Tackling Distribution Shifts in Federated Learning with Superquantile Aggregation

Krishna Pillutla[*†1] **Yassine Laguel**[*2] **Jérôme Malick**[3] **Zaid Harchaoui**[1]
[1] University of Washington, Seattle, WA, USA [2] Rutgers University, New Brunswick, NJ, USA
[3] CNRS, Grenoble, France

Abstract

Federated learning has emerged as the predominant framework for distributed machine learning over decentralized data, e.g. in mobile phones. The usual approaches suffer from a distribution shift: the model is trained to fit the average population distribution but is deployed on individual clients, whose data distributions can be quite different. We present a distributionally robust approach to federated learning based on a risk measure known as the superquantile and show how to optimize it by interleaving federated averaging steps with quantile computation. We demonstrate experimentally that our approach is competitive with usual ones in terms of average error and outperforms them in terms of tail statistics of the error.

1 Introduction

Federated learning is a distributed machine learning framework where many clients (e.g. mobile devices) collaboratively train a model under the orchestration of a central server (e.g. service provider), while keeping the training data private and local to the client throughout the training process [16, 10]. It has found widespread adoption across industry [1, 18] for applications ranging from smart device apps [22, 6] to healthcare [2, 9].

A key feature of federated learning is statistical heterogeneity, i.e., client data distributions are *not* identically distributed [10, 13]. Each client is a user who generates diverse data depending on their unique personal, cultural, regional, and geographical characteristics.

This data heterogeneity in federated learning manifests itself as a train-test distributional shift. Indeed, the usual approach minimizes the prediction error of the model on average over the population of clients available for training [16] while at test time, the same model is deployed on individual clients. This approach can be liable to fail on **tail clients** whose data distribution is far from most of the population or who may have less data than most of the population. It is highly desirable, therefore, to have a federated learning method that can robustly deliver good predictive performance across a wide variety of natural distribution shifts posed by individual clients.

We present in this paper a robust approach to federated learning that guarantees a minimum level of predictive performance to all clients even in situations where the population is heterogeneous. The approach we develop addresses these issues by minimizing a learning objective based on the notion of a superquantile [20, 19], a risk measure that captures the tail behavior of a random variable. Our algorithm relies on quantile statistics of the losses to filter out clients on which to run federated averaging steps. Experimental results on benchmark datasets shows that our approach yields improved performance on tail clients over a number of state of the art baselines while maintaining competitive performance on the average error.

[*]These authors contributed equally to this work. [†]Now at Google Research.

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Proposed Objective and Optimization Algorithm

Suppose we have n clients such as mobile phones. The loss incurred by the model $w \in \mathbb{R}^d$ on this client i is $F_i(w) := \mathbb{E}_{z\sim p_i}[f(w;z)]$, where p_i is the distinct data distribution on client i and $f(w;\xi)$ is the loss function e.g. cross entropy, on data point z. The usual objective of federated learning [16] is simply the empirical risk minimization (ERM) approach

$$\min_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} F_i(w)\,. \tag{1}$$

Owing the natural statistical heterogeneity in the data, the data distribution p encountered at test time on an unseen test client might be different from the population training distribution $p_{\text{train}} = (1/n)\sum_{i=1}^{n} p_i$, leading to poor performance on such clients. Our goal is to improve the performance on such tail clients.

To this end, we directly minimize the average loss across tail clients above a certain tail threshold. We formalize this through the notion of a risk measure known as the **superquantile**, a tail summary statistic of random variables [20]. The $(1-\alpha)$-superquantile is defined for a continuous random variable Z and $\alpha \in (0,1)$ as $\mathbb{S}_\alpha(Z) = \mathbb{E}[Z \mid Z > Q_\alpha(Z)]$, where $Q_\alpha(Z)$ is the $(1-\alpha)$-quantile of Z. A similar interpretation holds for discrete distributions; it is formally defined as

$$\mathbb{S}_\alpha(u_1,\cdots,u_n) := \max\left\{\sum_{i=1}^{n}\pi_i u_i \;:\; 0\le\pi_i\le\frac{1}{\alpha n}\,\forall\, i\in[n],\ \sum_{i=1}^{n}\pi_i = 1\right\}.$$

This is an instance of the continuous knapsack problem and can be solved optimally by a greedy algorithm [4]. Assuming $u_1 < \cdots < u_n$ and αn is an integer, the optimal solution $\pi^\star$ above satisfies $\pi_i^\star = 1/(\alpha n)$ for $i \ge (1-\alpha)n$ or that the u_i's larger than their $(1-\alpha)$ quantile are averaged.

The Δ-FL Objective and Distributional Robustness. Instead of minimizing the average loss as in (1), our proposed framework, called Δ-FL, minimizes the tail loss across clients, as measured by the superquantile. Concretely, at level $\alpha \in (0,1)$, we minimize

$$F_\alpha(w) := \mathbb{S}_\alpha\big(F_1(w),\cdots,F_n(w)\big)\,. \tag{2}$$

If we have a test client whose distribution $p_\pi = \sum_{i=1}^{n}\pi_i p_i$ can be written as a mixture of the training distributions $p_i,\ldots,p_n$, then the Δ-FL objective minimizes $\max_{\pi_i\le 1/(\alpha n)}\mathbb{E}_{z\sim p_\pi}[f(w;z)]$, the *worst-case loss over all mixture distributions with a weight constraint* $\pi_i \le 1/(\alpha n)$.

Federated Optimization of Δ-FL. In order to design a federated optimization algorithm to optimize the Δ-FL objective, we must overcome two challenges: (i) nonsmoothness, and (ii) biased gradient estimation. The superquantile $a \mapsto \mathbb{S}_\alpha(u_1,\cdots,u_n)$ is a nonsmooth function, leading to potential difficulties in optimization. We overcome this challenge by deriving an expression for the subgradient of the Δ-FL objective. Concretely, when αn is an integer, we have

$$\partial F_\alpha(w) \ni \sum_{i=1}^{n}\pi_i^\star F_i(w)\,,\quad \text{where}\quad \pi_i^* = \frac{\mathbb{I}(F_i(w)\ge Q_\alpha)}{\sum_{j=1}^{n}\mathbb{I}(F_j(w)\ge Q_\alpha)}\,, \tag{3}$$

and $Q_\alpha = Q_\alpha(F_1(w),\cdots,F_n(w))$ is the $(1-\alpha)$-quantile of the losses. See Appendix B for a proof.

The second challenge stems from the lack of unbiased gradient estimators for the superquantile. Given m i.i.d. copies $Z_1,\ldots,Z_m$ of a random variable Z, the empirical mean $\bar{Z}_m = (1/m)\sum_{i=1}^{m} Z_i$ is an unbiased estimate of the population mean, i.e., $\mathbb{E}[\bar{Z}_m] = \mathbb{E}[Z]$. This is no longer true for the superquantile, i.e., $\mathbb{E}[\mathbb{S}_\alpha(Z_1,\cdots,Z_m)] \ne \mathbb{S}_\alpha(Z)$. As a result, we do not have access to unbiased stochastic gradients (here, m is the batch size). In federated learning, it is not reasonable to assume that we have access to all the clients due to a diurnal availability pattern of clients [10]. We overcome this issue by actually minimizing the *expected minibatch superquantile* instead, defined as

$$\tilde{F}_{\alpha,m}(w) = \mathbb{E}_{(i_1,\cdots,i_m)\sim U_m}\left[\mathbb{S}_\alpha\big(F_{i_1}(w),\ldots,F_{i_m}(w)\big)\right]\,,$$

where U_m is the uniform distribution over all subsets of $\{1,\ldots,n\}$ of batch size m. This is a uniform close surrogate of the original objective [11, Prop. 1]

$$|F_\alpha(w) - \tilde{F}_{\alpha,m}(w)| \le \frac{3}{\sqrt{\alpha m}}\max_{i=1,\ldots,n}|F_i(w)|\,.$$

Using this expression, we design a federated optimization algorithm that steps of the usual federated averaging algorithm [16] with quantile estimation steps. Specifically, in each communication round, the local updates w_i^+ from the subsample of m selected clients $i \in S$ are aggregated to update the global model with the following two steps:

- estimate the quantile $\hat{Q}_\alpha \approx Q_\alpha(F_i(w) \,:\, i \in S)$ of the per-client losses to the server, and
- aggregate the updates from tail clients where $F_i(w) \geq \hat{Q}_\alpha$ to find the new global model w^+ as

$$w^+ = \frac{1}{|S_\alpha|} \sum_{i \in S_\alpha} w_i^+ \,, \quad \text{where} \quad S_\alpha = \{i \,:\, F_i(w) \geq \hat{Q}_\alpha\} \,.$$

The full algorithm is given in Appendix A. Similar to the standard FedAvg algorithm [16] for ERM objective (1), this aggregation rule enjoys a simplification in the case of a single local update per-client with a learning rate γ. Specifically, under the assumption of full client participation (i.e., $m = n$), if the local update $w - w_i^+ = \gamma \nabla F_i(w)$ is a single gradient step and $\hat{Q}_\alpha = Q_\alpha(F_1(w), \cdots, F_n(w))$ is the exact quantile of the per-client losses, the aggregated update is simply a subgradient step $w - w^+ = \gamma \nabla F_\alpha(w)$ where we denote the subgradient as $\nabla F_\alpha(w) \in \partial F_\alpha(w)$. Similar to FedAvg, our algorithm reduces the overall communication cost, which is often the bottleneck in bandwidth-constrained edge devices, while incurring a larger computation cost at each client.

3 Numerical Experiments

In this section, we demonstrate the effectiveness of Δ-FL in handling natural distribution shifts in federated learning.

Setup. We measure the 90^{th} percentile of the per-client misclassification errors, as a measure of the tail performance. We repeat all experiments 5 times and report the mean and standard deviation. We consider two learning tasks.

(a) *Character Recognition*: We use the EMNIST dataset [3], where the input x is a 28×28 grayscale image of a handwritten character and the output y is its label (0-9, a-z, A-Z). Each client is a writer of the character x. We train both a linear model and a LeNet-type convolutional network.
(b) *Sentiment Analysis*: We use the Sent140 dataset [5] where the input x is a tweet and the output $y = \pm 1$ is its sentiment. Each client is a distinct Twitter user. We train both a logistic regression and a Long-Short Term Memory neural network architecture (LSTM). The LSTM is built on the GloVe embeddings of the words of the tweet [8].

Baselines. We compare Δ-FL with the following baselines: We consider two methods which attempt to minimize the usual objective (1): FedAvg [16] and FedProx [14]. The latter augments FedAvg with a proximal term for more stable optimization. We also consider a few heterogeneity-aware objectives: Tilted-ERM [12], which is the analogue of Δ-FL but using the log-sum-exp function and AFL [17], whose objective is obtained as the limit $\lim_{\alpha \to 0} F_\alpha(w)$ of the Δ-FL objective. We also consider q-FFL [15], which raises the per-client loss F_i to the $(q+1)^{\text{th}}$ power, for some $q > 0$. We optimize q-FFL and Tilted-ERM with the federated optimization algorithms proposed in their respective papers. We use q-FFL with $q = 10$ in place of AFL, as it was found to have more stable convergence with similar performance.

Hyperparameters. We fix the number of clients per round to be $m = 100$ for each dataset-model pair except for Sent140-RNN, for which we use $m = 50$. We fix an iteration budget and tune a learning rate for FedAvg. The same iteration budget and learning rate schedule were used for *all* other methods including Δ-FL. All hyperparameters were tuned to find the best tail error (90^{th} percentile).

Results. The results are in Tables 1 and 2. We visualize in Figure 1 the distribution of test errors.

Δ-FL consistently achieves the smallest 90^{th} percentile error. Δ-FL achieves a 3.3% absolute (12% relative) improvement over any ERM objective on EMNIST-ConvNet. Among the heterogeneity aware objectives, Δ-FL achieves 1.8% improvement over the next best objective, which is Tilted-

Table 1: **90^{th} percentile** of the distribution of test misclassification errors (in %).

	EMNIST		Sent140	
	Linear	**ConvNet**	**Linear**	**RNN**
FedAvg	$49.66_{0.67}$	$28.46_{1.07}$	$46.83_{0.54}$	$49.67_{3.95}$
FedProx	$49.15_{0.74}$	$27.01_{1.86}$	$46.83_{0.54}$	$49.86_{4.07}$
q-FFL	$49.90_{0.58}$	$28.02_{0.80}$	$\mathbf{46.39}_{0.40}$	$48.66_{4.68}$
Tilted-ERM	$48.59_{0.62}$	$25.46_{1.49}$	$46.69_{0.49}$	$46.54_{3.27}$
AFL	$51.62_{0.28}$	$45.08_{1.00}$	$47.52_{0.32}$	$57.78_{1.19}$
Δ-FL, $\alpha = 0.8$	$49.10_{0.24}$	$26.23_{1.15}$	$46.44_{0.38}$	$\mathbf{46.46}_{4.39}$
Δ-FL, $\alpha = 0.5$	$\mathbf{48.44}_{0.38}$	$\mathbf{23.69}_{0.94}$	$46.64_{0.41}$	$50.48_{8.24}$
Δ-FL, $\alpha = 0.1$	$50.34_{0.95}$	$25.46_{2.77}$	$51.39_{1.07}$	$86.45_{10.95}$

Table 2: **Mean** of the distribution of test misclassification errors (in %).

	EMNIST		Sent140	
	Linear	**ConvNet**	**Linear**	**RNN**
FedAvg	$34.38_{0.38}$	$16.64_{0.50}$	$34.75_{0.31}$	$30.16_{0.44}$
FedProx	$\mathbf{33.82}_{0.30}$	$16.02_{0.54}$	$34.74_{0.31}$	$30.20_{0.48}$
q-FFL	$34.34_{0.33}$	$16.59_{0.30}$	$34.48_{0.06}$	$\mathbf{29.96}_{0.56}$
Tilted-ERM	$34.02_{0.30}$	$15.68_{0.38}$	$34.70_{0.31}$	$30.04_{0.25}$
AFL	$39.33_{0.27}$	$33.01_{0.37}$	$35.98_{0.08}$	$37.74_{0.65}$
Δ-FL, $\alpha = 0.8$	$34.49_{0.26}$	$16.09_{0.40}$	$\mathbf{34.41}_{0.22}$	$30.31_{0.33}$
Δ-FL, $\alpha = 0.5$	$35.02_{0.20}$	$\mathbf{15.49}_{0.30}$	$35.29_{0.25}$	$33.59_{2.44}$
Δ-FL, $\alpha = 0.1$	$38.33_{0.48}$	$16.37_{1.03}$	$37.79_{0.89}$	$51.98_{11.81}$

ERM. We note that q-FFL marginally outperforms Δ-FL on Sent140-Linear, but the difference 0.05% is much smaller than the standard deviation across runs.

Δ-FL is competitive at multiple values of α. For EMNIST-ConvNet, Δ-FL with $\alpha \in \{0.5, 0.8\}$ is better in 90^{th} percentile error than *all* other methods we compare to, and Δ-FL with $\alpha = 0.1$ is tied with Tilted-ERM, the next best method. We also empirically confirm that Δ-FL interpolates between FedAvg ($\alpha \to 1$) and AFL ($\alpha \to 0$).

Yet, Δ-FL is competitive in terms of average error. Perhaps surprisingly, Δ-FL gets the best test error performance on EMNIST-ConvNet and Sent140-Linear. This suggests that the average test distribution is shifted relative to the average training distribution p_α. In the other cases, we find that the reduction in mean error is small relative to the gains in the 90^{th} percentile error.

Minimizing superquantile loss over all clients performs better than minimizing worst error over all clients. Specifically, AFL which aims to minimize the worst error among all clients, as well as other objectives which approximate it (Δ-FL with $\alpha \to 0$, q-FFL with $q \to \infty$) tend to achieve poor performance. Δ-FL offers a more nuanced and more effective approach via the constraint set $\pi_i \leq 1/(n\alpha)$ than the straight pessimistic approach minimizing the worst error among all clients.

Δ-FL yields improved prediction on non-conforming clients. This can be observed from the histogram of Δ-FL in Figure 1, which exhibits thinner tails than FedAvg or Tilted-ERM. We see that the ERM objective of FedAvg sacrifices performance on the nonconforming clients. Tilted-ERM does improve over FedAvg in this regard, but Δ-FL has a thinner right tail than Tilted-ERM, showing better handling of heterogeneity.

Δ-FL yields improved prediction on data-poor clients. We observe in Figure 1 that Tilted-ERM and q-FFL mainly improve the performance on data-rich clients, that is clients with lots of data. On the other hand, Δ-FL gives a greater reduction in misclassification error on data-poor clients, that is clients with little data (< 200 examples per client).

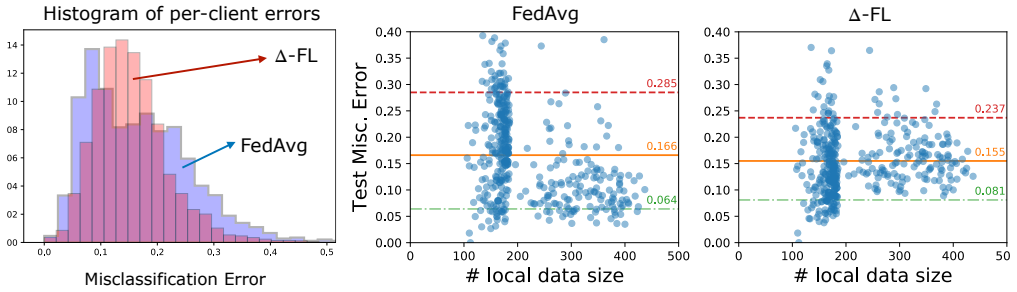

Figure 1: Per-client test misclassification error on EMNIST. **Left**: histogram of per-client errors. **Right two**: Scaltter plot of dataset size versus test error.

Acknowledgements

We acknowledge support from NSF DMS 2023166, DMS 1839371, CCF 2019844, the CIFAR program "Learning in Machines and Brains", faculty research awards, and a JP Morgan PhD fellowship. This work has been partially supported by MIAI – Grenoble Alpes, (ANR-19-P3IA-0003). This work was performed while Krishna Pillutla was at the University of Washington.

References

[1] Kallista A. Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards Federated Learning at Scale: System Design. In *Proceedings of Machine Learning and Systems 2019, MLSys 2019*, 2019.

[2] Theodora S. Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch. Paschalidis, and Wei Shi. Federated learning of predictive models from federated Electronic Health Records. *Int. J. Medical Informatics*, 112:59–67, 2018.

[3] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *arXiv Preprint*, 2017.

[4] George B Dantzig. Discrete-variable extremum problems. *Operations research*, 5(2):266–288, 1957.

[5] Alec Go, Richa Bhayani, and Lei Huang. Twitter Sentiment Classification using Distant Supervision. *CS224N Project Report, Stanford*, 2009.

[6] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated Learning for Mobile Keyboard Prediction. *arXiv Preprint*, 2018.

[7] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms I: Fundamentals*. Grundlehren der mathematischen Wissenschaften. 1996.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Li Huang, Andrew L Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu. Patient Clustering Improves Efficiency of Federated Machine Learning to Predict Mortality and Hospital stay time using Distributed Electronic Medical Records. *Journal of Biomedical Informatics*, 99, 2019.

[10] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021.

[11] Daniel Levy, Yair Carmon, John C. Duchi, and Aaron Sidford. Large-Scale Methods for Distributionally Robust Optimization. In *Advances in Neural Information Processing Systems*, 2020.

[12] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted Empirical Risk Minimization. In *International Conference on Learning Representations*, 2021.

[13] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[14] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated Optimization in Heterogeneous Networks. In *MLSys*. 2020.

[15] Tian Li, Maziar Sanjabi, and Virginia Smith. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations*, 2020.

[16] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, pages 1273–1282, 2017.

[17] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic Federated Learning. In *ICML*, 2019.

[18] Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier C. van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, Sudeep Agarwal, Julien Freudiger, Andrew Byde, Abhishek Bhowmick, Gaurav Kapoor, Si Beaumont, Áine Cahill, Dominic Hughes, Omid Javidbakht, Fei Dong, Rehan Rishi, and Stanley Hung. Federated Evaluation and Tuning for On-Device Personalization: System Design & Applications. *arXiv Preprint*, 2021.

[19] R Tyrrell Rockafellar, Stan Uryasev, and Michael Zabarankin. Risk tuning with generalized linear regression. *Mathematics of Operations Research*, 33(3):712–729, 2008.

[20] R Tyrrell Rockafellar and Stanislav Uryasev. Conditional Value-at-Risk for General Loss Distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

[21] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. 2009.

[22] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied Federated Learning: Improving Google Keyboard Query Suggestions. *arXiv Preprint*, 2018.

A Pseudocode

The pseudocode of the proposed optimization algorithm is given in Algorithm 1.

Algorithm 1 The Δ-FL Algorithm

Input: Initial iterate $w^{(0)}$, number of communication rounds T, number of clients per round m, number of local updates τ, local step size γ

for $t = 0, 1, \cdots, T-1$ **do**
  Sample m clients from $[n]$ without replacement in S
  Estimate the $(1-\alpha)$-quantile of $F_i(w^{(t)})$ for $i \in S$; call this $Q^{(t)}$
  for each selected client $i \in S$ in parallel **do**
    Set $\tilde{\pi}_i^{(t)} = \mathbb{I}\left(F_i(w^{(t)}) \geq Q^{(t)}\right)$
    Initialize $w_{k,0}^{(t)} = w^{(t)}$
    for $k = 0, \cdots, \tau - 1$ **do**
      $w_{i,k+1}^{(t)} = (1 - \gamma\lambda) w_{i,k}^{(t)} - \gamma \nabla F_i(w_{i,k}^{(t)})$
    end for
  end for
  $w^{(t+1)} = \sum_{i \in S} \tilde{\pi}_i^{(t)} w_{i,\tau}^{(t)} / \sum_{i \in S} \tilde{\pi}_i^{(t)}$
end for
return w_T

B Proofs

Proof of the Subgradient Expression (3). We first give a general expression for the subgradient. Define the notation

$$\mathcal{P}_\alpha = \left\{ \pi_i \in \mathbb{R}_n \,:\, 0 \le \pi_i \le \frac{1}{\alpha n} \,\forall\, i \in [n],\, \sum_{i=1}^n \pi_i = 1 \right\},$$

so that $\mathbb{S}_\alpha(u_1, \cdots, u_n) = \max_{\pi \in \mathcal{P}_\alpha} \pi^\top u$.

Proposition 1. *Fix a $w \in \mathbb{R}^d$ and let $\pi^\star \in \arg\max_{\pi \in \mathcal{P}_\alpha} \sum_{i=1}^n \pi_i F_i(w)$. Then, we have,*

$$\sum_{i=1}^n \pi_i^\star F_i(w) \in \partial F_\alpha(w)\,,$$

where $\partial F_\alpha(w)$ denotes the regular subdifferential of F_α.

Proof. Let $g_n(w) = (F_1(w), \cdots, F_n(w))$ denote the concatenation of the losses into a vector. Then, $F_\alpha(w) = \mathbb{S}_\alpha \circ g_n(w)$. Since $\mathbb{S}_\alpha$ is convex, we get that its (convex) subdifferential [e.g., 7, Cor. 4.4.4] is

$$\partial \mathbb{S}_\alpha(u) = \arg\max_{\pi \in \mathcal{P}_\alpha} \pi^\top u\,.$$

Since g_n is smooth and $\mathbb{S}_\alpha$ is convex with full domain, we obtain the regular subdifferential of $\mathbb{S}_\alpha \circ g_n$ by the chain rule [21, Thm. 10.6] as

$$\partial(\mathbb{S}_\alpha \circ g_n) = \nabla g_n(w) \partial \mathbb{S}_\alpha(u)\,,$$

where $\nabla g_n(w) \in \mathbb{R}^{d \times n}$ is the transpose of the Jacobian matrix of g_n. □

Let $Z(w)$ be a discrete random variable which takes the value $F_i(w)$ with probability $1/n$ for $i = 1, \ldots, n$, and let $Q_\alpha(Z(w))$ denote its $(1-\alpha)$-quantile. Consider the weights $\hat{\pi} \in \Delta^{n-1}$ given by a hard-thresholding based on whether $F_i(w)$ is larger than its $(1-\alpha)$-quantile:

$$\tilde{\pi}_i = \mathbb{I}\big(F_i(w) \ge Q_\alpha(Z(w))\big)\,, \quad \text{and,} \quad \hat{\pi}_i = \frac{\tilde{\pi}_i}{\sum_{i'=1}^n \tilde{\pi}_{i'}}. \tag{4}$$

The objective defined by these weights is $\hat{F}_\alpha(w) = \sum_{i=1}^n \hat{\pi}_i F_i(w)$. The next proposition shows that $\hat{F}_\alpha(w) = F_\alpha(w)$ when αn is an integer, or is a close approximation in general.

Proposition 2. *Assume $F_1(w) < \cdots < F_n(w)$ and let $i^\star = \lceil \alpha n \rceil$. Then, we have,*

(a) $\pi^\star = \arg\max_{\pi \in \mathcal{P}_\alpha} \sum_{i=1}^n \pi_i F_i(w)$ is unique,
(b) $Q_\alpha(Z(w)) = F_{i^\star}(w)$,
(c) if αn is an integer, then $\hat{\pi} = \pi^\star$ so that $\hat{F}_\alpha(w) = F_\alpha(w)$, and,
(d) if αn is not an integer, then

$$0 \le F_\alpha(w) - \hat{F}_\alpha(w) \le \frac{\max_{i=1,\ldots,n} |F_i(w)|}{\alpha n}\,.$$

Proof. We apply the property that the superquantile is a tail mean for discrete random variables [20, Proposition 8] to get

$$F_\alpha(w) = \frac{1}{\alpha n} \sum_{i=i^\star+1}^n F_i(w) + \left(1 - \frac{\lfloor \alpha n \rfloor}{\alpha n}\right) F_{i^\star}(w)\,.$$

Comparing with the definition $F_\alpha(w) = \sum_{i=1}^n \pi_1^\star F_i(w)$, this gives a closed-form expression for $\pi^\star$, which is unique because $F_{i^\star-1}(w) < F_{i^\star}(w) < F_{i^\star+1}(w)$. For (b), note that $Q_\alpha(Z(w)) = \inf\{\eta \in \mathbb{R} \,:\, \mathbb{P}(Z(w) > \eta) \le \alpha\}$ equals $F_{i^\star}(w)$ by definition of $i^\star$. Therefore, if αn is an integer, $\pi^\star$ coincides exactly with $\hat{\pi}$. When αn is not an integer, we have

$$\hat{F}_\alpha(w) = \frac{1}{n - i^\star + 1} \sum_{i=i^\star}^n F_i(w)\,.$$

The bound on $\hat{F}_\alpha(w) - F_\alpha(w)$ follows from elementary manipulations. □