



大数据云环境下 TDS 和 BUG 混合 k -匿名化方法

范晓峰¹, 闫凤², 刘洋³

(1. 内蒙古商贸职业学院, 内蒙古 呼和浩特 010070;

2. 内蒙古农业大学职业技术学院, 内蒙古 包头 014109;

3. 呼和浩特职业学院, 内蒙古 呼和浩特 010050)

摘要:针对一般子树匿名化方法处理大数据效率低和伸缩性较差的问题,提出了一种可伸缩的自下向上的泛化(BUG)方法,并在此基础上,结合已有的自上向下的特化(TDS),形成一种混合方法。在提出的方法中, k -匿名作为隐私模型,TDS 和 BUG 都是基于映射化简开发组成,并通过云的强大计算能力来获得较高的伸缩性。提出的映射化简 BUG 只需在几次泛化循环之后就可插入一个新的泛化候选,不会影响另一个泛化的信息损失。考虑到工作负载平衡点 K 与匿名参数 k 的复杂关系,将映射化简的 BUG 和 TDS 结合形成混合方法。实验结果验证了本文方法的有效性,与 TDS 和 BUG 相比,混合方法的效率和可伸缩性大为提高。

关键词:云计算;子树匿名化;大数据;泛化;特化;映射化简

中图分类号:TP391

文献标识码:A

doi: 10.11959/j.issn.1000-0801.2016135

Hybrid k -anonymity approach based on TDS and BUG under the environment of big data cloud

FAN Xiaofeng¹, YAN Feng², LIU Yang³

1. Inner Mongolia Business & Trade Vocational College, Hohhot 010070, China

2. Vocational and Technical College of Inner Mongolia Agricultural University, Baotou 014109, China

3. Hohhot Vocational College, Hohhot 010050, China

Abstract: As the issue of low efficiency and poor scalability in general sub-tree anonymous method of treating big data, a bottom-up generalization(BUG) method with scalability was proposed, and on this basis, combined with the existing top-down specialization(TDS), a hybrid approach was formed. In the proposed method, k -anonymity was being as a privacy model, the compositions of TDS and BUG were developed with mapping simplification, and higher scalability through powerful cloud computing capabilities were achieved. The proposed mapping simplification BUG could insert a new candidate after several cycles of generalization, and would not affect information loss of another generalization. Given the complexity of the relationship between workload balancing point K and anonymous parameter k , mapping simplifications of BUG and TDS were combined to form a hybrid approach. Experimental results demonstrate the effectiveness of the proposed method and compared with TDS and BUG, the efficiency and scalability of hybrid method are greatly improved.

Key words: cloud computing, sub-tree anonymous, big data, generalization, specialization, mapping simplification

1 引言

目前,云计算和大数据^[1]具有颠覆性的趋势,对IT信息产业和研究领域都产生了重要影响^[1,2]。云计算可以通过使用大量电脑弹性地提供大规模的计算和存储能力,使用户能够在没有大量基础设施的情况下合理地配置大数据应用。为了开发公共云平台提供的优势,越来越多的大数据应用移入云端,因此,如何保护这些数据集的隐私成为一个重要的挑战,在云上分析和分享数据集之前,迫切需要处理数据隐私的问题。

对数据的隐私保护已经获得了广泛的研究,并取得了一定的进展。 k -匿名^[3]和 I -多样性^[4]分别是测量隐私敏感信息、对抗记录链接攻击和属性链接攻击程度的两种基本隐私模型。而一些匿名化操作可用于匿名化数据集,如抑制^[5]、泛化^[6]和解分^[7]等,本文采用泛化来匿名化数据集。关于泛化,一般有4种:全域泛化^[5]使一个属性的所有域值泛化到分类树的同一级上;子树泛化^[8]需要所有的子域值,但分类树中所有的中间节点都没有被泛化;多维泛化^[9]在泛化域值时考虑了多个属性;细胞泛化^[10]是局部重新编码并将相同的实例泛化到域值的不同级别中。

前3种泛化都需要全部重新编码,而细胞泛化不需要。本文主要讨论子树泛化方法,不同于多维或细胞泛化方法,子树泛化可以产生一致的匿名数据,该数据可以直接用于现存的数据挖掘和数据分析工具。一般有两种方法可以实现子树匿名化,即自上而下的特化(top-down specialization, TDS)和自下而上的泛化(bottom-up generalization, BUG)。迄今为止,研究者们为TDS和BUG提出了一系列的方法。

参考文献[11]提出了一种分布式的TDS方法,然而该方法主要关注其他评估的隐私保护而不是伸缩性问题。该方法仍采用信息获得作为搜索标准,导致其数据效用低。

参考文献[12]利用映射化简通过TDS来实现大数据匿名化需要的密集型运算。但是,自上而下的特化在 k -匿名参数很小的情况下比自下而上的泛化的执行速度还要慢。

为了研究隐私保护中的匿名化算法伸缩性和效率,参考文献[13]引入多属性支持的 k -匿名方法,利用可伸缩的决策树和采样技术来获得较高的伸缩性和效率。然而,该方法可以用于多维泛化方案,不能为子树泛化。

参考文献[14]提出了一种基于BUG的方法,利用索引数据结构来提高效率,从而不能在云环境中达到较高的伸缩性和并行化。因此,如何开发带有映射化简的BUG算

法,从而提升伸缩性和效率是一个有前途的方法,但缺少对指定的 k -匿名参数的准确获得。

目前,现有的TDS和BUG方法分别是为了子树泛化方案而开发。两者都缺少对使用者指定的 k -匿名参数的了解。事实上, k -匿名参数的值可以影响其性能。如果参数 k 较大,TDS更合适,而BUG的性能会较差;当 k 较小时,情况相反。同时考虑到映射简化广泛应用于各种数据处理应用程序来增加伸缩性和效率^[15]。本文针对大数据中的子树匿名化提出了一种联合了TDS和BUG的高度可伸缩的混合方法。当给定一个数据集时,该方法可以通过对比使用者规定的 k -匿名参数和来自数据集的阈值自动决定哪个组成进行匿名化。本文的主要贡献有3条:

- 通过自动选择TDS和BUG,提升子树数据匿名化的伸缩性和效率;
- 为BUG设计了一组创新的映射化简任务,以一种高度伸缩性的形式推导计算;
- 提出了在一次循环中执行多个泛化操作,推进BUG的并行处理能力和伸缩性。

2 子树泛化方案

本节简要介绍子树泛化,表1所示为一些基本符号和标记,一个分类树 TT_i 的叶节点值是 D 中记录的原属性值,使用 QI 作为准标识符。为不失一般性,本文采用 k -匿名化^[3]作为隐私模型,对于任意 $qid \in QID$, $QID(qid)$ 必须为0或至少为 k ,以便一个准标识符不会与至少 $k-1$ 个准标识符区分开。

在子树泛化方案中,可以采用两个操作,即BUG泛化和TDS特化。泛化操作就是在分类树中用中亲值取代域值,而特化操作是用所有的子值取代域值。泛化在形式上的表示为 $gen:Child(q) \rightarrow q$,而特化表示为 $spec:q \rightarrow Child(q)$,其中, $q \in DOM_i$ 为一个域值,集合 $Child(q)$ 包含了 q 的所有子域值。利用匿名级的概念^[3]来捕获匿名化的程度,AL匿名化程度用域值集的矢量表示,即 $AL = \langle Cut_1, Cut_2, \dots, Cut_m \rangle$,其中, $Cut_i (1 \leq i \leq m)$ 是 TT_i 的式样。

为了选出匿名化过程中最好的操作,通过搜索标准测量候选的泛化或特化,使用信息/隐私权衡作为本文方法的搜索标准,即分别为TDS的信息增益/隐私损失(IGPL)和BUG的信息损失/隐私增益(ILPG)。

考虑到泛化 $gen:Child(q) \rightarrow q$,泛化的ILPG计算如下:

$$ILPG(gen) = IL(gen) / (PG(gen) + 1) \quad (1)$$

其中, $IL(gen)$ 是执行 gen 后的信息损失, $PG(gen)$ 代表



表 1 基本符号标识和定义

符号	定义
D	包含数据记录的数据集
M	属性数量
r	数据记录, $r \in D$ 且 $r = \langle v_1, v_2, \dots, v_m, sv \rangle$, 其中, $v_i (1 \leq i \leq m)$ 为属性值, sv 为敏感值
$Attr_i$	数据记录的第 i 个属性
TT_i	属性 $Attr_i$ 的分类树
DOM_i	TT_i 中所有域值的集合
SV	敏感值集合
qid	准标识符, $qid = \langle q_1, q_2, \dots, q_m \rangle, q_i \in DOM_i, 1 \leq i \leq m$
QID	准标识符集 $QID = \langle Attr_1, Attr_2, \dots, Attr_m \rangle$
$QID(qid)$	具有相同准标识符 qid 的包含所有记录的准标识符组
$AGSet$	可用的泛化集
$SGSet$	同科泛化集
$NGSet$	新的泛化集
$AQISet$	匿名准标识符
$CGSet$	关键泛化集
$NCGSet$	非关键泛化集
$RGSet$	竞赛泛化集
$ACGSet$	可用关键泛化集

隐私增益,两者都是通过来源于数据集的统计信息计算得到, R_x 表示可以泛化成 x 属性值的原始记录集, $|R_x|$ 是 R_x 中数据记录的数量, $I(R_x)$ 是 R_x 的熵。 $IL(gen)$ 定义如下:

$$IL(gen) = \sum_{c \in Child(q)} \left(\frac{|R_c|}{|R_q|} \right) I(R_c) - I(R_q) \quad (2)$$

$A_d(gen)$ 表示执行 gen 后的匿名, 而 $A_b(gen)$ 是执行 gen 之前的匿名, gen 的隐私增益计算如下:

$$PG(gen) = A_d(gen) - A_b(gen) \quad (3)$$

映射化简是一种可延伸的容错数据处理结构, 具有简易性、伸缩性和容错性 3 种主要特征。映射化简任务包含两个原函数: map 和 reduce, 通过键值对 ($key, value$) 的数据结构定义。map 函数可以表示为 $map:(k_1, v_1) \rightarrow (k_2, v_2)$, map 采用 (k_1, v_1) 作为输入, 输出另一中间键值 (k_2, v_2)。这些中间的键会作为 reduce 的输入而消耗。reduce 函数在形式上可以表示为 $reduce:(k_2, list(v_2)) \rightarrow (k_3, v_3)$, 即 reduce 采用了中间值 k_2 及其对应值 $list(v_2)$ 作为输入并输出另一对 (k_3, v_3)。通常 (k_3, v_3) 列是映射化简获得的结果。map 和 reduce 函数都是用户根据其特定应用程序决定。

3 提出的混合泛化

本节主要描述映射化简的自下而上泛化 (MRBUG),

实际的映射化简程序包含了 map 和 reduce 函数, 用一个驱动程序来协调映射化简任务, 然后提出了一种混合泛化方法。

3.1 自下而上泛化的映射化简驱动程序

匿名化的自下而上泛化是一种循环过程, 最低匿名级包括分类树最低级别中的内域节点。每一次循环包括 4 个主要步骤, 即检测当前数据集是否满足匿名要求、计算 ILPG、寻找最佳的泛化以及根据选择的最佳泛化方法泛化数据集。

本文为 ILPG 计算设计映射化简任务, 由于匿名级的概念是一个数据集匿名化状态, 因此不需要将数据集具体泛化到每次循环。算法 1 为自下而上的泛化映射化简 (BUGMR) 驱动程序。

算法 1 MRBUG 驱动程序

输入 数据集 D , 匿名级 AL_0 , 匿名参数 k

输出 最终匿名化数据集 D^*

(1) 通过任务 ILPG 计算为每个关于 AL_0 的泛化 $gen \in \bigcup_{j=1}^m Cut_j$ 初始化搜索标准;

(2) 当 $\exists gen, A_b(gen) < k$;

(3) 从所有的可用泛化候选中确定可用泛化集 $AGSet$;

(4) $\forall gen \in AGSet$, 将 gen 标识为不活跃从而在当下匿名级上执行 gen ;

(5) 如果 $\exists gen \in AGSet, \forall gen \in SGSet(gen)$, 那么 gen 已经标识为不活跃;

(6) 向 $NGSet$ 中插入一个新的泛化 $gen_{New}: Child(q) \rightarrow q$, 其中, $Child(q) = \{q_i | gen: Child(q_i) \rightarrow q_i, gen \in SGSet(gen)\}$

(7) 移除 $SGSet(gen)$ 中的所有泛化;

(8) 结束条件;

(9) $AL_{i+1} \leftarrow AL_i$; 通过 ILPG 计算为 AL_{i+1} 中的所有活跃泛化候选更新 ILPG 值;

(10) 结束如果;

(11) 根据 AL_i 通过任务数据泛化将 D 泛化到 D^*

算法 1 的细节如下: 首先, 预设所有泛化的 ILPG 值 (第 1 行); 第 2 行检测当下匿名化数据集是否满足 k -匿名化的需要; 第 3 行识别表示为 $AGSet$ 的可用泛化集。基本上, $AGSet$ 包括最高 ILPG 值, 只根据上述 BUG 方法的第 3 步最佳泛化 gen_{Best} 。但是本文在一次循环中执行多次泛化, 从而提高并行化和效率的程度, 这将在第 3.2 节中详细描述。第 4 行通过将其标识为不活跃来执行 $AGSet$ 中的泛化过程。将一个泛化过程标识为不活跃意味着在接下来的循

环中将不再考虑该泛化过程,理论上在数据集上实现匿名化。当 $SGSet(gen)$ 中的泛化都标识为不活跃时,将会在匿名级中插入一个新的更高级别的泛化取代不活跃的泛化(第 5~7 行)。由于检测 $AGSet$ 中的多泛化,因此很有可能产生多个新的泛化。第 9 行更新了每个活跃泛化的隐私增益,这是因为 $AGSet$ 中执行的泛化可能会改变数据集的匿名。而且,如果已经插入了新的泛化,那么就需要计算信息损失,即 $NGSet \neq \emptyset$ 。第 11 行根据最终的匿名级,匿名化数据集。

第 1、9 行需要的 ILPG 估算包括访问原始数据集和计算数据集的统计信息。第 11 行也需要处理所有数据集。本文利用映射化简来执行这些情况下的密集型运算。本文设计了映射化简任务:为完成第 1、9 行所需的计算,进行 ILPG 估算任务,为获得第 11 行中的具体匿名化,进行数据泛化任务。

3.2 执行泛化的并行性

一些观察值也可以有助于设计出有效的 ILPG 的映射化简任务。自下而上的泛化在几次泛化循环之后插入一个新的泛化候选。另一个是根据式(2)泛化,不会影响另一个泛化的信息损失。根据这个观察值,本文可以在一次循环中考虑多个泛化候选,从而提高本文提出方法的并行性。然而,执行一次泛化可能会改变数据集的匿名,并影响每个泛化候选的隐私增益。为了确定一次循环可以同时考虑哪些候选,本文给出下列定义。

定义 1 (匿名标识符) 如果 $|QIG(qid)| = \min_{qid' \in QID} |QIG(qid')|$, 那么准标识符 qid 就是一个匿名准标识符,其中, $| \cdot |$ 表示 QI -组的大小。

具有一个匿名准标识符的 QI -组的大小取决于定义 1 中的数据集匿名化。值得注意的是,在一个匿名级中可能具有多个匿名准标识符。 $AQISet$ 表示匿名准标识符的集合。本文定义关键词泛化如下。

定义 2 (关键的泛化) 如果 $\exists q' \in Child(q), q' \in \bigcup_{qid \in AQISet} \{q_i | q_i = Proj_i(qid), 1 \leq i \leq m\}$ 那么泛化 $gen: Child(q) \rightarrow q$ 是关键的,其中, $Proj_i(qid)$ 是 qid 的第 i 个坐标。

$CGSet$ 表示关键泛化的集合,即 $CGSet = \bigcup_{qid \in AQISet} \{q_i | q_i = Proj_i(qid), 1 \leq i \leq m\}$ 。非关键泛化的集合表示为 $NCGSet$,且 $NCGSet = (\bigcup_{i=1}^m Cut_i) / CGSet$,根据上述定义得出:

如果一个泛化 $gen \in CGSet$, 执行该泛化可能会改变数据集的匿名,即 $A_p(gen) - A_b(gen)$ 可能会大于 0。相反,如果 $gen \in NCGSet$, $A_p(gen) - A_b(gen) = 0$, 则 $PG(gen) = 0$ 。

假设所有泛化候选都根据 ILPG 值进行分类,则本文

可以在第一个关键泛化之前同时执行所有候选而不影响匿名化的结果。为了确定在相同的循环中可以泛化哪些关键泛化,给出下列定义。

定义 3 (竞赛泛化) 假设所有泛化候选根据 ILPG 值进行分类。第一个关键泛化及其之后的持续关键泛化都是竞赛泛化。这些一起构成了竞赛泛化的集合,表示为 $RGSet$ 。

第一个关键泛化可以在相同的循环中明确执行,而 $RGSet$ 中的其他泛化不能,值得注意的是,执行关键泛化可能会影响候选的 ILPG 值,因此需要强制更新 ILPG 值。为了确定 $RGSet$ 中一次循环的可用关键泛化,算法 2 中给出了一个子程序。 $ACGSet$ 表示合成的可用关键泛化集,即 $ACGSet$ 中的泛化以及第一个关键泛化之前的泛化可以在一次循环中执行。在该算法中,使用一个优先序列来保持关于 ILPG 的泛化分类。

算法 2 确认可用泛化

输入 $RGSet, AQISet$

输出 $ACGSet$

- (1) $Queue \leftarrow \emptyset$, 其中, $Queue$ 是关于 ILPG 的优先排队;
- (2) 如果 $RGSet = \emptyset$;
- (3) $Queue \leftarrow g_{\min}, g_{\min} \in RGSet$ 且 $g_{\min}$ 有着最低的 ILPG;
- (4) $ACGSet \leftarrow g_{\min}$;
- (5) 如果 $Queue \neq \emptyset$;
- (6) $g \leftarrow Queue$ 且 $RGSet \leftarrow RGSet / \{g\}$;
- (7) $Queue \leftarrow \bigcup_{qid \in \{qid' | qid' \text{ contains } g \text{ and } qid' \in AQISet\}} \{q_i | q_i = Proj_i(qid), 1 \leq i \leq m\} / \{g\}$;
- (8) $AQISet \leftarrow AQISet / \{qid | qid' \text{ contains } g\}$;
- (9) 结束如果;
- (10) 结束如果

在将所有活跃泛化候选进行分类并识别了关键泛化之后,可以获得输入参数 $RGSet$ 。根据算法 2 中其他输入参数即 $AQISet$, 来确定 $CGSet$ 。因此,确定 $AQISet$ 是算法 2 的关键。第 3.3 节给出了如何在映射化简任务 ILPG 估算中确定 $AQISet$ 。一旦确定 $AQISet$, 就可以轻松构建可用泛化集 $AGSet$ 。

3.3 ILPG 估算任务

ILPG 估算任务负责算法 1 第 1 行中的 ILPG 初始化以及第 9 行中的 ILPG 更新。ILPG 初始化所需要的算法和 ILPG 更新所需算法十分相似。算法 3 描述了 ILPG 估算中的 map 函数,而算法 4 显示了 reduce 函数。在算法 3 和算法 4 中,“#”用来确定是否发出一个密钥来计算信息增益



或匿名损失。“\$”用来区分一个密钥是为了计算 $A_p(spec)$ 还是 $A_b(spec)$ 。

算法 3 ILPG 估算中的 map

输入 数据记录 $(ID, r), r \in D$; 匿名级, $NGSet$

输出 中间键值对 $(key, count)$

(1) 对于 r 中的每个属性值 v_i , 找出其在当下 AL 中的泛化 gen_i 。 p_i 为 gen_i 中的中亲, 且 c_i 是 v_i 本身或者 p_i' 的子集也是 v_i' 的前身;

(2) 如果 $gen_i \in NGSet$, 那么发出 $(\langle p_i, c_i, sv \rangle, count)$;

(3) 构建准标识符 $qid = \langle q_1, q_2, \dots, q_m \rangle$, 其中, $q_i = \begin{cases} p_i, & \text{if } gen_i \text{ is INACTIVE,} \\ c_i, & \text{o.w.,} \end{cases} 1 \leq i \leq m$; Emit $(\langle qid, \$, \# \rangle, count)$;

(4) 对于每个 $i \in [1, m]$, 用 q_i 在 qid 中的中亲 p_i 取代 q_i , 如果 $q_i = c_i$, 产生结果准标识符 qid^* 发出 $(\langle qid^*, p_i, \# \rangle, count)$

算法 3 的细节: 对于 ILPG 初始化, $NGSet$ 是关于 AL_0 的所有初始泛化集合, 对于 ILPG 更新, 根据算法 1 的第 6 行设置。算法 3 的第 6 行根据当前的匿名级, 将一个原始记录转换为其匿名形式。为了计算式(2)中信息损失估算 $|R_d|, |R_{d,sv}|, |R_b|$ 和 $|R_{b,sv}|$, 第 2 行向 reduce 函数发射键值对进行信息损失计算, 在执行其他泛化或插入一个新的泛化时, 泛化的信息损失将不会受到影响, 而隐私增益则由于数据集匿名化的改变而受到影响。

算法 3 的第 3 行的目的在于在执行一个泛化前, 找出数据集的匿名, 即 $A_b(gen)$, 第 4 行在执行一次泛化之后发射键值对来获得该匿名, 即 $A_d(gen)$ 。由于 $A_b(gen)$ 在整体中是唯一的, 本文只是反射了当下的准标识符 qid 进行统计。至于 $A_d(gen)$, 将发射 qid 的潜在匿名准标识符来计算不同活跃泛化候选的 $A_d(gen)$ 。在获得了 $A_d(gen)$ 和 $A_b(gen)$ 后, 可以根据式(3)更新每个泛化的隐私增益。

算法 4 ILPG 估算中的 reduce

输入 中间键值对 $(key, list(count))$

输出 信息增益 $(gen, IL(gen))$ 和匿名 $(gen, A_b(gen), AQISet)$, 泛化的 $(gen, A_d(gen))$

(1) 对于每个 $key, sum \leftarrow \sum count$;

(2) 对于每个 key , if $key.sv \neq \#$, 更新统计计算;

(3) $|R_{b,sv}| \leftarrow sum, |R_b| \leftarrow sum + |R_b|, |R_{d,sv}| \leftarrow sum + |R_{d,sv}|, |R_d| \leftarrow sum + |R_d|$;

(4) 如果子集 c 中的所有敏感值都到达, 那么计算 $I(R_d)$;

(5) 如果亲本 p 中的所有子集 c 都到达, 那么计算

$I(R_d)$ 和 $IL(gen)$, 发射 $(gen, IL(gen))$;

(6) 对于每个 key , if $key.sv \neq \#$, 更新匿名;

(7) 如果 $key.c = \$$, 且 $sum < A_d(gen)$ 那么更新当下匿名: $A_d(gen) \leftarrow sum$;

(8) 如果 $key.c \neq \$$;

(9) 如果 $sum < A_b(gen)$, 更新 gen 的潜在匿名: $A_b(gen) \leftarrow sum$ 和 $AQISet \leftarrow key.c$;

(10) 如果 $sum = A_b(gen)$, 那么更新: $AQISet \leftarrow \{key.c\} \cup AQISet$;

(11) 发射 $(gen, A_d(gen))$ 且发射 $(gen, A_b(gen), AQISet)$

算法 4 中描述的 reduce 函数主要聚合估算信息损失和隐私增益的统计信息。由于键值对在传输给 reducer workers 之前通过映射化简内置机制进行分类, 该 reduce 函数可以依次计算泛化的信息损失而不需要大量记忆来保存统计信息。因此, reduce 函数在计算信息损失上具有很强的伸缩性。

计算数据集匿名的本质特征是找出 QI -组的最小值。第二部分, 第 6 行和第 11 行, 目的在于计算隐私增益以及确定 $AQISet$ 。在并行执行泛化的前后, reducer workers 会找出局部最小 QI -组大小。通过比较 reducer workers 的输出可以获得驱动程序的整体大小。在处理和构建 $AQISet$ 中会记录具有最小组大小的 QI -组的准标识符。值得注意的是, $AQISet$ 在确定下一个循环的可用泛化中起到十分重要的作用。综上所述, ILPG 估算 reduce 函数在信息损失和隐私增益计算中都具有较高的伸缩性。在获得信息损失和隐私增益后就可以根据式(1)计算 ILPG 的值。

3.4 结合 TDS 和 BUG

本文将自下而上泛化的映射化简(MRBUG)和自上而下特化的映射化简(MRTDS)^[12]结合在一起, 在使用者指定匿名参数 k 后确定使用哪个组成来匿名化数据。混合方法可以自动给出系统参数 k , 如果 $k \geq K$, 那么选择 MRTDS, 否则选择 MRBUG。

一旦确定了工作负载平衡点 K , 混合方法就可以确定选择哪个组成。如果 $k > K$, 那么选择 MRTDS, 这是因为 MRBUG 会引发更多计算; 而如果 $k < K$, 则选择 MRBUG。算法 5 简略描述如下:

算法 5 混合算法

输入 数据集 D, k -匿名参数

输出 匿名数据集

(1) 如果 $k \geq K$, 那么用 MRTDS 匿名化 D ;

(2)否则,用 MRBUG 匿名化 D

找出 K 的精确值较为困难,这是由于 K 主要依赖于数据集的一些性能,如数据分配和偏态。然而并不需要获得精确值,这是因为当 k 在 K 附近取值时, MRTDS 和 MRBUG 的性能区别很小。

4 实验评估

4.1 实验设置

为了评估提出方法的效率,将其与参考文献[12]中的 MRTDS 和 MRBUG 进行比较,本文方法用 Hyb 表示,取混合英文单词 Hybrid 的前 3 个字母,3 种方法的执行时间分别表示为 T_{Hyb} 、 T_{TDS} 和 T_{BUG} 。当 $k > K$ 时, T_{Hyb} 和 T_{TDS} 相似;而当 $k < K$ 时, T_{Hyb} 和 T_{BUG} 相似,估计 K 会引发额外开支,但是这与映射化简任务中进行的计算相比非常微小。

实验在 U-Cloud 的云环境下进行, U-Cloud 是一种经常测试使用的云计算环境。图 1 显示了 U-Cloud 的系统概述,在硬件和 Linux 操作系统(Ubuntu)上安装了可以虚拟化的基础设施,并提供可以统一计算和存储资源的 KVM 虚拟化软件[16,17]。为了创建虚拟化数据中心,安装了负责虚拟化管理、资源调度、任务分配和与用户的交互的 OpenStack 开源云环境。

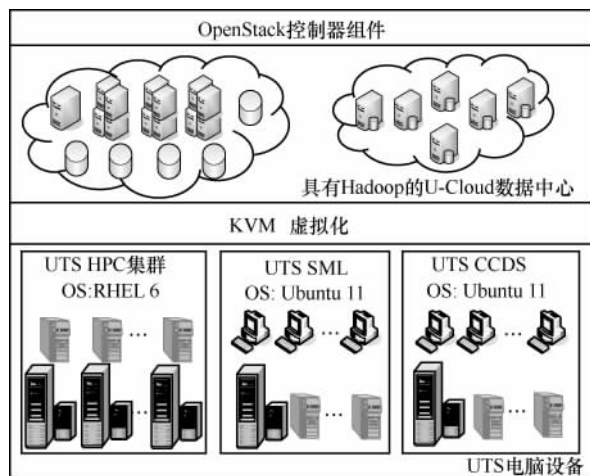


图 1 U-Cloud 的系统概述

所有方法都在 Java 和标准 Hadoop MapReduce API 中实现。Hadoop 集群包含 20 个具有 2 个虚拟 CPU 和 4 GB 内存的 m1 型媒介。每一轮实验重复 20 次,计算 20 次的均值作为测量结果。

4.2 实验结果与分析

本文测量了随着匿名参数 k 改变的执行时间 T_{Hyb} 、 T_{TDS}

和 T_{BUG} , 数据集的大小设置为 1 000 MB。该数据集包含 1.1×10^7 个数据记录,参数 α 为 0.5。

图 2 所示为 T_{Hyb} 、 T_{TDS} 和 T_{BUG} 随着 k 在 $0 \sim 1.1 \times 10^7$ 之间变化而发生的变化。为了表示简洁, k 用 exp 表示, 即 $k = 1.1 \times 10^{exp}$, 所有 exp 的取值范围为 $0 \sim 7$ 。从图 2 可以看出,混合算法的执行时间保持在一定水平之下,而 MRTDS 和 MRBUG 在 k 较小和较大时引发了较高的执行时间。此外, T_{Hyb} 曲线的左边部分十分靠近 T_{BUG} , 而右边部分十分靠近 T_{TDS} , 这是由于混合算法利用 MRTDS 和 MRBUG 作为具体计算的组件。因此,实验结果表明,本文的混合算法相较于现有方法,能够在不考虑 k -匿名参数的情况下较大地提高了子树匿名化的性能。

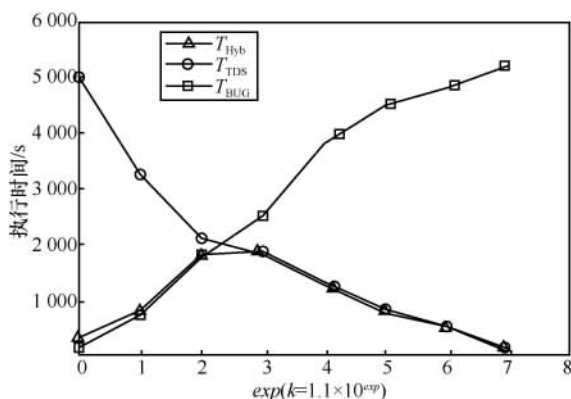


图 2 执行时间随着匿名参数 k 的改变

5 结束语

本文研究了云上大数据子树匿名化的伸缩性问题,并提出了一种结合了自上向下的特化(TDS)和自下而上的泛化(BUG)的混合方法。该混合方法通过比较用户指定的 k -匿名参数和负载平衡点自动选择两个组件中的一个。通过一系列特意设计的 MapReduce 任务,实验结果表明,与现有方法相比,本文方法极大地提高了子树数据匿名化的伸缩性和效率。

在云环境中,由于数据集体积越来越大,数据分析、分享和挖掘中的隐私保护是一项具有挑战性的研究课题,本文未来计划进一步探讨伸缩性的隐私保护感知分析和大规模数据集的调集。

参考文献:

- [1] 戚建国. 基于云计算的大数据安全隐私保护的研究 [D]. 北京: 北京邮电大学, 2015.

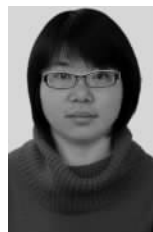


- QI J G. Research on security privacy protection of large data based on cloud computing [D]. Beijing: Beijing University of Posts and Telecommunications, 2015.
- [2] 康瑛石, 郑子军. 大数据整合机制与信息共享服务实现[J]. 电信科学, 2014, 30(12): 97-102.
- KANG Y S, ZHENG Z J. Big data integration mechanism and information sharing service realization [J]. Telecommunications Science, 2014, 30(12): 97-102.
- [3] 徐勇, 秦小麟, 杨一涛, 等. 一种考虑属性权重的隐私保护数据发布方法[J]. 计算机研究与发展, 2012, 49(5): 913-924.
- XU Y, QIN X L, YANG Y T, et al. A QI weighted-aware approach to privacy preserving publishing data set [J]. Journal of Computer Research and Development, 2012, 49 (5): 913-924.
- [4] MACHANAVAJJHALA A, GEHRKE J, KIFER D, et al. L-diversity: privacy beyond k -anonymity [J]. ACM Transactions on Knowledge Discovery from Data, 2007, 1(1): 24-32.
- [5] 李明庆. 一种基于抽样确定泛化区间的 K -匿名算法 [D]. 哈尔滨: 哈尔滨工程大学, 2013.
- LI M Q. A K -anonymity algorithm based on sampling to determine the generalized interval [D]. Harbin: Harbin Engineering University, 2013.
- [6] WANG K, FUNG B C M, YU P S. Handicapping attacker's confidence: an alternative to k -anonymization [J]. Knowledge & Information Systems, 2007, 11(3): 345-368.
- [7] TERROVITIS M, MAMOULIS N, LIAGOURIS J, et al. Privacy preservation by disassociation [J]. Proceedings of the Vldb Endowment, 2012, 5(10): 944-955.
- [8] 黄春梅, 费耀平, 李敏, 等. 基于多维泛化路径的 K -匿名算法[J]. 计算机工程, 2009, 35(2): 154-156.
- HUANG C M, FEI Y P, LI M, et al. K -anonymity algorithm based on multi-dimensional generalization path [J]. Computer Engineering, 2009, 35(2): 154-156.
- [9] FUNG B C M, WANG K, YU P S. Anonymizing classification data for privacy preservation [J]. IEEE Transactions on Knowledge & Data Engineering, 2007, 19(5): 711-725.
- [10] RAJALAKSHMI V, MALA G S A. Anonymization based on nested clustering for privacy preservation in data mining [J]. Indian Journal of Computer Science & Engineering, 2013, 34(3): 856-861.
- [11] MOHAMMED N, FUNG B C M, HUNG P C K, et al. Centralized and distributed anonymization for high-dimensional healthcare data [J]. ACM Transactions on Knowledge Discovery from Data, 2010, 4(4): 885-900.
- [12] ZHANG X, YANG L T, LIU C, et al. A scalable two-phase top-down specialization approach for data anonymization using MapReduce on cloud [J]. IEEE Transactions on Parallel & Distributed Systems, 2014, 25(2): 363-373.
- [13] 吕品, 钟路, 于文兵, 等. MA-Datafly: 一种支持多属性泛化的 k -匿名方法[J]. 计算机工程与应用, 2013, 49(4): 138-140.
- LV P, ZHONG L, YU W B, et al. MA-Datafly: k anonymity approaches for supporting multi-attribute generalization [J]. Computer Engineering and Applications, 2013, 49 (4): 138-140.
- [14] 刘盼盼. 基于聚类的数据匿名发布技术研究 [D]. 西安: 西安电子科技大学, 2013.
- LIU P P. Research on technology of data anonymous publishing based on clustering[D]. Xi'an: Xidian University, 2013.
- [15] CUI X, ZHU P, YANG X, et al. Optimized big data K -means clustering using map reduce [J]. Journal of Super Computing, 2014, 70(3): 1249-1259.
- [16] 王笑帝, 张云勇, 刘楠, 等. 云计算虚拟化安全技术研究[J]. 电信科学, 2015, 31(6): 1-5.
- WANG X D, ZHANG Y Y, LIU D, et al. Research on security of virtualization on cloud computing [J]. Telecommunications Science, 2015, 31(6): 1-5.
- [17] KVM[EB/OL]. (2011-04-10)[2012-02-10]. http://www.linux-kvm.org/page/Main_Page.

[作者简介]



范晓峰(1982-),女,内蒙古商贸职业学院讲师,主要研究方向为大数据、云计算。



闫凤(1982-),女,内蒙古农业大学职业技术学院讲师,主要研究方向为大数据。



刘洋(1975-),女,呼和浩特职业学院讲师,主要研究方向为大数据。