



(12)发明专利

(10)授权公告号 CN 109753797 B

(45)授权公告日 2020.11.03

(21)申请号 201811503421.0

(22)申请日 2018.12.10

(65)同一申请的已公布的文献号  
申请公布号 CN 109753797 A

(43)申请公布日 2019.05.14

(73)专利权人 中国科学院计算技术研究所  
地址 100080 北京市海淀区中关村科学院  
南路6号

(72)发明人 程学旗 刘盛华 喻文健 张嘉宝  
冯文杰 沈华伟

(74)专利代理机构 北京律诚同业知识产权代理  
有限公司 11006  
代理人 祁建国 梁挥

(51)Int.Cl.  
G06F 21/56(2013.01)

(56)对比文件

- CN 104954477 A, 2015.09.30
- CN 103268481 A, 2013.08.28
- CN 103338379 A, 2013.10.02
- CN 106100921 A, 2016.11.09
- CN 104820705 A, 2015.08.05
- CN 103400152 A, 2013.11.20
- CN 104598629 A, 2015.05.06
- CN 107928631 A, 2018.04.20
- US 2007055646 A1, 2007.03.08
- CN 104303153 A, 2015.01.21

审查员 李婧雯

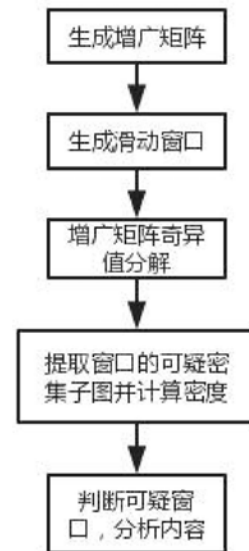
权利要求书3页 说明书7页 附图2页

(54)发明名称

针对流式图的密集子图检测方法及其系统

(57)摘要

本发明涉及一种针对流式图的密集子图检测方法和系统,包括:持续从社交网络获取三元组,该三元组由用户、对象和时间戳组成,以该三元组作为流式图建模为行增广矩阵;用滑动窗口访问行增广矩阵,并对每个窗口内的行增广矩阵进行奇异值分解,得到奇异矩阵,获取奇异矩阵的奇异向量对,根据向量阈值对该奇异向量对进行筛选,得到候选密集块及其密度;通过对候选密集块利用已有方法进一步进行密集子块筛选;最终密集块的用户为检测的异常用户、其中的目标物为检测的异常目标。本发明根据增广矩阵和滑动窗口对流式图建模,每次只存储一个步长的数据,每次检测一个窗口的数据,性能优于每插入一条新数据都要更新密集块的流式算法。



1. 一种针对流式图的密集子图检测方法,其特征在于,包括:

步骤1、持续从社交网络获取以流式图表示的三元组,该三元组由用户、对象和时间戳组成,通过将该时间戳拼接对象上作为行,用户作为列,将该流式图建模为行增广矩阵;

步骤2、用滑动窗口访问行增广矩阵,并对每个窗口内的行增广矩阵进行奇异值分解,得到奇异矩阵U、S和V,获取U、V奇异矩阵的奇异向量对 $(u, v)$ ,设置向量阈值筛选u和v向量上的值,得到候选密集块及该候选密集块的密度;

步骤3、集合每个窗口输出的密度,得到密度历史集合,设置密度阈值为 $\mu+3\sigma$ ,其中 $\mu$ 是该密度历史集合的平均值, $\sigma$ 是该密度历史集合的标准差,若t时刻窗口的密度 $D_t$ 大于该密度阈值,则判定t时刻窗口的候选密集块中的用户和对象分别为异常用户和异常目标,将该异常用户和该异常目标作为检测结果输出。

2. 如权利要求1所述的针对流式图的密集子图检测方法,其特征在于,该步骤1包括:

持续从社交网络获取三元组,并将所有的用户记作集合 $B_1$ ,所有的对象记作集合 $B_2$ ,构造二部图 $\mathcal{G} = (V, E)$ ,其中E是图的边集合,V是图的节点集合, $V = B_1 \cup B_2$ ,将 $\mathcal{G}$ 作为该流式图,建模为该行增广矩阵。

3. 如权利要求1所述的针对流式图的密集子图检测方法,其特征在于,该奇异值分解包括:

步骤21、定义随机矩阵 $\Omega$ 大小为 $n \times l$ ;窗口内的行增广矩阵的大小为 $m \times n$ ;其中 $l < \min(m, n)$ ;

步骤22、定义两个列表glist, hlist,结合滑动窗口,通过下式每次计算一个步长s的窗口内的行增广矩阵a,生成对应的矩阵g, h,分别存入glist, hlist;

$$g = a \Omega; h = a^T g$$

遍历完该行增广矩阵后,将glist中所有的矩阵g按行拼接生成矩阵G,将hlist中所有的矩阵h相加生成矩阵H;

步骤23、利用已有的Single-pass PCA算法,根据矩阵G和H得到矩阵Q和B,再根据矩阵Q和B生成奇异矩阵U、S和V。

4. 如权利要求1或2或3所述的针对流式图的密集子图检测方法,其特征在于,该步骤2中筛选过程包括:

步骤24、设置u向量的向量阈值为 $\tau_u = \frac{1}{\sqrt{m_t}}$ ,  $m_t$ 代表t时刻窗口对应的行增广矩阵的行数,v向量的向量阈值为 $\tau_v = \frac{1}{\sqrt{n_t}}$ ,  $n_t$ 代表t时刻窗口对应的行增广矩阵的列数;

步骤25、获取第t时刻窗口的奇异向量对 $(u_t, v_t)$ ,  $u_t \subseteq U_t$ ,  $v_t \subseteq V_t$ ,遍历 $u_t$ 向量的值,提取出所有不小于阈值 $\tau_u$ 的行,构成集合rowset;遍历 $v_t$ 向量的值,提取出所有不小于阈值 $\tau_v$ 的列,构成集合colset, rowset中的行和colset中的列构成了窗口 $[t, t+w]$ 的候选密集块 $B_t$ , w为窗口的大小,  $U_t$ 为第t时刻窗口内的奇异矩阵U,  $V_t$ 为第t时刻窗口内的奇异矩阵V;

步骤26、通过 $B_t$ 的边数和除以 $B_t$ 的行列数之和,得到候选密集块的密度。

5. 如权利要求4所述的针对流式图的密集子图检测方法,其特征在于,该步骤25包括:根据已有的密集块检测算法HoloScope或Fraudar,对候选密集块 $B_t$ 进一步进行密集子块筛

选,形成最终的候选密集块。

6. 一种针对流式图的密集子图检测系统,其特征在于,包括:

增广矩阵生成模块,用于持续从社交网络获取以流式图表示的三元组,该三元组由用户、对象和时间戳组成,通过将时间戳拼接到对象上作为行,用户作为列,将该流式图建模为行增广矩阵;

奇异值分解模块,用滑动窗口访问行增广矩阵,并对每个窗口内的行增广矩阵进行奇异值分解,得到奇异矩阵U、S和V,获取U、V奇异矩阵的奇异向量对(u,v),设置向量阈值筛选u和v向量上的值,得到候选密集块及该候选密集块的密度;

检测模块,集合每个窗口输出的密度,得到密度历史集合,设置密度阈值为 $\mu+3\sigma$ ,其中 $\mu$ 是该密度历史集合的平均值, $\sigma$ 是该密度历史集合的标准差,若t时刻窗口的密度 $D_t$ 大于该密度阈值,则判定t时刻窗口的候选密集块中的用户和对象分别为异常用户和异常目标,将该异常用户和该异常目标作为检测结果输出。

7. 如权利要求6所述的针对流式图的密集子图检测系统,其特征在于,该增广矩阵生成模块,包括:

持续从社交网络获取三元组,并将所有的用户记作集合 $B_1$ ,所有的对象记作集合 $B_2$ ,构造二部图 $\mathcal{G} = (V,E)$ ,其中E是图的边集合,V是图的节点集合, $V = B_1 \cup B_2$ ,将 $\mathcal{G}$ 作为该流式图,建模为该行增广矩阵。

8. 如权利要求6所述的针对流式图的密集子图检测系统,其特征在于,该奇异值分解包括:

定义随机矩阵 $\Omega$ 大小为 $n \times 1$ ;窗口内的行增广矩阵的大小为 $m \times n$ ;其中 $1 < \min(m,n)$ ;

定义两个列表glist,hlist,结合滑动窗口,通过下式每次计算一个步长s的窗口内的行增广矩阵a,生成对应的矩阵g,h,分别存入glist,hlist;

$$g = a \Omega; h = a^T g$$

遍历完该行增广矩阵后,将glist中所有的矩阵g按行拼接生成矩阵G,将hlist中所有的矩阵h相加生成矩阵H;

利用已有的Single-pass PCA算法,根据矩阵G和H得到矩阵Q和B,再根据矩阵Q和B生成奇异矩阵U、S和V。

9. 如权利要求6或7或8所述的针对流式图的密集子图检测系统,其特征在于,该奇异值分解模块中筛选过程包括:

设置u向量的向量阈值为 $\tau_u = \frac{1}{\sqrt{m_t}}$ , $m_t$ 代表t时刻窗口对应的行增广矩阵的行数,v向量的向量阈值为 $\tau_v = \frac{1}{\sqrt{n_t}}$ , $n_t$ 代表t时刻窗口对应的行增广矩阵的列数;

获取第t时刻窗口的奇异向量对 $(u_t, v_t)$ , $u_t \subseteq U_t$ , $v_t \subseteq V_t$ ,遍历 $u_t$ 向量的值,提取出所有不小于阈值 $\tau_u$ 的行,构成集合rowset;遍历 $v_t$ 向量的值,提取出所有不小于阈值 $\tau_v$ 的列,构成集合colset,rowset中的行和colset中的列构成了窗口 $[t, t+w]$ 的候选密集块 $B_t$ ,w为窗口的大小, $U_t$ 为第t时刻窗口内的奇异矩阵U, $V_t$ 为第t时刻窗口内的奇异矩阵V;

通过 $B_t$ 的边数和除以 $B_t$ 的行列数之和,得到候选密集块的密度。

10. 如权利要求9所述的针对流式图的密集子图检测系统,其特征在于,该筛选过程还

包括:根据密集块检测算法HoloScope或Fraudar,对候选密集块 $B_t$ 进一步进行密集子块筛选,形成最终的候选密集块。

## 针对流式图的密集子图检测方法及系统

### 技术领域

[0001] 本发明属于计算机技术领域,特别涉及一种针对流式图数据的密集子图检测方法和系统。

### 背景技术

[0002] 随着社交网络的兴起,网络诈骗已经成为一个越来越严重的问题,大量社交平台都存在各种各样的欺诈行为,如微博水军、淘宝刷单等,以及网络攻击行为,如DDoS攻击。如何检测这类异常行为已经越来越引起人们的重视,用图表示社交网络的数据,问题可以转换为基于大规模流式图挖掘检测异常行为。

[0003] 传统的异常检测算法都是检测静态数据,比如说,基于谱分解的EigenSpokes,还有很多算法基于图密度,比如说Fraudar,甚至有算法还考虑了攻击的爆发增长和回落,例如HoloScope。这些算法能精准的检测到异常行为,但是它们是基于静态图的,这会造成两个问题:第一,不能实时得到异常检测的反馈结果;第二,每次要计算所有的数据,计算量太大,耗时长。因此,人们偏向于检测流式图数据,以便能够及时得到反馈。

[0004] 传统的基于流式图的欺诈检测算法,只是通过相似函数来比较相邻图的变化,而不考虑整体的趋势,导致结果不准确。现有的很多流式算法都是检测密集子图,将动态图建模为流式张量,目的是近似地识别出topK个最密集的子块。然而,这些算法需要维持密集块,每读入一条新数据,都要更新密集块,性能不高,特别是当密集块很大的情况下,更新速度很慢。Spotlight基于随机草图映射的方法,能够实时检测出密集块的突然出现或消失,但是它只能检测出大的密集块,因为草图只包含原流式图的主要特征。还有一些方法是基于图分解和划分的,比如存储了基于张量分解的图结构摘要,并将变化点识别为异常。另外,随机算法定义了一个健壮的随机切割数据结构,可以用作输入流的草图或概要。但是这些方法都只能识别出大的密集块。

[0005] 通过分析,检测流式数据更符合实际应用情况,如何高效准确地识别出流式数据中的欺诈密集块是一个有待解决的问题。

### 发明内容

[0006] 本发明的目的是解决现有基于流式数据异常检测技术的缺陷,提出了一种基于流式图奇异值分解的密集子图检测方法。

[0007] 具体来说,本发明涉及一种针对流式图的密集子图检测方法,其中包括:

[0008] 步骤1、持续从社交网络获取以流式图表示的三元组,该三元组由用户、对象和时间戳组成,通过将时间戳拼接到对象上作为行,用户作为列,将该流式图建模为行增广矩阵;

[0009] 步骤2、用滑动窗口访问行增广矩阵,并对每个窗口内的行增广矩阵进行奇异值分解,得到奇异矩阵U、S和V,获取U、V奇异矩阵的奇异向量对 $(u, v)$ ,设置向量阈值筛选 $u, v$ 向量上的值,得到候选密集块及该候选密集块的密度;

[0010] 步骤3、集合每个窗口输出的该密度,得到密度历史集合,设置密度阈值为 $\mu+3\sigma$ ,其中 $\mu$ 是该密度历史集合的平均值, $\sigma$ 是该密度历史集合的标准差,若 $t$ 时刻窗口的密度 $D_t$ 大于该密度阈值,则判定 $t$ 时刻窗口的候选密集块中用户为异常用户,否则为正常用户,将该异常用户的对象作为异常目标,将异常用户和异常目标作为检测结果输出。

[0011] 该针对流式图的密集子图检测方法,其中该步骤1包括:

[0012] 持续从社交网络获取三元组,并将所有的用户记作集合 $B_1$ ,所有的对象记作集合 $B_2$ ,构造二部图 $\mathcal{G} = (V,E)$ ,其中 $E$ 是图的边集合, $V$ 是图的节点集合, $V = B_1 \cup B_2$ ,将 $\mathcal{G}$ 作为该流式图,建模为该行增广矩阵。

[0013] 该针对流式图的密集子图检测方法,其中该奇异值分解包括:

[0014] 步骤21、定义随机矩阵 $\Omega$ 大小为 $n \times 1$ ,其中 $1 < \min(m, n)$ ;窗口内的行增广矩阵的大小为 $m \times n$ ;

[0015] 步骤22、定义两个列表 $glist, hlist$ ,结合滑动窗口,通过下式每次计算一个步长 $s$ 的窗口内的行增广矩阵 $a$ ,生成对应的矩阵 $g, h$ ,分别存入 $glist, hlist$ ;

[0016]  $g = a \Omega ; h = a^T g$

[0017] 遍历完该行增广矩阵后,将 $glist$ 中所有的矩阵 $g$ 按行拼接生成矩阵 $G$ ,将 $hlist$ 中所有的矩阵 $h$ 相加生成矩阵 $H$ ;

[0018] 步骤23、利用已有的Single-pass PCA算法,根据矩阵 $G, H$ 得到矩阵 $Q, B$ ,再根据矩阵 $Q, B$ 生成奇异矩阵 $U, S$ 和 $V$ 。

[0019] 该针对流式图的密集子图检测方法,其中该步骤2中筛选过程包括:

[0020] 步骤24、设置 $u$ 向量的向量阈值为 $\tau_u = \frac{1}{\sqrt{m_t}}$ , $m_t$ 代表 $t$ 时刻窗口对应的行增广矩阵

的行数, $v$ 向量的向量阈值为 $\tau_v = \frac{1}{\sqrt{n_t}}$ , $n_t$ 代表 $t$ 时刻窗口对应的行增广矩阵的列数;

[0021] 步骤25、获取第 $t$ 时刻窗口的奇异向量对 $(u_t, v_t)$ , $u_t \subseteq U_t, v_t \subseteq V_t$ ,遍历 $u_t$ 向量的值,提取出所有不小于阈值 $\tau_u$ 的行,构成集合 $rowset$ ;遍历 $v_t$ 向量的值,提取出所有不小于阈值 $\tau_v$ 的列,构成集合 $colset$ , $rowset$ 中的行和 $colset$ 中的列构成了窗口 $[t, t+w]$ 的候选密集块 $B_t$ , $w$ 为窗口的大小;

[0022] 步骤26、通过 $B_t$ 的边数和除以 $B_t$ 的行列数之和,得到候选密集块的密度。

[0023] 该针对流式图的密集子图检测方法,其中该步骤25包括:根据密集块检测算法HoloScope或Fraudar,对候选密集块 $B_t$ 进一步进行密集子块筛选,形成最终的候选密集块。

[0024] 本发明还提供了一种针对流式图的密集子图检测系统,其中包括:

[0025] 增广矩阵生成模块,用于持续从社交网络获取以流式图表示的三元组,该三元组由用户、对象和时间戳组成,通过将时间戳拼接到对象上作为行,用户作为列,将该流式图建模为行增广矩阵;

[0026] 奇异值分解模块,用滑动窗口访问行增广矩阵,并对每个窗口内的行增广矩阵进行奇异值分解,得到奇异矩阵 $U, S$ 和 $V$ ,获取 $U, V$ 奇异矩阵的奇异向量对 $(u, v)$ ,设置向量阈值筛选 $u, v$ 向量上的值,得到候选密集块及该候选密集块的密度;

[0027] 检测模块,集合每个窗口输出的该密度,得到密度历史集合,设置密度阈值为 $\mu+3$

$\sigma$ , 其中 $\mu$ 是该密度历史集合的平均值, $\sigma$ 是该密度历史集合的标准差,若 $t$ 时刻窗口的密度大于该密度阈值,则判定 $t$ 时刻窗口的候选密集块中的用户和对象分别为异常用户和异常目标,将该异常用户和该异常目标作为检测结果输出。

[0028] 该针对流式图的密集子图检测系统,其中该增广矩阵生成模块,包括:

[0029] 持续从社交网络获取三元组,并将所有的用户记作集合 $B_1$ ,所有的对象记作集合 $B_2$ ,构造二部图 $\mathcal{G} = (V, E)$ ,其中 $E$ 是图的边集合, $V$ 是图的节点集合, $V = B_1 \cup B_2$ ,将 $\mathcal{G}$ 作为该流式图,建模为该增广矩阵。

[0030] 该针对流式图的密集子图检测系统,其中该奇异值分解包括:

[0031] 定义随机矩阵 $\Omega$ 大小为 $n \times l$ ,其中 $l < \min(m, n)$ ;窗口内的行增广矩阵的大小为 $m \times n$ ;

[0032] 定义两个列表 $glist, hlist$ ,结合滑动窗口,通过下式每次计算一个步长 $s$ 的窗口内的行增广矩阵 $a$ ,生成对应的矩阵 $g, h$ ,分别存入 $glist, hlist$ ;

[0033]  $g = a \Omega; h = a^T g$

[0034] 遍历完该行增广矩阵后,将 $glist$ 中所有的矩阵 $g$ 按行拼接生成矩阵 $G$ ,将 $hlist$ 中所有的矩阵 $h$ 相加生成矩阵 $H$ ;

[0035] 利用已有的Single-pass PCA算法,根据矩阵 $G, H$ 得到矩阵 $Q, B$ ,再根据矩阵 $Q, B$ 生成奇异矩阵 $U, S$ 和 $V$ 。

[0036] 该针对流式图的密集子图检测系统,其中该奇异值分解模块中筛选过程包括:

[0037] 设置 $u$ 向量的向量阈值为 $\tau_u = \frac{1}{\sqrt{m_t}}$ , $m_t$ 代表 $t$ 时刻窗口对应的行增广矩阵的行数, $v$ 向量的向量阈值为 $\tau_v = \frac{1}{\sqrt{n_t}}$ , $n_t$ 代表 $t$ 时刻窗口对应的行增广矩阵的列数;

[0038] 获取第 $t$ 时刻窗口的奇异向量对 $(u_t, v_t)$ ,  $u_t \subseteq U_t, v_t \subseteq V_t$ ,遍历 $u_t$ 向量的值,提取出所有不小于阈值 $\tau_u$ 的行,构成集合 $rowset$ ;遍历 $v_t$ 向量的值,提取出所有不小于阈值 $\tau_v$ 的列,构成集合 $colset$ , $rowset$ 中的行和 $colset$ 中的列构成了窗口 $[t, t+w]$ 的候选密集块 $B_t$ , $w$ 为窗口的大小;

[0039] 通过 $B_t$ 的边数和除以 $B_t$ 的行列数之和,得到候选密集块的密度。

[0040] 该针对流式图的密集子图检测系统,其中该筛选过程还包括:根据密集块检测算法HoloScope或Fraudar,对候选密集块 $B_t$ 进一步进行密集子块筛选,形成最终的候选密集块。

[0041] 本发明技术进步包括:

[0042] 用行增广矩阵和滑动窗口对流式图建模,提出了流式图的奇异分解算法AugSVD。每个窗口的行增广矩阵调用AugSVD算法,每次只需存储一个步长的数据在内存中,而传统的SVD分解需要存储一个窗口的数据,AugSVD节省了大量内存,扩展性良好。EigenPulse每次检测一个窗口的数据,相比起每插入一条新数据都要更新密集块的流式算法,性能大幅提升。DenseAlert是目前性能最好的流式图密集子图检测算法,比最快的批处理算法快了几百倍,如图2所示,EigenPulse在表1的前5个数据集上的运行速度比DenseAlert至少提高了2.53倍,在Amazon CellPhone数据集上甚至提高了12.2倍。

## 附图说明

[0043] 图1为滑动窗口示意图；

[0044] 图2为EigenPulse与DenseAlert运行时间对比图；

[0045] 图3为流式图异常检测模型的处理流程图；

[0046] 图4为EigenPulse在微博数据集上的密集块检测结果。

[0047] 具体实施细节

[0048] 本发明的发明步骤包括：

[0049] 1、社交网络不断生成形如三元组(用户,商品,时间戳)的数据,表示用户在该时间戳评价了商品。将所有的用户记作集合 $B_1$ ,所有的商品记作集合 $B_2$ ,构造二部图 $\mathcal{G} = (V,E)$ 表示数据,其中 $E$ 是图的边集合, $V$ 是图的节点集合, $V = B_1 \cup B_2$ ,边代表用户和商品之间的连接。用户是节点,构成了节点集合 $B_1$ ,商品是另一类的节点,构成节点集合 $B_2$ 。如果用户买了商品,就会在这个用户和这个商品之间形成一条边,这条边上记录了购买信息,比如说购买时间(即时间戳)。将流式图 $\mathcal{G}$ 建模为行增广矩阵 $A$ ,如果每条新数据对应的矩阵行号是递增的或者等于最后一行的行号,就是行增广矩阵。拼接商品和时间戳作为行,用户作为列,随着时间不断增长,矩阵行号一定是递增的。

[0050] 2、设计滑动窗口访问 $\mathcal{G}$ ,如图1所示。图1展示的是 $A^T$ ,行代表用户,列递增。定义时间单位的窗口大小为 $w$ ,时间单位的步幅大小为 $s$ ,每次窗口向前推进 $s$ 形成下一个窗口。假设窗口起始时间为 $t$ ,则结束时间为 $t+w$ ,对应的行增广矩阵为 $A_t$ ,下一个窗口的开始时间为 $t+s$ 。若 $w$ 无穷大,在每次步骤中考虑所有的历史数据;若 $w=s$ ,可以得到非重叠子图。

[0051] 3、结合滑动窗口和行增广矩阵,设计算法AugSVD做行增广矩阵的奇异值分解。AugSVD算法基于Single-pass PCA算法,改进了矩阵 $G,H$ 的生成过程,矩阵 $G,H$ 是用来生成矩阵 $Q,B$ 的中间矩阵。定义 $t$ 时刻的行增广矩阵 $A_t$ 的大小为 $m \times n$ ,算法输入 $A_t$ ,输出 $t$ 时刻窗口的奇异矩阵 $U_t, S_t, V_t$ 。

[0052] AugSVD算法步骤如下：

[0053] 1) 定义随机矩阵 $\Omega$ 大小为 $n \times 1$ ,其中 $1 < \min(m, n)$ 。

[0054] 2) 定义两个列表 $glist, hlist$ 。结合滑动窗口,每次计算一个步长 $s$ 的矩阵 $a$ ,生成对应的矩阵 $g, h$ ,分别存入 $glist, hlist$ 。

[0055]  $g = a \Omega ; h = a^T g$

[0056] 遍历完 $A_t$ 后,将 $glist$ 中所有的矩阵 $g$ 按行拼接生成矩阵 $G$ ,将 $hlist$ 中所有的矩阵 $h$ 相加生成 $H$ 。

[0057] 3) 和Single-pass PCA算法相同,根据矩阵 $G,H$ 生成矩阵 $Q,B$ ,效果类似于 $QB$ 分解。再根据 $Q,B$ 矩阵生成 $A_t$ 的奇异矩阵 $U_t, S_t, V_t$ 。

[0058] 4、设计EigenPulse算法提取窗口的可疑密集块(候选密集块)并计算可疑密集块的密度。在 $t$ 时刻,行增广矩阵 $A_t$ 对应的时间窗口为 $[t, t+w]$ ,行数为 $m_t$ ,列数为 $n$ ,输入奇异向量对 $(u_t, v_t)$ ,其中 $u_t \subseteq U_t, v_t \subseteq V_t$ ,输出可疑密集块的密度 $D_t$ 。

[0059] EigenPulse算法步骤如下：

[0060] 1) 设置 $u$ 向量的阈值为 $\tau_u = \frac{1}{\sqrt{m_t}}$ , $m_t$ 代表 $t$ 时刻窗口对应的行增广矩阵的行数, $v$



向量的阈值为 $\tau_v = \frac{1}{\sqrt{n_t}}$ ,  $n_t$ 代表t时刻窗口对应的行增广矩阵的列数。

[0061] 2) 遍历 $u_t$ 向量的值, 提取出所有不小于阈值 $\tau_u$ 的行, 构成集合rowset; 遍历 $v_t$ 向量的值, 提取出所有不小于阈值 $\tau_v$ 的列, 构成集合colset。rowset中的行和colset中的列构成了窗口 $[t, t+w]$ 的候选密集块 $B_t$ 。

[0062] 3) [可选步骤] 调用已有的密集块检测算法HoloScope或Fraudar在密集子块 $B_t$ 上进一步寻找更密集的子块, 构成可疑密集块 $B'_t$ 。

[0063] 计算 $B'_t$ 的密度 $D_t$ , 分子为 $B'_t$ 的边数和, 分母为 $B'_t$ 的行列数之和。

$$[0064] \quad D_t(\text{rowset}, \text{colset}) = \frac{\sum_{i \in \text{rowset}} \sum_{j \in \text{colset}} A_t(i, j)}{|\text{rowset}| + |\text{colset}|}$$

[0065] 5、将每个窗口输出的密度记作集合D, 根据正态分布性质, 设置密度阈值为 $\mu+3\sigma$ ,  $\mu$ 是D的历史平均值,  $\sigma$ 是D的历史标准差。若t时刻的密度 $D_t$ 大于阈值, 则 $B'_t$ 中的用户非常具有嫌疑。

[0066] 为了让本发明的上述特征和效果能阐述的更明确易懂, 下文特举实施例, 并配合说明书附图作详细说明如下。

[0067] 结合图3中的整个模型处理流程, 具体的实施步骤如下所示:

[0068] 步骤1、选取新浪微博的数据, 时间跨度1个月, 如表1所示。数据格式为(用户, 微博, 时间戳), 代表用户在该时间转发了这条微博。将时间戳拼接到微博上作为行, 用户作为列, 生成增广矩阵A, A的元素值为用户在该时间转发这条微博的次数。

[0069] 步骤2、设置滑动窗口参数,  $w=2h$ ,  $s=1h$ 。

[0070] 步骤3、拿第一个窗口举例, 行增广矩阵 $A_0$ 存储了初始两小时的数据, 调用AugSVD算法输出奇异矩阵U, S, V。

[0071] 步骤4、调用EigenPulse算法, 输入为U, V矩阵的第一个奇异向量对 $(u_0, v_0)$ 。首先提取出不小于阈值 $\tau_u$ 的行和不小于阈值 $\tau_v$ 的列, 构成密集块。再调用检测算法Fraudar检测密集块, 输出有异常嫌疑的行和列, 构成可疑密集块并计算可疑密集块的密度。

[0072] 步骤5、根据所有历史窗口的密度, 计算密度阈值 $\mu+3\sigma$ , 取出密度大于阈值的窗口, 这些窗口的可疑密集块非常具有嫌疑。

[0073] 所有历史窗口的密度曲线如图4所示, 可以看出, 有几个窗口输出的密度非常大, 爬取这些窗口的可疑密集块对应的微博, 经过观察和分析, 确定了这些可疑密集块对应的内容主题, 用多边形图标表示。这些可疑密集块的具体信息如表2所示, 特别值得注意的是, 有一个可疑密度块有953条边, 但是只有7用户 $\times$ 8消息, 这意味着每个用户在两小时内平均转发一个消息20次, 非常具有嫌疑。所以, EigenPulse可以检测到真实数据集中的存在异常的密集块。

[0074] 对比EigenPulse和DenseAlert在表1前5个数据集上的运行时间。设置两个算法的滑动窗口为 $w=30\text{day}$ ,  $s=10\text{day}$ , 运行时间如图2所示。可以看出, EigenPulse对比DenseAlert速度提高了至少2.53倍, 在Amazon CellPhone数据集上甚至提高了12.2倍。

[0075] 表1数据集信息表:

名字	节点数	边数	时间跨度
Amazon Electronic	4.20M × 476K	7.82M	1998.12 - 2014.7
Amazon Grocery	763K × 165K	1.29M	2007.1 - 2014.7
Amazon Cellphone	2.26M × 329K	3.45M	2007.1 - 2014.7
BeerAdvocate	26.5K × 50.8K	1.08M	2008.1 - 2011.11
Yelp	686K × 85.3K	2.68M	2004.10 - 2016.7
Sina Weibo	2.74M × 8.08M	50.06M	2013.11 - 2013.12

[0076] 表2可疑窗口的可疑子图信息表:

主题	可疑块大小	窗口时间跨度	可疑块边数
中国电信促销活动	39 × 57	6:00~8:00, 11.7	2,004
	78 × 58	7:00~9:00, 11.7	4,051
	151 × 119	8:00~10:00, 11.7	8,295
双十一广告	201 × 139	6:00~8:00, 11.10	7,012
	196 × 111	7:00~9:00, 11.10	9,668
	126 × 93	8:00~10:00, 11.13	638
歌手王栎鑫新专辑广告	7 × 8	22:00~24:00, 11.26	953
感恩节商家广告	26 × 36	23:00, 11.26 ~ 1:00, 11.27	629
	43 × 34	1:00~3:00, 11.27	263

[0079] 以下为与上述方法实施例对应的系统实施例,本实施方式可与上述实施方式互相配合实施。上述实施方式中提到的相关技术细节在本实施方式中依然有效,为了减少重复,这里不再赘述。相应地,本实施方式中提到的相关技术细节也可应用在上述实施方式中。

[0080] 本发明还提供了一种针对流式图的密集子图检测系统,其中包括:

[0081] 增广矩阵生成模块,用于持续从社交网络获取以流式图表示的三元组,该三元组

由用户、对象和时间戳组成,通过将时间戳拼接到对象上作为行,用户作为列,将该流式图建模为行增广矩阵;具体实施中对象视社交网络的不同而不同,例如若社交网络是购物网站,则对象是商品,若社交网络是微博平台,则对象是微博。

[0082] 奇异值分解模块,用滑动窗口访问行增广矩阵,并对每个窗口内的行增广矩阵进行奇异值分解,得到奇异矩阵 $U$ 、 $S$ 和 $V$ ,获取 $U$ 、 $V$ 奇异矩阵的奇异向量对 $(u, v)$ ,设置向量阈值筛选 $u, v$ 向量上的值,得到候选密集块及该候选密集块的密度;

[0083] 检测模块,集合每个窗口输出的该密度,得到密度历史集合,设置密度阈值为 $\mu+3\sigma$ ,其中 $\mu$ 是该密度历史集合的平均值, $\sigma$ 是该密度历史集合的标准差,若 $t$ 时刻窗口的密度大于该密度阈值,则判定 $t$ 时刻窗口的候选密集块中的用户和对象分别为异常用户和异常目标,将该异常用户和该异常目标作为检测结果输出。

[0084] 该针对流式图的密集子图检测系统,其中该增广矩阵生成模块,包括:

[0085] 持续从社交网络获取三元组,并将所有的用户记作集合 $B_1$ ,所有的对象记作集合 $B_2$ ,构造二部图 $\mathcal{G} = (V, E)$ ,其中 $E$ 是图的边集合, $V$ 是图的节点集合, $V = B_1 \cup B_2$ ,将 $\mathcal{G}$ 作为该流式图,建模为该行增广矩阵。

[0086] 该针对流式图的密集子图检测系统,其中该奇异值分解包括:

[0087] 定义随机矩阵 $\Omega$ 大小为 $n \times l$ ,其中 $l < \min(m, n)$ ;窗口内的行增广矩阵的大小为 $m \times n$ ;

[0088] 定义两个列表 $glist, hlist$ ,结合滑动窗口,通过下式每次计算一个步长 $s$ 的窗口内的行增广矩阵 $a$ ,生成对应的矩阵 $g, h$ ,分别存入 $glist, hlist$ ;

[0089]  $g = a\Omega; h = a^T g$

[0090] 遍历完该行增广矩阵后,将 $glist$ 中所有的矩阵 $g$ 按行拼接生成矩阵 $G$ ,将 $hlist$ 中所有的矩阵 $h$ 相加生成矩阵 $H$ ;

[0091] 利用已有的Single-pass PCA算法,根据矩阵 $G, H$ 得到矩阵 $Q, B$ ,再根据矩阵 $Q, B$ 生成奇异矩阵 $U, S$ 和 $V$ 。

[0092] 该针对流式图的密集子图检测系统,其中该奇异值分解模块中筛选过程包括:

[0093] 设置 $u$ 向量的向量阈值为 $\tau_u = \frac{1}{\sqrt{m_t}}$ , $m_t$ 代表 $t$ 时刻窗口对应的行增广矩阵的行数, $v$

向量的向量阈值为 $\tau_v = \frac{1}{\sqrt{n_t}}$ , $n_t$ 代表 $t$ 时刻窗口对应的行增广矩阵的列数;

[0094] 获取第 $t$ 时刻窗口的奇异向量对 $(u_t, v_t)$ , $u_t \subseteq U_t, v_t \subseteq V_t$ ,遍历 $u_t$ 向量的值,提取出所有不小于阈值 $\tau_u$ 的行,构成集合 $rowset$ ;遍历 $v_t$ 向量的值,提取出所有不小于阈值 $\tau_v$ 的列,构成集合 $colset$ , $rowset$ 中的行和 $colset$ 中的列构成了窗口 $[t, t+w]$ 的候选密集块 $B_t$ , $w$ 为窗口的大小;

[0095] 通过 $B_t$ 的边数和除以 $B_t$ 的行列数之和,得到候选密集块的密度。

[0096] 该针对流式图的密集子图检测系统,其中该筛选过程还包括:根据密集块检测算法HoloScope或Fraudar,在候选密集块 $B_t$ 上进一步寻找更密集的子块,形成最终的候选密集块。

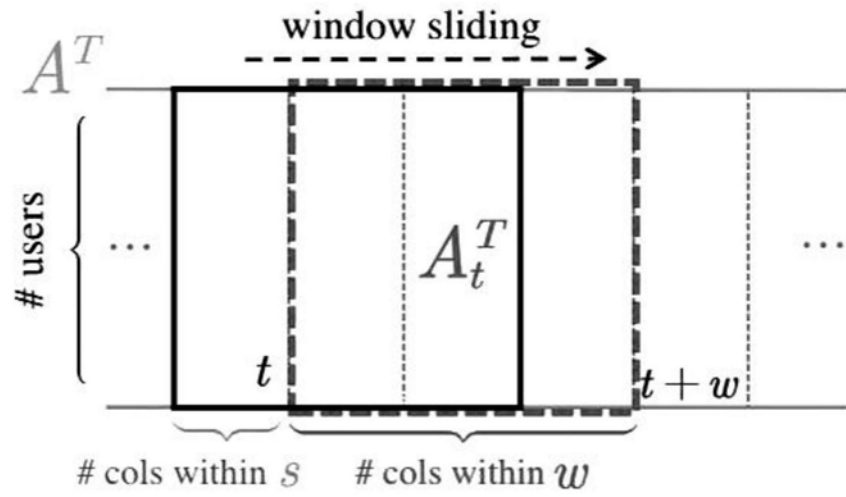


图1

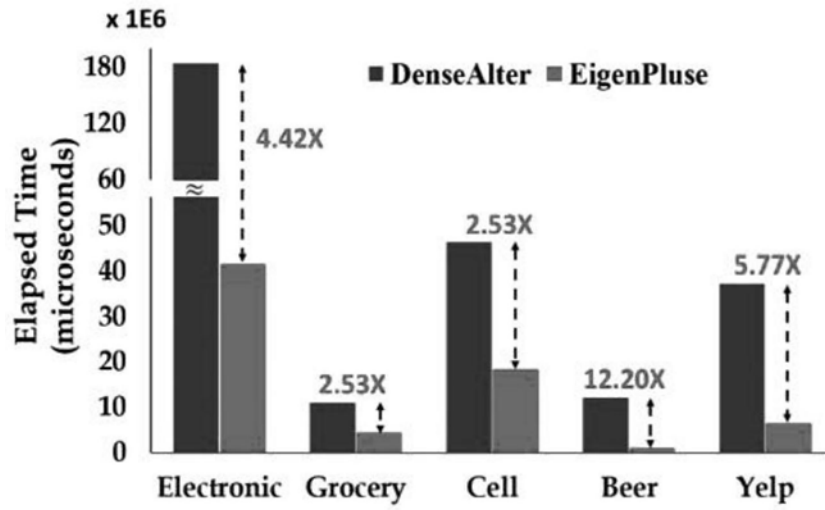


图2

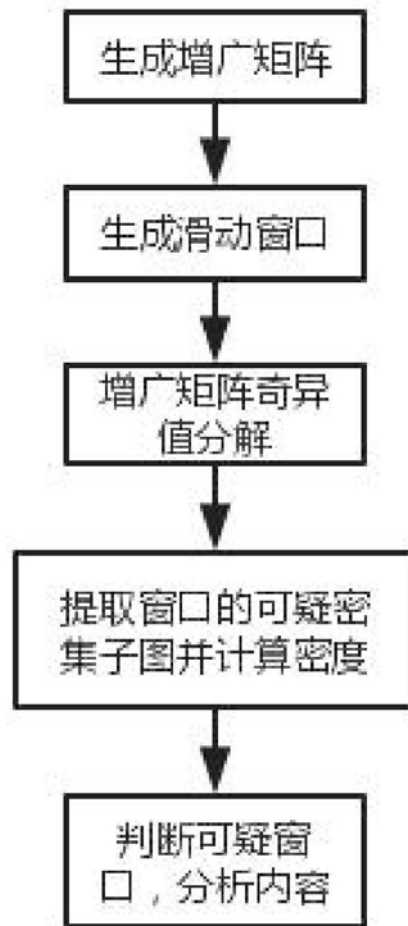


图3

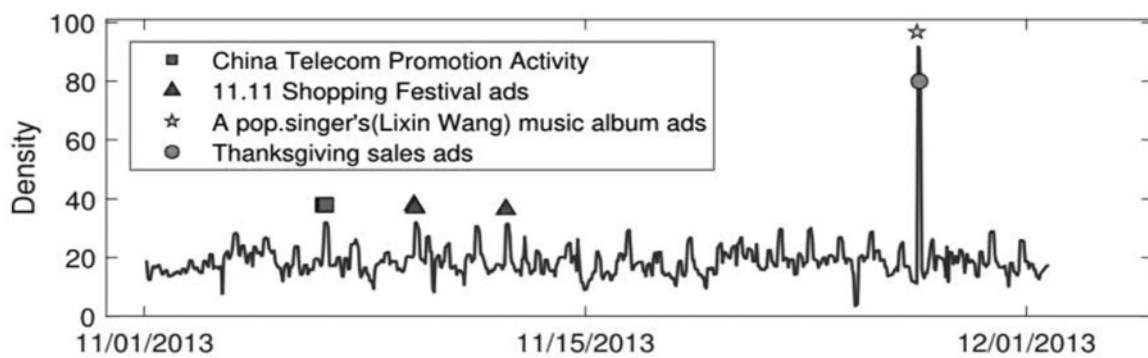


图4