

# D-VSSP: 分布式社会网络隐私保护算法

张晓琳 张 臣 张文超 张焕香 于芳名

(内蒙古科技大学信息工程学院 包头 014010)

**摘要** 针对传统社会网络隐私保护技术对大规模社会网络数据处理效率较低的问题,提出一种分布式结点分裂匿名社会网络隐私保护算法(Distributed-Vertex Splitting Social Network Privacy Preserving, D-VSSP)。D-VSSP 算法利用 MapReduce 和 Pregel-like 分布式计算模型处理社会网络图数据。首先基于 MapReduce 分布式计算模型对大图中的结点的标签信息进行标签平凡化,标签平凡化分组和精确分组处理;然后基于 Pregel-like 的消息传递机制,选举结点分裂,进行分布式结点分裂匿名。实验结果表明,在对大规模社会网络数据的处理效率上,D-VSSP 算法优于传统算法。

**关键词** 分布式算法,大规模社会网络,隐私保护,分布式结点分裂匿名

**中图分类号** TP309.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.02.012

## D-VSSP: Distributed Social Network Privacy Preserving Algorithm

ZHANG Xiao-lin ZHANG Chen ZHANG Wen-chao ZHANG Huan-xiang YU Fang-ming

(School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China)

**Abstract** The processing efficiency of traditional social network privacy preserving technology for large-scale social network data is low. To solve this problem, a distributed vertex splitting social network privacy preserving(D-VSSP) algorithm was proposed. D-VSSP algorithm deals the large-scale social network data in parallel with MapReduce computing model and Pregel-like model. Firstly, using MapReduce distributed model processes the vertex labels with method of label trivialization, grouping trivialized label and exact grouping. And then it realizes distributed vertex splitting anonymity based on the message passing mechanisms of Pregel-like through splitting vertex electing. The experimental results show that the D-VSSP algorithm is superior to the traditional algorithm in processing efficiency for large-scale social network data.

**Keywords** Distributed algorithm, Large-scale social networks, Privacy-preserving, D-VSSP

## 1 引言

社会网络是现实中许多真实复杂网络的抽象,蕴含着丰富的数据信息。现实世界中,社会网络随处可见,例如 Blog, Facebook 和 Twitter 等在线社交网站。社会网络数据发布可以为不同的研究工作者提供一些可靠的、实时的而且高价值的的数据。对社会网络数据的分析可用于知识决策、科学研究和数据共享。然而这些数据往往涉及用户隐私信息,发布和共享社会网络数据会导致隐私泄露<sup>[1]</sup>。

针对传统社会网络数据的隐私保护的技术有许多的研究成果。目前,有结点 K 匿名<sup>[2-5]</sup>、子图 K 匿名<sup>[6-10]</sup>、数据扰乱<sup>[11]</sup>和推演控制<sup>[12]</sup>等技术。Sun Y J 等人<sup>[13]</sup>提出了一种新颖的社会网络隐私保护技术,结点分裂匿名。通过这样的方式,结点的标签信息不会丢失,图中的边不会改变,匿名后的数据可用性高。结点匿名处理过程修改图结构,匿名后,对数

据隐私保护的程高度很高。但是,随着社会网络数据量的增加,传统社会网络隐私保护技术处理效率较差。目前,较少有针对大规模社会网络隐私保护的研究。Zakerzadeh H 等人<sup>[14]</sup>使用 MapReduce 对大规模社会网络图进行 t-diversity 匿名,处理过程中使用 MapReduce 模型对图进行处理,但是 MapReduce 处理图数据时,数据会反复迁移,处理时间长;提出的 SVFW 和 SVFG 算法存在过度保护,即将一些不需要保护的结点处理,使得数据可用性降低。

针对大规模社会网络数据,结合 MapReduce<sup>[15]</sup>和 Pregel-like<sup>[16]</sup>提出 D-VSSP 算法。使用 MapReduce 分布式平台处理原始的图数据,抽取原始数据中的标签信息并对结点信息匿名,处理后的结果供 D-VSSP 处理;使用 Pregel-like 系统以“结点为中心”的思想,通过结点之间的信息交换,在结点的邻域中选举分裂结点,被选举出来的结点进行分裂操作。实验结果证明,D-VSSP 算法能够提高对大规模图处理的效率。

到稿日期:2015-10-09 返修日期:2015-12-23 本文受国家自然科学基金:基于云计算的大规模社会网络隐私保护技术研究(61562065)资助。

张晓琳 女,博士,教授,主要研究方向为数据库理论与技术、信息安全,E-mail:zhangxl@imust.cn;张 臣 男,硕士生,主要研究方向为数据库理论与技术、大规模数据下社会网络隐私保护,E-mail:knightsark@163.com;张文超 女,硕士生,主要研究方向为数据库理论与技术、大规模数据下社会网络隐私保护;张焕香 女,讲师,主要研究方向为数据库理论与技术、大规模数据下社会网络隐私保护;于芳名 女,硕士生,主要研究方向为数据库理论与技术、大规模数据下社会网络隐私保护。

## 2 相关工作

### 2.1 社会网络图

社会网络可以抽象成图结构,一般的社会网络图可以通过带标签的无向图来表示。图中的结点代表了社会网络中的个体,结点之间的边代表了个体之间的关系,结点的属性代表了个体的特征。

**定义 1** 社会网络带标签无向图  $G = \{V, E, U\}$ , 其中,  $V$  表示结点集,  $V = \{v_1, v_2, \dots, v_n\}$ ,  $v_i$  代表图  $G$  中的结点,  $i \in [1, n]$ ;  $E$  表示边集,  $e_{ij} = e(v_i, v_j)$ , 其中  $v_i, v_j \in V$ ;  $U$  表示标签集合,  $u_i \in U, i \in [1, n]$ ,  $u_i$  表示结点  $v_i$  的标签集合。

社会网络带标签无向图结点如表 1 所列, 结点  $v_1$  的标签集合  $u_1 = \{\text{Alice}, \text{F}, 22, \text{UK}\}$  代表了结点  $v_1$  拥有 4 个属性, 分别为姓名、性别、年龄、国籍。图 1 中边  $e_{1,2} = (v_1, v_2)$  代表 Alice 和 Bob 具有敏感关系。

表 1 结点标签表

V	U				
	ID	Name	Gender	Age	Nationality
$v_1$	1	Alice	F	22	UK
$v_2$	2	Bob	M	27	UK
$v_3$	3	Carlo	M	23	USA
$v_4$	4	David	M	25	USA
$v_5$	5	Eve	F	26	USA
$v_6$	6	Francis	F	29	USA
$v_7$	7	Gerald	M	27	UK
$v_8$	8	Henna	F	22	UK

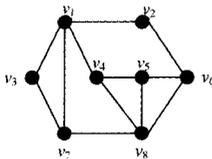


图 1 社会网络带标签无向图

### 2.2 结点信息匿名

**定义 2**(标签平凡化<sup>[6]</sup>) 给定原始图  $G$ , 能够唯一识别结点的标签集合  $S \subseteq U$ , 结点的标签集合为  $U$ , 则朴素匿名图为  $G' = \{V, E, U'\}$ , 其中  $U' = U - S = P\{u_1, u_2, \dots, u_r\}, r < n$ 。

标签平凡化的过程就是移除能够唯一识别结点的标签集合。经过结点标签平凡化处理, 得到的匿名图为  $G'$ , 其中, 结点集合  $V$  和结点边结合  $E$  不变。移除唯一标识标签后, 结点标签集合为  $U'$ 。

**定义 3**(平凡化标签分组<sup>[6]</sup>) 平凡化标签分组是将  $U_R$  通过  $G$  函数得到  $U_G = G(U_R, \Delta)$  的过程。其中  $U_G = \{u_{G1}, u_{G2}, \dots, u_{Gm}\}$  为分组后的标签集合,  $1 \leq m \leq n$ ;  $\Delta = \{\delta_1, \delta_2, \dots, \delta_r\}$  是数值标签的容差,  $r$  为数值型标签的个数。  $\forall u_{Ri} \in U_R, \exists u_{Gp} \in U_G$  满足对任意的非数值标签  $t_c$  都有  $u_{Ri}^c = u_{Gp}^c$ , 且对第  $s$  个数值标签  $t_{vs}$  都有  $|u_{Ri}^s - u_{Gp}^s| < \delta_s, 1 \leq s \leq r$ , 若对某个  $u_{R0}, \exists u_{Gp}, u_{Gq} \in U_G$  同时满足上述条件, 则  $p = q$  即  $u_{Gp} = u_{Gq}$ 。

在社会网络中, 两个相连的结点有很高的相似性, 将两个结点的数值型标签进行泛化后, 可以使一些相邻结点彼此不可区分。

**定义 4**(精确分组<sup>[6]</sup>) 将  $G$  按照数值标签容差集  $\Delta$  变成  $G_G = (V, E, U_G)$  的过程。  $U_G = \{u'_{G1}, u'_{G2}, \dots, u'_{Gn}\}$ , 其中  $u_i$  经平凡化标签分组后对应于  $u'_{Gi}, 1 \leq i \leq n$ 。

对表 1 中的结点标签进行处理, 通过标签平凡化, 得到 4 种标签类型。第一种类型为  $\{F, (20, 25], \text{UK}\}$ , 简记为 A; 第二种类型为  $\{F, (25, 30], \text{UK}\}$ ; 第三种类型为  $\{F, (20, 25], \text{USA}\}$ ; 第四种类型为  $\{F, (25, 30], \text{USA}\}$ 。  $u_1$  标签属于 A 类型。则最后结点  $v_1$  的标签为 A。

### 2.3 结点分裂匿名

**定义 5**(结点分裂匿名<sup>[6]</sup>) 将结点进行  $m$  阶分裂的过程。结点  $v$  的  $m$  阶分裂是用  $m$  个子结点  $v_1, v_2, \dots, v_m$  代替结点  $v$  的过程, 并且  $m$  阶分裂需要满足:

- 1) 所有的结点与原始结点有相同的标签。
- 2)  $\forall (v, v_x) \in E$ , 有且仅有一条对应的  $(v_i, v_x), 1 \leq i \leq m$ 。

当且仅当  $||v_i|_A - |v_j|_A| \leq 1$  时, 称为结点的  $m$  阶分裂, 其中  $|v_i|_A = |\{(v_i, v_x) | u_{v_x} = u_A, (v_i, v_x) \in E\}|$ 。图  $G$  进行结点分裂匿名后, 得到  $G' = \{V', E', U'\}$ 。其中,  $V'$  为匿名后的结点集合,  $E'$  为边集合,  $U'$  为标签集合。

图 1 中, 结点  $v_1$  的度为 4, 即有 4 个邻居, 分别为  $v_2, v_3, v_4, v_5$ 。如果进行 2 阶分裂, 就产生结点  $v_1$  的两个子结点  $v_{11}, v_{12}$ , 然后从结点  $v_1$  的邻居集合中随机选出两个邻居  $v_2, v_4$  分配给  $v_{11}$ , 其余的邻居分配给  $v_{12}$ , 得到边  $e_{11,2}, e_{1,3}, e_{11,4}, e_{1,7}$ 。

## 3 分布式环境下结点分裂隐私保护方法

### 3.1 分布式结点信息匿名

针对单机串行处理方式的不足, 提出分布式结点分裂隐私保护算法 D-VSSP。

首先使用 MapReduce 对原始数据进行数据处理。MapReduce 从原始数据中读取  $\langle \text{key}, \text{value} \rangle$  键值对, 通过条件判断, 移除唯一性结点标签, 实现标签平凡化。map 任务接收一个  $\langle u_{id}, id \rangle$  的键值对,  $u_{id}$  中存在数值型标签, 进行泛化处理, map 任务结束。MapReduce 框架会将 map 函数产生的中间键值对中键相同的值传递给同一个 reduce 函数, 这样有相同标签的结点就会放入到一个 reduce 任务中。reduce 任务会接收一个键, 以及与此键相关的一组值  $\langle u_{id}, id\text{-list} \rangle$ , 将这一组值合并, 得到结点标签的映射表, 供查询使用。得到标签映射表的过程为标签平凡化分组。reduce 产生的结果保存成文件。通过标签映射表将结点标签替换为简记标签编号, 实现精确分组。

#### 算法 1 分布式结点信息匿名

输入: 原始图  $G(V, E, U)$ , 阈值集合  $\Delta$   
 输出: 结点信息匿名图  $G'(V, E, U')$

1. Upload original graph to HDFS
2. TextInputFormat() /\* 从文件中读取数据 \*/
3. ReadFileByLine line
4. while(line is not null)
5. each line split by sign  
/\* 按标志划分每一行信息 \*/
6. deal the numeric label with threshold set  $\Delta$   
/\* 处理结点的数值型属性 \*/
7. mapper(key, value)  
/\* key 为结点的标签信息, value 为结点的 id \*/
8. emit(key, value)  
/\* 输出 Map 的中间运算结果 \*/
9. reducer(key, value)  
/\* 将 key 相同的数据放入同一个桶中 \*/

```

10. emit(key,value)
    /* 输出 reduce 的运算结果 */
11. SaveAsTextFile()
12. return Initial Graph
    
```

以表 1 为例,通过移除唯一型属性和泛化数值型属性后,得到 4 种类型的标签,分别为  $\{F, (20, 25], UK\}$ ,  $\{F, (25, 30], UK\}$ ,  $\{F, (20, 25], USA\}$ ,  $\{F, (25, 30], USA\}$ ,再给每一种标签编号。将每个结点的标签信息变成标签编号,如结点  $v_1$  的标签  $u_1 = A$ 。

### 3.2 分布式结点分裂匿名

社会网络结构复杂,将社会网络划分出层次结构很困难。结点分裂匿名过程中,如果一个分裂结点的 1-邻域内出现另一个分裂结点,这两个结点在进行严格的  $m$  阶分裂时,两个分裂结点之间的边有可能会分裂,破坏结点严格的  $m$  阶分裂。图中边的数量改变,会导致数据可用性降低。因此,为了保证分裂结点的 1-邻域内没有其他的分裂结点,结合 Pregel-like 计算模型中以结点为中心的原理,通过结点间的消息交换,在结点的 1-邻域内选举出分裂结点。

**定义 6(选举分裂结点)**  $V_{n_i}$  为结点  $v_i$  的邻居结点集合,选举的分裂结点集合为  $V_s, V_s$  满足:

$$V_s = \{v_i | d(v_i) > d(v_x), v_x \in V_{n_i} \text{ or } d(v_i) = d(v_x), i > x, v_x \in V_{n_i}\}$$

其中,  $d(v_i)$  为结点  $v_i$  的度。

经过 MapReduce 处理后,社会网络图数据将会被 master 划分到不同的 worker 上。任务开始时,每个结点为激活状态,结点之间使用 SendMessage 函数发送消息。在第一个超步中,每个结点发送自己的编号和度值给邻居;第二个超步中,结点接收到来自邻居的消息,能够构造出结点的 1 邻域子图。使用邻居消息进行分裂结点的选举,选举出来的分裂结点执行分裂操作,然后将分裂的信息反馈给邻居。分裂后的结点变成静默状态。当超步数不小于 2 时,结点接收到的消息为结点分裂消息。通过结点分裂消息将自己的邻居进行修改,把修改过的结点的消息从消息映射中删除。每次消息交换完成后,通过 VertexProgram 函数对消息进行处理,选举分裂结点。分裂后的结点进入静默状态。迭代执行,直到所有结点都为静默状态,程序停止,算法过程如下所示。

#### 算法 2 分布式结点分裂匿名

输入:初始化图  $G(V, E, U)$ , 分裂阶数  $m$   
 输出:匿名图  $G'(V', E', U')$

```

1. each vertex  $v \in V$  initial message queue and  $step = 0$ 
    /* 初始化消息队列超步数量为 0 */
2. for each vertex  $v \in V$  satisfied AllVertexIsNotSilent
    /* 所有结点不为静默状态 */
3. if current  $step = 1$  then
4.   SendMessage(E.destinateId, Message1)
    /* 发送 id 和度消息给邻居结点 */
5. else if queue is nonEmpty and the vertex hasnot splitted then
6.   sendMessage (E.destinateId, Message2)
    /* 发送分裂消息给邻居 */
7.   else
8.     sendMessage (Empty)
    /* 发送空消息给邻居 */
9.   end if
    
```

```

10. end if
11. MergeMessage() /* 合并消息 */
12. if (message.nonEmpty) then
    /* 接收到的消息不为空 */
13.   if ( $step = 1$ )
14.     AddMessageToMap(message)
    /* 接收到消息后,将消息放入 Map 映射中 */
15.   else
16.     DropTargetFromMap(message)
    /* 将消息从 Map 映射中删除 */
17.     ChangeTheEdge(message)
    /* 通过消息修改边 */
18.   end if
19.   if (selected as split vertex)
20.     SpiltTheNode(m) /* 结点 m 阶分裂 */
21.   else
22.     Silent
23.   end if
24. else
25.   Silent
26. end if
27.  $step = step + 1$ 
28. end for
29. return G
    
```

原始图进行信息匿名后再进行分布式结点分裂匿名。在超步 1 中,所有的结点处于激活状态,结点将自己的编号和度发送给邻居,消息的格式为(结点编号,结点的度)。结点  $v_2$  发送消息(2,3)给结点  $v_1$ 。图 2(a)中只列出了结点  $v_1$  收到的消息。在超步 2 中,结点接收消息,结点  $v_1$  接收到来自  $v_2, v_3, v_4, v_7$  的消息,消息内容分别为(2,3), (3,2), (4,3), (7,3)。通过结点选举  $v_1$  分裂。同理,结点  $v_8$  也分裂。假设结点分裂为 2 个,如图 2(b)所示,结点分裂成  $v_1$  和  $v_{11}$ ,  $v_1$  和  $v_{11}$  拥有相同的标签;然后将分裂后的原始标签和子结点标签发送给邻居。如图 2(c)所示,结点  $v_2$  和  $v_4$  接收到来自  $v_{11}$  的消息后,修改边信息,将结点  $v_1$  的消息从消息映射中删除;然后结点  $v_1$  和  $v_{11}$  转换成静默状态。结点  $v_8$  执行相同的操作。经过多次迭代执行,直到所有结点变成静默状态,程序停止。在结点分裂的过程中,结点分配边的过程随机,会导致结点分裂的结果不唯一。最后,一种可能的结点分裂结果如图 2(d)所示。

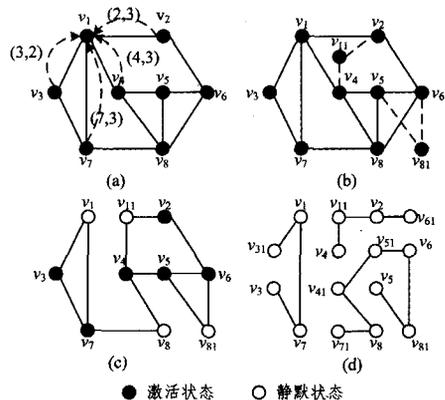


图 2 分布式结点分裂过程

### 4 隐私保护的安全性数据的可用性

#### 4.1 数据可用性

分布式节点分裂匿名过程中,首先对节点进行标签匿名,然后再进行分裂匿名,整个处理过程只是对节点的标签进行修改;然后节点分裂,但是节点的标签信息没有进行修改,节点之间的连接关系没有改变,节点的边也没有进行修改,所以边上的信息没有损失。

**定义 7** 图中的聚类查询分为两类<sup>[6]</sup>。第一类是只针对节点的存在性的聚类查询;第二类查询是在边上存在的数量的聚类查询。

**定理 1** 在经过严格阶分裂后的匿名数据可以在容差  $\Delta$  范围内无误差回答第 1 类查询<sup>[6]</sup>。

**定义 8(可预见误差)** 在查询结果给出前可以事先通过理论计算得到的查询误差。

**定理 2** 在经过严格  $m$  阶分裂后的匿名数据可以在容差  $\Delta$  范围内以可预见误差回答第 2 类查询<sup>[6]</sup>。

#### 4.2 安全性分析

针对社会网络的各种攻击,通过不同的背景知识能够导致不同的隐私泄露。攻击者在没有任何背景知识的情况下,对于节点分裂匿名后的数据,不存在隐私泄露。如果只知道单个节点的信息,比如度信息或者标签信息,则识别结点的概率为  $\frac{1}{C_{nm}^m}$ ,  $m$  为分裂的阶数,  $n$  为与目标节点有相同标签的节点的个数。假如攻击者知道节点的 1 邻域子图且节点数量为  $n$ ,则识别出单个节点的概率为:

$$\frac{1}{NC_n^a C_{n-a}^{a-1} C_{n-2a}^{a-2} \dots C_{n-ia}^{a-i} C_{n-ia-b}^b \dots C_{n-ia-j}^j}$$

其中,  $N$  为与目标节点有相同邻居数量且精确分组后节点标签相同的节点的数量,  $i+j=m, a=b+1, a=\frac{n}{m}$ , 这个概率很小。

由此可知,节点分裂匿名的隐私保护级别很高。

#### 4.3 开销分析

在 BSP 的一个超级计算步中,可以抽象出 BSP 的开销模型,一个超级计算步的开销为:

$$cost = \max(W_i) + \max \max(h_i, g) + L$$

其中,  $W_i$  是进程  $i$  的局部计算时间,  $h_i$  是进程  $i$  发送或接收的最大通信包数,  $g$  是带宽的倒数(时间步/通信包),  $L$  是同步时间。所以,在 BSP 计算中,如果用了  $s$  个超级步,则总的运行时间为:

$$T_{BSP} = \sum_{i=0}^{s-1} W_i + g \sum_{i=0}^{s-1} h_i + sL$$

通过上式, BSP 模型的开销主要由通信开销、计算开销和同步开销决定。因为 Pregel-like 模型是基于 BSP 实现的,所以整体开销是由计算、网络通信和同步决定的。在网络稳定的情况下,主要时间是由算法中的超级步和消息传递量决定的。D-VSSP 算法中,超级步的数量取决于图的直径。

### 5 实验与分析

#### 5.1 实验环境与数据集

本实验所使用的数据集均为斯坦福大学数据平台提供

的真实数据集<sup>1)</sup>。其中, Friendster 和 cit-Patents 均为真实社会网络数据集。以边为单位,将 cit-Patents 分割成若干个边数量相等的数据片段,然后将数据片段组合成 1:2:3 的数据片段 split1, split2 和 split3。其中具体参数如表 2 所列。实验中使用 12 台 worker 搭建的 Hadoop 2.5.2 集群,硬件配置为 CPU1.8GHz, RAM 16G。安装的 Spark 版本为 1.3.1, giraph 版本为 1.1.0。

表 2 数据集

Name	Vertices	Edges
com-Friendster	65608366	1806067135
cit-Patents	3774768	16518948
split1	806647	4000000
split2	1525139	8000000
split3	2203846	12000000
Epinions	131828	841372

#### 5.2 处理时间

使用分布式节点分裂匿处理大规模社会网络图数据,处理平台分别使用 spark, giraph 系统和原始算法。对原始数据进行处理时,分别使用不同大小的数据集,使用 2 个 worker 进行分布式处理。从图 3 中可以看出,在对 4 个不同的数据集进行处理时,若数据量小,原始算法和 D-VSSP 算法所用时间相差不多,两种平台使用 D-VSSP 算法运行时间相差不多;当数据量超过百万时, giraph 和 spark 的处理时间要少于原始算法的处理时间,同时 giraph 和 spark 的处理时间有差异,使用 Spark 平台要优于 giraph 平台,但两种平台相对于单机串行处理都有较高的处理效率。2 个 worker 的速度是单机的 1.5 倍左右,若 worker 数量增加,处理的效率将会提高。将其运用于大规模图数据的效率将会是显著提升的。

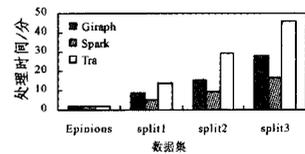


图 3 数据处理时间

处理时间随各参数的变化情况如图 4 所示。

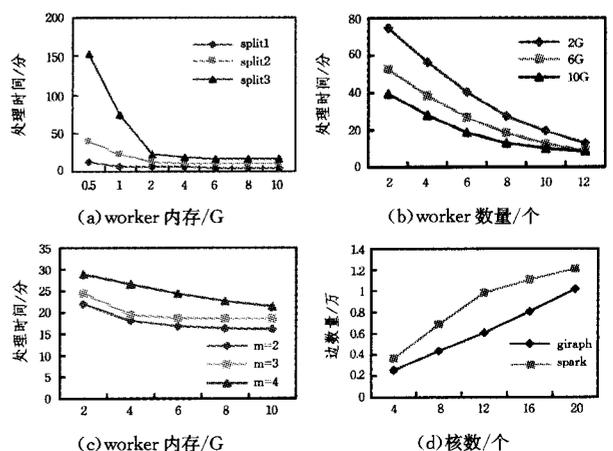


图 4 处理时间随各参数的变化情况

图 4(a)中,使用 4 个 worker,节点分裂数目  $m=2$  时,改变每个 worker 的处理内存。当数据集小时,修改内存,处理时间变化不明显。使用大的数据集时,处理时间变化明显。

1) 斯坦福大学数据集网址: <http://snap.stanford.edu/data/index.html>

当内存小数据集大时,处理时间尤为明显,主要是因为内存小时,数据集频繁地调入调出内存会消耗时间。当内存相同时,随着数据集的增大,处理时间增加。当内存足够时,增加内存不能明显加快计算速度。

图 4(b)中,worker 数量递增,处理数据集 split3 时,随着 worker 数目的增加,处理时间减少,但是处理时间的减少不是呈线性的。处理时间图线的斜率逐渐降低,说明随着 worker 数量的增加,各个结点的通信数量增加,通信延迟加快了速度的降低。

图 4(c)中使用 4 个 worker 对 cit-Patents 数据集做处理时,分裂的结点数  $m$  的变化对图处理效率的影响不大。当结点分裂  $m$  值变大时,处理时间变长。

图 4(d)中对 cit-Patents 数据集分析处理,其结果显示了每个 worker 的核每秒能够处理边的数量。可见,Spark 平台使用相同资源对大规模图处理的效果更加理想。

### 5.3 数据可用性

针对两类查询,提出 4 种查询条件,查询 1 是只针对单一结点信息的查询,查询 2 是精确约束查询,查询 3 是双边约束查询,查询 4 是单边约束查询。分别在原始数据集和匿名集合中进行查询,结果如图 5(a)所示。对于第一类查询,数据可用性高,与原始数据的查询结果相近。对于第二类查询,查询的相对误差主要由可预见误差构成,经过修正<sup>[6]</sup>后,数据的可用性高。当  $m=2$  时,针对不同的数据集,查询的效果与原始图相差不大,而且相对误差率在都在 5% 以下。图 5(b)中,对不同的数据集,查询相对误差与图性质的关系不大,同样,相对误差率在 5% 以下,数据可用性高。

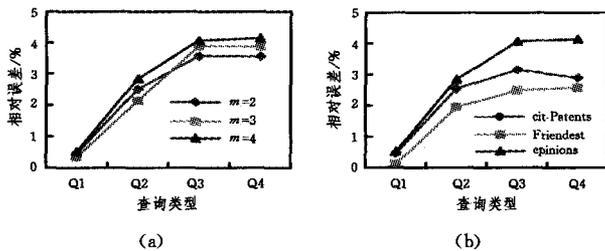


图 5 查询相对误差

### 5.4 数据安全性

通过对匿名后的数据进行测试,统计每个结点的候选结点。通过结点的候选结点数量的分布情况来反映结点的安全性,候选结点数量越少的结点的数量越多,则越不安全。图 6 为对匿名后结点的候选集合进行统计的结果。在较小的数据集中,能够以大于 0.1 的概率识别出来的结点的数量占整个图中结点数量的比例不足 4‰;在大数据集中,能够以大于 0.1 的概率识别出来的结点的数量占整个图中结点数量的比例不足 1‰。所以通过结点分裂匿名后的图数据的安全性较高。

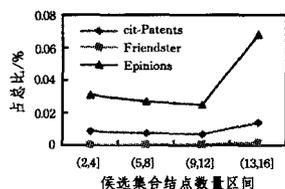


图 6 安全性测试

**结束语** 本文针对传统社会网络隐私保护技术对大规模社会网络数据处理效率较低的问题,提出一种分布式结点分

裂匿名社会网络隐私保护算法 D-VSSP。在不降低数据可用性和数据匿名性的前提下,利用传统社会网络隐私保护技术在分布并行的环境中进行大规模社会网络数据的处理。实验结果证明,D-VSSP 算法是一种高效的社会网络隐私保护算法。下一步计划将算法应用到动态社会网络中以及个性化社会网络隐私保护中。

### 参考文献

- [1] LIU X Y, WANG B, YANG X C. Survey on privacy preserving techniques for publishing social network data [J]. Journal of Software, 2014, 25(3): 576-590.
- [2] TAI C H, YU P S, YANG D N, et al. Privacy-preserving social network publication against friendship attacks [C] // ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, CA, USA, August 2011: 1262-1270.
- [3] CAMPAN A, TRAIAN M. A clustering approach for data and structural anonymity in social networks [J]. In Privacy, Security, and Trust in KDD Workshop, 2008, 2(8): 33-54.
- [4] CORMODE G, SRIVASTAVA D, YU T, et al. Anonymizing bipartite graph data using safe groupings [J]. Vldb Journal, 2008, 1(1): 833-844.
- [5] HAY M, MIKLAU G, JENSEN D, et al. Resisting structural re-identification in anonymized social networks [J]. Proceedings of the Vldb Endowment, 2008, 1(1): 797-823.
- [6] WANG Y, XIE L, ZHENG B, et al. Utility-Oriented K-Anonymization on Social Networks [C] // Database Systems for Advanced Applications-16th International Conference (DASFAA 2011). Hong Kong, China, 2011: 78-92.
- [7] CHENG J, FU W C, LIU J. K-isomorphism: Privacy preserving network publication against structural attacks [C] // ACM SIGMOD International Conference on Management of Data (SIGMOD 2010). Indianapolis, Indiana, USA, June 2010: 459-470.
- [8] WU W, XIAO Y, WANG W, et al. k-symmetry model for identity anonymization in social networks [C] // International Conference on Extending Database Technology. ACM, 2010: 111-122.
- [9] ZOU L, CHEN L, ZSU M T. k-automorphism: a general framework for privacy preserving network publication [J]. Proceedings of the Vldb Endowment, 2009, 2(1): 946-957.
- [10] LIU X, YANG X. A Generalization Based Approach for Anonymizing Weighted Social Network Graphs [M] // Web-Age Information Management. Springer Berlin Heidelberg, 2011: 118-130.
- [11] DAS S, EGECIOGLU O, EL ABBADI A. Anonymizing weighted social network graphs [C] // International Conference on Data Engineering. IEEE, 2010: 904-907.
- [12] LIU X Y, YANG X C. Protecting sensitive relationships against inference attacks in social networks [C] // International Conference on Database Systems for Advanced Applications. Springer-Verlag, 2012: 335-350.
- [13] SUN Y, YUAN Y, WANG G, et al. Splitting anonymization: a novel privacy-preserving approach of social network [J]. Knowledge and Information Systems, 2015, 1(2): 1-29.
- [14] ZAKERZADEH H, AGGARWAL C C, Barker K. Big Graph Privacy [C] // EDBT/ICDT Workshops. 2015: 255-262.
- [15] QIN L, YU J X, CHANG L, et al. Scalable big graph processing in MapReduce [C] // SIGMOD. 2014: 827-838.
- [16] SALIHOGLU S, WIDOM J. Optimizing graph algorithms on pregel-like systems [J]. Proceedings of the Vldb Endowment, 2014, 7(7): 577-588.