

基于垂直分割的个人数据隐私保护方法

阮华锋, 李睿, 罗凯伦

(东莞理工学院, 广东 东莞 523808)

摘要: 在分布式环境中, 垂直分割是一种保护用户隐私的有效方法。然而, 当前的垂直分割策略假设参与数据存储的各个云服务提供商 (cloud service provider, CSP) 之间不存在共谋。针对实际场景中CSP之间可能存在的共谋问题, 探讨了如何在这种情况下保护用户数据隐私。假设有 n 个CSP参与数据存储, 其中最多 k 个CSP可能会共谋, 给出了垂直分割的 (k, n) -安全定义, 并提出了MLVP (machine learning vertical partitioning) 方案。该方案利用机器学习算法分析属性之间的关联性, 对得到的所有关联性进行优化, 并将计算垂直分割方法问题转化成可满足性问题, 再利用可满足性问题求解器得到分割方法。此外, 对MLVP方案的安全性进行理论分析, 并在真实数据集上进行实验, 比较不同机器学习算法和隐私保护强度对分割效果和性能的影响; 与两个不考虑CSP存在共谋的垂直分割方案 (Oriol方案和Ciriani方案) 在计算速度和查询速度上进行了比较。实验结果表明: 在计算速度上, 因为要保证CSP共谋时的安全性, MLVP方案略慢, 在查询速度上, MLVP方案相较Oriol方案和Ciriani方案分别提升了32.6%和8.8%。

关键词: 垂直分割; 隐私保护; k -匿名模型; 机器学习; 可满足性问题

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2024076

Personal data privacy protection method based on vertical partitioning

RUAN Huafeng, LI Rui, LUO Kailun

Dongguan University of Technology, Dongguan 523808, China

Abstract: In distributed environments, vertical partitioning had been an effective method to protect user privacy. However, current vertical partitioning strategies assumed that there was no collusion among the CSPs (Cloud Service Providers) involved in data storage. This study explored how to protect user data privacy when collusion might exist between CSP. Assuming n CSP participated in data storage, with no more than k of these potentially colluding, this paper defined a (k, n) -security for vertical partitioning and introduced an automated computation scheme for vertical partitioning based on machine learning—the MLVP scheme. This MLVP scheme utilized machine learning algorithms to analyze the correlation between attributes, optimized all correlations, and transformed the vertical par-

收稿日期: 2024-01-29; 修回日期: 2024-05-15

基金项目: 国家重点研发计划 (2021YFB3101303); 国家自然科学基金 (61972089, 62206055)

Foundation Items: The National Key R & D Project of China (2021YFB3101303), The National Natural Science Foundation of China (61972089, 62206055)

引用格式: 阮华锋, 李睿, 罗凯伦. 基于垂直分割的个人数据隐私保护方法[J]. 网络与信息安全学报, 2024, 10(5): 175-187.

Citation Format: RUAN H F, LI R, LUO K L. Personal data privacy protection method based on vertical partitioning[J]. Chinese Journal of Network and Information Security, 2024, 10(5): 175-187.

tioning problem into a satisfiability problem, which was then solved using a satisfiability solver. Moreover, the security of the MLVP scheme was theoretically analyzed. To validate the effectiveness of the MLVP scheme, experiments were conducted on real datasets to compare the impact of different machine learning algorithms and levels of privacy protection on the effectiveness and performance of the vertical partitioning. The experiments also compared the MLVP scheme with two other schemes that did not consider collusion among CSP, Oriol's and Ciriani's schemes, in terms of computation and query speeds. The results showed that the MLVP scheme was slightly slower in computation speed to ensure security against partial CSP collusion. However, it improved the query speed by 32.6% and 8.8% compared to the aforementioned schemes, respectively.

Keywords: vertical partitioning, privacy protection, k -anonymity, machine learning, satisfiability problem

0 引言

随着云服务的普及, 以及云存储能提供便利、廉价的数据托管服务^[1], 众多企业开始倾向选择云服务提供商 (cloud service provider, CSP) 来构建其数据存储系统。然而, 数据一旦被存储在 CSP 中, 即意味着其控制权属于 CSP, 可能会产生数据被滥用以获取非法利益的风险^[2]。因此, 构建分布式存储系统时考虑数据泄露的风险很有必要。

在分布式环境中, 垂直分割是一种有效的用户隐私保护方法。数据分割问题可以表示为二维表的方式处理。在数据分割方法中, 常见的划分方式有水平分割^[3-4]、垂直分割^[5]和混合分割^[6]。水平分割数据按行分割数据, 通常适用于对数据值敏感的数据。然而, 在处理包含敏感属性的复杂数据时, 水平分割的隐私保护能力有限。混合分割结合了水平和垂直分割的特点, 虽然提高了数据处理的灵活性, 但显著增加了计算复杂性, 导致效率降低。相比之下, 垂直分割通过按列 (属性) 分割数据, 能够更精确地控制对敏感属性的访问。因此, 在多种数据分割策略中, 垂直分割适用于属性敏感的数据处理。

鉴于云存储服务中存在内部和外部攻击的风险, 用户不能完全信任第三方 CSP^[7]。在两个或多个 CSP 遭受攻击的情况下, 用户的敏感信息可能会被推断和泄露。例如, 表 1 展示了经过匿名处理的 Adult 数据集的一部分, 其中已删除了姓名属性以保护用户隐私。为了方便本文的讨论, 假设添加了姓名属性。在这个数据集中, 个人的薪资等级是一项敏感数据, 不公开个人薪资等级。假设将数据分为 {姓名, 年龄, 工作种类}、{性别, 婚姻状况, 教育程度}、{薪资等级} 3 个属

性集合, 并分别存储在 CSP1、CSP2 和 CSP3 中。当 CSP1 和 CSP2 中的数据被泄露, 根据年龄、教育程度两个属性的数据, 攻击者能推断存储在 CSP3 中个人薪资等级这一敏感数据。

表 1 Adult 数据集示例

Table 1 Example of adult dataset

姓名	性别	年龄	婚姻状况	教育程度	工作种类	薪资等级
张宏	男	40	已婚	12 年级	私人雇主	≤50 000
李四	男	49	已婚	硕士	私人雇主	>50 000
李明	男	44	离婚	11 年级	自我雇佣	>50 000
王婷	女	21	未婚	9 年级	自我雇佣	≤50 000
陈凯	女	31	已婚	11 年级	政府工作	≤50 000

本文考虑一个场景, 有 n 个 CSP 共同负责数据的存储, 其中 k 个 CSP 可能会共谋泄露信息 ($1 \leq k < n$)。针对此情形, 本文设计一种数据分割方案, 以确保用户隐私得到保护, 并引入了属性的 (k, n) -安全分割定义, 用于判断所给出的分割方法是否安全。为实现符合 (k, n) -安全定义的垂直分割方法, 本文提出了基于机器学习和可满足性问题的 MLVP (machine learning vertical partitioning) 方案。该方案包含 3 个步骤。①采用机器学习算法 L 进行递归式的分析, 识别所有可能泄露隐私的非敏感属性集。②对初步得到的属性集进行优化精简, 去除不必要的推断集, 以提高分割方案计算的效率。③将计算垂直分割问题转化成可满足性问题 (satisfiability problem, SAT), 并用 SAT 求解器进行求解, 得到最终的垂直分割方法。

本文通过实验, 比较不同的机器学习算法和隐私保护强度对 MLVP 方案的性能影响, 并将其

与现有垂直分割方案的计算速度和查询速度进行比较。实验结果表明：在测试的8种机器学习算法中，有4种产生了相同的结果，岭回归算法在这些算法中计算速度最快。本文还发现，在一定范围内增大隐私保护强度能提升隐私保护的效果。MLVP方案能够在最多 k 个CSP共谋的情况下保护用户隐私，相比不考虑共谋问题的Oriol方案^[5]和Ciriani方案^[8]，其在计算速度上虽有劣势，但在数据查询速度上占有明显优势。具体来说，与Oriol方案和Ciriani方案相比，MLVP方案的查询速度分别提高了32.6%和8.8%。

1 相关工作

1.1 k -匿名模型

k -匿名模型^[9]为公开发布的数据提供了一种隐私保护机制。在数据集中，隐私泄露往往并非源自敏感数据的直接泄露，而是敏感数据与用户的联系被发现导致的。为此， k -匿名模型引入了准标识符属性这一概念。这类属性本身不能确定单个数据项的身份，但与外源数据结合时，便可能识别出数据项身份。该模型将准标识符属性匿名处理，使数据中存在 k 项相同准标识符属性的值，从而混淆攻击者识别个人身份的可能性。

1.2 可满足性问题

在计算复杂性领域，SAT问题被认为是首个被证明的NP-完全问题。由于这一复杂组合问题在多个领域拥有广泛的应用，并且各种问题都能被归约至SAT问题，因此它一直吸引着研究者的关注^[10]。从理论的角度探讨，多数的组合问题能被转化为命题逻辑形式的条件约束表达式，并通过运用SAT求解器得到相应解决。

1.3 垂直数据分割

现有的垂直分割方法研究通常基于这样一个假设：不同的CSP之间不会发生共谋行为。然而，在实际应用场景中，CSP间的共谋是可能发生的。因此，为了防止CSP共谋导致的隐私泄露，数据所有者必须采用可信赖的服务器或加密技术来保护数据^[11]。

Aggarwal等^[12]提出先将数据集垂直分割为两部分，并对其中一部分进行加密，然后将它们分别存储在两台独立服务器上，以此来增强

数据的安全性。Yang等^[13]提出将医疗数据集垂直分割成3部分，其中包含识别患者信息的部分以密文形式存储，其余数据则以明文保存。需要指出的是，这些方案仅适用于数量有限的服务器。

Ganapathy等^[14]在Aggarwal等^[12]的基础上进行了进一步的研究，提出了一种基于贪婪爬山算法的启发式搜索方案，该方案能够将数据垂直分割到任意数量的服务器上。Ciriani等^[15]则引入了一种数据外包隐私保护范式，允许数据所有者保留部分数据，并将剩余数据存储在外部服务器上。此外，Sharad等^[16]专注于垂直分割方案中的查询安全性问题，提出了“查询分组”的安全方案，该方案确保在联合查询敏感与非敏感数据时不泄露任何敏感信息。Oriol等^[5]则将垂直分割任务建模为一个组合优化问题，并采用组合优化技术来解决分割问题。然而，这些方案的共同缺点是它们依赖于额外的可信服务器，这可能导致网络查询的延迟增加以及成本上升。

Ciriani等^[8]展示了一种结合垂直分割和加密技术以确保数据隐私的解决方案，并提出了两种启发式方法来得到分割策略。Sudha等^[17]基于垂直分割和数据匿名化技术，为银行数据设计了一种防欺诈交易的存储方案。此外，Denis等^[18]针对物联网环境中的发布/订阅模式，提出了一种结合属性加密和数据分割的隐私保护方案。然而，为了防止服务器之间的共谋导致隐私泄露，所需的加密操作不可避免地会增加计算成本。

2 定义及概念

本文考虑结构化数据，设一个结构化数据集 $\mathcal{D}$ ，其属性集 $\mathcal{A}=\{A_1, A_2, \dots, A_m\}$ 中包含 m 个属性，其中 $A_i (1 \leq i \leq m)$ 表示一个具体的属性。为了实现数据的有效分割，本文定义一个属性分割方案 $\mathcal{F}=\{F_1, F_2, \dots, F_n\}$ ，其中每个 $F_i (1 \leq i \leq n)$ 可能是一个单独的属性或一个属性的集合，并且满足以下条件：①所有分割的并集 $F_1 \cup F_2 \cup \dots \cup F_n = \mathcal{A}$ 等于整个属性集 $\mathcal{A}$ ；②任意两个不同的分割 F_i 和 F_j 满足 $F_i \cap F_j = \emptyset (i \neq j, 1 \leq i, j \leq n)$ ，保证每个属性只被分配到一个子集中。

进一步定义 $\mathcal{F}$ 对应的数据集为 $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$, 其中每个 $D_i (1 \leq i \leq n)$ 表示包含属性 F_i 值的所有数据项。通过映射函数 $\mathcal{D}(F_i) = D_i$, 可以从属性集 F_i 中提取相应的数据集 D_i 。举例来说, 假设存在数据集 $\mathcal{D} = \{d_1, d_2, \dots, d_m\}$, 每个数据项 d_i 携带属性 A_i 的值, 且给定属性分割 $\mathcal{F} = \{F_1, F_2\}$, 其中 $F_1 = \{A_1, \dots, A_g\}$ 包括一组属性。于是, 可以得出两个子数据集 D_1 和 D_2 , 这里 $D_1 = \{d_1, \dots, d_g\}$ 包括与 F_1 相对应的全部数据项, 而 $D_2 = \{d_{g+1}, \dots, d_m\}$ 则涵盖与 F_2 相关的所有数据项。

基于数据集中的信息敏感性, 将数据集属性分为敏感属性集和非敏感属性集。敏感属性集用 S 表示, D_S 表示 D 中具有 S 属性的值的全部数据, 借助函数 $\mathcal{D}(S) = D_S$, 可以轻松地从整个数据集中筛选出含有敏感属性的数据项。以表 1 中的数据为例, 假设敏感属性集 $S = \{\text{薪资等级}\}$, 其对应的敏感数据集 $D_S = \{\leq 50\ 000, > 50\ 000, > 50\ 000, \leq 50\ 000, \leq 50\ 000\}$ 。

为了增强垂直分割的效能, 引入 L -推断集定义。因为在匿名数据中, 攻击者仍然能用人工智能方法侵犯用户隐私, 所以本文利用机器学习方法推断属性间关系, 并试图切断这种关系。

定义 1 L -推断集。设 $\mathcal{D}$ 是数据全集, $D_i (D_i \subset \mathcal{D})$ 是一个数据集, $D_S (D_S \subset \mathcal{D})$ 是敏感数据集, 且 L 为一个机器学习算法。给定推断率阈值 c , 如果存在一个模型 M , 满足 $M = \text{Train}(L, D_i, D_S)$, 并且 $\text{Acc}(M, D_S | D_i) > c$, 其中 $\text{Acc}(M, D_S | D_i)$ 表示模型 M 利用 D_i 数据集得到 D_S 数据集的准确率, 则称 D_i 对应的属性集 F_i 是数据集的一个 L -推断集。

以表 1 为例, 假设攻击者获取了一家公司泄露了年龄、教育程度和薪资等级属性的数据。因为数据表中的姓名未泄露, 因此攻击者不能直接获取到李明的薪资等级。但是攻击者利用机器学习算法 L , 构建了一个预测准确率为 0.7 的模型, 可以输入个体的年龄、教育程度预测其薪资等级。当攻击者能够获取李明的年龄和教育程度等信息时, 能以 70% 的概率推测出李明的薪资等级信息。

在 L -推断集的基础上, 继续考虑在 CSP 共

谋的情况下保护用户隐私。在文献[19]中, 考虑如何利用秘密共享方案处理数据并且划分为 n 份, 使任何大于 k 份的子集都能重构出原有的数据。本文受到秘密共享方案的启发, 考虑在小于等于 k 个 CSP 共谋的情况下, 如何利用垂直分割方案保护用户隐私安全的问题。为了解决这种问题, 本文给出了属性的 (k, n) -安全分割定义。该定义保证在至多 k 个 CSP 的数据泄露时, L -推断集不遭到泄露, 因此攻击者无法训练得到准确率大于 c 的模型, 从而推测用户的敏感属性。

定义 2 属性的 (k, n) -安全分割定义。设 $\mathcal{D}$ 是数据全集, 有敏感属性集 S 对应的数据集 D_S 。设 $\mathcal{D}$ 的一个垂直分割方案 $\mathcal{F}$, 所有 L -推断集的集合 $\mathcal{P} = \{P_1, P_2, \dots, P_l\}$, 其中 P_i 是 $\mathcal{D}$ 的一个 L -推断集, 给定任意一个泄露的子分割属性集 $H \subset \mathcal{F} (1 \leq |H| \leq k)$, 并且 $k < n$, 使 $(S \cup \forall P_i) \not\subset \cup H$, 则称这种垂直分割方法是 (k, n) -安全的。

考虑表 1 中的数据作为 (k, n) -安全分割定义的示例。设定 $k=2, n=3$, 这意味着希望在至多两个 CSP 共谋的情形下, 通过垂直分割技术保护用户隐私。在此基础上, 假设算法识别的 L -推断集的集合 $\mathcal{P} = \{\{\text{年龄, 教育程度}\}\}$ 。此时, 一个符合 (k, n) -安全定义的垂直分割方案 $\mathcal{F}$ 可以被定义为 $\mathcal{F} = \{\{\text{性别, 薪资等级}\}, \{\text{年龄, 婚姻状况}\}, \{\text{教育程度, 工作种类}\}\}$ 。因为在任意两个子分割属性集合的组合中, 均不包含足够的信息训练出准确率大于特定阈值 c 的模型以推断用户的敏感属性。因此, 即使在最多两个 CSP 共谋的情况下, 攻击者也无法从泄露的数据中恢复出完整的 L -推断集, 从而确保敏感信息的安全。

3 MLVP 方案设计

本文提出的垂直分割计算方案包括 3 个步骤: 计算所有 L -推断集、优化 L -推断集的集合、计算垂直分割方法。MLVP 方案流程如图 1 所示, 流程如下: 首先, 数据集 $\mathcal{D}$ 、隐私属性集 S 、机器学习算法集和推断率阈值 c 被输入计算所有 L -推断集步骤, 选择算法集合中产生的最有效推断算法, 生成数据集 $\mathcal{D}$ 的 L -推断集的集合 $\mathcal{P}$; 然

后, $\mathcal{P}$ 经过优化 L -推断集的集合步骤, 对多余的 L -推断集精简; 最后, 将优化后的 L -推断集的集合与敏感数据集 S 、服务器个数 n 共同输入计算垂直分割方法步骤中, 得到符合 (k, n) -安全定义的垂直分割方法。

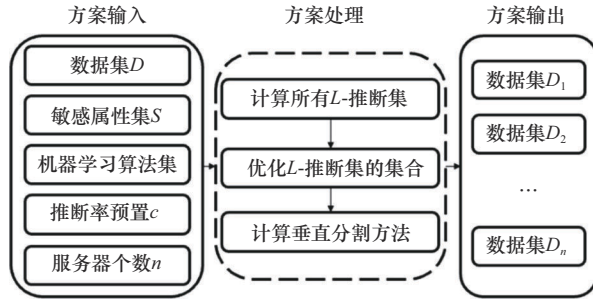


图1 MLVP方案流程

Figure 1 Flow of MLVP scheme

3.1 计算所有 L -推断集

根据定义2, 本文提出一种计算所有 L -推断集的方案: 输入数据集 $\mathcal{D}$ 的所有子集, 直接识别出所有的 L -推断集。然而, 在实际应用场景中, 通过这种方法得到的 L -推断集的集合并未全面反映攻击者可能获得的用户隐私。因为攻击者可能会推断出中间数据集, 然后利用中间数据集进一步推断出敏感数据集。在这种场景下存在属性值递推的可能性。因此, 这种方案在安全性层面上存在不足。

以表1中的属性为例, 假定一个属性分割数据集 $H = \{\{\text{年龄}, \text{教育程度}\}, \{\text{婚姻状况}, \text{工作种类}, \text{薪资等级}\}\}$ 被泄露, 泄露的属性如图2中的所有圆角矩形所示。设置推断率阈值 $c=0.7$ 。攻击者知晓李明的3个属性的值(年龄、教育程度、工作种类), 在图2(递推例子图示)中用红色圆角矩形标出。攻击者利用泄露的数据训练出一个从 $P_1 = \{\text{婚姻状况}, \text{工作种类}\}$ 推断出 $S = \{\text{薪资等级}\}$ 的模型 M_1 , 准确率为0.8。但攻击者不知道李明的婚姻状况。于是, 攻击者进一步训练了一个从 $P_2 = \{\text{年龄}, \text{教育程度}\}$ 推断出婚姻状况的模型 M_2 , 准确率为0.9。因此, 攻击者可以利用模型 M_2 推断李明的婚姻状况, 利用模型 M_1 推断李明的薪资等级。整体推断的准确率为 $0.9 \times 0.8 = 0.72$, 这一概率超过了设定的推断率阈值 c 。

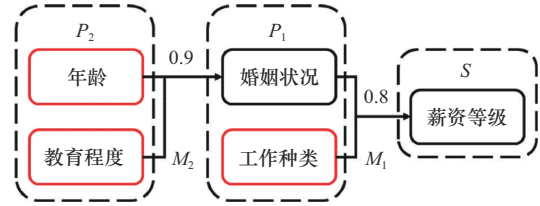


图2 递推例子

Figure 2 Recursion example

为了避免属性值递推的情况发生, 本文引入一种独特的数据结构—— L -推断集生成树。在这种树形结构中, 根节点代表最初的敏感属性集 S 。每一个子节点是以前节点作为敏感属性集时, 得到的 L -推断集。在得到 L -推断集生成树之后, 数据所有者可以根据具体情况, 选择整棵树或一部分子树作为数据集的 L -推断集, 进行后续的垂直分割计算。 L -推断集生成树的引入强化了安全性, 提供了一种更为精细和优化的隐私保护方法, 推断集生成算法如算法1所示。

算法1 推断集生成算法: $\text{GetInferences}(T, \mathcal{D}, \mathcal{A}, H, L, c)$

输入 空树 T , 数据集 $\mathcal{D}$, 属性集 $\mathcal{A}$, 属性集 H , 其中 $H \subset \mathcal{A}$, 机器学习算法 L , 隐私保护阈值 c ;

输出 推断集生成树 T

- 1) **if** T 是空树
- 2) $T \leftarrow$ 初始化为根节点为 S 的树
- 3) **end if**
- 4) $Q \leftarrow$ 空队列
- 5) $A^* \leftarrow \text{PowerSet}(\mathcal{A} - S)$;
- 6) **for each** $F_i \in A^*$
- 7) $D_i \leftarrow \mathcal{D}(F_i)$; //得到具有属性集 F_i 属性值的全部数据
- 8) $D_H \leftarrow \mathcal{D}(H)$; //得到具有属性集 H 属性值的全部数据
- 9) $M \leftarrow \text{Train}(L, D_i, D_H)$;
- 10) **if** $F_i \notin T$ and $\text{Acc}(M, D_S | D_i) > c$
- 11) $T.\text{InsertChild}(H, F_i)$;
- 12) $Q.\text{Enqueue}(F_i)$;
- 13) **end if**
- 14) **end for**
- 15) **for each** $F_i \in Q$
- 16) $\text{GetInferences}(T, \mathcal{D}, \mathcal{A}, F_i, L, c)$;

17) **end for**

18) **Return** T

算法步骤如下。首先, 遍历属性集除去敏感属性集后的所有子集。对每个子集 F_i , 如果同时满足以下两个条件: ① F_i 在推断集生成树 T 中不存在; ② F_i 是以 H 为敏感属性集的 L -推断集。那么将 F_i 作为 H 的子节点插入树 T 中。然后, 对每个新生成的 L -推断集 F_i , 视作敏感属性集再次调用算法进行递归计算, 直到不再产生新的 L -推断集为止。因为算法不会添加重复的集合, 所以本轮迭代的敏感属性集不需要包括上一轮的敏感属性集。最后, 构造出一棵 L -推断集生成树并返回。因为属性递推的可能性逐层降低, 所以在构造 L -推断集生成树时, 采用广度优先的策略。

推断集生成算法能避免攻击者通过递推的方式侵犯用户隐私。在上述的属性值递推例子中采用该算法后, 得知能推断出婚姻状况的属性集 $P_2 = \{\text{年龄}, \text{教育程度}\}$ 在 L -推断集生成树中存在。因此, P_2 将被视为 L -推断集输入计算垂直分割方法中。因此攻击者不能获取 P_2 关联的数据集, 也无法推断李明的薪资等级。

在现实世界的应用中, 数据拥有者通常无法准确预测攻击者会选择哪一种机器学习算法来进行数据分析和推断。因此, 提出针对一种特定的机器学习算法的推断集生成算法是不足的。在 MLVP 方案中, 要求数据拥有者提前定义一个可能被攻击者使用的机器学习算法集合。在这个集合中, MLVP 方案会计算出所有可能的 L -推断集, 并从中选择那些在给定算法集合上推断率最高的算法产生的推断集作为最终的 L -推断集。通过这种方法, 即使在数据拥有者不知道确切的攻击算法的情况下, MLVP 方案也能提供一种针对预测的机器学习算法集合中最有效的推断保护方法。这种方法增强了本文方案的适用性和安全性, 确保了即便在不同的攻击情景下, 用户的隐私仍能得到有效保护。

3.2 优化 L -推断集的集合

为了加快计算垂直分割方法的速度, 根据属性的 (k, n) -安全分割定义对生成的 L -推断集的集合 $\mathcal{P}$ 优化。

定理 1 考虑 L -推断集集合 P_i 和 P_j , 其中

$P_i \subset P_j$ 。定义所有满足 (k, n) -安全定义的集合为 A, B , 可以推测出 $A \subset B$, 即所有能够保护 P_i 不被泄露的分割方法也必然能够保护 P_j 不被泄露。

证明 在考虑两个 L -推断集 P_i 和 P_j 的情况下, 其中 $P_i \subset P_j$, 定义 A 和 B 分别为针对 P_i 和 P_j 的所有满足 (k, n) -安全定义的分割方法集合。由于 P_i 是 P_j 的子集, 根据分割方法的性质, 可以推导出 $A \subset B$, 这意味着所有能够保护 P_i 不被泄露的分割方法同样能保护 P_j 。然而, (k, n) -安全定义要求分割方法必须防止所有 L -推断集的泄露。如果某一分割方法能防止 P_j 的泄露却不能防止 P_i 的泄露, 则此方法将不会包含在集合 A 中, 而仅存在于 B , 即 $B - A$ 的部分。但这种情况与 (k, n) -安全定义的基本要求相矛盾, 因为 P_i 已被包含在 P_j 中, 故任何不能保护 P_i 的方法也无法保护 P_j 。这表明在 $B - A$ 中不存在任何有效的分割方法, 因而所有只能保护 P_j 而不能保护 P_i 的分割方法都不满足 (k, n) -安全定义。由此可见, P_j 相对 P_i 是冗余的, 其存在并不增加分割方法的有效性。因此, P_j 可以从 L -推断集集合 $\mathcal{P}$ 中移除, 而不影响垂直分割方案的 (k, n) -安全性。

以表 1 中的数据举例。假设敏感属性 $S = \{\text{薪资等级}\}$, $\mathcal{P} = \{P_1, P_2\}$, $P_1 = \{\text{教育程度}\}$, $P_2 = \{\text{教育程度}, \text{年龄}\}$, $k=1$ 。那么符合 P_1 的垂直分割方法必须要求教育程度、薪资等级不能存放在同一个服务器中, 而符合 P_2 的垂直分割方法要求教育程度、薪资等级和年龄不能存放在同一个服务器中。所以所有满足 P_1 的分割方法的集合是所有满足 P_2 的分割方法的集合的子集, 因此 P_2 可以从 $\mathcal{P}$ 中删除。

3.3 计算垂直分割方法

本文把计算垂直分割的问题转化为可满足性问题, 利用 SAT 求解器计算分割方法。

定理 2 给定属性的 (k, n) -安全分割定义, 设 L -推断集集合 $\mathcal{P}$ 中属性数量最少的集合为 $P_{\min}$, 则允许泄露的服务器数量 k 的取值范围应满足 $0 < k < \min(|P_{\min}| + |S|, n)$ 。

证明 根据基于属性的 (k, n) -安全分割定义, 数据集的单个属性被视为分割的最小单元。此定义的核心前提是, 当泄露的服务器数量 k 达到或超过 $|P_{\min}| + |S|$ 时, 意味着存在一个服务器集

合，其属性可以完全覆盖 $S \cup P_{\min}$ ，从而有可能推断出敏感信息。这种情况违反了 (k, n) -安全定义的基本宗旨。因此，可以推导出 k 的最大值应为 $k_{\max} = |P_{\min}| + |S| - 1$ 。此外， (k, n) -安全定义还要求 k 值必须小于服务器的总数 n ，进而形成了 $k < n$ 的额外约束。综合这两个条件，得出 k 的合理取值范围应该是 $0 < k < \min(|P_{\min}| + |S|, n)$ 。这一取值范围确保即使服务器发生数据泄露，敏感信息仍能得到有效的保护。

本文设定服务器的总数为 n ，该值由数据拥有者根据用户实际需求进行确定，并且引入符号 $V_{A_i}E_j$ 来表示属性 A_i 存储在服务器 E_j 中，而 $\neg V_{A_i}E_j$ 则代表属性 A_i 不存储在服务器 E_j 中。在后续部分，将阐述如何将计算分割方法问题转化为约束表达式并用 SAT 求解器计算分割方法。

3.3.1 确保属性的准确分配

确保属性分配的正确性是实施垂直分割方法过程中的首要任务。这要求每个属性必须且只能分配到一个服务器。假设属性集 $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$ 与服务器集 $\mathcal{E} = \{E_1, E_2, \dots, E_n\}$ ，其中 $A_i (1 \leq i \leq m)$ 代表一个属性，而 $E_j (1 \leq j \leq n)$ 代表一个服务器。为了描述属性 A_i 恰当地存储在服务器 E_j 上的情况，定义约束条件 $r_{A_i}E_j = \neg V_{A_i}E_1 \wedge \neg V_{A_i}E_2 \wedge \dots \wedge V_{A_i}E_j \wedge \dots \wedge V_{A_i}E_n$ ，其中 $V_{A_i}E_j$ 表示属性 A_i 存储在服务器 E_j 上。对于每个属性 A_i ，都存在一个约束条件 $r_{A_i} = r_{A_i}E_1 \vee r_{A_i}E_2 \vee \dots \vee r_{A_i}E_n$ ，保证 A_i 至少分配到一个服务器。最终，通过构建全体属性的分配约束 $R_1 = r_{A_1} \wedge r_{A_2} \wedge \dots \wedge r_{A_m}$ ，确保所有属性的正确分配。

例如，考虑表1中的数据，其中需确保除姓名以外的属性分配至3个服务器 $\mathcal{E} = \{E_1, E_2, E_3\}$ ，同时设定 $\mathcal{A} = \{A_1, A_2, A_3, A_4, A_5, A_6\} = \{\text{性别, 年龄, 婚姻状况, 教育程度, 工作种类, 薪资等级}\}$ 。首先，为性别属性 A_1 构建其存储于服务器 E_1 的条件约束式 $r_{A_1}E_1 = V_{A_1}E_1 \wedge \neg V_{A_1}E_2 \wedge \neg V_{A_1}E_3$ 。同理，构建性别属性存储于 E_2 和 E_3 的条件约束式 $r_{A_1}E_2$ 和 $r_{A_1}E_3$ 。接着，将这些约束式结合起来，形成性别属性的正确分配条件 $r_{A_1} = r_{A_1}E_1 \vee r_{A_1}E_2 \vee r_{A_1}E_3$ 。以此类推，对所有属性进行同样的操作，

最终得到整体的约束条件 $R_1 = r_{A_1} \wedge r_{A_2} \wedge r_{A_3} \wedge r_{A_4} \wedge r_{A_5} \wedge r_{A_6}$ ，以确保所有属性都被正确地分配。

3.3.2 确保分割方法不违反 (k, n) -安全定义

本小节构建条件约束式，以确保所计算的垂直分割方案满足 (k, n) -安全定义。设 L -推断集的集合为 $\mathcal{P} = \{P_1, P_2, \dots, P_l\}$ ，其中 $P_i (1 \leq i \leq l)$ 表示一个推断集。推断集与敏感属性集 S 的并集构成了新的属性集合 $\mathcal{I} = \{I_1, I_2, \dots, I_l\}$ ，且每个 $I_i = \{a_1, a_2, \dots, a_o\}$ 是由 L -推断集 P_i 与 S 合并而成的属性集 $(1 \leq i \leq o)$ ，其中 $a_x \in I_i (1 \leq x \leq o)$ 代表 I_i 中的属性。对 n 台服务器，考虑所有从中选出 k 台服务器的可能组合 $\mathcal{C} = \{C_1, C_2, \dots, C_q\}$ ， $C_j (1 \leq j \leq q)$ 是这些组合中的一个。对于属性集 I_i 存储在服务器组合 C_j 中的每种可能情况，构建约束 $r_{I_i}C_j = V_{a_1}e_1 \wedge V_{a_2}e_2 \wedge \dots \wedge V_{a_o}e_o$ ，其中 $\{e_1, e_2, \dots, e_o\} \subseteq C_j$ 代表 C_j 中选出的服务器。之后，对所有服务器组合中的这种存储情况进行否定，以构建属性集 I_i 不被泄露的约束式 $r_{I_i} = \neg r_{I_i}C_1 \wedge \neg r_{I_i}C_2 \wedge \dots \wedge \neg r_{I_i}C_q$ 。最终，通过合并所有的 r_{I_i} ，得到满足 (k, n) -安全定义的总约束 $R_2 = r_{I_1} \wedge r_{I_2} \wedge \dots \wedge r_{I_l}$ ，确保所有可能的垂直分割方案都不会违反 (k, n) -安全性。

以表1中的数据为例，假设数据集 D 的 L -推断集的集合 $\mathcal{P} = \{P_1\} = \{A_1, A_2\} = \{\{\text{年龄, 教育程度}\}\}$ ， $S = \{A_3\} = \{\text{薪资等级}\}$ ，那么 $\mathcal{I} = \{I_1\} = \{\{A_1, A_2, A_3\}\}$ ，服务器 $\mathcal{E} = \{E_1, E_2, E_3\}$ ，并且 $k=2$ 。首先确定所有可能泄露的服务器组合 $\mathcal{C} = \{C_1, C_2, C_3\} = \{\{E_1, E_2\}, \{E_1, E_3\}, \{E_2, E_3\}\}$ 。然后对组合 C_1 构建违反 (k, n) -安全定义的约束式 $r_{I_1}C_1 = V_{A_1}e_1 \wedge V_{A_2}e_2 \wedge V_{A_3}e_3$ ， $e_1, e_2, e_3 \in \{E_1, E_2\}$ 。约束式 $r_{I_1}C_1$ 代表在服务器组合 C_1 泄露时，隐私泄露的所有情况。再对剩下的服务器组合 C_2, C_3 用相同的方法构建约束式 $r_{I_1}C_2, r_{I_1}C_3$ 。将得到的约束式全部取反后用且符号连接，得到垂直分割约束 $r_{I_1} = \neg r_{I_1}C_1 \wedge \neg r_{I_1}C_2 \wedge \neg r_{I_1}C_3$ 。因为只有一个 L -推断集，所以可以直接得到最终的约束条件式 $R_2 = r_{I_1}$ 。

3.3.3 用户自定义约束

为了响应数据所有者对隐私保护的关注，并满足数据存储与访问的其他要求，本文所提的

MLVP 方案提供了自定义约束功能。例如, 某些属性可能因数据完整性或安全性的考虑而不宜存储在同一服务器上, 需采用特定的存储策略以提高联合查询的效率。这类由数据所有者定义的约束被标记为 R_3 。由于采用布尔条件表达式来处理属性分割问题, 用户可以将自定义的约束以相同的形式输入 MLVP 方案中。该方案随后会将这些自定义约束与其他约束整合, 确保最终的属性分割方案能够满足用户的特定需求。

完成所有约束表达式的整合后, 将其输入 SAT 求解器, 从而生成满足 (k, n) -安全标准的垂直分割方法 $\mathcal{F}$ 。在这个过程中, 约束式 R_1 确保属性被正确分配; 约束式 R_2 确保得到的垂直分割方法结果符合 (k, n) -安全定义; 而约束 R_3 确保结果的灵活性。这种综合约束表达式的形式为

$$\mathcal{R} = R_1 \wedge R_2 \wedge R_3$$

3.4 安全性分析

3.4.1 针对共谋攻击分析

MLVP 方案在缓解 CSP 间共谋攻击方面取得了显著进展。首先, 借助 (k, n) -安全定义, 为属性间的垂直分割提供了清晰的指导原则。这确保即使在最多 k 个服务器数据泄露的情况下, L -推断集仍能保持安全不被泄露。其次, 引入 L -推断集生成树的概念, 确保仅特定 L -推断集能被用于隐私保护计算, 有效预防属性值递推漏洞, 进一步提升了隐私保护操作的精确性和效率。最后, 将 (k, n) -安全所需的垂直分割问题转化为一个可满足性问题, 并通过求解该问题获得保障隐私安全的垂直分割方案。综上所述, 本文所提方案能有效抵御部分 CSP 之间的共谋攻击。

3.4.2 针对属性间关系的攻击

攻击者可能会针对属性间的关系进行挖掘分析。当外泄的数据被用作构建机器学习模型的输入时, 攻击者有可能通过用户的非敏感属性值推断出相关的敏感属性集。本文给出的 (k, n) -安全定义旨在这种情况下对用户隐私保护, 确保攻击者无法训练出一个推断准确率超过 c 的推断模型。

在极端情况中, 攻击者可能只需要再掌握一个属性的所有值, 用户隐私就会遭到泄露。

假设这个属性采用 v 位存储空间, 数据集存在 g 条数据记录。攻击者尝试猜测这一属性的值时, 针对属性间关系的攻击导致隐私泄露的概率为 $c \times 1/(g \times 2^v)$ 。

在真实数据集中, 隐私泄露的概率非常小。在本文所用的 Adult 数据集中, 包含约 48 000 条数据。假设推断率阈值 $c=0.8$, 缺失的属性值存储位数 $v=8$, 则隐私泄露概率是 6.51×10^{-8} 。这表明本文所给出的安全定义能显著遏制用户隐私被潜在泄露。

3.4.3 针对匿名模型的攻击分析

当泄露的数据包括匿名模型需要隐藏的信息时, 隐私安全将得到匿名模型的保护。以表 1 数据为例, 假设该数据经过 k -匿名处理, 其中标识符属性是性别、年龄, 敏感属性是薪资等级, 并假设这些属性对应的数据已经泄露。在此场景下, 攻击者可能会尝试将性别、年龄和已知的用户非敏感属性链接, 确定用户的身份。归功于 k -匿名模型的保护, 攻击者最终可能得到 k 个混淆用户身份。

数据拥有者可以自行输入与匿名模型相关的属性约束, 规避针对匿名模型的隐私攻击。在上面的例子中, 用户只需要将 {性别, 年龄, 薪资等级} 这个属性集作为自定义约束输入, 即可完全避免与外源数据链接导致泄露隐私的情况。

4 实验

4.1 实验方法

影响 MLVP 方案结果的关键因素包括所使用的机器学习算法及其推断率阈值。为评估垂直分割方案的有效性, 本文主要考察了方案的计算时间和查询时间两项指标。

(1) 实验数据集

本文实验在 Adult 数据集^[20]上进行。该数据集来源于 1994 年美国的人口普查数据库, 从中选取了 9 个属性作为实验的对象 (性别、年龄、种族、婚姻状况、教育程度、祖国、工作种类、职业、薪资等级), 并且设定敏感属性集 $S=\{\text{薪资等级}\}$ 。实验采用的数据集包括训练集和测试集部分, 共包含 48 843 条记录。

(2) MLVP 方案的实现细节

基于第 3 节所描述的 MLVP 方案, 本文采用

python 编程语言结合 scikit-learn 库和 SymPy 库进行实现。为稳定机器学习算法差异带来的实验效果变化,构建了一个统一的框架,包括数据预处理、机器学习模型构建、模型效果评估3大模块。数据预处理阶段,将所有原始数据转化为数值型数据,以适应机器学习模型的需求;模型构建阶段,采用默认参数进行训练;模型效果评估阶段,采用 R2 指标。

(3) 实验环境设置

本文方案实验在英特尔 i7-9700, 16 GB 运行内存的 Windows10 计算机上进行。数据查询速度实验在腾讯云轻量应用服务器 1 核 CPU 版本上进行,服务器运行环境为 Ubuntu20。

(4) 机器学习算法对 MLVP 方案的影响实验

在现实场景中,数据拥有者无法得知攻击者采取的机器学习算法,所以本文将机器学习算法作为一个输入参数,试图理解不同算法可能带来的分割结果差异。本文旨在通过比较各种算法的结果,以便挑选一个相对优秀的算法。此次实验输入的机器学习算法集合包含 8 种常用的算法(线性回归、岭回归、Lasso、贝叶斯岭回归、偏最小二乘回归、 k 近邻回归、决策树以及支持向量机),对比不同算法对 MLVP 方案结果的影响。

(5) MLVP 方案性能评估实验

推断率阈值 c 是一个关键参数,它决定了哪些属性集合被认定为 L -推断集,并且能显著影响方案的计算效率。本次实验对比分析了不同的 c 值对 MLVP 方案的影响。输入不同的隐私保护强度 $1/c$ 值(5.6, 6.3, 7.1, 8.3, 10),分析随着 $1/c$ 值越来越高,对垂直分割方法以及计算性能的影响。

(6) 不同垂直分割方案的数据计算速度与查询速度对比

垂直分割方案的数据计算速度和查询速度是评估垂直分割方案的重要指标。现有的垂直分割方案没有考虑 CSP 共谋的问题,因此在数据查询时需要额外的操作防止数据泄露的问题。Ciriani 方案^[8]通过第三方服务器进行查询,Oriol 方案^[4]则需要对部分数据进行加密处理。本文实验将 MLVP 方案与以上两个方案进行数据计算速度和

查询速度上的对比。

首先在数据计算速度上进行对比。因为 Ciriani 方案和 Oriol 方案与 MLVP 方案的计算方法都利用输入的属性间约束进行属性划分,所以本文选择以上两个方案作为垂直分割比较对象。本文将 MLVP 方案计算出的 L -推断集作为属性间的约束,输入 Ciriani 方案和 Oriol 方案中计算垂直分割方法,以对比不同方案的数据计算时间。

接着对比不同方案的数据查询速度。在真实网络环境中部署 3 个方案生成的垂直分割方法。MLVP 方案能在部分服务器泄露时保护隐私,因此当查询涉及的服务器少于或等于 k 个时,可以执行直接查询,否则通过可信服务器进行查询。Oriol 方案因为没有考虑 CSP 共谋的情况,查询必须全部通过可信服务器, Ciriani 方案则需要对属性进行加密。本文将 Adult 数据集拆分,分别为包含 9 600、19 200、28 800、38 400、48 000 数据的子集,对比数据集的增大对查询时间的影响。

4.2 机器学习算法对 MLVP 方案的影响

MLVP 方案产生 3 种符合 (k, n) -安全定义的分割方法:方法 A,方法 B 和方法 C。方法 $A=\{\{\text{性别, 种族, 教育程度, 祖国}\}, \{\text{婚姻状况, 工作种类, 薪资等级}\}, \{\text{年龄, 职业}\}\}$;方法 $B=\{\{\text{种族, 婚姻状况}\}, \{\text{性别, 年龄, 工作种类, 薪资等级}\}, \{\text{教育程度, 祖国, 职业}\}\}$;方法 $C=\{\{\text{教育程度, 工作种类}\}, \{\text{年龄, 性别, 种族, 祖国, 职业, 薪资等级}\}, \{\text{婚姻状况}\}\}$ 。

机器学习算法对 MLVP 方案的影响实验结果表明,机器学习算法对 MLVP 方案的结果有很大影响。输入不同机器学习算法时的计算结果如表 2 所示,8 种机器学习算法产生了 3 种结果,并且有两种算法无法得到垂直分割方法。线性回归、岭回归、Lasso 和贝叶斯岭回归 4 种算法产生了相同的方法 A,偏最小二乘回归产生了方法 B, k 近邻回归产生了方法 C,决策树算法和支持向量机算法都未能得到分割方法。在无法得到结果的机器学习算法输入中,决策树算法是因为得到的 L -推断集过于复杂,支持向量机算法则是无法得到有效的 L -推断集。

表2 输入不同机器学习算法时的计算结果
Table 2 Calculation results when inputting different machine learning algorithms

机器学习算法	推断率阈值 c	L -推断集个数	k	结果
线性回归	0.178	56	2	方法A
岭回归	0.178	56	2	方法A
Lasso	0.178	56	2	方法A
贝叶斯岭回归	0.178	56	2	方法A
偏最小二乘回归	0.178	36	2	方法B
k 近邻回归	0.27	49	2	方法C
决策树回归	0.24	52	2	无
支持向量机	0	0	0	无

在未确定攻击者使用机器学习算法的情况下, 岭回归是生成垂直分割方法的最佳选择。不同机器学习算法速度对比如图3所示, 在得到相同结果的4种机器学习算法中, 岭回归展现出最快的计算速度, 计算时间是2.12 s; 得到方法B的偏最小二乘回归的计算时间是2.08 s; 得到方法C的 k 近邻回归算法的计算时间是303.26 s; 未能得到分割方案的决策树算法的计算时间是9.24 s。因为支持向量机算法无法得到 L -推断集的集合, 因此计算时间不存在。数据拥有者在这个数据集中决定MLVP方案的机器学习算法时, 应该选择分割方案结果和大多数算法一致, 并且计算速度较快的岭回归算法。

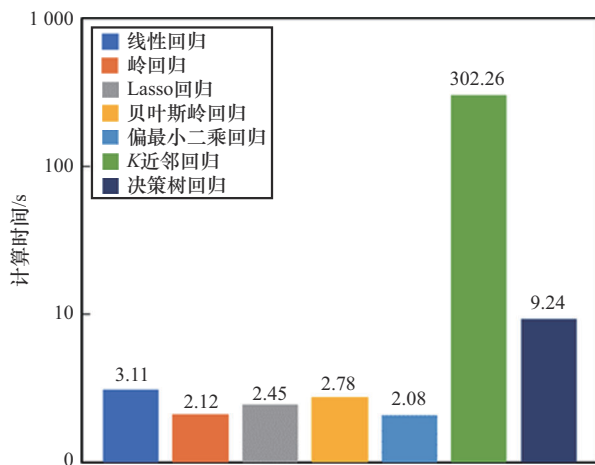


图3 不同机器学习算法速度对比
Figure 3 Comparison of speed of different machine learning algorithms

4.3 MLVP方案性能评估实验结果

MLVP方案性能评估实验结果表明: 在一定范

围内降低推断率阈值 c 的值能提高计算 L -推断集的效率。隐私保护强度对计算 L -推断集时间的影响如图4所示, 当隐私保护强度分别是5.6、6.3、7.1、8.3、10时, 对应的计算时间为1.61 s、2.54 s、3.63 s、6.51 s、7.34 s。这些结果表明, 随着隐私保护强度的增强, L -推断集的计算时间呈现逐渐增长的趋势。出现这种现象的原因是, 更高的隐私保护强度意味着更低的推断率阈值, 使更多推断率较低的属性集合被纳入 L -推断集的计算中, 导致整体计算时间增加。

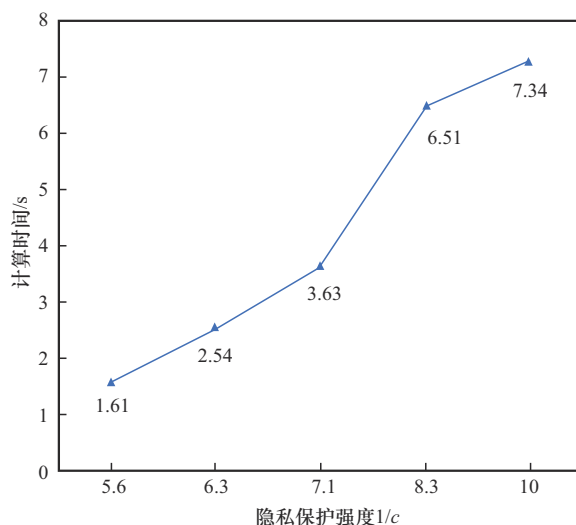
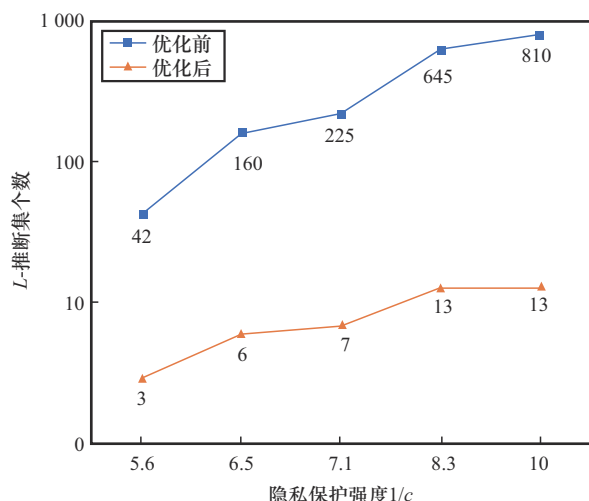


图4 隐私保护强度对计算 L -推断集时间的影响
Figure 4 Impact of privacy protection strength on the time to calculate L -inference set

在一定范围内降低推断率阈值 c 的值能有效增强隐私保护的效果。隐私保护强度对 L -推断集生成的影响如图5所示, 在隐私保护强度为5.6到8.3内, 随着隐私保护强度的提升, 优化后的 L -推断集个数从3增加到13, 这说明分割方法的隐私保护效果也在增强。而在隐私保护强度为8.3到10内, 尽管优化前的 L -推断集个数从645增加到810, 但是优化后的 L -推断集个数维持在13不变, 这说明隐私保护的效果并没有变强。

MLVP方案中对 L -推断集的集合进行优化是有效的。如图5所示, 在隐私保护强度 $1/c$ 分别为5.6、7.1、10时, 优化前的 L -推断集个数分别是42、225、810; 优化后的 L -推断集个数分别是3、7、13。可知优化后的 L -推断集的数量显著减少, 这将减少计算垂直分割方法步骤中生成的约束条件式, 从而使MLVP方案的计算速度加快。

图5 隐私保护强度对 L -推断集生成的影响Figure 5 The impact of privacy protection strength on L -inference set generation

机器学习算法是影响 MLVP 方案计算时间的主要因素。从图4和图6中可以看出,当隐私保护强度 $1/c$ 为 5.6、7.1、10 时,计算 L -推断集分别耗时 1.61 s、3.63 s、7.34 s; 计算垂直分割方法分别耗时 0.42 s、0.16 s、0.42 s。可以得知,计算 L -推断集的时间随着隐私强度的增加也在增加。因为计算垂直分割方法的耗时取决于具体的 L -推断集的内容,所以隐私强度对该步骤耗时的影响不大。结合图4和图6发现, L -推断集计算的时间平均占整体方案计算时间的 93.3%。

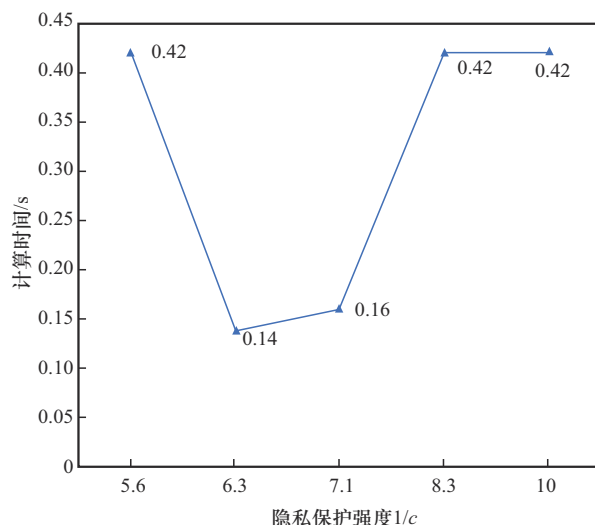


图6 隐私保护强度对计算垂直分割方法时间的影响

Figure 6 Impact of privacy protection strength on the time of computing attribute partitioning methods

4.4 不同属性分割方案计算与查询速度对比结果

不同属性分割方案计算与查询速度对比结果表明, MLVP 方案在计算速度上的劣势是可以接受的。不同垂直分割方案计算速度对比如图7所示,当 L -推断集的个数分别为 42、225、810 时, MLVP 方案所需要的计算时间分别是 426.9 ms、166 ms、421 ms; Ciriani 方案所需要的时间分别是 0.143 ms、0.345 ms、0.548 ms; Oriol 方案所需要的时间分别是 0.042 ms、0.082 ms、0.093 ms。因为 MLVP 方案相比 Ciriani 方案和 Oriol 方案需要考虑 CSP 共谋的情况,因此 MLVP 方案在计算上所需要时间较多,这一差距在实际应用中是可以接受的。主要原因是:① MLVP 方案的垂直分割方法计算是一个离线操作,仅需执行一次即可得到持久化结果;② 即便耗费最多时间的 426.9 ms,也并不是一个很长的时间。

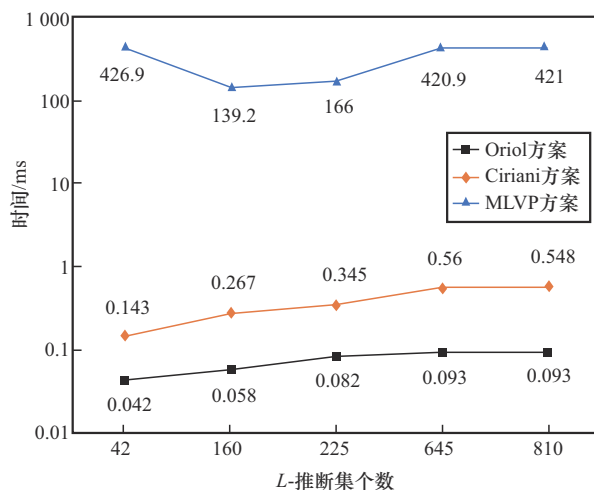


图7 不同垂直分割方案计算速度对比

Figure 7 Comparison of calculation speed of different vertical partitioning schemes

MLVP 方案在数据查询速度上优于 Ciriani 方案和 Oriol 方案。不同垂直分割方案数据查询速度对比如图8所示,当数据集大小为 9 600、28 800、48 000 时, MLVP 方案的平均查询速度分别是 81 ms、82.83 ms、83.7 ms; Ciriani 方案的平均查询速度分别是 89.03 ms、90.9 ms、88.44 ms; Oriol 方案的平均查询速度分别是 104 ms、108.15 ms、112.76 ms。可以看出,随着数据集的增大,各方案的查询时间增加不显著。MLVP 方案的查询速度优势主要是因为将属

性垂直分割后, 可以更有效地并行处理查询请求, 尤其是查询涉及多个属性时。Oriol 方案在执行数据查询时必须依赖可信服务器, 这增加了查询操作的延迟, 平均较 MLVP 方案慢 32.6%。Ciriani 方案对部分数据执行了加密操作, 尽管增加了数据安全性, 但也相应地增加了数据查询的计算复杂性, 使查询速度比 MLVP 方案慢 8.8%。

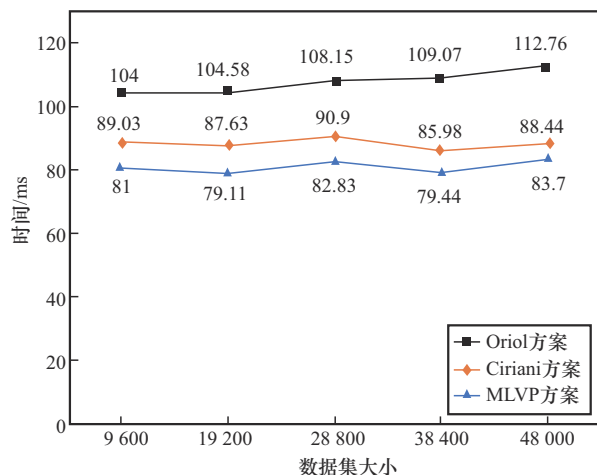


图8 不同垂直分割方案数据查询速度对比

Figure 8 Comparison of data query speed of different vertical partitioning schemes

5 结束语

垂直分割已被公认为一种行之有效的数据隐私保护方法。本文考虑到多个 CSP 之间可能存在的共谋风险, 针对匿名数据存储给出了属性的 (k, n) -安全分割定义, 设计并实现了自动计算满足该定义的分割方法的 MLVP 方案。实验结果表明, 在 8 种被评估的机器学习算法中, 采用岭回归算法的方案结果与其他 3 种算法一致, 并且在计算速度上表现最佳。此外, 隐私保护效果在一定范围内随着推断率阈值 c 下降而增强。尽管在计算速度上存在劣势, 但是本文方案在查询速度上占优势, 分别比 Ciriani 方案和 Oriol 方案快 32.6% 和 8.8%。

在属性分割方面, 本文的研究主要聚焦于确保分割方案符合 (k, n) -安全定义。在实际应用场景中, 数据分割中数据的均衡也是一个关键因素。数据的均衡不仅影响数据处理的效率, 还可能影响运行性能。因此, 未来的研究将着重关注在保证数据隐私的前提下, 优化数据分割的均衡

性, 确保各分割部分的数据量尽可能均等, 以实现更优的数据处理效果。

参考文献:

- [1] DU M X, WANG Q, HE M Q, et al. Privacy-preserving indexing and query processing for secure dynamic cloud storage[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(9): 2320-2332.
- [2] SINGH P, PANDEY A K. A review on cloud data security challenges and existing countermeasures in cloud computing[J]. International Journal of Data Informatics and Intelligent Computing, 2022, 1(2): 23-33.
- [3] DHINAKARAN D, JOE PRATHAP P M. Preserving data confidentiality in association rule mining using data share allocator algorithm[J]. Intelligent Automation & Soft Computing, 2022, 33(3): 1877-1892.
- [4] MEHROTRA S, SHARMA S, ULLMAN J, et al. Partitioned data security on outsourced sensitive and non-sensitive data[C]//Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE). Piscataway: IEEE Press, 2019: 650-661.
- [5] ORIOL F, RIBES-GONZÁLEZ J, RICCI S. Privacy-preserving data splitting: a combinatorial approach[J]. Designs, Codes and Cryptography, 2021, 89(7): 1735-1756.
- [6] BADRAN S, ARMAN N, FARAJALLAH M. An efficient approach for secure data outsourcing using hybrid data partitioning[C]//Proceedings of the 2021 International Conference on Information Technology (ICIT). Piscataway: IEEE Press, 2021: 418-423.
- [7] SOHAL M, BHARANY S, SHARMA S, et al. A hybrid multi-cloud framework using the IBBE key management system for securing data storage[J]. Sustainability, 2022, 14(20): 13561.
- [8] CIRIANI V, DE CAPITANI DI VIMERCATI S, FORESTI S, et al. Combining fragmentation and encryption to protect privacy in data storage[J]. ACM Transactions on Information and System Security, 2010, 13(3): 1-33.
- [9] SWEENEY L. K-anonymity: a model for protecting privacy[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(5): 557-570.
- [10] GUO W X, ZHEN H L, LI X J, et al. Machine learning methods in solving the Boolean satisfiability problem[J]. Machine Intelligence Research, 2023, 20(5): 640-655.
- [11] MEHROTRA S, OKTAY K Y, SHARMA S. Exploiting data sensitivity on partitioned data[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 274-299.
- [12] AGGARWAL G, BAWA M, GANESAN P, et al. Two can keep a secret: a distributed architecture for secure database services[J]. 2nd Biennial Conference on Innovative Data Systems Research, CIDR 2005, 2005: 186-199.
- [13] YANG J J, LI J Q, NIU Y. A hybrid solution for privacy preserving medical data sharing in the cloud environment[J]. Future Genera-

tion Computer Systems, 2015, 43: 74-86.

- [14] GANAPATHY V, THOMAS D, FEDER T, et al. Distributing data for secure database services[C]//Proceedings of the 4th International Workshop on Privacy and Anonymity in the Information Society. New York: ACM, 2011: 1 - 10.
- [15] CIRIANI V, DE CAPITANI DI VIMERCATI S, FORESTI S, et al. Selective data outsourcing for enforcing privacy[J]. Journal of Computer Security, 2011, 19(3): 531-566.
- [16] SHARAD M, SHARMA S, ULLMAN J, et al. Partitioned data security on outsourced sensitive and non-sensitive data[C]//Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE). Piscataway: IEEE Press, 2019: 650-661.
- [17] SUDHA R, POOJA G, REVATHY V, et al. Advances in parallel computing[M]. Amsterdam: IOS Press, 2021: 854-858.
- [18] DENIS N, CHAFFARDON P, CONAN D, et al. Privacy-preserving content-based publish/subscribe with encrypted matching and data splitting[C]//Proceedings of the 17th International Joint Conference on e-Business and Telecommunications. SCITEPRESS - Science and Technology Publications, 2020: 405 - 414.
- [19] ATTASENA V, DARMONT J, HARBI N. Secret sharing for cloud data security: a survey[J]. The VLDB Journal, 2017, 26(5): 657-681.
- [20] ASUNCION A, NEWMAN D. UCI machine learning repository[R]. 2007.

[作者简介]



阮华锋 (1999-), 男, 广东化州人, 东莞理工学院硕士生, 主要研究方向为数据存储隐私保护。



李睿 (1975-), 男, 湖南汨罗人, 东莞理工学院教授, 主要研究方向为数据隐私保护。



罗凯伦 (1989-), 男, 广东清远人, 东莞理工学院讲师, 主要研究方向为人工智能安全。