MAPE)衡量模型准确度,序列 S 的 MAPE 值被定义为:

$$\mathrm{MAPE}=\frac{\sum_{i=1}^{N}|s_i-\hat{s}_i|}{\sum_{i=1}^{N}|s_i|} \tag{8}$$

其中,$\hat{s}_i$ 为预测值,s_i 为实际值.

6.1.2　实验结果及分析

我们分别采用 HTM 和 ARIMA 算法对上述人造数据进行预测,其中 HTM 的参数如表 2 所示. 为了使 ARIMA 能够实时预测,我们采用了滑动窗口的方法,设定窗口长度为 100,对未来 5 条数据进行预测.

表 2　HTM 网络模型参数表

参数名	参数值
细胞柱数量 N	2048
细胞柱内细胞数量 M	32
树状分支的活跃阈值 θ	14
突触的初始连通值	0.21
突触连通的阈值	0.5
突触连通值的增益值 $p+$	0.1
突触连通值的衰退值 $p-$	0.1

图 4 显示了在不同数据集上 MAPE 值的变化情况. 图中每一个数据点为连续 100 条数据的 MAPE 值,共有 49 个数据点. 从图中我们可以看出,在 6 个数据集上 HTM 预测准确性都优于 ARIMA,特别是对于低噪声数据(数据集 S_1,S_2 和 S_3),HTM 预测的准确度明显高于 ARIMA 模型. 对于无噪声和低噪声的数据 $S_1 \sim S_3$,HTM 开始未学习到数据的变化模式,准确度较差,随着学习的进行,HTM 学习到数据的变化模式,MAPE 值迅速下降,无噪声数据集的 MAPE 值甚至趋于 0;而 ARIMA 预测准确度始终保持在一定范围内波动. 在高噪声模式下,HTM 性能略优于 ARIMA(如图 4(d)和 4(e)所示). 在多模式下(如图 4(f)),我们注意到当数据模式发生改变时,HTM 和 ARIMA 的预测准确性受到影响,但是在学习一段时间后,HTM 学习到新数据模式,其 MAPE 值不断下降,最终趋于稳定,可以看出 HTM 算法能够学习新模式. 使用滑动窗口的 ARIMA 算法不能够很好的适应新的模式,MAPE 增大之后不再下降.

设 HTM 和 ARIMA 的预测结果的 MAPE 值分别为 M_h 和 M_a,下面分别对每个数据集中成对的数据(M_h, M_a)进行统计显著性分析. 设

$$D_i=M_{ai}-M_{hi}(i=1,2,\cdots,n)$$

为来自正态总体 $N(\mu_d,\sigma_D^2)$ 的样本,基于这一样本检验假设:

$$H_0:\mu_D\leqslant 0(\text{HTM 的预测结果的误差较大})$$

$$H_1:\mu_D>0$$

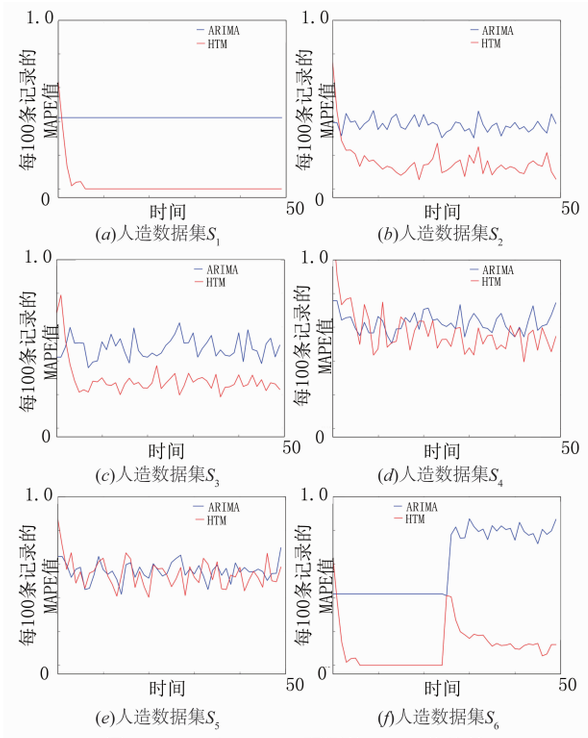


图4 HTM和ARIMA预测的MAPE值比较

取 $t=\frac{\overline{D}}{S/\sqrt{n}}$ 为检验统计量，其中 $\overline{D}$ 为样本均值，S 为数据集的样本方差，$n=49$ 为一个数据集中 D 的样本个数，分别计算数据集 S_1 – S_6 上的 p 值，结果如表 3 所示. 在数据集 S_5 上 p 取得最高值为 0.0525，在其他数据集上则基本为 0，所有数据集上的 p 值都接近或者远小于 0.05，因此我们可以否认原假设 H_0；由此可以确定 HTM 的预测的准确度要高于 ARIMA.

表 3 统计显著性结果

数据集	S_1	S_2	S_3	S_4	S_5	S_6
P	0	0	0	0.000071	0.0525	0

表 4 显示了，HTM 和使用滑动窗口的 ARIMA 在不同的人造数据集上的运行时间. ARIMA 使用了滑动窗口的方法，需要不断的建立模型，以学习新的规律，这也极大的降低了其预测效率；HTM 的效率则明显高于 ARIMA.

表 4 HTM 和 ARIMA 在不同数据集上预测的运行时间

运行时间	S_1	S_2	S_3	S_4	S_5	S_6
ARIMA	126.84	399.16	291.93	235.07	231.08	133.65
HTM	60.56	75.48	121.58	127.91	129.49	67.25

人造数据集上的实验表明 HTM 算法能够准确、快速的对时间序列进行预测，这也为本文所提出的基于层级实时记忆算法的时间序列异常检测算法提供了良好的基础.

6.2 真实数据集

本节在真实数据集上对基于层级实时记忆算法的时间序列异常检测算法进行测试. 选用纽约出租车乘客量数据 D1 和在线广告点击率数据 D2 进行对比实验. 数据集 D1 每 30 分钟采集一次乘客数，共采集了 10320 条已标注数据（数据发布者标明了异常点，如图 5 – D1 中用“○”标明了 4 处异常点）；数据集 D2 是每一个小时采集一次点击率，共采集了 1538 条，该数据集未被标注. 从图 5 和图 6 可以看出，这两组数据都是具有一定周期性的时间序列数据. 实验中本方法将和著名的商业异常检测软件 Skyline 以及 twitter 开源项目 AnomalyDetection 进行对比 AnomalyDetection 是由社交网络公司 twitter 开发的 R 语言工具包，这一项目基于混合季节性 ESD（Seasonal Hybrid ESD，SHESD）算法，能够同时检测出时间序列中的局部异常和全局异常. 实验结果如图 5 和图 6 所示，图中分别给出了数据实际值和三种算法对数据异常程度的评分. 该分值位于 0 到 1 之间，分数越高表明数据的异常程度越高，由于 twitter 的 AnomalyDetection 只能判断数据是否为异常，因此其异常评分非 0 即 1，1 表示异常，0 表示正常.

从图 5 和图 6 我们可以看出：本方法、skyline 和 AnomalyDetection 都能发现数据中明显的异常，即远远超过正常值的数据点，如图 5 和图 6 中箭头 1 所示. 另外，AnomalyDetection 将周末的乘客数的变化视为异常，没有发现一周为一个周期. HTM 能够发现出现周期性错误的数据，而其他两种方法无法做到这一点，如图 5 和图 6 的箭头 2 所示. 对于大量较稳定的数据集 D1，统计学方法 skyline 所给出的异常评分总是趋于一个较低水平，AnomalyDetection 只能给出 0 或 1 这两个值，分数不具有区分度，使得异常的程度区分不明显，预警阈值缺乏选择性. 数据运行初期，本方法给出的分值总是处于一个较高水平，随着程序的运行，其分值不断的降低，这一过程，程序正在不断的学习新的模式，并且做出准确的预测，给出异常评分.

为正确评估比较以上的异常检测算法，我们将实验结果分为 4 类，如表 5 所示.

表 5 算法检测结果分类

实际情况 \ 检测结果	异常	正常
异常	TP（True Positive）	FN（False Negative）
正常	FP（False Positive）	TN（True Negative）

（注：TP：表示异常被正确检测，FN：表示异常被算法误检为正常 FP：表示正常被误检为异常，TN：表示正常被算法视为正常）

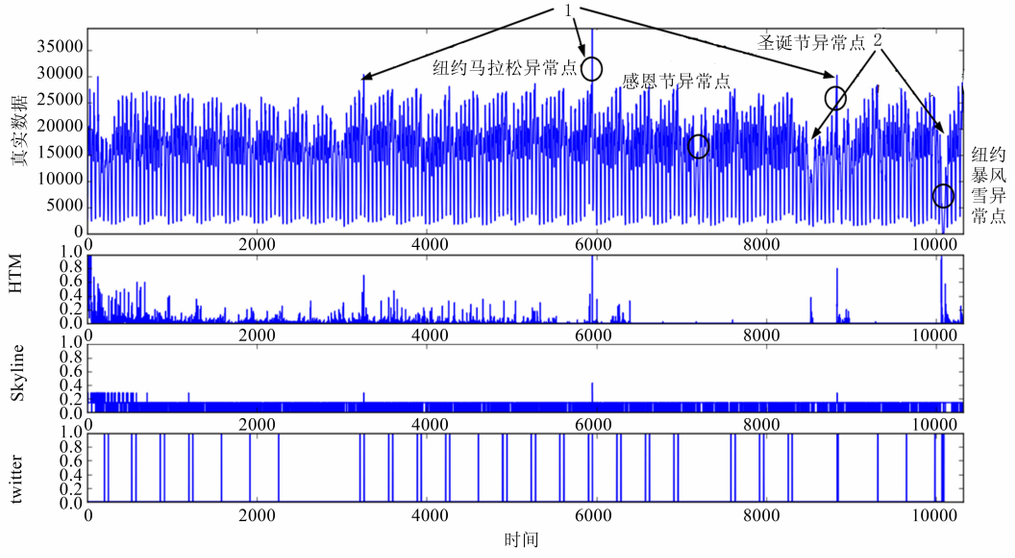


图5　D1：纽约出租车乘客量数据异常检测

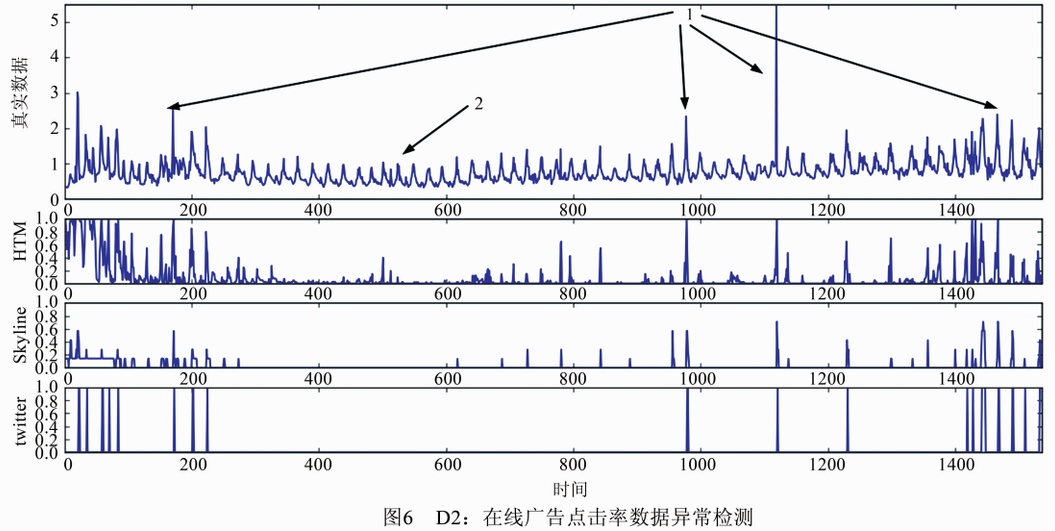


图6　D2：在线广告点击率数据异常检测

按表 5 中的分类对已标注数据集 D1 的检测结果进行统计,统计结果如表 6 所示.

表 6　各算法异常检测结果统计

算法检测结果	TP	FN	FP
HTM	3	1	1
Skyline	0	4	0
twitter	3	1	35

(注:1、由于基于 HTM 的方法需要对数据模式进行学习,该表从 D1 的第 1000 条检测结果开始统计

2、异常判断的阈值默认为 0.5,检测算法的异常评分低于 0.5,则视为检测正常,否则,为检测异常)

由表 6 我们可以看出,对于数据集 D1,HTM 和 twitter 都能较好的检测到其中的异常点,而 twitter 则存在许多误报(FP)的情况;基于数据统计特征和滑动窗口技术的 Skyline 对于 D1 中的异常并不敏感,不能够发现其中的异常,因此漏报现象严重.

人造数据集和真实数据集上的对比结果表明本方法的异常检测效果明显优于其他方法. 本方法能够不断的从数据中学习规律,发现时间序列中的周期性,准确、快速地检测时间序列中的异常,并且对不同程度的异常给出合理的异常评分.

7　结论

随着信息化程度的不断提高,产生了大量时间序列数据,使得异常检测等一系列数据挖掘技术显得异常重要. 传统的异常检测算法很少涉及数据在时间维度上的关联,而近些年的研究,如 LSTM,能够捕捉到数据在时间维度上的信息,却又受制于算法需要进行离

线训练,使得该算法的自适应性不高.针对现有方法的不足,本文提出基于层级实时记忆算法的时间序列异常检测方法.该方法利用 HTM 算法实现对数据模式的存储和预测,其巧妙的结构和简洁的运作模式,使得本方法具有较强的自适应性,能够及时发现多时间序列中潜在模式,以实现对数据异常的有效检测.在后续的研究中我们将丰富数据编码的方式,以实现对更多的不同的类型的数据进行异常检测;同时完善异常检测的评价机制,以提升本方法的准确性和可靠性.

参考文献

[1] Weiss G M. Mining with rarity:a unifying framework[J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1):7-19.

[2] Basu M. Gaussian-based edge-detection methods—a survey [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2002, 32(3):252-260.

[3] Schölkopf B, Williamson RC, Smola AJ, et al. Support vector method for novelty detection[A]. NIPS[C]. ACM, 1999. 582-588.

[4] 程光,等. 基于抽样测量的高速网络实时异常检测模型[J]. 软件学报,2002,13(4):594-599.
CHENG Guang, et al. A real-time anomaly detection model on sampling measurement in a high-speed network[J]. Journal of Software, 2002, 13(4):594-599. (in Chinese)

[5] Etsy. Skyline [CP]. https://github.com/etsy/skyline, 2017.

[6] Esling P, Agon C. Time-series data mining[J]. ACM Computing Surveys (CSUR), 2012, 45(1):12.

[7] Salvador S, Chan P, Brodie J. Learning states and rules for time series anomaly detection[A]. FLAIRS Conference [C]. USA, 2004. 306-311.

[8] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM[J]. Neural Computation, 2000, 12(10):2451-2471.

[9] Hawkins J, Ahmad S. Why neurons have thousands of synapses, a theory of sequence memory in neocortex[J]. arXiv preprint arXiv:1511.00083, 2015.

[10] Hebb D O. The Organization of Behavior: A Neuropsychological Theory[M]. Psychology Press, 2005.

[11] Massey P V, Bashir Z I. Long-term depression: multiple forms and implications for brain function[J]. Trends in Neurosciences, 2007, 30(4):176-184.

[12] Ahmad S, Hawkins J. How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites[J]. arXiv preprint arXiv:1601.00720, 2016.

[13] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model[J]. Neurocomputing, 2003, 50:159-175.

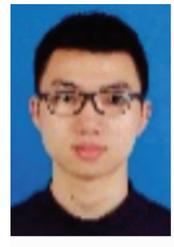
[14] Rajagopalan V, Ray A. Symbolic time series analysis via wavelet-based partitioning[J]. Signal Processing, 2006, 86(11):3309-3320.

[15] Twitter. Anomaly Detection R Package[CP]. https://github.com/twitter/AnomalyDetection, 2017

[16] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553):436-444.

[17] Purdy S. Encoding data for HTM systems[J]. arXiv preprint arXiv:1602.05925, 2016.

作者简介



曾惟如 男,1995 年生于江西赣州,电子科技大学本科生,本科期间发表多篇论文,研究方向为数据挖掘、机器智能.

吴 佳 女,1980 年生于成都.博士研究生,电子科技大学副教授,研究方向为机器智能、数据分析.
E-mail:jiawu@uestc.edu.cn

闫 飞 男,1983 年生于河南省三门峡市.博士,西南交通大学讲师,研究方向为控制理论与控制工程、智能算法、智能交通系统.