ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746007

PRIVACY-PRESERVING FEDERATED MULTI-TASK LINEAR REGRESSION: A ONE-SHOT LINEAR MIXING APPROACH INSPIRED BY GRAPH REGULARIZATION

Harlin Lee, Andrea L. Bertozzi*, Jelena Kovačević†, Yuejie Chi‡*

*Dept. of Mathematics, University of California, Los Angeles
†Tandon School of Engineering, New York University
‡Dept. of Electrical and Computer Engineering, Carnegie Mellon University

ABSTRACT

We investigate multi-task learning (MTL), where multiple learning tasks are performed jointly rather than separately to leverage their similarities and improve performance. We focus on the federated multi-task linear regression setting, where each machine possesses its own data for individual tasks and sharing the full local data between machines is prohibited. Motivated by graph regularization, we propose a novel fusion framework that only requires a one-shot communication of local estimates. Our method linearly combines the local estimates to produce an improved estimate for each task, and we show that the ideal mixing weight for fusion is a function of task similarity and task difficulty. A practical algorithm is developed and shown to significantly reduce mean squared error (MSE) on synthetic data, as well as improve performance on an income prediction task where the real-world data is disaggregated by race.

***Index Terms*—** multi-task learning, linear regression, federated learning, graph regularization

1. INTRODUCTION

In many real-world situations, learning comes with multiple related tasks, especially in personalized learning settings such as federated learning [2]. Instead of solving them independently, multi-task learning (MTL) tackles these related tasks together to take advantage of their similarities while respecting their differences. For example, if they have varying levels of difficulty in terms of sample sizes or signal-to-noise ratios (SNRs), it is advantageous for the harder problem to borrow information from the easier problem. MTL also often occurs in a distributed setting, that is, tasks and datasets are assigned to different machines (e.g. phones, hospitals, countries). A naive approach is to give all n sets of full local data to a central server or fusion center for centralized processing. However, this poses difficulties due to privacy concerns, ownership, communication cost, or storage constraints. Therefore, this work focuses on *privacy-preserving federated multi-task learning*, where related tasks in different machines are solved jointly in a communication-efficient manner without sharing the full data.

Graph regularization is a flexible framework that drives the solutions of an optimization problem to have desired properties with respect to a graph. It is an intuitive approach to MTL that can easily integrate the task relationship information into the problem formulation, with a clear connection to communication as well as relational networks [3]. This work starts with the classic graph regularization approach, which uses a penalty function that requires joint optimization of all estimates and therefore data sharing. However, we observe that a completely different framework that does *not* require data sharing can achieve the same set of solutions under certain settings. This new perspective on graph regularization leads to the following novel approach to federated MTL, which is the focal point of this paper.

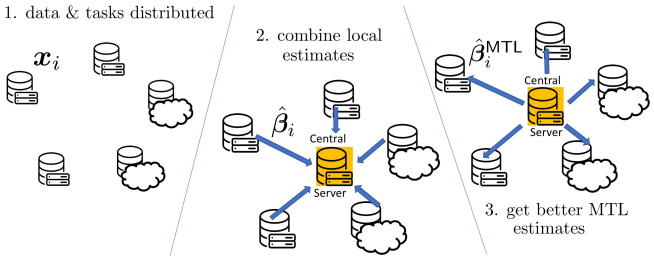


Fig. 1: Outline of the the proposed one-shot fusion method for federated multi-task learning. Only the local estimates, not the full data, are shared between the machines in a single communication round.

Specifically, we consider a scenario of n machines, where the ith machine observes the ith local dataset $\boldsymbol{x}_i$, and $\boldsymbol{x}_i$ cannot be shared outside machine i. Our goal is then to share information from $\{\boldsymbol{x}_i\}_{i=1}^n$ in a meaningful and feasible way such that we can faithfully estimate the ground truth signals $\{\boldsymbol{\beta}_i^\star\}_{i=1}^n$. Our proposed fusion approach, motivated by graph regularization, is summarized in Fig. 1. For $i = 1, \dots, n$, the ith machine calculates a linear unbiased local estimator $\widehat{\boldsymbol{\beta}}_i$ and sends it to the central server. The central server then *linearly combines* the local estimates $\{\widehat{\boldsymbol{\beta}}_i\}_{i=1}^n$ according to a mixing matrix, and produces the improved MTL estimates $\{\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}}\}_{i=1}^n$, i.e. $\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}} = \sum_{j=1}^n \mathcal{W}_{ij}\widehat{\boldsymbol{\beta}}_j$ for some matrix $\boldsymbol{\mathcal{W}} = [\mathcal{W}_{ij}] \in \mathbb{R}^{n\times n}$. This approach circumvents the aforementioned privacy concerns regarding data sharing, and only calls for a one-shot communication between the machines. Under very mild assumptions on the noise, we show that the optimal $\boldsymbol{\mathcal{W}}$ depends on task similarity and task difficulty, e.g. noise level and sample complexity, and propose a practical and straightforward algorithm for estimating $\boldsymbol{\mathcal{W}}$.

The rest of the paper is organized as follows. The proposed fusion framework is motivated via graph regularization in Section 2, defined in Section 3, and demonstrated on both synthetic and real-world data in Section 4. Finally Section 5 discusses related works, and we conclude in Section 6. The complete proofs are deferred to

Emails: {harlin, bertozzi}@math.ucla.edu, jelenak@nyu.edu, yuejiechi@cmu.edu.

This work was supported in part by the grants NSF CCF-2007911, ECCS-1818571 and DMS-1952339, ARO W911NF-18-1-0303, and ONR N00014-19-1-2404. Part of this work was completed when the first author was a graduate student at Carnegie Mellon University [1].

[1] due to length limits.

Throughout the paper, boldface letters $\boldsymbol{a}$ and $\boldsymbol{A}$ represent vectors and matrices, respectively. $\|\boldsymbol{a}\|_2$ is the ℓ_2 norm of $\boldsymbol{a}$, $\boldsymbol{A}^\top$ is the transpose of $\boldsymbol{A}$, and $\boldsymbol{A}^{-1}$ is the inverse of $\boldsymbol{A}$. $\boldsymbol{I}_p$ is the $p \times p$ identity matrix, and $\mathsf{diag}(\boldsymbol{a})$ is the diagonal matrix whose diagonal elements are $\boldsymbol{a}$. Expectation is denoted with $\mathbb{E}$, and multivariate normal distribution with $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

2. MOTIVATION: MTL VIA GRAPH REGULARIZATION

We define the multi-task linear regression problem as follows. At each machine $i = 1, \ldots, n$,

$$\boldsymbol{x}_i = \boldsymbol{A}_i \boldsymbol{\beta}_i^\star + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \sigma_i^2 \boldsymbol{I}_{m_i}), \tag{1}$$

where $\boldsymbol{x}_i$ is the observation signal or data, $\boldsymbol{\epsilon}_i \in \mathbb{R}^{m_i}$ is the noise, $\boldsymbol{\beta}_i^\star \in \mathbb{R}^d$ is the ground truth signal or model, and $\boldsymbol{A}_i \in \mathbb{R}^{m_i \times d}$ is the sensing or feature matrix. The unknown noise level $\sigma_i > 0$, and $\boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_j$ are uncorrelated for $i \neq j$. The goal is to estimate $\{\boldsymbol{\beta}_i^\star\}_{i=1}^n$ from $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{A}_i\}_{i=1}^n$.

Let us assume that we have access to (or derived) the similarity information between n tasks as an adjacency matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times n}$. In this graph, the ith node corresponds to the ith dataset, task, or machine, and the edge weight $\Gamma_{ij} \geq 0$ represents the similarity between the ith and jth nodes. Then, given some regularization parameter $\lambda > 0$, the graph-regularized MTL problem [3, 4, 5] solves for

$$\begin{aligned}&(\widehat{\boldsymbol{\beta}}_1^\lambda, \ldots, \widehat{\boldsymbol{\beta}}_n^\lambda) \\ &= \underset{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n \in \mathbb{R}^d}{\operatorname{argmin}} \left[\sum_{i=1}^n \|\boldsymbol{x}_i - \boldsymbol{A}_i \boldsymbol{\beta}_i\|_2^2 + \lambda \sum_{i,j=1}^n \Gamma_{ij} \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2^2 \right] \end{aligned} \tag{2}$$

For simplicity of exposition, set $\Gamma_{ii} = 0$ and scale $\sum_{j=1}^n \Gamma_{ij} = 1$ for all i. Also assume $\boldsymbol{A}_i$ is tall and orthogonal, i.e. $m_i \geq d$ and $\boldsymbol{A}_i^\top \boldsymbol{A}_i = \boldsymbol{I}$, and denote the local ordinary least squares (OLS) estimate as

$$\widehat{\boldsymbol{\beta}}_i^{\mathsf{OLS}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^d}{\operatorname{argmin}} \|\boldsymbol{x}_i - \boldsymbol{A}_i \boldsymbol{\beta}\|_2^2 = \boldsymbol{A}_i^\top \boldsymbol{x}_i. \tag{3}$$

Then Theorem 1 states that $\widehat{\boldsymbol{\beta}}_i^\lambda$'s are in fact *convex combinations* of local OLS estimates. Proof follows from KKT stationarity conditions, i.e. setting the gradient of (2) to be 0, and properties of $\boldsymbol{\Gamma}$.

Theorem 1 (Graph-regularized Multi-task Linear Regression)**.** *Under the assumptions of Section 2, the minimizers $\{\widehat{\boldsymbol{\beta}}_i^\lambda\}_{i=1}^n$ of graph-regularized linear regression* (2) *for $\lambda > 0$, are convex combinations of local OLS estimates $\{\widehat{\boldsymbol{\beta}}_i^{\mathsf{OLS}}\}_{i=1}^n$ (cf.* (3)*). More precisely,*

$$\widehat{\boldsymbol{\beta}}_i^\lambda = \sum_{j=1}^n \Theta_{ij} \, \widehat{\boldsymbol{\beta}}_j^{\mathsf{OLS}}$$

for mixing matrix $\boldsymbol{\Theta} \in \mathbb{R}^{n \times n}$, which is a right stochastic matrix and defined as

$$\boldsymbol{\Theta} = \frac{1}{\lambda + 1} \sum_{k=0}^\infty \left(1 - \frac{1}{\lambda + 1}\right)^k \boldsymbol{\Gamma}^k.$$

While the quadratic nature of (2) is well-studied, Theorem 1 under the simplifying assumptions highlights the privacy-preserving aspect of the MTL solutions. On one hand, we can solve (2) which uses the classical optimization-based graph regularization framework. On the other hand, we can arrive at the same answers by taking convex combinations of local OLS estimates, which no longer requires data sharing. This fresh view on graph regularization motivates a general privacy-preserving approach to federated MTL.

3. A ONE-SHOT LINEAR MIXING APPROACH TO FEDERATED MTL

Theorem 1 suggests that linearly combining local estimates is a valid approach to combining information without combining data. Building on that intuition, we propose MTL estimates

$$\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}} = \sum_{j=1}^n \mathcal{W}_{ij} \widehat{\boldsymbol{\beta}}_j, \quad i = 1, \ldots, n, \tag{4}$$

which are linear combinations of local estimates $\{\widehat{\boldsymbol{\beta}}_j\}_{j=1}^n$ according to a mixing matrix $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{n \times n}$. This may be viewed as learning a new graph from the data such that $\boldsymbol{\mathcal{W}}$ is the diffusion operator or the averaging operator for the local estimates defined on the graph. Theorem 2 specifies the mixing matrix $\boldsymbol{\mathcal{W}}$ with maximum mean squared error (MSE) reduction for any linear local estimates $\{\widehat{\boldsymbol{\beta}}_j\}_{j=1}^n$. The proof of Theorem 2 follows from directly minimizing the MSE of $\widehat{\boldsymbol{\beta}}^{\mathsf{MTL}}$ with respect to $\boldsymbol{\mathcal{W}}$ and assumptions on noise such as uncorrelation between tasks. Note that unlike the motivating example in Section 2, we no longer assume tall and orthogonal $\boldsymbol{A}_i$, nor OLS $\widehat{\boldsymbol{\beta}}_i$ for each machine $1 \leq i \leq n$.

Theorem 2 (Fusion of Linear Estimators)**.** *Assume observation model* (1)*. Let $\widehat{\boldsymbol{\beta}}_i$ be any linear unbiased local estimator of $\boldsymbol{\beta}_i^\star$, which has an expected value $\boldsymbol{\beta}_i^\star$ and variance $\mathbb{E}\|\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i^\star\|_2^2$, where the expectation is taken with respect to randomness in the ith dataset. For $\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}}$ as defined in* (4)*, the MSE $\mathbb{E}\|\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}} - \boldsymbol{\beta}_i^\star\|_2^2$ is minimized for all $i = 1, \ldots, n$ by the mixing matrix*

$$\boldsymbol{\mathcal{W}} = \boldsymbol{C} \left(\boldsymbol{C} + \boldsymbol{V}\right)^{-1},$$

where

$$\boldsymbol{C} = \left[\langle \boldsymbol{\beta}_i^\star, \boldsymbol{\beta}_j^\star \rangle\right]_{i,j=1}^n, \boldsymbol{V} = \mathsf{diag}\left(\left[\mathbb{E}\|\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i^\star\|_2^2\right]_{i=1}^n\right).$$

It is straightforward to see that the fusion estimate $\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}}$ is always at least as accurate as $\widehat{\boldsymbol{\beta}}_i$ in terms of MSE by the optimization criteria, i.e. for each $i = 1, \ldots, n$,

$$\mathbb{E}\|\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}} - \boldsymbol{\beta}_i^\star\|_2^2 \leq \mathbb{E}\|\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i^\star\|_2^2.$$

Theorem 2 states that the ideal mixing weights $\boldsymbol{\mathcal{W}}$ depend on task difficulties and task similarities. For one, $V_{ii} = \mathbb{E}\|\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i^\star\|_2^2$ is precisely the variance of the local estimate $\widehat{\boldsymbol{\beta}}_i$, which captures information about the randomness in the ith dataset, e.g. noise level σ_i and number of samples m_i. Therefore we describe $\boldsymbol{V}$ as the *task difficulty* term. Meanwhile, we categorize $\boldsymbol{C}$ as *task similarity* term: $C_{ij} = \langle \boldsymbol{\beta}_i^\star, \boldsymbol{\beta}_j^\star \rangle$ is clearly proportional to the cosine similarity or angle between the two ground truth signals.

3.1. Fusion Algorithm

An obvious shortcoming of directly using the results of Theorem 2 to calculate $\boldsymbol{\mathcal{W}}$ is that $\boldsymbol{C}$ uses the inner product with the ground truth signal, which is impossible to implement in practice. Therefore, we adopt an iterative approach to address this issue, giving rise to the proposed Iterative Fusion algorithm (cf. Alg. 1). We approximate $\boldsymbol{\beta}_i^\star \approx \widehat{\boldsymbol{\beta}}_i$ to calculate the key matrix $\boldsymbol{C}$.

This algorithm avoids concerns about data sharing by compressing the necessary information from each dataset into $\widehat{\boldsymbol{\beta}}_i$ and V_{ii}. For

Algorithm 1 Iterative Fusion for Multi-task Linear Regression

inputs local datasets $\{\boldsymbol{x}_i\}_{i=1}^n$, $\{\boldsymbol{A}_i\}_{i=1}^n$.
Each machine *locally* calculates a linear unbiased estimate $\widehat{\boldsymbol{\beta}}_i$ from $\boldsymbol{x}_i$ and $\boldsymbol{A}_i$.
Each machine *locally* estimates $\mathbb{E}\|\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i^\star\|_2^2$, e.g. by bootstrapping [6].
$\boldsymbol{V} \leftarrow \mathsf{diag}\left(\left[\mathbb{E}\|\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i^\star\|_2^2\right]_{i=1}^n\right)$
repeat
  $\boldsymbol{C} \leftarrow \left[\langle\widehat{\boldsymbol{\beta}}_i, \widehat{\boldsymbol{\beta}}_j\rangle\right]_{i,j}$
  $\boldsymbol{\mathcal{W}} \leftarrow \boldsymbol{C}\left(\boldsymbol{C}+\boldsymbol{V}\right)^{-1}$
  (Optional) Threshold elements of $\boldsymbol{\mathcal{W}}$.
  $\widehat{\boldsymbol{\beta}}_i \leftarrow \sum_{j=1}^n \mathcal{W}_{ij}\widehat{\boldsymbol{\beta}}_j$.
until termination
(Optional) Project $\widehat{\boldsymbol{\beta}}_i$ onto a constraint set.
$\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}} \leftarrow \widehat{\boldsymbol{\beta}}_i$.
outputs MTL estimates $\{\widehat{\boldsymbol{\beta}}_i^{\mathsf{MTL}}\}_{i=1}^n$.

example, if we have tall and orthogonal $\boldsymbol{A}_i$s and decide to combine OLS local estimates, then each machine will estimate σ_i from their dataset, and pass $d\sigma_i^2$ with $\widehat{\boldsymbol{\beta}}_i$ to the central server. Note that the communication between the local machines and the central server is limited to one round, while the fusion step at the central server alternates between updating the weights $\boldsymbol{\mathcal{W}}$ and the local estimates $\{\widehat{\boldsymbol{\beta}}_i\}_{i=1}^n$. The number of iterations for the fusion step can be fixed (simulations suggest 2 to 3), or chosen for the specific dataset via cross validation.

4. NUMERICAL EXPERIMENTS

Python code for the following experiments are published on `https://github.com/HarlinLee/multitask-fusion`.

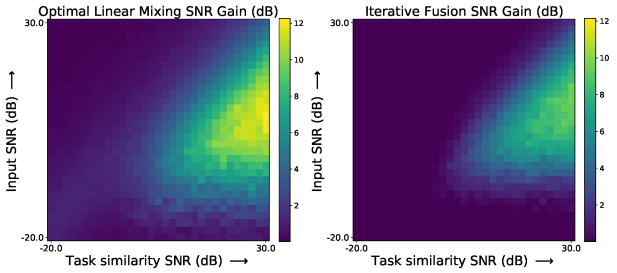
4.1. Simulations

We test our algorithm by combining local OLS estimates according to our mixing matrix. We simulate tall and orthogonal sensing matrix $\boldsymbol{A}_i \in \mathbb{R}^{m_i \times d}$ by sampling each element from $\mathcal{N}(0,1)$ i.i.d, and orthogonalizing the matrix. We choose $m_i = d$ w.l.o.g. and gather observations via (1). Orthogonal $\boldsymbol{A}_i$ is not necessary for our algorithm, but is chosen to simplify the experiment. Variables $\boldsymbol{\beta}_i^\star$, σ_i, and n are determined as follows for different experiments.

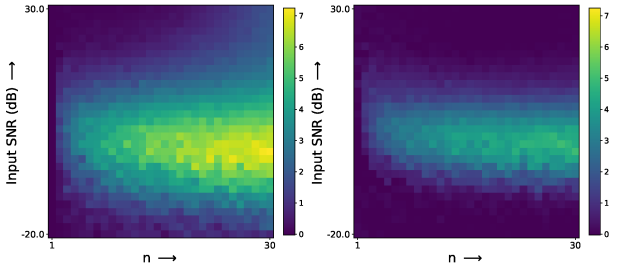
Central Model: This model assumes that all tasks are similar to each other by roughly the same degree. More concretely, set

$$\boldsymbol{\beta}_i^\star \sim \mathcal{N}(\boldsymbol{\beta}^\star, \sigma_\star^2 \mathbf{I}_d),\ \boldsymbol{\beta}^\star \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d),\ \sigma = \sigma_1 = \ldots = \sigma_n, \quad (5)$$

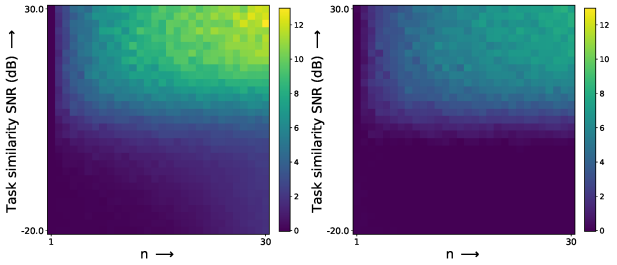
such that ground truth task similarity is uniformly determined by $\sigma_\star$, and task difficulty σ is identical across i. Note that $\sigma_\star = 0$ reduces the model to a distributed consensus scenario. We define *task similarity SNR* as $\mathsf{SNR}_{\mathsf{sim}} = 10\log_{10}(\|\boldsymbol{\beta}^\star\|_2^2/d{\sigma_\star}^2)$ dB, task input SNR as $\mathsf{SNR}_{\mathsf{in}} = 10\log_{10}(\|\boldsymbol{\beta}^\star\|_2^2/d\sigma^2)$ dB, and task output SNR as $\mathsf{SNR}_{\mathsf{out}} = -10\log_{10}(\mathsf{MSE}_{\mathsf{out}})$ dB, where $\mathsf{MSE}_{\mathsf{out}} = \frac{1}{nd}\sum_{i=1}^n \|\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i^\star\|_2^2$.



(a) Fusion helps when tasks are more similar to each other ($\mathsf{SNR}_{\mathsf{sim}}$ ↑), and if ground truth signals are more similar to each other than they are to noise ($\mathsf{SNR}_{\mathsf{sim}} \geq \mathsf{SNR}_{\mathsf{in}}$). $n = d = 20$.



(b) Fusion helps when tasks are hard enough that collaboration helps, but not too hard that there are no useful information to share (mid-to-low $\mathsf{SNR}_{\mathsf{in}}$). $\mathsf{SNR}_{\mathsf{sim}} = 10$dB, $\sigma_\star = \sqrt{0.1}$.



(c) Fusion helps when there are more tasks (n ↑). $\mathsf{SNR}_{\mathsf{in}} = 0$dB, $\sigma_i = 1$.

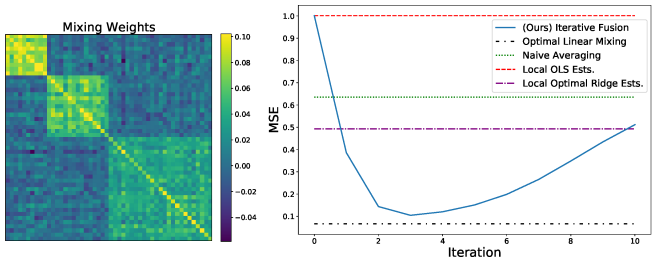
Fig. 2: SNR gain compared to the *optimal* local ridge regression estimates. Data are simulated under the central model (5). These phase diagrams visualize the regimes where the fusion method is effective. Here, $d = 20$. Averaged over 10 trials.

To understand the effect of $\mathsf{SNR}_{\mathsf{sim}}$, $\mathsf{SNR}_{\mathsf{in}}$, and n on the proposed method, we varied $\sigma_\star \in \{10^{-1.5}, \ldots, 10^1\}$, $\sigma_i \in \{10^{-1.5}, \ldots, 10^1\}$, and $n \in \{1, \ldots, 30\}$. Then under each set of parameters, we simulated data under the central model, combined *local OLS estimates* via Alg. 1, and for comparison, solved *ridge regression* for each local dataset with optimal λ_i such that the MSE of $\widehat{\boldsymbol{\beta}}_i$ is minimized. Alg. 1 is run for 10 steps, and the lowest MSE in hindsight is reported as $\mathsf{MSE}_{\mathsf{out}}$.

The SNR gain is summarized as phase transition diagrams in Fig. 2. We stress that while our methods combined the local OLS estimates, $\mathsf{SNR}_{\mathsf{out}}$ of our methods are compared against $\mathsf{SNR}_{\mathsf{out}}$ of the *optimal local* ridge regression estimates. When compared to the local OLS estimates, the SNR gain is even more substantial. Furthermore, these results suggest that fusion algorithms are beneficial when 1) ground truth signals are more similar to each other than they are to noise, 2) tasks are more similar to each other, 3) tasks are not too difficult that there is no useful information to be borrowed from

others, but also not too easy that there is *no need to* borrow information from others, and 4) there are more tasks.

Community model: The n tasks are divided into 3 groups (each with a portion of 20%, 30%, and 50%), and within each group, data are simulated following the central model (5). It is notable that the community structure is captured by the mixing weights in Fig. 3a, since the algorithm does not make that assumption a priori, nor take in parameters such as the number of groups. Fig. 3b demonstrates the successful MSE reduction by the proposed fusion algorithm.



(a) (b)

Fig. 3: (a) Mixing weights produced by fusion algorithm under the community model recover the community structure among the tasks, which were not provided a priori. (b) Combining OLS estimates can yield better estimates than local optimal ridge regression estimates. MSE reduction by fusion algorithms under the community model. Here, $\sigma^\star = 0.1, \sigma = 1, n = 50, d = 100$. Averaged over 20 trials.

4.2. Income Prediction

It is becoming increasingly evident that many aspects of life in the United States are segregated by race and socioeconomic class. Across these different subpopulations, the optimal models for a task may be related but not identical, and data may not be shared. To simulate such a setting that can benefit from federated MTL, we divide the popular "Adult" UCI dataset [7] according to the race of the individual (White, Black, Asian-Pac-Islander, Amer-Indian-Eskimo, Other), and assign each racial group to a machine, i.e. $n = 5$. For each group, we calculate local OLS estimates that predict from the census data whether yearly income exceeds \$50K. The estimates are then combined according to our fusion method, and their prediction powers are measured on each machine's test sets. The area under the receiver operating characteristic (AUC) is used as a surrogate for MSE of the prediction model, as we do not have access to the ground truth in this real-world data; AUC closer to 1.0 indicates a more accurate classifier. We estimate V_{ii} in Alg. 1 using $\sigma_i^2 \approx \|\boldsymbol{x}_i^{\text{train}} - \boldsymbol{A}_i^{\text{train}} \widehat{\boldsymbol{\beta}}_i^{\text{OLS}}\|_2^2 / m_i$. Table 1 demonstrates that our fusion method improves the AUC significantly on the subsets boxed in red, which have fewer samples.

Race	Train, test (samples)	OLS (AUC)	Fusion (AUC)
White	25933, 12970	0.800	0.801
Black	2817, 1411	0.832	0.825
A-P-I	895, 408	0.785	0.783
A-I-E	286, 149	0.670	0.805
Other	231, 122	0.780	0.836

Table 1: Classification results of Adult dataset. Our fusion method improves the AUC significantly on the subsets with fewer samples (boxed in red). Here, A-P-I stands for Asian-Pac-Islander, and A-I-E for Amer-Indian-Eskimo.

5. RELATED WORKS AND CONNECTIONS

Taking a linear combination of local estimates is not a new idea in distributed learning. However, existing literature focuses on reaching a consensus, and forces the weights to sum to 1 (i.e. weighted average) [8, 9], or simply takes the naive average [10, 11]. Our work unifies all averaging-based methods in distributed (consensus) learning literature as a special case of distributed MTL with identical tasks. In [12], the unity constraint in averaging is eliminated, but the scope is limited to distributed consensus, ridge regression local estimates, and random-effects model assumption on $\boldsymbol{\beta}^\star$, which is integral to their analysis based on random matrix theory. In comparison, our framework is applicable to other linear local estimators and does not rely on any assumptions on the ground truth signal $\boldsymbol{\beta}^\star$. However, it *can* be specialized for each application as in [12], which highlights its potential for future works in linear mixing.

A different approach is to combine the local gradient updates instead of the local model weights, which is proposed in many federated learning and distributed learning on graph [4, 13, 14] literature as they are tied to (stochastic) gradient descent. The MOCHA algorithm [15] updates their models and relationship matrix iteratively, but it requires multiple communication rounds between the local machines and the parameter server, while ours is limited to one. Model interpolation for personalized federated learning in [16] and [17] are related, but the local machine only takes a weighted average of a central model and its own model.

Another popular body of work for MTL is shared architecture or shared representation learning [18, 19, 20]. These approaches require joint optimization, and are fundamentally different from our "post-hoc" method. Unlike [21, 22], we do not assume group sparsity. Lastly, meta-learning-based methods [23, 24, 25] differ from ours in that they focuses on finding a good initialization model—a central model that can go through a few gradient updates at the local machines. Their origins are closer to transfer learning, where model from the source task is used to initialize the target task.

6. CONCLUSIONS

We proposed a novel fusion framework for federated MTL that linearly combines local estimates to get improved estimates for each task, while bypassing the restrictions on data sharing. Motivated by graph regularization solutions, we developed a concrete but simple and communication-efficient algorithm for multi-task linear regression with any unbiased linear estimators. While we use graph regularization to motivate the fusion approach, the resulting method diverges significantly. When tested on simulated data, combining local OLS estimates according to our proposed methods significantly surpassed the performance of *optimal* local ridge regression estimates under a wide range of conditions. Its performance was also demonstrated on an income prediction task with real data that has been disaggregated by race. Ongoing works include extensions of this framework to biased estimators for linear regression, e.g. ridge regression estimates, and other MTL problems, e.g. principal components analysis (PCA).

7. REFERENCES

[1] Harlin Lee, *Better Inference with Graph Regularization*, Ph.D. thesis, Carnegie Mellon University, Aug 2021.

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[3] Roula Nassif, Stefan Vlaski, Cédric Richard, Jie Chen, and Ali H Sayed, "Multitask learning over graphs: An approach for distributed, streaming machine learning," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14–25, 2020.

[4] Weiran Wang, Jialei Wang, Mladen Kolar, and Nathan Srebro, "Distributed stochastic multi-task learning with graph regularization," *arXiv preprint arXiv:1802.03830*, 2018.

[5] Rohan Varma, Harlin Lee, Jelena Kovačević, and Yuejie Chi, "Vector-valued graph trend filtering with non-convex penalties," *IEEE transactions on signal and information processing over networks*, vol. 6, pp. 48–62, 2019.

[6] Bradley Efron and Robert J Tibshirani, *An introduction to the bootstrap*, CRC press, 1994.

[7] Ronny Kohavi and Barry Becker, "Adult," UCI Machine Learning Repository, 1996.

[8] Edgar Dobriban and Yue Sheng, "Distributed linear regression by averaging," *The Annals of Statistics*, vol. 49, no. 2, pp. 918–943, 2021.

[9] Sen Lin, Guang Yang, and Junshan Zhang, "A collaborative learning framework via federated meta-learning," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 289–299.

[10] Yuchen Zhang, John C Duchi, and Martin J Wainwright, "Communication-efficient algorithms for statistical optimization," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3321–3363, 2013.

[11] Vasileios Charisopoulos, Austin R Benson, and Anil Damle, "Communication-efficient distributed eigenspace estimation," *arXiv preprint arXiv:2009.02436*, 2020.

[12] Edgar Dobriban and Yue Sheng, "Wonder: Weighted one-shot distributed ridge regression in high dimensions.," *Journal of Machine Learning Research*, vol. 21, no. 66, pp. 1–52, 2020.

[13] Sulin Liu, Sinno Jialin Pan, and Qirong Ho, "Distributed multi-task relationship learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 937–946.

[14] Shuxiao Chen, Qinqing Zheng, Qi Long, and Weijie J Su, "A theorem of the alternative for personalized federated learning," *arXiv preprint arXiv:2103.01901*, 2021.

[15] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar, "Federated multi-task learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4427–4437.

[16] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.

[17] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[18] Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei, "Few-shot learning via learning the representation, provably," *arXiv preprint arXiv:2002.09434*, 2020.

[19] Nilesh Tripuraneni, Chi Jin, and Michael I Jordan, "Provable meta-learning of linear representations," *arXiv preprint arXiv:2002.11684*, 2020.

[20] Sen Wu, Hongyang R Zhang, and Christopher Ré, "Understanding and improving information transfer in multi-task learning," *arXiv preprint arXiv:2005.00944*, 2020.

[21] Jialei Wang, Mladen Kolar, and Nathan Srerbo, "Distributed multi-task learning," in *Artificial intelligence and statistics*. PMLR, 2016, pp. 751–760.

[22] Dominic Richards, Sahand N. Negahban, and Patrick Rebeschini, "Decentralised sparse multi-task regression," *arXiv preprint arXiv:1912.01417*, 2019.

[23] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He, "Federated meta-learning with fast convergence and efficient communication," *arXiv preprint arXiv:1802.07876*, 2018.

[24] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar, "Personalized federated learning: A meta-learning approach," *arXiv preprint arXiv:2002.07948*, 2020.

[25] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.