·安全技术· 文章编号：1000-3428(2018)06-0122-08 文献标志码：A 中图分类号：TP391

基于流式聚类及增量隐马尔可夫模型的实时反欺诈系统

李旭瑞[1,2]，邱雪涛[2]，赵金涛[2]，胡 奕[2]

（1. 复旦大学 计算机科学技术学院，上海 200433；
2. 中国银联股份有限公司 电子商务与电子支付国家工程实验室，上海 201201）

摘 要： 针对目前金融支付行业交易中存在的欺诈风险复杂化和高频化的问题，提出一种基于密度分布演化的流式聚类算法（DDE-Stream）。利用 CLIQUE 算法对账户交易特征进行网格聚类，结合隐马尔可夫算法构建账户交易行为档案模型，根据该模型进行实时的欺诈侦测，并在模型自更新阶段，利用 DDE-Stream 算法对多维度交易特征进行实时聚类。实验结果表明，该算法能够实时侦测交易欺诈风险，且在验证集上获得的准确率相比传统随机森林分类算法超过 50% 。

关键词： 实时风控；欺诈侦测；行为档案；流式聚类；增量隐马尔可夫

中文引用格式： 李旭瑞，邱雪涛，赵金涛，等. 基于流式聚类及增量隐马尔可夫模型的实时反欺诈系统[J]. 计算机工程，2018，44(6)：122-129.

英文引用格式： LI Xurui，QIU Xuetao，ZHAO Jintao，et al. Real-time anti-fraud system based on stream clustering and incremental hidden Markov model[J]. Computer Engineering，2018，44(6)：122-129.

Real-time Anti-fraud System Based on Stream Clustering and Incremental Hidden Markov Model

LI Xurui[1,2]，QIU Xuetao[2]，ZHAO Jintao[2]，HU Yi[2]

（1. School of Computer Science，Fudan University，Shanghai 200433，China；2. National Engineering Laboratory for Electronic Commerce and Electronic Payment，China UnionPay Co.，Ltd.，Shanghai 201201，China）

【Abstract】 Aiming at the complexity and high frequency of fraud risk existing in the current financial payment industry transactions，a new method called the Density Distribution Evolution-based Stream（DDE-Stream）clustering algorithm is proposed. The CLIQUE algorithm is used to cluster the trading features of the account，and an account trading behavior archive model is constructed by combining with hidden Markov algorithm. According to the model，real-time fraud detection can be performed. In the stage of model self-renewal，DDE-Stream algorithm is used to perform effective real-time clustering of multi-dimensional trading features. Experimental results show that the algorithm can detect the risk of transaction fraud in real time，and the accuracy rate obtained on the verification set exceeds 50% ，compared with the traditional random forest classification algorithm.

【Key words】 real-time wind control；fraud detection；behavior archives；stream clustering；incremental hidden Markov

DOI： 10.19678/j.issn.1000-3428.0047241

0 概述

在金融科技快速发展的形势下，金融支付行业在交易量增长的同时，也伴随着更加复杂化、高频化的欺诈风险。目前，交易欺诈风险侦测的方式主要分为两大类：基于规则和基于机器学习算法。其中，规则系统主要依赖专家经验，存在一定的主观因素，且难免疏漏，基于机器学习算法的反欺诈方案相对规则系统具有更好的客观性及准确性[1]。逻辑回归、支持向量机以及随机森林等分类算法最为常用[2]。此外，频繁项集挖掘[3]和神经网络[4]等算法也在反欺诈领域有着较好的效果，但上述方法仍然存在以下问题：

1）上述模型主要是通过对大量账户的交易特征进行统计分析后训练得出的一个较为通用的结果，然而具体到单个账户的情况下，每个账户的交易行为很有可能与通用的规则或模型有偏差。

2）目前绝大多数的机器学习反欺诈模型在实时场景并不适用。例如最为主流的分类算法需要事先对训练样本标注。但是在实时交易场景中，新产生

基金项目： 国家发展和改革委员会国家信息安全专项（[2015]289）；上海市青年科技英才扬帆计划项目（17YF1425800）。

作者简介： 李旭瑞（1989—），男，博士，主研方向为大数据、人工智能、金融风险防控等；邱雪涛、赵金涛、胡 奕，工程师、硕士。

收稿日期： 2017-05-17 **修回日期：** 2017-07-13 **E-mail：** xurui.lee@msn.com

的交易数据不可能及时被标记为是否欺诈。

3)随着欺诈手段的不断更新,很多规则甚至模型都有可能会失效。面对这种潜在的失效可能,目前的一般做法是每隔一段时间重新进行模型训练。然而,这种做法一方面需要耗时耗力地进行周期性的人工维护,另一方面也不能及时地根据当前信息做出最准确的判断。

针对通用方法的不足,文献[5]提出可以通过隐马尔可夫模型(Hidden Markov Model,HMM)建立账户级别的历史交易序列模型。不过由于基本的 HMM 算法只能针对单个特征的序列进行分析,因此仅将交易金额分为高、中、低 3 类作为 HMM 模型的观察状态变量。显然,这对于欺诈交易风险分析是远远不够的。其他一些改进算法例如最大熵马尔科夫[6]、条件随机场[7]以及循环神经网络[8]能够满足多维度的特征分析,但是往往需要较大的计算量,不适用于实时反欺诈场景。文献[9]提出自组织映射算法在实时反欺诈场景有一定的效果,但是该方法和上面提到的算法一样,仍然只是通过离线训练好的模型进行判别,不能根据新的数据实时地进行在线更新。

针对以上方法的不足,本文提出一种基于流式聚类及增量隐马尔可夫模型的实时反欺诈系统。该系统利用 CLIQUE 算法对账户交易特征进行网格聚类,并结合 HMM 算法对账户交易行为建模,根据该模型进行实时的欺诈侦测。

1　本文算法

本文算法系统的主要思想是根据特定账户的历史交易模式判断当前交易是否可疑。系统结合改进的流式聚类及增量隐马尔科夫算法,能够实时处理及更新多特征行为模型,更加适用于实时反欺诈的应用场景。

1.1　交易行为模型预训练

1.1.1　账户交易特征提取

HMM 是一种统计模型,用来描述一个含有隐含未知参数的马尔可夫过程。在该模型中,观察值是关于状态的随机过程,而状态是关于时间的随机过程,因此,HMM 是一个双重随机过程。

一个 HMM 模型[10]是由一个五元组来描述,即$(Q,O,\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$。其中:

$Q=\{q_1,q_2,\cdots,q_N\}$为隐藏状态集合;

$O=\{o_1,o_2,\cdots,o_M\}$为观察状态集合;

$\boldsymbol{A}=\bar{\boldsymbol{a}}_{ij}^{\mathrm{T}},a_{ij}=P(Q_{t+1}=q_j\mid Q_t=q_i)$为隐藏状态间的转移概率矩阵;

$\boldsymbol{B}=\{b_{ik}\},b_{ik}=P(Q_t=o_k\mid Q_t=q_i)$为隐藏状态到输出状态的概率矩阵;

$\boldsymbol{\pi}=\{\pi_i\},\pi_i=P(Q_1=q_i)$为隐藏状态的初始概率分布。

具体到交易上,在上面的五元组中,隐藏状态表示某一特定的交易状态。观察状态就是每一笔交易的可观测的某种状态。然而,原始的 HMM 模型假设前提使得它不能处理多于一个标记的特征,而每一笔交易中的多个特征都会对欺诈分析起到一定的作用。因此,这里是将每一笔交易的多个可观测特征组合为一个符号来表示。

本文根据经验及统计分析等方法挑选原始变量。在实际交易行为中,像交易金额、交易时间、交易地点、交易网络 IP、设备指纹、操作习惯等都可以作为用于风险监控的变量。实时系统需要将交易记录进行特征映射,处理完毕后,每条交易记录就对应一个交易特征向量。

1.1.2　账户交易行为特征预聚类

由于 HMM 方法只能针对单变量进行处理,因此对交易特征向量进行聚类。每一个类别代表一群最相似的集合,处于同一集合内的交易被认为是具有相似模式的交易行为,接着将每笔交易对应的类别作为 HMM 模型的变量即可。在反欺诈场景中,单纯使用 K-means 这一类算法的效果不是很好,因为它一开始就需要指定聚类的个数,并且使用基于距离度量准则,聚类结果均趋于球形,对交易数据这种高维分布的效果较差。这里首先选用基于网格聚类的 CLIQUE 算法[11],主要思想是在 d 维的网格空间 $g(s_1,s_2,\cdots,s_d)$中,每一维 s_i 被分为 p_i 份,这样整个属性的空间被分为 $S=\prod_{i=1}^{d}p_i$ 个网格。对一个样本的特征向量表示为 $\boldsymbol{x}_j=(a_{j1},a_{j2},\cdots,a_{jd})$。如果映射到某一网格的对象数超过该密度阈值则该网格是稠密的。首先根据简单的贪婪搜索方法,从一个任意稠密单元开始,找出覆盖该单元的最大区域,然后在尚未被覆盖的剩余的稠密单元上继续这一过程,直到稠密单元都被覆盖,这时连成一片的稠密单元就是一个聚类簇。

聚类完成后,会得到 K 个类别和对应的类别中心。类别号分别用$\{c_1,c_2,\cdots,c_K\}$中的元素来表示。由于初始聚类模型是由较大量的历史交易数据计算得到的结果,能比较准确地代表样本分布特征,因此在后续模型更新的过程中,维持 K 值不变。

1.1.3　基于历史交易序列的 HMM 模型预训练

令 $\lambda=\{\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi}\}$为 HMM 的参数。根据给定账户的历史交易序列(观察状态集合)来学习模型参数 λ。使用向前-向后(Baum-Welch)算法来计算模型参数 λ。交易序列分析中的更新步骤如下:

1)参数 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{\pi}$ 的初始值分配

假设隐藏状态一共有 s 个(具体可以在后期步进调试选择最佳的参数)。由于并不知道每种隐藏的交易状态到底代表什么,因此这里对于隐藏状态的初始概率分布 $\boldsymbol{\pi}$ 和隐藏状态间的转移概率矩阵 $\boldsymbol{A}$ 只进行简单的均分,即每个隐藏状态的初始概率都

为 $1/s$,从一个隐藏状态跳到另一个隐藏状态的概率也为 $1/s$(包括隐藏状态转移到其本身的情况)。而隐藏状态到输出状态的概率矩阵 $\boldsymbol{B}$,可以根据交易特征聚类类别的分布情况进行初始化赋值。有效的初始化值有助于更快更好的收敛。对于特定的账户,根据其历史交易对应类别号集合的 $\{c_1,c_2,\cdots,c_k\}$ 占其总交易总数的百分比进行概率初始化。

2)最大似然(EM)估计

给定观察序列 $O=o_1,o_2,\cdots,o_T$,调节模型参数 λ,使得期望最大。

E 步骤:

首先根据当前的参数 λ 计算向前变量 $\alpha_t(i)$ 和向后变量 $\beta_{t+1}(i)$:

$$\alpha_t(i)=P(O_1,O_2,\cdots,O_t,q_t=i\mid\lambda)$$

$$\beta_{t+1}(i)=P(O_{t+1},O_{t+2},\cdots,O_T\mid q_t=i,\lambda)$$

其次计算在时间 t 位于状态 q_i、时间 $t+1$ 位于状态 q_j 的概率 $\xi_t(i,j)$:

$$\xi_t(i,j)=P(Q_t=q_i,Q_{t+1}=q_j\mid O,\lambda)=\frac{\alpha_t(i)\times a_{ij}\times b_j(O_{t+1})\times\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)\times a_{ij}\times b_j(O_{t+1})\otimes\beta_{t+1}(j)}$$

然后计算在时间 t 位于状态 q_i 的概率 $\gamma_t(i)$:

$$\gamma_t(i)=\sum_{j=1}^{N}\xi_t(i,j)$$

M 步骤:

重新估计初始状态($t=1$ 时刻隐藏状态 q_i 的概率):

$$\bar{\boldsymbol{\pi}}_i^{\mathrm{T}}=\gamma_1(i)$$

重新估计转移概率矩阵 $\bar{\boldsymbol{a}}_{ij}^{\mathrm{T}}$:

$$\bar{\boldsymbol{a}}_{ij}^{\mathrm{T}}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

重新估计输出概率矩阵,$\bar{\boldsymbol{b}}_j^{\mathrm{T}}(k)$ 为 Q 从状态 q_j 发出观测状态 o_k 的期望 Q 到达状态 q_j 的期望:

$$\bar{\boldsymbol{b}}_j^{\mathrm{T}}(k)=\frac{\sum_{t=1}^{\mathrm{T}}\gamma_t(j)\times\delta(O_t,o_k)}{\sum_{t=1}^{\mathrm{T}}\gamma_t(j)}$$

如果 $O_t\neq o_k$,则 $\delta(O_t,o_k)=0$;否则 $\delta(O_t,o_k)=1$。

根据期望 $\xi_t(i,j)$ 和 $\gamma_t(i)$,带入 c、d、e 式重新得到 $\bar{\boldsymbol{\pi}}_i^{\mathrm{T}}$、$\bar{\boldsymbol{a}}_{ij}^{\mathrm{T}}$、$\bar{\boldsymbol{b}}_j^{\mathrm{T}}(k)$,即得到新的参数 $\boldsymbol{\lambda}^{\mathrm{T}}$。

3)前向与后向变量的循环计算直至收敛

根据当前的参数 $\boldsymbol{\lambda}^{\mathrm{T}}$ 重新计算向前和向后变量,开始下一轮重新计算。依次迭代计算直到 $\bar{\boldsymbol{\pi}}_i^{\mathrm{T}}$、$\bar{\boldsymbol{a}}_{ij}^{\mathrm{T}}$、$\bar{\boldsymbol{b}}_j^{\mathrm{T}}(k)$ 收敛。

1.2 实时交易欺诈侦测方法

1.2.1 账户参数存储

训练完参数 λ 之后,得到一个训练好的隐马尔可夫模型。根据该模型就得到了一个账户的历史交易行为的模型,后续根据该模型即可判断一笔交易是否具有风险。后台系统需要维护一个专门的数据表来存储每个账户对应的参数,实例存储结构如图 1 所示。

账户索引	参数1	参数2	参数3	观测序列	序列概率	时间戳
*id*1	*A*1	*B*1	*π*1	01	*ρ*1	*t*1
*id*2	*A*2	*B*2	*π*2	02	*ρ*2	*t*2
*id*3	*A*3	*B*3	*π*3	03	*ρ*3	*t*3
…	…	…	…	…	…	…

图 1 账户参数存储结构示例

可以使用 MongoDB 或者 HBase 这类非关系型数据库来存储这些参数,因为每个账户对应的参数中的 3 个矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{\pi}$ 的格式都不是固定的,在训练阶段会根据验证不断调整优化参数结构。其中,每个账户的观察序列只存取最近的 R 笔交易 $O=\{o_1,o_2,\cdots,o_R\}$,o_i 就是上面计算出来的 $\{c_1,c_2,\cdots,c_k\}$ 中的一个。每个序列被设计为一个空间为 R 大小的循环队列,以此保证空间利用率。只有历史交易笔数达到 R 笔的账户才会被进行模型训练,并将训练好的参数结果进行保存。图 1 中的序列概率是指根据当前账户模型计算出来的当前观察状态序列的概率值。R 的数值大小可以根据后续业务经验进行调整。后台服务器需要每隔一段时间对该数据库进行一次更新,每次更新时只对最近有交易记录变化的账户进行操作。图 1 最后一列对应每次更新的时间戳。服务器后台再定期更新数据库时,还可以设置一个时长参数 P_e,将时间戳大于 P_e 的账户记录给删除,这样不仅可以避免数据库存储空间的不足,还能使得账户模型更加符合近期交易行为。

1.2.2 实时交易欺诈判别

对于特定账号新进的一笔交易,在对该笔交易特征化之后,映射到相应的网格内,该网格属于某一聚类簇,用该类别号表示该笔交易。由于成千上万的账户会不断产生交易信息,因此可以采用 Spark Stream 等流式计算框架来实时地对每一笔交易进行处理和判别。Spark Stream 是将流式数据按照设定的批大小(如 1 s)分成一段一段的数据,每一段数据都转换成 Spark 中的 RDD 后在内存中进行计算。

通过对每一笔交易预处理完之后,取该账号的前 R 笔交易 $O=\{o_1,o_2,\cdots,o_R\}$,其中 o_i 就是上面计算出来的 $\{c_1,c_2,\cdots,c_k\}$ 中的一个。对应根据训练好的模型计算该观测序列的概率 $\rho_1=P(o_1,o_2,\cdots,o_R\mid\lambda)$。然后丢弃最早的 o_1,将当前交易作为 o_{R+1} 与剩余的历史交易合并为新的序列,求这个新序列的概率 $\rho_2=P(o_2,o_3,\cdots,o_{R+1}\mid\lambda)$。令概率的变化率 $\Delta\rho=\frac{\rho_1-\rho_2}{\rho_1}$。若 $\Delta\rho>0$,说明当前概率小于原来概率,则存在欺诈可能。定义一个阈值 θ,如果 $\Delta\rho\geqslant\theta$,则说明当前的交易行为模式与其前段时间的交易行为模

型差别较大,因此判别为欺诈交易[12]。如果 $\Delta\rho < \theta$,那么认为该笔交易正常,并将它添加到该账户的历史交易序列中去,用于模型更新。具体的阈值 θ 大小可以在后续业务中根据相应的指标进行适当的调整。

1.3　在线模型自更新

随着互联网 + 的兴起,在线支付变得越来越频繁,单账户的交易行为也有可能随着时间的变化而不断变化,离线训练风控模型很有可能不能及时地获取到最新的信息。并且人工周期性地对大量的账户进行模型训练本身也是耗时耗力的工作。如果能够根据当前交易及时地更新模型,就能够获取到最新的信息,从而做出更准确的判断。具体的模型更新规则如下所示。

1.3.1　更新数据的选取

如果上文计算出来的概率变化率 $\Delta\rho < \theta$,则说明当前交易符合该账户的正常交易行为。这时,将这笔交易作为新的正常样本数据来对模型进行在线更新。

1.3.2　流式聚类的更新

由于每次新进来的交易数据都会对前面的聚类结果产生影响,因此需要先在线更新聚类模型。CluStream 算法是目前比较权威的一种流式聚类算法,但它是基于 Kmeans 算法的实时改进版本,所以,在实时反欺诈场景中效果并不是很好[13]。D-Stream (Density-based Stream)聚类算法借鉴了 CluStream 算法中的在线微聚类和离线宏聚类的框架思想,实现了一种基于时间衰减技术的实时网格聚类算法,有效避免了欧式距离度量的不足,能够对任意形状的分布进行聚类[14]。但是在该算法中,每个网格密度的定义仅考虑了当前网格内样本随时间变化的分布情况,而没有整体考虑每个网格中样本在整个空间所占比例随时间的变化。这样,就会导致网格密度计量的偏差。鉴于交易行为特征的复杂性和多变性,本文提出基于密度分布演化的实时聚类 DDE-Stream (Density Distribution Evolution-Based Stream)算法,具体方法如下:

1)网格密度在线预更新

给每一个样本分配一个密度权重系数 w_j,当该样本刚加入到某一网格时密度权重系数为 1,然后按下式进行衰减:$w_j^{t+\delta t} = \lambda^{\delta t} w_j^t$,其中 $0 < \lambda < 1$。设某个时间点为 t_0,第 i 个网格内所有样本的密度权重之和为 $W_i^{t_0} = \sum_{n=1}^{N_i^{t_0}} w_n$。设定一个阈值 D_l,定义当第 i 个网格的密度 $D_i < D_l$ 时该网格为稀疏网格,否则即为稠密网格。该定义简化了原始 D-Stream 中过渡网格带来的计算复杂度。而原始 D-Stream 算法已经证明,在初始化阶段过后,如果在某一时刻删除一个稀疏网格,则不会对聚类结果产生影响。因此,仍然遵循这个原理,只在初始化时计算一次整个网格空间,后续则只考虑当前网格占稠密网格空间的比重。设整个稠密网格空间中的所有样本的密度权重之和为 $W_A^{t_0} = \sum_{m=1}^{N_A^{t_0}} w_m$,那么定义第 i 个网格此时的密度为 $D_i^{t_0} = \dfrac{W_i^{t_0}}{W_A^{t_0}}$。

2)在线疏密判断

每经过一个较短的时间 δt 后就更新一次网格的密度值。假设第 i 个网格内在 δt 时间段内一共新增了 $k_i^{t_0}$ 个样本,可以认为此时这 k_i 个样本的密度权重都是 1,那么第 i 个网格内所有样本的密度权重之和为:

$$W_i^{t_0+\delta t} = \sum_{n=1}^{N_i^{t_0}} \lambda^{\delta t} \times w_n^{t_0} + k_i^{t_0} = \lambda^{\delta t} \times W_i^{t_0} + k_i^{t_0}$$

整个稠密网格空间中的所有样本的密度权重之和为:

$$W_A^{t_0+\delta t} = \sum_{m=1}^{N_A^{t_0}} \lambda^{\delta t} \times w_m + \sum_{j=1}^{s} k_i^{t_0} = \lambda^{\delta t} \times W_A^{t_0} + \sum_{j=1}^{s} k_j^{t_0}$$

其中,r 为 δt 时间内有新样本进入的网格数。所以,在经过 δt 时间段后,网格密度的首次更新公式为:

$$D_{i0}^{t_0+\delta t} = \frac{\lambda^{\delta t} \times W_i^{t_0} + k_i^{t_0}}{\lambda^{\delta t} \times W_A^{t_0} + \sum_{j=1}^{r} k_j^{t_0}}$$

3)在线二次密度更新

根据首次更新公式 $D_{i0}^{t_0+\delta t}$ 判断网格是稠密还是稀疏,如果是稀疏,则直接将该网格删除。假设一共删除了 p 个稀疏网格,那么对于剩下的稠密网格,需要进行二次密度更新。最终得到第 i 个稠密网格的密度更新为:

$$D_i^{t_0+\delta t} = \frac{\lambda^{\delta t} \times W_i^{t_0} + k_i^{t_0}}{\lambda^{\delta t} \times \left(W_A^{t_0} - \sum_{l=1}^{p} w_p^{t_0}\right) + \sum_{j=1}^{q} k_j^{t_0}}$$

其中,q 为在 δt 时间段内有新样本加入的稠密网格的数量。

综上所述,该算法仍然只负责维护一个稠密网格集,并且对该集合中的网格进行后续离线聚类,不仅能够很好地表示密度分布演化,而且能有效地避免全网格的复杂计算。

4)离线宏聚类

根据以上计算,可以获得任意时刻的稠密网格。设在所有 S 个网格空间中存在任意 2 个网格单元 s_1 和 s_2,当这 2 个网格单元在一个维度上有交集时,则它们互为邻接网格单元。互为邻接单元的网格所连接成的最大连通子网格空间即为一个稠密连通簇。这时计算每一个稠密连通簇在网格空间中的坐标质心点,然后使用简单的 K-means 算法对多个质心点根据初始聚类 k 值对连通聚类网格进行宏聚类。这样,在整个空间得到的若干类簇即为当前在线聚类结果。

1.3.3 增量 HMM 模型更新

根据以上方法,就可以得到实时更新的聚类。对于每一个新的交易样本进来之后,都能够保证它能被映射到最新的聚类簇中,即每个交易样本都能被准确地对应于 HMM 模型的某一个观测状态。当一个新的交易被映射后,就可以利用增量改进的 HMM 模型进行模型更新,这样能够保证模型的时效性[15]。接下来根据以下方法进行 HMM 模型的更新[16]:

1)参数 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{\pi}$ 的初始值分配。

取上一轮计算好的 HMM 参数作为当前参数的初始值。

2)最大似然(EM)估计。

仍然需要根据当前的参数 λ 计算向前变量 $\alpha_t(i)$,方法不变。但是由于实时情况下不能观测到后面时间的状态,因此后向变量 $\beta_{t+1}(i)$ 不能进行流式计算,这里假设 $\beta_t(i)=\beta_{t+1}(i)=\cdots=\beta_T(i)=1$。

E 步骤:

期望的增量修正公式如下:

$$\gamma_T(i)=\frac{\alpha_T(i)}{\sum_{i=1}^{N}\alpha_T(i)}$$

$$\xi_{T-1}(i,j)=\frac{\alpha_{T-1}(i)\times a_{ij}\times b_j(O_T)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{T-1}(i)\times a_{ij}\times b_j(O_T)}$$

M 步骤:

最大化计算的增量修正如下:

估计初始状态方法不变:

$$\bar{\boldsymbol{\pi}}_i^{\mathrm{T}}=\gamma_1(i)$$

重新估计转移概率矩阵 $\bar{\boldsymbol{a}}_{ij}^{\mathrm{T}}$ 的流式修正公式:

$$\bar{\boldsymbol{a}}_{ij}^{\mathrm{T}}=\frac{\bar{a}_{ij}^{T-1}\left(\sum_{t=1}^{T-2}\gamma_t(i)\right)+\xi_{T-1}(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

重新估计输出概率矩阵 $\bar{\boldsymbol{b}}_j^{\mathrm{T}}(k)$ 的流式修正公式:

$$\bar{\boldsymbol{b}}_j^{\mathrm{T}}(k)=\frac{\bar{b}_j^{T-1}(k)\left(\sum_{t=1}^{T}\gamma_t(j)\right)+\psi(T,j,k)}{\sum_{t=1}^{T}\gamma_t(j)}$$

如果 $O_t\neq o_k$,则 $\psi(T,j,k)=0$;否则 $\psi(T,j,k)=\gamma_T(j)$。

根据 M 步骤的 3 个公式即得到新的参数 $\boldsymbol{\lambda}^{\mathrm{T}}$。

3)循环计算直至收敛。

每进来一个新的正常交易,就进行 EM 循环至收敛得到一个新的模型。

以上步骤在一个判别为正常的交易被加入到对应的观察序列中时进行,保证系统只对产生交易的账户进行模型更新,而没有新交易的模型则保持原状,这样系统的资源可以达到更好的利用。

1.4 整体系统框架

整体的实时自更新反欺诈系统的框架如图 2 所示。首先是账户交易模型初始化,该流程主要步骤用虚点线标出。当一条交易发生时,风控系统的实时欺诈侦测主要流程用实线标出。主要是利用当前的聚类模型和 HMM 模型进行欺诈侦测。点划线标出的是聚类模型和 HMM 模型的更新流程,穿插在实时欺诈侦测流程中的。

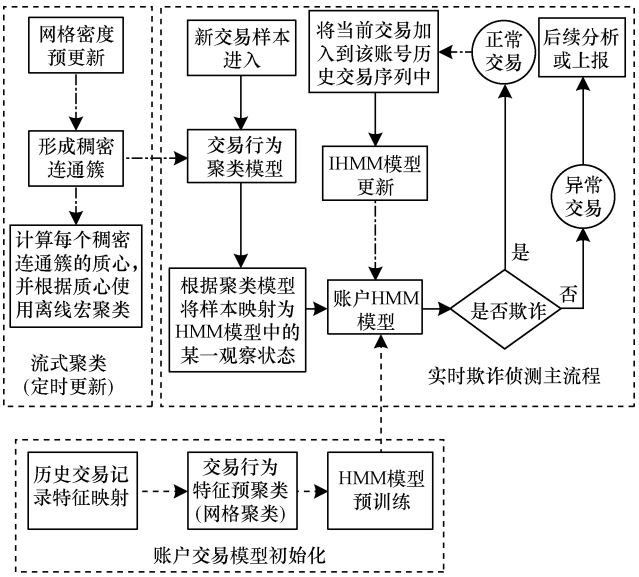

图 2 实时反欺诈系统整体框架

模型更新需要注意的是,聚类模型是针对所有账户的所有交易进行聚类的。它不是每时每刻都进行更新,而是每隔一小段时间 δt 自动更新一次聚类模型,这样基本可以满足行为模型更新的需求(因为所有账户整体交易的聚类分布在较短时间内变化不会很大),也减轻了服务器的压力。而对于具体一条交易样本,对应账户短时间内的交易行为模型可能会变化较大,所以,每当有效样本进入时,都要进行一次 IHMM 模型更新。

2 实验结果与分析

2.1 实验方法步骤

按照以下 2 个步骤来完成实验设计:

1)利用通用数据集,将 DDE-Stream 算法与其他流式聚类算法进行性能对比。

2)利用真实的交易数据,验证本文提出的实时欺诈侦测系统的性能。

2.2 实验设计

2.2.1 DDE-Stream 算法性能验证

本文实验采用的硬件环境 CPU 为 Intel Core i5,主频 3.2 GHz,内存为 8 GB。聚类实验使用的是 KDD-CUP-99 数据集[17]。该数据是一组网络攻击

侦测领域的权威验证集。由于数据量较大,本文仅从该数据集随机抽取了 10% 的已知攻击行为数据,共计 494 021 条数据(约 71 MB)进行验证。每条数据分别被标记为 23 个类别中的一种(1 类正常数据以及 22 小类攻击数据)。其中每条数据具有 32 维连续属性和 9 维离散。这里针对归一化后的连续属性进行计算。

在实验中,对于 DDE-Stream 算法的参数设置如下:衰减系数 $\lambda = 0.85$;密度阈值 $D_l = 0.8$。要注意的是,在整个实际的实时反欺诈模型中,并不需要提前指定离线宏聚类的 K 值,而是在模型预训练阶段通过 CLIQUE 算法得到的类别数作为 K 值即可。这里作为数据验证阶段,假设离线宏聚类阶段的 K 值为 23。

定义聚类纯度:

$$Pur = \frac{\sum_{i=1}^{K} \frac{c_i^D}{c_i}}{K} \times 100\%$$

其中,K 代表类别总数,定义聚类后每个类别中对应样本个数最多的原始类别标记为当前类别的标记。对于聚类后的类 i,其样本个数为 c_i,而 c_i^D 代表原始类别标记为 i 的样本出现在类 i 的个数。

考虑到样本点的权重随时间的推移而逐渐衰减,这里在计算聚类纯度时,只统计某一固定时刻点在给定时间窗口内的纯度统计。时间窗口大小可以调节,这里暂取为 1 s,数据流速设置为 1 000 点时间窗口。

在固定其他参数情况下,聚类纯度随时间窗口增长的变化情况如图 3 所示。结果显示,当时间窗口数增长到 100 ~ 150 之间时,聚类纯度值达到最大,为 93.5% 左右。当时间窗口数再继续增长之后,聚类纯度一直稳定在 92.8% 上下。

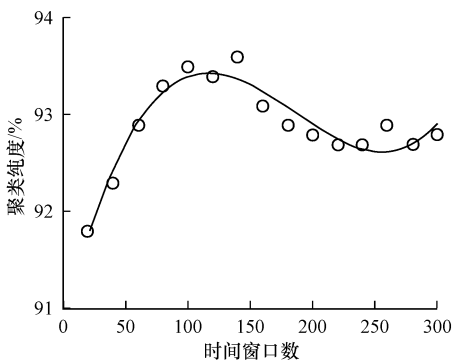

图 3　聚类纯度随时间窗口的变化规律

在固定其他参数的情况下,聚类纯度随每一维网格空间划分数 P_{split} 的变化规律如图 4 所示。从图 4可以看出,当 P_{split} 超过 40 以后,聚类纯度超过 90% 。而当 P_{split} 继续增加之后,聚类纯度的上升趋势变缓。考虑到实时计算的速度要求,网格数量不宜划分太多,在后续实际应用中还需折中考虑。后续实验中 P_{split} 统一取 50。

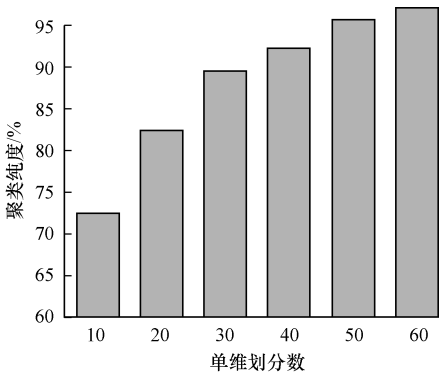

图 4　聚类纯度随网格划分数的变化规律

在同样条件下,DDE-Stream 算法与 CluStream、D-stream 算法的速度对比如图 5 所示。从图 5 可以看出,CluStream 算法的速度最慢,而在数据流量很少的情况下,D-stream 相对 DDE-Stream 算法更为快速,但当流量超过一定范围时,DDE-Stream 算法的速度优势开始展现,明显比 D-stream 快。

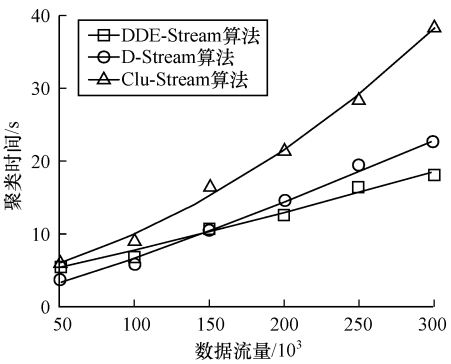

图 5　不同流聚类算法的速度对比

DDE-Stream 算法与 CluStream、D-stream 算法的聚类效果对比如图 6 所示。从图 6 可以看出,DDE-Stream 算法的聚类纯度是最高的。

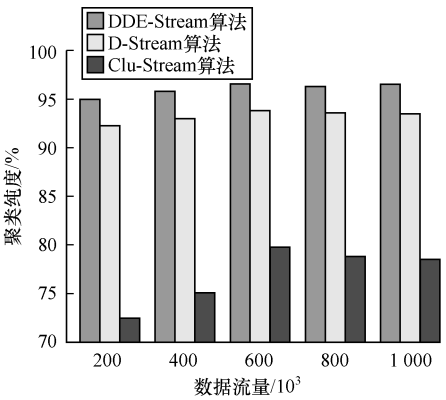

图 6　不同流聚类算法的聚类效果对比

2.2.2 实时欺诈侦测系统的性能验证

首先从 2016 年 1 月—2016 年 6 月的真实交易数据中提取数据。从欺诈样本库中选取一批欺诈账号,而该批账号在 2016 年 1 月之前的半年内是没有任何欺诈交易记录的。然后利用该批账号再去正常交易库中寻找对应的交易记录。对这批欺诈账号进行统计,发现该批账号在这半年内平均交易总次数约为 27。因此,这里作为示例,将总交易记录小于 30 笔的账号筛除。这时,剩余的欺诈交易样本约 10 万条,对应相关账号 2 471 个。考虑到数据的平衡性,选择欺诈账号样本和正常账号样本的比例为 1∶10,即在该半年时间内随机抽取了正常交易笔数大于 30 的账号 24 710 个。然后将这批账号对应的所有交易数据提取出来。将以上所有样本组合起来,作为有标签的训练样本供该模型使用。

根据相关业务知识,选取了可以被实时获取的数据作为待选特征,具体细节不详细描述。最终每一笔交易可以被实时地映射为一个 61 维的向量。在预训练阶段,对所有交易数据采用 CLIQUE 算法进行聚类,结果获得 11 个类别。接着把每一笔交易都映射为对应类别。将每个账号最早的 20 笔交易作为最初的训练样本,利用 HMM 算法分别进行建模。训练完成后,剩余样本作为验证集按一定的流速进行模拟实时分析。在参数选择上,隐藏状态个数的变化从 6 以步进 1 的方式变化到 10,而欺诈概率变化阈值 θ 从 0.1 以步进 0.2 的方式变化到 0.9。

最终在验证集上得到的准确率对比结果如图 7 所示。结果显示,在同样情况下,θ 越大准确率越高(但覆盖率稍有降低),因此实际使用时也需进行折中考虑。至于隐藏状态数,当该参数设置为 6 时达到峰值。整个系统在最优化参数下的准确率可以超过 50% 。

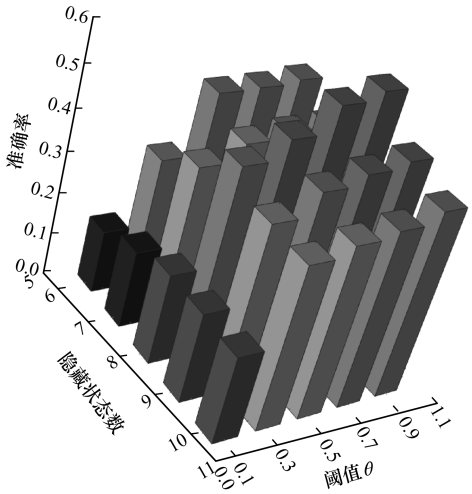

图 7 不同参数下欺诈侦测模型的准确率对比

此外,将本文中的实时欺诈侦测效果与逻辑回归、支持向量机以及随机森林等效果进行对比。分类算法本文并没有进行账户级别的建模,而是将所有账号的所有交易混在一起进行相应的训练和预测。为了和上文的样本分配比例类似,这里随机抽取了上述整体样本的 2/3 作为训练集,剩余 1/3 的样本作为验证集。各个算法的最优准确率如图 8 所示。从图 8 可以看出,随机森林在分类算法中的效果最好,但即使是静态批量预测,在验证集上最高准确率只有 31.4% ,大约只有本文方案的一半。可见本文提出的欺诈风险控制系统不仅能够进行实时侦测和更新,并且在准确度上相对通用分类算法也有了较大的提升。

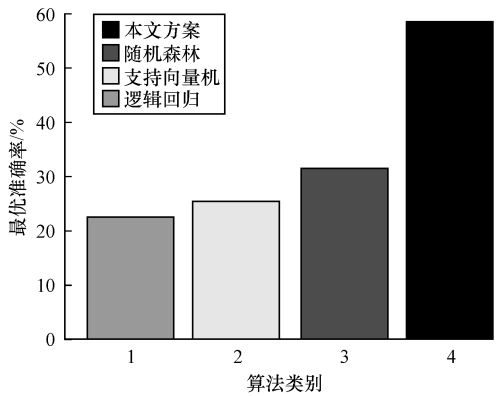

图 8 本文方案与其他分类算法的最优准确率对比

3 结束语

本文利用 CLIQUE 网格聚类算法,结合 HMM 模型进行账户行为档案级别的建模,可以有效反映复杂多维度的交易特性,并且根据当前交易与行为模型的差异进行实时欺诈侦测。此外,本文提出一种基于密度分布演化的实时聚类(DDE-Stream)算法,该算法不仅能够实时处理任意形状的高维交易数据的分布,并且考虑了密度网格占整体网格空间比随时间的演化,较准确地反映了样本数据的分布,有效地对交易行为数据进行聚类。通过结合 DDE-Stream 算法及增量 HMM 模型,可以针对账户交易行为进行自适应的模型更新。该方法既能够实时地侦测欺诈交易风险,又能及时地更新账户的交易行为模式,在一定程度上提高了系统的稳定性。实验结果表明,本文提出的反欺诈系统方案不仅能实时地进行欺诈风险侦测及模型更新,并且在验证集上的欺诈识别准确率与传统的静态分类算法相比提升近 1 倍。

参考文献

[1] AGRAWAL A, KUMAR S, MISHRA A K. Implementation of novel approach for credit card fraud detection[C]// Proceedings of International Conference on Computing for Sustainable Global Development. Washington D. C., USA: IEEE Press, 2015: 8-11.

[2] BHATTACHARYYA S, JHA S, THARAKUNNEL K, et al. Data mining for credit card fraud: a comparative study[J]. Decision Support Systems, 2011, 50(3): 602-613.

[3] SEEJA K R, ZAREAPOOR M. FraudMiner: a novel credit card fraud detection model based on frequent itemset mining[J]. Scientific World Journal, 2014(2014): 1-10.

[4] NATH D M, JAMI S, JOG D. Credit card fraud detection using neural network[J]. International Journal of Students Research in Technology & Management, 2014, 2(2): 84-88.

[5] SRIVASTAVA A, KUNDU A, SURAL S, et al. Credit card fraud detection using hidden markov model[J]. IEEE Transactions on Dependable and Secure Computing, 2007, 5(1): 37-48.

[6] 冯　冲. 统计方法信息抽取中的若干关键技术研究[D]. 北京:中国科学技术大学, 2005.

[7] 朱莎莎,刘宗田,付剑锋,等. 基于条件随机场的中文时间短语识别[J]. 计算机工程, 2011, 37(15): 164-167.

[8] 古　勇,苏宏业,褚　健. 循环神经网络建模在非线性预测控制中的应用[J]. 控制与决策, 2000, 15(2): 254-256.

[9] QUAH J T S, SRIGANESH M. Real-time credit card fraud detection using computational intelligence[J]. Expert Systems with Applications, 2008, 35(4): 1721-1732.

[10] 谭小彬,王卫平,奚宏生,殷保群. 基于隐马尔可夫模型的异常检测[J]. 小型微型计算机系统, 2004, 25(8): 1546-1549.

[11] AGRAWAL R, GEHRKE J, GINOUULOS D, et al. Automatic subspace clustering of high dimensional data[J]. Data Mining and Knowledge Discovery, 2005, 11(1): 25-33.

[12] BHUSARI V, PATIL S. Application of hidden Markov model in credit card fraud detection[J]. International Journal of Distributed & Parallel Systems, 2011, 2(6): 33-36.

[13] CHEN Y, TU L. Density-based clustering for real-time stream data[C]//Proceedings of ACM Sigkdd International Conference on Knowledge Discovery & Data Mining. New York, USA: ACM Press, 2007: 133-142.

[14] ALI M H, SUNDUS A, QAISER W, et al. Applicative implementation of d-stream clustering algorithm for the real-time data of telecom sector[C]//Proceedings of International Conference on Computer Networks & Information Technology. Washington D. C., USA: IEEE Press, 2011: 293-297.

[15] CAVALIN P R, SABOURIN R C, SUEN Y, et al. Evaluation of incremental learning algorithms for HMM in the recognition of alphanumeric characters[J]. Pattern Recognition, 2009, 42(12): 3241-3253.

[16] FLOREZ-LARRAHONDO G, BRIDGES S, HANESAN E A. Incremental estimation of discrete hidden Markov models based on a new backward procedure[C]//Proceedings of IEEE National Conference on Artificial Intelligence. Washington D. C., USA: IEEE Press, 2005: 758-763.

[17] LEE W, STOLFO S J, MOK K W. A data mining framework for building intrusion detection models[C]// Proceedings of IEEE Symposium on Security and Privacy. Washington D. C., USA: IEEE Press, 1999: 120-132.

编辑　索书志

(上接第 121 页)

[8] Scanning data for entropy anomalies[EB/OL]. [2017-07-31]. http://blog. dkbza. org/2007/05/scanning-data-for-entropy-anomalies. html.

[9] PENG F, SCHUURMANS D. Combining naive Bayes and n-gram language models for text classification[C]// Proceedings of European Conference on Information Retrieval. Berlin, Germany: Springer, 2003: 335-350.

[10] NORVIG P. Natural language corpus data[EB/OL]. [2017-07-31]. http://www. norvig. com/ngrams/ch14. pdf.

[11] WANG Wei, SHIRLEY K. Breaking bad: detecting malicious domains using word segmentation[EB/OL]. [2017-07-31]. https://arxiv. org/abs/1506. 04111 2015.

[12] 邓爱萍. 程序代码相似度度量算法研究[J]. 计算机工程与设计, 2008, 29(17): 4636-4638.

[13] 石　野,黄龙和,车天阳,等. 基于语法树的程序相似度判定方法[J]. 吉林大学学报(信息科学版), 2014, 32(1): 95-100.

[14] COMANICIU D, MEER P. Mean shift: a robust approach toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 24(5): 603-619.

[15] 张华伟,王明文,甘丽新. 基于随机森林的文本分类模型研究[J]. 山东大学学报(理学版), 2006, 41(3): 139-143.

[16] 张希翔,赵　欢. 基于随机森林的语音人格预测方法[J]. 计算机工程, 2017, 43(6): 253-258.

[17] KOHAVI R. A study of cross-validation and bootstrap for accuracy estimation and model selection[C]// Proceedings of International Joint Conference on Artificial Intelligence. New York, USA: ACM Press, 1995: 1137-1145.

[18] GOUTTE C, GAUSSIER E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation[C]//Proceedings of European Conference on Information Retrieval. Berlin, Germany: Springer, 2005: 345-359.

编辑　顾逸斐