使用日志的异常检测①

陈海宇，曾德胜

(罗定职业技术学院 电子信息系，罗定 527200)

摘 要：提出了使用日志的孤立点分析方法，对日志数据进行预处理，确立合适的挖掘粒度，刻画出正常模式。改进的方法可对规模较大的数据集进行异常检测时，在降低误报率的同时，大大提高了检测率，并达到理想的时间效率。使系统定期分析用户日志，从其自动找到可疑的日志，及时预防或者处理非法操作的现象，提高检测系统的智能化、准确性和检测效率。

关键词：日志；数据挖掘；分类模型；孤立点；高维数据

Anomaly Detection on the Use of Log

CHEN Hai-Yu, ZENG De-Sheng

(Electronic Information Department, Luoding Vocational Technical College, Luoding 527200, China)

Abstract: The use of log analysis of the outlier was proposed, on the log data preprocessing to establish the appropriate mining size, with depicting a normal mode. The improved method can be used for the large-scale anomalous detection of data sets, reducing the false alarm rate, while greatly improving the detection rate to achieve the desired time efficiency. The system can be with the regular analysis of the user logs, to automatically find the suspect from the log, in a timely manner to prevent or deal with the illegal operation, in order to make the detection system more intelligent, accurate and efficient.

Key words: log; data mining; classification model; outlier; high-dimensional

1 背景

为了维护系统资源的运行状况，每个系统都会有相应的日志记录系统，记录着有关日常用户操作的各种信息，并可从记录的日志中抽取出各种正常以及异常的操作如：误操作警报的日期、时间戳信息等。当发现系统的异常症状后不容易或不能找到非法用户对系统造成的伤害。针对以上的行为，目前常用的方法就是在系统中增加日志功能，记录所有用户的状态信息以及对系统所做的操作。由于日志数据十分庞大，某些异常行为难以被直接发现，地域信息已经不再是发现可疑信息的线索。孤立点检测的任务就是从大量的复杂的数据集中发现小部分异常数据所隐含的、与常规数据模式不同的数据模式[1]。基于系统日志使用数据挖掘中异常点分析的技术，就可以从海量日志中得到正常和异常的行为模式。该方法首先对日志数据进行预处理，确定合适的挖掘粒度，得到以用户使用会话为单位的日志数据，使用统计的方法，统计出可以表现用户行为特征的属性值，使用多种聚类算法对不同的特性值进行实验，找到对于聚类算法效果最好的特征值组，最后建立聚类模型。

2 相关研究及技术分析

日志就是程序运行时为了记录当前状态而产生信息，它由软件设计者根据具体需求设置具体内容，如可以记录错误，程序当前状态等，用户访问信息等。模型检测技术主要有误用检测和异常检测，异常检测是指使用已建立的正常模式，通过判断当前模式与正常模式之间的偏差来识别入侵[2]。主机的异常检测是要检测主机或系统的数据，它对系统的审计日志、系统的行为数据或受保护系统的文件系统等进行分析，

① 收稿时间:2010-12-21;收到修改稿时间:2011-02-14

发现异常或越权行为，从而引起系统管理员的注意。异常检测是通过统计正常情况下的用户行为规律，运用预测模式生成技术，生成用户每种行为下一步的行为概率，若某种正常情况下概率极小的行为发生，认为该用户有异常行为，实质上是对系统运行日志进行挖掘。用户行为异常检测的关键在于分类模型的建立，而分类模型的准确度与覆盖度取决于分类特征的选择。目前有关日志挖掘的研究主要有：软件数据挖掘，web 使用挖掘，工作流挖掘等。

2.1 软件挖掘

软件挖掘就是对软件生命期中产生的数据使用数据挖掘的方法找到有趣的模式从而辅助软件的各个阶段的需求。软件数据挖掘对运行时系统数据的研究从其目的上分为：系统完善与重构；源代码潜在错误检测，逆向工程等。其中主要研究有：S. Breu 分析程序产生的日志信息，寻找被重复执行的函数，从而找到程序的切面。在挖掘方向中，分析程序日志是一个主要的研究方法。Liu 在他的文章中记录测试程序函数调用的日志，然后绘制出函数调用的频繁图，在图中抽出主要特性构造出基于 svm 的分类器，来检测异常程序。El-Ramly 使用遗留系统中嵌入特定日志记录功能，使用频繁序列挖掘的方法找到了用户使用老系统的交互模式，然后根据找到的模式在新系统开发中用于自动生成用户的图形界面。本文的研究工作也属于软件挖掘，其目的就是增强系统在执行阶段的安全性。是数据挖掘技术在软件领域的一次应用。

2.2 Web 使用挖掘

Web 使用挖掘就是利用数据挖掘技术对 Web 日志信息进行分析从而找到浏览者的使用模式。可用于分析网站流量模式，发现系统性能瓶颈，优化站点结构、提高站点效率，提高用户访问的有效性，发现用户的需要和兴趣等。但是传统的 Web 日志很难还原会话状态这一信息，虽然 Tanasa 提出了一个方法来还原用户会话，以角色用户行为模式分析对传统的 Web 日志挖掘还是很困难的，主要的研究有 Web 服务器性能改进，包括：页面缓存，预读取，交换；定制访问者会话服务，分类浏览者等，主要使用数据挖掘的方法是：基于数据结构 WAS 树的算法，频繁模式和分类[3,4]。

2.3 工作流挖掘

工作流挖掘的大体流程是：各种事务信息系统，例如客户关系管理(CRM)，企业资源计划(ERP)运行过程中都产生一些日志数据，为了对不同系统异构的日志数据进行挖掘，需要将这些日志数据统一转换成 XML 格式，然后将日志文件数据储存到日志数据仓库中。通过各种流程挖掘技术或者工具对数据仓库中的日志数据进行处理，挖掘出实际运行的流程模型，然后将其与期望的流程模型或者预先设计好的流程模型进行一致性测试，并且将测试的结果用以改进流程设计。工作流挖掘中的日志也是要求还原出用户的会话状态来追踪其流程的步骤。

系统日志就是系统运行时状态的记录，充分利用系统日志，可以最大程度地对潜在的恶意操作做出记录和预测，而日志挖掘就是利用数据挖掘方法通过分析系统运行时状态寻找系统运行时的有趣模式。本文综合以上的研究与技术，深入分析日志数据，通过对日志的合适处理，并通过统计建日志数据模型，采用基于密度的聚类方法，可以过滤“噪声”数据，发现任意形状的簇。

3 异常行为模型的建立

3.1 日志数据预处理

预处理过程经过四个步骤：数据净化、用户识别、会话识别和事务识别。会话是日志数据挖掘中一些常被采用的粒度，在构造日志函数时，主要是考虑到不同用户的会话状态和用户在同页面对数据库执行的操作。在原有模型的基础上给运行系统引入专门的日志切面，其关注点是系统中业务逻辑层中的每个函数。系统及网络中的日志内容是由干若特征属性描述的多个事件的数据对象 E 的集合，$E=\{e_1,e_2,...,e_n\}$ 如：$\text{session}=\{\text{session}_1,session_2,...session_n\}$，$n$ 为会话中日志的数目。集合 E 中的每个对象是由 m 个特征属性 $(F_1,F_2,...,F_m)$ 描述，即会话定义如下：$session=\{\{T_i,SID_i,UserType_i,Page_i,Fun_i\},i\in m\}$，因此，集合 E 中任一个事件 e_i（即 $session_i$）可表示为一个 m 维空间的特征向量（$f_{i1},f_{i2},...,f_{im}$），其中 f_{ij} 为特征 F_i 的一个具体值，如(T1, S1, USER_ADMIN, P1, C1.F1)等。得到如表 1 所示形式的日志内容：

如表 2 中所示，其中 SID 用来追踪用户的会话长度，对于 C/S 模式，这个属性也是需要的。有时，数据日志并不能反应用户的真实行为，用户在使用系统时，必然会经过登录页面还有一些不进行任何操作的信息页面，如：利用客户端的 Cookies 和后退按钮，

用户可以方便地浏览网页，所以这样的页面是没有价值的，根据 Page 属性将其过滤掉。普通类型用户不是研究的范围，可以根据 UserType 属性将其过滤掉，剩下管理类型的用户的日志信息。完成数据清洗后，便很容易根据 SID 统计出某个管理类型用户的会话数据：

表 1　日志数据集

时间（T）	会话标识符(SID)	用户类型(UserType)	用户界面(Page)	执行的函数(Fun)
T1	S1	USER_ADMIN	P1	C1.F1
T2	S2	USER_ADMIN	P2	C2.F4
T3	S3s	USER_ADMIN	P2	C2.F5
T4	S1	USER_ADMIN	P7	C2.F4
T5	S2	USER_ADMIN	P4	C1.F2

表 2　预处理后的日志数据集

会话标识符(SID)	时间（T）	用户类型(UserType)	用户界面(Page)	执行的函数(Fun)
S1	T1	USER_ADMIN	P1	C1.F1
S1	T2	USER_ADMIN	P1	C1.F1
S1	T3	USER_ADMIN	P2	C1.F1
S1	T4	USER_ADMIN	P2	C1.F1
S1	T4	USER_ADMIN	P4	C1.F1

然后我们对每个 SID 的数据计算如下数据：

表 3　统计处理后的日志数据集

SID	UserType	T	PN	FN	T/F
S1	ADMIN	420s	6	12	T1
S2	ADMIN	730s	8	34	T2
S3	ADMIN	100s	2	4	T3
S4	ADMIN	450s	8	23	T4
S5	ADMIN	600s	7	7	T4

T：会话总时间

PN：会话中遍历的页面数

FN：会话中执行的业务逻辑函数总数

T/F：每次操作的平均时间间隔

人为构造属性如下：

F/PN：每个页面的操作数

T/PN：每个页面停留的平均时间

3.2 实验算法介绍

离群点通常被当作聚类挖掘的副产物，许多聚类挖掘算法都将其作为噪声删除。一个离群点是这样的数据点，基于某种度量，该数据点与数据集中其他的数据点有着明显的不同。离群点检测的目的是为了发现数据集中的一小部分对象，与数据集中其余的大部分对象相比，这一小部分对象有着特殊的行为或者具有反常的属性[5,6]。Knorr 和 Ngr 提出基于距离的离群点数据挖掘方法,但是这种方法中的距离难以确定,而且没有离群数据的离群衡量测度。

密度的定义是 DBSCAN 聚类算法中的密度的定义，即数据对象的密度就是该数据对象领域半径内的数据对象的数量。基于密度的离群点概念由 Breuning 等人提出，基于密度的离群点算法（DBOMA，Density-Based Outlier Mining Algorithm）可以发现任意形状的数据布局中的离群数据，它的基本思想是：对于数据集中的每一个离群数据对象，不能包含任何一个给定半径和该半径领域内包含指定数据对象数目的核心对象的领域内。基于密度的离群数据挖掘为了发现所有的离群数据，需要对每个数据进行处理。首先，从数据集 D 中任意找一个数据对象 p，并查找出 D 中 p 的关于半径领域内包含的所有的领域对象，若 p 的 ε 领域内某一个数据对象的 ε 领域内包含 Minpts 或多于 Minpts 个数据对象，即 p 的领域内存在一核心数据对象，则 p 不是离群数据，反之，若 p 的 ε 领域内所有数据对象的 ε 领域内包含的数据对象个数都少于 Minpts 个或者 p 的 ε 领域内没有数据对象即 p 是数据集 D 中关于 ε 领域的孤立点，则 p 是离群数据；接着处理数据集中的下一个数据，直至数据集中的所有数据都被处理完[7,8]。DBSCAN 算法是一个有代表的基于密度的方法，它根据一个密度域值来控制簇的增长。OPTICS 算法是另一个基于密度的方法，它为自动和交互的聚类分析计算一个聚类循序。本文在综合考虑基于密度的离群点算法 DBSCAN 和 CURE 算法、OPTICS 算法的基础上，对其进行改进：计算一个数据对象的领域时，当发现它是密集的，则不必对其邻居进行领域计算，从而提高效率，可以采用该算法来选择离群点。

通过以上表 1 可以看出描述日志的特征通常包括数值特征，也包含符号特征，为了有效计算涉及多种不同类型描述的数据对象间的距离，可以利用特征相关性作为计算数据对象之关的距离的依据。首先把数据对象的特征集 F 分为两个不相交的子集 Fs 和 Fr，Fs 为符号特征集，Fr 为数值特征集。为了能准确判断两个对象之间的距离，对数据集作如下处理：

1) 计算所有数据对象的 Fs 特征集的平均值：

$$V_k = \frac{1}{n}\sum_{i=1}^{n} f_{ik}, k \in [1, s] \quad (1)$$

2) 计算平均的绝对偏差:

$$s_k = \frac{1}{n}\sum_{i=1}^{n} \left| f_{ik} - v_k \right| \quad (2)$$

3) 计算标准化度量值:

$$z_{ik} = \frac{f_{ik} - v_k}{s_k} \quad (3)$$

经过上述处理后两对象间的距离为：

$$d_s\left(e_i^s, e_j^s\right) = \sqrt{\sum_{k=1}^{m_s} \left(z_{ik} - z_{jk}\right)^2} \quad (4)$$

对于符号变量，则将不同类型的变量组合在单个相异度矩阵中，数据对象 e_i^r 和 e_j^r 之间的相异度为：

$$d\left(e_i^r, e_j^r\right) = \frac{\sum_{f=1}^{r} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{r} \delta_{ij}^{(f)}} \quad (5)$$

如果 f_{jf} 或 f_{jf} 为空或 f_{jf}=f_{jf}=0 且变量 f 是不对称的二元变量则指示项 $\delta_{ij}^{(f)} = 0$，否则 $\delta_{ij}^{(f)} = 1$。变量 f 对 i 和 j 之间的相异度的计算方式与其具体类型有关:

1) 当 f 是标称变量时,若 $f_{jf} = f_{jf}$, $d_{ij}^{(f)} = 0$,否则 $d_{ij}^{(f)} = 1$;

2) 当 f 是区间标度变量时：

$$d_{ij}^{f} = \frac{\left| f_{if} - f_{jf} \right|}{\max_h f_{hf} - \min_h f_{hf}} \quad (6)$$

根据上述公式，可得任意两件事件的距离为：

$$d(e_i, e_j) = \alpha d_s\left(e_i^s, e_j^s\right) + (1-\alpha) d_r\left(e_i^r, e_j^r\right) \quad (7)$$

其中，α 为权重因子，利用 α 权重因子可以更好地表达符号特征子集和数值特征子集在计算数据对象之间差异（或距离）时所起的不同作用。

算法的基本思路分两步骤，第一，粗选，将每个单元的数据对象集 D 置为空，S 为空，再将数据集的每个数据对象映射到一个单元内，如 f(d(d1,d2,…dm))->U。如果 U.D>Minpts,不是离群数据；否则为离群数据。 第二，精选，对第一步暂定为离群数据的单元进行处理。若 U 的邻居为空单元，那么该单元内的数据为离群数据；若 U 的邻居为非空单元，则进一步计算其偏离指数，如果偏离指数小于指定的阈值，则该数据对象不是离群数据，否则为离群数据。该算法的描述如下:

```
FDBSCAN()
输入:Minpts, L, f
输出:离群点数据集合 Os-set
For 数据集中的每一个数据对象 d do
{将每个数据对象 d 映射到相应的 U 单元中}
For 每个单元 U do
{ If U 是 DU（即 U.D>=Minpts），单元里的数据都不是离群数据
    Continue
  Else if U 的邻居单元为空
  单元内的数据都是离群数据
  Else
  If LDI(d)>f
    d->Os-set
}
```

3.3 实验结果

根据以上算法原理，利用 VC++ 6.0 编程实现，测试程序在 Pentium Dual 1.73GHz,1G 内存，120Gbyte 硬盘，Windows 2000 XP 上运行的，实验数据来自网络中心某台服务器 2010 年 3 月 12 日一天的日志记录，实验结果如下表 4 所示。

检测率为实际检测到的异常数目与数据集中包含异常的数目之比；误报率（包括假报、漏报）为误报的数据与正常行为的数目之比。其中，DOS 为拒绝服务异常；Intrusion Attempt (企图攻击),尝试猜测口令或企图越权操作而导致操作异常；Penetration（合法用户的攻击），本地用户权限提升异常，一般会访问那些原来不允许的数据库或数据库对象；R2L 为远程异常。

4 结语

以上采用的聚类属性是人为构造的，这样并不能确定那一个属性的效果更好，所以需要使用反复实验的方法找到更好的聚类属性。本文对日志文件进行充分分析的基础上，先对系统日志数据进行事务还原处理，找到代表每次事务的特征属性，然后使用聚类算法找到孤立点，而这个孤立点就代表了对系统进行异

常操作的用户。以上方法只是检测异常用户，即用户在单一会话中的异常行为检测，对于那些多会话的异常行为是无能为力的。而大多数系统使用者本身的操作行为，必然存在很多有趣的行为模式，如熟练用户与不熟练用户的操作模式一定是不同的，经常偷懒的使用者和积极的工作者在实用系统中也必然存在巨大的差异，什么样的使用者会使系统出现更多的异常。下一阶段将在现有的模式上建立系统使用者的行为模型，当挖掘出用户的使用模式后，可以使用 svm 等方法对以上描述类用户的使用模式建立一个模型，然后使用这个模型区分出不同类型的用户，准确地预测非法用户的入侵。

表 4　测试结果

数据集	异常类型	异常事件(个)	检测异常事件(个)	检测率	误报率
抽取 6386 条样本，异常 90 条,误报 158 条	DOS	18	15	81.3953%	158/(6386-90)=25.138
	Intrusion Attempt	20	16	81.25%	
	Penetration	45	31	68.5185%	
	R2L	7	5	70.5882%	

参考文献

1 Hawkins D. Identification of Outliers. London: Chapman and Hall, 1980.

2 柴平瑄,程时端.入侵检测技术分析.计算机工程与应用,2003,14:164–166.

3 肖国强,肖铁.一种从 WEB 日志中挖掘访问模式的新算法.华中科技大学学报(自然科学版),2004,32(5):70–72.

4 H Z, Xu X, Deng S. FP-outlier: frequent pattern based outlier detection. ComSIS, 2005,2(1):103–118.

5 Han J, Kamber M.范明,孟小峰,译.数据挖掘概念与技术.北京:机械工业出版社,2001.3–22.

6 Knorr E, Roymond NG. Algorithms for mining distance-based outlier in large databases. Proc. of the VLDB Conf. New York: 1998: 390–405.

7 崔贯勋.基于密度的离群数据挖掘算法研究[硕士学位论文].重庆:重庆大学,2007.

8 徐翔,刘建伟,罗雄.离群点挖掘研究.计算机应用研究,2009,26(1):34–40.

(上接第 97 页)

梅尔倒频谱参数提取，即可获得对应语音的基频特征向量与 MFCC 特征向量，设标准语音的 MFCC 特征向量为 $M_1=[m_1(1), m_2(2),\cdots m_1(T)]$，基频特征向量为 $P_1=[p_1(1), p_2(2),\cdots p_1(T)]$（T 为标准语音长度）；待评价语音的 MFCC 特征向量为 $M_2=[m_2(1), m_2(2),\cdots m_2(S)]$，基频特征向量为 $P_2=[p_2(1), p_2(2),\cdots p_2(S)]$，（S 为待评价语音长度），系统只需进行一次 DTW 操作，就可按以下公式求取基频变化相似度 P 以及 MFCC 特征相似度 M。

$$C=\begin{pmatrix}C_1\\C_2\end{pmatrix}=DTW(M_1,M_2)$$，（C 是特征比较矩阵）

$$\begin{pmatrix}P\\M\end{pmatrix}=\begin{pmatrix}P_1 & P_2\\M_1 & M_2\end{pmatrix}\begin{pmatrix}C_1\\C_2\end{pmatrix}$$

由于各语音特征参数间存在着关联性，可根据评分的侧重点不同，在机评分计算公式中引入各特征参数的权值，实现机评分与专家评分之间的最佳映射。

$$Scores(P,M)=k_1P+k_2M+k_3PM$$

4　结语

本文在功能强大的 ARMS3C2410X 硬件平台上，构建了一个嵌入式平台的普通话发音质量评价系统，分别从语音的准确性与朗读的韵律两个方面对普通话的发音质量进行评价，在保证评分质量的情况下，对基于特征比较的语音评价算法进行了改进，大大降低了系统实现平台的硬件资源配置要求，对研制嵌入语音识别片上系统具有很好的参考价值。

参考文献

1 易克出,田斌.语音信号处理.北京:国防业出版社,2000.16-22,331–335.

2 杜普选,马庆龙.实时 DSP 技术及浮点处理器的应用.第 2 版.北京:清华大学出版社,北京交通大学出版社,2007: 86–90.

3 陈彩华,龙卫兵,刘彬.基于 ARM-Linux 的家用网络平台设计与实现.计算机测量与控制,2010,(9):2176–2177,2193.

4 韩纪庆,张磊,郑铁然.语音信号处理.北京:清华大学出版社,2004.133–135.