一种新的在线流数据异常检测方法

丁智国 莫毓昌 杨 凡
(浙江师范大学数理与信息工程学院 金华 321004)

摘 要 流数据的海量、无限、分布动态变化且不均衡等特征使得对流数据的在线异常检测成为当前一个研究热点。分析了异常数据的少而不同且更容易通过随机空间的分割而孤立出来的特征,基于在线集成学习理论,提出了一种基于隔离森林的在线流数据异常检测算法。在4个UCI标准数据集上的实验结果表明提出的方法有效。

关键词 流数据,异常检测,隔离森林,在线集成学习
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.10.011

Novel Anomaly Detection Method of Online Streaming Data

DING Zhi-guo MO Yu-chang YANG Fan
(College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China)

Abstract Streaming data has some unique characteristics such as massive, unlimited generation, dynanmic variation distribution and unbalanced data distribution and so on, which make the anomaly detection for streaming data become a research hot spot. An obvious characteristic of anomalous sample is "few and different" compared to these normal data, which makes it more easily to be isolated than normal data by the partition of stochastic space. In this paper, a novel online anomaly detection method for streaming data was proposed based on the isolation principle and online ensemble learning theory. Experiments conducted on four UCI datasets demonstrate the effectiveness of the proposed method.

Keywords Streaming data, Anomaly detection, Isolation forest, Online ensemble learning

1 引言

流数据(Streaming Data, SD)通常是指一组顺序、大量、快速、持续到达的数据序列[1]。流数据的产生来源和应用场景有很多,如科学计算、网络安全、财经数据、医疗数据等[2,3]。在这些应用领域中,数据的显著特点就是量大且产生速度快,数据的规模从GB开始,不断向TB和PB级发展。这些获取的数据中大部分是正常的,只代表一些普通的信息,相对于海量的数据来说只有小部分数据(占总量的0.01%到5%不等)附带有更值得关注的信息。比如,信用卡交易中偶然的一次异常记录可能意味着一场交易欺诈;网络中不正常的流量模式有可能预示着网络正在遭受恶意攻击[4];在医学领域,从海量的医学图像中识别出危险病灶[5]也逐渐成为疾病诊断学研究的重要方向。纵观这些应用不难发现,这些异常事件的发生频率同大量的正常事件相比所占的比例非常小,而其蕴含的价值却非常巨大。因此,对这些异常数据进行实时在线的检测、预警、追踪并采取相应的措施,对科学研究和行业应用都具有非常重要的价值和意义[6]。

2 流数据异常检测的挑战和发展状况

在机器学习和数据挖掘研究领域,对流数据的研究通常将其建模为一个随时间延续而无限增长的动态数据集合。对这样的对象进行研究一个不容忽视的事实是:数据流自身的一些显著特点使得对其的研究不同于传统的应用于静态数据集的方法。比如针对流数据持续产生、数据海量且无限的特点,传统的需要存储所有数据的静态、离线异常检测的方法不能胜任流数据处理所需要的"one-pass"特性[7];针对流数据的数据分布动态进化特性[8],很难建立精确预测模型;针对数据分布不均衡的特征[9],常用的基于数据挖掘和机器学习的分类算法很难学到分类边界来建立合适的模型[10,11]。

针对上述困难和挑战,国内外学者针对流数据的异常检测已提出了一些新的思路和方法[12]。现有的文献中对流数据异常检测的研究有两种方案:第一种是对静态方法的改进;第二种是引入增量学习[13]、在线集成学习[14]的概念,从数据集的不同侧面构建相应的检测器。研究结果表明,集成学习能普遍提高学习的泛化性能且在线的集成学习能在一定程度上处理因流数据的动态特性而造成的概念漂移现象[15]。基于上述两种思路,已经出现很多方法[16]。在这些方法中,基于隔离规则[10]和集成学习方法设计了一种专门用来对流数据进行异常检测的算法[7]。该方法的主要优点是在构建初始模型时不需要任何实际的数据,从而能快速构建初始探测模型。它符合数据流的分布特性且在流数据异常检测问题上达

到稿日期:2015-07-12 返修日期:2015-11-12 本文受浙江省计算机科学与技术重中之重学科开放项目(ZC323014100),浙江省自然科学基金项目(Y13F020080)资助。

丁智国(1979—),男,讲师,主要研究方向为数据挖掘、大数据处理,E-mail:dzg_jsj@zjnu.cn;莫毓昌(1980—),男,副教授,主要研究方向为复杂计算系统的可靠性;杨 凡(1957—),男,教授,主要研究方向为图形图像处理。

到了一定的探测精度，然而其不足也很明显，如在探测器更新阶段，无论是否需要，只要当前窗口数据缓冲区满时，都需要对检测器进行强制更新。这种策略虽然能实时应对动态变化，但浪费了计算资源。本文在上述方法的研究基础上，提出了一种基于隔离的在线集成流数据异常检测方法。

3 隔离树定义及基于隔离的异常检测

异常数据的一个显著的特点就是“少且不同”，所以更容易通过对数据空间的随机分割而独立出来[10]。基于隔离的方法的主要思想是通过随机选定的样本属性及其值将样本空间进行随机划分，隔离的过程可以描述成一个建立树结构的过程，对新的样本，基于建立的隔离树求其分割深度，深度值越小，表示越容易被隔离，也就意味着为异常的概率越大；反之则为正常样本。

定义 1(隔离树) 有数据集 X，其幂集为 2^X，如果存在一棵树 T，它是节点的有限集，设 T 的任一节点 N 存在 k 个孩子节点，记为 N_i，$1\leqslant i\leqslant k$ 且存在映射 $f:N\to 2^X$。如果下述 3 个条件成立，则树 T 为数据集的一棵隔离树。

(1) $\forall i,j\in[1,k], i\neq j, M(N_i)\cap M(N_j)=\varnothing$；

(2) $\bigcup_{i=1}^{k} M(N_i)=M(N)$；

(3) $M(N_0)=X$。

其中 $M\in 2^X$，N_i，N_j 表示树 T 中的节点，N_0 表示树 T 的根节点。

上述定义只是一般的定义，如果对其进行特殊化，则得到二叉隔离树定义。

定义 2(二叉隔离树) 令 $k=\{0,2\}$，T 或者为空的二叉隔离树，或者为具有以下特征的二叉隔离树中的一个节点，该节点对应的样本集为 X，$x(x\in X)$ 为一个 d 维样本，则 T 或者为一个无孩子节点的外部节点，或者为一个拥有两个孩子节点 (T_l,T_r) 的内部节点，其中 T_l 和 T_r 分别为其左、右孩子节点。在节点 T，针对其样本集 X，对于数据 x，随机选择其某一属性 q 并随机在其值域空间取值 p，将数据集 X 划分为两个子集，即样本集中数据的 q 属性值小于 p 的值构成的子集形成左孩子样本集 X_l，而 $X-X_l$ 形成右孩子样本集 X_r，且 T_l 和 T_r 也是二叉隔离树。

在实现过程中，给定一个包含 n 个 d 维数据的样本集 $X=\{x_1,x_2,\cdots,x_n\}$，即 $x_i=(x_{i,1},x_{i,2},\cdots,x_{i,d})$，$x_i\in X$，构建二叉隔离树只需要递归地随机选择分割属性 q 和对应的分割值 p，将样本空间 X 进行划分即可，该过程在满足下列预定的条件之一时结束。

(1)树的深度达到预定义的最大值；

(2)节点只有一个样本数据，即 $|X'|=1$，$X'\subseteq X$；

(3)某个节点的数据子集 X' 中的所有样本相同。

上述生成的二叉隔离树具有如下特征：

(1)树中的每个节点或者没有孩子节点，或者有且仅有两个孩子节点。

(2)如果原始样本集 $X(|X|=n)$ 中的每个节点都是可区分的，则该二叉隔离树包含的最大节点个数为 $2*n-1$，且其内部节点个数为 $n-1$，则隔离树的存储空间基于其样本空间线性增长，即空间复杂度为 $O(n)$。

上述方法与现有的基于距离和基于密度的异常检测方法不同，该方法不需要距离或密度计算，且具有线性时间空间复杂度[10]，能适应在线异常检测活动。

4 基于隔离和在线集成学习的异常检测算法

根据 Bagging 的学习策略，样本随机选择，样本集中每个样本被选择的概率为 $1/N$，则样本集中某一个样本被选择 k 次作为训练集的概率为：

$$p(K=k)=C_N^K\left(\frac{1}{N}\right)^k\left(1-\frac{1}{N}\right)^{N-k} \tag{1}$$

该分布满足二项式概率分布。当 $N\to\infty$ 时，即数据量无穷时，每个样本被选择的概率很小，则上述的二项分布近似于泊松分布 $p(K=k)=\frac{\lambda^k}{k!}e^{-\lambda}(\lambda=1)$，即 $p(K=k)=\frac{e^{-1}}{k!}$。这表明可以在线按如下策略执行 Bagging 算法。当有新样本到达时，对于每个学习器个体，基于样本被选择 k 次的概率更新个体学习器。算法描述如下。

算法 1 Online_Bagging(H,d)

输入：H 为$\{h_1,h_2,\cdots,h_M\}$，d 为新样本

输出：$H^*=\{h_1^*,h_2^*,\cdots,h_M^*\}$

1. For 每个基学习器 $h_m(m=1\cdots M)$

2. 根据 Poisson(1)设置 k 值

3. For j=1…k

4. $h_m^*=L_o(h_m,d)$

其中，H 为现有的 M 个个体学习器，而 d 为流数据中新的样本。当异常处理的对象为流数据时，在学习初始阶段，可以基于历史数据采用离线的 Bagging 模式进行学习获得初始的集成探测器模型，随着流数据的到达，采用在线的 Bagging 方法对该探测器模型进行更新。

基于上述分析，本文提出的基于在线集成学习的异常检测算法主要包括 3 个关键阶段。

(1)探测器构建。采用 iForest 算法[10]，基于历史数据构建初始的异常探测器。

(2)在线异常检测。将异常探测器应用于流数据，即对每个到达的数据，判断其异常状况，并基于 Possion(1)分布，判断该样本是否作为更新样本添加到缓冲区。

(3)探测器更新。根据预定的应用规模计算异常率，如果超过了预定义的阈值或样本缓冲区已满，则执行探测器进行更新。

上述的阶段(1)、(3)对应的算法为分别为算法 2(Building_AIForest)和算法 3(Updating_AIForest)。其中初始的基于 iForest 算法构建个体探测器的 iTree 算法可参考文献[10]，采样算法可参考上述的在线集成学习的方法。

4.1 初始探测器构建

基于历史数据，首先学习构建隔离探测器，算法伪代码如下。

算法 2 Building_AIForest(X,N,L)

输入：X 为训练数据集，L 为集成规模(隔离树个数)，N 为采样规模，缓冲区大小

输出：AIForest 为集成隔离树(异常探测器)

1. Initialize AIForest={};

2. set iTree height h=ceiling($\log_2^N$);

3. for i=1 to L do

X′=Sample_with_Replacement (X,N);

AIForest=AIForest∪iTree (X′)

4. return AIForest;

4.2 在线异常探测器更新

在线地对流数据进行异常判断时，如果在指定的规模下，异常率超过了阈值或样本缓冲区已满，则执行探测器更新算法，伪代码如下。

算法 3 Updating_AIForest(X', X_{buffer}, u, u', k, $AIForest$)

输入：X′为当前窗口样本；X_{buffer}为缓冲区样本；u 为异常率阈值；u′为当前窗口的异常率；k 为更新个体；AIForest 为当前集成隔离树（异常探测器）

输出：AIForest′为更新后的集成隔离树（异常探测器）

```
AIForest′={}, X*={};
if |X_buffer|>=M then
   X*=X_buffer
if u′> u then
   X*= X′ ∪X_buffer
for i= 1 to k do
   AIForest′←AIForest′∪iTree (X*);
从 AIForest 中丢弃最旧的 k 个隔离树;
AIForest′←AIForest′∪AIForest;
return AIForest′;
```

在探测阶段，异常检测的活动即对新的观测值 x 计算其被容易隔离的程度，即基于集成隔离树，计算样本 x 的平均隔离深度，平均隔离深度用来表征其异常度，如果该值越小，表明该样本越容易被隔离，其为异常的概率就越大；反之亦然。该值的计算根据其在隔离森林 AIForest 中的平均隔离深度来进行，而对于每棵隔离树 iTree，计算样本 x 从树根到被隔离节点的遍历深度作为其隔离路径长度。样本 x 的最终异常分 $S(x,M)$就是隔离深度的平均值。可以用式(2)表示。

$$S(x,M)=2^{-\frac{E(h(x))}{c(M)}}$$

$$E(h(x))=\frac{1}{N}\sum_{i=1}^{N}h_i(x)$$

$$c(M)=\begin{cases}2H(M-1)-2(M-1)/M, & \text{for } M>2\\ 1, & \text{for } M=2\\ 0, & \text{otherwise}\end{cases} \tag{2}$$

$$H(n)=\ln(n)+0.5772156649$$

其中，$S(x,M)$表示样本 x 的异常分，M 为样本集规模的大小，$E(h(x))$表示平均隔离深度，$h_i(x)(i\in\{1,2,\cdots,N\})$是样本 x 的针对森林中第 i 棵 iTree 树的隔离深度，$c(M)$是由 M 个样本构成的数据集的隔离树深度 $h(x)$的期望值，即 $c(M)$可以用 $h(x)$的平均值来代替，该值用来对 $h(x)$进行正则化处理。$H(M)$是谐波数(harmonic number)或调和数，可以用欧拉常数来估计，即 $H(M)=\ln(M)+0.572156649$(Euler's constant)[10]。当 $E(h(x))\to 0$，即 $S(x,M)\to 1$ 时，表示样本 x 确定为一个异常。当 $E(h(x))>c(M)$，即 $S(x,M)<0.5$ 时，则可以确定认为样本 x 为正常。

5 实验和算法性能分析

5.1 实验数据集和初始参数确定

为了验证所提算法(AIForest)的有效性，采用 4 个 UCI 机器学习库[17]中的真实数据集(见表 1)来对提出的算法进行评估验证。由于 AIForest 属于无监督学习方法，在实验中，类别标签被选用的主要目的是评估算法的性能。在实验中，为了模拟流数据的时序特征，一个合理的设想就是将数据的输入顺序作为流数据的顺序。

表 1 4 个实验数据集

数据集	总样本数	异常数	异常率(%)
HTTP	567498	2211	0.39
SMTP	95156	30	0.03
FORESTCOVER	286048	2747	0.96
SHUTTLE	49097	3511	7.15

针对上述的每个数据集，运行实验 50 次，实验结果取 50 次的平均值，在每次实验中，将数据集的 30%作为历史数据训练集成异常探测器，其余 70%模拟流数据对算法进行验证。ROC 曲线下的面积(Area Under Curve，AUC)通常被用来评估异常检测算法的性能[18]。ROC_{Area} 越接近 1，表示算法性能越好，即 AUC 值越大，异常检测算法的性能越好。

通过对历史数据的学习可知，对 4 个数据集，将集成规模设定为 40，不同采样大小(分别设置 128，256，512 和 1024)的实验结果表明，4 个数据集中，HTTP 和 SHUTTLE 的采样规模为 256，SMTP 和 FORESTCOVER 的采样规模在 512 时都基本达到了性能收敛。因此，在后续的实验中选择采样大小为 256。

5.2 在线算法性能分析

由于流数据分布可能会发生变化，原始的探测器不能或不能很好地胜任新的情况，需要对集成的探测器进行更新，更新算法的 k 值设定如下：

$$k=\begin{cases}\lceil M*\delta\rceil, & u_{xu}<u_0\\ \lceil M*(1-\delta)\rceil, & u_{xu}\geqslant u_0\end{cases} \tag{3}$$

其中，⌈ ⌉表示向上取整操作，M 为集成规模大小。阈值参数 $\delta\in[0,0.3]$，当 $\delta=0$ 即只有当前集成探测器完全不满足要求时才对其进行全部更新。δ 值的大小直接影响到系统的计算开销和系统性能，通常的策略是：如果流数据分布变化不大且在探测精度可接受的范围内，δ 值应该越小越好，因为这样可以避免持续更新所造成的计算资源浪费和由于计算所导致的预测延迟问题；如果流数据分布变化大且为正常的概念漂移情况，应设置较大的 δ 值，这样能及时反映流数据分布变化的情况从而及时更新探测器。然而，由于流数据的未知特性，选择合适的 δ 值是个很困难的事情。现在已经有些研究者对在线探测器的更新问题展开了大量研究，在线的增量式的学习流数据的分布可以为该值的设定提供一些参考[13]。

表 2 列出了集成规模为 40，采样窗口大小设置为 256，参数 δ 分别设置为 0，0.1，0.2 和 0.3 时探测器的性能。其中 $\delta=0$ 表示对探测模型不更新。对于传统的 iForest 算法，选择相同的参数设置进行实验比较，结果如表 2 第 6 列所示。

表 2 4 个数据集的 AUC 性能

数据集	AIFOREST(δ)				iForest
	0	0.1	0.2	0.3	
HTTP	0.7693	0.8910	0.9337	0.9264	0.9987
SMTP	0.5978	0.8072	0.7920	0.7793	0.8910
FORESTCOVER	0.7302	0.8745	0.8981	0.8849	0.9021
SHUTTLE	0.8065	0.9120	0.9260	0.9258	0.9973

从上述的实验结果来看，当 $\delta=0$ 时表示仅仅只根据部分的历史数据学习得到的集成探测器的 AUC 值相对较低。而当 $\delta\neq 0$、取不同的值时，AUC 的值有不同程度的提高。这一方面说明了在流数据环境下，随着样本分布的变化，探测器的更新是必须的；另一方面说明提出的算法具有一定的实用性，对流数据的在线异常探测具有一定的效果。然而，在静态数

(下转第 80 页)

[6] Khurshid A, Zhou W, Caesar M, et al. Veriflow: verifying network-wide invariants in real time[J]. ACM SIGCOMM Computer Communication Review, 2012, 42(4): 467-472

[7] Al-Shaer E, Al-Haj S. FlowChecker: Configuration analysis and verification of federated OpenFlow infrastructures[C]// Proceedings of the 3rd ACM Workshop on Assurable and Usable Security Configuration. ACM, 2010: 37-44

[8] Ni H T. Design and Implementation of the Conformance Test System for OpenFlow Protocol[D]. Beijing: Beijing Jiaotong University, 2015(in Chinese)
倪海铜. OpenFlow 协议一致性测试系统设计与实现[D]. 北京: 北京交通大学, 2015

[9] OpenFlow Switch Consortium. OpenFlow Switch Specification Version1. 3. 4 (2014) [OL]. https://www. opennetworking. org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1. 3. 4. pdf

[10] Lin H M, Zhang W H. Model checking: Theories, techniques and applications[J]. Chinese Journal of Electronics, 2002, 30(S1): 1907-1912(in Chinese)
林惠民, 张文辉. 模型检测: 理论, 方法与应用[J]. 电子学报, 2002, 30(12): 1907-1912

[11] Emerson E A. Temporal and modal logic[M]. MIT Press Cambridge, 1995

[12] Clarke E M, Grumberg O, Peled D. Model checking[M]. MIT Press, 1999

[13] Zhang Y Z. Research on symbolic model-checking algorithm [D]. Jilin: Jilin University, 2009(in Chinese)
张衍志. 符号化模型检测算法的研究[D]. 吉林: 吉林大学, 2009

[14] Granas A, Dugundji J. Fixed point theory [M]. Springer Science & Business Media, 2013

(上接第 65 页)

据集上运行的 iForest 算法的结果可以看作期望值，而在流环境下，每个数据集的 AUC 值却相对较低，这说明在机器学习领域中对整体样本分布的认知程度直接影响着探测器的性能。

其次，从上述 δ 参数在不同值情况下的 AUC 值分析可知，对于不同的数据集，由于样本的进化特征不同，不可能有统一的 δ 值用来指导集成探测器的更新，因此实际应用中需要根据历史数据及其在线学习理论，掌握针对具体应用的流数据产生机理，学习其进化特征。

结束语 本文基于异常数据的本质特征和在线集成学习理论，提出了一种基于隔离森林的在线流数据异常检测方法。4 个标准数据集的实验结果表明，提出的方法能获得较好结果。但需要进一步对该算法中的一些关键参数(如窗口尺寸、集成规模等参数的设置)进行研究以提高算法的性能。其次，在构建隔离树时，属性及其属性值的随机选择没有考虑样本属性值的分布情况，后续将进一步对其展开研究。

参 考 文 献

[1] Gupta M, Gao J, Aggarwal C C, et al. Outlier Detection for Temporal Data: A Survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(9): 2250-2267

[2] Zheng Li-ming, Zou Peng, Jia Yan. How to Extract and Train the Classifier in Traffic Anomaly Detection System[J]. Chinese Journal of Computers, 2012, 35(4): 719-730(in Chinese)
郑黎明, 邹鹏, 贾焰. 网络流量异常检测中分类器的提取与训练方法研究[J]. 计算机学报, 2012, 35(4): 719-730

[3] Shen M X, Liu D R, Shann S H. Outlier detection from vehicle trajectories to discover roaming events[J]. Information Sciences, 2015, 294: 242-254

[4] Lu You, Li Wei, Luo Jun-zhou, et al. A Network User's Abnormal Behavior Detection Approach Based on Selective Collaborative Learning[J]. Chinese Journal of Computers, 2014, 37(1): 28-41(in Chinese)
陆悠, 李伟, 罗军舟, 等. 一种基于选择性协同学习的网络用户异常行为检测方法[J]. 计算机学报, 2014, 37(1): 28-41

[5] Srinivasan U. Anomalies Detection in Healthcare Services[J]. It Professional, 2014, 16(6): 12-15

[6] Zhou Dong-hua, Wei Mu-heng, Si Xiao-sheng. A Survey on Anomaly Detection, Life Prediction and Maintenance Decision for Industrial Processes[J]. Acta Automatica Sinica, 2013, 39(6): 711-722(in Chinese)
周东华, 魏慕恒, 司小胜. 工业过程异常检测、寿命预测与维修决策的研究进展[J]. 自动化学报, 2013, 39(6): 711-722

[7] Tan S C, Ting K M, Liu T F. Fast anomaly detection for streaming data[C]// Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence. AAAI Press, 2011, 2: 1511-1516

[8] Widmer G, Kubat M. Learning in the presence of concept drift and hidden contexts[J]. Machine Learning, 1996, 23(1): 69-101

[9] Li D, Liu S L, Zhang H L. A negative selection algorithm with online adaptive learning under small samples for anomaly detection[J]. Neurocomputing, 2015, 149: 515-525

[10] Liu F T, Ting K M, Zhou Z H. Isolation-Based Anomaly Detection[J]. ACM Transactions on Knowledge Discovery from Data, 2012, 6(1): 1-39

[11] Daneshpazhouh A, Sami A. Entropybased outlier detection using semi-supervised approach with few positive examples[J]. Pattern Recognition Letters, 2014, 49: 77-84

[12] Quinn J A, Sugiyama M. A least-squares approach to anomaly detection in static and sequential data[J]. Pattern Recognition Letters, 2014, 40: 36-40

[13] He H, Chen S, Li K, et al. Incremental learning from stream data[J]. IEEE Transactions on Neural Networks and Learning Systems, 2011, 22(12): 1901-1914

[14] Ando S, Thanomphongphan T, Seki Y, et al. Ensemble anomaly detection from multi-resolution trajectory features [J]. Data Mining and Knowledge Discovery, 2015, 29(1): 39-83

[15] Minku L L, Yao X. DDD: A New Ensemble Approach for Dealing with Concept Drift[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(4): 619-633

[16] Chang W C, Cho C W. Online Boosting for Vehicle Detection [J]. IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics, 2010, 40(3): 892-902

[17] UCI Machine Learning Repository [OL]. http://archive. ics. uci. edu/ml/datasets. html

[18] Hand D J, Tillr J. A simple generalisation of the area under the ROC curve for multiple class classification problems[J]. Machine Learning, 2001, 45(2): 171-186