计算机工程
Computer Engineering
ISSN 1000-3428,CN 31-1289/TP

《计算机工程》网络首发论文

题目：高效的非交互式隐私保护逻辑回归模型
作者：唐敏，张宇浩，邓国强
DOI：10.19678/j.issn.1000-3428.0065549
网络首发日期：2022-11-04
引用格式：唐敏，张宇浩，邓国强．高效的非交互式隐私保护逻辑回归模型[J/OL]．计算机工程. https://doi.org/10.19678/j.issn.1000-3428.0065549

网络首发时间：2022-11-04 14:57:00
网络首发地址：https://kns.cnki.net/kcms/detail/31.1289.TP.20221104.1352.005.html

计 算 机 工 程
Computer Engineering

本文源代码链接：https://github.com/anndme/SLRT

高效的非交互式隐私保护逻辑回归模型

唐敏，张宇浩，邓国强

(桂林电子科技大学 数学与计算科学学院 广西高校数据分析与计算重点实验室 广西 桂林 541004)

摘 要：逻辑回归作为一种典型的机器学习算法，广泛应用于医疗诊断、金融预测等领域。由于单个用户没有足够的样本构建高精度模型，传统的集中式训练又会导致隐私泄漏，因而构建具有隐私保护的逻辑回归模型受到广泛关注。现有的要求用户和服务器之间进行交互的方案导致了高昂的计算成本和通信负担。本文提出一种高效的非交互式逻辑回归训练协议，利用具有“良可分离结构”的梯度更新公式，解耦了样本数据和模型参数之间的计算耦合性，保证用户与服务器之间的单向单次传输性，即用户将本地数据整合并以秘密共享的方式上传给云服务器后即可离线；训练阶段设计了基于矩阵和向量运算的协议，保证服务器在每次迭代中使用固定的信息更新参数，降低了计算成本和通信开销。同时，提供了协议的安全性分析和数值实验，对来自 UCI 库的 4 个真实数据集上训练逻辑回归模型的实验结果表明，在保证模型精度的前提下，与最新的隐私保护逻辑回归方案 VANE 相比效率有较大提升（80~120 倍），且与明文域中的训练时间相近。

关键词：逻辑回归；隐私保护；良可分离结构；秘密共享；向量化

开放科学（资源服务）标志码（OSID）：

Efficient Non-Interactive and Privacy-Preserving Logistic Regression Model

TANG Min, ZHANG Yuhao, DENG Guoqiang

(School of Mathematics and Computing Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China)

【Abstract】 As a typical machine learning algorithm, logistic regression is widely used in medical diagnosis, financial forecasting and other fields. Since a single user does not have enough samples to build a high-precision model, and the traditional centralized training will lead to privacy leakage, building a logistic regression model with privacy preserving has attracted extensive attention. The existing schemes that require communication between users and servers lead to high computing costs and communication burden. This paper proposes an efficient non-interactive logistic regression training protocol. Using the gradient update formula with a "well-separable structure", the computational coupling between sample data and model parameters is decoupled to ensure one-direction single transmission between users and servers. That is, users can go offline after integrating local data and uploading it to the cloud servers in a secret sharing manner; In the training phase, a protocol based on matrix and vector operation is designed to ensure that the server uses fixed information update parameters in each iteration, reducing the calculation cost and communication overhead. Meanwhile, the protocol security analysis and numerical experiments are provided. The experimental results of training the logistic regression model on four real datasets from the UCI library show that, under the premise of ensuring the accuracy of the model, the efficiency is greatly improved (80~120 times) compared with the latest privacy preserving logistic regression scheme VANE, and the training time is similar to that in the plaintext domain.

【Key words】 logistic regression; privacy-preserving; well-separable structure; secret sharing; vectorization

DOI:10.19678/j.issn.1000-3428.0065549

0 概述

大数据时代下，机器学习对人们生活产生了巨大的影响[1-3]。逻辑回归作为典型的机器学习算法，广泛应用在医疗诊断[4]、文本识别[5]、物联网[6]等多个领域。通常来说，单一的用户或组织没有足够的数据构建高精度的模型，应对本地训练样本不足的方案[7]是聚合不同来源的数据。然而，出于隐私限制很难将带有敏感信息的数据直接集中进行模型训练。因此，如何在隐私保护下构建逻辑回归模型引起人们的广泛关注。

基金项目：广西科技基地和人才专项 (AD18281024)；桂林电子科技大学研究生教育创新计划资助项目 (2022YCXS144)。
作者简介：唐敏 (1980-)，女，副教授，博士，主研方向为计算机代数，机器学习算法的隐私保护；张宇浩，男，硕士研究生，主研方向为隐私计算、机器学习；邓国强 (通信作者)，男，副教授，博士研究生，主研方向为机器学习模型的安全与隐私、分组密码的设计与分析。
E_mail: d9801242@guet.edu.cn

研究者利用密码技术，针对逻辑回归模型隐私保护(Privacy-Preserving Logistic Regression, PPLR)做出了许多工作。其中，同态加密 (Homomorphic Encryption, HE)是最常用的数据安全保护技术。Guo 等人[8]采用 BGN 全同态加密 (Fully Homomorphic Encryption, FHE)系统设计了一个逻辑回归预测阶段的隐私保护医疗预诊方案。Fan 等人[9]使用了一种更高效的 SEAL 全同态库加密数据，提出了在训练阶段的隐私保护逻辑回归算法 (PPLRA)。出于实用性考虑，基于 Chen 等人[10]的 HEAAN 全同态方案，Xu 等人[11]将二分类逻辑回归隐私保护模型推广到多分类。为避免全同态方案的高计算复杂性，Song 等人[12]使用部分同态 (Partial Homomorphic Encryption，PHE)保护数据，通过异步梯度共享算法交换训练中间结果而不暴露隐私，实现了对垂直分区数据的安全训练。尽管同态加密允许在不解密数据的情况下对密文直接进行计算，其输出与用同一方法处理明文的结果一致[13]。然而，针对机器学习模型训练问题，由于迭代次数较多、数据规模较大[14]，无论基于 FHE 还是 PHE 的方案都需要进行代价较大的加解密处理以及多次同态运算，效率不高；另外，用户之间或用户与服务器之间的多次交互也导致大量通信负担。

为了缩小密文训练与明文训练在计算效率上的差距，有学者采用秘密共享技术保护私有数据。Mohassel 等人[15]提出了 SecureML，极大地提高了数据维度较大时逻辑回归训练的效率。在 SecureML 的基础上，Martine 等人[16]引入了一个可信的第三方生成乘法三元组，进一步提高了服务器交互训练阶段的效率；Zheng 等人[17]提出了基于茫然传输的安全矩阵计算方案，利用 OT 扩展协议和批处理，缩减了交互轮数，使训练过程所需的通信开销更低。基于秘密共享的隐私保护方案避免了同态加密所需的巨大计算量，在效率上有很大的提高。然而，由于训练过程由 2 台[16]或 3 台[17]非共谋的服务器协同完成，服务器之间需要多次交互，对网络的可靠性提出更高的要求。

最近，有学者提出非交互式的隐私保护机器学习方案[18-20](用户一次上传加密数据，不参与训练)。典型的代表是 2021 年 Wang 等人[20]设计的 VANE-基于梯度下降的非交互式 PPLR 方案，其迭代训练过程中参数更新操作在明文下进行，因而效率较高。然而，该方案在训练前要求用户使用 Paillier 系统对 $m\cdot(d+1)^2$ （m 为用户数量，d 为数据维度）个元素进行加密并上传给服务器，服务器端也需要聚合并解密同等规模的数据，当属性较多或用户数较多时，该方案的效率有所下降。

为了解决上述方案的局限性，本文提出一个高效的具有隐私保护的逻辑回归训练方案 SLRT (Secure Logistic Regression Training)，不仅实现了用户与服务器之间的非交互式训练，也避免了使用同态导致的高计算复杂性，相比直接拆分原始数据的秘密共享方案，极大地减轻了服务器的通信负担，针对大规模数据时仍能保证高精度和效率。本文的贡献如下：定义了“良可分离结构”，结合逻辑损失函数的近似替换策略，解耦了梯度更新公式中用户数据与模型参数之间的计算耦合性，用户整合本地数据并以秘密共享方式上传给云服务器后即可离线；设计了明文空间上基于矩阵和向量运算的训练协议，保证服务器在每轮迭代中使用固定的信息进行协同训练，减少了传统秘密共享方案中服务器之间的计算开销和通信负担。

1 预备知识

1.1 逻辑回归

逻辑回归[21]是一种广义的线性回归分析模型，常用于解决二分类问题。给定由 n 个样本组成的数据集 $(X,Y)=\{(\boldsymbol{x}_1,y_1),...,(\boldsymbol{x}_n,y_n)\}$，其中 $\boldsymbol{x}_i=(1,x_i^1,x_i^2,...,x_i^d)^{\mathrm{T}}$，$x_i^j$ 表示样本 $\boldsymbol{x}_i$ 的第 j 个特征，$i=1,2,...,n$，$j=1,2,...,d$，第一个元素 1 用于偏置项的计算；$\boldsymbol{x}_i$ 对应的类标签 $y_i\in\{-1,1\}$。

在逻辑回归算法中，使用 sigmoid 函数来构建样本 $\boldsymbol{x}_i$ 所属类别的概率，

$$P(y_i=1\mid\boldsymbol{x}_i,\boldsymbol{\omega})=\frac{1}{1+e^{-y_i\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}},$$

其中权重向量 $\boldsymbol{\omega}=(\omega^0,\omega^1,...,\omega^d)^{\mathrm{T}}$ 是优化的模型参数。

逻辑回归中用损失函数[22]来评估模型的预测值和真实值之间的误差，定义为

$$L(X,Y,\boldsymbol{\omega})=\frac{1}{n}\sum_{i=1}^{n}\log(1+e^{-y_i\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}). \tag{1}$$

通常使用梯度下降最小化 (1) 式来获取最优的模型参数 $\boldsymbol{\omega}$。在第 t 次迭代时，$\boldsymbol{\omega}$ 通过

$$\boldsymbol{\omega}^{t+1}=\boldsymbol{\omega}^{t}-\frac{\eta}{n}\sum_{i=1}^{n}\left(\frac{1}{1+e^{-y_i\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}}-1\right)y_i\boldsymbol{x}_i \tag{2}$$

进行更新，其中 η 为学习率。当模型参数 $\boldsymbol{\omega}^{t}$ 和 $\boldsymbol{\omega}^{t+1}$ 之间的值小于给定的阈值 ε 或达到最大迭代次数时，终止训练。

1.2 秘密共享

1.2.1 加秘密共享

加秘密共享[15]包含两个算法，共享算法将消息 a 分发给两个非共谋的参与方，重构算法根据共享值将原始数据 a 恢复出来。

(1) 共享算法 $Shr^{A}(a)$：为了在两个参与方 P_0，P_1 之间加秘密共享一个原始消息 $a\in\mathbb{R}$。首先，产生随机值 $a_0\in\mathbb{R}$ 作为 a 的一个共享值发送给参与方 P_0，记作 $\langle a\rangle_0$。然后，$\langle a\rangle_1=a-a_0\in\mathbb{R}$ 作为 a 的另一共享值发送给参与方 P_1。这样 a 就在 P_0，P_1 之间共享，但任何一方都无法了解 a 的真实信息。

(2) 重构算法 $Rec^{A}(\langle a\rangle_0,\langle a\rangle_1)$：设 P_0 拥有消息 a 的共享值 $\langle a\rangle_0$，P_1 拥有 a 的共享值 $\langle a\rangle_1$。为了重构出一个被加共享的值 a，P_r 将 $\langle a\rangle_r$ 发送给另一方 P_{1-r}，$r\in\{0,1\}$，或共同发给第三方，通过计算 $\langle a\rangle_0+\langle a\rangle_1$ 重构出 a。

1.2.2 加秘密共享乘法（$SSM(a,b)$）

设 P_0 有原始消息 $a\in\mathbb{R}$ 和 $b\in\mathbb{R}$ 的共享值 $\langle a\rangle_0$，$\langle b\rangle_0$，P_1 有 $\langle a\rangle_1$，$\langle b\rangle_1$，计算完成时双方分别拥有 $c_r=\langle ab\rangle_r\in\mathbb{R}$，$r\in\{0,1\}$，加秘密共享乘法执行过程如下：

(1) 由第三方可信机构 TPA 随机生成乘法三元组 (u,v,z)，其中 $u,v,z\in\mathbb{R}$，$z=uv$。执行共享算法 $Shr^{A}(\cdot)$，将 $\langle u\rangle_r$，$\langle v\rangle_r$ 和 $\langle z\rangle_r$ 分发给 P_r。

(2) P_r 本地计算 $\langle e\rangle_r=\langle a\rangle_r-\langle u\rangle_r$ 和 $\langle f\rangle_r=\langle b\rangle_r-\langle v\rangle_r$，并将 $\langle e\rangle_r$ 和 $\langle f\rangle_r$ 发给另一方 P_{1-r}，双方各自执行重构算法 $Rec^{A}(\langle e\rangle_0,\langle e\rangle_1)$，$Rec^{A}(\langle f\rangle_0,\langle f\rangle_1)$，恢复出 e 和 f。

(3) P_r 计算 $\langle c\rangle_r=r\cdot e\cdot f+f\cdot\langle u\rangle_r+e\cdot\langle v\rangle_r+\langle z\rangle_r$。

P_r 可以通过将 $\langle c\rangle_r$ 发送给 P_{1-r}，或共同发给第三方执行 $Rec^{A}(\langle c\rangle_0,\langle c\rangle_1)$ 重构得到 c，其中 $c=ab$。

1.2.3 加秘密共享内积（$SSIP(\boldsymbol{a},\boldsymbol{b})$）

假设 P_0 有向量 $\boldsymbol{a}\in\mathbb{R}^{1\times n}$ 和 $\boldsymbol{b}\in\mathbb{R}^{n\times 1}$ 的共享值 $\langle\boldsymbol{a}\rangle_0$，$\langle\boldsymbol{b}\rangle_0$，$P_1$ 有共享值 $\langle\boldsymbol{a}\rangle_1$，$\langle\boldsymbol{b}\rangle_1$。参与方 P_0，P_1 通过执行 $SSIP(\boldsymbol{a},\boldsymbol{b})$ 分别得到 $\langle z\rangle_r=\langle\boldsymbol{a}\cdot\boldsymbol{b}\rangle_r\in\mathbb{R}$，

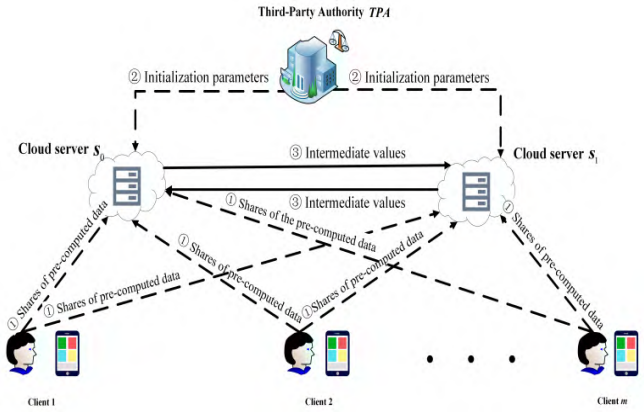


图 1 系统模型

Fig.1 System model

$r\in\{0,1\}$。加秘密共享内积协议执行过程如下：

(1) TPA 随机生成 Beaver's 乘法三元组[23] $(\boldsymbol{g},\boldsymbol{f},h)$，其中 $\boldsymbol{g}\in\mathbb{R}^{1\times n}$，$\boldsymbol{f}\in\mathbb{R}^{1\times n}$，$h=\boldsymbol{g}\cdot\boldsymbol{f}\in\mathbb{R}$。执行共享算法 $Shr^{A}(\cdot)$，并将共享值 $\langle\boldsymbol{g}\rangle_r$，$\langle\boldsymbol{f}\rangle_r$ 和 $\langle h\rangle_r$ 分发给 P_r。

(2) 参与方 P_r 收到乘法三元组的共享值后，本地计算 $\langle\boldsymbol{d}\rangle_r=\langle\boldsymbol{a}\rangle_r-\langle\boldsymbol{g}\rangle_r$，$\langle\boldsymbol{e}\rangle_r=\langle\boldsymbol{b}\rangle_r-\langle\boldsymbol{f}\rangle_r$，并将 $\langle\boldsymbol{d}\rangle_r$，$\langle\boldsymbol{e}\rangle_r$ 发送给另一方 P_{1-r}。P_r 双方各自执行重构算法 $Rec^{A}(\langle\boldsymbol{d}\rangle_0,\langle\boldsymbol{d}\rangle_1)$，$Rec^{A}(\langle\boldsymbol{e}\rangle_0,\langle\boldsymbol{e}\rangle_1)$，得到 $\boldsymbol{d}$ 和 $\boldsymbol{e}$。

(3) P_r 计算 $\langle z\rangle_r=r\cdot\boldsymbol{d}\cdot\boldsymbol{e}+\langle h\rangle_r+\langle\boldsymbol{g}\rangle_r\cdot\boldsymbol{e}+\boldsymbol{d}\cdot\langle\boldsymbol{f}\rangle_r$。

P_r 可以将 $\langle z\rangle_r$ 发送给 P_{1-r}，或共同发给第三方执行 $Rec^{A}(\langle z\rangle_0,\langle z\rangle_1)$ 重构得到 z，其中 $z=\boldsymbol{a}\cdot\boldsymbol{b}$。

2 模型与安全性需求

2.1 系统模型

系统模型由三部分组成：1) 第三方可信机构 TPA；2) 用户 C_i (i=1,...,m)；3) 云服务器 S_0，S_1。如图 1 所示。

(1) 第三方可信机构 TPA 主要负责系统初始化，生成训练参数以及乘法三元组并分发给云服务器 S_0，S_1。

(2) 用户 C_i 拥有本地数据集 D_i，每个样本包含完整的属性。C_i 首先在本地对其私有数据进行预处理，然后使用秘密共享将整合的本地局部数据矩阵拆分成两部分，分别上传给云服务器 S_0，S_1

(3) 两台非共谋云服务器 S_0 和 S_1 分别聚合来自多个用户的本地数据共享矩阵，负责协同训练获得全局模型参数的共享。

2.2 威胁模型与安全需求

SLRT 的主要目标是在保护用户数据和模型参数的隐私的前提下训练逻辑回归模型。本文考虑了以下两种攻击类型：

(1) 诚实且好奇的攻击：在 SLRT 系统中涉及的用户 C_i 和云服务器 S_0 和 S_1 都是诚实且好奇的，即正确地执行训练协议，但会在训练过程中尝试了解更多信息。

(2) 共谋攻击：虽然在 SLRT 系统中两台云服务器是非共谋的，这在安全两方的隐私保护协议中被广泛采用[15-17]。但允许云服务器 S_0，S_1 通过生成合法客户端与一些本地客户端共谋，并试图从获取的数据中推断出一些隐私信息。

3 SLRT 方案

本节详细介绍非交互式逻辑回归安全训练方案 SLRT。给出“良可分离结构”的定义，结合逻辑损失函数近似替换策略构建新的梯度计算格式；用户对其本地数据进行预处理，使用秘密共享将局部数据矩阵拆分，分别上传给两台云服务器 S_0，S_1；经 S_0, S_1 协同训练，得到模型参数的共享值。

定义 1 良可分离结构

函数 $f(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\omega})$ 具有良可分离结构，如果 $f(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\omega})$ 可以表示成

$$f(\boldsymbol{x},y;\boldsymbol{\omega}) \coloneqq \sum\nolimits_j g_j(\boldsymbol{x},y)h_j(\boldsymbol{\omega}),$$

其中 $g_j(\boldsymbol{x},y)$ 是关于 $\boldsymbol{x}$ 和 y 的函数，$h_j(\boldsymbol{\omega})$ 是关于 $\boldsymbol{\omega}$ 的函数。

注意到，具有良可分离结构的函数 f 可表示函数 g_j 和 h_j 的乘积的和式，其中 g_j 和 h_j 具有完全不同的自变量。当 $\boldsymbol{x}$，y 分别表示样本特征和类标签，$\boldsymbol{\omega}$ 表示模型参数时，可以将函数 f 看作是机器学习训练中的梯度更新公式。如果其可表示成良可分离结构形式，意味着函数 g_j 的计算只依赖于样本，h_j 只依赖于 $\boldsymbol{\omega}$。也就是说，g_j 可由用户在本地计算；另一方面，拥有参数 $\boldsymbol{\omega}$ 的服务器一旦获得 g_j，即可获得梯度 $f(\boldsymbol{x},y;\boldsymbol{\omega})$。

3.1 良可分离结构的构建

SLRT 方案利用梯度计算公式的良可分离结构，实现用户与服务器的非交互式训练。由于梯度更新公式（2）涉及幂运算和除法，很难将其转变为样本信息和参数信息完全分离的计算形式。

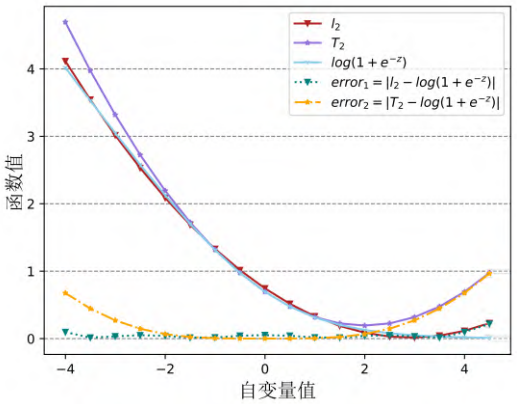


图 2 拟合效果示意图

Fig.2 Schematic diagram of fitting effect

Wang 等人[20]提出了一种基于二阶泰勒多项式 T_2 近似逻辑函数的良可分离结构。考虑到使用低阶的泰勒多项式可能导致精度的损失，本文采用连续最小二乘近似（Continuous Least Squares Approximation，CLSA）[24]构造替代函数，克服了泰勒多项式近似在远离原点时偏离原函数的局限。

3.1.1 逻辑函数近似

CLSA 的目标是在整个给定区间 $[a,b]$ 上找到一个近似函数，满足与原函数之间的误差平方和最小。使用 CLSA 算法，区间 $[-4,4]$ 上近似 $\log(1+e^{-z})$ 的二阶多项式为

$$l_2 = \xi_2 z^2 + \xi_1 z + \xi_0, \tag{3}$$

其中系数 $\xi_0 = 0.744204$，$\xi_1 = -0.5$，$\xi_2 = 0.085660$。图 2 给出了 T_2 和 l_2 逼近逻辑函数的效果。

将二阶多项式（3）代入到（1）式中的逻辑函数，损失函数的近似表示为：

$$\tilde{L}(X,Y,\boldsymbol{\omega}) = \frac{1}{n}\left[\sum_{i=1}^{n}\xi_2(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\omega}^t)^2 + \xi_1 y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\omega}^t) + \xi_0\right]. \tag{4}$$

此时，逻辑回归的参数更新公式转换为：

$$\omega_0^{t+1} = \omega_0^t - \frac{\eta}{n}\sum\nolimits_{i=1}^{n}(2\xi_2(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\omega}^t) + \xi_1 y_i), \tag{5}$$

$$\omega_j^{t+1} = \omega_j^t - \frac{\eta}{n}\sum\nolimits_{i=1}^{n}(2\xi_2(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\omega}^t) + \xi_1 y_i)\boldsymbol{x}_i^j, \tag{6}$$

其中 t 为当前迭代次数，$j = 1,2,\ldots,d$。

3.1.2 解耦的梯度计算格式

经二阶多项式 l_2 近似逻辑函数，模型参数更新公式（5）和（6）具有良可分离结构，其等价于

$$\begin{aligned}\omega_0^{t+1} &= \sum\nolimits_{j=1}^{3} g_j(\boldsymbol{x},y)h_j(\boldsymbol{\omega}) \\ &= \omega_0^t - \frac{2\eta}{n}\xi_2\left(\sum\nolimits_{i=1}^{n}\boldsymbol{x}_i^{\mathrm{T}}\right)\cdot\boldsymbol{\omega}^t - \frac{\eta}{n}\xi_1\sum\nolimits_{i=1}^{n} y_i,\end{aligned} \tag{7}$$

$$\omega_j^{t+1}=\sum_{j=4}^{6}g_j(\boldsymbol{x},y)h_j(\boldsymbol{\omega})$$
$$=\omega_j^t-\frac{2\eta}{n}\xi_2(\sum_{i=1}^{n}\boldsymbol{x}_i^{\mathrm{T}}\cdot x_i^j)\cdot\boldsymbol{\omega}^t-\frac{\eta}{n}\xi_1\sum_{i=1}^{n}y_i x_i^j, \quad (8)$$

其中，

$$g_1(\boldsymbol{x},y)=0,\qquad h_1(\boldsymbol{\omega})=\omega_0^t,$$
$$g_2(\boldsymbol{x},y)=\sum_{i=1}^{n}\boldsymbol{x}_i^{\mathrm{T}},\qquad h_2(\boldsymbol{\omega})=\frac{2\eta}{n}\xi_2\boldsymbol{\omega}^t,$$
$$g_3(\boldsymbol{x},y)=\sum_{i=1}^{n}y_i,\qquad h_3(\boldsymbol{\omega})=-\frac{\eta}{n}\xi_1,$$
$$g_4(\boldsymbol{x},y)=0,\qquad h_4(\boldsymbol{\omega})=\omega_j^t,$$
$$g_5(\boldsymbol{x},y)=\sum_{i=1}^{n}\boldsymbol{x}_i^{\mathrm{T}}\cdot x_i^j,\qquad h_5(\boldsymbol{\omega})=\frac{2\eta}{n}\xi_2\boldsymbol{\omega}^t,$$
$$g_6(\boldsymbol{x},y)=\sum_{i=1}^{n}y_i\cdot x_i^j,\qquad h_6(\boldsymbol{\omega})=-\frac{\eta}{n}\xi_1.$$

注意到

$$g_2(\boldsymbol{x},y),\ g_3(\boldsymbol{x},y),\ g_5(\boldsymbol{x},y),\ g_6(\boldsymbol{x},y) \quad (9)$$

仅与样本有关且每次迭代过程中保持不变。结合(7)和（8），注意到（9）式不泄漏单个样本数据，同时能为服务器提供模型训练所需的信息，所以用户在本地计算(9)式并上传给服务器后即可离线。为保护模型参数和提高计算效率，(9)式将以秘密共享的形式上传给 S_0 和 S_1。

注记: 若采用高次近似多项式替代逻辑函数，仍可构造具有良可分离的结构的参数更新公式，但形式上比二次近似要复杂得多。通过训练第 5 节表 1 中列出的数据集，发现使用高阶多项式并不能提高 LR 模型的精度。主要原因是 CLSA 方法获得的二阶多项式与逻辑函数在整个区间上已经具有相同的行为。出于效率的原因，下面给出基于 (7) 和 (8) 的本地数据预处理方法，而不采用更高阶的近似多项式。

3.2 局部数据共享矩阵生成

根据 3.1.2 节设计的梯度计算格式，为了让 S_0，S_1 获得（9）式的共享，用户 C_i 首先对本地数据集进行预处理。设用户 C_i (i=1,2,...,m)拥有数据集

$$D_i=\{(\boldsymbol{x}_{i1},y_{i1}),(\boldsymbol{x}_{i2},y_{i2}),...,(\boldsymbol{x}_{in_i},y_{in_i})\},$$

其中，$(\boldsymbol{x}_{ik},y_{ik})=(1,x_{ik}^1,x_{ik}^2,...,x_{ik}^d,y_{ik})$，$n_i$ 表示用户 C_i 拥有的样本数。

首先，用户 C_i 对数据集 D_i 中的每个样本 $(\boldsymbol{x}_{ik},y_{ik})$ 计算

$$A_{ik}=(1,x_{ik}^1,x_{ik}^2,...,x_{ik}^d)^{\mathrm{T}}(x_{ik}^1,x_{ik}^2,...,x_{ik}^d,y_{ik}),$$

得到样本矩阵

$$A_{ik}=\begin{bmatrix}x_{ik}^1 & \cdots & x_{ik}^d & y_{ik}\\ x_{ik}^1\cdot x_{ik}^1 & \cdots & x_{ik}^d\cdot x_{ik}^1 & y_{ik}\cdot x_{ik}^1\\ \vdots & \ddots & \vdots & \vdots\\ x_{ik}^1\cdot x_{ik}^d & \cdots & x_{ik}^d\cdot x_{ik}^d & y_{ik}\cdot x_{ik}^d\end{bmatrix}. \quad (10)$$

然后，C_i 计算本地局部数据矩阵

$$A_i=\sum_{k=1}^{n_i}A_{ik}$$
$$=\begin{bmatrix}\sum_{k=1}^{n_i}x_{ik}^1 & \cdots & \sum_{k=1}^{n_i}x_{ik}^d & \sum_{k=1}^{n_i}y_{ik}\\ \sum_{k=1}^{n_i}x_{ik}^1\cdot x_{ik}^1 & \cdots & \sum_{k=1}^{n_i}x_{ik}^d\cdot x_{ik}^1 & \sum_{k=1}^{n_i}y_{ik}\cdot x_{ik}^1\\ \vdots & \ddots & \vdots & \vdots\\ \sum_{k=1}^{n_i}x_{ik}^1\cdot x_{ik}^d & \cdots & \sum_{k=1}^{n_i}x_{ik}^d\cdot x_{ik}^d & \sum_{k=1}^{n_i}y_{ik}\cdot x_{ik}^d\end{bmatrix}. \quad (11)$$

最后，对矩阵 A_i 中每个元素 $(A_i)_{pq}$，$p=1,2,...,d+1$，$q=1,2,...,d+1$，执行 $Shr^A((A_i)_{pq})$，将 A_i 拆分成

$$\langle A_i\rangle_r=\begin{bmatrix}\langle\sum_{k=1}^{n_i}x_{ik}^1\rangle_r & \cdots & \langle\sum_{k=1}^{n_i}x_{ik}^d\rangle_r & \langle\sum_{k=1}^{n_i}y_{ik}\rangle_r\\ \langle\sum_{k=1}^{n_i}x_{ik}^1\cdot x_{ik}^1\rangle_r & \cdots & \langle\sum_{k=1}^{n_i}x_{ik}^d\cdot x_{ik}^1\rangle_r & \langle\sum_{k=1}^{n_i}y_{ik}\cdot x_{ik}^1\rangle_r\\ \vdots & \ddots & \vdots & \vdots\\ \langle\sum_{k=1}^{n_i}x_{ik}^1\cdot x_{ik}^d\rangle_r & \cdots & \langle\sum_{k=1}^{n_i}x_{ik}^d\cdot x_{ik}^d\rangle_r & \langle\sum_{k=1}^{n_i}y_{ik}\cdot x_{ik}^d\rangle_r\end{bmatrix},$$

其中 $r\in\{0,1\}$，满足 $A_i=\langle A_i\rangle_r+\langle A_i\rangle_{1-r}$。

用户 C_i 将矩阵 $\langle A_i\rangle_0$ 上传给 S_0，将 $\langle A_i\rangle_1$ 上传给 S_1，用于生成全局训练数据矩阵，随即离线。

3.3 全局逻辑回归模型训练

本节给出 S_0 和 S_1 进行全局逻辑回归模型训练的详细过程。

3.3.1 全局数据共享矩阵生成

当 S_r 收到所有用户 C_i 上传的本地数据共享矩阵后，通过计算

$$\langle A\rangle_r=\sum_{i=1}^{m}\langle A_i\rangle_r$$
$$=\begin{bmatrix}\sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle x_{ik}^1\rangle_r & \cdots & \sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle x_{ik}^d\rangle_r & \sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle y_{ik}\rangle_r\\ \sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle x_{ik}^1\cdot x_{ik}^1\rangle_r & \cdots & \sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle x_{ik}^d\cdot x_{ik}^1\rangle_r & \sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle y_{ik}\cdot x_{ik}^1\rangle_r\\ \vdots & \ddots & \vdots & \vdots\\ \sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle x_{ik}^1\cdot x_{ik}^d\rangle_r & \cdots & \sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle x_{ik}^d\cdot x_{ik}^d\rangle_r & \sum_{i=1}^{m}\sum_{k=1}^{n_i}\langle y_{ik}\cdot x_{ik}^d\rangle_r\end{bmatrix}$$

得到全局数据共享矩阵。由于云服务器得到的是经过预计处理的本地训练数据矩阵，其每个元素为多个样本属性之和的共享值，因此 S_0，S_1 即使共谋也不可能窃取单个样本信息，同时避免了传统的秘密共享方案每次迭代过程中，由于样本和参数的计算耦合性而造成的服务器之间多轮通信。

为了简化记号，用 $\langle v_{ij}\rangle_r$ 表示矩阵 $\langle A\rangle_r$ 中的元素 $(\langle A\rangle_r)_{ij}$，$\langle u_i\rangle_r$ 表示矩阵 $\langle A\rangle_r$ 中的元素 $(\langle A\rangle_r)_{i(d+1)}$，

$i=1,2,...,d+1$， $j=1,2,...,d$。于是$\langle A\rangle_r$可表示为

$$\langle A\rangle_r=\begin{bmatrix}\langle v_{01}\rangle_r & \cdots & \langle v_{0d}\rangle_r & \langle u_0\rangle_r\\ \langle v_{11}\rangle_r & \cdots & \langle v_{1d}\rangle_r & \langle u_1\rangle_r\\ \vdots & \ddots & \vdots & \vdots\\ \langle v_{d1}\rangle_r & \cdots & \langle v_{dd}\rangle_r & \langle u_d\rangle_r\end{bmatrix}. \tag{12}$$

3.3.2 全局模型参数更新

回顾3.1.2节不难发现，非交互式逻辑回归训练的关键是获得（9）式，而（9）式可以通过C_i预处理本地数据，再经S_0，S_1聚合得到。此时，（7）式和（8）式可通过下式计算

$$\omega_0^{t+1}=\omega_0^{t}-\frac{\eta}{n}(2\xi_2(\boldsymbol{v}_0\cdot\boldsymbol{\omega}^t)+\xi_1u_0), \tag{13}$$

$$\omega_j^{t+1}=\omega_j^{t}-\frac{\eta}{n}(2\xi_2(\boldsymbol{v}_j\cdot\boldsymbol{\omega}^t)+\xi_1u_j), \tag{14}$$

其中$j=1,2,...,d$，$\boldsymbol{v}_0$，$\boldsymbol{v}_j$，u_0，u_j可由矩阵（12）得到

$$\begin{aligned}&\boldsymbol{v}_0=(Rec^A(\langle n\rangle_0,\langle n\rangle_1),Rec^A(\langle v_{01}\rangle_0,\langle v_{01}\rangle_1),...,Rec^A(\langle v_{0d}\rangle_0,\langle v_{0d}\rangle_1)),\\&\boldsymbol{v}_j=(Rec^A(\langle v_{0j}\rangle_0,\langle v_{0j}\rangle_1),Rec^A(\langle v_{j1}\rangle_0,\langle v_{j1}\rangle_1,...,Rec^A(\langle v_{jd}\rangle_0,\langle v_{jd}\rangle_1)),\\&u_0=Rec^A(\langle u_0\rangle_0,\langle u_0\rangle_1),\\&u_j=Rec^A(\langle u_j\rangle_0,\langle u_j\rangle_1).\end{aligned} \tag{15}$$

一旦获得（15）式，服务器就能计算（13）式和（14）式，从而自行完成整个逻辑回归模型的训练。接下来，介绍S_0，S_1如何利用聚合生成的全局数据共享矩阵$\langle A\rangle_r$协同训练模型。

3.3.3 全局逻辑回归安全训练

S_0，S_1生成全局数据共享矩阵$\langle A\rangle_r$后，通过下述操作进行全局逻辑回归模型训练。

(1) *TPA*首先负责初始化[16]工作，选择学习率η，最大迭代次数*itermax*，随机选择乘法三元组$(\boldsymbol{g},\boldsymbol{f},h)$，模型参数初始值$\boldsymbol{\omega}^0=(\omega_0,\omega_1,...,\omega_d)$，利用共享算法$Shr^A(\cdot)$将$(\boldsymbol{g},\boldsymbol{f},h)$和$\boldsymbol{\omega}^0$拆分成两部分，分发给$S_0$，$S_1$。

(2) $S_r(r\in\{0,1\})$利用$SSIP(\boldsymbol{v}_0,\boldsymbol{\omega})$，$SSIP(\boldsymbol{v}_j,\boldsymbol{\omega})$计算

$$\langle\boldsymbol{v}_0\cdot\boldsymbol{\omega}^t\rangle_r，\langle\boldsymbol{v}_j\cdot\boldsymbol{\omega}^t\rangle_r. \tag{16}$$

(3) S_r更新模型参数

$$\langle\omega_0^{t+1}\rangle_r=\langle\omega_0^{t}\rangle_r-\frac{\eta}{n}(2\xi_2\langle\boldsymbol{v}_0\cdot\boldsymbol{\omega}^t\rangle_r+\xi_1\langle u_0\rangle_r), \tag{17}$$

$$\langle\omega_j^{t+1}\rangle_r=\langle\omega_j^{t}\rangle_r-\frac{\eta}{n}(2\xi_2\langle\boldsymbol{v}_j\cdot\boldsymbol{\omega}^t\rangle_r+\xi_1\langle u_j\rangle_r), \tag{18}$$

其中$j=1,2,...,d$。

(4) S_r重复执行(2)，(3)直至达到最大迭代次数，最终获得模型参数的共享值$\langle\boldsymbol{\omega}^*\rangle_r$。

算法1给出了全局逻辑回归安全训练的形式化描述。

算法1 全局逻辑回归安全训练算法

输入：用户C_i数据样本集$\{(\boldsymbol{x}_{ik},y_{ik})\}$，其中$i=1,2,...,m$，$k=1,2,...,n_i$.

输出：模型参数$\boldsymbol{\omega}^*$的共享$\langle\boldsymbol{\omega}^*\rangle_0,\langle\boldsymbol{\omega}^*\rangle_1$.

1. C_i：
2. 利用本地样本$(\boldsymbol{x}_{ik},y_{ik})$生成样本矩阵$A_{ik}$.
3. 聚合样本矩阵A_{ik}获得本地局部数据矩阵A_i.
4. 对A_i中的每个元素执行$Shr^A((A_i)_{pq})$，得到本地数据共享矩阵$\langle A_i\rangle_0$，$\langle A_i\rangle_1$.
5. 将$\langle A_i\rangle_0$上传给S_0，$\langle A_i\rangle_1$上传给S_1.
6. *TPA*：
7. 初始化学习率η，最大迭代次数*itermax*.
8. 随机生成乘法三元组$(\boldsymbol{g},\boldsymbol{f},h)$和$\boldsymbol{\omega}^0$，执行共享算法$Shr^A(\cdot)$，将共享值分发给服务器$S_r$.
9. S_r：
10. 根据全局数据共享矩阵$\langle A\rangle_r$得到

$\langle\boldsymbol{v}_0\rangle_r$，$\langle\boldsymbol{v}_j\rangle_r$，$\langle u_0\rangle_r$，$\langle u_j\rangle_r$.

11. $t\leftarrow0$.
12. While $t\le itermax$ do
13. S_r之间执行$SSIP(\boldsymbol{v}_0,\boldsymbol{\omega})$，$SSIP(\boldsymbol{v}_j,\boldsymbol{\omega})$计算(16).
14. 利用（17）和（18）式更新模型参数$\boldsymbol{\omega}$.
15. $t\leftarrow t+1$.
16. End While
17. 返回 $\langle\boldsymbol{\omega}^*\rangle_0,\langle\boldsymbol{\omega}^*\rangle_1$.

4 安全性分析

在SLRT框架中存在两方面的信息需要保护：1) 用户C_i的私有数据；2) 服务器S_r，$r\in\{0,1\}$训练完成的模型参数。如第2.2节所述，敌手主要包括：1) 诚实且好奇的云服务器$\mathcal{A}_1$(Type-I)；2) 诚实且好奇的用户$\mathcal{A}_2$(Type-II)；3)云服务器与部分用户共谋的敌手$\mathcal{A}_3$(Type-III)。在整个训练过程中，*TPA*只负责初始化和分发参数，无法访问用户和模型参数信息。

因此，专注于分析以下潜在的信息泄露情况，证明SLRT框架是安全的。

4.1 数据隐私

4.1.1 抗诚实且好奇的攻击

Type-I 攻击。证明 C_i 的数据不能被敌手 $\mathcal{A}_1$ 获得。用户 C_i 使用共享算法 $Shr^A(\cdot)$ 将整合得到的本地局部数据矩阵 A_i 拆分成 $<A_i>_0$，$<A_i>_1$ 并将其上传给 $\mathcal{A}_1$。对 A_i 的每个元素 $(A_i)_{pq} \in A_i$，算法 $Shr^A((A_i)_{pq})$ 利用 $a_0 \in \mathbb{R}$ 将其拆分为 $<(A_i)_{pq}>_0$，$<(A_i)_{pq}>_1$。由于 a_0 是 C_i 均匀随机选择的，因此 $\mathcal{A}_1$ 不能获得 $(A_i)_{pq}$ 的真实值。

Type-II 攻击。证明某个特定用户 C_i 的数据不能被其他用户（敌手 $\mathcal{A}_2$）获得。在 SLRT 中，训练过程是非交互式的，用户独立地将其私有数据经预处理后上传到云服务器 S_0 和 S_1，随即离线，$\mathcal{A}_2$ 与用户 C_i 之间不存在信息交换，因此无法获得 C_i 的数据。

4.1.2 抗共谋攻击

Type-III 攻击。SLRT 允许云服务器与一些本地客户端共谋。用户 C_i 将数据以秘密共享的形式上传，在威胁模型中假设 S_0，S_1 是非共谋的，因此，$\mathcal{A}_3$ 除了得到的其他用户局部数据矩阵 A_i 的共享 $<A_i>_r$，此外无法获得更多信息，即 SLRT 的数据隐私是抗共谋攻击的。

4.2 模型隐私

4.2.1 抗诚实且好奇的攻击

Type-I 攻击。证明模型参数不能被 $\mathcal{A}_1$ 获得。模型参数 $\boldsymbol{\omega}$ 由 *TPA* 随机初始化为 $\langle\boldsymbol{\omega}\rangle_0$，$\langle\boldsymbol{\omega}\rangle_1$ 分发给 S_0、S_1。在训练阶段，S_r 通过加秘密共享内积协议计算 $\langle\boldsymbol{v}_0\cdot\boldsymbol{\omega}^t\rangle_r$，$\langle\boldsymbol{v}_j\cdot\boldsymbol{\omega}^t\rangle_r$，所需乘法三元组通过 *TPA* 预生成并分发给 S_r。参数更新（17）、（18）式仅包含秘密共享加法和乘法操作，而秘密共享协议是满足 UC (Universal Composability) 安全的[15]，因此，在训练期间，$\mathcal{A}_1$ 无法根据模型参数中间结果或优化结果的共享推断出模型参数。

Type-II 攻击。实际上 SLRT 考虑的是一种完全非交互的情形，训练得到的模型参数不会返回到 C_i。S_0 和 S_1 可以利用以秘密共享的形式存储的模型参数为用户提供预测服务，即 $\mathcal{A}_2$ 不会得到有关模型的任何信息。

4.2.2 抗共谋攻击

Type-III 攻击。模型参数中间结果或优化结果是共享在两台云服务器之间的，即 S_r 拥有 $\langle\boldsymbol{\omega}\rangle_r$ 及其他中间结果记作 $\langle D\rangle_r$，C_i 拥有其本地数据 A_i，显然 S_r 和部分 C_i 共谋，$\mathcal{A}_3$ 也无法根据 $\{A_i,\langle\boldsymbol{\omega}\rangle_r,\langle D\rangle_r\}$ 推断出完整的模型 $\boldsymbol{\omega}$，因此 SLRT 的模型隐私是抗共谋攻击的。

从上述分析可以得出 SLRT 满足系统的安全性需求。

表 1 数据集信息

Table 1 Information of the datasets

数据集	特征数	样本数	正例数	负例数
DD[25]	8	768	500	268
WIBC[26]	9	699	458	241
HDD[27]	13	294	150	120
ACAD[28]	14	690	307	383

5 实验与结果分析

本节从精度、计算开销和通信开销等方面分析和测试 SLRT 的性能。考虑到：

(1) 现有的交互式 PPLR 方案[12,22,29,30]，数据所有者的通信开销随着迭代次数的增加呈线性增长，不同于本文提出的非交互式方案，用户开销保持不变。

(2) 已有的基于秘密共享的 PPLR 方案[15-17]，由于未经数据预处理，在参数更新过程中无法使用固定的信息，需要不经意传输或混淆电路协议，产生较大的通信量和计算负担。

(3) 与本文最相近的工作是 VANE[20]，它是目前最新的支持线性回归、岭回归和逻辑回归的非交互式联邦学习方案，与之前的 PPLR 方案相比效率有较大提升。

因此，以下通过理论分析和数值实验将 SLRT 与 VANE 进行对比。

5.1 实验设置

数值实验在局域网环境下完成，使用 3.20GHz，Intel 4 核 8GB 内存，i5 处理器，搭载 Win10 系统的计算机分别模拟用户和云服务器，并将它们托管在同一区域的局域网上执行训练任务，因此整个实验过程仅存在少量网络延迟。

使用 Python 3.7 对来自 UCI 的 4 个公开的真实数据集 Diabetes Datasets (DD)[25]，Wisconsin Breast cancer data (WIBC)[26]，Heart Disease Data (HDD)[27]，Australian Credit Approval Data (ACAD)[28]分别进行逻辑回归训练来评估 SLRT 和 VANE 的精度和计算开销。数据集的特征数和样本数见表 1。

表 2 SLRT 和 VANE 计算量比较

Table 2 Comparison of computational overhead between SLRT and VANE

参与方	SLRT	VANE
用户	$m\cdot(t_{\text{mat}}+(d+1)^2\cdot t_{\text{add}})$	$m\cdot t_{\text{mat}}+2m\cdot(d+1)^2\cdot(t_{\text{exp}}+t_{\text{mul}})$
云服务器	$l\cdot t_{\text{smul}}$	$(d+1)^2\cdot(t_{\text{inv}}+(m+2)t_{\text{mul}}+t_{\text{exp}})$

5.2 SLRT 和 VANE 理论计算量与通信量对比

仅记录 SLRT 和 VANE 中计算复杂度高的步骤和操作。具体来说，令 t_{mat}、t_{add}、t_{smul}、t_{exp}、t_{inv} 和 t_{mul} 分别表示生成本地局部数据矩阵、加秘密共享、加秘密共享内积、模指数、模逆和模乘运算。使用 m，d，n 和 l 分别表示用户数量、特征维度、训练样本量和迭代次数。

(1) SLRT 和 VANE 计算量对比

在 SLRT 和 VANE 中完整的训练过程包含预处理和服务器训练两个阶段，其中预处理包含用户生成本地局部数据矩阵和数据加密操作(SLRT 中采用秘密共享，VANE 采用部分同态)；训练阶段均不与用户交互，由服务器协同或单独完成。

在 SLRT 中，预处理的所有操作均在明文下进行，包含生成本地局部数据矩阵和对矩阵元素进行 $Shr^A(\cdot)$ 操作，总计算开销为 $m\cdot t_{\text{mat}}+m\cdot(d+1)^2\cdot t_{\text{add}}$。VANE 除了包含生成本地局部数据矩阵的计算开销 $m\cdot t_{\text{mat}}$，代价更大的操作在于使用 Paillier 系统加密本地数据矩阵，计算量为 $2m\cdot(d+1)^2\cdot(t_{\text{exp}}+t_{\text{mul}})$。

服务器训练阶段，VANE 首先利用 Paillier 系统的加同态性，在密文下聚合的来自用户的本地局部数据矩阵，进而解密获得全局训练数据矩阵，计算量为 $(d+1)^2\cdot(t_{\text{inv}}+2t_{\text{mul}}+t_{\text{exp}})$，最后在明文下执行训练，此操作运算量极小，可忽略不计。SLRT 由服务器协同执行加秘密共享内积协议完成模型训练，计算量为 $l\cdot t_{\text{smul}}$。

表 2 汇总了用户和服务器在 SLRT 和 VANE 下的理论计算量。由于加秘密共享操作的时间远小于 Paillier 密码系统加密数据的时间；SLRT 中服务器在明文下执行加秘密共享内积协议的时间远小于 VANE 中使用 Paillier 密码系统聚合和解密数据矩阵的时间，因而无论在预处理阶段还是训练阶段，SLRT 的效率都优于 VANE 方案，在数值实验中可进一步得到验证。

(2) SLRT 和 VANE 通信量对比

表 3 不同数据集的模型精度

Table 3 Model accuracy of different datasets

Dataset	Number of iterations	Precision(%)			Recall(%)		
		SLRT	VANE	basline	SLRT	VANE	baseline
DD	2×10^3	78.2	77.6	78.5	78.3	78.0	78.8
WIBC	2×10^3	97.5	98.7	98.8	96.8	98.7	98.7
HDD	10^3	97.5	97.4	97.5	96.0	95.1	97.7
ACAD	10^3	97.4	97.4	97.6	98.4	97.7	98.5

在 VANE 中，用户经预处理获得的本地局部数据矩阵的维度是 $(d+1)\times(d+1)$，使用 Paillier 系统对矩阵中每个元素进行加密并上传给云服务器。每个密文的长度为 2κ，其中 κ 是安全参数，一般取 1024 或 2048，VANE 的通信量为 $2m\cdot(d+1)^2\cdot\kappa$。

SLRT 中用户经预处理同样获得 $(d+1)^2$ 个元素的本地局部数据矩阵，拆分成两部分后上传给 S_0 和 S_1。在训练阶段，每次迭代过程中 S_0，S_1 执行 $SSIP(\cdot)$，需要交互传输 2 个 $(d+1)$ 维的向量，整个训练阶段传输数据个数为 $2l\cdot(d+1)$。因此 SLRT 的总通信量为 $2(d+1)^2+2l\cdot(d+1)$ 个明文数据。由于传输一个明文数据所需通信量远远小于一个 Paillier 密码系统的密文数据，迭代次数 l 通常在数千次以内，SLRT 的通信开销优于 VANE。

5.3 精度测试

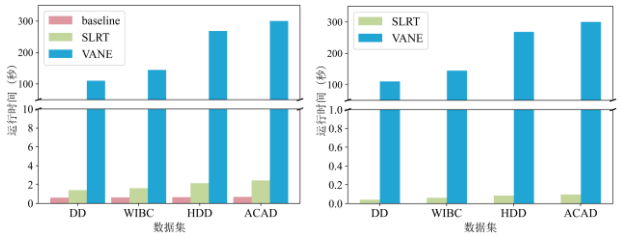
精准率(Precision)和召回率(Recall)为评价机器学习模型精度的常用指标。其中，精准率(Precision)表示正确预测为正的占全部预测为正的比例，召回率(Recall)表示正确预测为正的占全部实际为正的比例。使用 5-折交叉验证法，如表 3 所示，SLRT 的模型精度与 VANE 相当。同时，给出了原始模型 (1) 在非隐私保护下的训练结果 baseline，发现 SLRT 的模型精度也与明文下的结果相当。

5.4 性能评估

为测试两种方案在实际机器上的运行效率，记录了 SLRT 和 VANE 的预处理时间、总训练时间，并评估样本数、用户数、迭代次数的变化对两种方案的影响。

(1) 原始数据集下总训练时间的对比

图 3(a)比较了原始数据集下 SLRT 和 VANE 进行逻辑回归训练的总时间，包括本地预处理时间(图 3 (b))与服务器训练时间。SLRT 的效率比 VANE 提高了至



(a) 总时间　　(b) 预处理时间

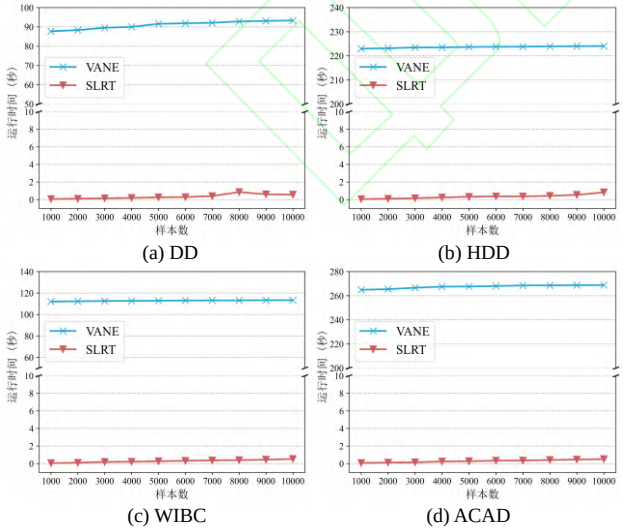
图 3 原始数据集下SLRT与VANE总训练时间和预处理时间对比（$l=1000, m=10$）

Fig.3 Comparison of total training time and preprocessing time between SLRT and VANE ($l=1000$, $m=10$)

少 10^2 倍。主要原因在于：在迭代次数 l 和用户数量 m 相同的情况下，影响两种方案的重要因素在于使用的密码技术不同，SLRT 中的主要操作为明文下执行的 $Shr^A(\cdot)$ 和 $SSIP(\cdot)$，其计算量远小于 VANE 中 Paillier 系统加密、聚合和解密数据矩阵。

(2) SLRT 与 VANE 的预处理时间随样本数量变化对比

SLRT 与 VANE 均包含有预处理阶段，对原始数据集进行复制使样本数量达到 104 个，以观察样本数量对两种方案预处理时间的影响。从图 4(a)-(d)中可以看出，随着训练样本数量的增加，SLRT 和 VANE 的预处理时间几乎没有变化。然而，SLRT 仅花费 0.56s 到 0.82s 便可完成 4 个数据集的预处理任务，较 VANE 提高了 80~120 倍。主要原因是：SLRT 需要拆分的样本矩阵大小仅与样本维度 d 有关，VANE 需要加密的数据量也仅与 d 有关，这两个操作均与样本个数 n 无关。在 Paillier 系统加密操作时

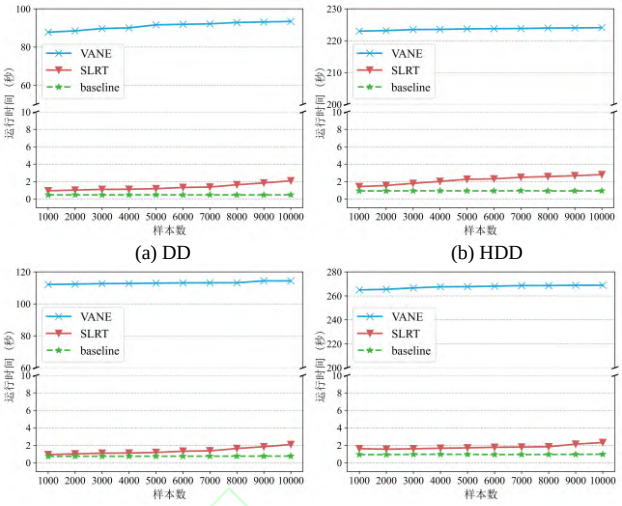


(a) DD　　(b) HDD

(c) WIBC　　(d) ACAD

图 4 SLRT 与 VANE 预处理时间随样本数量变化对比（$l=1000, m=10$）

Fig.4 Comparison of preprocessing time of SLRT and VANE with the number of samples ($l=1000$, $m=10$)



(a) DD　　(b) HDD

(c) WIBC　　(d) ACAD

图 5 总训练时间随样本数量变化对比（$l=1000, m=10$）

Fig.5 Comparison of total training time with the number of samples ($l=1000$, $m=10$)

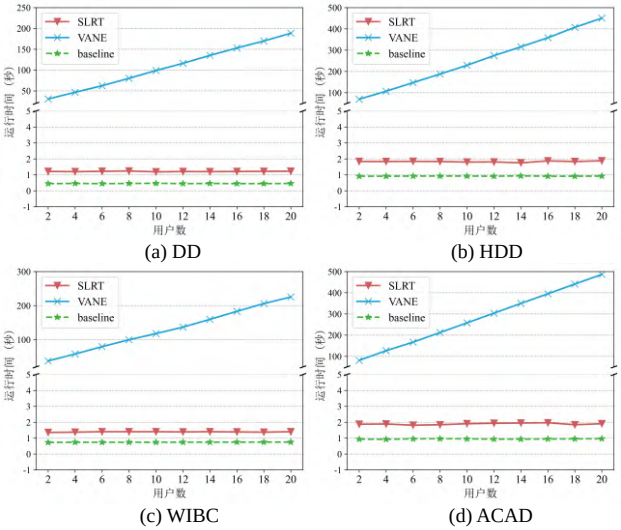
间远大于共享算法 $Shr^A(\cdot)$ 的情形下，SLRT 的预处理时间显著低于 VANE。

(3) 总训练时间随样本数量变化对比

图 5(a)-(d)展示了样本数 n 从 10^3 增加到 10^4 时，方案总训练时间的对比，可以看出随着样本数量的增加，SLRT 和 VANE 总训练时间几乎没有影响，原因在于这两种方案都经过了预处理，在服务器训练阶段无需单个样本参与，因而训练时间与样本数无关。由于 SLRT 在预处理阶段表现得比 VANE 好，训练阶段也没有 VANE 方案中复杂度较高的聚合解密等操作，在总训练时间上优于 VANE。

(4) 总训练时间随用户数量变化对比

SLRT 和 VANE 都是针对水平分布的数据集，



(a) DD　　(b) HDD

(c) WIBC　　(d) ACAD

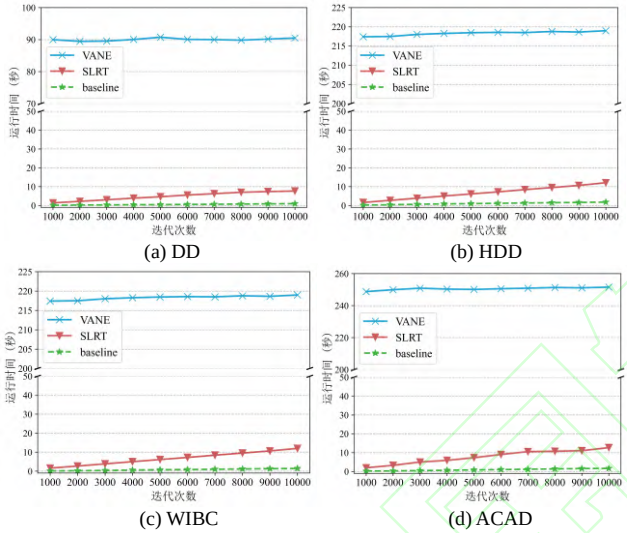
图 6 总训练时间随用户数变化对比（$l=1000, m=10$）

Fig.6 Comparison of total training time variation with the number of clients($l=1000$, $m=10$)

为考察用户数量增加对训练时间的影响，本文给出了图 6(a)-(d)，当从 2 增加到 20 时，VANE 的时间随用户数线性增长，而 SLRT 几乎不受影响。原因在于：VANE 每增加一个用户，都需要在 Paillier 系统下增加 $(d+1)^2$ 个数据的加密和聚合操作，而 SLRT 每增加一个用户，增加的操作仅为明文下执行的共享算法，因而不受用户数量的影响。

(5) 总训练时间随迭代次数变化对比

SLRT 和 VANE 都是非交互式的方案，迭代训练过程都是在明文下进行，不同之处是 VANE 由一个服务器完成训练，而 SLRT 由两个服务器协同完成。从图 7(a)-(d)可以看出，随着迭代次数的增加，



(a) DD (b) HDD

(c) WIBC (d) ACAD

图 7 总训练时间随迭代次数变化对比($m=10$, $n=10000$)

Fig.7 Comparison of total training time variation with the number of iterations ($m=10$, $n=10000$)

VANE 方案总训练时间几乎不受影响，而 SLRT 方案略有增加，但比起 VANE 在效率上还是有明显优势。原因在于 VANE 在服务器训练之前需要花费大量时间进行加密、聚合和解密运算。

5.5 多种方案功能对比

本节对 SLRT 和几种流行的 logistic 回归隐私保护方案在功能上进行对比，可以看出，当数据分布式地存储在各个用户中时，SLRT 方案在保证数据信息、模型参数隐私的同时，仍能保持高精度和高性能，见表 4。

6 结束语

本文提出了一种高效的非交互式逻辑回归隐私保护训练方案，基于“良可分离结构”，结合逻辑损失函数的二阶多项式替换策略，构建新的梯度计算格式，保证用户将数据进行预处理并以秘密共享方式上传给服务器后即可离线。与基于同态的交互式方案相比，极大地降低了用户的通信量和计算量；与未经预处理的秘密共享方案相比，减少了服务器之间的交互；与最新的非交互式方案 VANE 相比，效率上至少提高 10^2 倍。因此，SLRT 方案适合于分布式、高维度、多用户、大样本场景下的隐私保护模型训练。下一步，将针对垂直分区数据集上的逻辑回归隐私保护问题展开研究，并在时间及通信成本方面做进一步优化。

表 4 多种隐私保护方案功能对比

Table 4 Comparison of functions of various schemes

方案	高效率		高精度	低通信量	不需硬件支持
	中等规模数据集	大规模数据集			
Our SLRT	✓	✓	✓	✓	✓
VANE[20]	✓	✗	✓	✓	✓
PPLRA[9]	✓	✗	✓	✓	✓
SecureML[15]	✓	✓	✓	✗	✓
SecureLR[29]	✓	✗	✓	✓	✗

参考文献

[1] PALLONETTO F, DE ROSA M, MILANO F, et al. Demand response algorithms for smart-grid ready residential buildings using machine learning models[J]. Applied Energy, 2019, 239: 1265-1282.

[2] WANG W, ZHENG H S, WU Y C. Prediction of fundraising outcomes for crowdfunding projects based on deep learning: a multi model comparative study[J]. Soft Computing, 2020, 24(11): 8323-8341.

[3] REHOUMA R, BUCHERT M, CHEN Y P. Machine learning for medical imaging-based COVID-19 detection and diagnosis[J]. International Journal of Intelligent Systems, 2021, 36(9): 5085-5115.

[4] HOOSHMAND A. Accurate diagnosis of prostate cancer using logistic regression[J]. Open Medicine, 2021, 16(1): 459-463.

[5] CHOWANDA A, SUTOYO R, MEILIAN A, et al. Exploring text-based emotions recognition machine learning techniques on social media conversation[J]. Procedia Computer Science, 2021, 179(1): 821-828.

[6] CVITIC I, PERAKOVIC D, PERISA M, et al. Ensemble

machine learning approach for classification of IoT devices in smart home[J]. International Journal of Machine Learning and Cybernetics, 2021, 12(11): 3179-3202.

[7] HOU R, KONG Y Q, CAI B, et al. Unstructured big data analysis algorithm and simulation of internet of things based on machine learning[J]. Neural Computing and Applications, 2020, 32(10): 5399-5407.

[8] GUO W, SHAO J, LU R X, et al. A privacy-preserving online medical prediagnosis scheme for cloud environment[J]. IEEE Access, 2018, 6: 48946-48957.

[9] FAN Y K, BAI J R, LEI X, et al. Privacy preserving based logistic regression on big data[J]. Journal of Network and Computer Applications, 2020, 171: 102769.

[10] CHEN H, GILAD-BACHRACH R, HAN K, et al. logistic regression over encrypted data from fully homomorphic encryption[J]. BMC Medical Genomics, 2018, 11(4): 3-12.

[11] 许心炜，蔡斌，向宏等. 基于同态加密的多分类逻辑回归模型[J]. 密码学报, 2020, 7(2): 179-186.
XU X W, CAI B, XIANG H, et al. Multinomial logistic regression model based on homomorphic encryption[J]. Journal of Cryptologic Research, 2020, 7(2): 179-186. (in Chinese)

[12] 宋蕾，马春光，段广晗等. 基于数据纵向分布的隐私保护逻辑回归[J]. 计算机研究与发展，2019，56(10): 2243-2249.
SONG L, MA C G, DUAN G H, et al. Privacy-preserving logistic regression on vertically partitioned data[J]. Journal of Computer Research and Development,2019,56(10): 2243-2249. (in Chinese)

[13] 郭娟娟，王琼霄，许新等. 安全多方计算及其在机器学习中的应用[J]. 计算机研究与发展，2021，58(10): 2163-2186.
GUO J J, WANG Q X, XU X, et al. Secure multiparty computation and application in machine learning[J]. Journal of Computer Research and Development, 2021, 58(10): 2163-2186. (in Chinese)

[14] GUYON I, LI J W, MADER T, et al. Competitive baseline methods set new standards for the NIPS 2003 feature selection benchmark[J]. Pattern Recognition Letters, 2007, 28(12):1438-1444. http://archive.ics.uci.edu/ml/datasets/Arcene

[15] MOHASSEL P, ZHANG Y. SecureML: A system for scalable privacy-preserving machine learning[C]//2017 IEEE symposium on security and privacy (S&P), IEEE, 2017: 19-38.

[16] DE COCK M, DOWSLEY R, NASCIMENTO A C A, et al. High performance logistic regression for privacy-preserving genome analysis[J]. BMC Medical Genomics, 2021, 14(1): 1-18.

[17] 郑云涛，叶家炜. 基于 OT 协议的 FATE 联邦迁移学习方案[J/OL]. 计算机工程. https://doi.org/10.19678/j.issn.1000-3428.0064452
ZHENG Y T, YE J W. FATE federated transfer learning scheme based on OT protocol[J/OL]. Computer Engineering. (in Chinese) https://doi.org/10.19678/j.issn.1000-3428.0064452

[18] LI T, LI J, CHEN X F, et al. NPMML: A framework for non-interactive privacy-preserving multi-party machine learning[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 18(6): 2969-2982.

[19] MA X, CHEN X F, Zhang X Y. Non-interactive privacy-preserving neural network prediction[J]. Information Sciences, 2019, 481: 507-519.

[20] WANG F W, ZHU H, LU R X, et al. A privacy-preserving and non-interactive federated learning scheme for regression training with gradient descent[J]. Information Sciences, 2021, 552: 183-200.

[21] PREGIBON D. logistic regression diagnostics[J]. The annals of statistics, 1981, 9(4): 705-724.

[22] HARDY S, HENECKA W, IVEY-LAW H, et al. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption[J]. arXiv preprint arXiv: 1711.10677, 2017.

[23] Beaver D. Commodity-based cryptography[C]//Proceedings of The Twenty-ninth Annual ACM Symposium on Theory of Computing. 1997: 446-455.

[24] KINCAID D, CHENEY W. Numerical Analysis: Mathematics of Scientific Computing[M]. 3th ed. China Machine Press, Beijing, China, 2003.

[25] BENNETT P H, BURCH T A, MILLER M. Diabetes mellitus in American (Pima) Indians[J]. The Lancet, 1971, 298(7716): 125-128. https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes

[26] MANGASARIAN O L, SETIONO R, WOLBERG W H. Pattern recognition via linear programming: Theory and Application to Medical Diagnosis[J]. 1990. http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

[27] DETRANO R, JANOSI A, STEINBRUNN W, et al. International application of a new probability algorithm for the diagnosis of coronary artery disease[J]. The American Journal of Cardiology, 1989, 64(5): 304-310. http://archive.ics.uci.edu/ml/datasets/Heart+Disease

[28] Quinlan J R. Simplifying decision trees[J]. International Journal of Man-machine Studies, 1987, 27(3): 221-234. http://archive.ics.uci.edu/ml/datasets/Credit+Approval

[29] JIANG Y C, HAMER J, WANG C H, et al. SecureLR: Secure logistic regression model via a hybrid cryptographic protocol [J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2018, 16(1): 113-123.

[30] XU R, BARACALDO N, ZHOU Y, et al. FedV: Privacy-preserving federated learning over vertically partitioned data[C]//Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security. 2021: 181-192.