Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach

贾京

摘要

在联邦学习中，我们的目标是跨多个计算单元（用户）训练模型，而用户只能与一个公共的中央服务器进行通信，而不需要交换他们的数据样本。这种机制利用了所有用户的计算能力，并允许用户获得更丰富的模型，因为他们的模型是在更大的数据点集上进行训练的。但是，该方案只为所有用户开发了一个公共输出，因此，它并不适合每个用户。这是一个重要的缺失特性，特别是考虑到不同用户的底层数据分布的异构性。在本文中，我们研究个性化的联邦学习，我们的目标是找到一个初始共享模型，当前或新用户可以很容易地适应他们的本地数据集通过执行一个或几个步骤的梯度下降对自己的数据。这种方法保留了联邦学习体系结构的所有好处，并且通过结构，为每个用户提供了更个性化的模型。我们展示了这个问题可以在模型不可知的元学习（MAML）框架内进行研究。受到这种联系的启发，我们研究了一个个性化变体。

关键词：联邦学习；元学习

1 引言

联邦学习是在多个不相交的局部数据集上学习模型的任务。当本地数据由于隐私、存储或通信限制而无法共享时，它特别有用。例如，在物联网应用程序中，在边缘设备上创建大量数据的情况下，或使用由于隐私而无法共享的医疗数据。在联邦学习中，所有客户机都集体训练一个共享模型，而不共享数据，并试图最小化通信。不幸的是，当不同客户端的数据分布不同时，学习单个全局模型可能会失败。例如，用户数据可能来自不同的设备或地理位置，并且可能是异构的。在极端情况下，每个客户端可能都需要解决不同的任务。为了处理客户之间的这种异质性，在所选择的论文中，它通过考虑一个包含个性化的联邦学习问题的修正公式来克服这个问题。建立在模型不可知论元学习（MAML）问题公式引入，这个新公式的目标是找到一个初始点共享所有用户之间执行后每个用户更新它对自己的损失函数，可能通过执行几个步骤的基于梯度的方法。这样，虽然初始模型是在所有用户上以分布式方式导出的，但每个用户实现的最终模型不同于基于她或他自己的数据的其他模型。这篇论文的算法是 FedAvg 算法的一个个性化变体，称为 Per-FedAvg，设计用于解决所提出的个性化 FL 问题。

2 相关工作

2.1 联邦学习的挑战

联邦学习面临的挑战有：隐私保护（必须保证联邦学习中的模型训练不会泄露用户的隐私信息）、数据量不足（在传统机器学习中，如果想要得到一个较好的模型，往往需要大量的数据，但在分布式环境中，每个移动设备上的数据量不足。另一方面，以集中的方式收集所有的数据可能导致巨大的费用。因此，联邦学习要求每个设备使用本地数据来训练本地模型，然后将所有本地模型上传到服务器

上聚合成全局模型）和异质性（联邦环境中存在大量边缘设备，这些设备所持有的数据可能是非独立同分布的）[1]。在开发解决 FL 中不同挑战的一些方法中都有些取得了重大进展。特别是，在包括保护用户的隐私和降低通信成本。而与本文的论文更相关的是，也有几项工作研究了 FL 中用户数据点的统计异质性，但他们并没有试图为每个用户找到个性化的解决方案，对此本文提出了解决方法。

2.2 模型不可知论元学习（MAML）

首先简要地回顾一下 MAML 的公式。在 MAML 中，给定一组来自潜在分布的任务，与传统的监督学习设置相比，目标不是找到一个在所有期望任务上表现良好的模型。相反，在 MAML，我们假设我们有一个有限的计算预算更新我们的模型一个新任务到达后，在这个新设置，我们寻找一个初始化执行后更新对这个新任务，可能通过一个或几个步骤的梯度下降。特别地，如果我们假设每个用户接受初始点，并使用一步关于它自己的损失函数来更新它。在模型不可知论元学习（MAML）问题的集中式版本有多篇论文研究了它的经验特性及其收敛特性在这项工作中，本论文也关注 MAML 方法对 FL 设置的收敛，因为节点在向服务器发送更新之前执行多个本地更新，这在以前的元学习理论工作中没有考虑到。当然也有一些其他的论文从经验角度研究了 MAML 型方法与 FL 体系结构的不同组合。本文除了对算法的参数对其性能的作用外，在数值实验部分，展示了研究的方法在某些情况下可能不能很好地表现出来，并提出了另一个算法来解决这个问题。

3 本文方法

3.1 本文方法概述

在论文中，其是利用不可知论元学习（MAML）框架背后的基本思想来设计 FL 问题的个性化变体。对于 MAML 的公式。我们可以将问题：

$$\min_{w\in R^d} f(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w). \tag{1}$$

改变为：

$$\min_{w\in R^d} F(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w - \alpha\nabla f_i(x)). \tag{2}$$

这个公式的优势是，它不仅允许我们保持 FL 的优势，而且它捕获用户之间的区别作为现有或新用户可以把这个新问题的解决方案作为初始点和稍微更新它对自己的数据。这意味着用户可以通过检查自己的初始化，并更新自己的数据，只执行一个或几个梯度下降步骤，以获得一个适合自己数据集的模型。

如前所述，对于所考虑的数据分布的异构模型，解决问题 (1) 并不是理想的选择，因为它返回一个单一的模型，即使经过几步的本地梯度，也可能不会快速调整到每个用户的本地数据。另一方面，通过求解 (2)，我们找到了一个初始模型（元模型），它的训练方式是，经过一步局部梯度后，可以为每个用户提供一个很好的模型。这个公式也可以扩展到用户运行几个梯度更新步骤的情况。

为了遵循一个类似于 FedAvg 的方案来解决问题 (2)，第一步是计算局部函数的梯度，在这种情况下，梯度由下表示：

$$\nabla F_i(w) = (I - \alpha\nabla^2 f_i(w))\nabla f_i(w - \alpha\nabla f_i(w)). \tag{3}$$

通常每一轮计算梯度计算代价很高。因此，我们取一批关于分布 pi 的数据 Di，得到一个无偏的

估计，表示为：

$$\nabla f_i(w, D^i) := \frac{1}{|D^i|} \sum_{(x,y)\in D^i} \nabla l_i(w; x, y). \tag{4}$$

类似地，(3) 中的黑森矩阵的二阶导数可以被它的无偏估计所取代。

在 Per-FedAvg 的 k 轮，类似于 FedAvg，首先服务器将当前全局模型发送给大小均匀随机选择的部分用户。每个用户 i 在局部和对梯度执行随机梯度下降的 τ 步骤。特别是，这些本地更新生成一个本地序列

$$w^i_{k+1,t} = w^i_{k+1,t-1} - \beta \nabla F_i(w^i_{k+1,t-1}). \tag{5}$$

其中，β 为局部学习率（步长），梯度为 (3) 中梯度的估计值。注意，所有局部迭代的随机梯度使用独立的批次分别计算，它的计算如下：

$$\nabla F_i(w^i_{k+1,t-1}) := (I - \alpha \nabla^2 f_i(w^i_{k+1,t-1}, D^{''i}_t) \nabla f_i(w^i_{k+1,t-1} - \alpha \nabla f_i(w^i_{k+1,t-1}, D^i_t), D^{'t}_t) \tag{6}$$

3.2 特征提取、超参数与数据划分

论文中使用一个具有两个大小为 80 和 60 的隐藏层的神经网络，并使用指数线性单位（ELU）激活函数。在网络中获取 n 个 = 50 个用户，并为 K = 1000 轮来运行算法。在每一轮中，选择使用 r = 0.2 的 r • n 个代理来运行 τ 次本地更新。批量大小为 D=40，学习速率为 β=0.001。特别地，考虑了 MNIST 数据集上的多类分类问题，并将训练数据在 n 个用户之间分配训练数据如下：(i) 一半的用户，每个用户都有前五类的 a 个图像；（ii）其余的，每个用户都只有前 5 个类中的一个类 a/2 个图像和其他 5 个类中的一个类的 2a 个图像。在 MNIST 数据集中，我将参数 a 分别设置为 = 196。这样，我们就创建了一个例子，其中所有用户上的图像分布都是不同的。类似地，我们将测试数据划分在与训练数据分布相同的节点上。请注意，对于这个用户的分布有显著不同的特定例子，我们的目标不是要实现最先进的准确性。相反，我们的目的是提供一个例子来比较在异构设置中获得个性化的各种方法。事实上，通过使用更复杂的神经网络，所有考虑的算法的结果[2]。

3.3 两种近似梯度的方法

由于 Per-FedAvg 的实现需要访问二阶信息，这对计算成本很高。为了解决这个问题，降低计算，论文中考虑了两种不同的近似方法来近似 (6):

第一种 (FO) 直接用一阶导来代替整个梯度 (3)

第二种 (HF) 用二阶偏导的定义来近似，即用如下公式来近似.

$$\nabla f_i(w - \alpha \nabla f_i(w, D), D') - \alpha d_i(w) \tag{7}$$

其中：

$$d_i(w) := \frac{\nabla f_i(w + \delta \nabla f_i(w - \alpha \nabla f_i(w, D), D'), D'') - \nabla f_i(w - \delta \nabla f_i(w - \alpha \nabla f_i(w, D), D'), D'')}{2\delta} \tag{8}$$

4 复现细节

本篇论文没有源码，在复现的过程中参考了变体之前的算法 FedAvg 算法的代码，并进行了改进。FedAVg 算法和 Per-FedAvg 算法如下：

Algorithm 1: The proposed Personalized FedAvg (Per-FedAvg) Algorithm

Input: Initial iterate w_0, fraction of active users r.
for $k : 0$ to $K-1$ **do**
  Server chooses a subset of users $\mathcal{A}_k$ uniformly at random and with size rn;
  Server sends w_k to all users in $\mathcal{A}_k$;
  for all $i \in \mathcal{A}_k$ **do**
    Set $w^i_{k+1,0} = w_k$;
    for $t : 1$ to τ **do**
      Compute the stochastic gradient $\tilde{\nabla} f_i(w^i_{k+1,t-1}, \mathcal{D}^i_t)$ using dataset $\mathcal{D}^i_t$;
      Set $\tilde{w}^i_{k+1,t} = w^i_{k+1,t-1} - \alpha \tilde{\nabla} f_i(w^i_{k+1,t-1}, \mathcal{D}^i_t)$;
      Set $w^i_{k+1,t} = w^i_{k+1,t-1} - \beta(I - \alpha \tilde{\nabla}^2 f_i(w^i_{k+1,t-1}, \mathcal{D}^{''i}_t))\tilde{\nabla} f_i(\tilde{w}^i_{k+1,t}, \mathcal{D}^{'i}_t)$;
    end for
    Agent i sends $w^i_{k+1,\tau}$ back to server;
  end for
  Server updates its model by averaging over received models: $w_{k+1} = \frac{1}{rn} \sum_{i \in \mathcal{A}_k} w^i_{k+1,\tau}$;
end for

对于上述算法，我首先对其更新的地方做了改进，对于 FedAvg 算法 users 端每次迭代只需要一步更新，而对于 Per-FedAvg 算法需要两步更新，并且我使用了文中提及的两种方法来更新第二步。其次，由于 Per-FedAvg 算法的两步更新，需要对优化器的某些功能进行改进，在复现中我通过对优化器类的继承，编写了一个用于 Per-FedAvg 算法更新的优化器。在论文中，其数据划分也和之前不同，需要将数据集划分不等分的大小，并且要保证它们是非独立同分布的。

5 实验结果分析

实验结果如下表：

Dataset	Parameters	PerFedAvg(FO)	PerFedAvg(HF)
MNIST	τ=10 α=0.001	89.98%	89.35%
	τ=10 α=0.01	70.22%	60.12%
	τ=4 α=0.001	87.14%	86.38%
	τ=4 α=0.01	73.81%	72.17%

从实验结果来看对于 α = 0.001 和 τ = 10，PerFedAvg（FO）和 Per-FedAvg（HF）的表现几乎相似，并且效果最好。此外，降低 τ 会导致算法性能的下降，这是随着总迭代次数的减少。接下来，研究对于 α 的作用，通过将 α 从 0.001 增加到 0.01，算法的性能下降，这是由于随着迭代，较高 α 不能够很好的保证算法的收敛。对于降低 τ，由于总的迭代次数下降会导致也会算法的性能下降。但是对于 α = 0.01 和 τ = 10 的时候，提高 τ 两个算法的性能都下降了，这是可能是由于总迭代次数多和学习率高的共同作用下导致了算法产生了过拟合的现象，也可能是对于 τ = 10 的时候，模型在测试时能更好的适应用户的数据。之后，经过多次测试并输出迭代的时候产生的图像，发现在经过大约 300 次左右的迭代后，其产生的损失值有明显的波动，由此可以得出其发生了过拟合的现象。由上述可见，显然对于较大的 α，无论是 PerFedAvg（FO）还是 Per-FedAvg（HF）的性能都有显著的下降。总之，更准确的 Per-FedAvg 的实现，即 Per-FedAvg（FO），在所有情况下都优于 FedAvg，并导致了更个性化的解决方案。

6 总结与展望

在联邦学习（FL）问题在异构情况下，本篇论文研究了一个个性化的经典 FL 公式，它的目标是找到一个适当的初始化模型，可以迅速适应每个用户的本地数据后训练阶段。并且强调了其公式与模型不可知的元学习（MAML）的联系，并展示了如何使用 MAML 的分散实现，称之为 Per-FedAvg，来解决所提出的个性化 FL 问题。可惜都是在实现过程中，并没有很好的把第二种方法的优势展现出来。总之，实验表明由 Per-FedAvg 得到的解是具有个性化的解。而对于联邦学习的展望来说，其更大的影响，是其为高效和分布式地训练机器学习模型提供了一个框架。由于这些有利的特性，它已经获得了广泛的关注，并已被广泛地应用于具有关键的社会效益的应用中。这些应用程序从医疗保健系统，其中机器学习模型可以被训练，同时保护患者的隐私，到图像分类和 NLP 模型，科技公司可以改进他们的神经网络，而不需要用户与服务器或其他用户共享他们的数据[3]。在这篇论文的工作中，其研究了 FL 的挑战之一，即个性化方面。他试图从理论角度回答的主要问题是，即是否可以有一个面向用户的经典 FL 算法变体，能够在享受 FL 分布式架构的同时适应每个用户数据。在论文中证明了其答案是积极的，并为算法提供了严格的理论保证，这些算法可以用于上述所有应用，以实现在 FL 框架中更个性化的模型。事实上，这一结果可能会对提高用户模型的质量产生广泛的影响

参考文献

[1] IMTEAJ A, THAKKER U, WANG S, et al. A Survey on Federated Learning for Resource-Constrained IoT Devices[J]. IEEE Internet of Things Journal, 2022, 9(1): 1-24. DOI: 10.1109/JIOT.2021.3095077.

[2] ALIREZA FALLAH A O, Aryan Mokhtari. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach[J]. NeurIPS, 2020.

[3] IEEE Approved Draft Guide for Architectural Framework and Application of Federated Machine Learning[J]. IEEE P3652.1/D6.1, July 2020, 2020: 1-70.