·先进计算与数据处理· 文章编号：1000-3428(2019)07-0006-07 文献标志码：A 中图分类号：TP391

一种高效的分布式水军群组检测算法

张 璐[1]，朱海婷[2]

（1. 南京财经大学 信息工程学院，南京 210046；2. 南京邮电大学 物联网学院，南京 210023）

摘 要： 为在电子商务水军群组检测中快速处理真实环境中的大规模用户数据，提出一种分布式水军群组检测算法。设计基于余弦模式挖掘的候选群组提取算法，通过余弦相似度衡量群组成员间的耦合性，以精准提取候选群组并降低后续识别的计算量。结合组投影技术与 Spark 计算框架，提出一种分布式群组提取算法，从而提升群组检测的运行速度。在真实数据集上的实验与案例研究结果表明，该算法能够保证检测准确率，且具有较高的运行效率。

关键词： 水军群组检测；检测效率；余弦模式；紧耦合群组；组投影；分布式计算框架

中文引用格式： 张璐，朱海婷. 一种高效的分布式水军群组检测算法[J]. 计算机工程，2019，45(7)：6-12.

英文引用格式： ZHANG Lu，ZHU Haiting. An efficient distributed detection algorithm for spammer group[J]. Computer Engineering，2019，45(7)：6-12.

An Efficient Distributed Detection Algorithm for Spammer Group

ZHANG Lu[1]，ZHU Haiting[2]

（1. College of Information Engineering，Nanjing University of Finance and Economics，Nanjing 210046，China；
2. College of Internet of Things，Nanjing University of Posts and Telecommunications，Nanjing 210023，China）

【Abstract】 In order to quickly process large-scale user data in real environment in e-commerce spammer group detection，a distributed detection algorithm for spammer group is proposed. A candidate group extraction algorithm based on cosine pattern mining is designed to measure the coupling between group members by cosine similarity，so as to extract candidate groups accurately and reduce the computational complexity of subsequent recognition. Combining group projection technology with the Spark computing framework，a distributed group extraction algorithm is proposed to further improve the speed of group detection. Results of experiments and case studies on real data sets show that the proposed algorithm can guarantee the detection accuracy and has high efficiency.

【Key words】 spammer group detection；detection efficiency；cosine pattern；tightly-coupled group；group projection；distributed computing framework

DOI： 10.19678/j.issn.1000-3428.0052048

0 概述

随着电子商务与 Web 2.0 的快速发展，在线评论已成为消费者在购买产品或服务时的重要参考依据[1]。与此同时，大量水军通过发布虚假评论影响用户的购买决策，从而攫取不正当的商业利益[2]。虽然已有研究试图通过异常的评论特征、行为特征等对水军进行检测[3-5]，但是，近年来越来越多的水军通过组成群组的方式，共同发布针对目标商品的虚假评论。在这种有组织的行动下，水军个体能够以更加轻微的异常行为隐藏自己的身份，且由于群组中包含大量水军，往往能够形成言论优势，进而完全控制关于目标商品的评价[6]。水军群组对社会产生了巨大危害，其检测问题成为广受关注的研究热点。

主流水军群组检测方法一般包含 2 个阶段[7]：提取若干候选群组（提取阶段），从候选群组中识别出水军群组（识别阶段）。现有研究工作主要聚焦于检测方法准确率的提升[8-9]，较少关注检测效率等性能问题。然而，随着电子商务平台中用户和商品规模的急剧扩张，检测方法的效率与扩展性成为限制其广泛应用的瓶颈。水军群组识别阶段主要使用分

基金项目： 国家重点研发计划（2017YFD0401002）；国家自然科学基金（71801123，91646204，61502250）；南京邮电大学引进人才科研启动基金（NY214188）。

作者简介： 张 璐（1983—），男，讲师、博士，主研方向为数据挖掘、分布式计算；朱海婷（通信作者），讲师、博士。

收稿日期： 2018-07-09 **修回日期：** 2018-08-29 **E-mail：** luzhang@njue.edu.cn

类、排序等机器学习算法,已有研究多数致力于解决这些算法的效率问题[10-11],且目前很多分布式框架中已集成了成熟的机器学习工具包(如 Hadoop 中的 Mahout、Spark 中的 MLlib 等)。另一方面,识别算法的运行速度主要受待识别样本量的影响,如果提取阶段产生的候选群组过多,且其中包含大量的非水军群组,将会增加识别阶段的压力,导致算法整体检测性能降低。

综上,精准且高效的候选群组提取是提升水军群组检测算法整体性能的关键。为此,本文提出一种基于余弦模式挖掘的群组提取算法,以精准提取候选水军群组。在此基础上,利用 Spark 框架设计基于分布式余弦模式挖掘的群组提取算法,以进一步提升群组提取的效率、扩展性以及算法的整体性能。

1　相关工作

进行水军群组检测的第 1 步是从所有用户中提取群组(提取阶段)。在电子商务水军检测领域,群组即一组多次针对不同目标商品共同发布评论的用户。文献[7]提出一种基于频繁模式挖掘的群组提取方法,该方法将共同评论某一商品的用户视为关联规则中的一条事务,进而将群组提取转化为从多个事务中挖掘多次出现的用户组合,相当于从事务集中挖掘频繁模式。由于该群组提取方法简单易行,因此被很多后续的水军群组检测研究所沿用[8-9],成为群组提取的主流方法。

利用频繁模式挖掘提取出的群组中既包含水军群组,也包含由于某些特殊原因偶然形成的群组(如拥有共同的兴趣爱好、同时购买流行商品等)。因此,此时提取出的群组仅仅是水军群组的候选,还需进一步构建群组特征,利用分类、排序等机器学习方法从中识别出真正的水军群组(识别阶段)。目前,水军群组识别工作主要基于以下 3 个思路:

1)构建群组特征与分类器,将候选群组分类为水军群组和非水军群组,如文献[8]基于群组评论相似性、群组规模等特征,利用 k 近邻分类器对候选群组进行分类。

2)利用群组特征建立关联关系网络,然后通过图排序算法对候选群组的受怀疑程度或恶意性进行排序,再将可疑度或恶意性高于预设阈值的群组判定为水军群组。如文献[7]建立群组-用户、群组-商品以及用户-商品 3 种关联关系并融合成关系网络,然后利用类似 PageRank 的排序算法计算群组恶意性,从而获得水军群组。

3)提取群组特征或群组目标商品的特征,将每种特征转化为分值并按一定的方式进行加权,根据最终的分值识别水军群组。如文献[9]利用主成分分析对有关群组恶意程度的主要特征值进行加权,根据最终加权分值判定水军群组。

在群组提取阶段,如何高效地从大规模用户中提取群组,且提取出的群组足够精确,即尽可能多地排除非水军群组,减轻识别阶段的计算量,对于提升水军群组检测的整体性能至关重要,本文将围绕这一问题展开研究。需要说明的是,由于识别阶段所采用的多种机器学习算法已有较成熟的分布式工具包,这一阶段的算法性能分析暂不纳入本文的研究范围。

2　基于余弦模式挖掘的群组提取

利用频繁模式挖掘提取群组时需设置支持度参数,即定义构成群组的用户需同时出现的次数。为避免遗漏水军群组,通常会设置较低的支持度,由此带来的负面影响是提取出的群组过多,可能包含大量的非水军群组,为后续水军群组的识别带来困难。造成这一现象的原因是频繁模式挖掘仅考虑用户间的共现次数,忽略了对用户间关系的度量。从内部看,群组成员间不仅有共同评论的商品,各成员也有各自独立评论的商品,即非共同评论商品。对一个群组而言,成员共同评论的商品与被群组成员评论过的商品总和间的比值衡量了群组成员间的紧密程度,本文用耦合度表示这一概念。相关研究结果表明,耦合度较高的群组(下文称为紧耦合群组)属于水军群组的概率更高[12]。如图 1 所示,水军群组成员以集体行动为主,共同评论的商品比例较高,而由于兴趣聚合而成的群组,成员往往还有各自购买/评论的需求和偏好,独立评论的商品比例较高。将群组的耦合度纳入群组提取过程中进行考虑,将更具针对性并能够过滤掉大量偶然形成的群组。本文从紧耦合群组的定义开始,利用余弦模式对其进行建模,设计基于余弦模式挖掘的紧耦合群组提取算法。

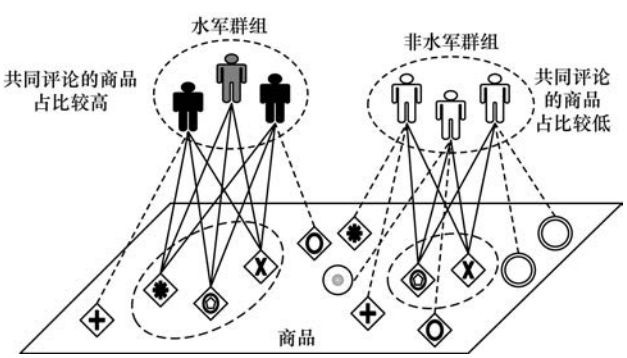

图 1　水军群组与非水军群组示意图

2.1 紧耦合群组

水军群组由一群多次共同评论特定商品或商品集合的用户所组成。令 $U=\{u_1,u_2,\cdots,u_n\}$ 表示一组用户的集合，P_l 表示用户 u_l 所评论的商品集合。若 U 中各用户所评论的商品集合完全相同，即 $P_1=P_2=\cdots=P_n$，则称这样的用户集合为完全耦合群组，并有：

$$\frac{\left|\bigcap_{l=1}^{n}P_l\right|}{|P_1|}=\frac{\left|\bigcap_{l=1}^{n}P_l\right|}{|P_2|}=\cdots=\frac{\left|\bigcap_{l=1}^{n}P_l\right|}{|P_n|} \tag{1}$$

其中，$|\cdot|$ 表示集合中元素的数量。定义用户 u_l 的共同评论率 $R_l=\left|\bigcap_{l=1}^{n}P_l\right|/|P_l|$，其表示群组中所有用户共同评论的商品数量与用户 u_l 所评论的商品数量间的比值。式(1)表明完全耦合群组中用户的共同评论率都为 1。根据这一思路，可以用共同评论率作为水军群组的衡量指标。由于在实际情况中，完全耦合群组出现的概率极低，因此需对共同评论率为 1 这一要求进行必要的松弛。在此之前，先给出平均共同评论率这一概念的定义。

定义 1（平均共同评论率）给定一个用户集合 $U(|U|\geqslant 2)$，其平均共同评论率 AR 定义为集合中所有用户的共同评论率的几何平均数，即：

$$AR(U)=\sqrt[|U|]{\prod_{u_l\in U}R_l} \tag{2}$$

由式(2)可以看出，平均共同评论率的取值在 0~1 之间，其值越高，说明用户间共同评论的商品所占比例越大，用户间的耦合度也越高。因此，紧耦合群组可视为一个平均共同评论率 $AR\geqslant\tau_{AR}$ 的用户集合，其中，τ_{AR} 表示预设的阈值。

此外，为过滤掉部分共同评论的商品数量过少的用户集合，还需对共同评论商品的最小数量进行限制。为适应不同规模的商品集合，本文用共同评论商品占商品总量的比例描述这一限制，即要求用户集合 $U(|U|\geqslant 2)$ 满足如下条件：

$$MS(U)=\frac{|P_1\cap P_2\cap\cdots\cap P_{|U|}|}{m}\geqslant\tau_{MS} \tag{3}$$

其中，m 表示商品总量，τ_{MS} 为预设的阈值。由此，可得到紧耦合群组的定义。

定义 2（紧耦合群组）当且仅当给定阈值 τ_{AR}、$\tau_{MS}\in[0,1]$ 满足 $AR(U)\geqslant\tau_{AR}$ 且 $MS(U)\geqslant\tau_{MS}$ 时，用户集合 U 为紧耦合群组。

特别地，当 $\tau_{AR}=1$ 时紧耦合群组将转化为完全耦合群组。所有群组成员的评论商品完全相同，这表明紧耦合群组是完全耦合群组的一种松弛版本，两者的区别随着 τ_{AR} 值的增大而减小。

2.2 紧耦合群组的余弦模式建模

在对紧耦合群组进行建模前，先给出余弦模式的定义。余弦模式是频繁模式的变种，除采用支持度外，还引入余弦相似度作为模式的兴趣度度量指标[13]。假设 $I=\{i_1,i_2,\cdots,i_k\}$ $(k\geqslant 2)$ 为事务集 D 中的一个 k 项集，I 的余弦相似度定义为：

$$Cos(I)=\frac{Supp(I)}{\sqrt[k]{\prod_{p=1}^{k}Supp(\{i_p\})}} \tag{4}$$

其中，$Supp(\cdot)$ 表示支持度，其计算公式为：

$$Supp(I)=\frac{SC(I)}{|D|} \tag{5}$$

其中，$SC(I)=|\{T|I\in T,T\in D\}|$ 表示 I 的支持度计数，T 为事务集 D 中包含项集 I 的事务。根据余弦相似度，可给出余弦模式的定义。

定义 3（余弦模式）分别给定支持度阈值 $\tau_s\in[0,1]$ 和余弦相似度阈值 $\tau_c\in[0,1]$，如果项集 I 满足 $Supp(I)\geqslant\tau_s$ 且 $Cos(I)\geqslant\tau_c$，则 I 为 τ_s 和 τ_c 约束下的余弦模式。

在定义 3 中，如果将 τ_c 设置为 0，则余弦模式退化为频繁模式，此时支持度成为唯一的兴趣度度量指标。如果将评论某一特定商品的用户集合视为一条事务，那么每个商品将产生一条事务并共同组成事务集 D，此时可得到如下命题。

命题 1 给定阈值 τ_{AR} 和 τ_{MS}，提取紧耦合群组等价于从事务集中根据阈值 $\tau_c=\tau_{AR}$ 和 $\tau_s=\tau_{MS}$ 挖掘余弦模式。

证明 给定一个用户集合 U，如果 q 为 U 中所有用户共同评论的某一商品，即 $q\in P_1\cap P_2\cap\cdots\cap P_{|U|}$，则可认为 U 包含在商品 q 对应的事务 $t_q\in D$ 中，因此，可得到：

$$|P_1\cap P_2\cap\cdots\cap P_{|U|}|=|\{t_q|U\subseteq t_q,1\leqslant q\leqslant m\}|=SC(U)$$

其中，$SC(U)$ 为 U 在事务集 D 中的支持度计数。则可得到：

$$MS(U)=\frac{|P_1\cap P_2\cap\cdots\cap P_{|U|}|}{m}=Supp(U) \tag{6}$$

进一步将共同评论率 R_l 的定义代入式(2)，易得：

$$\begin{aligned}AR(U)&=\sqrt[|U|]{\left|\bigcap_{l=1}^{|U|}P_l\right|^{|U|}/\prod_{u_l\in U}P_l}=\\&|P_1\cap P_2\cap\cdots\cap P_{|U|}|/\sqrt[|U|]{\prod_{u_l\in U}P_l}=\\&SC(U)/\sqrt[|U|]{\prod_{u_l\in U}SC(\{u_l\})}=\\&Supp(U)/\sqrt[|U|]{\prod_{l=1}^{|U|}\prod_{u_l\in U}Supp(\{u_l\})}=\\&Cos(U)\end{aligned} \tag{7}$$

由式(6)、式(7)以及定义 2、定义 3 可得，提取紧耦合群组等价于挖掘余弦模式，证毕。

2.3 候选水军群组提取

在紧耦合群组建模为余弦模式后，可采用余弦模式挖掘算法提取紧耦合群组。为易于进行后续的分布式扩展，本文选用文献[13]提出的余弦模式挖

掘算法 CIP-Growth 进行紧耦合群组提取。该算法是经典频繁模式挖掘算法 FP-Growth 的扩展,其基本流程与 FP-Growth 类似,如下:

1)扫描事务集并计算每个项的支持度计数,对其降序排列生成头部表 *Head Table*。

2)再次扫描事务集构建 FP 树,将其作为基础数据结构。

3)以递归的方式生成条件 FP 树,从中搜索余弦模式。

基于 CIP-Growth 的群组提取算法伪代码如下,具体的设计思想与细节可参考文献[13]。

算法 1　基于 GIP-Growth 的紧耦合群组提取算法

输入　条件 FP 树 *Tree*,初始化由事务集 *D* 构建;*Tree* 的前缀模式 *S*,初始化为空;支持度阈值 τ_s;余弦相似度阈值 τ_c

输出　余弦模式集合 *G*,即紧耦合群组集合,初始化为空

```
1. CIP-Growth(Tree,S,G, τs, τc)
2. FOR Tree 的头部表中的项 ik ;//按降序遍历
3. 生成候选余弦模式 S′ = S∪{ik};
4. IF Cos(S′)≥τc THEN
5. G = G∪S′;
6. 按 τs创建 S′的条件 FP 树 TreeS′;
7. IF TreeS′≠∅ THEN
8. CIP-Growth(Tree,S,G, τs, τc);
9. 返回 G.//G 中包含所有的余弦模式(紧耦合群组)
```

鉴于余弦模式与紧耦合群组间的等价关系,挖掘出事务集中的所有余弦模式相当于提取出所有的紧耦合群组。

3　候选群组提取算法的分布式实现

真实的电子商务平台中包含大量的商品与用户,由此建立的事务集规模巨大。为应对这一挑战,本文将对基于余弦模式挖掘的提取算法进行分布式扩展,以提升其性能。在类 FP-Growth 算法中,进行分布式设计的核心是利用组投影技术[14],通过将头部表划分成若干组,根据每组包含的项集,在事务数据集上进行投影,从而将事务集分割为互不相交的数据子集,然后对每个数据子集分别构建 FP 树并进行模式挖掘,最终汇总得到挖掘结果。本文利用组投影技术对事务集进行分割,然后基于内存计算框架 Spark(http://spark.apache.org/)设计分布式候选群组提取算法。

3.1　组投影技术

组投影技术是文献[14]为设计并行 FP-Growth 算法而提出的一种事务集转换切割方法,其形式化定义为:

定义 4(组投影)　假设按降序排列的头部表 *FList* 中的各项自上而下被分为 *K* 个不相交的子集,即 $FList = \beta_1 \cup \beta_2 \cup \cdots \cup \beta_K$,其中,$\beta_k = \{i_1^k, i_2^k, \cdots, i_r^k\}$ 由 *FList* 中 *r* 个连续项组成,则事务集 *D* 在第 *k* 个组 β_k 上的组投影数据集为 $D_k = \{T_p \cap (\bigcup_{j=1}^{k} \beta_j) \mid T_p \cap \beta_k \neq \varnothing, T_p \in D\}$。

为便于理解组投影的含义,以表 1 所示事务集为例,其按各项支持度降序排列的头部表为{*E*(7),*D*(5),*C*(5),*B*(4),*A*(3)}(括号中的数据代表相应项的支持度计数)。假设 *FList* 被分割为 3 个组{*D*,*E*}、{*B*,*C*}、{*A*},则在这种分割下通过组投影技术生成的数据集如表 2 所示。以组{*B*,*C*}投影得到的数据为例,当删除 *FList* 中位于项 *B* 之后的 *A* 后,提取所有包含 *B* 或 *C* 的事务,得到表 2 中组{*B*,*C*}所对应的包含 7 个事务的投影数据集,其中,括号中的数字表示对应事务在数据集中的数量。

表 1　事务集示例

事务 ID	项集	事务 ID	项集
1	*B*,*C*,*D*,*E*	5	*C*
2	*A*,*B*,*D*,*E*	6	*B*,*C*,*D*,*E*
3	*B*,*E*	7	*A*,*C*,*D*,*E*
4	*C*,*D*,*E*	8	*A*,*E*

表 2　组投影示例

组	投影数据集
{*D*,*E*}	{*D*,*E*}(5),{*E*}(2)
{*B*,*C*}	{*B*,*C*,*D*,*E*}(2),{*B*,*D*,*E*},{*C*,*D*,*E*}(2),{*B*,*E*},{*C*}
{*A*}	{*A*,*B*,*D*,*E*},{*A*,*C*,*D*,*E*},{*A*,*E*}

3.2　基于 Spark 框架的分布式候选水军群组提取

在 Spark 框架下,本文设计的基于余弦模式的分布式水军群组提取算法可封装为 2 个 Map-Reduce 过程,如下:

1)横向切割事务集 *D* 并加载到 Spark 的弹性分布式数据集(Resilient Distributed Datasets,RDD)中,触发第 1 次 Map-Reduce 过程。在 Map 阶段计算每个 RDD 上各项的支持度计数,然后以项为键、对应的支持度计数为值,在 Reduce 阶段汇总计算项在整个事务集中的支持度计数,并排序生成 *FList*,这一过程类似于经典的 Word Count 算法,在此不再赘述。

2)将 *FList* 分割为 *K* 个子集,广播到各个 RDD 上,触发第 2 次 Map-Reduce 过程。在 Map 阶段根据 *FList* 的分割对 RDD 中的事务进行组投影,在 Reduce 阶段,以组为键,汇总投影结果生成 *K* 个新的 RDD,每个 RDD 中保存对应组的投影数据。最后,在新产生的 RDD 中执行第 2.3 节的紧耦合群组提取算法,汇总得到候选群组。

基于 Spark 框架的分布式群组提取算法结构如图 2 所示。

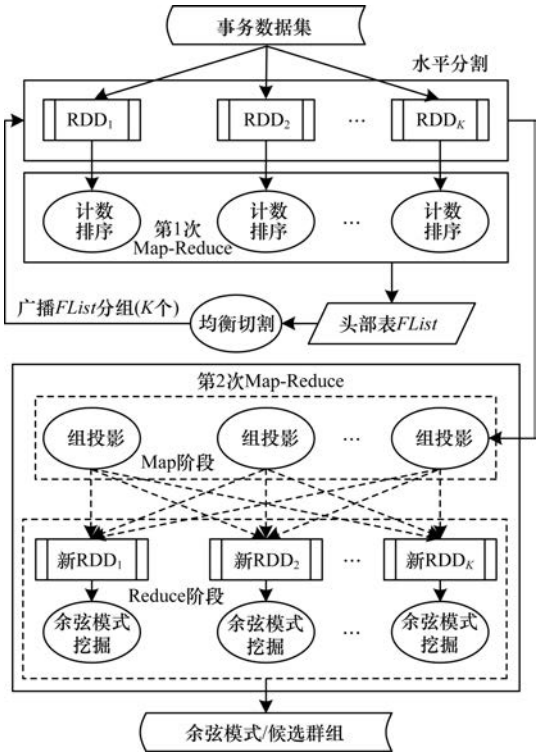

图 2 基于 Spark 框架的分布式群组提取算法结构

在得到候选群组后,可选用已有的水军群组识别方法从中识别水军群组。这些方法大多基于分类、排序等机器学习算法,如 SVM、PageRank 等,其分布式版本已在 Spark 的机器学习库 MLlib 中实现,可与本文分布式水军群组提取算法进行有效结合。

4 实验结果与分析

本节利用真实数据集从性能和准确性 2 个方面对分布式水军群组检测算法进行验证和分析。

4.1 实验设置

实验具体设置如下:

1)数据集:实验数据爬取自亚马逊中国网站(https://www.amazon.cn/),共收集了自 2009 年 9 月—2017 年 12 月间 834 547 个用户针对 4 837 386 个商品发布的 190 435 346 条评论数据。从中移除冷门商品和用户,仅保留发布 3 次以上评论的用户和被评论 10 次以上的商品以及对应的评论数据,建立如表 3 所示的实验数据集。

表 3 实验数据集

名称	用户数	商品数	评论数
amazon.cn	488 631	358 764	85 464 145

2)实验环境:本文使用 Spark 集群运行分布式水军群组检测算法,集群共包含 8 台服务器,使用 InfiniBand 万兆网络互联,每台服务器配置为英特尔 E5-2650v2 CPU(4 核,主频 2.6 GHz),128 GB 内存,240 GB SSD 硬盘以及 600 GB SAS 硬盘,操作系统为 64 位 RedHat Linux 7.0,Spark 版本为 1.6.2,实验代码采用 Java 语言编写。

4.2 性能验证

由于水军群组识别阶段的算法未纳入本文研究范围,因此本次实验主要关注群组提取阶段的性能。本文基于余弦模式挖掘的提取算法能够提取出紧耦合群组,较频繁模式挖掘方法能够大幅降低候选群组的数量,有利于提升后续识别的效率。设置余弦阈值 $\tau_c=0.2$,支持度 τ_s 在 0.01% ~0.05% 间变化,图 3 所示为余弦模式挖掘和频繁模式挖掘所提取出的群组数量对比。从图 3 可以看出,余弦模式挖掘提取出的候选群组数量远少于频繁模式挖掘,且支持度越小,两者的差异越大。由于支持度越小,越不容易遗漏群组,因此较小支持度下两者的显著性差异更能凸显余弦模式挖掘对于提升检测效率的优势。

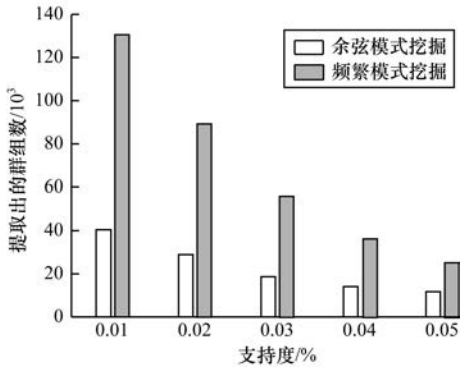

图 3 2 种模式提取群组的数量对比

从运行时间方面比较分布式版本和单机版本的余弦模式挖掘算法的性能差异。设置余弦阈值 $\tau_c=0.2$,支持度 $\tau_s=0.01\%$,*FList* 的分割数 K 从 4 变化到 20,经测算单机版算法的运行时间为 1 430.27 s,分布式版本算法的运行时间如图 4 所示。从图 4 可以看出,随着并行度(即 K 的数值)的增大,提取群组所消耗的时间持续减少,且分布式版本的运行时间远小于单机版本,当 $K=20$ 时,分布式版本算法比单机版本算法所消耗的时间减少 90% 以上。该实验结果表明,分布式余弦模式挖掘算法能够有效提升群组提取的效率,最终提升水军群组检测的整体性能。

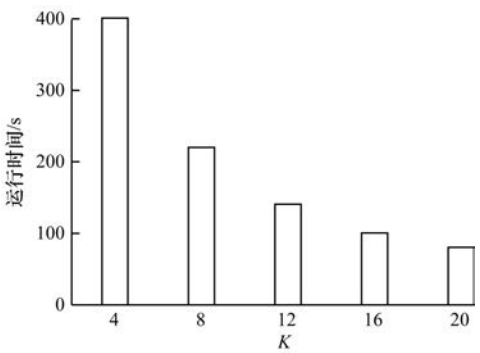

图 4 分布式余弦模式挖掘算法运行时间情况

4.3　准确性验证

为验证本文群组提取算法的检测准确性,选用文献[7]提出的 GSRank 算法从候选群组中识别水军群组。GSRank 算法利用群组、用户及商品特征建立群组-用户、群组-商品以及用户-商品 3 种关联关系并融合成关系网络,然后利用类似 PageRank 的算法计算群组恶意度并排序,选择排序列表中恶意度位于前列的群组,将其判定为水军群组。

由于实验所使用的数据集中并未给出用户或群组是否为水军的标记,即缺乏 GroundTruth。因此,本文采用文献[15]提出的间接验证思路对水军群组识别的准确性进行验证。由于 GSRank 算法会给出群组的恶意度,在按恶意度排序生成的列表中(假设按降序排列),如果检测算法是准确的,那么排在列表前列的群组和排在末尾的群组间必然具有较高的区分度,即前列群组的恶意度很高,末尾群组的恶意度很低,截取首尾部分群组组成新的测试集,使用分类器对测试集进行分类也将取得较好的分类效果。反之,如果检测算法准确性较低,则列表首尾部分群组的区分度较低,分类器将难以取得好的分类效果。在进行间接验证时,本文设定余弦阈值 $\tau_c = 0.2$,支持度 $\tau_s = 0.01\%$,提取候选群组并采用 GSRank 算法进行排序,然后分别截取列表首尾 5% 、10% 、15% 的群组建立测试集,再使用 SVM 和 C4.5 决策树对测试集进行分类,分类效果采用准确率(P)、召回率(R)和综合评价指标(F)进行描述,实验结果如表 4 所示。由表 4 可以看出,本文余弦模式提取算法与频繁模式提取算法在水军检测中所取得的准确率相差无几,说明本文算法在提升检测效率的同时,也能保证准确率。由表 4 还可以看出,在排序列表首尾截取的群组越少,群组间的区分度越高,分类器的分类效果越好。

表 4　2 种模式检测性能对比

截取首尾比例/%	提取算法	SVM			C4.5		
		P	R	F	P	R	F
5	余弦模式	0.95	0.93	0.94	0.91	0.90	0.90
	频繁模式	0.95	0.94	0.94	0.90	0.91	0.90
10	余弦模式	0.92	0.90	0.91	0.88	0.86	0.87
	频繁模式	0.93	0.91	0.92	0.87	0.87	0.87
15	余弦模式	0.87	0.85	0.86	0.85	0.84	0.84
	频繁模式	0.87	0.86	0.86	0.84	0.84	0.84

使用余弦模式挖掘算法提取出的群组数量远少于频繁模式挖掘,因此,余弦模式是否会造成水军群组遗漏值得关注。本文分别使用余弦模式和频繁模式提取群组,然后用 GSRank 排序得到 2 个列表并按恶意度降序排列。由于水军群组排在列表的前列,本文分别截取 2 个列表的前 Y 个群组,考查两者间的重合率,实验结果如图 5 所示。由图 5 可以看出,2 个列表中排序靠前的群组间重合度很高,当余弦阈值 $\tau_c = 0.2$、截取的群组位于列表前 2 000 时,2 个列表中超过 90% 的群组是相同的。因此,相对频繁模式挖掘,余弦模式挖掘所排除的群组主要是非水军群组,几乎没有造成水军群组遗漏。

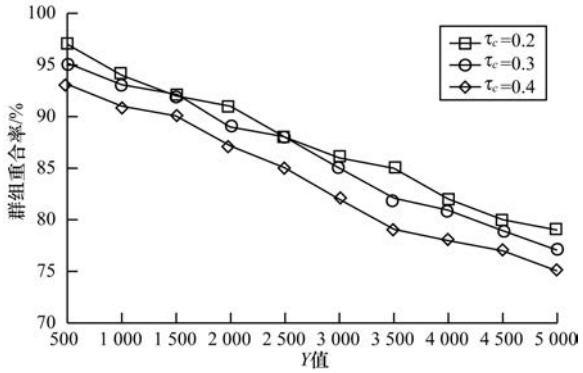

图 5　2 个水军群组列表中前 Y 个群组的重合率情况

5　案例研究

为进一步阐述本文水军群组检测算法的有效性,本节展现 2 个由该算法从实验数据集中检测出的水军群组,并分析其异常行为。本文实验数据取自 2009 年 9 月—2013 年 12 月,由于亚马逊网站本身也会开展水军检测工作,案例中展现的水军 ID 可能已被网站删除,但这也从侧面说明了本文算法的有效性。2 个水军群组信息如下:

1)水军群组 1 共包含 8 名成员,在 2010 年 5 月 21 日—25 日的 5 天里,这些用户一共评论了15 个商品,所有商品均给予 5 星评价。图 6 所示为用户评论商品的时间分布情况,可以看出,这些用户多次在同一天共同评论同一商品,这种行为有异于偶然形成的普通群组。普通群组的成员虽然有共同评论的商品,但评论时间通常是不同的。此外,通过在亚马逊中国网站上搜索群组成员的 ID,发现很多 ID 或相关评论已被删除,这说明亚马逊的检测算法与本文得出了一致的结论,上述用户组成的群组为水军群组。

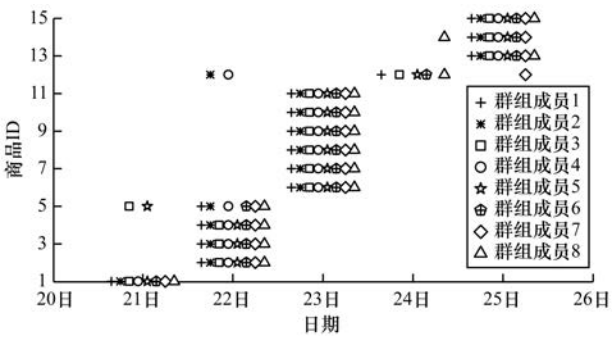

图 6　水军群组 1 的评论时间分布情况

2)水军群组 2 包含 5 名成员,并于 2010 年 9 月 30 日共同评论了 3 件商品,其中,3 名成员还分别额

外评论了2件商品。图 7 所示为该群组的目标商品及评论行为。首先,该群组所评论的商品集中属于“书中缘”品牌的图书,但图书类别各不相同,因此,其并非由兴趣形成的群组;其次,3 名成员发布了重复评论,这是水军发表评论的典型特征;最后,所有成员均给出了 4 星或 5 星的高评分,其目的在于提升“书中缘”品牌图书的排名,即该群组属于品牌营销水军。

图 7 水军群组 2 的目标商品及评论行为

6 结束语

本文提出紧耦合群组的概念并用余弦模式进行建模,基于余弦模式挖掘精准提取候选群组,在此基础上,运用组投影和 Spark 框架设计一种分布式群组提取算法,以提升算法的运行效率与扩展性。下一步将通过合理划分头部表,使组投影技术得到更加均衡的数据切割,以提高分布式算法的运行效率。

参考文献

[1] CHEVALIER J A, MAYZLIN D. The effect of word of mouth on sales: online book reviews[J]. Journal of Marketing Research, 2006, 43(3): 345-354.

[2] 李璐旸, 秦兵, 刘挺. 虚假评论检测研究综述[J]. 计算机学报, 2018, 41(4): 946-968.

[3] JINDAL N, LIU Bing. Opinion spam and analysis[C]// Proceedings of 2008 International Conference on Web Search and Data Mining. New York, USA: ACM Press, 2008: 219-230.

[4] 莫倩, 杨珂. 网络水军识别研究[J]. 软件学报, 2014, 25(7): 1505-1526.

[5] 金礼仁. 基于结构与内容的社交网络水军团体识别[D]. 南京: 南京邮电大学, 2016.

[6] YE Junting, AKOGLU L. Discovering opinion spammer groups by network footprints[C]//Proceedings of 2015 Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin, Germany: Springer, 2015: 267-282.

[7] MUKHERJEE A, LIU Bing, GLANCE N. Spotting fake reviewer groups in consumer reviews[C]//Proceedings of International Conference on World Wide Web. New York, USA: ACM Press, 2012: 191-200.

[8] XU Chang, ZHANG Jie, CHANG Kuiyu, et al. Uncovering collusive spammers in Chinese review websites[C]// Proceedings of ACM International Conference on Information and Knowledge Management. New York, USA: ACM Press, 2013: 979-988.

[9] WANG Youquan, WU Zhang, BU Zhan, et al. Discovering shilling groups in a real e-commerce platform[J]. Online Infor-mation Review, 2016, 40(1): 62-78.

[10] 常家伟, 戴牡红. 基于 PageRank 和谱方法的个性化推荐算法[J]. 计算机科学, 2018, 45(增刊): 398-401.

[11] LUO Dijun, DING C, HUANG Heng. Parallelization with multiplicative algorithms for big data mining[C]// Proceedings of IEEE International Conference on Data Mining. Washington D. C., USA: IEEE Computer Society, 2012: 489-498.

[12] WANG Zhuo, HOU Tingting, SONG D, et al. Detecting review spammer groups via bipartite graph projection[J]. Computer Journal, 2016, 59(6): 861-874.

[13] CAO Jie, WU Zhang, WU Junjie. Scaling up cosine interesting pattern discovery: a depth-first method[J]. Information Sciences, 2014, 266(5): 31-46.

[14] GRAHNE G, ZHU Jianfeng. Mining frequent itemsets from secondary memory[C]//Proceedings of IEEE International Conference on Data Mining. Washington D. C., USA: IEEE Computer Society, 2004: 91-98.

[15] MUKHERJEE A, KUMAR A, LIU Bing, et al. Spotting opinion spammers using behavioral footprints[C]// Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2013: 632-640.

编辑 吴云芳