A Bi-Objective Optimization Approach for Enhancing FedUL

Abdullah Alzahrani
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
alzaha@rpi.edu

May 2, 2024

Abstract

Federated learning (FL) enables clients to collectively train a machine learning model without sharing their data, thus preserving privacy. Federated learning is particularly effective in supervised settings. However, in unsupervised settings, where clients prefer not to disclose their data labels, the optimization of the objective function faces additional challenges. We propose a Bi-Objective optimization method that enhances Federated Unsupervised Learning (FedUL) algorithm to address these issues. This report validates the effectiveness of our method experimentally using publicly accessible data to ensure reproducibility of the results.

1 Introduction

In this project, we aim to minimize the cost function associated with multi-class classification tasks. Training local models in unsupervised federated learning settings using decentralized datasets can encounter numerous obstacles. These include the absence of a definitive objective function for model evaluation. Unlike unsupervised learning, where generative models like GANs or diffusion models excel at modeling data distributions, federated learning contends with heterogeneous datasets. This heterogeneity can prevent local models from acquiring generalizable knowledge and complicate the aggregation of these models on the server side.

1.1 Related Work

FedUL is an algorithm designed to overcome the challenge of unclear objectives in the unsupervised FL setting. It begins with the natural segregation of data across clients, splitting each client's dataset into multiple subsets. Each subset

is indexed relative to the client's dataset, serving as the surrogate label. This approach facilitates training on surrogate data using a supervised task. Thus, the objective becomes minimizing the loss function associated with predicting the surrogate labels. The algorithm assumes we have access to the class priors and a shared class-conditional distribution among clients. The primary goal of FedUL is to use an injective transition function Q to recover the optimal model from the surrogate task, ensuring that the knowledge acquired is applicable for classifying the actual data.

This goal is achieved by implementing Theorem 1 from the paper, which establishes a method to correlate the posterior probabilities of the real and surrogate data. Specifically, it assesses the likelihood of each surragate dataset chosen per client. Which is approximated as the number of features for that dataset over the total number of features for the client. Then it maps posterior probability from the real space to the space of the surrogate data. Then it utilizes the prior probability of the real classes to assess the likelihood of a class's presence in a client's dataset, effectively normalizing the posterior probability predicted by the global model, which is the last matrix in the unnormalized representation of the injective function. By optimizing the surrogate classifier, the classifier for the actual task is concurrently enhanced, supported by other assumptions detailed in the paper.

Theorem 1

For each client $c \in [C]$, let $\bar{\eta}_c : \mathcal{X} \to \Delta^{M-1}$ and $\eta : \mathcal{X} \to \Delta^{K-1}$ be the surrogate and original class-posterior probability functions, respectively, where $(\bar{\eta}_c(x))_m = \bar{p}_c(\bar{y} = m \mid x)$ for $m \in [M]$, and $(\eta(x))_k = p(y = k \mid x)$ for $k \in [K]$. Here, Δ^{M-1} and Δ^{K-1} are the M-dimensional and K-dimensional simplexes, respectively.

Let $\bar{\pi}_c = [\bar{\pi}_{1c}, \ldots, \bar{\pi}_{Mc}]^\top$ and $\pi = [\pi_1, \ldots, \pi_K]^\top$ be vector forms of the surrogate and original class priors, respectively. Let $\Pi_c \in \mathbb{R}^{M \times K}$ be the matrix form of $\pi^c_{m,k} = p^c_m(y = k)$. Then we have:

$$\bar{\eta}_c(x) = Q_c(\eta(x); \pi, \bar{\pi}_c, \Pi_c)$$

where the unnormalized version of the vector-valued function Q_c is given by:

$$\tilde{Q}_c(\eta(x); \pi, \bar{\pi}_c, \Pi_c) = D_{\bar{\pi}_c} \cdot \Pi_c \cdot D_\pi^{-1} \cdot \eta(x).$$

Here, D_a denotes the diagonal matrix with the diagonal terms being vector a, and '$\cdot$' denotes matrix multiplication. Q_c is normalized by the sum of all entries, i.e., $(Q_c)_i = \frac{(\tilde{Q}_{ec})_i}{\sum_j (Q_c)_j}$.

2 Our Contribution

2.1 Overview

We introduce an enhanced algorithm designed to enhance the performance of FedUL in terms of accuracy and robustness. Recognizing FedUL's effectiveness, our approach integrates FedUL with an additional optimization phase utilizing supervised learning techniques. This integration involves training local models using FedUL, aggregating these models at the server, and subsequently implementing a bi-objective optimization task trained on publicly accessible labeled data, aiming for Pareto optimality. This approach has shown to improve FedUL's performance, especially in non-IID settings.

2.2 Significance

Despite the scarcity of publicly accessible labeled datasets in many areas, our method leverages any applicable available small-sized labeled dataset in conjunction with a larger decentralized dataset. This approach preserves the privacy of client data while enhancing the model's performance through bi-objective optimization techniques.

2.3 Integration Techniques

To integrate the updates from local models with the supervised model's updates, we explored two methods:

2.3.1 Method 1: Feedback Optimization

In this approach, we fine-tune the aggregated global model using the labeled data retained at the server. This phase optimizes the objective of the server's model based on the labeled data. Following this, we establish a feedback loop by redistributing the fine-tuned model back to the clients. This loop allows the clients to further optimize the model with their local data, enhancing the overall accuracy of the model.

2.3.2 Method 2: Independent Enhancement

Alternatively, we execute the FedUL process to aggregate the local models into a global model. Parallel to this, we optimize an independent model, intialized uniformly with the global model, specifically on the labeled data. The parameters learned from this supervised task are then combined with the global model's parameters. This combination is regulated through a manually tuned hyperparameter, effectively blending the insights of both the local models and the supervised model. A similar loop to method 1 is implemented.

3 Proposed Algorithm

Algorithm 1 Federation of unsupervised learning (FedUL)

Server Input: initial f, global step-size α_g, and global communication round R
Client c's Input: local model f_c, unlabeled training sets $U_c = \{U_c^m\}_{m=1}^{M_c}$, class priors Π_c and π, local step-size α_l, and local updating iterations L

Start with initializing clients with Procedure A.
for $r = 1 \to R$ **do**
    Run Procedure B and Procedure C iteratively.
end for
procedure A. ClientInit(c)
    Transform U_c to a surrogate labeled dataset U_c according to (6)
    Modify f to $g_c = Q_c(f)$, where Q_c is computed according to Theorem 1
end procedure
procedure B. ClientUpdate(c)
    $f_c \leftarrow f$ ▷ Receive updated model from ServerExecute
    $g_c \leftarrow Q_c(f_c)$
    for $l = 1 \to L$ **do**
        $g_c \leftarrow g_c - \alpha_l \nabla J_b(g_c; U_c)$ ▷ SGD update based on objective (7)
        $f_c \leftarrow f_c - \alpha_l \nabla J_b(Q_c(f_c); U_c)$ ▷ Update on g_c induces update on f_c
    end for
    Send $f_c - f$ to ServerExecute
end procedure
procedure C. ServerExecute(r)
    if using Method 1 **then**
        Fine-tune f on the server's labeled data
    else if using Method 2 **then**
        Independently train a model f' on labeled data
        $f \leftarrow (1 - \text{hyperparameter}) \times f + \text{hyperparameter} \times f'$
        Broadcast updated f to ClientUpdate
    end if
end procedure

4 Empirical Results

4.1 Experimental setup

The experiments were done on the MNIST dataset using a CNN with the exact dimensions and parameters as conducted in the experiments of FedUL, with exception of a batch size of 100 instead of 128 for splitting considerations. The goal is achieve a performance better than that of FedUL's and validate the approach of bi-objective optimization. Below is a table comparing our model's mean error compared to FedUL's, with FedUL+Mx being a place holder for our algorithm with method $\#x$. The best performing model on the validation data is chosen.

Table 1: Setup

Clients	Parameters	Data Splitting
#C = 5	$\alpha = 1e-4$	Test: 10k
Set/C = 10	batch = 100	FedUL: 36k
	epochs = 100	Supervised task: 12k
	method 2's hyperparameter: 0.5	Validation: 12k

4.2 Results

Table 2: Mean Error Rates

Algorithm	Mean Error (IID)	Mean Error (non-IID)
FedUL	0.78	2.98
FedUL+M1	0.69	0.77
FedUL+M2	0.69	0.79

Note

Additionally, plots showing the mean error and losses per epoch are included and briefly discussed in the appendix.

5 conclusion and Future Work

In conclusion, we found that implementing a bi-objective optimization with FedUL and a supervised model can significantly enhance the performance of a standalone FedUL model, particularly when applied to non-IID datasets. Our research has demonstrated two key findings. First, FedUL is capable of learning concurrently with a supervised federated learning technique without the objectives interfering destructively with each other. Second, the integration of knowledge from a supervised model into FedUL not only boosts its accuracy but also facilitates faster convergence. In the current setting, method 1 seems to be perform better than method 2.

Looking ahead, future work could focus on several areas. Extensive validation of the algorithm under diverse settings and with various models could further establish its robustness and adaptability. Additionally, optimizing the hyperparameter tuning process to enhance performance and ease of use. Finally, examining the impacts of different data distributions and more extreme cases of non-IID data could provide deeper understanding for broader applications in real-world scenarios.

References

[1] Lu, N., Wang, Z., Li, X., Niu, G., Dou, Q., & Sugiyama, M. (2022) Federated Learning from Only Unlabeled Data with Class-Conditional-Sharing Clients. In *International Conference on Learning Representations*. Available: `https://openreview.net/forum?id=WHA8009laxu`

Code: The code used to provide the empirical results is a heavily modified version of the code the authors Lu, N et. al. used and can be found here, under federated. choose fedulmnist.py:

`https://rpi.box.com/s/y86pvaxxs5sn2sj6huwhrvs2xahszmi6`

Instructions: This is the link for the GitHub repository of FedUL algorithm, which mostly covers how to run the code. For extra information please contact me:

`https://github.com/lunanbit/FedUL?tab=readme-ov-file`

Appendix

Discussion

5.1 Loss

Shown in Figure 1 is the supervised learning loss and surrogate task loss when performing the hybrid model, compared to the loss of performing standalone FedUL. The loss of the surrogate task and FedUL seems to be exact, with the exception that the surrogate task's drops faster initially. This shows the positive influence of the hybrid model on adjusting the parameters of the clients' end. Additionally, the loss of all models seem to consistently decrease, indicating a shared objective.

It is also worth mentioning that Figures 2 and 4 show the loss dropping much faster than the Figures 1 and 3. This indicates that method 2 can lead to faster convergence.

5.2 Error Rate

In figure 5, we see a clearer picture of how the hybrid model outperforms the Standalone FedUL Algorithm. It consistently has lower error rates and converges faster. Additionally, in figures 7 and 8, the mean error shows a steep decrease which indicates that the hybrid model handles the non-IID datasets well.

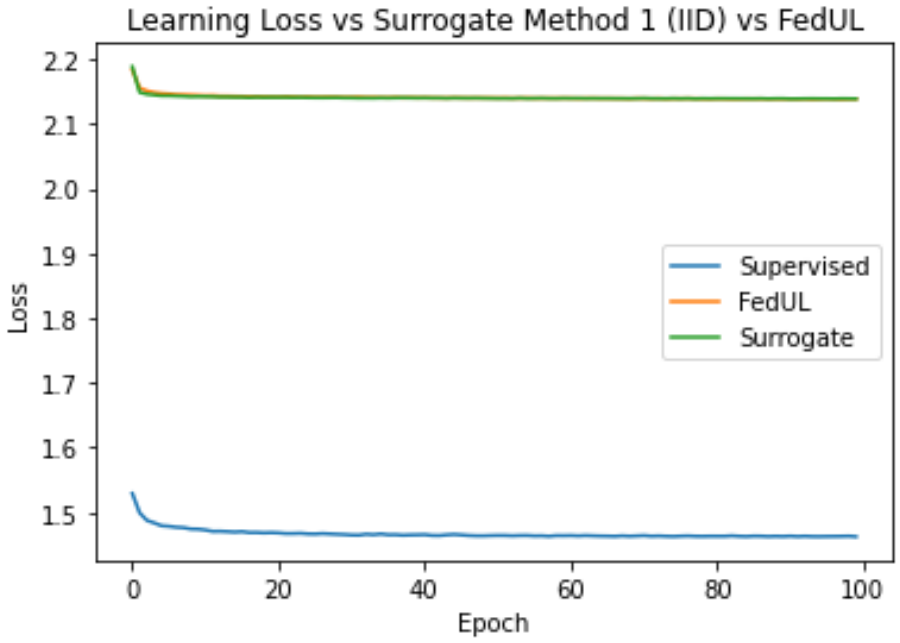

Figure 1: Modified Figure Sourced From Lecture Notes

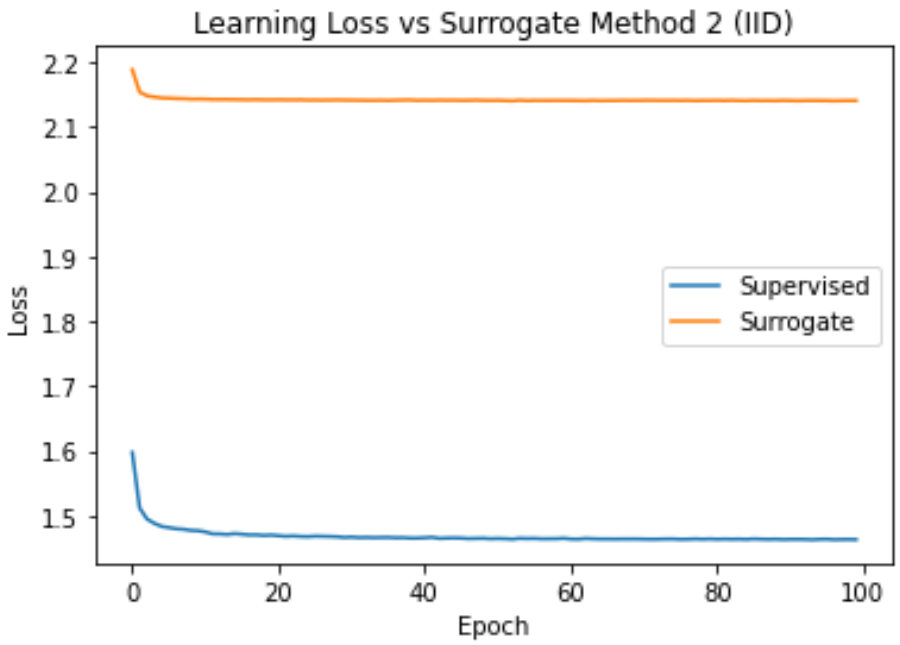

Figure 2: Modified Figure Sourced From Lecture Notes

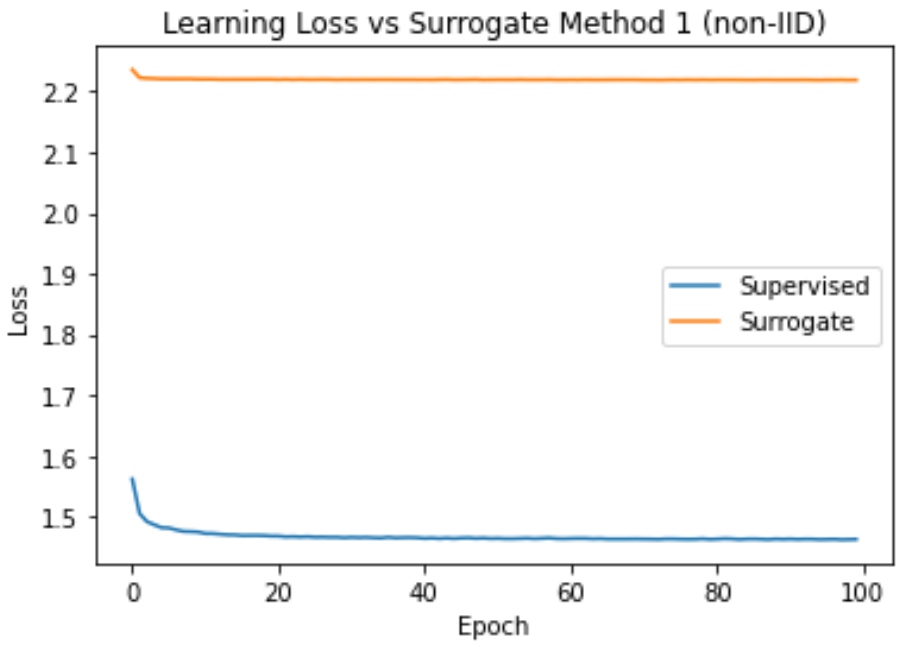

Figure 3: Modified Figure Sourced From Lecture Notes

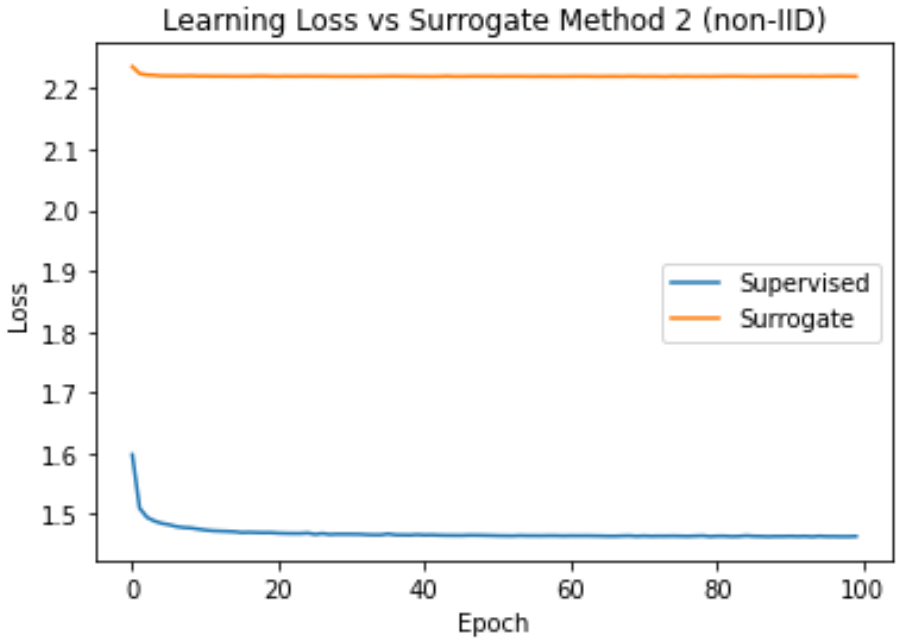

Figure 4: Modified Figure Sourced From Lecture Notes

Figure 5: Modified Figure Sourced From Lecture Notes

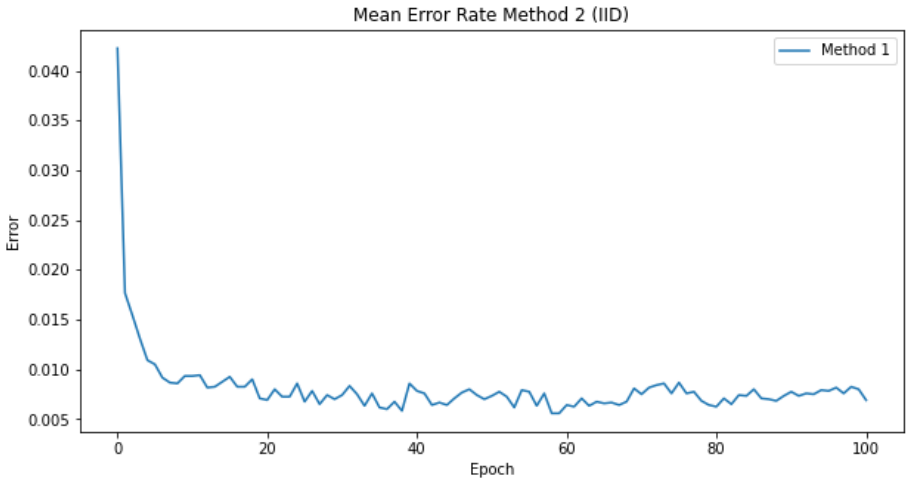

Figure 6: Modified Figure Sourced From Lecture Notes

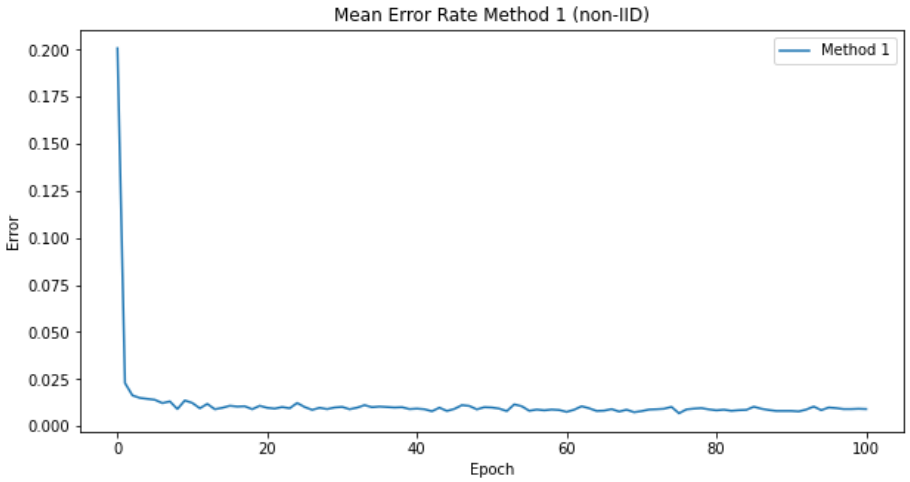

Figure 7: Modified Figure Sourced From Lecture Notes

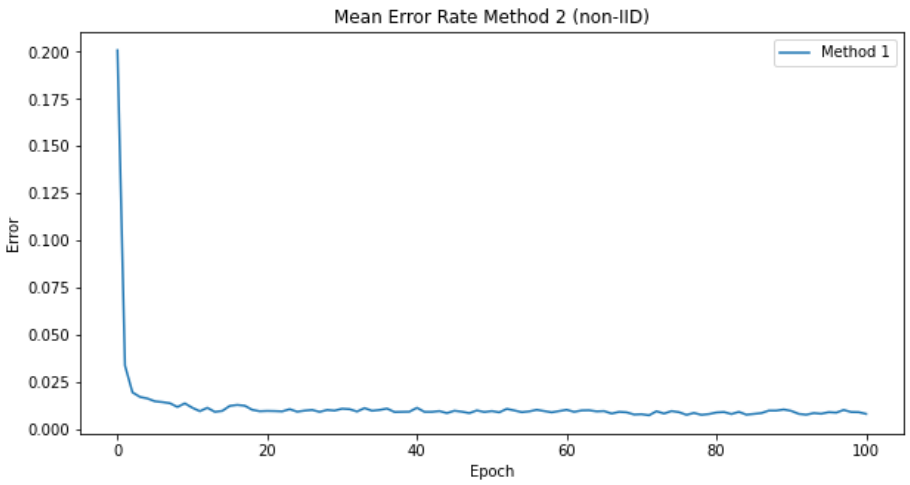

Figure 8: Modified Figure Sourced From Lecture Notes