INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Communication-Efficient Agnostic Federated Averaging

Jae Ro[1], *Mingqing Chen*[1], *Rajiv Mathews*[1], *Mehryar Mohri*[1,2], *Ananda Theertha Suresh*[1]

[1]Google Inc., U.S.A.
[2]Courant Institute of Mathematical Sciences, U.S.A.
{jaero,mingqing,mathews,mohri,theertha}@google.com

Abstract

In distributed learning settings such as federated learning, the training algorithm can be potentially biased towards different clients. [1] proposed a domain-agnostic learning algorithm, where the model is optimized for any target distribution formed by a mixture of the client distributions in order to overcome this bias. They further proposed an algorithm for the cross-silo federated learning setting, where the number of clients is small. We consider this problem in the cross-device setting, where the number of clients is much larger. We propose a communication-efficient distributed algorithm called AGNOSTIC FEDERATED AVERAGING (or AGNOSTICFEDAVG) to minimize the domain-agnostic objective proposed in [1], which is amenable to other private mechanisms such as secure aggregation. We highlight two types of naturally occurring domains in federated learning and argue that AGNOSTICFEDAVG performs well on both. To demonstrate the practical effectiveness of AGNOSTICFEDAVG, we report positive results for large-scale language modeling tasks in both simulation and live experiments, where the latter involves training language models for Spanish virtual keyboard for millions of user devices.

1. Introduction

In *federated learning* (FL), a global model is trained on decentralized data from a large number of clients, which may be mobile phones, other edge devices, or sensors [2, 3, 4]. The training data remains distributed over the clients, thus providing a layer of privacy during model training. However, FL also raises several types of issues, both practical and algorithmic, that have been the topic of multiple research efforts. This includes efficient communication strategies [2, 3, 5, 6, 7], differential privacy algorithms [8, 9], lower bound guarantees for parallel stochastic optimization [10], better optimization algorithms [11, 12, 13, 14, 15, 16], and algorithms for adaptation, multi-task learning, and personalization [17, 18, 19, 20, 21, 22]. We refer readers to [23] and [24] for a detailed literature survey on FL. FL is typically studied in two scenarios: *cross-silo* and *cross-device*. In cross-silo FL, the number of clients is small, where as in cross-device FL, the number of clients is very large and can be in the order of millions.

Fairness is a key objective in general machine learning [25, 26] and especially FL [27, 28], where the network of clients can be massive and heterogeneous. Standard learning objectives in FL minimize the loss with respect to the uniform distribution over all samples. [1] argued that, in many common instances, the uniform distribution is not the natural objective distribution as the data observed during training and inference in FL can differ. This is, in part, because models are typically trained on client devices under certain conditions (e.g. device is charging, is connected to an un-metered network, is idle, etc.), whereas during inference, these conditions need not be met. Hence it's risky to seek to minimize the expected loss with respect to a specific distribution. To overcome this, they proposed a new framework, *agnostic federated learning*, where the centralized model is optimized for any possible target distribution formed by a mixture of the client distributions. Instead of optimizing for a specific distribution, which has the high risk of a mismatch with the target, they defined an agnostic and more risk-averse objective. They further showed generalization guarantees for this new objective and proposed a stochastic mirror descent type algorithm to minimize this objective.

However, their approach and algorithm did not address some key scenarios in FL. Firstly, their algorithm is feasible in the cross-silo setting, where the number of clients is small and the samples per client is large. However, in the cross-device setting, where the number of clients is very large, we argue that their model yields very loose generalization bounds. Secondly, their algorithm did not fully address the important communication bottleneck and decentralized data issues [4] inherent in the cross-device FL setting. A straightforward implementation of their approach requires running a federated algorithm for a few hundred thousand rounds, which is not feasible in the cross-device setting.

In this paper, we overcome these bottlenecks and propose a communication-efficient federated algorithm called AGNOSTIC FEDERATED AVERAGING (or AGNOSTICFEDAVG) to minimize the agnostic learning objective in the cross-device setting. AGNOSTICFEDAVG is not only communication-efficient, but also amenable to privacy preserving techniques such as secure aggregation [29]. The rest of the paper is organized as follows. In Section 2, we state the notation and overview existing results, in Section 3, we define the framework, and in Section 4, we propose our algorithm. Finally, in Section 5, we evaluate the proposed algorithm on different synthetic and live user datasets.

2. Preliminaries and Previous Work

We start with some general notation and definitions. Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the output space. A distribution $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$.

We will primarily discuss a multi-class classification problem where $\mathcal{Y}$ is a finite set of classes, but much of our results can be extended straightforwardly to regression and other problems. The hypotheses we consider are of the form $h: \mathcal{X} \to \Delta_{\mathcal{Y}}$, where $\Delta_{\mathcal{Y}}$ stands for the simplex over $\mathcal{Y}$. Thus, $h(x)$ is a probability distribution over the classes or categories that can be assigned to $x \in \mathcal{X}$. We will denote by $\mathcal{H}$ a family of such hypotheses h. We also denote by ℓ a loss function defined over $\Delta_{\mathcal{Y}} \times \mathcal{Y}$ taking non-negative values. The loss of $h \in \mathcal{H}$ for a labeled sample $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is given by $\ell(h(x), y)$. One key example in applications is the cross-entropy loss, which is defined as $\ell(h(x), y) = -\log(\mathbb{P}_{y' \sim h(x)}[y' = y])$. We will denote by $\mathcal{L}_{\mathcal{D}}(h)$ the expected loss of a hypothesis h with respect to a

http://dx.doi.org/10.21437/Interspeech.2021-153

Algorithm 1 AGNOSTICFEDAVG

procedure SERVER
    $w_0 \in \mathcal{W}$, $\lambda_0 \in \Delta_p$, $\mathbf{N}_0 \in \mathbb{N}^p$
    for round $t = 1$ to T **do**
        $\alpha_t \leftarrow \frac{\lambda_{t-1}}{\sum_{j=t-r}^{t-1} \mathbf{N}_j / r}$
        $C_t \leftarrow$ (random set of c clients)
        for client $k \in C_t$ **do**
            $w_t^k, \beta_t^k, \mathbf{L}_t^k, \mathbf{N}_t^k \leftarrow \text{CLIENT}(k, w_{t-1}, \alpha_t)$
        end for
        $w_t \leftarrow \sum_{k \in C_t} \beta_t^k w_t^k / \beta_t$
        $\mathbf{N}_t \leftarrow \sum_{k \in C_t} \mathbf{N}_t^k$
        $\mathbf{L} \leftarrow \sum_{k \in C_t} \mathbf{L}_t^k / \mathbf{N}_t$
        $\lambda_t \leftarrow \frac{\lambda_{t-1} \cdot \exp(\gamma_\lambda \mathbf{L})}{\sum_{i=1}^p \lambda_{t-1}^i \cdot \exp(\gamma_\lambda \mathbf{L}_i))}$
    end for
end procedure

procedure CLIENT(k, w, α) ▷ Run on client k
    for domain $i = 1$ to p **do**
        $\mathbf{L}_i^k \leftarrow |S_k \cap \widehat{\mathcal{D}}_i| \cdot \mathsf{L}(w, S_k \cap \widehat{\mathcal{D}}_i)$
        $\mathbf{N}_i^k \leftarrow |S_k \cap \widehat{\mathcal{D}}_i|$
        $\beta^k \leftarrow \sum_{i=1}^p \alpha_i |S_k \cap \widehat{\mathcal{D}}_i|$
    end for
    $\mathbf{B} \leftarrow$ (split S_k into batches of size B)
    for epoch $e = 1$ to E **do**
        for batch $b \in \mathbf{B}$ **do**
            $w \leftarrow w - \gamma_w \nabla \left(\frac{\sum_{i=1}^p \alpha_i \sum_{j \in b \cap \widehat{\mathcal{D}}_i} \mathsf{L}(w, x_j, y_j)}{\beta^k} \right)$
        end for
    end for
    return $w, \beta^k, \mathbf{L}^k, \mathbf{N}^k$
end procedure

distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, $\mathcal{L}_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$ and by $h_{\mathcal{D}}$ its minimizer: $h_{\mathcal{D}} = \text{argmin}_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(h)$. In standard learning scenarios, the distribution $\mathcal{D}$ is the test or target distribution, which typically coincides with the distribution of the training samples. However, in FL, this is often not the case.

In FL, the data is distributed across many heterogeneous clients and the data distribution is different for each client [24]. Let q be the total number of clients. Let $\mathcal{D}_k$ denote the data distribution for client k. The client does not have access to the true distribution $\mathcal{D}_k$ and instead has access to S_k where $S_k = ((x_{k,1}, y_{k,1}), \ldots, (x_{k,n_k}, y_{k,n_k})) \in (\mathcal{X} \times \mathcal{Y})^{n_k}$. Let $\widehat{\mathcal{D}}_k$ denote the empirical distribution associated to sample S_k of size n_k. A natural goal is to minimize the empirical risk on the average risk given by

$$\mathcal{L}_{\overline{U}}(h),$$

where $\overline{U} = \frac{1}{q} \sum_k \widehat{\mathcal{D}}_k$ is the uniform distribution over all clients data. However, as argued by [1], due to differences between the train and test distributions, minimizing this objective is risky. Hence, they proposed to minimize the loss on the worst case distribution. More concretely, for distributions $\mathcal{D}_k$, $k = 1, \ldots, q$, let $\mathcal{D}_\lambda = \sum_{k=1}^q \lambda_k \mathcal{D}_k$ for some $\lambda \in \Delta_q$, where Δ_q is the probability simplex over q clients. Thus, the learner minimizes the *empirical agnostic loss* (or *agnostic risk*) $\mathcal{L}_{\mathcal{D}_{\Delta_q}}(h)$ associated to a predictor $h \in \mathcal{H}$ as

$$\mathcal{L}_{\overline{\mathcal{D}}_{\Delta_q}}(h) = \max_{\lambda \in \Delta_q} \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h), \tag{1}$$

where $\overline{\mathcal{D}}_\lambda = \sum_{k=1}^q \lambda_k \widehat{\mathcal{D}}_k$. For simplicity, we allow any $\lambda \in \Delta_q$ in the above definition. However, the generalization bounds [1, Theorem 1] depends on $\min_k n_k$, the minimum number of samples of any client. In the cross-device setting, this yields loose bounds as each client typically only has a few hundred samples. Hence, instead of treating each client as a domain, we treat collections of clients or data pooled from clients as domains.

3. Proposed Formulation

As stated before, treating each client as a separate domain yields loose generalization bounds. Hence, we treat collections of clients as domains, which naturally leads to two types of partitions. Let there be p domains $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_p$.

1. *Data partition*: Each client has data from one or more domains and domains represent different types of data. For example, for virtual keyboard applications [30], the domains could be the application source of client inputs, such as messaging, emails, or documents. In this case, the data distribution $\mathcal{D}_k$ for client k is given by

$$\mathcal{D}_k = \sum_{i=1}^p \lambda_i \mathcal{D}_i,$$

where $\sum_{i=1}^p \lambda_i = 1$ and $\lambda_i \geq 0$ for all $i \leq p$.

2. *Client partition*: Each client has data from exactly one domain and domains represent clusters of clients. For example, clustering clients based on their geographic location yields this domain type. In this case, the data distribution $\mathcal{D}_k$ of client k is given by $\mathcal{D}_k = \mathcal{D}_i$ for some $i \leq p$.

In both of the above formulations, even though there are q different clients, the number of underlying distinct domains is p, which we argue is considerably smaller. Hence, we have a large number of samples from each of the domains and get strong generalization bounds. Since each client distribution $\mathcal{D}$ can be written as a linear combination of domain distributions,

$$\max_{\lambda \in \Delta_q} \mathcal{L}_{\mathcal{D}_\lambda}(h) \leq \max_{\lambda \in \Delta_p} \mathcal{L}_{\mathcal{D}_\lambda}(h). \tag{2}$$

However, we do not have access to the true domain distributions $\mathcal{D}_i$ and instead have samples from $\widehat{\mathcal{D}}_i$, where $\widehat{\mathcal{D}}_i$ is the empirical distribution obtained by pooling all the data of domain i. Let m_i be the number of samples in domain i. By (2), the true agnostic loss over clients is smaller than the true agnostic loss over domains. Hence we propose to minimize the empirical agnostic loss over domains,

$$\max_{\lambda \in \Delta_p} \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h),$$

where $\overline{\mathcal{D}}_\lambda = \sum_{i=1}^p \lambda_i \widehat{\mathcal{D}}_i$. The previous known generalization bounds from [1, Lemma 3, Corollary 4] yields the following generalization bound. Let $\epsilon > 0$. With probability at least $1 - \delta$, for any client k and any hypothesis h

$$\mathcal{L}_{\mathcal{D}_k}(h) \leq \max_{\lambda \in \Delta_p} \mathcal{L}_{\overline{\mathcal{D}}_\lambda}(h) + \sqrt{\frac{\ell_c}{\min_i m_i}} \left(\sqrt{d \log \frac{\sum_i m_i}{d} + p \log \frac{1}{\epsilon\delta}} \right) + \epsilon,$$

for some constant ℓ_c which depends on the maximum value of the loss and d is the Vapnik–Chervonenkis (VC) dimension of the hypothesis class $\mathcal{H}$. The above generalization bound scales inversely with $\min_i m_i$, which is the minimum number of samples in any domain. Since the number of domains p is small, as long as the domains are well-distributed, we would have a relatively large number of samples per domain and thus a favorable generalization bound in the cross-device setting.

We now propose a communication-efficient algorithm to minimize agnostic loss (1) in the cross-device setting.

4. AGNOSTICFEDAVG

[1] showed that agnostic learning can be treated as a two-player game, where a learner tries to find the best hypothesis and an adversary tries to find the domain weights λ that maximize the loss. They proposed a stochastic mirror descent algorithm and showed that the objective reaches the optimum value at a rate of $\mathcal{O}(1/\sqrt{T})$ after T rounds of training. Similar to how FEDERATEDAVERAGING (FEDAVG) of [4] is based on stochastic gradient descent (SGD) but is more communication-efficient, we propose AGNOSTICFEDAVG, that is based on [1] and is communication-efficient. In fact, a direct implementation of [1] would be infeasible in the cross-device setting, as the number of steps can be in the order of millions. Furthermore, a direct implementation of [1] requires the clients to reveal their domain to the server, which can be privacy-invasive. In contrast, the proposed algorithm AGNOSTICFEDAVG can be used with privacy preserving techniques such as secure aggregation. We design AGNOSTICFEDAVG with the following properties.

- Each round of FL uses only a single round of transmission, which includes model download from the server to the clients and upload from the clients to the server.
- The clients train with multiple local SGD steps similar to FEDAVG.
- The server does not have access to individual clients data but only aggregated statistics, making it compatible with other cryptographic techniques such as secure aggregation [29]. This provides another layer of security and prevents the server from retrieving or rebuilding privacy-sensitive information from individual client parameter updates without additional side information.

Let $\mathcal{W}$ be the set of parameters of the hypothesis class. The algorithm first initializes the weights to $w_0 \in \mathcal{W}$, domain weights to $\lambda_0 \in \Delta_p$, and the number of examples per domain to $\mathbf{N}_0 \in \mathbb{N}^p$, where $\mathbf{N}_{t,i}$ denotes the number of samples for domain i at round t and $\mathbf{N}^k$ denotes the number of samples for client k, split by domain. We keep a sliding window of the number of examples per domain $\mathbf{N}_t$ over the last r training rounds. The algorithm uses learning rate γ_λ for learning domain weights. In the following, let L denote the loss function as a function of hypothesis parameter w.

At each round of training t, the algorithm computes a scaling vector α_t by taking the ratio of domain weights λ_{t-1} and the average number of samples per domain for the last r rounds $\sum_{j=t-r}^{t-1} \mathbf{N}_j/r$. The algorithm then selects c clients randomly C_t and sends the parameters w_{t-1} and scaling vector α_t to each of them. First, each selected client k computes the number of samples per domain $\mathbf{N}^k$, initial loss per domain $\mathbf{L}^k \in \mathbf{R}^p$, and scaled client weight β^k for their local dataset. Then, each client updates the parameters w_{t-1} based on α_t and β^k by running E epochs of SGD with batch size B and learning rate γ_w. Finally, the client transmits the updated parameters w^k, weight per client β^k, initial loss per domain $\mathbf{L}^k$, and number of samples $\mathbf{N}^k$ per domain back to the server. Since this is done using secure aggregation, the server only observes the total number of samples $\mathbf{N}$ and loss $\mathbf{L}$ per domain across clients. The server then computes the new parameters w_t by averaging the client updates weighted by β_t^k and does an exponentiated gradient (EG) step for the domain weights λ_t,

$$\lambda_t = \frac{\lambda_{t-1} \cdot \exp(\gamma_\lambda \mathbf{L})}{\sum_{i=1}^p \lambda_{t-1}^i \cdot \exp(\gamma_\lambda \mathbf{L}_i))}.$$

If a round does not have any samples from a particular domain, we set $\mathbf{L}$ to zero for that round. This process is repeated for T rounds. The complete pseudo-code is given in Algorithm 1.

To see why the above algorithm aims to minimize the agnostic loss, consider the weighted average of all the client losses

$$\begin{aligned}
&\sum_{k \in C_t} \beta_t^k \frac{\sum_{i=1}^p \alpha_t^i \sum_{j \in S_k \cap \widehat{\mathcal{D}}_i} \mathsf{L}(w, x_j, y_j)}{\sum_{i=1}^p \alpha_t^i |S_k \cap \widehat{\mathcal{D}}_i|} \\
&= \sum_{k \in C_t} \sum_{i=1}^p \alpha_t^i \sum_{j \in S_k \cap \widehat{\mathcal{D}}_i} \mathsf{L}(w, x_j, y_j) \\
&= \sum_{i=1}^p \alpha_t^i \sum_{k \in C_t} \sum_{j \in S_k \cap \widehat{\mathcal{D}}_i} \mathsf{L}(w, x_j, y_j) \\
&\overset{(a)}{\approx} \sum_{i=1}^p \lambda_t^i \frac{\sum_{k \in C_t} \sum_{j \in S_k \cap \widehat{\mathcal{D}}_i} \mathsf{L}(w, x_j, y_j)}{N_i} = \sum_{i=1}^p \lambda_t^i \mathsf{L}_i(w),
\end{aligned}$$

where $\mathsf{L}_i(w)$ is the average loss for domain i and N_i is the number of samples in domain i from the selected clients at round t. The approximation (a) assumes that the moving average of $\mathbf{N}_{t,i}$ is close to the number of samples in domain i from the selected clients at round t. Thus, AGNOSTICFEDAVG aims to minimize the domain agnostic objective defined in (1). We further note that by using secure aggregation [29], the server only observes aggregated statistics rather than learning domains or gradients of individual clients and provides an additional layer of privacy.

The communication costs of FEDAVG and AGNOSTICFEDAVG are given in Table 1. For a given round t, AGNOSTICFEDAVG adds a small additional cost on top of the communication cost of FEDAVG, as the number of domains p is typically much smaller than the number of model parameters $|\mathcal{W}|$. Furthermore, in practice, AGNOSTICFEDAVG can use fewer communication rounds than FEDAVG, thereby reducing or eliminating this overhead entirely (Appendix A).

Table 1: *Total communication cost in number of parameters per round, where $|\mathcal{W}|$ is the number of parameters in the model.*

algorithm	number of parameters per round
FEDAVG	$2c \cdot |\mathcal{W}|$
AGNOSTICFEDAVG	$2c \cdot |\mathcal{W}| + 4c \cdot p$

5. Experiments

We report the results for the English Stack Overflow language model simulation task and a Spanish language modeling live experiment for millions of virtual keyboard user devices. We implemented all algorithms and experiments using the open-source FedJAX [31] and TensorFlow Federated [32] libraries. For all experiments, we compare three algorithms:

- **FEDAVG (uniform)**: Trained uniformly on all available data.
- **FEDAVG (target-only)**: Trained only on data from the target.
- **AGNOSTICFEDAVG**: Trained on all available data.

Table 2: *Perplexity and in-vocab-accuracy for Stack Overflow test dataset with the standard deviation for three trials in parentheses.* AGNOSTICFEDAVG *attains lowest perplexity and highest in-vocab-accuracy for the harder domain* ***answer***.

algorithm	**answer**		question		difference	
	perp.	acc.	perp.	acc.	perp.	acc.
FEDAVG (uniform)	53.1 (.06)	24.5 (.02)	**43.4 (.23)**	**27.5 (.06)**	9.7	3.0
FEDAVG (answer)	52.9 (.15)	24.9 (.02)	64.2 (.39)	21.2 (.14)	11.3	3.7
AGNOSTICFEDAVG	**51.9 (.22)**	**25.1 (.004)**	52.7 (.36)	23.5 (.05)	**0.8**	**1.6**

Table 3: *Perplexity and in-vocab-accuracy for Spanish virtual keyboard.* AGNOSTICFEDAVG *attains lowest perplexity for the harder domain* ***es-AR***.

algorithm	**es-AR**		es-419*		es-US	
	perp.	acc.	perp.	acc.	perp.	acc.
FEDAVG (uniform)	56.0	11.9	**50.5**	**11.3**	**44.2**	**10.5**
FEDAVG (es-AR)	55.0	12.1	62.2	10.2	52.8	9.5
AGNOSTICFEDAVG	**53.4**	**12.3**	60.6	10.2	52.2	9.6

Table 4: *Statistics per domain in the Stack Overflow dataset.*

	train	held-out	test
clients	342K	38.8K	204K
sentences	136M	16.5M	16.6M
answers	78.0M	9.33M	9.07M
questions	57.8M	7.17M	7.52M

We demonstrate that AGNOSTICFEDAVG attains a lower perplexity compared to FEDAVG (uniform) and FEDAVG (target-only) for both the experiments on the harder domain: answer domain for Stack Overflow and es-AR for the Spanish language model.

To verify that AGNOSTICFEDAVG correctly minimizes the domain agnostic objective and to showcase its effectiveness on non-language tasks, we also include experiments on a synthetic toy regression example and the EMNIST-62 image recognition task in Appendices B and C, respectively.

5.1. Stack Overflow Language Model

We consider the language model task for the Stack Overflow dataset from [33]. This dataset contains two domains, questions and answers, from the Stack Overflow forum grouped by client ids. This corresponds to the *data partition* domain type since an individual client can post both questions and answers. Table 4 summarizes the statistics per domain.

We match the model and training setup from [33] and train a single layer LSTM language model over the top 10K words with an Adam server optimizer and 50 clients participating per training round for 1500 rounds. For AGNOSTICFEDAVG, we use the same set up with domain weight learning rate 0.005.

For the Stack Overflow experiments, we report perplexity and in-vocab-accuracy, where in-vocab-accuracy is the number of correct predictions, without UNK (out-of-vocabulary) or EOS (end-of-sentence) tokens, divided by the number of words without the EOS token. Defining in-vocab-accuracy this way allows valid comparisons for different vocabulary sizes. The results are in Table 2. For the baseline FEDAVG (uniform), of the two domains, the answer domain is harder and has higher perplexity and lower accuracy. Given this, we also train an additional baseline FEDAVG (answer) on answer examples only. While FEDAVG (answer) does improve answer performance over FEDAVG (uniform), it results in significantly worse question performance. However, AGNOSTICFEDAVG outperforms both FEDAVG (uniform) and FEDAVG (answer) on the answers domain, while also significantly decreasing the performance disparity between answers and questions. This suggests that there could be important features in the questions that can augment performance on answers that are leveraged by AGNOSTICFEDAVG but aren't optimally weighted in FEDAVG (uniform) or are completely ignored in FEDAVG (answer).

5.2. Spanish Virtual Keyboard Language Model

We further use AGNOSTICFEDAVG to train a Coupled Input and Forget Gate (CIFG) [34] language model for Spanish on virtual keyboard client devices. We follow the same settings and FL requirements for client participation as [30]. We consider three domains based on the Spanish locales: es-US for US, es-AR for Argentina, and a subset of countries belonging to es-419[1]. Since each user device falls in a single region, this task corresponds to the *client partition*.

Similar to Section 5.1, we report perplexity and in-vocab-accuracy. For all algorithms, we use the momentum server optimizer, using Nesterov accelerated gradient [35], and 500 clients participating per training round for 3000 rounds. Over the course of training, approximately 141 million sentences are processed by 1.5 million clients. The results are in Table 3. For the baseline FEDAVG (uniform), of the three languages, es-AR has the worst perplexity. Similar to Section 5.1, training FEDAVG (es-AR) on es-AR clients only improves es-AR performance over FEDAVG (uniform) but also results in much worse performance for es-US and es-419*. Again, AGNOSTICFEDAVG improves the perplexity and accuracy on es-AR over FEDAVG (uniform) while also decreasing the regression on es-US and es-419* when compared to FEDAVG (es-AR).

6. Conclusion

We presented an algorithmic study of domain agnostic learning in the cross-device FL setting. We also examined the two types of naturally occurring domains in FL: *data partition* and *client partition* and provided example learning tasks for both in large-scale language modeling. Finally, we defined AGNOSTICFEDAVG, a communication-efficient federated algorithm that aims to minimize the domain agnostic objective proposed in [1] and can provide additional security using secure aggregation and demonstrated its practical effectiveness in simulations and real live experiments. We hope that our efforts will spur further studies into improving the practical efficiency of FL algorithms.

7. Acknowledgements

We thank our colleagues Kaan Ege Ozgun, Gary Sivek, Zachary Garrett, and Keith Rush for their help with training and experiment infrastructure, and Andrew Hard, Sean Campbell, and Françoise Beaufays for their helpful comments and discussions.

[1] Defined by UN M.49 region code. We use "es-419*" to denote these countries.

8. References

[1] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *International Conference on Machine Learning*, 2019, pp. 4615–4625.

[2] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[3] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[5] J. Hamer, M. Mohri, and A. T. Suresh, "Fedboost: A communication-efficient algorithm for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3973–3983.

[6] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, "Qsparse-local-sgd: Distributed sgd with quantization, sparsification, and local computations," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 217–226, 2020.

[7] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, "Federated learning with compression: Unified analysis and sharp guarantees," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2350–2358.

[8] N. Agarwal, A. T. Suresh, F. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed SGD," in *Proceedings of NeurIPS*, 2018, pp. 7575–7586.

[9] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete gaussian mechanism for federated learning with secure aggregation," *arXiv preprint arXiv:2102.06387*, 2021.

[10] B. E. Woodworth, J. Wang, A. D. Smith, B. McMahan, and N. Srebro, "Graph oracle models, lower bounds, and gaps for parallel stochastic optimization," in *Proceedings of NeurIPS*, 2018, pp. 8505–8515.

[11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[12] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[13] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 7611–7623. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/564127c03caab942e503ee6f810f54fd-Paper.pdf

[14] S. P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Mime: Mimicking centralized stochastic algorithms in federated learning," *arXiv preprint arXiv:2008.03606*, 2020.

[15] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *International Conference on Learning Representations*, 2021.

[16] Z. Li and P. Richtárik, "A unified analysis of stochastic gradient methods for nonconvex federated optimization," *arXiv preprint arXiv:2006.07013*, 2020.

[17] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4427–4437.

[18] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.

[19] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.

[20] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020, pp. 794–797.

[21] A. Agarwal, J. Langford, and C.-Y. Wei, "Federated residual learning," *arXiv preprint arXiv:2003.12880*, 2020.

[22] Y. Mansour, M. Mohri, J. Ro, A. T. Suresh, and K. Wu, "A theory of multiple-source adaptation with limited target labeled data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2332–2340.

[23] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[24] H. B. McMahan *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1, 2021.

[25] P. J. Bickel, E. A. Hammel, and J. W. O'Connell, "Sex bias in graduate admissions: Data from Berkeley," *Science*, vol. 187, no. 4175, pp. 398–404, 1975.

[26] M. Hardt, E. Price, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf

[27] A. Abay, Y. Zhou, N. Baracaldo, S. Rajamoni, E. Chuba, and H. Ludwig, "Mitigating bias in federated learning," 2020.

[28] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=ByexElSYDr

[29] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1175–1191.

[30] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[31] J. H. Ro, A. T. Suresh, and K. Wu, "FedJAX: Federated learning simulation with JAX," 2020. [Online]. Available: http://github.com/google/fedjax

[32] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," *CoRR*, vol. abs/1902.01046, 2019. [Online]. Available: http://arxiv.org/abs/1902.01046

[33] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," 2020.

[34] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[35] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $o(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983. [Online]. Available: https://ci.nii.ac.jp/naid/10029946121/en/

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