# Federated Learning on Non-IID Data Silos: An Experimental Study

丁子倩

**摘要**

由于日益增加的隐私问题和数据法规，训练数据已经越来越分散，形成多个“数据孤岛”的分布式数据库（例如，在不同的组织和国家）。为了开发有效的机器学习服务，必须在不交换原始数据的情况下利用这些分布式数据库中的数据。最近，联邦学习（FL）已经成为一种越来越受关注的解决方案，它使多方能够在不交换本地数据的情况下协作训练机器学习模型。分布式数据库面临的一个关键和共同的挑战是各方之间数据分布的异构性。不同方的数据通常是非独立且相同分布的（即，非IID）。已经有许多FL算法来解决非IID数据设置下的学习有效性。然而，缺乏一个实验研究，系统地了解他们的优点和缺点，由于以往的研究都采用非常严格的数据划分策略，很难代表性和全面性。为了帮助研究人员更好地理解和研究联邦学习中的非IID数据设置，本文提出了全面的数据划分策略来覆盖典型的非IID数据情况。此外，我们进行了广泛的实验，以评估国家的最先进的FL算法。我们发现，非IID的FL算法的学习精度带来了重大挑战，现有的最先进的FL算法在所有情况下都优于其他算法。我们的实验为未来解决“数据孤岛”挑战的研究提供了见解。

**关键词：** FL算法；实验研究；数据划分策略

## 1 引言

随着隐私问题和数据法规的日益增加，训练数据变得越来越分散，形成了多个“数据孤岛”的分布式数据库。这些分布式数据库存在于不同的组织和国家之间。然而，为了开发有效的机器学习服务，我们需要在不交换原始数据的情况下利用这些分布式数据库中的数据。联邦学习（FL）是一种解决方案，它允许多方在不交换本地数据的情况下协作训练机器学习模型。然而，在处理非独立且相同分布（非IID）的数据时，分布式数据库面临着挑战。

目前已经有一些FL算法用于解决非IID数据设置下的学习有效性问题。然而，以往的研究存在一些限制：缺乏系统性的实验研究，无法全面了解这些算法的优点和缺点；数据划分策略过于僵化，无法代表性和彻底地反映出真实的非IID数据情况。

该论文旨在帮助研究人员更好地理解和研究联邦学习中的非IID数据设置。为此，论文提出了全面的数据划分策略，以覆盖典型的非IID数据情况，并进行广泛的实验评估国际上最先进的FL算法。通过这些实验，研究人员可以更深入地了解非IID的FL算法在不同情况下的学习精度和效果，并为未来解决“数据孤岛”挑战提供见解。该研究对于促进联邦学习的发展，推动解决隐私问题和数据法规带来的挑战具有重要意义。

## 2 相关工作

通过对非IID数据问题的探讨、FL算法的改进、数据划分策略的研究以及实验研究的评估，为解决非IID数据设置下的联邦学习挑战提供了有用的见解和方法。这些工作为研究人员提供了指导，以更好地处理非IID数据并改善联邦学习的性能。

## 2.1 非IID数据问题

探讨了非独立且相同分布（非 IID）的数据在联邦学习中的应用。这种情况下，不同方（组织或国家）之间的数据存在异构性，形成了数据孤岛。研究人员针对这个问题进行了深入研究，旨在理解非 IID 数据的特点以及它们对联邦学习的挑战。数据 IID 和 Non-IID 设置下，FL 中 local model 聚合并且经过多个 round 进行训练的 global model 结果：

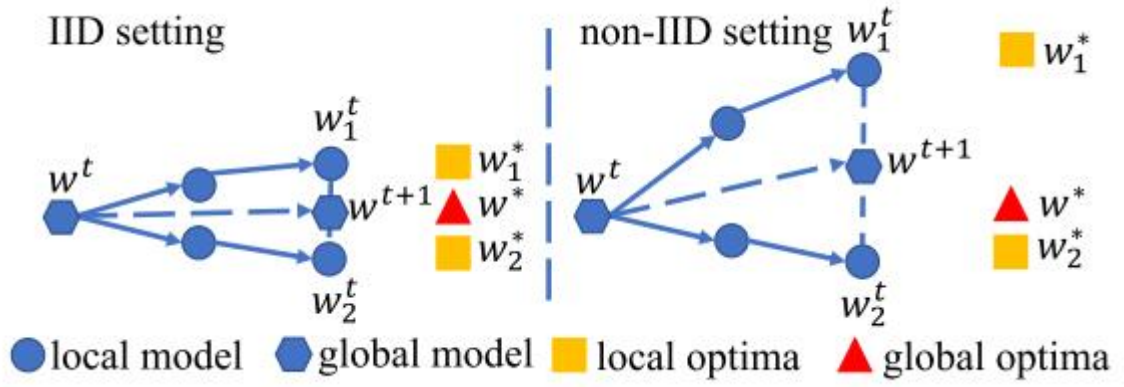



图 2.1 在非IID数据设置下FedAvg的问题

## 2.2 联邦学习算法

重点是研究和开发适用于非 IID 数据设置的 FL 算法。FL 算法旨在实现多方之间协作训练机器学习模型的方法，而无需共享原始数据。研究人员将尝试提出新的 FL 算法或改进现有算法，以克服非 IID 数据所带来的学习效果下降的问题。经典的 FedAvg 的算法图，之后的大部分 FL 算法框架基于此进行优化改进对于 FedAvg 来说提供了一个协同训练的有效标准，并且在不同 round 的训练过程中将计算开销分担给多个本地客户端来减少中央服务器的计算开销并且减少协同训练的通信开销。

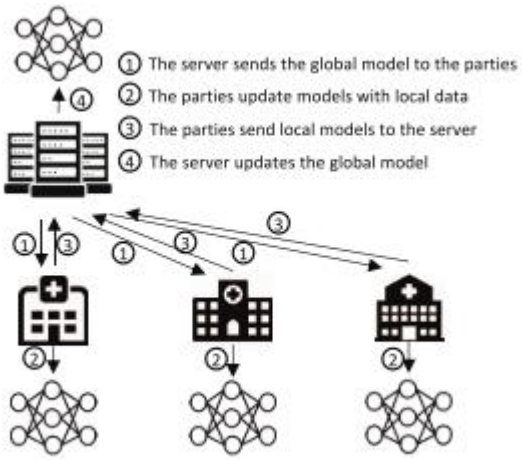



图 2.2 FedAvg框架

### 2.2.1 FedProx （Algorithm 1）

限制本地模型的参数更新（在本地模型的目标函数中引入一个 ℓ 2 正则化项来控制本地模型和全局模型的距离）—> 直观上的方法，来限制本地模型不要过度远离 global optima。

其优点在于相较于 FedAvg 来说没有引入额外不可接受的计算开销，因此从通信开销角度来看是高效的。

缺点在于每一个 local model 在进行训练的时候需要对 ℓ 2 正则化项的权重参数进行调优实现较好的效果。

### 2.2.2 FedNova （Algorithm 1）

基于 FedAvg 优化中央服务器和本地客户端进行模型聚合过程。考虑不同的本地模型贡献不同（local model 中的 different mini-batches 可能导致贡献不同）。在 FedAvg 中如果某一个本地模型存在较多的 train data 会导致模型聚合产生的 global model 产生偏好，FedNova 基于本地训练的 local steps 引入一个正则化项来限制对 global model 的影响。

同样该模型的优点是在 FedAvg 基础上仅增加有限的计算开销实现了较好的效果。

**Algorithm 1:** A summary of FL algorithms including FedAvg/FedProx/FedNova. We use red and orange colors to mark the part specially included in FedProx and FedNova, respectively.

**Input:** local datasets $\mathcal{D}^i$, number of parties $N$, number of communication rounds $T$, number of local epochs $E$, learning rate $\eta$

**Output:** The final model $w^T$

**Server executes**:
initialize $x^0$
**for** $t = 0, 1, ..., T-1$ **do**
  Sample a set of parties $S_t$
  $n \leftarrow \sum_{i \in S_t} |\mathcal{D}^i|$
  **for** $i \in S_t$ **in parallel do**
    send the global model $w^t$ to party $P_i$
    $\Delta w_i^t, \tau_i \leftarrow$ **LocalTraining**$(i, w^t)$
  For FedAvg/FedProx:
    $w^{t+1} \leftarrow w^t - \eta \sum_{i \in S_t} \frac{|\mathcal{D}^i|}{n} \Delta w_k^t$
  For FedNova:
    $w^{t+1} \leftarrow w^t - \eta \frac{\sum_{i \in S_t} |\mathcal{D}^i| \tau_i}{n} \sum_{i \in S_t} \frac{|D^i| \Delta w_i^t}{n \tau_i}$
return $w^T$

**Party executes**:
For FedAvg/FedNova: $L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y)$
For FedProx:
  $L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y) + \frac{\mu}{2} \|w - w^t\|^2$
**LocalTraining**$(i, w^t)$:
$w_i^t \leftarrow w^t$
$\tau_i \leftarrow 0$
**for** epoch $k = 1, 2, ..., E$ **do**
  **for** each batch $\mathbf{b} = \{\mathbf{x}, y\}$ of $\mathcal{D}^i$ **do**
    $w_i^t \leftarrow w_i^t - \eta \nabla L(w_i^t; \mathbf{b})$
    $\tau_i \leftarrow \tau_i + 1$
$\Delta w_i^t \leftarrow w^t - w_i^t$
return $\Delta w_i^t$, $\tau_i$ to the server

图 2.3 Algorithm 1

### 2.2.3 SCAFFOLD （Algorithm 2）

将 Non-IID 建模为数据持有者之间的差异，并且引入方差减少技术处理该问题。在实现中，引入服务器和本地客户端的控制变量，用于估计服务器模型的更新方向和每个客户端的更新方向。然后，用这两个更新方向的差异来近似局部训练的偏移。SCAFFOLD 在全局模型处计算局部数据的梯度或通过重复使用先前计算的梯度来个更新局部控制变量。计算开销更小。

**Algorithm 2:** The SCAFFOLD algorithm. We use blue color to mark the part specially included in SCAFFOLD compared with FedAvg.

**Input:** same as Algorithm 1
**Output:** The final model $w^T$

**Server executes:**
initialize $x^0$
$c^t \leftarrow \mathbf{0}$
**for** $t = 0, 1, ..., T-1$ **do**
  Randomly sample a set of parties $S_t$
  $n \leftarrow \sum_{i \in S_t} |\mathcal{D}^i|$
  **for** $i \in S_t$ **in parallel do**
    send the global model $w^t$ to party $P_i$
    $\Delta w_i^t, \Delta c \leftarrow$ **LocalTraining**$(i, w^t, c^t)$
  $w^{t+1} \leftarrow w^t - \eta \sum_{i \in S_t} \frac{|\mathcal{D}^i|}{n} \Delta w_k^t$
  $c^{t+1} \leftarrow c^t + \frac{1}{N} \Delta c$
return $w^T$

**Party executes:**
$L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y)$
$c_i \leftarrow \mathbf{0}$
**LocalTraining**$(i, w^t, c^t)$:
$w_i^t \leftarrow w^t$
$\tau_i \leftarrow 0$
**for** epoch $k = 1, 2, ..., E$ **do**
  **for** each batch $\mathbf{b} = \{\mathbf{x}, y\}$ of $\mathcal{D}^i$ **do**
    $w_i^t \leftarrow w_i^t - \eta(\nabla L(w_i^t; \mathbf{b}) - c_i^t + c)$
    $\tau_i \leftarrow \tau_i + 1$
$\Delta w_i^t \leftarrow w^t - w_i^t$
$c_i^* \leftarrow (i) \nabla L(w_i^t), or (ii) c_i - c + \frac{1}{\tau_i \eta}(w^t - w_i^t)$
$\Delta c \leftarrow c_i^* - c_i$
$c_i \leftarrow c_i^*$
return $\Delta w_i^t$, $\Delta c$ to the server

图 2.4 Algorithm 2

### 2.2.4 FedDyn

在本地模型更新时基于 global model 和前几轮通信中的模型添加正则化项。

### 2.2.5 FedBN

针对 feature distribution skew（属于 Non-IID 数据分布的一种情况，后面会详细介绍），在客户端上传模型参数之前在本地执行 batch-norm layers。

### 2.2.6 Addressing Class Imbalance in Federated Learning

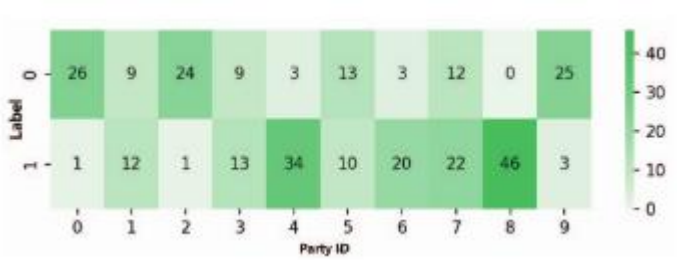
增加一个 monitor 来监控 imbalance class distribution 并基于此提出新的损失函数。

### 2.2.7 MOON: model-contrastive federated learning

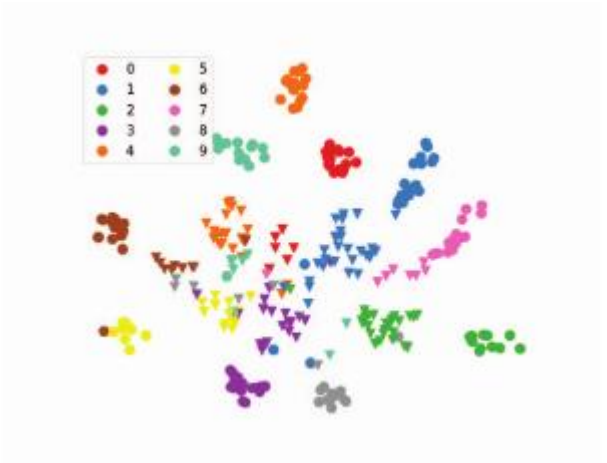
基于模型对比的思想，通过比较 current local model、local model from the previous round、global model 来修正本地模型训练的结果。

## 2.3 数据划分策略

为了解决非 IID 数据设置的挑战，论文提出了全面的数据划分策略。这些策略考虑了数据的异质性，并致力于确保代表性和彻底地反映真实的非 IID 数据情况。通过合理地划分数据（标签分布偏斜、特征分布偏斜、相同的标签但不同的特征、相同的特征但不同的标签、数量偏斜），研究人员可以模拟真实场景并提高联邦学习的效果。




(a) Criteo的标签分发




(b) Digits的功能分布

图 2.5 Criteo 和 Digits 的非 IID 属性

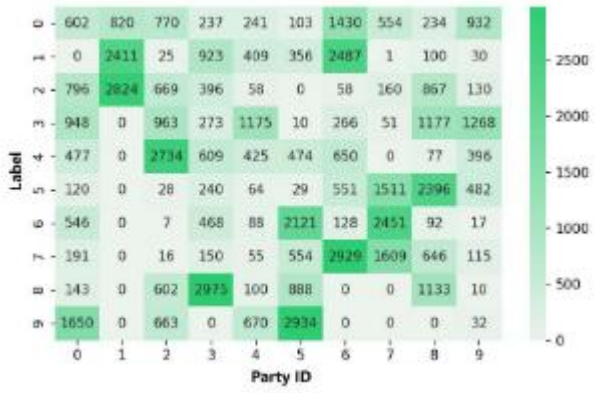



图 2.6 MNIST数据集上基于分布的标签不平衡分区示例

### 2.4 实验研究

进行了广泛的实验研究，评估了在非IID数据设置下国际上最先进的FL算法的性能。通过实验对比和分析，研究人员揭示了各种FL算法在非IID数据情况下的优点和缺点。这些实验有助于深入理解FL算法在不同情况下的表现，并为研究人员提供指导，以选择适合特定非IID数据场景的FL算法。

## 3 本文方法

### 3.1 本文方法概述

为了研究现有FL算法在非IID数据集上的有效性，在公共数据集上进行了广泛的实验。对于图像数据集，我们使用CNN，它有两个 5 x 5 卷积层，然后是 2 x 2 最大池化（第一个有6个通道，第二个有16个通道）和两个带有ReLU激活的全连接层（第一个有120个单元，第二个有84个单元）。对于表格数据集，我们使用具有三个隐藏层的MLP。三层的隐藏单元数分别为32、16和8。默认情况下，参与方数设置为10，FCUBE除外，其参与方数设置为4。每一轮都有各方参与，以消除默认方抽样带来的随机性影响。使用SGD优化器，rcv1的学习率为0.1，其他数据集的学习率为0.01（从{0.1，0.01，0.001}调整），动量为0.9。默认情况下，批量大小设置为64，本地epoch的数量设置为10。我们使用测试数据集上的top-1准确度作为度量来比较所研究的算法。为了公平比较，将所有研究的算法运行相同的轮数。除非指定，否则默认情况下回合数设置为50。

现有研究中的实验设置和我们的基准。请注意，现有研究中的基于数量、基于噪声和数量偏斜的划分策略与我们研究中提出的策略不同。

| Partitioning strategies | | FedAvg | FedProx | SCAFFOLD | FedNova | NIID-Bench |
|---|---|---|---|---|---|---|
| Label distribution skew | quantity-based | ✓ | ✓ | ✗ | ✗ | ✓ |
| | distribution-based | ✗ | ✗ | ✓ | ✓ | ✓ |
| Feature distribution skew | noise-based | ✗ | ✗ | ✗ | ✗ | ✓ |
| | synthetic | ✗ | ✓ | ✗ | ✗ | ✓ |
| | real-world | ✗ | ✓ | ✗ | ✗ | ✓ |
| Quantity skew | | ✗ | ✗ | ✗ | ✓ | ✓ |

图 3.1 实验设置和基准

为了研究现有FL算法在非IID数据集上的有效性，我们在九个公共数据集上进行了广泛的实验，包括六个图像数据集。

| Datasets | #training instances | #test instances | #features | #classes |
|---|---|---|---|---|
| MNIST | 60,000 | 10,000 | 784 | 10 |
| FMNIST | 60,000 | 10,000 | 784 | 10 |
| CIFAR-10 | 50,000 | 10,000 | 1,024 | 10 |
| SVHN | 73,257 | 26,032 | 1,024 | 10 |
| adult | 32,561 | 16,281 | 123 | 2 |
| rcv1 | 15,182 | 5,060 | 47,236 | 2 |
| covtype | 435,759 | 145,253 | 54 | 2 |
| FCUBE | 4,000 | 1,000 | 3 | 2 |
| FEMNIST | 341,873 | 40,832 | 784 | 10 |

图 3.2 数据集的统计

根据观察，绘制了一个决策树来总结每个非IID设置的合适FL算法，如图 3.3 所示。该决策树有助于用户根据非IID分布和数据集选择学习算法。

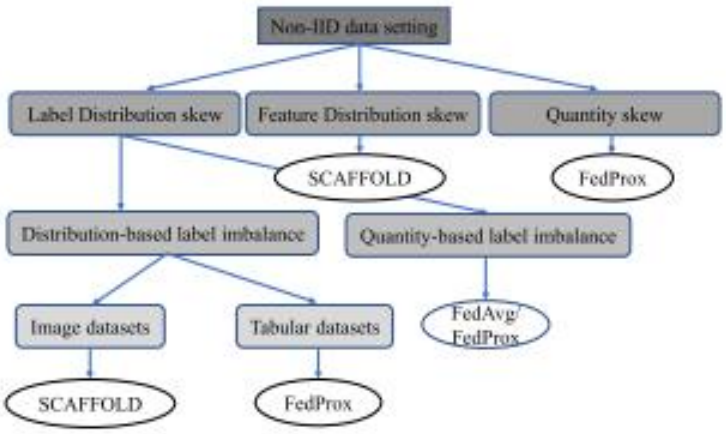



图 3.3 在非IID设置下确定最佳FL算法的决策树

在几个非 IID 数据设置上的现有算法的准确性与在均匀设置上的现有算法的准确性之间存在差距。

| category | dataset | partitioning | FedAvg | FedProx | SCAFFOLD | FedNova |
|---|---|---|---|---|---|---|
| Label distribution skew | MNIST | $p_k \sim Dir(0.5)$ | 98.9%±0.1% | 98.9%±0.1% | **99.0%±0.1%** | 98.9%±0.1% |
| | | $\#C = 1$ | 29.8%±7.9% | **40.9%±23.1%** | 9.9%±0.2% | 39.2%±22.1% |
| | | $\#C = 2$ | **97.0%±0.4%** | 96.4%±0.3% | 95.9%±0.3% | 94.5%±1.5% |
| | | $\#C = 3$ | **98.0%±0.2%** | 97.9%±0.4% | 96.6%±1.5% | **98.0%±0.3%** |
| | FMNIST | $p_k \sim Dir(0.5)$ | 88.1%±0.6% | 88.1%±0.9% | 88.4%±0.5% | **88.5%±0.5%** |
| | | $\#C = 1$ | 11.2%±2.0% | **28.9%±3.9%** | 12.8%±4.8% | 14.8%±5.9% |
| | | $\#C = 2$ | **77.3%±4.9%** | 74.9%±2.6% | 42.8%±28.7% | 70.4%±5.1% |
| | | $\#C = 3$ | 80.7%±1.9% | **82.5%±1.9%** | 77.7%±3.8% | 78.9%±3.0% |
| | CIFAR-10 | $p_k \sim Dir(0.5)$ | 68.2%±0.7% | 67.9%±0.7% | **69.8%±0.7%** | 66.8%±1.5% |
| | | $\#C = 1$ | 10.0%±0.0% | **12.3%±2.0%** | 10.0%±0.0% | 10.0%±0.0% |
| | | $\#C = 2$ | 49.8%±3.3% | **50.7%±1.7%** | 49.1%±1.7% | 46.5%±3.5% |
| | | $\#C = 3$ | **58.3%±1.2%** | 57.1%±1.2% | 57.8%±1.4% | 54.4%±1.1% |
| | SVHN | $p_k \sim Dir(0.5)$ | 86.1%±0.7% | 86.6%±0.9% | **86.8%±0.3%** | 86.4%±0.6% |
| | | $\#C = 1$ | 11.1%±0.0% | **19.6%±0.0%** | 6.7%±0.0% | 10.6%±0.8% |
| | | $\#C = 2$ | **80.2%±0.8%** | 79.3%±0.9% | 62.7%±11.6% | 75.4%±4.8% |
| | | $\#C = 3$ | 82.0%±0.7% | **82.1%±1.0%** | 77.2%±2.0% | 80.5%±1.2% |
| | adult | $p_k \sim Dir(0.5)$ | 78.4%±0.9% | **80.5%±0.7%** | 76.4%±0.0% | 52.3%±26.7% |
| | | $\#C = 1$ | **82.5%±2.2%** | 76.4%±0.0% | 23.6%±0.0% | 50.8%±0.9% |
| | rcv1 | $p_k \sim Dir(0.5)$ | 48.2%±0.7% | **70.3%±13.3%** | 64.4%±24.3% | 49.3%±2.1% |
| | | $\#C = 1$ | **51.8%±0.7%** | **51.8%±0.7%** | **51.8%±0.7%** | **51.8%±0.7%** |
| | covtype | $p_k \sim Dir(0.5)$ | **77.2%±7.4%** | 70.9%±0.7% | 67.7%±14.9% | 74.8%±12.9% |
| | | $\#C = 1$ | 48.8%±0.1% | **59.1%±2.1%** | 49.6%±1.4% | 50.4%±1.4% |
| number of times that performs the best | | | 8 | 11 | 4 | 3 |
| Feature distribution skew | MNIST | $\hat{\mathbf{x}} \sim Gau(0.1)$ | **99.1%±0.1%** | **99.1%±0.1%** | **99.1%±0.1%** | **99.1%±0.1%** |
| | FMNIST | | 89.1%±0.3% | 89.0%±0.2% | **89.3%±0.0%** | 89.0%±0.1% |
| | CIFAR-10 | | 68.9%±0.3% | 69.3%±0.2% | **70.1%±0.2%** | 68.5%±1.3% |
| | SVHN | | **88.1%±0.5%** | **88.1%±0.2%** | **88.1%±0.4%** | **88.1%±0.4%** |
| | FCUBE | synthetic | **99.8%±0.2%** | **99.8%±0.0%** | 99.7%±0.3% | 99.7%±0.1% |
| | FEMNIST | real-world | **99.4%±0.0%** | 99.3%±0.1% | **99.4%±0.1%** | 99.3%±0.1% |
| number of times that performs the best | | | 4 | 3 | 5 | 2 |
| Quantity skew | MNIST | $q \sim Dir(0.5)$ | **99.2%±0.1%** | **99.2%±0.1%** | 99.1%±0.1% | 99.1%±0.1% |
| | FMNIST | | 89.4%±0.1% | **89.7%±0.3%** | 88.8%±0.4% | 86.1%±2.9% |
| | CIFAR-10 | | **72.0%±0.3%** | 71.2%±0.6% | 62.4%±4.1% | 10.0%±0.0% |
| | SVHN | | 88.3%±1.0% | **88.4%±0.4%** | 11.0%±7.4% | 41.3%±21.1% |
| | adult | | 82.2%±0.1% | **84.8%±0.2%** | 81.6%±4.5% | 43.2%±33.9% |
| | rcv1 | | 96.7%±0.3% | **96.8%±0.4%** | 49.0%±1.9% | 51.8%±0.7% |
| | covtype | | **88.1%±0.2%** | 84.6%±0.2% | 63.2%±20.8% | 51.2%±3.2% |
| number of times that performs the best | | | 3 | 5 | 0 | 0 |
| Homogeneous partition | MNIST | IID | 99.1%±0.1% | 99.1%±0.1% | **99.2%±0.0%** | 99.1%±0.1% |
| | FMNIST | | 89.6%±0.3% | 89.5%±0.2% | **89.7%±0.2%** | 89.4%±0.2% |
| | CIFAR-10 | | 70.4%±0.2% | 70.2%±0.1% | **71.5%±0.3%** | 69.5%±1.0% |
| | SVHN | | **88.5%±0.5%** | **88.5%±0.8%** | 88.0%±0.8% | 88.4%±0.5% |
| | FCUBE | | 99.7%±0.1% | 99.6%±0.2% | 99.8%±0.1% | **99.9%±0.1%** |
| | FEMNIST | | 99.3%±0.1% | **99.4%±0.1%** | **99.4%±0.0%** | 99.3%±0.0% |
| | adult | | 82.6%±0.4% | **84.8%±0.2%** | 83.8%±2.5% | 82.6%±0.0% |
| | rcv1 | | **96.8%±0.4%** | 96.6%±0.6% | 80.9%±27.8% | 96.6%±0.4% |
| | covtype | | 87.9%±0.1% | 85.2%±0.0% | **88.0%±2.3%** | 87.9%±0.2% |
| number of times that performs the best | | | 2 | 3 | 5 | 1 |

图 3.4 不同方法的最佳准确度

### 3.2 特征分布偏斜模块

在特征分布偏斜中，尽管知识 P（yi）在各方之间变化，但是特征分布 P（xi）在各方之间变化。P(yi|xi)相同。例如，猫在不同的区域可能会有不同的毛色和图案。在这里，我们引入三种不同的设置来模拟特征分布偏斜：基于噪声的特征不平衡，合成特征不平衡和真实世界特征不平衡。

基于噪声的特征不平衡：首先将整个数据集随机平均地分为多方。对于每一方，我们将不同水平的高斯噪声添加到其本地数据集中，以实现不同的特征分布。具体而言，给定用户定义的噪声水平 σ，我们为参与方 Pi 添加噪声 x~ Gau（σ • i/N），其中 Gau（σ • i/N）是均值为 0 且方差为 σ • i/N 的高斯分布。用户可以改变 σ 以增加各方之间的特征相异度。图 3.5 是 FMNIST 数据集上基于噪声的特征不平衡的示例。为了便于展示，我们使用 x ~Gau（σ）来展示这样的划分策略。左图，从 Gau（0.001）采样的噪声被添加到其图像中。右图，从 Gau（0.01）采样的噪声被添加到其图像中。

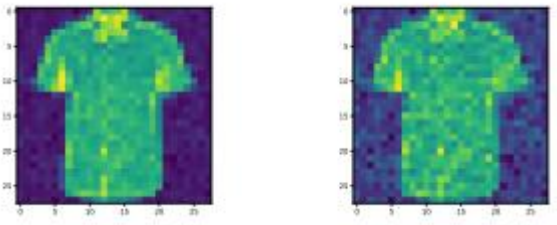


图 3.5 FMNIST 数据集上添加噪声的示例

合成特征不平衡：我们生成一个合成的特征不平衡联邦数据集命名为 FCUBE。假设数据点的分布是一个三维立方体（即（x1，x2，x3）），它有两个不同的标签，由平面 x1 =0 分类。如图 3.6 所示，我们通过平面 x1 =0、x2 =0 和 x3 =0 将立方体分成 8 部分。然后，我们分配两个对称的部分（0，0，0）的一个子集为每一方。通过这种方式，各方之间的特征分布不同，而标签仍然是平衡的。

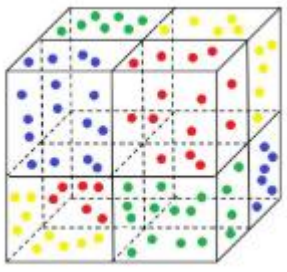


图 3.6 FCUBE 数据集的可视化

真实世界特征不平衡：真实世界的特征不平衡：EMNIST数据集收集了来自不同作者的手写字符/数字。然后根据作者将数据集划分为不同的部分是很自然的。由于书写者之间的字符特征通常不同（例如，笔画宽度、倾斜度），因此在不同方之间存在自然的特征分布偏斜。具体来说，对于EMNIST的数字图像，我们将作者（及其数字）随机平均地划分并分配给每一方。由于每一方都有不同的作者，所以特征分布在各方之间是不同的。我们将此联邦数据集称为FEMNIST。

### 3.3 损失函数定义

损失函数使用主要是在模型的训练阶段，每个批次的训练数据送入模型后，通过前向传播输出预测值，然后损失函数会计算出预测值和真实值之间的差异值，也就是损失值。得到损失值之后，模型通过反向传播去更新各个参数，来降低真实值与预测值之间的损失，使得模型生成的预测值往真实值方向靠拢，从而达到学习的目的。损失函数就是用来度量模型的预测值f(x)与真实值Y的差异程度的运算函数，它是一个非负实值函数，通常使用L(Y, f(x))来表示，损失函数越小，模型的鲁棒性就越好。

对本地更新的鲁棒性：

局部历元的数量可以对现有算法的精度有很大的影响。局部历元数的最优值对非IID分布非常敏感。

一方面，我们可以发现局部历元的数目对FL算法的精度有很大的影响。例如，当#C =2时，当局部历元的数量设置为80时，所有算法的准确度通常显著降低。

另一方面，局部时期的最佳数量在不同的设置中不同。例如，当#C =1和#C =2时，对于FedAvg，局部时期的最佳数量是20，而对于设置pk = Dir（0.5）和 #C =3。总之，现有的算法对大的局部更新不够鲁棒。必须考虑非IID分布以确定局部时期的最佳数量。

## 4 复现细节

### 4.1 与已有开源代码对比

参考本论文的源代码。设计自己的数据加载器。编写一个数据加载器以元组形式返回数据集（X_train、y_train、X_test、y_test）。得到 （X_train， y_train， X_test， y_test） 后，该函数将返回 .partition.py partition_data utils.pyMNIST_truncated dataset.pypartition_datanet_dataidx_map

### 4.2 实验环境搭建

操作系统：Linux

开发语言：Python 3.8.13

内存：220G

在服务器运行 requirements.txt。

requirements.txt 里面包括：

scikit-learn==0.22.1

numpy==1.18.1

scipy==1.4.1

torch==1.1.0

torchvision==0.3.0

pandas==0.24.2

requests==2.23.0

### 4.3 使用说明

下面是运行代码的一个示例：

```
python experiments.py --model=simple-cnn \
   --dataset=cifar10 \
```

```
--alg=fedprox \
--lr=0.01 \
--batch-size=64 \
--epochs=10 \
--n_parties=10 \
--mu=0.01 \
--rho=0.9 \
--comm_round=50 \
--partition=noniid-labeldir \
--beta=0.5\
--device='cuda:0'\
--datadir='./data/' \
--logdir='./logs/' \
--noise=0 \
--sample=1 \
--init_seed=0
```

| 参数 | 描述 |
| --- | --- |
| model | 模型体系结构。选项：。默认值 =。simple-cnn vgg resnet mlp mlp |
| dataset | 要使用的数据集。选项：。默认值 =。mnist cifar10 fmnist svhn generated femnist a9a rcv1 covtype mnist |
| alg | 训练算法。选项：。默认值 =。fedavg fedprox scaffold fednova moon fedavg |
| lr | 局部模型的学习率，默认值 =。0.01 |
| batch-size | 批大小，默认值 =。64 |
| epochs | 本地训练周期数，默认值 =。5 |
| n_parties | 参与方数量，默认值 =。2 |
| mu | FedProx 的近端术语参数，默认值 =。0.001 |
| rho | 控制动量 SGD 的参数，默认值 =。0 |
| comm_round | 要使用的通信轮数，默认值 =。50 |
| partition | 分区方式。选项：、、（或 2、3、…，表示每一方拥有的固定数量的标签）、、默认值 = homo noniid-labeldir noniid-#label1 real iid-diff-quantity homo |
| beta | 异构分区的狄利克雷分布的浓度参数，默认值 =。0.5 |
| device | 指定运行程序的设备，默认值 =。cuda:0 |
| datadir | 数据集的路径，默认值 =。./data/ |
| logdir | 存储日志的路径，默认值 =。./logs/ |
| noise | 我们添加到局部方的高斯噪声的最大方差，默认值 =。0 |
| sample | 参与每轮沟通的各方比率，默认值 =。1 |
| init_seed | 初始种子，默认值 =。0 |

图 4.1 使用说明

可以调用 function in 来访问 。 是一本字典。其键是参与方 ID，每个键的值是一个列表，其中包含分配给该参与方的数据索引。对于实验，重复某些设置的实验时，变为 1 或 2。除非另有说明，否则默认值为 0。列出了获取数据分区的方法如下。
get_partition_dict()experiments.pynet_dataidx_mapnet_dataidx_mapinit_seed=0init_seednoise

基于数量的标签不平衡：=，或 partitionnoniid-#label1noniid-#label2noniid-#label3

基于分布的标签不平衡：=、= 或 partitionnoniid-labeldirbeta0.50.1

基于噪声的特征不平衡：=，=（实际上噪声不影响 partitionhomonoise0.1net_dataidx_map)

合成特征不平衡和真实世界特征不平衡：partition=real

数量偏差：=、= 或 partitioniid-diff-quantitybeta0.50.1

IID 设置：partition=homo

混合偏斜：= 基于分布的标签不平衡和数量偏斜的混合; = 和 = 表示基于分布的标签不平衡和基于噪声的特征不平衡的混合。

以下是函数参数的解释。get_partition_dict()

| 参数 | 描述 |
| --- | --- |
| dataset | 要使用的数据集。选项：。 mnist cifar10 fmnist svhn generated femnist a9a rcv1 covtype |
| partition | Tha 分区方式。选项：、、（或 2、3、…，表示每一方拥有的固定数量的标签）、、 homo noniid-labeldir noniid-#label1 real iid-diff-quantity |
| n_parties | 当事方数量。 |
| init_seed | 初始种子。 |
| datadir | 数据集的路径。 |
| logdir | 存储日志的路径。 |
| beta | 异构分配的狄利克雷分布的浓度参数。 |

图 4.2 参数解释

### 4.4 创新点

每一方不需要交换数据和进行集中培训，而是将其模型发送到服务器，服务器在每一轮中更新全局模型并将其发回给各方。机器学习的有效性很大程度上依赖于大量高质量的训练数据，但是在 FL 的设定下，无法保证高质量数据的集中训练从而产生了数据 Non-IID 分布问题，这也是 FL 的重点研究方向。之前的研究在不同的数据持有者之间规定了严格的数据划分策略从而产生满足数据 Non-IID 分布的训练数据，但是其在数据划分方面的硬性规则导致其不具有普适性。因此本文提出了一个全面的数据划分策略，旨在指定数据 Non-IID 分布的测试标准，并且从更加全面的角度评测现有方法的有效性和特点。

## 5 实验结果分析

对比了经典 FL 算法包括 FedAvg、FedProx、FedNova、SCAFFOLD 面对数据 Non-IID 分布时的效果，基于个性化联邦和鲁棒性框架设计的方法不在本文讨论范围内。

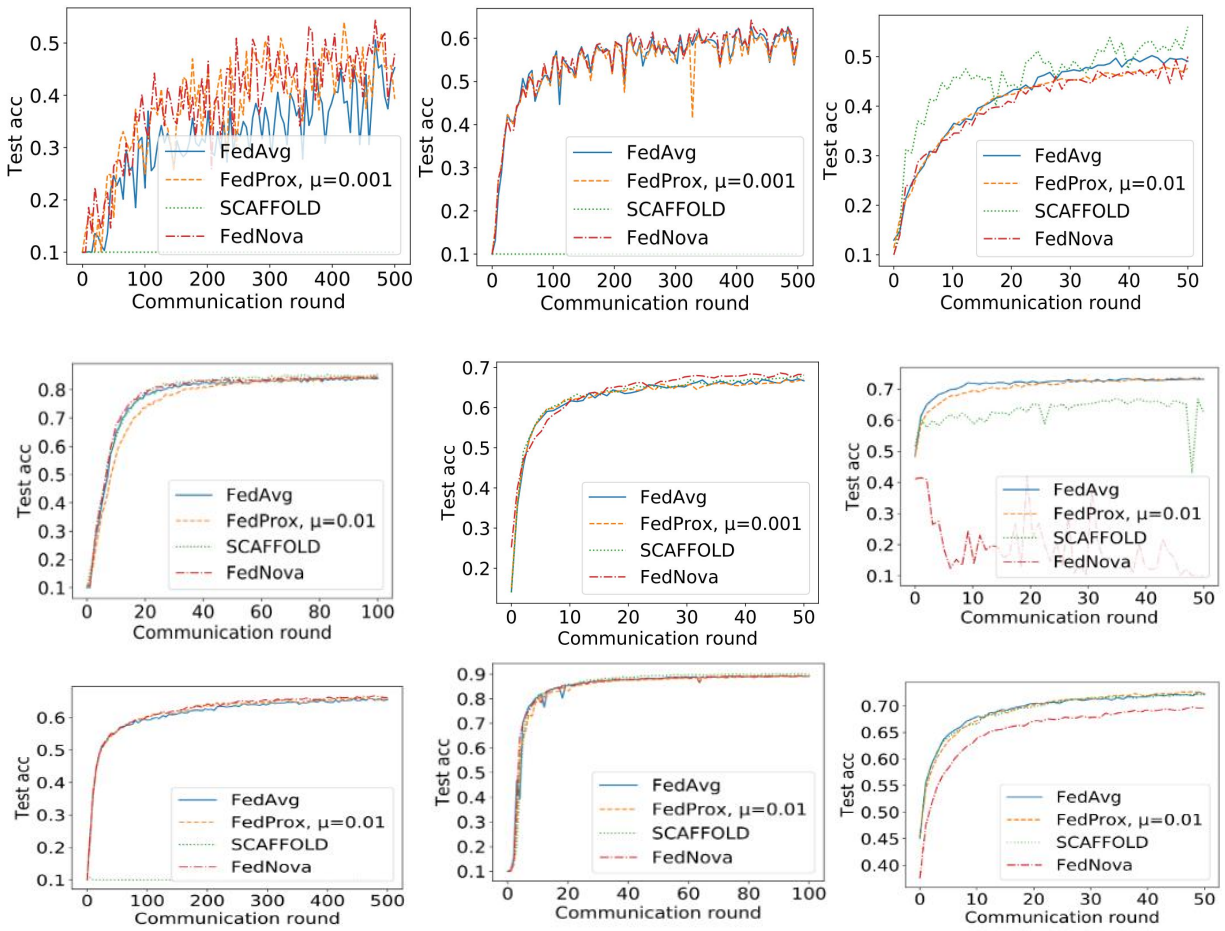



图 5.1 效果对比图

## 6 总结与展望

对利用分布式数据库（例如，在不同的组织和国家），以提高机器学习服务的有效性。在本文中，研究了非 IID 数据在这样的分布式数据库中的一个关键挑战，并开发了一个基准程序 NIIDbench。具体来说，介绍了六个数据划分策略，这是比以前的研究更全面。此外，进行了全面的实验，比较现有的算法，并证明他们的优点和缺点。这项研究揭示了在分布式数据库上构建有效的机器学习服务的未来方向。提出了一些有前途的未来方向的数据管理（与学习数据库系统集成、用于分析非 IID 数据的轻量级数据技术、部分参与的非 IID 抗性抽样、隐私保护数据挖掘、联邦数据库查询）和联邦学习非 IID 分布式数据库（一个标签、快速训练、FL 的自动参数调整、针对不同非 IID 设置的鲁棒算法、异构批处理规范化的聚合）。

## 参考文献

[1] S. AbdulRahman, H. Tout, A. Mourad, and C. Talhi. Fedmccs: multicriteria client selection model for optimal iot federated learning. IEEE Internet of Things Journal, 8(6):4723–4735, 2020.

[2] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama. Federated learning based on dynamic regularization. In International Conference on Learning Representations, 2021.

[3] R. Agrawal and R. Srikant. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pages 439–450, 2000.

[4] M. Andreux, J. O. du Terrail, C. Beguier, and E. W. Tramel. Siloed federated learning for multi-centric histopathology datasets. In Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning, pages 129–139. Springer, 2020.

[5] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. M. Kiddon, J. Koneˇcn´y, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander. Towards federated learning at scale: System design. In SysML, 2019.

[6] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Koneˇcn`y, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. arXiv preprint arXiv:1812.01097, 2018.

[7] S. Chaudhuri, R. Motwani, and V. Narasayya. Random sampling for histogram construction: How much is enough? ACM SIGMOD Record, 27(2):436–447, 1998.

[8] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. Emnist: Extending mnist to handwritten letters. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 2921–2926. IEEE, 2017.

[9] Z. Dai, B. K. H. Low, and P. Jaillet. Federated bayesian optimization via thompson sampling. Advances in Neural Information Processing Systems, 33, 2020.

[10] Y. Deng, M. M. Kamani, and M. Mahdavi. Distributionally robust federated averaging. Advances in Neural Information Processing Systems,33, 2020.