基于频繁项集和高效用项集挖掘的银行间市场对倒交易检测

刘 丹[1] 金天成[1] 窦 亮[1] 肖春芸[1*] 詹杭龙[2] 卢艳民[2]

[1](华东师范大学计算机科学与技术学院 上海 200062)

[2](中汇信息技术(上海)有限公司 上海 201203)

摘 要 传统的银行间市场对倒交易检测采用直接建立规则的方法,忽略了对倒交易的策划性、协同性和交易主体的差异性,存在运行时间长、效率低和滞后性等问题。基于频繁项集和高效用项集挖掘找到多次共同交易的群体,结合对倒交易模式检测出对倒交易链。实验结果表明,该方法识别率高于 97%,且检测时间减少了 45%,在效率上有明显的提高,对检测对倒交易有一定的预判指导意义。

关键词 对倒交易 数据挖掘 高效用项集 频繁项集 市场操纵

中图分类号 TP3 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.12.052

THE DETECTION OF WASH SALES IN INTERBANK MARKET BASED ON FREQUENT ITEMSETS AND HIGH UTILITY ITEMSETS MINING

Liu Dan[1] Jin Tiancheng[1] Dou Liang[1] Xiao Chunyun[1*] Zhan Hanglong[2] Lu Yanmin[2]

[1](*College of Computer Science and Technology, East China Normal University, Shanghai 200062, China*)

[2](*CFETS Information Technology(Shanghai) Co., Ltd., Shanghai 201203, China*)

Abstract The traditional detection of wash sale in interbank market ignores the strongly planning, cooperation of wash sale and the difference between transaction subjects, and has the limitations of long running time, low efficiency and lag. In view of the problems, this paper introduced frequent itemsets and high utility itemsets to find out the trading groups who have participated in transaction together in many times, and detected the wash sale chain combining with its mode. The experiments show that the proposed method can recognize more than 97% of all wash sales and the detection time is reduced by 45%. The efficiency of detection is significantly improved. This work has a certain guiding significance for the detection of wash sale.

Keywords Wash sale Data mining High utility itemsets Frequent itemsets Market manipulation

0 引 言

随着我国金融改革的全面深化、金融开放水平的不断提高,金融行业逐渐向综合经营的方向发展[1],金融机构在银行间市场的参与程度越来越高,银行间市场的发展状况直接或间接地影响着整个金融行业的健康发展。对倒交易是一种典型的“自买自卖”的异常交易,其目的是震仓、吸引跟风盘或拉升股价,会严重扰乱投资者的视线,而银行间市场的对倒交易产生的影响极有可能经过银行间市场传递到金融市场、企业和个人,造成巨大的破坏力。因此,制定相关的法律法规,并利用金融科技创新技术,对交易市场进行实时监控,实现早发现、早预警、早处理,对维护市场稳定和谐具有重要意义。

目前,国家颁布了一系列法律法规来对对倒交易进行认定与管治[2],除此之外,也有相关的研究人员和机构提出了一些对倒交易相关的检测方法[3]。独道刚[4]为了识别出股票中庄家的真实交易意图,提出了一种成交数据修正算法,能够克服庄家间对倒交易的

收稿日期:2021-06-10。上海市科学技术委员会科研计划项目(18511103802)。**刘丹**,硕士生,主研领域:机器学习。**金天成**,博士。**窦亮**,副教授。**肖春芸**,讲师。**詹杭龙**,博士。**卢艳民**,硕士。

干扰,在识别庄家意图的基础上实现对股市主力资金流向的有效监测。王楠[5]选取银行间市场现券交易样本,分析市场参与者价格偏离容忍度,来监测包括对倒等其他超出价格偏离阈值的异常交易。Golmohammadi 等[6]通过决策树、朴素贝叶斯和神经网络等监督学习算法去检测股票市场中包括对倒交易在内的一些操纵市场行为。Martínez-Miranda 等[7]将强化学习框架应用到市场操纵的分析和检测中去,找出通过对倒交易等异常交易策略来操纵市场的交易者,并通过这些交易者的欺骗原因来分析潜在违法行为。

上交所、深交所于 2019 年上线新一代风险检测平台,新增舆情分析、机构画像等功能。这些检测系统往往以规则化数据统计功能为主(交易规模、交易量和差价等),根据已知的异常交易模式建立规则并设置数据阈值来检测其中的异常交易。

通过以上方法能够为监管部门取证提供技术支持,同时也有助于规范市场秩序,促进经济的健康发展。然而在新的发展要求和环境下,通过特征或者规则建立模型去做对倒交易检测的传统检测方法已经难以满足当前的需求,从效率和实用性而言,也存在局限性:

(1) 在检测过程中,忽略了对倒交易的组织性和交易主体的差异性。在市场交易中,对倒交易往往是有组织有预谋的,具有一定的群体特性,只有大量的交易频繁主体才是市场变化的直接驱动者,大量频繁的对倒交易对于金融交易市场的健康发展有着较大的风险与隐患[4,8-9],而少量无序的买卖很难改变走势。因此,充分考虑交易主体间的差异性和对倒交易的规律性很有必要。

(2) 目前缺少针对对倒交易这一重要异常交易的深入研究,对于交易数据的分析和挖掘深度还存在一些差距。《银行间债券市场债券交易检测工作方案》(中汇交发〔2009〕243 号)对异常交易进行了定义,即成交价格大幅度偏离市场公允价值、单笔报价或交易量远高于实际需求、交易双方或多方以虚增交易量为目的的对倒交易等行为,异常交易的类型多样,并具有各自的特点。当前大多数的检测手段都是以规则化数据统计功能为主,通过价格偏离度或者其他的指标去判定是否存在异常交易[5,10],但未将其定位到异常交易中的一个具体的类型,例如对倒、做尾盘和对敲等。

(3) 直接通过已知交易模式建立规则的传统检测方法需要进行多次的迭代递归,效率低下且存在无效递归的情况,并且一旦数据量急剧增加,需要消耗大量的时间和空间,算法性能表现极差。

针对以上问题,本文提出两种专门针对检测银行间市场对倒交易的算法 WSD_MFI 和 WSD_MHUI,充分考虑对倒交易的组织性和主体差异性,基于频繁项集[11]和高效用项集[12]思想,可有效甄别每一笔对倒交易的触发者和交易群体信息,识别出资金的来源、流通路径、最终去向,同时检测效率也有了很大的提升。WSD_MFI 和 WSD_MHUI 这两种算法在提取交易数据特征后,分别采用频繁项集挖掘算法和高效用项集挖掘算法,挖掘出频繁项集和高效用项集,找到多次共同进行交易的主体,并将其结合到传统的对倒交易检测方法中,最终检测出对倒交易链。通过在频繁或者高效用的交易群体间去做对倒交易的检测,既考虑到交易主体的差异性和对倒交易的群体性,有效检测出整个对倒交易链,又降低了存储空间和检测时间,提高了检测的效率。

1　相关概念和算法

1.1　对倒交易模式

对倒交易,又被称为洗售(wash sale),具体是指资金经过多方周转又回到卖方的交易,是一种通过虚增交易量来造成市场活跃假象的异常交易[6]。对倒交易常常表现为一种异常的群体共谋行为,涉及到多个交易主体有预谋的交易过程,所以往往是有组织和有策划的,具有一定的群体特性。如果一些交易主体经常在同一交易日进行操作,那么这些主体相对于其他主体而言进行对倒交易的嫌疑更大,参与对倒交易的数量和规模也更大。

常见的对倒操作有两种目的:(1) 抬高市场价格,以制造假象,吸引其他交易者跟进;(2) 压低价格使信心不足的交易者为止损而离开市场。机构对于常见对倒交易模式的认定如表 1 所示,按照交易主体的数量以及不同的交易指标,对倒交易模式主要分为两方对倒、代持、三方对倒、四方对倒、五方对倒。本文主要研究交易主体数量在 5 个及以下的对倒交易模式。

表 1　交易机构模式分析指标

模式	额度	交易日	资金流向
两方对倒	无限制	相同	A⇔B
代持	相等	相同	A⇔B
三方对倒	无限制	相同	A→B→C→A
四方对倒	无限制	相同	A→B→C→D→A
五方对倒	无限制	相同	A→B→C→D→E→A

1.2 频繁项集挖掘

关联规则是指各个项集出现的联系和规律,最早主要应用于交易数据库进行购物分析。关联规则挖掘包括挖掘频繁项集和发现强关联规则,频繁项集挖掘指挖掘出数据库中所有符合条件的项集,是发现关联规则的核心部分,早在 20 世纪 90 年代国内外学者对其进行了广泛的研究。下面以本文的研究背景为例,简要地介绍如何挖掘频繁项集。

在本文中每个项表示一个交易主体,而一个项目集合包含在同一交易日进行交易的交易主体。其相关定义如下:事务数据库 $D_1=\{T_1,T_2,\cdots,T_n\}$,$T_i=\{p_1,p_2,\cdots,p_k\}$,其中 T_i 表示在同一个交易日内进行交易的主体 p_j 的集合,不包括重复的交易主体。表 2 为一个具体的示例,事务数据库共包含 4 个交易日的数据,每个交易日有若干个交易主体,使用 p_j 表示。

表 2 事务数据库 D_1 示例

事务编号	交易主体集合
T_1	p_1,p_2,p_5
T_2	$p_2,p_4,$
T_3	p_2,p_3
T_4	p_1,p_2,p_4

定义 1 事务数据库所有交易主体的集合 A,交易主体集合 $X\subseteq A$,并且 X 中包含 k 个交易主体,则 X 称为 k 项集,X 在事务数据库 D_1 中出现的次数被称为支持数 $S(X,D_1)$。

定义 2 项集 X 在事务数据库 D_1 的支持数定义为 $S(X,D_1)=\mathrm{Count}(X,D_1)$,其中 Count( ) 表示计数,例如在表 2 的事务数据库中 $S(\{p_1,p_2\},D_1)=2$。

定义 3 给定一个最小支持数 m_{inSup},如果 $S(X,D_1)\geqslant m_{\mathrm{inSup}}$,则称 X 是频繁项集。例如,当设定 $m_{\mathrm{inSup}}=2$ 时,$X=\{p_1,p_2\}$ 是一个频繁项集。

性质 1 如果一个项集 X 是频繁项集,并且项集 $Y\subseteq X$,那么 Y 也是频繁项集。

频繁项集挖掘(Mining Frequent Itemsets,MFI)算法根据挖掘的先后次序分为深度优先搜索和广度优先搜索,FP-Growth[11] 和 Apriori[13] 分别是两类方法的经典算法。由于 Apriori 算法需要多次迭代,并且每次迭代均需要扫描数据库,而 FP-Growth 算法只需扫描原始数据两遍,并通过 FP-tree 数据结构对原始数据进行压缩,效率较高。由于 FP-Growth 算法的高效性,目前被广泛地应用在各个领域[14-19],本文也基于 FP-Growth 算法进行对倒交易的检测。

1.3 高效用项集挖掘

高效用项集挖掘(Mining High Utility Itemsets,MHUI)是针对频繁项集挖掘算法过程繁琐、内存消耗大和数据结构复杂等不足提出的一种用来挖掘高共现频率特征的新方法。高效用项集算法用效用值表示项目的重要性和项目间关联性,它不仅考虑一个项集在事务中是否出现,并且能表示项集出现的次数。通过在交易数据中挖掘高效用项集,可以得到更具有针对性的信息,为下游数据分析提供依据,目前被应用在各个领域[20-21]。

结合本文背景,高效用项集挖掘相关定义如下:事务数据库 $D_2=\{T_1,T_2,\cdots,T_n\}$,其中 T_i 表示同一个交易日进行交易的主体的集合,在同一交易日进行多次交易的主体在事务数据库中也会出现相应的次数,$T_i=\{p_1,p_2,\cdots,p_k\}$,其中 p_j 为交易主体,可重复出现。例如表 3 中共包含 4 个交易日的数据,每个交易日有若干个交易主体,使用 p_j 表示。

表 3 事务数据库 D_2 示例

事务编号	交易主体集合
T_1	p_1,p_1,p_5,p_5
T_2	p_2,p_4
T_3	p_2,p_3
T_4	p_1,p_2,p_2,p_4

定义 4 项集 X 在事务中的效用值指 X 中项在事务中出现次数的和,可被定义为 $U(X,T_i)=\mathrm{Count}(X,T_i)$,其中 Count( ) 表示计数,例如在表 3 中,$U(\{p_1,p_5\},T_1)=4$。

定义 5 项集 X 在事务数据库中的效用值指的是 X 在每个事务的效用和,可被定义为 $U(X,D_2)=\sum_{i}^{n}U(X,T_i)$,例如表 3 的事务数据库中 $U(\{p_2,p_4\},D_2)=0+2+0+3=5$。

定义 6 在一个事务数据库 D_2 中,给定一个最小效用值 m_{inUti},如果 $U(X,D_2)\geqslant m_{\mathrm{inUti}}$,则称 X 是高效用项集。例如,当设定 $m_{\mathrm{inUti}}=2$,$X=\{p_2,p_4\}$ 是一个高效用项集。

性质 2 如果一个项集 X 是高效用项集,并且项集 $Y\subseteq X$,那么 Y 也是高效用项集。

高效用项集挖掘一般采用两种方法,(1) 通过构造和支持度类似的属性值,然后借助频繁项集挖掘算法展开;(2) 首先生成候选项集,然后在过程中采用剪枝策略进行剪枝,最后对候选项目集进行测试,典型的

算法有 Two_phase[22]、IHUP[23]、HUI-miner[12]、FHM[24]等,其中 FHM 算法较于前三者效率更高,本文使用 FHM 算法挖掘出高效用频繁项集,用于对倒交易的检测。

2　对倒交易检测算法

2.1　传统的对倒交易检测算法

基于已知交易模式建立模型进行检测的思想,传统的对倒交易检测算法(Wash Sales Detection,WSD)主要分为三个步骤。

(1) 按照交易日期将所有的交易记录分组。首先按照“交易日”与“结算日”将交易记录分成多个组,“交易日”与“结算日”相同的交易记录分到同一组,得到“交易日”与“结算日”相同的各组交易记录。

(2) 遍历每个组,跟踪每组的资金流向。从“交易日”与“结算日”相同的各组交易记录中找出对倒交易,分别以同组记录的一条交易记录作为“起点”,以它的拆出方为“源”,跟踪它所借出的资金是否会在同一个交易日、同一个结算日回到该拆出方,并记录下资金流。

(3) 跟踪资金流向,检测对倒交易。在跟踪借出资金的资金流时,遍历交易时间在上一交易记录之后的交易,判断上一交易记录的拆入方与当前遍历的拆出方是否相同,如果相同,则将资金流涉及到的公司总数加 1。接着,判断“起点”交易记录与当前遍历的拆入方是否相同,如果相同,就找到一条对倒交易,加入到对倒交易的集合中,并在当前这条对倒交易的基础上再去找是否有更长的对倒交易序列;否则将当前遍历的拆出方加入到当前交易链路的末端,再去找该条交易链是否存在对倒交易。

下面以伪代码的形式给出对倒交易检测算法,如算法 1 所示。

算法 1　传统的对倒交易检测算法

输入:交易序列,包括交易时间、拆入方、拆出方。
输出:对倒交易。

```
Detect_Sales_in_one_day(sales_in_one_day)
BEGIN
    //遍历同一交易日的交易记录
FOR sale IN sales_in_one_day THEN
    temp_source = [sale.拆出方,sale.拆入方]
            //以当前交易为起点,跟踪资金流向
     Follow_Wash_Sale(sales_in_one_day,sale,temp_source)
    END FOR
END
Follow_Wash_Sale(sales_in_one_day,sale,temp_source)
BEGIN
  FOR t_sale IN sales_in_one_day THEN
    IF t_sale.order > sale.order THEN              //交易次序
      //资金发生流动
      IF sale.拆入方 == t_sale.拆出方 THEN
      //对资金流涉及的主体进行计数
      temp_source.公司数 = temp_source.公司数 + 1
  //资金经过多次周转又流入源交易主体,检测到对倒交易链
      IF temp_source.源公司 == t_sale.拆入方 THEN
              //记录交易主体数量在五方及以下的对倒交易
       IF temp_source.公司数 < 5 THEN
        temp_source.add(sale.拆入方)
        wash_sales.add(temp_source)
       //在该对倒交易链上去检测是否有更长的对倒交易链
        Follow_Wash_Sales(sales_in_one_day,t_sale,temp_
source)
       END IF
      ELSE THEN
 //当前未形成对倒交易回路,继续寻找以当前交易链为前缀的
                                          //对倒交易
      IF temp_source.公司数 < 5 THEN
       Follow_Wash_Sales(sales_in_one_day,t_sale,temp_
source + t_sale.拆入方)
      END IF
      END IF
      END IF
     END IF
    END FOR
END
MAIN():
    BEGIN
    FOR sales_in_one_day IN all_sales THEN
          //同一交易日的交易记录进行对倒交易检测
          Detect_Sales_in_one_day(sales_in_one_day)
    END FOR
END
```

对倒交易检测算法,需要以一个固定的拆出方为源,探索它借出的资金经过多家银行后是否会在同一个交易日、同一个结算日回到该拆出方,以此来筛选对倒交易(本文最多考虑 5 方对倒)。所以,直接根据对倒规则进行检测的算法有以下两个缺点。

(1) 需要多次迭代递归,具有很高的时间复杂度,在处理海量的交易数据时,会消耗大量的内存开销和时间。

(2) 徒劳无功的可能性较大,一方面,在所有的交易主体中检测,而参与对倒交易的主体一般来讲只是小部分,存在无效迭代的情况;另一方面,检测到的对

倒交易序列可能是两个交易不频繁的小额交易,对于交易市场并没有多大的影响。

2.2 基于频繁项集的对倒交易检测算法

本文考虑到对倒交易的群体特性,为了找出多次共同交易的交易群体,基于频繁项集的思想,将频繁项集挖掘算法 FP-Growth 应用在对倒交易的检测中。

基于频繁项集的对倒交易检测算法主要分为两个步骤:

(1) 在所有的交易记录中筛选出除了频繁一项集外的频繁项集中项参与的交易记录作为频繁交易记录,这是因为频繁一项集只有一个项,不具有群体特性。为了便于描述,下文中将除了频繁一项集外的频繁项集中项的集合称为群体频繁项。由于 1.2 节中的性质 1 可知,频繁二项集包含所有的群体频繁项,所以只需要通过 FP_Growth 算法挖掘出频繁二项集即可。

(2) 将 WSD 算法应用在频繁交易记录中,得到对倒交易链。伪代码如算法 2 所示。

算法 2 基于频繁项集的对倒交易检测算法

输入:交易序列,包括交易时间、拆入方、拆出方、群体频繁项 **frequent_items**。

输出:对倒交易。

```
Wash_sale_FP-Growth(frequent_items)
BEGIN
  FOR sale IN all_sales THEN
      //筛选出频繁交易记录
    IF sale.拆入方 IN frequent_items AND sale.拆出方 IN frequent_items THEN
      frequent_sales.add(sale)
    END IF
  END FOR
      //对倒交易检测,算法 1
      WSD(frequent_sales)
END
```

基于频繁项集的对倒交易检测算法(WSD_MFI)从大量的交易数据中挖掘出有关联规则的频繁交易对象,并在这些交易对象的交易记录去检测是否存在对倒交易,有目的和有针对性地去做检测,可以有效提高检测的效率,但是其性能局限于频繁项集挖掘算法例如 Apriori、FP-Growth 等,所以在处理大量的交易数据下,内存消耗比较大。

2.3 基于高效用项集的对倒交易检测算法

本文进一步考虑主体差异性,将高效用项集挖掘算法 FHM 应用在对倒交易的检测中,既考虑交易主体在事务中出现的次数,对交易主体进行区分,又可以有效地改进基于频繁项集检测对倒交易算法的效率。

基于高效用项集的对倒交易检测算法主要分为两个步骤:

(1) 考虑群体特性和主体差异性,在所有的交易记录中筛选出除了高效用一项集外的高效用项集中项参与的交易记录作为高效用交易记录。为了描述,下文中将除了高效用一项集外的高效用项集中的项称为群体高效用项。同样由 1.3 节中的性质 2 可知,只需要挖掘出高效用二项集,便可获得所有的群体高效用项。

(2) 将 WSD 算法应用在高效用交易记录中,得到对倒交易链。伪代码如算法 3 所示。

算法 3 基于高效用项集的对倒交易检测算法

输入:交易序列,包括交易时间,拆入方和拆出方,群体高效用项 **high_utility_items**。

输出:对倒交易。

```
Wash_Sale_MHUI(high_utility_items)
BEGIN
      //筛选出高效用交易记录
  FOR sale IN all_sales THEN
      IF sale.拆入方 IN high_utility_items AND sale.拆出方 IN high_utility_items THEN
        high_utility_sales.add(sale)
      END IF
  END FOR
        //对倒交易检测,算法 1
        WSD(high_utility_sales)
END
```

基于高效用项集的对倒交易检测算法(WSD_MHUI)考虑交易主体在同一个交易日的交易次数,并从大量的交易数据中挖掘出共现频率高的交易群体,然后在这些交易对象的交易记录中去检测是否存在对倒交易,可以有效提高检测的效率,并且在处理大量的交易数据时具有较高的效率。

3 实验与分析

3.1 实验数据集

实验采用的数据集是基于银行间市场信用拆借业务场景模拟得到的某年全年的交易数据(以下简称为数据集 IBO2016),参与交易的金融机构有 1 073 家,属性数为 76,交易数量 11 万余条,数据样式如表 4 所示(只列出部分属性,数据已脱敏)。信用拆借交易是指与中国外汇交易中心暨全国银行间同业拆借中心(以下简称为交易中心)联网的金融机构之间通过交易中心的交易平台进行的无担保资金融通行为。

表 4　数据样式

交易日	拆入方	拆出方
2016/1/4	**1 号定向计划	** 信托 1 号
2016/1/4	** 一年定期开放混合基金	** 债券
2016/1/4	** 证券股债双赢一号	** 灵活配置混合基金

3.2　实验环境和参数设定

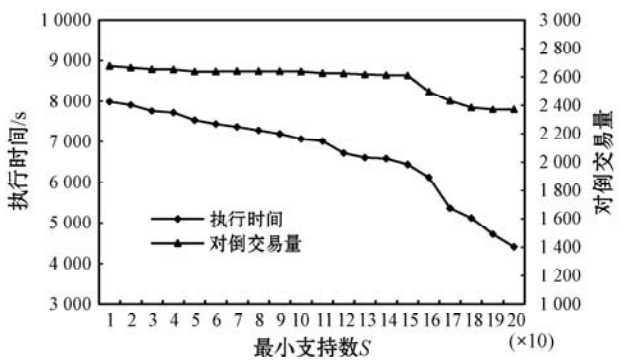
使用 Python 编程语言实现本文提出的两种算法,实验环境参数如表 5 所示。

表 5　实验环境参数

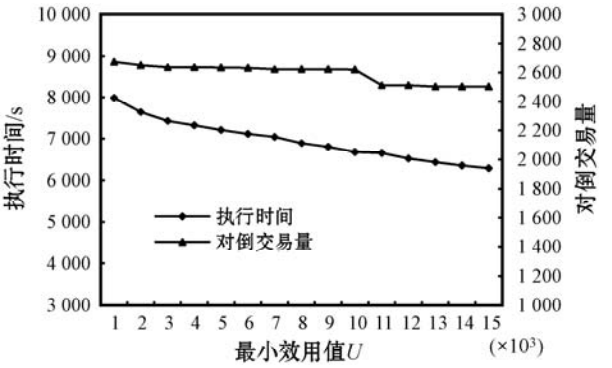
设备	参数
CPU	AMD EPYC 7K62 48-Core Processor
内存/GB	8
操作系统	Linux

(1) 基于频繁项集的对倒交易检测算法的 S 参数设定。结合本文的研究背景可以发现,在基于频繁项集的对倒交易检测算法中,如果最小支持数 S 设定越大,发现的交易群体对倒交易的嫌疑越大,但是群体的规模和数量也会随之减少,那么检测到的对倒交易量也会降低,运行时间也会减少;反之,S 设定越小,发现的交易主体参与对倒交易的嫌疑越小,但群体规模和数量会增大,检测到的对倒交易量也会增多,运行时间也会相应增加。所以,需要设定合理的 S,以发现一定数量且对倒交易嫌疑较高的交易群体,确保能在较短的时间内检测出更多的对倒交易。

图 1(a)为在不同最小支持数 S 下 WSD_MFI 的执行时间和识别的对倒交易数量。可以看出,当 S 设定小于 150,检测出的对倒交易量较多,但是执行时间没有明显减少,说明项集规模较大,S 的设定过于宽松;当 S 大于 150,执行时间降低,但是检测出的对倒交易量明显减少,说明项集规模过小,S 的设定过于严格;而当 S 等于 150,可以看出 WSD_MFI 在较短的时间内检测出了一定规模的对倒交易,说明得到了适当规模的项集,S 取值合理,所以将 S 的取值设定为 150。



(a) 基于频繁项集的对倒交易检测算法



(b) 基于高效用项集的对倒交易检测算法

图 1　在不同支持数和不同效用值下的执行时间和对倒交易量

(2) 基于高效用项集的对倒交易检测算法的 U 参数设定。同理,在基于高效用项集的对倒交易检测算法中也需要设定合理的 U。图 1(b)为在不同最小效用值下 WSD_MHUI 的执行时间和识别的对倒交易数量。可以看出,当 U 设定为 10 000 时,WSD_MHUI 可以在较短的时间内检测出一定规模的对倒交易,U 取值合理,所以实验中将 U 的取值设定为 10 000。

3.3　实验结果分析

数据集中交易主体共 1 073 个,其中群体频繁交易主体 151 个,群体高效用交易主体 142 个;交易记录共 111 020 条,其中群体频繁交易主体参与的交易即频繁交易 80 002 条,群体高效用交易主体参与的交易即高效用交易记录 80 053 条。使用 WSD、WSD_MFI 和 WSD_MHUI 三种算法检测出的对倒交易分别为 2 686 条、2 610 条和 2 621 条,部分对倒交易资金链例子如表 6 所示(数据已脱敏)。

表 6　检测出的对倒交易例子

交易日	对倒交易资金链
2016/1/11	** 基金国际-bc→ **2 号特定单一客户→ ** 基金国际-bc
2016/1/12	** 银行→ ** 信托卓越 10 号→ ** 银行
2016/1/12	** 银行→ ** 信托卓越 10 号→ ** 银行→ ** 灵活配置混合基金→ ** 银行

根据实验统计,群体频繁主体参与的对倒交易条数占总对倒交易量的 97.19%,符合对倒交易往往是交易群体预谋组织的特点;其次,群体高效用主体参与的对倒交易条数占总对倒交易量的 97.58%,占比高于前者,这说明考虑到主体差异性,即如果群体中交易主体参与的交易越多,参与对倒交易的概率和数量也越大,与前文描述的对倒交易特点相一致。

(1) 对倒交易检测性能分析。为了验证本文算法

的高效性,即尽可能在较短的时间内发现较多的对倒交易,本文对比三种算法的执行时间降低率和对倒交易的识别率。假设 T 是检测的执行时间,W^* 是检测出的对倒交易数量,W 是全部的对倒交易量,那么算法的识别率 P 计算式如下:

$$P=\frac{W^*}{W}\times 100\% \tag{1}$$

由表 7 可以看出,WSD_MFI 算法的执行时间较 WSD 方法降低了 48.7%,检测出了 2 610 条对倒交易,识别率达到了 97.17%,而 WSD_MHUI 算法执行时间降低了 46.8%,识别率达到 97.58%,两种算法的检测时间降低至将近传统方法的一半,而识别出的对倒交易量几乎相同,漏检率不到 3%。在实际应用中,这样的识别率可以接受,一方面由本文的研究背景可知,多次进行对倒交易的交易主体参与的对倒交易是有组织有预谋的,这些具有群体性的对倒交易才会影响金融市场的发展,基于频繁项集和高效用项集的检测方法检测出的对倒交易本身具有组织性和频繁性,而未被识别出的对倒交易不具备这种性质;另一方面,在追求实时性的检测场景下,可以允许少量的对倒交易漏检。

表 7 算法性能对比结果

算法	W	T/s	$\Delta T/T$/%	P/%
WSD	2 686	12 559.3	—	—
WSD_MFI($S=150$)	2 610	6 440.0	48.7	97.17
WSD_MHUI($U=10\ 000$)	2 621	6 676.7	46.8	97.58

(2) 频繁二项集和高效用二项集挖掘性能对比。本文提出的两种算法基于频繁项集和高效用项集,且只需要挖掘出频繁二项集和高效用二项集即可,现在以上实验的基础上,进一步对挖掘出频繁二项集和高效用二项集的 FP-Growth 和 FHM 算法的性能(内存消耗和执行时间)进行比较。为了比较在不同规模数据集上的内存消耗和执行时间,本文将 IBO2016 数据集作为较大规模的数据集;IBO2016 中第一季度的交易数据作为小规模的数据集 IBO2016_Q1,该数据集共 24 197 条交易数据,交易主体 717 个,属性数有 76 个,经过实验,在该数据集上合理的 S、U 分别为 17、1 900。其对比结果如表 8 和表 9 所示。

表 8 在数据集 IBO2016_Q1 上的算法性能对比结果

算法	内存/MB	执行时间/ms
FP-Growth($S=17$)	339	2 448
FHM($U=1\ 900$)	332	167

表 9 在数据集 IBO2016 上的算法性能对比结果

算法	内存/MB	执行时间/ms
FP-Growth($S=150$)	624	119
FHM($U=10\ 000$)	260	423

表 8 是在 IBO2016_Q1 数据集上挖掘频繁二项集和高效用二项集的执行时间和使用的内存消耗,可以看出,在数据集较小的时候,挖掘高效用二项集的执行时间远低于频繁二项集,内存消耗两者相差无几。在较大规模数据集 IBO2016 上的对比结果如表 9 所示,高效用二项集挖掘算法的内存消耗远小于频繁二项集挖掘算法,其执行时间高于频繁二项集挖掘算法。

综合在两个数据集上的对比可知,在少量数据的情况下,高效用二项集挖掘算法的执行时间也远低于频繁二项集,而在处理大量数据时,前者的执行时间高于后者;在内存消耗上,前者均小于后者,且这种现象在大规模数据集上更为明显。

本文认为基于高效用项集的对倒交易检测方法更具有应用价值,这是因为银行间市场往往会产生大规模的交易数据,在内存有限的硬件系统上减小内存占用量是非常重要的;另一方面,合理的 U 和 S 值往往较大,例如,本文实验在 IBO2016 数据集上 $S=150$,$U=10\ 000$,所以短时间内的交易数据的变化并不会引起频繁项集和高效用项集的变化,遂只需要对数据库中的频繁项集和高效用项集进行定期更新,而不是在每次检测前都去挖掘新的频繁二项集和高效用二项集,所以少数次执行的少量时间差异可以接受。

4 结 语

在新的发展要求和形势下,通过监管技术提高银行间市场的监管能力势在必行。本文在传统的对倒交易检测方法的基础上提出基于频繁项集的对倒交易检测方法和基于高效用项集的对倒交易检测方法,经过在银行间市场的交易数据上的测试,本文方法相对于传统的对倒交易检测算法具有更强的实用价值,在检测效率上有显著的提高。基于频繁项集的对倒交易检测方法和基于高效用项集的对倒交易检测方法,既考虑到交易主体的差异性,又降低了无关数据带来的运行成本,使得检测更有针对性,运行效率得到明显的提升。其中,高效用项集挖掘算法较频繁项集挖掘算法的内存消耗更小,所以在处理大量的交易数据时,基于高效用项集的对倒交易检测算法效率提升更加明显,更具有应用价值。

由于银行间市场交易是动态的,在后续工作中,将探索在交易数据日益增长的情况下,如何实现高效用项集的自动维护与更新,进一步提高方法的实用性。

参 考 文 献

[1] 李政,梁琪,方意. 中国金融部门间系统性风险溢出的监测预警研究——基于下行和上行 ΔCoES 指标的实现与优化[J]. 金融研究,2019(2):40 - 58.

[2] 蔡奕. 解读《证券法》关于市场操纵的法律规范[J]. 证券市场导报,2005(5):19 - 23.

[3] 刘凤元,陈俊芳. 股票价格操纵研究与政策建议[J]. 价格理论与实践,2003(7):52 - 53.

[4] 独道刚. 规划识别在监测股市个股主力资金流向中的应用[D]. 镇江:江苏科技大学,2011.

[5] 王楠. 银行间市场现券交易价格偏离度研究——以中小型商业银行为例[J]. 金融理论与教学,2020(1):56 - 59,62.

[6] Golmohammadi K, Zaiane O, Díaz D. Detecting stock market manipulation using supervised learning algorithms[C]// 2014 International Conference on Data Science and Advanced Analytics,2014:435 - 441.

[7] Martínez-Miranda E, McBurney P, Howard M. Learning unfair trading: A market manipulation analysis from the reinforcement learning perspective[C]//2016 IEEE Conference on Evolving and Adaptive Intelligent Systems,2016:103 - 109.

[8] 廖志良,潘文娣. 我国证券业洗钱风险分析与防范措施[J]. 海南金融,2010(5):55 - 57.

[9] 刘俊. 西藏矿业:对倒交易分析师当"托"[J]. 股市动态分析,2009(3):14,39.

[10] 金升平,刘钊. 银行异常交易检测方法研究[J]. 武汉金融,2018(2):61 - 66.

[11] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation[C]//2000 ACM SIGMOD International Conference on Management of Data,2000:1 - 12.

[12] Liu M, Qu J. Mining high utility itemsets without candidate generation[C]//21st ACM International Conference on Information and Knowledge Management,2012:55 - 64.

[13] Agrawal R, Imielinski T, Swami A. Database mining: A performance perspective[J]. IEEE Transactions on Knowledge and Data Engineering,1993,5(6):914 - 925.

[14] Zhang F, Zhao X, Li Y, et al. Based on FP-Growth algorithm to excavate medication rule of Chinese materia medica for radiation esophagitis[J]. World Journal of Integrated Traditional and Western Medicine,2020,6(7):31 - 38.

[15] 何望,林果园. 基于 FP-Growth 改进算法的云服务器故障数据分析[J]. 计算机工程与科学,2020,42(5):770 - 775.

[16] Jia Y, Liu L, Chen H, et al. A Chinese unknown word recognition method for micro-blog short text based on improved FP-growth[J]. Pattern Analysis and Applications,2019,23:1011 - 1020.

[17] 黄伟,李国和,吴卫江,等. 基于 FP_Growth 的消费行为关联分析系统设计与实现[J]. 计算机应用与软件,2015,32(8):34 - 37,79.

[18] Van L, Huong P, Thuan L, et al. Improving the feature set in IoT intrusion detection problem based on FP-Growth algorithm[C]//2020 International Conference on Advanced Technologies for Communications,2020:18 - 23.

[19] 刘琰,张进,陈静,等. 基于最大频繁项集挖掘的微博炒作群体发现方法[J]. 计算机工程与应用,2017,53(4):90 - 97.

[20] 吴玉佳,李晶,宋成芳,等. 基于高效用神经网络的文本分类方法[J]. 电子学报,2020,48(2):279 - 284.

[21] 穆晓芳,邓红霞,郭虎升,等. 基于快速高效用项集挖掘的大规模消息流预测算法研究与应用[J]. 计算机应用与软件,2019,36(11):243 - 249.

[22] Liu Y, Liao W, Choudhary A. A two-phase algorithm for fast discovery of high utility itemsets[C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining,2005:689 - 695.

[23] Ahmed C, Tanbeer S, Jeong B, et al. Efficient tree structures for high utility pattern mining in incremental databases[J]. IEEE Transactions on Knowledge and Data Engineering,2009,21(12):1708 - 1721.

[24] Fournier-Viger P, Wu C, Zida S, et al. FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning[C]//International Symposium on Methodologies for Intelligent Systems,2014:83 - 92.

(上接第 359 页)

[11] Kosba A, Miller A, Shi E, et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts[C]//IEEE Symposium on Security and Privacy,2016:839 - 858.

[12] Cachin C, Vukolic M. Blockchain consensus protocols in the wild[EB]. arxiv:1707.01873,2017.

[13] Hazari S, Mahmoud Q H. A parallel proof of work to improve transaction speed and scalability in blockchain systems[C]//9th Annual Computing and Communication Workshop and Conference,2019:916 - 921.

[14] Asif W, Lestas M, Qureshi H K, et al. Spectral partitioning for node criticality[C]//IEEE Symposium on Computers and Communication,2015:877 - 882.

[15] Fiedler M. Algebraic connectivity of graphs[J]. Czechoslovak Mathematical Journal,1973,23(98):298 - 305.