Analyze over Multi-objective Optimization in Federated Learning

Amirahmad Shafiee[1]

***Abstract*— Federated learning (FL) tackles the challenge of training models across distributed data sources. Real-world deployment unveils challenges in privacy, robustness, fairness, and convergence time. This paper analyzes a seminal work, FEDERATED LEARNING MEETS MULTI-OBJECTIVE OPTIMIZATION, focusing on their approaches to addressing fairness and robustness constraints. We delve into the methodologies proposed by this paper, examining their programming setup in tackling the complexities of federated learning scenarios. Furthermore, we inspect the constraints they consider. Through this analysis, we aim to examine optimization techniques used in state-of-the-art research to address FL challenges. Hence, one of our main concentrations will be to emphasize introducing the techniques in the literature of optimization. Moreover, since Privacy is one of the most important aspects of Federated Learning, we introduce a constraint into the Programming setup to satisfy privacy to some extent.**

I. INTRODUCTION

Federated Learning (FL) is a novel approach in Machine Learning (ML) that deals with distributed optimization in the context of increasingly popular small-user devices and applications that can benefit from ML. Instead of moving the training data to a central location, FL utilizes the computational power available on user devices and distributes the training process across participating nodes.

FL is closely related to conventional distributed optimization, but it was developed to tackle new challenges in the mobile era that traditional distributed optimization algorithms were not designed for. These challenges include non-independent and identically distributed (Non-IID) data within edge devices, limited user communication, privacy concerns, fairness issues, and robustness.

Ensuring fairness among users has become one of the serious goals in FL, as it largely determines users' willingness to participate and ensures some degree of robustness against malicious user manipulations. FL algorithms are eventually deployed in the wild, hence subject to malicious attacks. Indeed, adversarial attacks have been constructed recently to reveal vulnerabilities of FL systems against malicious manipulations on the user side.

FEDERATED LEARNING MEETS MULTI-OBJECTIVE OPTIMIZATION has addressed the mentioned challenges. In this work, we will focus on the optimization problems mentioned within this work. We first give a background in section II on the introduced methods and concepts that we will face in the entire paper. In section III, we introduce challenges regarding fairness and robustness, define constraints to be satisfied, and write down the final form of the optimization problem. We finally wrap up this section by referring to FedMGDA+, an efficient and proven convergent algorithm. In section IV we refer to MIST, a work on privacy in defense of Membership Inference Attacks (MIAs), and with its inspiration, introduce a constraint to our setup, to further guarantee some degree of privacy.

II. BACKGROUND

A. Multi-objective Optimization

Multi-objective minimization (MoM) encompasses the simultaneous minimization of multiple scalar objective functions, often incompatible with each other, necessitating the optimization of a vector-valued function. Mathematically, MoM can be expressed as:

$$\min_{w \in \mathbb{R}^d} f(w) := (f_1(w), f_2(w), ..., f_m(w))$$

[1]Amirahmad Shafiee, 99104027, Department of Mathematical Sciences

Where the minimum is defined with respect to the partial ordering:

$$f(w) \leq f(z) \Leftrightarrow \forall i = 1, ..., m,\ f_i(w) \leq f_i(z)$$

Unlike single-objective optimization, in MoM, it is possible to encounter scenarios where $f(w) \not\leq f(z)$ and $f(z) \not\leq f(w)$, indicating that w and z are not comparable. A solution w^* is considered **Pareto optimal** if its objective value $f(w^*)$ is a minimum element with respect to the partial ordering, implying that it cannot be further improved in any objective without worsening at least one other objective. **Weakly Pareto optimal** solutions represent instances where no other solution can improve all component objectives simultaneously. It's important to note that optimal solutions in MoM typically form a set, with all Pareto optimal solutions considered equally good without additional subjective preference information.

In the context of federated learning, where optimization is distributed across multiple nodes, finding Pareto optimal solutions presents significant challenges. Instead, Pareto stationary solutions, which satisfy specific first-order necessary conditions, are often pursued. These solutions are characterized by convex combinations of gradients vanishing, indicating optimality in a multi-objective sense. Several algorithms exist for finding Pareto stationary solutions, including the weighted approach, o-constraint approach, and Chebyshev approach, each offering unique insights into multi-objective optimization.

B. Standard Approach to Solving FL

In the setup of federated learning (FL), the characteristics of data distribution from which our training examples (x_i, y_i) will be drawn are as follows:

1) **Massively Distributed:** Data points are stored across a large number K of nodes. In particular, the number of nodes can be much bigger than the average number of training examples stored on a given node (n_K).
2) **Non-IID:** Data on each node may be drawn from a different distribution, i.e., the data points available locally are far from being a representative sample of the overall distribution.
3) **Unbalanced:** Different nodes may vary by orders of magnitude in the number of training examples they hold.

Adapting the objective function to these characteristics, the problem can be defined as introduced in the following paragraphs.

We have K nodes and n data points, a set of indices P_k $(k = 1, \ldots, K)$ of data stored at node k, and $n_k = |P_k|$ is the number of data points at P_k. We assume that $P_k \cap P_l = \varnothing$ whenever $l \neq k$, thus $\sum_{k=1}^{K} n_k = n$.

We can then define the local loss for node k as $F_k(w) \frac{1}{n_k} \sum_{i \in P_k} f_i(w)$. Thus, the problem to be minimized becomes:

$$\min_{w \in \mathbb{R}^d} f(w) = \frac{1}{n} \sum_{k=1}^{K} n_k F_k(w)$$

To solve the problem, the simplest algorithm is FederatedSGD, which is equivalent to mini-batch gradient descent over all of the data and is a simple application of distributed synchronous Stochastic Gradient Descent (SGD) for the described setup.

III. Multi-objective Optimization in FL

In this section, we will discuss the challenges of fairness and robustness. In the first part, we will bring up different limitations and introduce different constraints to satisfy. We will then bring up the main problem and discuss its superiority over standard and basic solutions of FL. Lastly, FedMGDA+ is being introduced and analyzed briefly.

Early approaches to optimizing the performance in an FL system were FedAvG and AFL. Both of the methods lacked in some way. We consider FL with m users (edge devices), where the i-th user is interested in minimizing a function $f_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, ..., m$, defined on a shared model parameter $w \in \mathbb{R}^d$. FedAVG consisted of the following setup:

$$\min_{w} \sum_{i=1}^{m} \lambda_i f_i(w)$$

Where the weights λ_i need to be specified beforehand. As can be easily seen, this setup of the objective functions is abundant in several problems. It only refers to finding the best weights, considering a preset setup of hyperparameters.

Hence, the challenges of fairness hold. Moreover, it is open to the malicious activity of users. Simply adding a constant to some $f_i(w)$ may result in significant changes in w. Hence, the challenge of robustness holds. To address these challenges, AFL was introduced. It follows the following setup:

$$\min_{w} \max_{\lambda \in \Lambda} \mathcal{A}^0_{f,\lambda}(w)$$

where

$$\mathcal{A}^0_{f,\lambda}(w) := \sum_{i=1}^{m} \lambda_i f_i(w).$$

Basically, it tries to minimize the loss in the worst-case scenario. In the introduced system, the set might cover reality better than any specific and provide some minimum guarantee for all users. However, this system is even more non-robust. Given that we first maximize over λ and then minimize over w, a user's malicious activity affects the result even more.

Now, consider the general objective function of multi-objective optimization. It is evident that each f_i can represent an objective function. Hence, putting the objective function as:

$$\min_{w \in \mathbb{R}^d} f(w) := (f_1(w), f_2(w), ..., f_m(w))$$

where $f_i(.)$ is the local loss function of user i, which defines a new optimization problem whose purpose is to reach a balance among its components. The provided system, if reaching a Pareto-stationary solution, guarantees fairness. The prior statement is derived from the definition of Pareto-stationary points. In such positions, directions through which an increase in one utility causes the other to decrease are considered worsening directions, hence reducing the overall utility. Therefore, such directions do not lead to other Pareto-stationary points, thus guaranteeing fairness.

To solve this problem, in the main paper, they have proposed an iterative method based on a Chebysev approach followed by applying a quadratic bound, assuming smoothness for f_i. Via applying the mentioned methods, the problem turns into:

$$w_{t+1} = \underset{w}{argmin} \max_{\lambda \in \Delta} \lambda^\top (f(w) - f(\tilde{w}_t))$$

and then:

$$w_{t+1} = \underset{w}{\text{argmin}} \max_{\lambda \in \Delta} \lambda^\top J_f^\top(w_t)(w - w_t) + \frac{1}{2\eta}\|w - w_t\|^2$$

where $J_f = [\nabla f_1, \ldots, \nabla f_m] \in \mathbb{R}^{d \times m}$ is the Jacobian and $\eta > 0$ is the step size. Since the equation is convex in w and concave in λ, we can swap min with max and obtain the dual.

$$w_{t+1} = \max_{\lambda \in \Delta} \min_{w} \lambda^\top J_f^\top(w_t)(w - w_t) + \frac{1}{2\eta}\|w - w_t\|^2$$

Now, applying the MGDA step by setting derivatives regarding w to 0, we get:

$$w_{t+1} = w_t - \eta d_t, \quad d_t = J_f(w_t)\lambda_t^*$$
$$where \quad \lambda_t^* = \underset{\lambda \in \Delta}{argmin} \|J_f(w_t)\lambda\|^2$$

To further satisfy different aspects and rules of FL, containing robustness, several other constraints are being added. Two of the major ones are mentioned in the following.

- **Balancing user average performance and fairness**: During the update phase, we aim to maintain both fairness and average performance by imposing a constraint that ensures an adequate performance level. Suppose that for some λ_0, the performance of the algorithm regarding the overall accuracy is being maximized.

  $$\lambda_t^* = \underset{\lambda \in \Delta, \{\lambda - \lambda_0\}_\infty \leq \epsilon}{argmin} \|J_f(w_t)\lambda\|^2$$

  It can be seen that tuning ϵ from $[0, 1]$ biases towards fairness.
- **Robustness against malicious users through normalization**: The proposed method is still open to malicious user-level activity. As proposed in the paper, normalizing gradients would provide the algorithm with robustness since it stops inflation by bounding the effect of an increase in one's loss function. Moreover, this method still holds fairness.

Overall, the main step of the algorithm is an optimization problem itself in the format of the following:

$$\begin{aligned} \underset{\lambda \in \Delta}{\text{minimize}} \quad & \|J_f(w_t)\lambda\|^2 \\ \text{subject to} \quad & \|\lambda - \lambda_0\|_\infty \leq \epsilon, \\ & \sum_{i=1}^{n} \lambda_i = 1 \end{aligned}$$

Which is equivalent to finding the best dual variables, subject to estimated minimizer w_t. Notice that when $min\,\|J_f(w_t)\lambda\|^2 \to 0$ it indicates that no direction further optimizes the vector of objectives, hence we have reached a Pareto-stationary solution.

The process of finding the common descent direction involves solving a standard quadratic programming problem. For a moderate number of users, a standard QP solver is sufficient. However, for a large number of users, we can efficiently solve using the conditional gradient algorithm.

Algorithm 1 FedMGDA+

for $t = 1, 2, \ldots$ **do**
  Choose a subset I_t of $\lceil p \cdot m \rceil$ clients/users
  for $i \in I_t$ **do**
    $g_i \leftarrow$ ClientUpdate(i, w_t)
    $\bar{g}_i \leftarrow g_i / \|g_i\|$ {Normalize}
  end for
  $\lambda^* \leftarrow \underset{\lambda \in \Delta, \|\lambda - \lambda_0\|_\infty \leq \epsilon}{\arg\min} \sum_i \|\lambda_i \bar{g}_i\|^2$
  $d_t \leftarrow \sum_i \lambda_i^* \bar{g}_i$ {Common direction}
  Choose (global) step size η_t
  $w_{t+1} \leftarrow w_t - \eta_t d_t$
end for
Function ClientUpdate(i,w):
$w_0 \leftarrow w$
for k epochs **do**
  {Split local data into r batches}
  $D_i \to D_{i,1} \cup \ldots \cup D_{i,r}$
  for $j = 1, 2, \ldots, r$ **do**
    $w \leftarrow w - \eta \nabla f_i(w; D_{i,j})$
  end for
end for
return $g := w_0 - w$ {To serve}

The convergence of the proposed algorithm is examined in a couple of statements.

Theorem 1a. Let each user function f_i be L-Lipschitz smooth and M-Lipschitz continuous, and choose step size η_t such that $\sum_t \eta_t = \infty$ and $\sum_t \sigma_t^2 \eta_t < \infty$, where $\sigma_t^2 := E\|d_t - \hat{d}_t\|^2$ with

$$d_t := J_f(w_t)\lambda_t, \lambda_t = \underset{\lambda \in \Delta}{argmin} \|J_f(w_t)\lambda\|$$

$$\hat{d}_t := \hat{J}_f(w_t)\hat{\lambda}_t, \hat{\lambda}_t = \underset{\lambda \in \Delta}{argmin} \|\hat{J}_f(w_t)\lambda\|$$

Then, with $k = r = 1$, we have:

$$\min_{t=0,\ldots,T} E\|J_f(w_t)\lambda_t\|^2 \to 0.$$

where k is the number of local updates, and r is the number of minibatches in each local update. The convergence rate depends on how quickly the "variance" term t of the stochastic common descent direction $\hat{dt}$ diminishes.

In case of deterministic gradient updates, we can prove convergence even with more local updates.

Theorem 1b. Let each user function f_i be L-Lipschitz smooth and M-Lipschitz continuous. For any number of local updates k, if the global step size $\eta_t \to 0$ with $\sum_t \eta_t = \infty$, local learning rate $\eta_{lt} \to 0$, and $\epsilon_t := \lambda_t - \hat{\lambda}_t \to 0$, then we have:

$$\min_{t=0,\ldots,T} \|J_f(w_t)\lambda_t\|^2 \to 0.$$

Now, when we have even stronger situations, such as convexity of f_i, the following theorem holds.

Theorem 2. Suppose each user function f_i is convex and M-Lipschitz continuous. Suppose at each round FedMGDA+ includes a strongly convex user function whose weight is bounded away from 0. Then, with the choice $\eta_t = \frac{2}{c(t+2)}$ and $k = r = 1$, we have:

$$E\|w_t - w_t^*\|^2 \leq \frac{4M^2}{c^2(t+3)},$$

and $w_t - w_t^* \to 0$ almost surely, where w_t^* is the nearest Pareto stationary solution to w_t, and c is some constant.

proofs of all of the theorems are thoroughly examined in the source paper.

IV. MIST with Relaxing Parameter to Address Privacy

Machine Learning (ML) models and, as an example, FL, are subject to Membership Inference Attacks (MIA). In MIAs, the adversary tries to determine whether an instance is used to train an ML model. MIAs constitute a significant privacy concern when using private data to train ML models.

One of the significant concerns and purposes of FL is privacy. Data within FL comes from highly private resources, and while the data is not directly used to train the main model, several distributions of locally sensitive data might bring up privacy concerns, especially when given sufficiently large weights.

In work **MIST: Defending Against Membership Inference Attacks Through Membership-Invariant Subspace Training**, a method to mitigate privacy concerns has been thoroughly examined. In this work, a theoretical support of the algorithm is introduced, which provides sufficient privacy guarantee.

$$\begin{aligned} &\text{xdiff}(\theta_D, \theta_{D\setminus\{(x_M,y_M)\}}) \\ &= \sup \left\| F(x;\theta_D)y - F(x;\theta_{D\setminus\{(x_M,y_M)\}})y \right\|_1 \end{aligned}$$

Where supremum is taken over all data pairs. Recall that the second term within the norm is the model trained on all data, excluding the m-th pair. Now, the model is said to be membership invariant if the following holds.

$$\sum_{M\in\{1,\dots,n\}} \text{xdiff}(\theta_D, \theta_{D\setminus\{(x_M,y_M)\}}) = 0$$

It is obvious from the formula why the statement holds true value. But for more data on proof, you can refer to the source paper.

The above equation tends to be strict and, more importantly, not possible due to the nature of FL. If we abuse the notation and consider the pair (x_M, y_M), a full set of data coming from user m, then a relaxed version of the Membership Invariance equation may be an effective constraint.

$$\max_{M\in\{1,\dots,n\}} \text{xdiff}(\theta_D, \theta_{D\setminus\{(x_M,y_M)\}}) \leq \epsilon$$

Where ϵ is a satisfying, beforehand tuned hyperparameter. One other way to address this issue is to add the left-hand side of the above non-equality to the objective function and via putting some limitations over its weights, seek privacy and fairness in a system reaching of Pareto-stationary solution.

V. Conclusion

FL is abundant with optimization challenges. As a distributed optimization paradigm itself, it comes from the nature of optimization. With the emergence of computationally capable edge devices, FL becomes more and more critical and faces more challenges, from accuracy to ethical concerns like privacy and fairness. Optimization techniques like the discussed Multi-objective optimization have shown potential in tackling such challenges. Moreover, we managed to introduce constraints satisfying privacy to some extent, further proving the capabilities of optimization techniques within FL.