基于衰减模型的混合属性数据流离群检测

苏晓珂[1] 兰 洋[2] 秦玉明[1] 程耀东[3]
(东华大学信息科学与技术学院 上海 201620)[1] (信阳师范学院计算机与信息技术学院 信阳 464000)[2]
(中国科学院高能物理研究所计算中心 北京 100049)[3]

摘 要 数据流离群检测因内存容量限制和实时检测需求而成为离群检测的一个难点。介绍了一种快速混合属性数据流离群检测算法。在衰减模型下增量聚类数据流,生成代表数据分布的聚类特征集合,半径阈值动态变化;当接收到检测请求时,计算满足条件的每个簇的离群因子,具有高离群因子的簇作为结果输出。同时提出了一种可有效区分离群簇与数据进化初始阶段的方法。算法的时间与空间复杂度同数据流规模近似成线性关系,在真实数据集上的实验结果显示,该算法可有效检测混合属性数据流中的离群点。

关键词 混合属性,数据流,增量聚类,离群检测,衰减模型
中图法分类号 TP391 **文献标识码** A

Outlier Detection Based on the Damped Model in Mixed Data Streams

SU Xiao-ke[1] LAN Yang[2] QIN Yu-ming[1] CHENG Yao-dong[3]
(College of Information Science and Technology, Donghua University, Shanghai 201620, China)[1]
(School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China)[2]
(Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China)[3]

Abstract Outlier detection in data streams poses great challenges due to the limited memory availability and real time detection requirement. A fast outlier detection algorithm in mixed data streams was introduced by clustering the data streams incrementally based on the damped model and generating the cluster features on behalf of the data distribution. The radius threshold value changed dynamically. When detection requirement was received the outlier factor of specified clusters was calculated and the clusters with high outlier factor were taken as the abnormal clusters. At the same time the method is proposed to distinguish between the abnormal cluster and the initial stage of data evolution. The complexity of the time and space were nearly linear with the size of data streams. The experimental results on the KDDCUP99 dataset demonstrate that the method can effectively detect the outliers in mixed data streams.

Keywords Mixed attribute, Data streams, Incremental clustering, Outlier detection, Damped model

1 引言

在网络、金融、保险和电信等众多领域中,识别离群数据比正常数据更有价值,因为它们表示一种偏差的开始,这可能对用户产生危害,造成巨大损失。在这些领域中,大量数据短时间内以流的形式到达,数据维数高、动态变化,但现有离群检测方法绝大部分对中、小规模静态数据集具有很好的性能,却不适用于流数据。

现有离群检测方法用于数据流中时主要存在以下问题:1)算法的时间和空间复杂度较高,不能满足数据流按顺序一趟扫描的要求。2)由于内存容量限制,数据流挖掘中只关注近期到达的数据,但现有离群检测方法对新旧数据同等对待,不做区分。3)现有算法将学习得到的模型作为检测离群点的唯一依据,不能适应数据流环境下数据分布随时间变化的特性。4) 数据流中的数据往往同时包含分类属性与数值属性,但现有算法不能处理混合属性数据流。

针对实际应用,迫切需要采用一种有效的离群检测算法在有限时间内处理大量数据流。本文基于衰减模型,采用"在线增量聚类、离线离群检测"思想,动态维护至多 k 个聚类特征,为混合属性数据流构建新的信息汇总方式,采用时间相关的增量聚类方法进行聚类,计算满足条件的簇的离群因子,根据用户检测需求,得到离群簇集合。算法应用假设为:1)数据流中的正常数据占绝大部分,离群点偏离正常数据并且位于数据分布的边界区域;2)正常数据与离群点聚集在不同类中。

2 相关工作

离群检测可处理的数据流模型分为界标模型、滑动窗口模型和衰减模型。现有数据流挖掘主要是以 CluStream 算法为代表的界标模型聚类算法[1],然而该算法并不适用于衰减模型,因为在线聚类阶段无法及时消除过期数据的影响。

到稿日期:2009-07-20 返修日期:2009-09-13 本文受国家863高技术研究发展计划(2006AA01A120),国家自然科学基金(10871040)资助。
苏晓珂(1979-),女,博士生,CCF会员,主要研究方向为模式识别等,E-mail:suxiaoke07@126.com;兰 洋(1978-),男,硕士生,讲师,主要研究方向为信息管理等;秦玉明(1963-),男,教授,博士生导师,主要研究方向为控制理论与控制工程等;程耀东(1977-),男,博士后,主要研究方向为网格计算技术等。

Aggarwal 等人提出的 HPStream 主要贡献在于引入了一个衰减聚类结构和对数据流进行投影聚类的思想[2]。Cao 等人提出了一种基于密度的聚类算法 DenStream[3]，用以挖掘噪声环境下数据流中任意形状的簇。

文献[4]提出了一种基于 k 均值分区的流数据离群检测算法，该算法先对数据流分区做 k 均值聚类，生成均值参考点集，随后在均值参考点中，根据离群点定义找出可能存在的离群点，严格来讲只是一种分区挖掘方式，并未体现数据流的特点。文献[5]提出了一种基于动态网格划分的数据流离群检测算法，利用动态网格对空间中的稠密和稀疏区域进行划分，对于稀疏区域中的候选离群点，采用近似方法计算其离群度，具有高离群度的数据作为离群点输出。文献[6]提出了一种基于数据流聚类的两阶段入侵检测方法，在线生成数据的统计信息，利用最能反映当前网络行为的统计信息检测入侵行为，该方法本质是数据流上的聚类算法，需要人工标识各簇的类型。以上几种方法均未考虑混合属性数据流，多数算法基于界标模型，无法区分离群簇与数据进化初始阶段。文献[7]提出了分类属性流数据加权频繁模式离群因子，它能动态发现和维护频繁模式，但对数值属性需要预先作离散化处理，若数据流包含多个数值属性，将导致较高的时间复杂度，检测过程同样体现不出数据流的进化特性。

3 相关定义

我们在文献[2]提出的衰减模型下考虑数据流离群检测问题，所有对象的权值随时刻 t 流逝即不断以指数 $f_{(t)}=2^{-\lambda\cdot t}$ 衰减，衰减因子 $\lambda>0$。随着新对象到来，旧对象持续衰减，直至消亡，数据流 DS 被不断更新。具体描述为：根据处理时间和内存空间限制，动态维护至多包含 k 个元素的聚类特征集合 CS。当接收到离群检测请求时，算法计算 CS 中特定簇的离群因子，按离群因子大小降序排列，得到前若干个簇组成的离群簇集合 OS 提供给用户。

假定每个对象具有 m 个属性，分类属性有 m_c 个，数值属性有 m_n 个，$m=m_c+m_n$，D_i 表示第 i 个分类属性。将每个对象看成 m 维空间中的一个点，用距离来度量对象之间的相似度，距离越小的对象越相似。

定义 1 聚类特征 $CF=(aF,lS,w,t_0,t_u,sts)$

aF 是簇 C 分类属性不同取值的绝对频率，$aF=(aF_1,aF_2,\cdots,aF_{m_c})$，$aF_i=\{(a,aF_{C|D_i}(a))\mid a\in D_i\}$，$1\leqslant i\leqslant m_c$；$lS$ 是数值属性的线性和，$lS=(ls_1,ls_2,\cdots,ls_{m_n})$，$ls_i=\sum_{j=1}^{n_C}p_{ji}$，$1\leqslant i\leqslant m_n$，$1\leqslant j\leqslant n_C$，$n_C$ 是 C 包含对象数，随着新对象加入，旧对象消亡，n_C 不断变化；w 指 C 的权重，t_0 指 C 产生时刻，t_u 指 C 更新时刻，sts 是 C 的标志位。

给定当前时刻 T，权重 $w=\sum_{j=1}^{n_C}f(T-t_0^{(j)})$，即 C 中所有对象的权重之和。取值 a 在 C 中的绝对频率 $aF_{C|D_i}(a)$ 指 C 在 D_i 上的投影 $C|D_i$ 包含 a 的次数 $o_{(C|D_i=a)}$，即 $aF_{C|D_i}(a)=o_{(C|D_i=a)}$，$0\leqslant aF_{C|D_i}(a)\leqslant o_{C|D_i}$。

标志位 sts 有 3 种状态：给定权重阈值 ε，若 $w>\varepsilon$，簇 C 标记为正常簇，$sts=0$；若 $w\leqslant\varepsilon$，C 标记为候选离群簇，$sts=1$。候选离群簇包括真正的离群簇 $sts=2$、数据进化初始阶段及正常簇的边界簇。

性质 1 CF 具有可加性。

两个簇 C_1 与 C_2 合并后形成新簇，用 $C_1\cup C_2$ 表示，聚类特征更新为：

$$CF_{(C_1\cup C_2)}\cdot aF_{(C_1\cup C_2)|D_i}(a)=CF_{C_1}\cdot aF_{C_1|D_i}(a)+CF_{C_2}\cdot aF_{C_2|D_i}(a) \tag{1}$$

$$CF_{(C_1\cup C_2)}\cdot ls_i=CF_{C_1}\cdot ls_i+CF_{C_2}\cdot ls_i \tag{2}$$

$$CF_{(C_1\cup C_2)}\cdot t_0=\min(CF_{C_1}\cdot t_0,CF_{C_2}\cdot t_0) \tag{3}$$

$$CF_{(C_1\cup C_2)}\cdot t_u=T \tag{4}$$

$$CF_{(C_1\cup C_2)}\cdot w=CF_{C_1}\cdot w+CF_{C_2}\cdot w \tag{5}$$

$$CF_{(C_1\cup C_2)}\cdot sts=\begin{cases}1 & CF_{(C_1\cup C_2)}\cdot w\leqslant\varepsilon\\ 0 & CF_{(C_1\cup C_2)}\cdot w>\varepsilon\end{cases} \tag{6}$$

性质 2 若时间间隔 δt 内，簇 C 中没有新对象加入，则 C 的聚类特征衰减为：$CF=(2^{-\lambda\cdot\delta t}\cdot aF,2^{-\lambda\cdot\delta t}\cdot lS,2^{-\lambda\cdot\delta t}\cdot w,t_0,t_u,sts)$。

此处仅证明权重 w 的衰减，aF，lS 类似。

证明：C 在时刻 t 的权重用 w_t 表示，由 $w_t=\sum_{j=1}^{n_C}f(t-t_0)=\sum_{j=1}^{n_C}2^{-\lambda\cdot(t-t_0)}$，可得 $w_{(t+\delta t)}=\sum_{j=1}^{n_C}f(t+\delta t-t_0)=\sum_{j=1}^{n_C}2^{-\lambda\cdot(t+\delta t-t_0)}=2^{-\lambda\cdot\delta t}\cdot\sum_{j=1}^{n_C}2^{-\lambda\cdot(t-t_0)}=2^{-\lambda\cdot\delta t}\cdot w_t$，由此得证。

将文献[8]中簇间距离做时间维上的扩展，使之能够运用到数据流中。

定义 2 簇间距离 $d(C_1,C_2)$

簇 C_1 与 C_2 间的距离 $d(C_1,C_2)=\dfrac{\sum_{i=1}^{m}dif(C_i^{(1)},C_i^{(2)})}{m}$，即在 m 个属性上距离和的平均值，$dif(C_i^{(1)},C_i^{(2)})$ 为 C_1 中所有对象与 C_2 中所有对象关于第 i 个属性的距离。对于第 i 个数值属性 $dif(C_i^{(1)},C_i^{(2)})=\left|\dfrac{ls_i^{(1)}}{w_1}-\dfrac{ls_i^{(2)}}{w_2}\right|$；对于分类属性 D_i，

$$\begin{aligned}dif(C_i^{(1)},C_i^{(2)})&=1-\frac{1}{w_1\cdot w_2}\sum_{a\in C_1}aF_{C_1|D_i}(a)\cdot aF_{C_2|D_i}(a)\\&=1-\frac{1}{w_1\cdot w_2}\sum_{a\in C_2}aF_{C_1|D_i}(a)\cdot aF_{C_2|D_i}(a)\end{aligned} \tag{7}$$

推论 1 当簇 C_1 中仅含一个对象 p 时，由 $d(C_1,C_2)$ 可得对象与簇的距离 $d(p,C)=\dfrac{\sum_{i=1}^{m}dif(p_i,C_i)}{m}$，$dif(p_i,C_i)$ 指 p 与 C 中所有对象在第 i 个属性上的距离。对第 i 个数值属性 $dif(p_i,C_i)=\left|p_i-\dfrac{ls_i}{w}\right|$；对于 D_i，

$$dif(p_i,C_i)=1-\frac{aF_{C|D_i}(a)}{w} \tag{8}$$

假定在接收到离群检测请求时，CS 中包含 k 个簇，为区分离群簇与数据进化初始阶段，给定时刻阈值 θ，仅考察满足条件 $\{C_i\mid(CF_{C_i}\cdot sts=1)\&(T-CF_{C_i}\cdot t_0>\theta),1\leqslant i\leqslant k\}$ 的簇的离群因子，其中 T 为当前时刻。

定义 3（簇 C 的离群因子） 簇 C 的离群因子定义为

$$OF(C)=\frac{1}{(k-1)}\sum_{j=1}^{k}d(C,C_j) \tag{9}$$

$OF(C)$ 度量了簇 C 偏离其它 $k-1$ 个簇的程度，其值越大，说明 C 偏离整体越远，离群程度越高[8]。增量聚类阶段生成类球形簇，可以将一个大的正常簇分成若干个小的候选离群簇。分布在数据流边界的离群簇的离群因子要高于小规

模正常簇的离群因子。

数据进化初始阶段表现为离群簇，当前检测阶段内，不能有效区分两者，随着数据的不断流入，数据进化初始阶段形成的簇不断吸收新数据，权重增加幅度大于对象衰减幅度，当$w>\varepsilon$时形成正常簇，而真正的离群簇因为没有新数据或仅有少量新数据加入，权重逐渐减小，直至最终消亡。因此在条件$\{C_i|(CF_{C_i}\cdot sts=1)\&(T-CF_{C_i}\cdot t_0>\theta),1\leqslant i\leqslant k\}$下，能够区分离群簇与数据进化初始阶段。

4 算法

算法分为在线聚类和离线离群检测两个阶段，在线部分增量聚类数据流中的对象，动态维护至多 k 个 CF，以保证有效利用有限的内存空间。当接收到离群检测请求时，计算满足$\{C_i|(CF_{C_i}\cdot sts=1)\&(T-CF_{C_i}\cdot t_0>\theta),1\leqslant i\leqslant k\}$条件的簇的离群因子，检测出真正的离群簇。

4.1 在线聚类

输入：顺序到达的数据流 DS，初始半径阈值 s，簇数最大值 k

输出：簇集合 CS

1：离线初始化 CS

2：while $|DS|\neq 0$ do

3：　从 DS 读入数据 p

4：　计算 $d(p,C_i),C_i\in CS$

5：　$d_{\min}=\min\{d(p,C_i)|0<i<k\}$

6：　if $d_{\min}\leqslant s$ then

7：　$C_n=C_n+p$(其中 $d(p,C_n)=d_{\min}$)

8：　else

9：　$CS=CS\cup\{p\}$

10：　if $|CS|>k$ then

11：　$l=0$

12：　while $l<|CS|$ do

13：　从 CS 读入簇 C_l

14：　if C_l 已过期 then

15：　$CS=CS-C_l$

16：　endif;

17：　l++

18：　endwhile;

19：　if 不存在过期簇 then

20：　$d_{mc}=\min\{d(C_i,C_j)|C_i,C_j\in CS\}$

21：　$C_i=C_i\cup C_j$

22：　$CS=CS-C_j$

23：　if $d_{mc}>s$ then

24：　$s=d_{mc}$

25：　endif;

26：　endif;

27：　endif;

28：　$DS=DS-\{p\}$

29：endwhile;

第 1 步，离线初始化数据流的前若干个对象。初始时，簇集合 CS 为空，从 DS 读入一个新的对象，以此构造一个新的类。读入下一新对象 p，计算它与每个已有类间的距离，选择最小的距离，若最小距离超过给定的半径阈值 s，以 p 构造一个新的类，继续读入下一对象，否则将 p 并入具有最小距离的类中，重复读入过程，直到离线初始化完毕。

第 7 步添加新对象是性质 1 的特例。对新到达的对象 p，根据推论 1 计算 p 与簇的距离，令 $d(p,C_n)=\min\{d(p,C_i)|0<i<k\}$，$C_n$ 为距离 p 最近的簇。当 $d(p,C_n)\leqslant s$ 时，将 p 添加到 C_n 中，利用可加性更新 C_n 的聚类特征。将 p 的分类属性 D_i 取值 a 与 C_n 中 aF_i 比较，若 aF_i 中有相同的取值，则 $CF_n\cdot aF_{C|D_i}(a)=CF_n\cdot aF_{C|D_i}(a)+1$，否则 $CF_n\cdot aF_{C|D_i}(a)=1$；$CF_n\cdot ls_i=CF_n\cdot ls_i+p_i$，$CF_n\cdot t_u=T$，$T$ 为当前时刻，$CF_n\cdot w=CF_n\cdot w+1$，若 $CF_n\cdot w>\varepsilon$，则 $CF_n\cdot sts=0$。

第 9 步增加新簇，当 $d(p,C_n)>s$ 时，将 p 作为一个新簇，添加到簇集合 CS 中。$CF_p\cdot aF_{C|D_i}(a)=1$，$CF_p\cdot ls_i=p_i$，$CF_p\cdot t_0=T$，$CF_p\cdot t_u=T$，$CF_p\cdot w=1$，$CF_p\cdot sts=1$。

第 14 步，若 CS 中的簇数超过 k，考察 CS 中是否存在过期簇，若存在，将其删除。当簇数达到上限时，对满足$\{C_i|(CF_{C_i}\cdot sts=0)||(CF_{C_i}\cdot sts=2),1\leqslant i\leqslant k\}$条件的簇进行删除判断，采用文献[9]提出的数据到达时间近似为泊松过程，即“到达时间间隔服从指数分布”的思想，若簇 C_i 满足 $T-CF_{C_I}\cdot t_u>\xi\cdot(CF_{C_I}\cdot t_u-CF_{C_I}\cdot t_0)/CF_{C_I}\cdot w$，则将 C_i 从 CS 中删除。

第 19 步，若 CS 中不存在过期簇，则根据定义 2 计算任意两个簇间的距离，由性质 1，合并距离最近的两个簇，若最小距离超过 s，则用此值替换 s。由于簇一旦被合并将无法拆分，因此仅当不存在过期簇时才考虑合并操作。

4.2 离线离群检测

输入：此阶段的簇集合 CS、时刻阈值 θ、离群簇比例 Q

输出：此阶段的离群簇集合 OS

1：if 收到检测请求 then

2：　$i=0$

3：　while $i<|CS|$ do

4：　从 CS 中读入 C_i

5：　if $(CF_{C_i}\cdot sts=1)\&(T-CF_{C_i}\cdot t_0>\theta)$

6：　计算 C_i 的离群因子（据定义 3）

7：　endif;

8：　i++

9：　endwhile;

10：　将 CS 按离群因子降序排列

11：　$i=0$

12：　while $i<k\cdot Q\%$ do

13：　$OS=OS\cup C_i$

14：　i++

15：　endwhile;

16：endif;

收到检测请求后，计算当前维护的簇集合 CS 中满足条件$\{C_i|(CF_{C_i}\cdot sts=1)\&(T-CF_{C_i}\cdot t_0>\theta),1\leqslant i\leqslant k\}$的簇到其他所有簇的距离和，根据定义 3，计算其的离群因子。将 CS 中的簇按离群因子降序排列，排在前面的 $k\cdot Q\%$ 个簇即为检测到的离群簇，其相应的 sts 设为 2。

4.3 复杂度分析

在线增量聚类阶段的时间和空间复杂度依赖于此阶段数据流中的对象数 N、对象包含的属性数 m、产生 CF 的个数 k，时间复杂度为 $O(m\cdot N\cdot k)$；内存中只需维护 k 个聚类特征，假定每个分类属性有 n_i 个不同取值，因此空间复杂度为 $O(k\cdot(\sum_{i=1}^{m_C}n_i+m_N))$。离线离群检测阶段，需计算满足条件的簇到其他 $k-1$ 个簇的距离和并排序，假定满足条件的簇有 l 个，$1\leqslant l\leqslant k$，则时间复杂度为 $O(m\cdot l\cdot k)$，由于 $k<<N$，因此整

体时间复杂度为 $O(m \cdot N \cdot k)$，与属性个数、对象个数以及簇个数成近似线性关系。

5 实验

实验环境为：2.2GHz Intel Pentium IV 处理器，512MB 内存，Windows XP professional 操作系统，编程语言使用 Visual C++ 6.0。评估采用文献[8]中的检测率 *DR* 和误报率 *FR*，检测率表示被正确检测的离群记录数占整个离群记录数的比例；误报率表示正常记录被检测为离群点的记录数占整个正常记录数的比例。

实验使用的 KDDCUP99 数据集中的记录分为 5 大类，包括 Normal，DOS，U2R，R2L 和 Probing，后面 4 种作为离群记录。对此数据集作适当修改，使离群记录只占数据集的 2%。取 400000 条记录模拟数据流，离群记录有 8022 条，其中 DOS 有 7868 条，Probing 有 85 条，U2R 有 52 条，R2L 有 17 条，每段检测间隔内包含大致相同比例的离群点。离线初始化阶段含 2000 条记录，数据流速为 2000 条记录/时间单位，聚类结果按离群因子降序排列，前 33%作为检测到的离群簇。

5.1 相关算法对比

选取文献[7]提出的 FODFP-Stream、文献[10]提出的 FindFPOF 和文献[11]提出的 EvolutionaryOutlierSearch 作为对比算法。这些算法都能处理大规模数据集，但后两者仅针对静态数据集，不存在时间维度的概念，因此在整个数据集上进行测试，对比结果如图 1 所示。

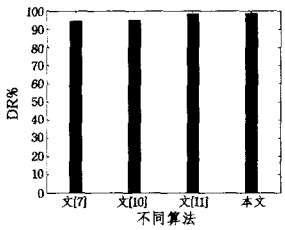

图 1 相关算法检测率对比

从图 1 可看出，在 4 种算法中，本文的检测率最高，接近 100%。

5.2 数据流检测结果

令 $t1=100000$，$t2=200000$，$t3=300000$ 及 $t4=400000$，得到的检测结果如表 1 所列。

表 1 不同阶段的检测结果

阶段	FR	DR				
		total	DOS	U2R	R2L	Probing
t1	1.44%	98.84%	99.90%	0.00%	0.00%	90.91%
t2	1.46%	99.52%	99.82%	16.67%	0.00%	92.68%
t3	0.46%	99.49%	99.68%	0.00%	0.00%	90.32%
t4	3.15%	99.31%	99.58%	0.00%	50.00%	86.25%

从表 1 可看出，4 个阶段整体检测率都达到 98.00%以上，$t4$ 阶段误报率最高，为 3.15%，原因在于 $t4$ 阶段作为模拟数据流最后阶段，没有后续记录对离群簇和数据进化初始阶段做出区分。在 4 种类型的离群记录中，算法对 DOS 的检测效果最好，Probing 次之，R2L 和 U2R 效果最差。这是因为在模拟数据流中，高达 98.08%的离群记录为 DOS，由 DOS 形成的簇在检测中压倒多数，使得部分 Probing，R2L 和 U2R 混杂在 DOS 或正常簇中。实际数据流环境中，每种离群记录比例大致均衡，算法对各类离群记录的检测结果也会有一定程度的改善。

5.3 衰减因子 λ 变化

当 λ 分别取 0.15，0.25，5 时，实验结果如图 2 所示。

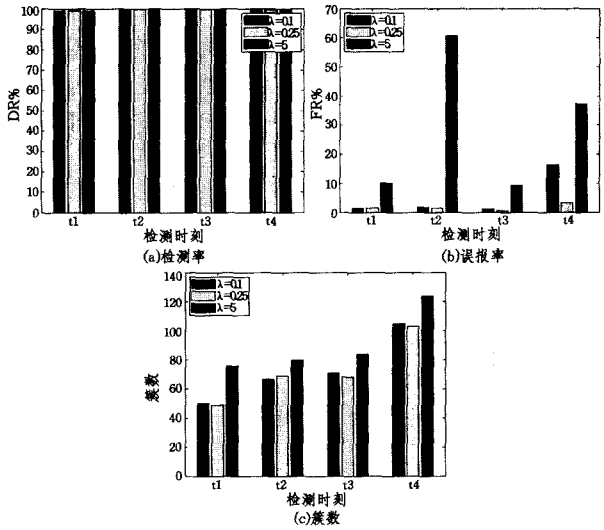

图 2 衰减因子变化

λ 取不同值时，图 2 中(a)显示检测率变化不大，都接近 100%，(b)表明误报率受影响较大。λ 越大，每条记录衰减得越快，对后期记录影响越小，取相同比例的簇作为离群簇时，检测率变化不大，但误报率增加很多。λ 越小，当前阶段的检测结果对后期阶段影响较大，误判为离群簇的正常簇在下一阶段可能仍被误判，导致误报率仍然较高。实验结果表明，$\lambda=0.25$ 时，检测结果较理想，具有高的检测率和低的误报率。

由图 2 中(c)可看出每个阶段生成的簇个数与误报率呈相同变化趋势，λ 越大，*CS* 中的簇衰减越快，过期簇越多，对新记录的影响越小，形成的新簇个数越多。

5.4 离群簇比例 Q 变化

当 Q 分别取 13，33，50 时，实验结果如图 3 所示。

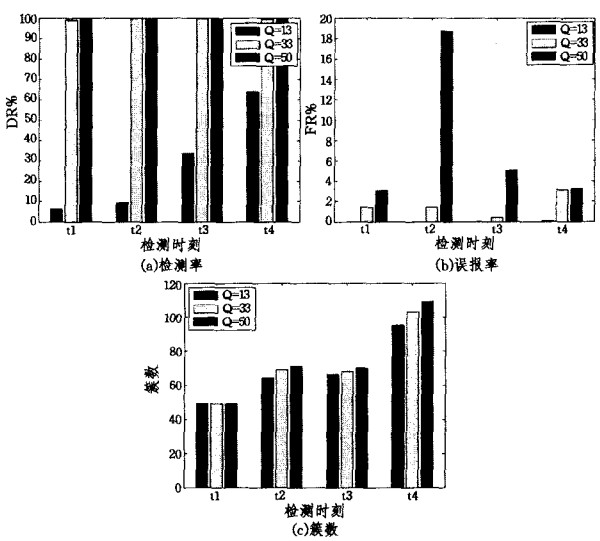

图 3 离群簇比例变化

由图 3 中(a)、(b)可看出 $Q=13$ 时检测率与误报率都很低，$Q=33$ 时的检测率与 $Q=50$ 时相差不大，但 $Q=33$ 时的误报率明显低于 $Q=50$ 时的误报率，原因在于离群簇比例越高，检测到的离群记录越多，检测率越高，但越有可能将正常

簇误判为离群簇，误报率相应也就越高。图3中(c)表明每阶段簇个数受Q变化影响不大。随着数据的不断流入，生成的簇个数整体呈上升趋势。

结束语 数据流离群检测不同于静态数据集的离群检测，因检测对象具有动态性、不可复读性、数据量大等特点而成为离群检测的一个难点，引起了许多研究人员的关注。本文探讨了混合属性数据流离群检测问题，提出了基于衰减模型的聚类特征结构，用以近似估计数据流的分布状况。同时提出了数据流中簇的离群因子定义，通过计算特定簇的离群因子，可得到离群簇集合，作为结果提交给用户，合理区分了离群簇与数据进化初始阶段，改进了数据流离群检测的质量。实验结果表明，本文提出的算法是有效的，检测性能优于其它相关算法。进一步提高算法的实现效率，将其扩展到更一般的数据流模型，以及与数据流中其他相关的数据挖掘算法进行结合，是下一步的研究方向。

参 考 文 献

[1] Aggarwal C C, Han Jia-wei, Wang Jian-yong, et al. A Framework for Clustering Evolving Data Streams[C]// Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, 2003: 81-92

[2] Aggarwal C C, Han Jia-wei, Wang Jian-yong, et al. A Framework for Projected Clustering of High Dimensional Data Streams[C]// Proceedings of the 30th International Conference on Very Large Data Bases. Toronto, 2004: 852-863

[3] Cao Feng, Ester M, Qian Wei-ning, et al. Density-based Clustering over an Evolving Data Stream with Noise[C]// Proceedings of the 6th SIAM International Conference on Data Mining. Bethesda, 2006: 326-337

[4] 倪巍伟，陆介平，陈耿，等. 基于k均值分区的数据流离群点检测算法[J]. 计算机研究与发展，2006，43 (9)：1639-1643

[5] 杨宜东，孙志挥，朱玉全，等. 基于动态网格的数据流离群点快速检测算法[J]. 软件学报，2006，17(8)：1796-1803

[6] 俞研，郭山清，黄皓. 基于数据流的异常入侵检测[J]. 计算机科学，2007，34(5)：66-71

[7] 周晓云，孙志挥，张柏礼，等. 高维类别属性数据流离群点快速检测算法[J]. 软件学报，2007，18(4)：933-942

[8] Jiang Sheng-Yi, Song Xiao-Yu. A Clustering-based Method for Unsupervised Intrusion Detections[J]. Pattern Recognition Letters, 2006, 27(5): 802-810

[9] 杨春宇，周杰. 一种混合属性数据流聚类算法[J]. 计算机学报，2007，30(8)：1364-1371

[10] He Zeng-you, Xu Xiao-fei, Huang Zhe-xue, et al. FP-Outlier: Frequent Pattern Based Outlier Detection[J]. Computer Science and Information System, 2005, 2(1): 103-118

[11] Aggarwal C, Yu P. An Effective and Efficient Algorithm for High-dimensional Outlier Detection[J]. The VLDB Journal, 2005, 14(2): 211-221

(上接第126页)

学方法尽量减少安全性分析的随意性和不确定性，实现了软件安全性的分析和评估，并给出软件安全等级的置信度水平。随着分析人员的估计次数和人数的增加，置信度水平会更接近于实际的软件等级。

但是，该方法未对软件安全性的保障技术以及构件失效概率等问题进行探讨。下一步需要结合具体应用背景，对复杂软件的安全性分析做更深入研究，并对计算模型和综合算法做一定程度的优化和改进。

参 考 文 献

[1] John C K. Safety-Critical System: Challenges and Directions [C]// Proceedings of the 24th International Conference on Software Engineering. May 2002: 547-550

[2] Wang J. A Subjective Methodology for Safety Analysis of Safety Requirements Specifications [J]. IEEE Transactions on Fuzzy Systems, 1997, 5(3): 418-430

[3] Dempster A P. A generalization of Bayesian inference(with discussion)[J]. Journal of the Royal Statistical Society Series B, 1968, 30(2): 205-247

[4] Shafer G. A Mathematical Theory of Evidence[M]. Princeton: Princeton University Press, 1976

[5] Atkinson C, Bunse C, Gross H-G, et al. Component-based Software Development for Embedded Systems[M]. Berlin Heidelberg, Germany: Springer-Verlag, 2005

[6] Schmucker K J. Fuzzy sets, Natural Language Computations and Risk Analysis [M]. Rockville, MD: Computer Science Press, 1984

[7] DO-178B. Software Considerations in Airborne Systems and Equipment Certification[S]. RTCA/EUROCAE, December 1992

[8] MIL-STD-882C. System Safety Program Requirements[S]. Department of Defense. USA Military Standard, 1993

[9] Karwowski M. Potential Applications of Fuzzy Sets in Industrial Safety Engineering [J]. Fuzzy Sets and Systems, 1986, 19: 105-120

[10] Liu J, Yang J B, Wang J, et al. Safety analysis and synthesis using fuzzy rule-based evidential reasoning approach [C] // the 2003 UK Workshop on Computational Intelligence. University of Bristol, September 2003

[11] Zadeh L A. Fuzzy Sets, Information and Control[M]. 1965: 338-353

[12] Herrera F, Martinez L. A 2-tuple fuzzy linguistic representation model for computing with words [J]. IEEE Transactions on Fuzzy Systems, 2000, 8(6)

[13] Bowles J B, Pelaez C E. Fuzzy logic Prioritization of failures in a system failure mode, effects and criticality analysis[J]. Reliability Engineering and System Safety, 1995, 50: 203-213

[14] Anderson L. The theory of possibility and fuzzy sets: new ideas for risk analysis and decision making[M]. Swedish Council for Building Research, 1988: 165-167

[15] Zimmerman H J. Fuzzy Set Theory and Its Application[M]. Norwell, MA: Kluwer, 1991

[16] 张锦. 三余度飞控计算机系统软件的研究与设计[D]. 西安：西北工业大学，2006：7-16