软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007636] [CSTR: 32375.14.jos.007636]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于隐私保护分组生成的高效联邦学习*

周众泽[1,2], 张 俊[1,2], 寿黎但[1,2], 陈 珂[1,2], 陈 刚[1,2]

[1](区块链与数据安全全国重点实验室 (浙江大学), 浙江 杭州 310027)

[2](杭州高新区 (滨江) 区块链与数据安全研究院, 浙江 杭州 310052)

通信作者: 寿黎但, E-mail: should@zju.edu.cn

摘 要: 联邦学习是目前安全多方机器学习领域最为先进的技术, 其因为解决了数据不离开本地的多节点联合模型训练问题而得到了广泛的关注. 然而, 现实中客户端数据的非独立同分布 (Non-IID) 会显著降低全局模型的性能. 现有的主流方法一类聚焦于在训练时校正模型的参数偏差, 但其在数据分布越极端的情况下效果越差. 另一类方法旨在本地训练前校正数据分布的偏差, 但其会有隐私泄露的风险. 提出一种框架, 遵循严格的联邦学习隐私保护要求, 利用差分隐私、安全多方计算等技术, 在客户端数据分布保持黑盒的前提下, 选择出对全局模型性能提升最有益的若干标签, 针对这些标签生成高质量合成数据并分发, 使补齐这些数据后的客户端本地数据分布趋向于IID, 从而大大提高全局模型的表现. 具体来说, 首先设计一个隐私安全的算法, 对标签在客户端上的分布进行捕获, 并依据该标签分布特征对所有客户端进行组别的划分. 随后挑选出最值得生成的标签, 在标签对应的组内协调客户端合作训练高质量全局生成模型存放于服务器. 最后在任务模型的训练阶段, 基于标签分布特征利用这些生成模型有选择地合成样本并分发, 来使本地数据分布更加均匀, 进而缩小本地模型差距, 在聚合后得到高质量的全局模型. 实验表明, 所提方法能够有效提高全局模型在测试集上的准确率, 且能减少全局模型收敛前所需要的联邦学习通信轮次, 其有效性超越了各基线方法, 在不同的数据集上得到了验证.

关键词: 联邦学习; 非独立同分布; 隐私保护

中图法分类号: TP311

中文引用格式: 周众泽, 张俊, 寿黎但, 陈珂, 陈刚. 基于隐私保护分组生成的高效联邦学习. 软件学报. http://www.jos.org.cn/1000-9825/7636.htm

英文引用格式: Zhou ZZ, Zhang J, Shou LD, Chen K, Chen G. Efficient Federated Learning with Privacy-preserving Grouping and Generation. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7636.htm

Efficient Federated Learning with Privacy-preserving Grouping and Generation

ZHOU Zhong-Ze[1,2], ZHANG Jun[1,2], SHOU Li-Dan[1,2], CHEN Ke[1,2], CHEN Gang[1,2]

[1](State Key Laboratory of Blockchain and Data Security (Zhejiang University), Hangzhou 310027, China)

[2](Hangzhou High-tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou 310052, China)

Abstract: Federated learning is currently the most advanced technology in secure multi-party machine learning and has attracted significant attention for its ability to enable collaborative model training across multiple nodes without requiring data to leave local devices. However, in real-world scenarios, client data are often non-independent and identically distributed (Non-IID), which significantly degrades the performance of the global model. Current mainstream methods primarily focus on correcting model parameter biases during training, but their effectiveness diminishes as data distribution becomes more extreme. Another category of methods aims to correct data distribution biases before local training, but this approach poses privacy risks. This study proposes a framework that adheres to strict federated learning privacy protection requirements and utilizes techniques such as differential privacy and secure multi-party computation. Assuming that client data distributions remain black boxes, the most beneficial labels for improving global model performance are selected.

* 基金项目: 浙江省尖兵研发攻关计划 (2024C01021); 浙江省科技创新领军人才计划 (2023R5214)
收稿时间: 2024-05-10; 修改时间: 2025-03-13; 采用时间: 2026-01-19; jos 在线出版时间: 2026-05-20

High-quality synthetic data are generated for these labels and distributed to clients, thus making the local data distribution closer to IID and significantly enhancing the performance of the global model. Specifically, a privacy-preserving algorithm is first designed to capture the distribution of labels across clients, and clients are grouped according to these label distribution characteristics. Then, the most worthwhile labels for generation are selected, and within the corresponding groups, clients are coordinated to collaboratively train high-quality global generative models, which are stored on the server. During the training phase of the task model, based on label distribution characteristics, samples are selectively synthesized using these generative models and distributed to clients, resulting in a more uniform local data distribution. This process reduces local model disparities and yields a high-quality global model after aggregation. Experimental results demonstrate that the proposed method can effectively improve the accuracy of the global model on the test set and reduce the number of federated learning communication rounds required before convergence. Its effectiveness surpasses that of baseline methods and has been validated on different datasets.

Key words: federated learning; non-independent and identically distributed (Non-IID); privacy protection

人工智能的发展离不开大数据的积累, 然而现有的数据孤岛问题使数据难以交流整合. 为了解决人工智能中存在的数据孤岛问题, 谷歌在 2016 年提出了联邦学习框架[1], 该框架通过全局服务器对多个客户端本地训练的模型参数迭代聚合从而进行模型优化. 与中心化的机器学习不同, 联邦学习不暴露客户端的原始数据, 可有效地保护数据隐私, 因此得到广泛关注.

然而, 实际情况下客户端的数据往往是异质的, 即非独立同分布 (non-independent and identically distributed, Non-IID), 包括特征分布、标签分布和数据数量的偏移 (skew). 先前的研究[2] 表明在数据异质的联邦中训练模型, 将对模型性能产生严重的负面影响. 其中, 标签分布偏移 (label distribution skew) 的影响尤为显著, 每个客户端持有的标签类别数越少、越不均衡, 全局模型的收敛越慢且准确度越低. 本文重点关注这一类问题.

图 1 可视化地展示了 Non-IID 现象对联邦学习中全局模型性能影响的成因, 以及当前的研究现状. 其中 w^t 是每轮服务器下发的全局模型 (也是每轮本地模型训练的起点), 客户端向着不同的方向进行更新得到本地模型 w_1^t 和 w_2^t, 这些本地模型在服务器上聚合得到新一轮的全局模型 w^{t+1}. 如图 1(a) 所示, 在联邦学习的 IID 场景中, 每个客户端的本地模型的优化目标 (图中黄色圆形) 很靠近全局模型的最优点 (图中黄色五边形), 因此相互接近的本地模型聚合得到的全局模型性能较好. 而如图 1(b) 所示, 在 Non-IID 的场景中, 由于每个客户端的本地数据分布显著的区别于全局的数据分布, 每个客户端的本地优化目标偏离全局最优点 (即服务器的优化目标), 这种现象被称为客户端漂移 (drift)[2]. 这导致在本地训练阶段, 本地模型将收敛到一个远离全局最优的点 (图中蓝色圆形). 这种差异使由所有客户端通过加权平均聚合产生的全局模型 (图中蓝色五边形) 与全局最优点 (图中黄色五边形) 产生了显著的偏离 (即②标识的实际上得到的全局模型与理论上最优点的距离远大于①). 因此, 在 Non-IID 场景下收敛后的全局模型的准确率明显低于 IID 场景.

为了减轻 Non-IID 对模型性能的影响, 现有的方法可分为分布校正和参数校正两类. 分布校正发生在学习过程之前, 通过将非原始数据引入客户端来减轻数据的统计偏差, 使数据分布更加均匀、更接近全局分布. 如图 1(c) 所示, 分布校正调整本地最优点使其更接近全局最优点, 从而使模型在本地训练期间更有效地收敛. 参数校正发生在学习过程中, 其通过从全局模型中提取其他客户端本地模型的优化方向等知识, 利用这些知识在本地训练的过程中约束参数更新, 使聚合后的全局模型更靠近最优点. 如图 1(d) 所示, 参数校正在客户端之间规范优化方向, 使本地模型参数更新更加一致.

然而, 这些方法要么存在隐私泄露的风险, 要么在极端的数据异质场景下效果并不理想. 对于分布校正的方法, 现有方法需要暴露部分原始的数据 (FAug[3] 等) 或者数据分布 (Astraea[4] 等) 给服务器, 这在高度隐私敏感的现实场景是不可取的. 对于参数校正的方法, 尽管本地训练时模型参数都向着互相靠近的方向被优化, 但由于起始各客户端的模型在非同质数据集上进行本地训练导致参数差异明显, 且每一轮中作为基准的上一轮的全局模型参数可能都远远偏离了全局最优, 故其只能缓慢校正客户端的参数 (这导致④标识的距离大于③), 且无法保证本地模型优化时是向着全局最优的方向进行. 现有的方法 (FedProx[5]、MOON[6] 等) 在极端的数据异质的场景下表现得都不尽如人意.

为了提高数据极端异质场景下联邦学习的性能, 我们从分布校正出发应对数据 Non-IID 挑战. 分布校正的核

心是通过补齐客户端的缺失/稀少类数据来让客户端数据尽可能服从独立同分布, 进而使本地模型的优化方向尽可能一致. 然而, 为了保护用户的隐私, 在联邦学习的场景下客户端的原始数据是不允许离开本地的, 所以客户端间直接共享自身的多数类数据来校正数据分布并不可行. 现有的工作[3,7]提出客户端利用自身本地数据去训练一个生成网络去产生合成数据, 这些合成数据既保留了原始数据特征, 又无法逆向还原出原始数据. 然而, 由单个客户端训练得到的模型, 由于其训练数据均来自本地, 该模型容易暴露客户端的数据分布情况. 例如若某生成网络生成了大量的袋鼠样本, 则表明训练该网络的客户端拥有大量的袋鼠数据, 而袋鼠是仅在大洋洲存在的物种, 那么可以进一步缩小该客户端所归属的机构范围, 这在很多隐私敏感的场景是客户端不希望被知晓的信息. 因此, 客户端训练生成网络的过程应该“融于群体”, 即多个客户端协同训练一个生成模型, 这样得到的模型才不会直接暴露作为个体的客户端的样本分布信息. 这样一方面保证了无法从协同训练得到的网络的输出特征推断每个客户端的数据特征, 另一方面也利用了这些客户端的本地数据扩大样本空间来提高该生成网络的质量, 这与联邦学习的初衷不谋而合. 但注意到, 此时全局上数据分布处于 Non-IID, 若所有的客户端一同训练这个生成模型反而会降低该生成模型质量. 以上促使我们探索: 是否可设计一个框架, 在保护客户端具体数据分布的前提下, 围绕某个共性对客户端进行隐私化分组, 在组内协调客户端训练高质量的生成模型, 最后合理地分发这些生成模型合成数据来补齐客户端的数据分布?

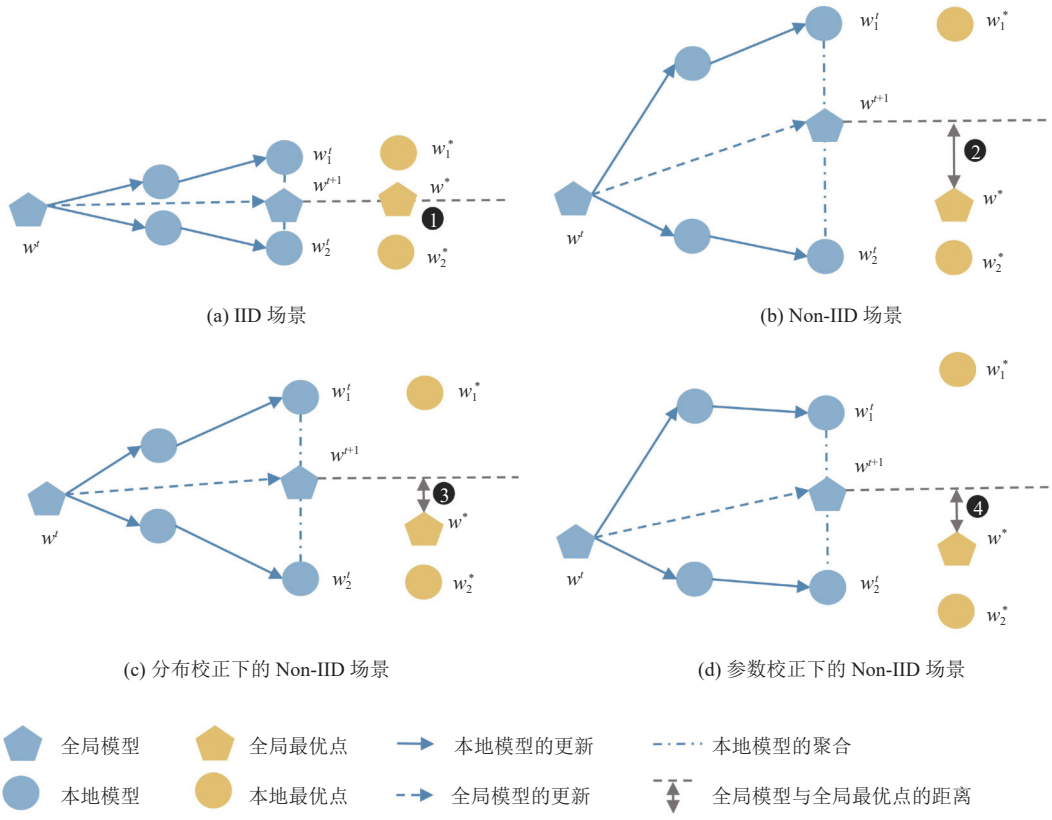


图 1 联邦学习中 Non-IID 现象和解决办法的可视化展现

为此我们提出了一种能有效应对极端数据非匀质问题的隐私安全的联邦学习框架 GGC, 包含 3 个阶段: 分布隐私捕获并分组 (distribution-aware privacy-preserving grouping, DPG)、组内联邦生成模型训练 (intra-group federated generative model training, IGT)、数据补齐下的任务联邦学习 (federated learning with data complement, FLC). GGC 以标签作为共性, 隐私地捕获各个客户端一定的分布特征. 依据这些特征对客户端进行分组, 将具有高

质量同类样本的客户端集中在一组, 随后在组内协调客户端仅针对该类样本训练一个全局生成模型. 最后在任务模型的本地训练前利用这些生成模型生成合成样本, 基于分布特征有选择地补齐各客户端的数据分布, 使本地任务模型优化时更靠近全局最优点. GGC 区别于现有其他工作的是: (1) 关注到了对客户端数据分布的保护, 在捕获分布特征时引入了多方安全计算和差分隐私等相关技术加强隐私保护. (2) 包含一种标签选择策略来帮助服务器确定哪些标签被合成并分发收益最大, 以便更好地补齐客户端数据分布的同时可以控制开销. (3) 尽可能地利用所有客户端的集体数据来扩大样本空间, 以得到可靠且高质量的生成模型. (4) 基于分布特征有选择地为客户端补齐数据分布, 并平衡了合成样本与原始样本的数目比例以在任务模型的联邦学习阶段加速收敛. (5) GGC 与主流基线方法兼容, 在某些场景下同时采用二者可以进一步提高全局模型的性能, 我们在正交实验中证明了这一点. (6) 性能更好. 在实验章节中进行了验证, GGC 对比其他现有方法取得了更好的效果.

本文第 1 节介绍应对联邦学习 Non-IID 问题的相关方法和研究现状. 第 2 节介绍本文所需的基础知识以及提出问题定义. 第 3 节对本文提出的方法进行详细的说明. 第 4 节围绕 7 个问题通过多种实验验证本文方法的有效性. 最后总结全文并对不足之处进行展望.

1 相关工作

联邦学习的基础算法 FedAvg[1] 在聚合时对客户端的本地模型的参数或梯度进行加权平均, 其权重由每个客户端本地的样本数量决定. 由于基于样本量的加权聚合只关注到了客户端在不同样本量上的本地训练对于全局模型参数贡献的权重, 而没有关注到由于本地数据分布的不同导致的各个本地模型参数之间的差异化, 因此该方法聚合得到的全局模型在非独立同分布 (Non-IID) 数据分布的场景下仍难以收敛到全局最优. 有很多工作着眼于此, 可以分为在联邦学习之前对数据进行分布校正和在联邦学习过程中对模型进行参数校正两种.

1.1 分布校正——联邦学习过程前

分布校正致力于在本地训练前调整各客户端的数据分布使其更接近全局分布, 来降低本地训练过程中的客户端漂移现象.

一类方法通过调度, 将不同客户端分为不同集群, 旨在集群内解决数据 Non-IID 问题. 如 Astraea[4], 其要求客户端向服务器发送本地数据分布情况, 服务器将能相互弥补数据分布的客户端放在一个集群内, 即将集群内的数据分布变为 IID, 之后集群内的每个客户端依次异步更新全局模型后上传到集群代理服务器, 最后再聚合每个集群代理服务器上的模型获得全局模型. PFA[8]从各客户端本地训练的模型的卷积层提取通道参数, 计算它们的稀疏度作为特征将不同客户端分为不同的集群, 但 PFA 需要提前指定聚类数量这个重要参数, 这在现实场景中很难做到. 总的来说, 这类方法存在集群间协作不足、暴露客户端的数据分布、高通信和集群管理成本的问题. FedConcat[9] 提出每个客户端发送本地数据分布信息, 服务器用聚类算法对客户端进行聚类, 每个类内的客户端联邦学习训练特征提取层, 在服务器上重新训练分类层以克服数据非独立同分布的影响.

另一类方法致力于从原始数据中抽取特征得到合成数据, 通过这些合成数据参与训练来缓解各方的数据分布差异导致的本地模型的差异. 如 FAug[3]提出客户端上传少量本地数据于服务器作为种子, 服务器联网搜索相似样本构建数据集, 再利用该数据集在服务器上训练生成网络. 其问题在于发送客户端样本违背了联邦学习的原则, 网络搜集样本也无法保证质量. SDA-FL[7]在每个客户端本地训练一个生成模型来生成合成数据, 这些合成数据会在服务器上被收集并打上伪标签参与之后的本地训练. 其弱点在于伪标签不够准确会影响本地模型的性能, 同时有限的训练样本导致客户端单独训练的生成模型生成的样本质量偏低. 此外, FedHome[10]简单地将若干客户端组成联邦, 训练一个生成式卷积自编码器 (generative convolutional autoencoder, GCAE), 然而在 Non-IID 情况下训练出的生成模型往往质量欠佳. FedMix[11]提出客户端在本地对样本和标签做平均 (Mixup), 将平均后的数据发送于服务器用于数据增强. 但该方法仍有很高的隐私泄露风险问题, 混合后的数据很容易暴露原数据、原标签或本地数据分布的情况. FedDPGAN[12] 针对二分类问题提出通过 GAN 生成样本, 但其只能生成一类数据, 且只考虑保护样本隐私, 并未考虑保护数据分布隐私.

1.2 参数校正——联邦学习过程中

参数校正旨在客户端本地训练或服务器聚合时通过上一轮模型参数等辅助信息校正本地或全局模型的参数, 使模型更接近于全局最优, 或限制客户端参数的更新幅度, 进而使聚合后的全局模型更接近全局最优.

如 FedProx[5], 其通过在本地模型训练的损失函数中添加了一个正则项, 使本地模型更新时不仅着眼于对任务的优化还要缩小与上一轮全局模型的距离. MOON[6]受到对比学习的启发, 利用模型表示的相似性校正客户端的本地训练, 其通过最大化本地模型和全局模型的相似性, 同时最小化本地模型与先前本地模型的相似性, 来减轻数据异质性带来的影响. FedNova[13]的核心思想是采用归一化平均方法消除目标间的不一致性. 其在本地训练时对每个客户端的更新步长进行归一化, 使得每个客户端的贡献与其计算量成正比, 同时允许客户端在不同的本地计算轮数后提交更新. 这种方法确保了每个客户端的更新对全局模型的影响是均衡的, 避免了某些客户端由于数据分布差异而过度影响全局模型. FedAvgM[14]在服务器更新参数时添加了动量, 期望全局模型聚合时更容易摆脱本地最优达到全局最优. FEDOPT[15]借鉴了梯度下降中的参数优化方法, 指出可以在联邦学习中采用自适应的服务器优化来提高收敛性, 其核心思想是算法需要在聚合时引入参数来约束那些梯度比较大而且震荡的方向, 同时要保证有益的方向尽量不会被惩罚. SCAFFOLD[16]在每个客户端和服务器端都引入了控制变量用于校正每轮本地训练中的差异. 其中每个客户端维护一个本地控制变量, 用于纠正本地梯度更新, 服务器维护一个全局控制变量, 用于在客户端上传更新时进行全局校正. FedUV[17]可以直接促使本地模型模仿 IID 环境, 而不是依赖于全局模型来进行正则化, 其专注于分类器方差和超球均匀性这两个正则化项来提高本地模型的表现. FedFTG[18]是一种与主流方法兼容的联邦学习方法, 其使用一种无数据的知识蒸馏方法利用本地模型的知识微调全局模型.

然而在极端的数据分布情形下, 由于初始聚合的全局模型都无法保证是接近于全局最优的, 并且各客户端数据分布差异巨大, 导致训练过程中本地模型的参数校正效果并不理想, 进而影响到了聚合得到的全局模型的表现.

与上述方法不同, 本文提出的框架 GGC 不仅高效解决了联邦学习中 Non-IID 造成的全局模型性能下降问题, 且更关注各客户端本身数据分布的隐私安全.

2 基础知识及问题定义

GGC 框架主要基于联邦学习、生成对抗网络和隐私保护手段. 下面对相关概念和基础知识进行介绍, 最后做出问题定义, 表 1 列出了本文的常用记号.

表 1 本文常用记号

符号	含义
R/r	总通信轮次/第 r 轮
E/e	本地训练的总迭代轮次/第 e 轮
K/k	客户端总数/第 k 个客户端
C/c	标签类别总数/第 c 个类
$\{\boldsymbol{x}^c, y^c\}$	标签为 c 的样本
N_k	客户端 k 的样本总数
$\mathcal{D}_k$	客户端 k 上的本地数据集
$\mathcal{D}_{\text{test}}$	服务器上存储的测试数据集
$\boldsymbol{\omega}^r$	第 r 轮的全局任务模型
$\boldsymbol{\omega}_k^{r+1}$	在客户端 k 上迭代 r 轮后的本地任务模型
$\boldsymbol{\theta}$	生成模型
B	批大小即一次本地参数更新中用于训练的样本数量
η	学习率即本地每次参数更新的速度

2.1 联邦学习

联邦学习 (federated learning, FL) 概念的提出最初是为了解决手机终端用户数据的模型训练问题, 目前该技术

主要用于解决数据在不共享情况下的联合模型训练问题. 联邦学习系统通常由一个联邦服务器和多个客户端(参与者) 组成, 每个客户端都持有一定的本地数据集, 在联邦学习中使用本地数据集训练本地模型并上传至服务器, 服务器接收到各客户端的本地模型参数后进行聚合操作并更新全局模型. 联邦学习的具体训练过程如下.

在每个训练轮次 $r=0,1,\ldots,R-1$ 中, 由服务器确定的集合 $\mathcal{S}_r$ 中的每个客户端下载全局模型 $\boldsymbol{\omega}^r$, 将其参数复制给本地模型 $\boldsymbol{\omega}_k^r \leftarrow \boldsymbol{\omega}^r$, 通过随机梯度下降 (SGD) 最小化损失函数 $\ell(\cdot)$, 在本地数据集 $\mathcal{D}_k$ 上以学习率 η 训练 $\boldsymbol{\omega}_k^r$ 共 E 个迭代轮次, 第 r 轮训练如下所示:

$$\boldsymbol{\omega}_k^{r+1} \leftarrow \boldsymbol{\omega}_k^r - \eta \nabla_{\omega} \ell(\boldsymbol{\omega}^r; \mathcal{D}_k),\ k \in \mathcal{S}_r \tag{1}$$

训练完成后, 客户端 k 将本地模型 $\boldsymbol{\omega}_k^{r+1}$ 上传到服务器. 服务器将所有接收到的本地模型的参数基于训练数据量进行加权平均来聚合本地模型, 从而更新全局模型参数:

$$\boldsymbol{\omega}^{r+1} \leftarrow \sum_{k \in \mathcal{S}_r} \frac{N_k}{N} \boldsymbol{\omega}_k^{r+1} \tag{2}$$

其中, $N=\sum_{k\in\mathcal{S}_r} N_k$ 为参与本轮的所有客户的本地样本数量之和. 以上步骤共重复 R 轮.

然而, 客户端之间高度倾斜的标签分布容易导致严重的本地模型漂移现象, 从而降低全局模型的性能. 为了在不共享原始数据的前提下解决这个问题, 我们将采用生成对抗网络 (generative adversarial network, GAN)[19]生成高质量的合成数据, 在任务模型训练阶段补充客户端上稀少或缺失标签对应的数据, 以校正客户端间的数据分布差异.

2.2 生成对抗网络

生成模型是一类旨在学习原始数据的分布, 并生成保留特征但与原始数据不同的新样本的机器学习模型. 生成对抗网络 (GAN) 是其中一个广泛应用的模型, 其在训练后能通过噪声生成新样本. 深度卷积生成对抗网络 (deep convolutional generative adversarial network, DCGAN)[20] 是 GAN 的一种变体, 它利用深度卷积神经网络 (deep convolutional neural network, DCNN) 生成高质量的图像. 与依赖全连接层的传统 GAN 不同, DCGAN 采用卷积层来捕获空间依赖关系并学习数据的分层表示.

生成对抗网络由两个神经网络组成: 生成器 G 和鉴别器 D. 生成器的优化目标是从噪声中生成能欺骗鉴别器的样本, 而鉴别器的优化目标是辨别输入是生成器产生的真实样本还是合成样本. GAN 的目标函数 $V(\cdot)$ 表示为:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{3}$$

其中, $p_{\text{data}}(x)$ 为真实数据分布, $p_z(z)$ 为噪声分布. 在训练过程中, G 和 D 被交替更新.

由于 GAN 是无监督学习, 从噪声生成的样本没有标签, 所以我们使用分组操作来固定某个生成模型生成样本的标签并扩大样本空间. 在第 2.3 节中, 我们将介绍隐私保护手段和相关定义, 以便在分组过程中引入相关技术.

2.3 隐私保护手段

隐私是联邦学习的基本属性之一, 联邦学习致力于通过多层次的隐私和安全保障机制保护用户隐私不被侵犯[21–23]. 在机器学习领域, 隐私保护关注在利用数据训练模型的同时, 如何保护敏感信息. 差分隐私 (differential privacy, DP)[24,25]、$\mathcal{K}$-匿名 ($\mathcal{K}$-Anonymity)[26]和安全多方计算[27,28]是其中的 3 个关键技术. 差分隐私和 $\mathcal{K}$-匿名是用数学语言描述的隐私定义, 前者针对算法后者针对数据集, 它们对隐私保护的程度进行了量化. 安全多方计算 (secure multi-party computation, MPC) 是密码学的一个子领域, 其主要目标是使多个参与方能够在保护各自输入隐私的前提下共同计算一个函数.

差分隐私是算法具有的属性, 无论输入数据集中是否包含或排除了任何个体的数据, 满足差分隐私的算法的输出在统计特性上很大程度上保持不变, 从而降低了泄露有关任何特定个体的敏感信息的风险. 差分隐私通过在计算过程中引入随机性来实现该目标, 其数学定义如下.

定义 1. (ϵ,δ)-差分隐私. 对于所有相邻的数据集 D 和 D' (若两个数据集间只有一条样本不同则它们被称为相邻数据集), 一个算法 $\mathcal{A}$ 如果满足 (ϵ,δ)-差分隐私, 用 $Range(\mathcal{A})$ 表示算法可能的输出范围, 那么对于任何输出子集 $S \subseteq Range(\mathcal{A})$, 都应该满足:

$$\Pr[\mathcal{A}(D)\in S]\leqslant \mathrm{e}^{\epsilon}\times\Pr[\mathcal{A}(D')\in S]+\delta \tag{4}$$

其中, Pr 是发生概率. $\epsilon\in(1,\infty)$ 是隐私预算, 用来调整差分隐私定义所能提供的“隐私程度”. ϵ 较小时, 意味着 $\mathcal{A}$ 需要为相似的输入提供非常相似的输出, 因此提供更高等级的隐私性. $\delta\in(0,1)$ 是失效概率, 描述了该算法在多大程度上引入的随机性会失效, δ 的值越小不等式成立概率越大即越不容易失效, δ 取 0 时为 ϵ-差分隐私.

$\mathcal{K}$-匿名 ($\mathcal{K}$-Anonymity) 描述数据集所具有的属性: 即一部分辅助信息不应该“过多地”缩小个体所属记录的可能范围. 换句话说, $\mathcal{K}$-匿名确保了在数据集中的每个个体都与至少另外 $\mathcal{K}-1$ 个个体具有相同的属性值, 从而难以通过某几个属性就单独识别某个具体的个体. 这种技术旨在防止通过数据集的发布或共享来识别特定个体的可能性, 从而保护个人隐私, $\mathcal{K}$ 值越大这个数据集隐私保护程度越高. 其数学形式定义如下.

定义 2. $\mathcal{K}$-匿名. 对于特定的 $\mathcal{K}$, 如果对于任何记录 $r_1\in D$, 存在至少有 $\mathcal{K}-1$ 条其他的记录 $r_2,r_3,\ldots,r_{\mathcal{K}}\in D$, 使得 $\Pi_{qi(D)}r_1=\Pi_{qi(D)}r_2,\Pi_{qi(D)}r_2=\Pi_{qi(D)}r_3,\ldots,\Pi_{qi(D)}r_1=\Pi_{qi(D)}r_{\mathcal{K}}$, 其中 $qi(D)$ 是 D 的伪标识符 (quasi-identifier), $\Pi_{qi(D)}r$ 表示包含伪标识符的记录的列 (即伪标识符的投影), 那么数据集 D 满足 $\mathcal{K}$-匿名.

安全多方计算最早由 Yao[27]在百万富翁问题中提出. 这个问题描述了两方如何在不透露各自财富的情况下比较谁更富有. 此概念后来被推广, 允许涉及任意数量参与方的计算. 其定义如下.

定义 3. 安全多方计算. MPC 协议中具有 n 个参与方, 每个参与方拥有一个私有输入 x_i, 所有参与方共同计算一个函数 $f(x_1,x_2,\ldots,x_n)$, 并确保有以下属性. (1) 正确性: 协议正确地计算函数 f, 即在协议结束时, 所有参与方都能获得正确的函数输出. (2) 隐私性: 除输出信息外, 任何参与方都不能学习到关于其他参与方输入的更多信息. 具体而言, 计算过程中泄露的信息不会危及各方输入的隐私. (3) 输入独立性: 每个参与方的输入独立于其他参与方, 确保输入是在没有任何外部影响下提供的. (4) 公平性: 所有参与方要么都获得输出, 要么都不获得输出. 没有参与方能比其他参与方更早获得输出, 从而获得优势.

2.4 问题定义

$\mathcal{D}_{\text{test}}$ 表示服务器上的测试集. 全局的训练数据集 (即各客户端本地数据集之和) 表示为 $\mathcal{D}_{\text{glob}}=\sum_{k=1}^{K}\mathcal{D}_k$. 客户端本地数据集 $\mathcal{D}_k$ 中标签为 c 的所有样本的总数记为 n_c^k, 那么客户端 k 上的本地数据分布可以用一个 C 维数组 $\boldsymbol{Q}_k=(n_0^k,n_1^k,\ldots,n_{C-1}^k)$ 来表示. 为定量比较任意两个分布 $\boldsymbol{Q}_1$ 和 $\boldsymbol{Q}_2$ 间的差异, 我们可通过 *JS* 散度 (Jensen-Shannon divergence) 来计算它们之间的距离. *JS* 散度是 *KL* 散度 (Kullback-Leibler divergence) 的一种变体, 消除了 *KL* 散度的不对称问题, 其定义如下:

$$\begin{cases} JS(\boldsymbol{Q}_1,\boldsymbol{Q}_2)=\dfrac{1}{2}KL\left(\boldsymbol{Q}_1\,\|\,\overline{\boldsymbol{Q}}\right)+\dfrac{1}{2}KL\left(\boldsymbol{Q}_2\,\|\,\overline{\boldsymbol{Q}}\right) \\ KL(\boldsymbol{Q}_1\,\|\,\boldsymbol{Q}_2)=\sum\boldsymbol{Q}_1\log\dfrac{\boldsymbol{Q}_1}{\boldsymbol{Q}_2},\ \overline{\boldsymbol{Q}}=\dfrac{\boldsymbol{Q}_1+\boldsymbol{Q}_2}{2} \end{cases} \tag{5}$$

我们定义公式 (6) 以便于衡量各客户端数据分布的 Non-IID 程度对全局模型性能的影响. 公式 (6) 计算了每个客户端训练集与全局训练数据分布之间的距离, 其表征了本地模型在聚合时的差异:

$$\mathbb{js}=\frac{1}{K}\sum_{k=1}^{K}JS\left(\boldsymbol{Q}_k',\boldsymbol{Q}'_{\text{glob}}\right) \tag{6}$$

为了使该计算结果仅代表客户端本地各标签数目之间的比例差距 (即描述标签分布偏移程度的大小), 我们对输入作归一化 (normalization) 操作, 即: $\boldsymbol{Q}_k'=\dfrac{\boldsymbol{Q}_k}{\|\boldsymbol{Q}_k\|_1}$, 对 $\boldsymbol{Q}_{\text{glob}}$ 做出了相同的处理得到 $\boldsymbol{Q}'_{\text{glob}}$. 随着客户端本地数据标签分布偏移的加重, $\mathbb{js}$ 的值将从 0 逐渐增大 (由 *JS* 散度的定义可知 $\mathbb{js}$ 值的上界是 1), 全局模型的性能也将逐渐下降. 注意为了避免客户端中没有某个标签 (即 $\boldsymbol{Q}_k$ 中某项为 0) 而导致计算过程中出现除零异常, 在代入计算前向量中的每个元素均加上了最小浮点数精度.

本文的目标是: 当进行一个 $\mathbb{js}$ 的值大于 0 的联邦学习任务时, 在客户端不暴露数据分布的前提下, 优化全局模型 $\boldsymbol{\omega}^R$ 在 $\mathcal{D}_{\text{test}}$ 上的表现.

3 基于隐私分组生成的高效联邦学习 GGC

3.1 总 览

在数据非独立同分布条件下, 标签分布的偏移现象会严重影响联邦学习中的全局模型性能. 为缓解上述问题, 现有的方法大多侧重于在网络层面进行参数校正, 而忽略了数据分布差异的根本问题, 因此这些方法在全局模型性能上的提升有限.

本文从减小客户端相互之间数据本身的分布差异着手, 总体思路是: 通过生成网络获得合成样本, 并利用这些合成样本在任务模型训练前使本地数据分布更加均匀, 进而减少本地模型的参数差异. 然而, 如引言中的探讨, 生成网络既不能在单个客户端上训练, 也不适合在所有客户端上联邦训练. 前者易暴露单个节点的数据分布且跨节点样本无法得到利用, 后者的效果会受到全局数据 Non-IID 影响. 因此, 本文以标签为单位, 将本地持有大量某个标签的客户端标记在一个组内, 组内仅针对这个标签进行生成网络的联邦学习. 基于标签分组的优点是: 1) 组内消除了 Non-IID 的影响, 通过组间隔离降低了不同分布间的相互干扰, 有效保证了生成模型的质量. 2) 生成模型合成的样本标签可以通过组别确定, 不需要有监督的输入标签训练生成网络. 然而该策略面临如下问题: 1) 如何将这些高质量客户端相对准确地标识出来, 并分在同一组, 同时满足一定的数据分布隐私保护要求? 2) 如何合理地对不同标签组的生成网络进行联邦训练? 3) 如何考虑到不同客户端本地的不同数据分布, 合理地进行个性化本地数据分布校正以提高效果?

针对以上挑战, 我们设计了一种框架 GGC 来高效地解决联邦学习 Non-IID 数据分布造成的全局模型性能下降问题. GGC 分为以下 3 个阶段.

(1) 隐私地捕获客户端的主要标签分布, 基于该分布特征将客户端进行分组 (DPG). 本文提出一种数据分布二值化算法来抽取客户端的主要标签分布, 并引入扰动来加强隐私保护. 具体来说, 客户端通过“随机切分”在本地计算出全局各类标签数目的均值, 随后基于这些均值构建可以表征主要标签分布的向量, 再之后对该向量进行“随机翻转”来进行扰动提供隐私保护. 最后客户端向服务器发送这些向量, 服务器依据它们将相似客户端分到同一组别.

(2) 服务器统计全局上的标签在客户端的分布结果, 挑选出分布更集中的标签 (即仅由少数客户端持有的大量标签), 在这些标签对应的组别内各联邦地训练一个生成模型 IGT. 服务器先消除 DPG 引入的扰动, 计算出全局上标签分布的无偏估计, 挑选出最应该生成并分发的标签, 再协调这些标签对应组别内的客户端仅使用该标签对应的样本作为训练集, 无监督地协作训练一个生成模型并保存在服务器上, 即每个组别最后将在服务器上得到一个全局生成模型.

(3) 服务器依据客户端的主要标签分布, 利用训练完成的各全局生成模型生成合成样本并分发, 随后客户端进行对任务模型的联邦学习 (FLC). 对每个客户端, 服务器依据其主要标签分布, 利用全局生成模型生成该客户端稀缺的样本类并分发. 客户端将接收到的合成样本集合并入训练集以使本地数据分布更加均匀, 然后开始针对任务模型执行联邦学习.

其中 DPG 和 IGT 发生在对任务的联邦学习过程前, FLC 发生在任务中. 整体流程如图 2 所示 (图例下文通用), 图中编号 0 为准备阶段, 其余编号分别对应各阶段序号, 下面将详述框架的设计.

3.2 详细方案

在正式阶段开始前, 服务器将经过初始化的任务模型 $\boldsymbol{\omega}$ 和生成模型 $\boldsymbol{\theta}$ 一同发送给各个客户端. 为了明确 GGC 中引入的噪声所能提供的隐私量, 同时服务器设置全局的差分隐私参数为 (ϵ,δ).

3.2.1 分布隐私捕获并分组

分布隐私捕获并分组 (DPG) 阶段在每个客户端本地将不应该被服务器知晓的样本分布 $\boldsymbol{Q}_k$, 使用数据分布二值化算法做脱敏处理, 变成客户端的主要标签分布 $\widetilde{\boldsymbol{Q}_k}$, 然后 $\widetilde{\boldsymbol{Q}_k}$ 被发送到服务器, 服务器依据其对全部客户端进行分组. 如图 3 举例所示, K 个客户端共有 4 类样本, 类别编号为 0、1、2、3. 客户端首先在本地通过“随机切分”计

算得到每类标签在全局上的数目平均值, 再基于这个平均值对 $\boldsymbol{Q}_k$ 的每个元素进行“截断”以保留一定特征, 之后通过“随机翻转”进行进一步混淆得到主要标签分布 $\widetilde{\boldsymbol{Q}}_k$. 最后各客户端向服务器发送 $\widetilde{\boldsymbol{Q}}_k$, 服务器将其中为 1 的元素作为分入该组的标识, 如客户端 1 被分到组别 group_0 和 group_3 中. 数据分布二值化算法主要分为两步, 具体步骤如下.

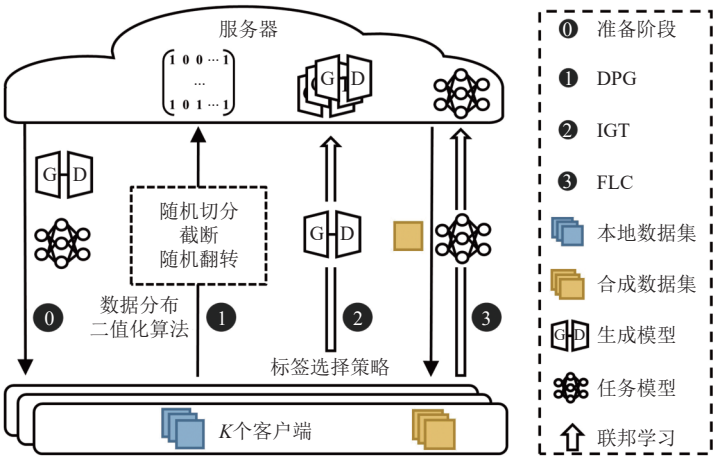


图 2　GGC 的整体框架示意图

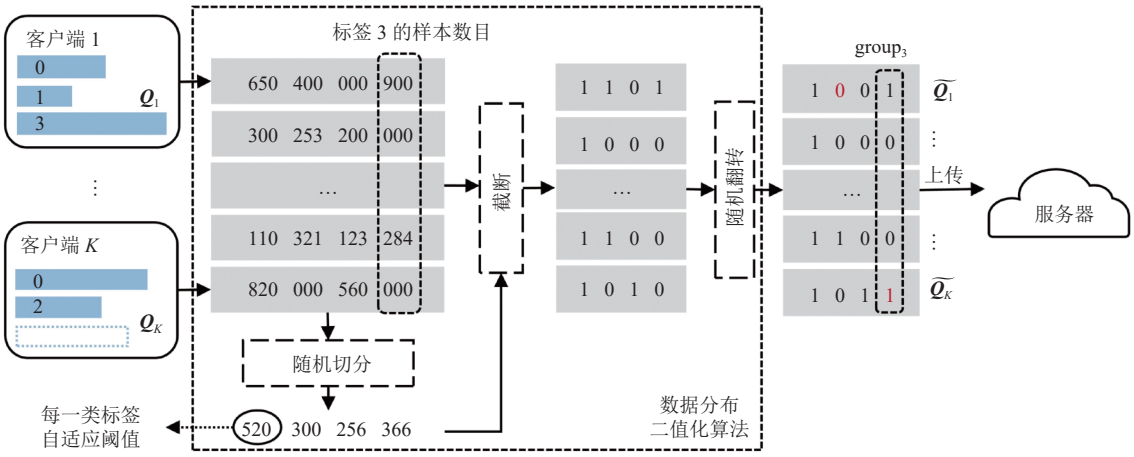


图 3　DPG 的流程示意图

(1) 随机切分后截断

首先, 每个客户端的 $\boldsymbol{Q}_k$ 相当于一条敏感记录, 其在发送给服务器进行分组前需进行处理以保护隐私. 受到图像处理的二值化 (binarization) 和 $\mathcal{K}$-匿名的概括 (generalization) 这两项技术的启发, 我们基于阈值将 $\boldsymbol{Q}_k$ 的每个元素映射到 0 或 1 两个离散的值. 我们将这一步的操作称为截断 (truncation). 现在基于 $\mathcal{K}$-匿名的定义来说明这一步提供的隐私保证, 如图 3 所示, 若以第 1 列作为伪标识符, 处理前 $650 \neq 300 \neq 110 \neq 820$ 则 $\mathcal{K}$ 为 1, 而处理后该列属性值均变为 1 则 $\mathcal{K}$ 增大到 4.

具体来说, 为了隐去每类样本具体个数的同时表征客户端的本地数据分布情况, 所有客户端基于标签样本数量均值对 $\boldsymbol{Q}_k$ 进行修改. 对每类标签 c, 设定一个自适应阈值 $T_c = \left(\sum n_c^k\right)/K$, 若客户端 k 的本地样本数量 $n_c^k > T$, 则

将 $\boldsymbol{Q}_k$ 的第 c 个元素赋值为 1, 否则赋值为 0. 记元素全部被修改后的向量为 $\widetilde{\boldsymbol{Q}_k}=(m_0^k,m_1^k,\ldots,m_{C-1}^k)$, 其表征了客户端的主要标签分布. 为了能使 n_c^k 的值在不被任何参与方知晓的前提下计算得到阈值 T_c, 本文引入安全多方计算的思想, 采用随机切分在每个客户端上计算得到阈值 T_c.

定理 1. 随机切分 (random split). 每个客户端采取如下方式可以在 n_c^k 完全保持未知的情况下在本地计算出 K 个 n_c^k 的均值, 即为阈值 T_c. 以客户端 k 为例: (1) 客户端 k 将 n_c^k 随机分割成 K 份, 即 $n_c^k=\sum_{i=1}^{K}a^{ki}$, 本地保留一份, 然后将其中的 $K-1$ 份随机一对一分发给其他 $K-1$ 个客户端. (2) 客户端 k 待接收到其他客户端发送来的 $K-1$ 个数据 a, 将它们和自己保留的那份数据进行求和, 得到 $b^k=\sum_{i=1}^{K}a^{ik}$, 随后将 b^k 分发给其他客户端. (3) 客户端 k 待收到其他 $K-1$ 个数据 b 后, 将这些数据累加得到总和 $sum=\sum_{j=1}^{K}b^j$, 那么阈值 $T_c=\dfrac{sum}{K}$.

证明: 随机切分在 n_c^k 不暴露的情况下, 使每个客户端计算得到了阈值 T_c. 依据上述步骤我们可得 sum 为:

$$sum=\sum_{j=1}^{K}b^j=\sum_{j=1}^{K}\left(\sum_{i=1}^{K}a^{ij}\right)=\sum_{i=1}^{K}\left(\sum_{j=1}^{K}a^{ij}\right)=\sum_{i=1}^{K}n_c^i \tag{7}$$

由于对数值的切分和分发过程都是完全随机的, 那么在 n_c^k 仍仅由客户端 k 知晓具体数值的前提下, 在每个客户端本地均计算得到了 K 个 n_c^k 的总和 sum, 此时 T_c 即为所求阈值.

(2) 随机翻转后分组

每个客户端随后向 $\widetilde{\boldsymbol{Q}_k}$ 进行进一步修改再将其上传服务器. 虽然此时每个客户端都尽可能地融入了“群体”, 但 $\widetilde{\boldsymbol{Q}_k}$ 仍蕴含部分特征, 为了防止恶意服务器明确地识别客户端的优势标签. 我们在每个客户端本地使用随机翻转来为 $\widetilde{\boldsymbol{Q}_k}$ 引入扰动, 以使 $\widetilde{\boldsymbol{Q}_k}$ 在脱离客户端本地前就具有 ϵ 的隐私保证.

定理 2. 随机翻转 (random flip). 客户端对 $\widetilde{\boldsymbol{Q}_k}$ 中的每个元素以 $q=(1+\mathrm{e}^{\epsilon})^{-1}$ 的概率进行翻转 (即 1 变为 0, 0 变为 1), 该翻转过程满足 (ϵ, 0)-差分隐私, 记翻转前的元素为 $\widetilde{\boldsymbol{Q}_k}[c]$, 翻转后的元素为 $\widetilde{\boldsymbol{Q}_k}[c]'$, 公式中 $c\in[0,C-1]$, $i\in\{0,1\}$, 形式化描述如下:

$$\Pr\left[\widetilde{\boldsymbol{Q}_k}[c]'=i\right]=\begin{cases}1-q, & \widetilde{\boldsymbol{Q}_k}[c]=i\\ q, & \widetilde{\boldsymbol{Q}_k}[c]\neq i\end{cases} \tag{8}$$

证明: 随机翻转满足 (ϵ, 0)-差分隐私. 若记翻转前的元素为 m, 翻转后的元素为 m', 对于 $i\in\{0,1\}$ 经过处理后的输出均满足如下公式:

$$\frac{\Pr[m'=i|m=i]}{\Pr[m'=i|m\neq i]}=\frac{1-q}{q}=\frac{\dfrac{\mathrm{e}^{\epsilon}}{1+\mathrm{e}^{\epsilon}}}{\dfrac{1}{1+\mathrm{e}^{\epsilon}}}=\mathrm{e}^{\epsilon} \tag{9}$$

依据差分隐私的定义公式 (4), 即证明该操作满足 $(\epsilon,0)$-差分隐私.

被翻转的元素在图 3 中用红色进行了标注. 翻转后每个客户端将 $\widetilde{\boldsymbol{Q}_k}$ 发送给服务器. 服务器依据 $\widetilde{\boldsymbol{Q}_k}$ 对客户端进行分组, 若 $m_c^k=1$ 则服务器将客户端 k 分到组别 group_c 中. 此时, 服务器只有 $(1-q)^C$ (通常这是一个远小于 1 的值, 如 $C=4$, $\epsilon=1$ 时约为 0.28) 的把握能认定 $\widetilde{\boldsymbol{Q}_k}$ 是翻转前的向量, 且对每个标签我们至少提供了 ϵ 的隐私量, 加之对数据分布的截断处理可以认为我们对客户端数据分布做出了足够的隐私保护.

3.2.2 组内联邦生成模型训练

组内联邦生成模型训练 (IGT) 阶段为了扩大样本空间以提高合成样本的质量, 在 DPG 划分好的组内使用联邦学习的方法训练一个生成模型, 并固定每个客户端仅使用组别对应的样本作为训练集. 同时我们注意到不同标签样本的生成所能缓解的数据异质性也存在差异, 具体来说, 集中在少数客户端内的标签样本应该被优先生成 (即分发它们对全局模型性能的提升贡献最大), 而在客户端之间分布均匀的标签样本应该暂缓生成. 考虑到生成模型的训练成本我们挑选 n_L 个对提高全局模型性能贡献最大的标签 (类), 在这些标签对应的组内联邦地训练生成模

型. 如图 4 举例所示, 服务器挑选出了贡献最大的类 3, 然后协调 group_3 中记录的客户端仅用样本 $\{\boldsymbol{x}^3\}$ 作为训练集协作训练一个全局生成模型 $\boldsymbol{\theta}_3$. 具体步骤如下.

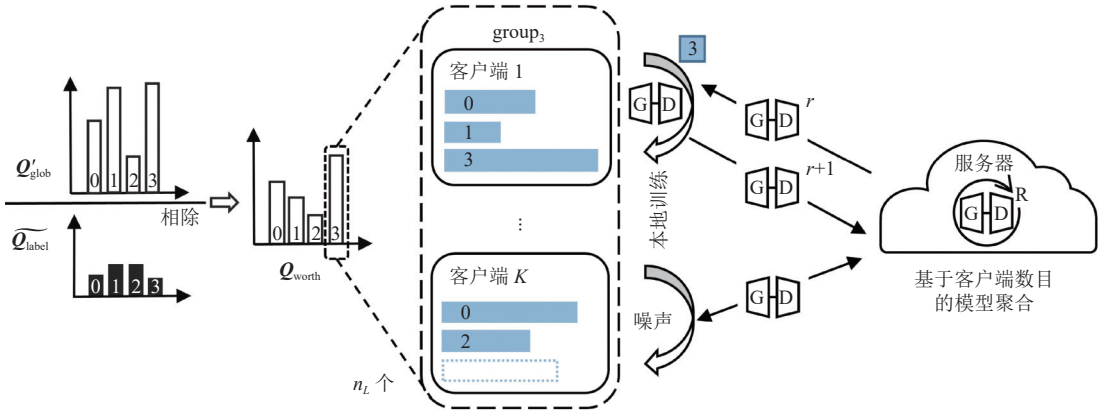


图 4 IGT 的流程示意图

(1) 标签选择策略

我们记全局的标签分布为 $\widetilde{\boldsymbol{Q}}_{\text{label}}$, 服务器首先通过下式计算其中元素值:

$$\widetilde{\boldsymbol{Q}}_{\text{label}}[c]=\frac{\sum_{k=1}^{K}\widetilde{\boldsymbol{Q}}_k[c]-Kq}{1-2q} \tag{10}$$

直观解释如下: DPG 引入的随机翻转将 0 以 q 的概率翻转成 1, 那么累加后的标签总数 $\sum_{k=1}^{K}\widetilde{\boldsymbol{Q}}_k[c]$ 需要先减去被翻转成 1 的个数 Kq. 再考虑到 1 被翻转成 0 的比例, 该结果再除以 $1-2q$ 可还原出原始标签分布. 易证明 $\widetilde{\boldsymbol{Q}}_{\text{label}}[c]$ 是 $\sum_{k=1}^{K}\widetilde{\boldsymbol{Q}}_k[c]$ 的无偏估计. 此时 $\widetilde{\boldsymbol{Q}}_{\text{label}}$ 索引位置为 c 的元素的值越小, 代表该标签在客户端上的分布越不均匀 (即少量客户端本地具有大量标签).

与此同时还需要考虑到不同标签的数目差异. 一般希望全局模型能更好地泛化测试集中占比较大的标签. 我们用 $\boldsymbol{Q}'_{\text{glob}}$ 来代表这种偏好性, $\boldsymbol{Q}'_{\text{glob}}$ 是服务器由测试集模拟地进行了归一化的全局样本分布 (若客户端信任服务器, 也可通过 DPG 中得到的 T_c 来模拟), 那么 $\boldsymbol{Q}'_{\text{glob}}$ 索引位置为 c 的元素的值越大, 则代表 $\{\boldsymbol{x}^c, y^c\}$ 在测试集中越占据主导地位. 我们记贡献向量为 $\boldsymbol{Q}_{\text{worth}}$, 其中每个元素值为:

$$\boldsymbol{Q}_{\text{worth}}[c]=\frac{\boldsymbol{Q}'_{\text{glob}}[c]}{\widetilde{\boldsymbol{Q}}_{\text{label}}[c]} \tag{11}$$

此时 $\boldsymbol{Q}_{\text{worth}}$ 中索引位置为 c 的元素的值即代表着生成这类标签带来的价值. $\boldsymbol{Q}_{\text{worth}}[c]$ 越大意味着 $\{\boldsymbol{x}^c, y^c\}$ 在全局上数量越多或在客户端中分布得更集中, 则 c 类样本更值得被生成并分发给其他客户端.

(2) 组内联邦训练生成网络

随后服务器选择 $\boldsymbol{Q}_{\text{worth}}$ 中值最大的 n_L 个标签形成标签集 S_L. 对标签集 S_L 中的每个标签对应的组别 group_c, 服务器协调组别内所有客户端作为参与方, 向本地持有的 $\{\boldsymbol{x}^c\}$ 中加入少量噪声 (这是惯用的生成模型训练技巧且能为原始样本提供一定的隐私保护) 后作为训练集, 采取优化目标为公式 (3) 的无监督联邦学习, 最后在服务器上聚合产生一个全局生成模型 $\boldsymbol{\theta}_c$. 此阶段中为防止客户端关于数据分布的隐私被泄露, 采用以下两个实现细节.

1) 服务器基于客户端数量对本地模型做平均聚合. 因为客户端仅使用 $\{\boldsymbol{x}^c\}$ 作为训练集, 若仍采取基于样本数量的聚合方式产生全局生成模型则需要将 $\{\boldsymbol{x}^c\}$ 的数目作为参数发送给服务器, 但具体某类样本的数目作为隐私不应被服务器知晓. 因此作为对公式 (1) 的替代, 客户端并不上传训练集大小. 经过 DPG 的筛选, 每个参与的客户端

使用的训练集规模相似, 可以对全局模型做出平等的贡献, 因此在服务器上使用下式对各本地模型的参数作基于客户端数目的平均聚合, 其中 n 是参与该次联邦学习的客户端总数:

$$\boldsymbol{\theta}^{r+1} \leftarrow \frac{1}{n}\sum \boldsymbol{\theta}_k^{r+1} \tag{12}$$

2) 被错误分组的客户端改变本地训练方式. 注意到由于随机翻转, 组别中可能存在被错误分组的客户端, 如图 3 所示, 客户端 K 在本地并没有样本 $\boldsymbol{x}^3$ 但被分到了 group_3 中. 这类客户端为避免被服务器标识出来, 仍需要上传更新过的模型参数, 它们在本地训练时采用如下手段: 仅使用噪声生成的合成样本进行网络训练或者用噪声代替梯度更新网络参数 (关于向网络参数中添加噪声, 在差分隐私机器学习、联邦学习下的搭便车攻击等领域均有论述, 如直接引入随机噪声, 或可以先对网络的参数或者训练梯度进行裁剪 (clip), 将裁剪量作为敏感度应用高斯机制等, 这里不属于本文研究重点故未展开讨论). 前者仅需迭代少量次数使网络参数发生变化即可; 与后者引入噪声都是一种常见的可以稳定生成对抗网络训练的技巧[29,30], 噪声亦扩展了网络参数的探索空间使得生成模型生成的样本更具多样性, 故该做法基本不带来负面影响.

IFT 结束后在服务器上得到了 n_L 个能从噪声生成样本的生成模型, 服务器仅保留其生成器.

3.2.3 数据补齐下的任务联邦学习

数据补齐下的任务联邦学习 (FLC) 利用 IGT 产生的若干生成模型 $\boldsymbol{\theta}$ 的生成器 G, 在各客户端本地生成合成样本并将其加入训练集, 补齐各客户端在任务模型训练时的数据分布, 并控制了原始样本和合成样本的数目比例, 以避免合成样本影响本地模型对原始样本的拟合. 如图 5 所示, 服务器参考 $\widetilde{\boldsymbol{Q}}_1$ 认为客户端 1 缺少 1 类和 2 类样本, 那么会向客户端 1 合成并分发这些样本, 客户端将这些样本合并入训练集. 当所有客户端数据分布校正完毕后, 开展任务模型的联邦学习. 具体如下.

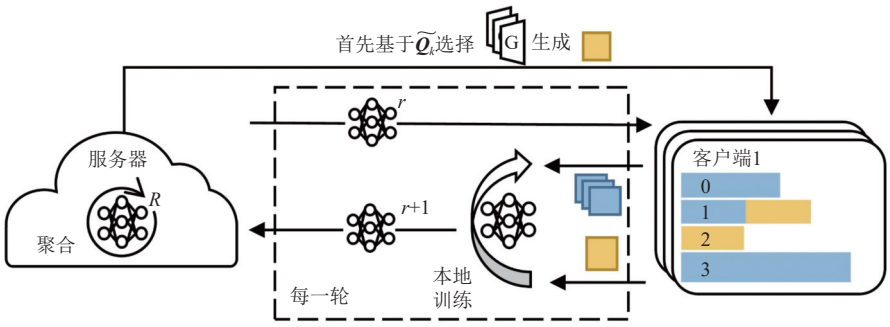


图 5 FLC 的流程示意图

对于每个客户端, 将客户端 k 的 $\widetilde{\boldsymbol{Q}}_k$ 中所有零元素的下标组成的集合记为标签集 S_F^k. 在任务联邦学习开始前, 服务器选取 S_F^k 中记录的标签对应的生成器, 向这些生成器输入噪声生成合成样本集 $\widetilde{\mathcal{D}}_k$, 并发送给客户端 k. 客户端 k 将 $\widetilde{\mathcal{D}}_k$ 与本地的原始样本集合并作为训练集. 当所有客户端的数据补齐完成后, 开始对任务模型的联邦学习, 以公式 (1) 更新本地任务模型, 以公式 (2) 在服务器上聚合得到新一轮的全局任务模型.

为了控制开销并防止过多的合成样本使本地模型对原始样本的拟合能力下降, 我们引入参数 $\alpha \in (0,\infty)$ 控制合成样本和原始样本在训练集中的比例, 来调节原始样本和合成样本在本地训练时对网络参数的贡献. 由于合成样本质量低于原始样本将引入噪声, 故合成样本数目不应超过原始样本的数目, 即 α 应取小于 1 的值. 合成样本集 $\widetilde{\mathcal{D}}_k$ 可由下式表达:

$$\widetilde{\mathcal{D}}_k = \bigcup_{c\in S_F^k \cap S_L} \{(\widetilde{\boldsymbol{x}},\widetilde{y}) | \widetilde{\boldsymbol{x}} = G_c(\mathbf{z}), \widetilde{y} = c\}_i^{(\alpha\times N_k)/(|S_F^k \cap S_L|)} \tag{13}$$

其中, $\mathbf{z}$ 是向生成器输入的随机噪声向量, 一般采样自标准正态分布, G_c 是由 group_c 训练得到的生成模型 $\boldsymbol{\theta}_c$ 的生成器. 总结 FLC 的流程如算法 1 所示.

算法 1. 数据补齐下的任务联邦学习 (FLC).

输入: 全局任务模型参数 $\boldsymbol{\omega}^0$, 通信轮次数目 R, 本地迭代轮次数目 E, 学习率 η, 调节参数 α;
输出: 最终的全局任务模型 $\boldsymbol{\omega}^R$.

1. 初始化全局模型 $\boldsymbol{\omega}^0$
2. 服务器计算每个客户端的 S_F^k
3. 服务器生成 $\widetilde{\mathcal{D}_k}$ 并分发 // $\widetilde{\mathcal{D}_k}$ 由公式 (13) 得到
4. **for** 每个通信轮次 $r=0,1,\ldots,R-1$ **do**: //在服务器上进行
5. 随机选若干个客户端组成 $\mathcal{S}_r$
6. **for** 每个客户端 $k\in\mathcal{S}_r$ 并行的 **do**:
7. 服务器发送客户端 k 全局模型 $\boldsymbol{\omega}^r$
8. $\boldsymbol{\omega}_k^{r+1}\leftarrow$ 客户端更新 $(k,\boldsymbol{\omega}^r)$
9. **end for**
10. $N\leftarrow\sum_{k\in\mathcal{S}_r}N_k$
11. $\boldsymbol{\omega}^{r+1}\leftarrow\sum_{k\in\mathcal{S}_r}\frac{N_k}{N}\boldsymbol{\omega}_k^{r+1}$
12. **end for**
13. **return** $\boldsymbol{\omega}^R$
14. 客户端更新 $(k,\boldsymbol{\omega})$: //在客户端上进行
15. **for** 每个迭代轮次 $i=1,2,\ldots,E$ **do**:
16. $\boldsymbol{\omega}\leftarrow\boldsymbol{\omega}-\eta\nabla_{\boldsymbol{\omega}}\ell\left(\boldsymbol{\omega}^r;\mathcal{D}_k\cup\widetilde{\mathcal{D}_k}\right)$
17. **end for**
18. **return** $\boldsymbol{\omega}$

3.3 代价分析

在 GGC 框架中, DPG 和 IGT 阶段发生在任务模型训练之前, 而 FLC 阶段贯穿于任务模型的训练过程中. 因此, 本节的代价分析主要围绕 FLC 阶段展开, 具体包括以下 5 部分: 服务器的额外存储代价、客户端的额外存储代价、服务器与客户端的额外通信代价、服务器的额外计算代价、客户端的额外计算代价.

上述 5 种代价均与生成模型及合成样本的使用密切相关. 值得注意的是, 在任务模型的训练过程中, 实际参与训练的是由生成器生成的合成样本集, 而生成器在此阶段并不会被进一步更新. 因此, 服务器在存储上有两种可选方案: (1) 保留生成器. 服务器持续存储生成器, 以支持动态生成合成样本. (2) 保留合成样本集. 每当完成一个生成模型的训练后, 将其生成的合成样本存储下来, 随后删除生成模型.

本文选择保留生成器有以下两方面的考量: 一方面保留生成器能够更自由地选择合成样本的数量, 适应不同的任务需求, 另一方面在训练时动态从噪声中采样生成样本, 有助于提高样本的多样性, 进而更全面地捕获其他客户端某类样本的特征. 此外, 由于每次仅需拟合一类样本, 生成模型的参数量通常较小, 存储开销相对较低. 因此, 从综合效益的角度出发, 保留生成器是一种更优的选择. 不过在一些特殊场景下, 如果需要补充的合成样本集远小于生成器的存储开销, 仅保留合成样本集也不失为一种可行的替代方案.

记生成器的大小为 γ, 每个样本的大小为 τ, 所有客户端的本地样本数目的最大值为 $N_{\text{Max}}=\max(N_1,N_2,\ldots,N_K)$. (1) 若保留生成器, 服务器需要保留每个生成器并生成后分发合成样本, 客户端需要让合成样本参与本地训练. 那么服务器额外存储代价为 γn_L、客户端的存储代价和通信代价均为 $\tau\alpha N_k$、服务器的额外计算代价为 $\Theta\left(\alpha\sum N_k\right)$、客户端的额外计算代价为 $\Theta(\alpha N_k)$. (2) 若服务器保留合成样本集, 那么对每类样本都需要存储所有客户端可能需求的最大值, 服务器额外存储代价变为 $\alpha N_{\text{Max}} n_L$, 客户端的存储代价和通信代价不变为 $\tau\alpha N_k$、服务器的额外计算

代价为$\Theta(\alpha K N_{\text{Max}})$、客户端的额外计算代价不变为$\Theta(\alpha N_k)$. 需要注意的是, 现代深度学习框架如 PyTorch 通常批量处理以利用 GPU 的并行计算能力来加速计算, 因此, 计算代价并不严格线性增长.

4 实验评估

本节我们评估并对比了 GGC 与基线方法在 Non-IID 数据分布情况下的表现. 为了更加清晰地讨论实验结果, 本文希望通过实验评估回答以下 7 个问题.

Q1: GGC 与经典的解决 Non-IID 问题的基线方法相比在不同的数据分布下表现是否均更有优势?

Q2: GAN 一般只用于图像数据的生成, 将其迁移到非图像数据集上时 GGC 效果如何?

Q3: 本文提出的分组生成的方法是否有效提高了合成样本的质量?

Q4: 通过全局生成模型生成的合成样本与原始样本的差异性如何, 会不会暴露原样本隐私?

Q5: GGC 受超参数的影响如何, 应该如何选择这些超参数?

Q6: GGC 与其他基线方法结合时在各场景下表现怎么样?

Q7: GGC 对计算资源的消耗情况怎么样?

4.1 实验设置

(1) 数据集

为了验证所提方法的有效性, 我们选用了在联邦学习领域中常用的 3 个知名公开数据集进行实验, 它们具有不同的特点.

CIFAR-10: 包含来自 10 个不同类别的 60000 张 32×32 像素的彩色图像, 涵盖了多种日常物体, 如飞机、汽车、鸟类、猫、鹿等. 其中每个类别共有 6000 张图像, 训练集共 50000 张, 测试集 10000 张 (类别均衡三通道).

Fashion-MNIST: 是一个用于图像分类的数据集, 包含 10 个类别的灰度服装图像, 共有 70000 张图片, 训练集每个类别有 6000 张 28×28 像素的单通道图片, 剩下为数量均等的测试集 (类别均衡单通道).

SVHN: 是一个用于字符识别的数据集, 主要用于识别房屋门牌上的数字, 包含来自 Google 街景图像的彩色 32×32 像素的彩色数字图像. 数据集共有 10 万张图片, 其中每个类别数量不均衡 (类别失衡三通道).

(2) 数据分布场景设置

对于数据 Non-IID 场景设置, 我们主要关注对模型性能影响最大的标签分布偏移, 依据前人工作按照以下两类划分方法将全局数据集分配到每个客户端中.

基于数量的偏移划分: 划分完成后每个客户端本地的标签类别是总标签类别的一个子集. 具体来说, 首先随机为每个客户端分配 l 个不同的标签编号, 随后将数据集中所有具有该标签的样本, 随机且数量均等地分配到拥有该标签编号的客户端中. 为了表示方便, 在后文中使用 $C=l$ 来指代这样的划分策略.

基于分布的偏移划分: 划分完成后客户端持有基于狄利克雷分布的样本. 具体来说, 对每个标签, 首先从 $p_c \sim Dir_K(\lambda)$ 中进行采样, 随后将所有标签为 c 的样本按照 $p_{c,k}$ 的比例分配给客户端 k. 其中 $Dir(\cdot)$ 为狄利克雷分布, λ 为分布的浓度参数, 其值越大生成的分布越均匀, 即会导致每个标签在客户端上的分布越接近 IID, 反之标签在客户端上的分布越极端. 为了表示方便, 在后文中使用 $Dir(\lambda)$ 来指代这样的划分策略.

具体来说设置了以下 5 种场景, 根据公式 (6) 可计算它们的 $\mathrm{j_S}$ 值按照由高到低排列分别为: C=1、C=2、C=3、Dir(0.2)、Dir(0.5).

(3) 评估指标

我们使用所有轮次中, 全局模型在测试集上的准确率的最大值 (下文同 ACC, 百分数表示) 作为性能的主要评价指标, 同时必要时加以比较方法的收敛速度.

(4) 基线方法

我们采用了 10 个经典的能应对 Non-IID 问题的联邦学习算法作为基准, 进行对比实验, 如下所示.

FedAvg[1]: 是联邦学习领域中最重要的基石算法, 每个客户端使用自己的数据训练本地模型, 然后在中央服务

器上对各更新后的模型进行加权平均.

FedProx[5]: 通过在本地更新过程中加入一个正则项来限制本地模型与全局模型之间的偏离程度以提高全局模型性能.

MOON[6]: 在本地更新过程中, 通过引入对比学习机制, 在模型表示的层面进行对比学习, 使得不同客户端的模型能够更好地捕捉和理解各自数据的特性, 从而提升全局模型的泛化能力.

SCAFFOLD[16]: 通过在客户端和服务器上引入包含模型更新方向的控制量, 来校正客户端本地更新过程中的偏差.

FedNova[13]: 引入归一化平均和一种新的权重分配机制, 有效地平衡了不同客户端在参与联邦学习过程中的贡献度. 这种机制考虑到了客户端的本地数据量和更新次数, 确保了在梯度聚合过程中更加有效地整合不同客户端的更新信息.

FedAVGM[3]: 在服务器聚合时添加之前轮次计算得到的动量, 以期望在各本地优化目标不一致时改变全局模型的更新方向突破局部最优点.

FedAdagrad[15]: 在服务器上根据各客户端本地模型参数更新的历史信息来自适应地调整服务器端的参数聚合幅度. 更新幅度越大, 学习率越小; 更新幅度越小, 学习率越大.

FedAdam[15]: 对客户端本地模型参数的更新, 计算每个参数的一阶矩和二阶矩的指数移动平均, 在服务器上利用这些矩的估计值来调整每个参数的学习率, 以修正更新幅度和方向.

FedYogi[15]: 引入了符号函数对参数更新提供约束, 以在服务器上更新全局模型参数时, 更少地被离群客户端的更新影响.

FedUV[17]: 引入分类器方差和超球均匀性这两项损失来规范化本地模型的训练.

(5) 联邦学习参数及其他设置

公平起见, 本节遵循前人的工作[2]对任务联邦学习的参数和任务模型进行统一设置, 且为了保证实验结果的稳定, 服务器会挑选所有客户端参与联邦训练: 设置本地训练的批大小 B 为 64, 学习率 η 为 0.01, 本地迭代轮次 E 为 10 轮, 客户端总数 K 为 10 个, 联邦学习通信轮次 R 为 50 轮. 生成模型参考 DCGAN, 判别器和生成器分别由 4 个卷积层组成. 因各标签在各客户端均持有较少, 故为保证生成质量, 生成模型在各客户端本地共至少迭代 150 个 epoch. 实际上服务器可监控全局生成模型的质量, 动态调整训练过程及超参数, 根据需要提前结束或者延长训练. GGC 的参数设置如下: $n_L = C$, $\alpha = 0.2$. 同时依据苹果公司发布的产品隐私参数[31]来设置本文提出的方法中相应的参数 $\epsilon = 4$, $\delta = 0$. 所有客户端及服务器均在 CPU 为 Intel(R) Xeon(R) Silver 4110、GPU 为 NVIDIA RTX A5000 的机器上模拟运行.

4.2 Q1: 主要对比实验

为了使实验结果展示得更加清晰, 我们在图 6 中绘制了准确率随轮次的变化情况, 在表 2 中记录了各方法不同场景下的 ACC, 在表 3 中统计了不同场景下达标轮次对比结果, 本文表中的最优结果被加粗标出. 我们分析实验结果得到如下结论.

(1) 所有的方法均受到原始数据分布的影响, 数据分布越极端全局模型的性能损失越剧烈. 对于基线方法来说, 数据分布越极端, 从模型参数获取的知识越不可靠, 越难指导本地模型正确更新. 而在绝大多数的实验场景中 GGC 的表现都远优于各基线方法, 这证明了我们方法的有效性.

(2) 同时在所有的实验场景中, GGC 的表现总优于 FedAvg, 这表明各数据分布设置下合成样本的质量总是可靠的, 向客户端分发这些合成样本不会导致全局模型的性能受到损害. 这也是 GGC 相比其他基线方法能够稳定提升全局模型性能的一个重要原因: GGC 提前提取到了稳定可靠的样本知识指导本地模型训练, 而不依赖于任务模型训练过程中不可靠的模型参数知识规范优化方向.

(3) 得益于对数据分布的校正, GGC 相比于其他基线方法, 在不同的数据分布下都展现出了鲁棒性. 其能使全局模型更快更稳定的收敛, 不会如基线方法产生强烈的波动, 其往往能使准确率提高 3% 以上, 同时使达到目标准

确率的联邦学习轮次减半, 具有较高的加速倍率, 并且不易受到数据分布的影响, 能在早期轮次使全局模型性能迅速攀升.

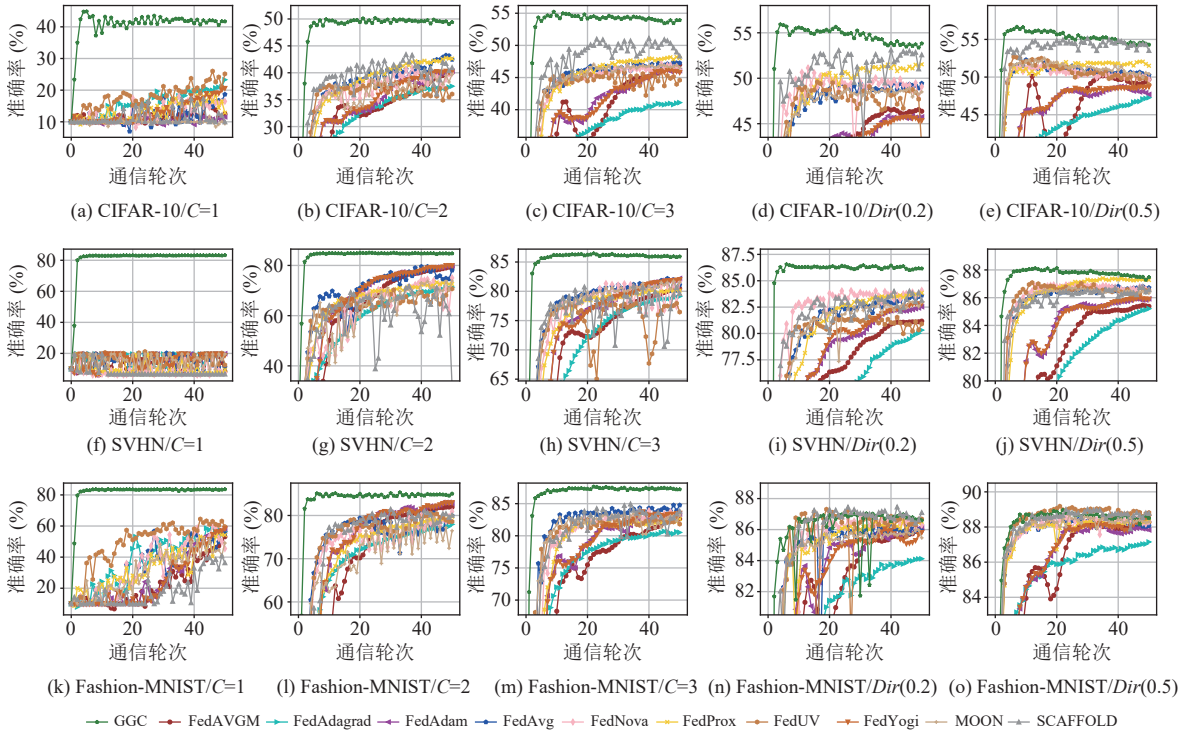


图 6　全局模型性能随着轮次的变化情况

表 2　各方法在 3 个数据集上的 ACC 比较 (%)

Method	CIFAR-10					SVHN					Fashion-MNIST				
	C=1	*C*=2	*C*=3	*Dir*(0.2)	*Dir*(0.5)	*C*=1	*C*=2	*C*=3	*Dir*(0.2)	*Dir*(0.5)	*C*=1	*C*=2	*C*=3	*Dir*(0.2)	*Dir*(0.5)
FedAvg	18.67	43.25	47.39	49.53	52.29	20.23	79.55	82.18	83.60	86.82	56.73	80.66	84.77	86.42	88.56
FedNova	18.09	40.90	47.13	51.12	51.45	19.58	75.01	81.66	84.08	86.94	60.41	80.92	83.91	86.83	88.53
FedProx	18.67	42.77	48.19	51.54	52.05	19.59	73.25	80.20	83.84	87.40	55.81	79.70	82.61	86.55	88.37
MOON	14.37	39.73	47.06	48.90	52.49	19.58	73.79	80.71	83.05	86.74	52.01	79.56	83.49	85.99	88.67
SCAFFOLD	17.36	43.40	51.10	53.11	55.13	19.58	71.74	80.94	84.02	86.74	40.79	81.46	84.95	**87.45**	89.01
FedAVGM	16.25	40.00	47.00	46.64	50.18	19.58	79.81	80.45	81.17	85.46	53.09	81.95	82.63	86.02	88.32
FedAdagrad	23.29	37.50	41.09	43.18	47.27	19.63	71.35	79.12	80.32	85.20	59.28	77.95	80.70	84.11	87.15
FedAdam	12.68	40.31	46.36	45.87	48.75	19.81	79.81	82.19	82.53	85.92	58.19	83.04	83.39	86.15	88.11
FedYogi	20.54	40.29	46.21	45.75	48.69	19.64	79.96	81.98	82.82	85.98	58.24	82.96	83.73	86.04	88.57
FedUV	26.02	38.58	46.14	49.58	52.69	21.16	70.68	77.98	81.57	87.08	64.48	80.54	82.77	87.31	**89.18**
GGC	**44.84**	**50.17**	**55.22**	**55.92**	**56.66**	**83.33**	**85.07**	**86.46**	**86.28**	**88.13**	**83.77**	**85.34**	**87.64**	87.03	88.96

(4) SVHN/*Dir*(0.5) 和 Fashion-MNIST/*Dir*(0.5) 中 GGC 效果较差的原因是, 由于隐私保护引入的噪声和扰动, GGC 对各客户端数据分布的估计是不够可靠的, 那么原始数据分布越接近 IID (此时也越不需要其他方法对联邦学习纠偏, 仅采用基础方法时全局模型表现就足够优良), GGC 引入的合成样本越容易干扰到对原始样本的拟合, 使泛化能力下降.

(5) 所有基线方法中对 ACC 提高仅次于 GGC 的方法是 SCAFFOLD, 但同时它也是训练最不稳定的方法, 不同轮次间准确率波动剧烈. 我们认为这是由于其虽然通过控制变量引入了相对可靠的知识, 但仍然未有效校正客户端分布间的极端偏差, 客户端间优化目标依然显著背离.

表 3　不同场景下到达目标准确率所需轮次

数据集	场景	js	FedAvg	Fed-Nova	Fed-Prox	MOON	SCAFFOLD	Fed-AVGM	Fed-Adagrad	Fed-Adam	Fed-Yogi	FedUV	GGC	加速倍率
CIFAR-10 (59.05%)	*C*=1	0.525	—	—	—	—	—	—	—	—	—	—	—	—
	C=2	0.424	—	—	—	—	—	—	—	—	—	—	**4**	∞
	C=3	0.349	48	—	30	—	8	—	—	—	—	—	**3**	×2.7
	Dir(0.2)	0.289	11	7	11	11	4	—	—	—	—	9	**2**	×2
	Dir(0.5)	0.187	4	4	4	4	3	11	50	20	21	4	**2**	×1.5
SVHN (89.26%)	*C*=1	0.528	—	—	—	—	—	—	—	—	—	—	**2**	∞
	C=2	0.440	21	38	36	40	36	28	—	23	22	—	**2**	×10.5
	C=3	0.351	5	8	9	6	5	13	20	10	10	6	**2**	×2.5
	Dir(0.2)	0.341	5	4	7	6	3	13	21	11	11	5	**2**	×1.5
	Dir(0.5)	0.231	3	3	3	3	3	9	11	8	8	3	**2**	×1.5
Fashion-MNIST (89.88%)	*C*=1	0.525	—	—	—	—	—	—	—	—	—	—	**2**	∞
	C=2	0.424	7	8	16	20	9	22	19	14	11	7	**1**	×7
	C=3	0.349	4	6	7	4	5	11	10	8	8	5	**1**	×4
	Dir(0.2)	0.271	3	**2**	5	3	**2**	10	8	7	8	3	**2**	×1
	Dir(0.5)	0.165	2	2	2	2	2	4	4	3	3	2	**1**	×2

注: 第1列括号内是IID时FedAvg全局模型的准确率, 0.8倍该值为目标准确率, —为不能达到, 最后一列是相比对比方法GGC最低加速倍率

(6) FedProx 限制客户端向全局最优更新参数的优化策略被证明总能稳定的提升全局模型性能, 而 MOON 引入的模型对比学习在数据分布极端时对本地模型的指导一般是有害的.

(7) 由于联邦学习场景、设置复杂多变, FedYogi 等方法过于依赖选择合适的超参数, 导致其性能在具体场景下往往不如人意.

4.3　Q2: 文本数据集实验

为了探究 GGC 在其他数据集上的表现, 我们在文本数据集 AG-News 上开展了实验. AG-News 是一个经典的文本分类数据集, 其涵盖了 4 类新闻文本, 训练集包含 12 万个样本, 每个类别各 3 万个样本, 测试集包含 7600 个样本, 每个类别各 1900 个样本.

为了便于 GAN 进行生成, 我们利用 doc2vec (又名 paragraph vector)[32,33] 对样本进行预处理, 将样本映射到 100 维的向量. 任务模型采用 4 层全连接层组成的神经网络, 生成模型采用原始的 GAN 结构. 实验结果如表 4 所示.

表 4　文本数据集 AG-News 上各方法的性能比较

Method	*C*=1	*C*=2	*C*=3	*Dir*(0.2)	*Dir*(0.5)
FedAvg	35.96/—	74.53/4	74.88/3	67.43/13	75.07/3
FedNova	37.20/—	73.53/4	74.91/3	65.07/8	77.34/**2**
FedProx	29.16/—	73.66/5	76.49/3	67.25/19	77.14/3
MOON	36.54/—	73.70/3	74.17/3	63.84/—	75.71/3
SCAFFOLD	40.97/—	75.05/3	**77.99**/2	35.53/—	76.04/**2**
FedAVGM	31.76/—	75.64/10	74.28/9	66.81/14	74.46/8
FedAdagrad	36.85/—	58.07/13	72.39/8	55.39/-	71.51/14
FedAdam	40.98/—	73.75/—	72.80/5	66.06/20	74.76/5
FedYogi	40.18/—	73.51/13	72.97/5	66.38/20	74.57/5
FedUV	43.97/—	**75.97**/3	76.98/2	**73.30**/8	**78.23**/**2**
GGC	**63.46**/—	74.69/**2**	75.85/**1**	67.02/**4**	76.52/**2**

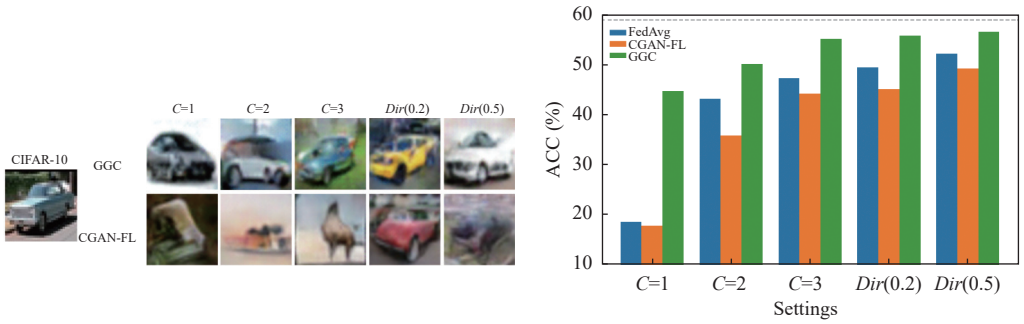
注: "/"前内容是ACC, "/"后内容是到达79.98%的0.8倍时所需轮次, 其中, 79.98%为IID时FedAvg全局模型的准确率

可见, GGC 在 AG-News 上的表现弱于图像数据集, 仅在 *C*=1 和 *Dir*(0.2) 两个场景下表现较好, 在大多数场景

下 FedUV 表现均较为优秀. 我们认为有以下两个原因导致该现象. (1) AG-News 是四分类任务, 在 C=3 和 Dir(0.5) 的场景中, 原始数据分布已经趋近于 IID, GGC 引入的合成样本没有有效的校正偏差反而带来了噪声, 这与在图像十分类的 Dir(0.5) 场景上表现较差的原因是相同的. (2) 原始的 GAN 难以捕捉到文本复杂的语义特征, 使合成样本的质量较多的低于原始样本, 需要寻求一种更高质量的文本生成方案来生成合成样本. 值得注意的是, GGC 的表现总是不远弱于 FedAvg, 并不像其他基线方法在某几个场景中会损害全局模型的性能 (如 SCAFFOLD 在 Dir(0.2) 中), 这进一步地证明了 GGC 的鲁棒性.

4.4 Q3: 分组生成对比实验

我们补充设置了一项对比实验, 证明分组生成能有效提高生成模型的质量, 进而使补充的样本在缓解数据分布偏差的同时不引入过多的干扰. 为了在不分组的前提下获得带标签的样本, 我们在所有客户端上联邦训练一个条件对抗生成网络 (conditional generative adversarial network, CGAN)[34], 该网络可以在输入时指定类别使生成器从噪声合成特定类别的样本. 我们将其网络结构参照本文实验所用结构进行同样更改, 并按照相同数目分发合成样本再进行任务模型的联邦学习, 记为 CGAN-FL. 如图 7(a) 展示了不同场景下 GGC 和 CGAN-FL 生成图像的差异. 可见二者生成的图片均区别于原始样本. 但 GGC 生成的图像质量更高, 并且在不同的原始数据分布下, 图像都较为稳定地保留了车的特征. 而 CGAN-FL 生成的图像模糊, 且随 js 的变化图像中包含的特征也产生波动, 如在 C=3 的场景下图像混入了鹿的特征, 在 C=1 时已经难以识别出合成的图片类别. 如图 7(b) 所示 (其中, 虚线表示 FedAvg 在 IID 场景下全局模型的准确率), 将这些低质量的图像加入本地训练集后均降低了全局模型的性能. 这是因为这些样本不仅没能带来其他类别的知识来指导本地模型训练, 而且混杂的特征干扰到对原始样本的特征识别. 与此同时, GGC 在所有场景下均能有效提升全局模型的表现, 证明了分组生成的必要性.



(a) 图像对比 (b) ACC对比

图 7 采用 GGC 或 CGAN-FL 带来的差异

4.5 Q4: 原始数据隐私安全实验

因存放在服务器上的全局生成模型是由各客户端的本地数据训练得到, 其生成的合成样本与原始数据可能相似, 故在本节实验中探讨合成样本和原始样本的差异性以证明本文方法对原始数据是隐私安全的.

为了量化比较两种样本的差异性, 本节选用结构相似性指数 (structural similarity index measure, SSIM) 和特征相似性指数 (feature similarity index measure, FSIM) 来作为相应的评价指标. SSIM 是一种衡量两幅图像相似度的重要方法, 它被广泛应用于图像质量评估和图像比较等领域, 其强调对图像结构信息的捕获, 基于亮度、对比度和结构来评价两幅图像的相似性. SSIM 的计算结果的取值范围为 [−1, 1], 其中 1 表示两幅图像完全相同, 0 表示两幅图像没有相似性, 负值表示两幅图像存在反相关. 一般可认为 $|\mathrm{SSIM}| < 0.4$ 时图像相似度较低. FSIM 与传统的图像质量评估方法 (如 MSE) 不同, 其主要基于图像的低级特征 (例如相位一致性和梯度幅度) 来衡量图像的相似性, 更加贴近人类视觉系统的感知方式. FSIM 的计算结果的取值范围为 [0, 1], 其中 1 表示两幅图像完全相同, 0 表示两幅图像没有相似性. 一般可认为 $|\mathrm{FSIM}| < 0.5$ 时图像相似度较低. 本节在 CIFAR-10 数据集上的 5 个场景展开实验, 每个场景中, 随机选取数据集中的 1 个样本, 随后使用该样本类别对应的生成器生成 1 000 个合成样本, 分别计

算真实样本和这 1 000 个合成样本的 SSIM 和 FSIM 的平均值, 并统计两幅图像相似度较低的个数, 结果如表 5 所示.

可见, 在所有场景中, 真实样本和随机生成的合成样本总是差异显著, 结合图 7 的各图像的人类肉眼感知 (两类图像也有显著区别), 可认为全局生成模型不会暴露原始数据的隐私, 本文提出的方法对原始样本是隐私安全的.

表 5　1 个真实样本和 1 000 个合成样本的相似性对比

指标	C=1	C=2	C=3	Dir(0.2)	Dir(0.5)
SSIM的均值	0.111	0.106	0.109	0.111	0.111
SSIM值落入相似区间的个数	0	0	0	0	0
FSIM的均值	0.092	0.090	0.090	0.092	0.091
FSIM值落入相似区间的个数	0	0	0	0	0

4.6 Q5: 参数影响实验

为了充分探讨本文提出的方法受各参数的影响状况, 分别在不同条件下进行实验, 来检验分析不同参数对全局模型性能表现的影响.

(1) 客户端数量 K 的影响

如图 8 所示, 我们将客户端数量增加到 20 个观察在更多客户端参与联邦学习时各方法的表现. 此时, 在基于数目的偏移场景下, 每类样本分散在更多的客户端中, 而在基于分布的偏移场景下每类样本分布并未产生显著变化. 因此所有方法在 C=1、C=2 和 C=3 的场景下全局模型的准确率略有上升, 而 Dir(0.2) 和 Dir(0.5) 场景下变化不大. 此外, 这也提高了生成模型的联邦学习效果, 导致训练出的生成模型质量更高, 因此 GGC 对全局模型的性能提升更加显著, 即 K = 20 时比 K = 10 时对比基线方法的优势更明显.

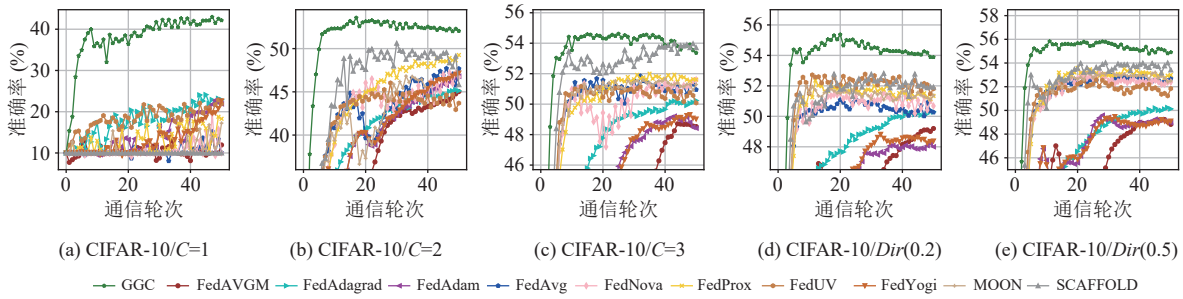


图 8　K =20 时全局模型表现随着轮次的变化

(2) 生成类别个数 n_L 的影响

n_L 控制了一共可以生成并分发多少类的合成样本. 我们取 n_L = 10, 9, 8, 7, 6 在 CIFAR-10 的 C=3 和 Dir(0.5) 两种分布情况下进行了实验, 结果如图 9 所示.

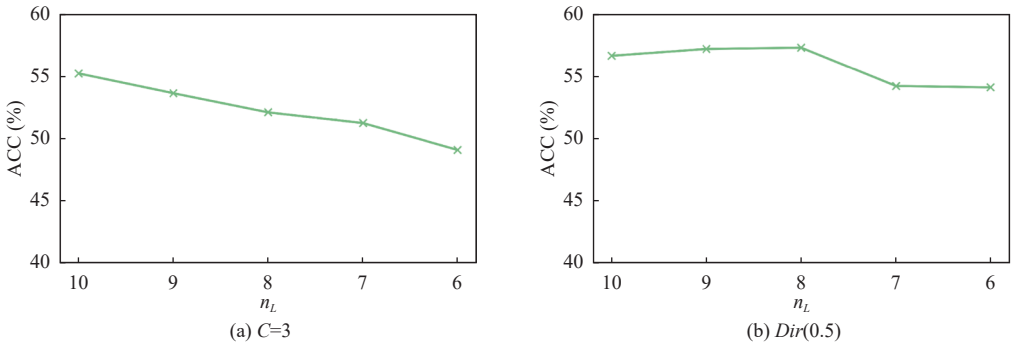


图 9　n_L 不同取值对性能的影响

在 C=3 的场景下, 因每类标签的分布没有显著区别, 均集中在少数客户端, 那么无论剔除哪类样本不合成并分发, 均会使全局模型更难泛化到这些未补充的类别样本, 导致在测试集上的准确率近似线性的下降. 而在 *Dir*(0.5) 的分布下, 有某几类标签均匀地分散在客户端中 (由图 9 可知在取 10、9、8 时准确率基本保持不变, 故此时应为 3 类样本在客户端中较为均匀散布), 这些样本并不需要合成并分发, 因此 GGC 放弃生成这些标签对全局模型性能几乎没有影响, 故随着 n_L 的减少, ACC 的变化趋势为先基本不变再突然下降. 这同时也验证了 GGC 标签选择策略的有效性.

(3) 合成样本数目比例 α 的影响

α 控制了合成样本和原始样本在客户端训练集中的数量占比, 它们以不同比例参与任务模型的训练将带来不同的效果, 我们取 α = 0.1, 0.2, 0.5, 1.0, 1.5, 在 CIFAR-10/C=2 进行了实验, 结果如表 6 所示.

表 6 α 的不同取值对性能的影响

α	ACC (%)
0.1	49.85
0.2	50.17
0.5	**50.94**
1.0	49.00
1.5	48.63

实验表明: α 的不同取值对准确率的影响较小, 模型性能波动不大. 其中, 在 α = 0.2 保持较低的开销时, 就能有效校正分布使全局模型达到较优的表现. 而在 α = 0.5 时达到了最优的性能, 但此时也补充了较多的合成样本增大了开销. 因此 α 取较小值 (小于 1.0) 相比与取较大值是一个更合理的选择. 同时注意到, α = 1.5 时相比 α = 1.0 的 ACC 已经开始下降, 这是因为合成样本的数目已经远超过了原始样本数目, 本地任务模型学习到了更多低质量的特征使全局模型在测试集上的泛化表现下降.

4.7 Q6: 正交实验

尽管我们通过补齐部分样本有效地校正了客户端本地标签分布的偏差, 但本地数据分布仍然是 Non-IID 的, 因此在联邦学习过程中对网络参数校正的方法仍然能起到一定作用. 为了验证 GGC 与基线方法的正交性, 我们在对任务模型的联邦学习训练中将 FedAvg 替换为 FedProx, 在 CIFAR-10 数据集上进行了正交实验.

如表 7 所示, 在所有 Non-IID 场景中 GGC 总能增强 FedProx 的性能, 使得到的全局模型的表现更接近于 IID, 这证明了 GGC 与基线方法的正交性. 此外, GGC 与 FedProx 结合在 C=2 和 *Dir*(0.5) 场景中优于 FedAvg 与 GGC 结合, 说明在不同场景下选择合适的方法进行结合, 可进一步提高全局模型的性能. 需要注意的是, FedProx 相比 FedAvg 在本地训练时引入了额外的计算, 而 FedProx+对 FedAvg+提升是有限的, 因此普通的 GGC 即能适应多数场景的需求.

表 7 正交试验 (%)

Method	C=1	C=2	C=3	*Dir*(0.2)	*Dir*(0.5)
FedAvg	18.67	43.25	47.39	49.53	52.29
FedAvg+	**44.84**	50.17	**55.22**	**55.92**	56.66
FedProx	18.67	42.77	48.19	51.54	52.05
FedProx+	44.29	**51.00**	55.20	54.21	**58.23**
up↑ (%)	137.22	19.24	14.54	5.18	11.87

注: 方法后的+号表示原方法与本文方法进行了结合, 最后一行列出了FedProx+相比FedProx对ACC的提升幅度

4.8 Q7: 额外计算代价对比

本文提出的方法产生的额外计算开销主要发生在以下 3 部分: 客户端训练生成模型、服务器生成合成样本、客户端使用合成样本进行任务模型本地训练. 本节先横向统计了这 3 部分的时间消耗, 之后纵向对比了各方法在

任务模型训练期间的理论上的计算代价.

以 Fashion-MNIST 数据集为例记录了各操作所用时间 (各阶段均在 GPU 上完成, 使用 Wall-clock), 如表 8 所示.

表 8 不同阶段花费的时间 (s)

操作	所需时间
生成模型迭代一个epoch (共6000个样本)	8
生成6000个合成样本	8
任务模型迭代一个epoch (共6000个样本)	4

需要说明的有以下 4 点: (1) 由于本文方法基于阈值对各客户端的数据进行了筛选, 每个客户端往往实际参与生成模型训练的样本数目约占总数的一半, 那么生成模型迭代一个 epoch 和任务模型迭代一个 epoch 的用时相差无几. (2) 本文方法仅需对数据分布进行一次性纠偏, 之后可用于多个联邦学习任务, 在这种情况下, 其计算时间开销可进一步忽略. (3) 综合表 3, 本文方法对任务模型仅需极少量的联邦学习轮次就能使全局模型性能超越其他基线方法的多轮联邦学习, 其能取得成效的原因可通俗地表示为: 用代价大致相当的, 数据独立同分布的生成模型联邦训练, “置换了”数据非独立同分布的任务模型的部分联邦训练过程. (4) 由于显卡是并行计算, 生成更多的合成样本的时间消耗不是线性增长, 故在更大的数据集上采用本文的方法带来的计算开销并不显著. 综上所述, 本文方法的计算代价在可以接受的范围内.

在任务模型训练期间各方法相比 FedAvg 带来的理论额外时间复杂度如表 9 所示, 其中 B 代表批大小, C 代表类别数, d 代表表征的维度数, w 代表模型大小, N 代表总训练批次数:

表 9 各方法任务模型训练时理论额外计算代价对比

方法	本地模型训练时	服务器模型聚合时
FedAvg	—	—
FedNova	—	$\mathrm{O}(N)$
FedProx	$\mathrm{O}(w\cdot N)$	—
MOON	$\mathrm{O}(d\cdot N)$	—
SCAFFOLD	$\mathrm{O}(w)$	$\mathrm{O}(w)$
FedAVGM	—	$\mathrm{O}(w)$
FedAdagrad	—	$\mathrm{O}\left(w^2\right)$
FedAdam	—	$\mathrm{O}\left(w^2\right)$
FedYogi	—	$\mathrm{O}\left(w^2\right)$
FedUV	$\mathrm{O}(C\cdot B\cdot N)+\mathrm{O}\left(B^2\cdot d\cdot N\right)$	—
GGC	$\mathrm{O}(\alpha\cdot N)$	—

实际上, 得益于显卡并行计算的能力, 各对比方法的实际开销均较小, 而本文方法由于是变相增加了训练集的规模, 则降低开销需要通过减小 α 或增大 B 使 N 减小实现.

5 总 结

本文针对联邦学习中非独立同分布 (Non-IID) 数据的标签分布偏移问题进行了深入研究, 这一问题在实际情境中非常常见, 对全局模型最后在测试集上的泛化表现有着关键性的影响. 现存的工作要么没有从校正数据分布的角度入手导致性能提升有限, 要么忽略了过程中对隐私的保护. 为了解决这些问题, 我们提出了新的联邦学习框架 GGC, 综合考虑隐私保护和标签分布偏移问题, 隐去了每个客户端具体的数据分布, 合理地挑选了若干标签, 有针对性地生成合成样本并分发, 有效缓解各客户端本地数据分布的偏差, 进而提升全局模型的性能. 实验结果表明, 我们的方法在多个场景和数据集下有效.

然而, GGC 一方面在全局客户端数据接近 IID 时表现一般, 且有可能会造成过拟合. 另一方面其会带来计算

等额外开销, 且其依赖于对每类样本的有效生成, 标签类别增多时会使开销线性增长. 在面对类别数较多的任务时, 可以遵循我们的隐私保护方法, 对标签进行分组和挑选, 但需要在后一阶段中寻求更有效的生成方案, 例如可将标签分布类似的客户端划分到一个更大的组中, 认为组内已经达到了某种程度上的数据同质, 在组内使用一个条件生成网络对多类样本进行生成来降低对生成模型个数的要求. 另一方面, 本文主要关注客户端数据分布相关的隐私保护问题, 但模型参数或合成样本也可能泄露客户端希望保护的隐私信息. 未来的研究可以进一步深入探讨与联邦学习相关的攻防领域, 探索加强本地模型和全局模型参数隐私保护的方法, 以防止通过这些参数推测出原始数据等, 从而更有效地实现联邦学习中的隐私保护目标. 将本文提出的方法与该领域现有技术结合是值得探索的方向.

References

[1] McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proc. of the 20th Int'l Conf. on Artificial Intelligence and Statistics. Fort Lauderdale: PMLR, 2017. 1273–1282.

[2] Li QB, Diao YQ, Chen Q, He BS. Federated learning on Non-IID data silos: An experimental study. In: Proc. of the 38th IEEE Int'l Conf. on Data Engineering. Kuala Lumpur: IEEE, 2022. 965–978. [doi: 10.1109/ICDE53745.2022.00077]

[3] Jeong E, Oh S, Kim H, Park J, Bennis M, Kim SL. Communication-efficient on-device machine learning: Federated distillation and augmentation under Non-IID private data. arXiv:1811.11479, 2023.

[4] Duan MM, Liu D, Chen XZ, Tan YJ, Ren JT, Qiao L, Liang L. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In: Proc. of the 37th IEEE Int'l Conf. on Computer Design. Abu Dhabi: IEEE, 2019. 246–254. [doi: 10.1109/ICCD46524.2019.00038]

[5] Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. In: Proc. of the 3rd Conf. on Machine Learning and Systems. Austin: MLSyS.org, 2020. 429–450.

[6] Li QB, He BS, Song D. Model-contrastive federated learning. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 10708–10717. [doi: 10.1109/CVPR46437.2021.01057]

[7] Li ZJ, Shao JW, Mao YY, Wang JH, Zhang J. Federated learning with GAN-based data synthesis for Non-IID clients. In: Proc. of the 1st Int'l Workshop on Trustworthy Federated Learning. Vienna: Springer, 2023. 17–32. [doi: 10.1007/978-3-031-28996-5_2]

[8] Liu BY, Guo Y, Chen XQ. PFA: Privacy-preserving federated adaptation for effective model personalization. In: Proc. of the 2021 Web Conf. Ljubljana: ACM, 2021. 923–934. [doi: 10.1145/3442381.3449847]

[9] Diao YQ, Li QB, He BS. Exploiting label skews in federated learning with model concatenation. In: Proc. of the 38th AAAI Conf. on Artificial Intelligence. Vancouver: AAAI Press, 2024. 11784–11792. [doi: 10.1609/aaai.v38i10.29063]

[10] Wu Q, Chen X, Zhou Z, Zhang JS. FedHome: Cloud-edge based personalized federated learning for in-home health monitoring. IEEE Trans. on Mobile Computing, 2022, 21(8): 2818–2832. [doi: 10.1109/TMC.2020.3045266]

[11] Yoon T, Shin S, Hwang SJ, Yang E. FedMix: Approximation of mixup under mean augmented federated learning. In: Proc. of the 9th Int'l Conf. on Learning Representations. OpenReview. net, 2021. 1–19.

[12] Zhang LL, Shen BC, Barnawi A, Xi S, Kumar N, Wu Y. FedDPGAN: Federated differentially private generative adversarial networks framework for the detection of COVID-19 pneumonia. Information Systems Frontiers, 2021, 23(6): 1403–1415. [doi: 10.1007/s10796-021-10144-6]

[13] Wang JY, Liu QH, Liang H, Joshi G, Poor HV. Tackling the objective inconsistency problem in heterogeneous federated optimization. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 7611–7623.

[14] Hsu TMH, Qi H, Brown M. Measuring the effects of non-identical data distribution for federated visual classification. arXiv:1909.06335, 2019.

[15] Reddi SJ, Charles Z, Zaheer M, Garrett Z, Rush K, Konečný J, Kumar S, McMahan HB. Adaptive federated optimization. In: Proc. of the 9th Int'l Conf. on Learning Representations. OpenReview.net, 2021. 1–38.

[16] Karimireddy SP, Kale S, Mohri M, Reddi SJ, Stich SU, Suresh AT. SCAFFOLD: Stochastic controlled averaging for federated learning. In: Proc. of the 37th Int'l Conf. on Machine Learning. JMLR.org, 2020. 5132–5143.

[17] Son HM, Kim MH, Chung TM, Huang C, Liu X. FedUV: Uniformity and variance for heterogeneous federated learning. In: Proc. of the 2024 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2024. 5863–5872. [doi: 10.1109/CVPR52733.2024.00560]

[18] Zhang L, Shen L, Ding L, Tao DC, Duan LY. Fine-tuning global model via data-free knowledge distillation for Non-IID federated

learning. In: Proc. of the 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 10164–10173. [doi: 10.1109/CVPR52688.2022.00993]

[19] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Proc. of the 28th Int'l Conf. on Neural Information Processing Systems. Montreal: MIT Press, 2014. 2672–2680.

[20] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proc. of the 4th Int'l Conf. on Learning Representations. San Juan, 2016. 1–16.

[21] Gu YH, Bai YB. Survey on security and privacy of federated learning models. Ruan Jian Xue Bao/Journal of Software, 2023, 34(6): 2833–2864 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6658.htm [doi: 10.13328/j.cnki.jos.006658]

[22] Liu XQ, Xu F, Ma Z, Yuan M, Qian HW. Research on privacy protection technology in federated learning. Journal of Information Security Research, 2024, 10(3): 194–201 (in Chinese with English abstract). [doi: 10.12379/j.issn.2096-1057.2024.03.01]

[23] Tang LT, Chen ZN, Zhang LF, Wu D. Research progress of privacy issues in federated learning. Ruan Jian Xue Bao/Journal of Software, 2023, 34(1): 197–229 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6411.htm [doi: 10.13328/j.cnki.jos.006411]

[24] Dwork C, Roth A. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 2014, 9(3-4): 211–487. [doi: 10.1561/0400000042]

[25] Gong MG, Xie Y, Pan K, Feng KY, Qin AK. A survey on differentially private machine learning [review article]. IEEE Computational Intelligence Magazine, 2020, 15(2): 49–64. [doi: 10.1109/MCI.2020.2976185]

[26] Sweeney L. k-Anonymity: A model for protecting privacy. Int'l Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(5): 557–570. [doi: 10.1142/S0218488502001648]

[27] Yao AC. Protocols for secure computations. In: Proc. of the 23rd Annual Symp. on Foundations of Computer Science. Chicago: IEEE, 1982. 160–164. [doi: 10.1109/SFCS.1982.38]

[28] Evans D, Kolesnikov V, Rosulek M. A pragmatic introduction to secure multi-party computation. Foundations and Trends® in Privacy and Security, 2018, 2(2–3): 70–246. [doi: 10.1561/3300000019]

[29] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training GANs. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016. 2234–2242.

[30] How to train a GAN? Tips and tricks to make GANs work. 2018. https://github.com/soumith/ganhacks

[31] Differential Privacy Team, Apple. Learning with privacy at scale. 2017. https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf

[32] Le QV, Mikolov T. Distributed representations of sentences and documents. In: Proc. of the 31st Int'l Conf. on Machine Learning. Beijing: JMLR.org, 2014. II-1188–II-1196.

[33] Lau JH, Baldwin T. An empirical evaluation of doc2vec with practical insights into document embedding generation. In: Proc. of the 1st Workshop on Representation Learning for NLP. Berlin: ACL, 2016. 78–86. [doi: 10.18653/v1/W16-1609]

[34] Mirza M, Osindero S. Conditional generative adversarial nets. arXiv:1411.1784, 2014.

附中文参考文献

[21] 顾育豪, 白跃彬. 联邦学习模型安全与隐私研究进展. 软件学报, 2023, 34(6): 2833–2864. http://www.jos.org.cn/1000-9825/6658.htm [doi: 10.13328/j.cnki.jos.006658]

[22] 刘晓迁, 许飞, 马卓, 袁明, 钱汉伟. 联邦学习中的隐私保护技术研究. 信息安全研究, 2024, 10(3): 194–201. [doi: 10.12379/j.issn.2096-1057.2024.03.01]

[23] 汤凌韬, 陈左宁, 张鲁飞, 吴东. 联邦学习中的隐私问题研究进展. 软件学报, 2023, 34(1): 197–229. http://www.jos.org.cn/1000-9825/6411.htm [doi: 10.13328/j.cnki.jos.006411]

作者简介

周众泽, 硕士生, 主要研究领域为隐私计算, 联邦学习.

张俊, 博士生, 主要研究领域为高效机器学习, 大语言模型, 多模态大模型.

寿黎但, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为非结构化数据管理, 数据智能, 多媒体挖掘.

陈珂, 博士, 副研究员, CCF 专业会员, 主要研究领域为大数据处理, 数据挖掘, 隐私保护.

陈刚, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为数据库, 大数据管理系统, 大数据智能计算.