Redundancy in cost functions for Byzantine fault-tolerant federated learning

Shuo Liu
sl1539@georgetown.edu
Georgetown University
Washington DC, USA

Nirupam Gupta
nirupam.gupta@epfl.ch
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Nitin H. Vaidya
nitin.vaidya@georgetown.edu
Georgetown University
Washington DC, USA

ABSTRACT

Federated learning has gained significant attention in recent years owing to the development of hardware and rapid growth in data collection. However, its ability to incorporate a large number of participating agents with various data sources makes federated learning susceptible to adversarial agents. This paper summarizes our recent results on server-based Byzantine fault-tolerant distributed optimization with applicability to resilience in federated learning. Specifically, we characterize *redundancies in agents' cost functions* that are necessary and sufficient for provable Byzantine resilience in distributed optimization. We discuss the implications of these results in the context of federated learning.

CCS CONCEPTS

• **Computing methodologies** → **Distributed algorithms**; **Machine learning**.

KEYWORDS

Federated learning, Fault-tolerance, Data redundancy

ACM Reference Format:
Shuo Liu, Nirupam Gupta, and Nitin H. Vaidya. 2021. Redundancy in cost functions for Byzantine fault-tolerant federated learning. In *Systems Challenges in Reliable and Secure Federated Learning (ResilientFL '21), October 25, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3477114.3488761

1 INTRODUCTION

Federated learning has gained significant attention in recent years due to rapid development in machine learning hardware and data collection. It is a distributed algorithm that can train large machine learning models with the help of a large number of participating agents. Unlike the traditional distributed machine learning models that assume homogeneous data sampling, in federated learning different agents may sample data from different distributions, and therefore, data sampling is generally heterogeneous [8]. This feature is critical to federated learning's wide applicability [3, 9, 10, 16, 17]. However, participation of a large diverse set of agents makes federated learning vulnerable to adversaries. This has motivated studies on the resilience of federated learning, including both privacy preservation [1, 2, 6, 15] and Byzantine fault-tolerance [7, 11, 18, 20, 21].

In this paper, we consider the problem of Byzantine fault-tolerance under a server-based architecture where there is a server which preserves and updates the parameters of a distributed optimization task, communicating with n agents where each agent i has a local cost function $Q_i(x)$. Suppose there are up to f Byzantine faulty agents (i.e., agents that can behave arbitrarily) in the system. An ideal goal of Byzantine fault-tolerant optimization problem would be to minimize the sum of cost functions of all non-faulty agents $\mathcal{H} \subsetneq \{1, ..., n\}$, i.e., to find a $x_{\mathcal{H}}$ such that [19]

$$x_{\mathcal{H}} \in \arg\min_x \sum_{i \in \mathcal{H}} Q_i(x). \tag{1}$$

The results of the above questions can be applied to federated learning, as it is a type of distributed optimization.

Under the above system architecture, we review our progress in Byzantine fault-tolerant distributed optimization utilizing existing *redundancy in cost functions*, and discuss the applicability of these results to Byzantine fault-tolerant federated learning.

2 REDUNDANCY FOR RESILIENCE

The aim of our research is to study the solvability of Byzantine fault-tolerant optimization problem. We first started with the ideal fault-tolerance goal, formally defined as f-resilience below [5]:

Definition 1 (f-resilience). *A deterministic algorithm is f-resilient, if it finds a minimizer $x_{\mathcal{H}}$ of all non-faulty agents with up to f Byzantine faulty agents in presence.*

That is, the output of a f-resilient algorithm finds solves Byzantine fault-tolerant optimization problems *exactly*. To achieve this goal, we proposed the notion of $2f$-redundancy property [5]:

Definition 2 ($2f$-redundancy). *For a given set of non-faulty agents $\mathcal{H}$, their non-faulty cost functions are said to satisfy $2f$-redundancy if for every subset $S \subseteq \mathcal{H}$ of size at least $n - 2f$,*

$$\arg\min_x \sum_{i \in S} Q_i(x) = \arg\min_x \sum_{i \in \mathcal{H}} Q_i(x). \tag{2}$$

Such a property can be utilized when solving Byzantine fault-tolerant distributed optimization problems. We showed the important role of $2f$-redundancy in Byzantine distributed optimization problems by the following theorem:

Theorem 1 ([5]). *Suppose all non-faulty cost functions are differentiable and convex. There exists a f-resilient deterministic algorithm, if and only if the cost functions of non-faulty agents satisfies the $2f$-redundancy.*

ResilientFL '21, October 25, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-8708-8/21/10…$15.00
https://doi.org/10.1145/3477114.3488761

That is to say, the solvability of a Byzantine fault-tolerant distributed optimization problem is associated with the redundancy property of cost functions. The sufficiency of $2f$-redundancy is shown in [5] by constructing a distributed gradient descent (DGD)-based algorithm with robust gradient aggregation rule named CGC.

$2f$-redundancy is achievable in many practical optimization problems, including distributed sensing and distributed learning (including federated learning) [5], but such a condition may still be too strong. To expand the applicability of our analysis, we considered the goal of approximate fault-tolerance, formally defined as (f, ϵ)-resilience [12]:

Definition 3 ((f, ϵ)-resilience). *A deterministic algorithm is said to be (f, ϵ)-resilient if the output $\widehat{x}$ of this algorithm satisfies*

$$\text{dist}\left(\widehat{x},\ \arg\min_x \textstyle\sum_{i\in\mathcal{H}} Q_i(x)\right) \leq \epsilon. \tag{3}$$

That is, the output of a (f, ϵ)-resilient algorithm is within ϵ-distance to the true minimum set. Correspondingly, we introduced a more generalized notion of $(2f, \epsilon)$-redundancy [12]:

Definition 4 ($(2f, \epsilon)$-redundancy). *A group of non-faulty cost functions is said to satisfy $(2f, \epsilon)$-redundancy, if for any pair of subsets $S, \widehat{S} \in [n]$, $|S| = n - f$, $\left|\widehat{S}\right| \geq n - 2f$, and $\widehat{S} \subsetneq S$,*

$$\text{dist}\left(\arg\min_x \textstyle\sum_{i\in S} Q_i(x),\ \arg\min_x \textstyle\sum_{i\in\widehat{S}} Q_i(x)\right) \leq \epsilon. \tag{4}$$

We were able to show the impact of $(2f, \epsilon)$-redundancy on the approximate results produced by an algorithm.

Theorem 2 (cf. [12]). *Suppose for any non-empty set of non-faulty agents S, the set $\arg\min_x \sum_{i\in S} Q_i(x)$ is non-empty and closed. We have*

1. *There exists a deterministic (f, ϵ)-resilient distributed optimization algorithm with $\epsilon \geq 0$, only if the agents' cost functions satisfy $(2f, \epsilon)$-redundancy, and*
2. *For a real value $\epsilon \geq 0$, if the agents' cost functions satisfy $(2f, \epsilon)$-redundancy, $(f, 2\epsilon)$-resilient is achievable.*

These two results indicate that even though the exact solution of Byzantine fault-tolerance problems relies on the existence of the strong $2f$-redundancy, a weaker property of $(2f, \epsilon)$-redundancy in cost functions can still imply approximate resilience. In practice, it is further shown in [12] that a synchronous DGD-based algorithm using robust gradient aggregation rules can achieve $(2f, O(\epsilon))$-resilience, where the actual value of $O(\epsilon)$ depends on the gradient aggregation rule [4, 22] in use, and also depends on f/n.

Comparing to $2f$-redundancy, $(2f, \epsilon)$-redundancy has a wider applicability; in fact, it provides a trade-off between resilience of an algorithm (f) and the accuracy one can expect from such an algorithm (ϵ), implicated by cost functions in question themselves and nothing else. $(2f, \epsilon)$-redundancy is a way of describing the cost functions, since for any group of non-faulty cost functions, once f is decided, an $\epsilon \geq 0$ always exists. Specially, $(2f, 0)$-redundancy is the same as $2f$-redundancy [12]. Similar to $2f$-redundancy, these results can also be applied to many scenarios including distributed sensing and distributed machine learning.

3 IMPLICATIONS ON RESILIENT FEDERATED LEARNING

The results in Section 2 can be generally applied to federated learning, a type of distributed optimization, specifically for those regarding solvability in Theorems 1 and 2. Consider a server-based federated learning system of n agents, among which up to f can be faulty. Instead of a cost function, each agent i has a data generation distribution $\mathcal{D}_i$ over a real vector space $\mathbb{R}^m$. The goal is to train a machine learning model Π, parameterized as a d-dimensional vector $x \in \mathbb{R}^d$. Each data point $z \in \mathbb{R}^m$ incurs a loss value $\ell(z; x)$; each agent i therefore can define a *expected local cost function*

$$Q_i(x) = \mathbb{E}_{z\sim\mathcal{D}_i}\ell(z; x). \tag{5}$$

To train a model Π is to minimize the sum of non-faulty expected cost functions. Thus, a federated learning task is of the same form as in (1).

In practice, DGD can sometimes be too expensive or impractical for federated learning, as the gradients $\nabla Q_i(w^t)$ for each t might be difficult to obtain. Instead, distributed stochastic gradient descent (D-SGD) is more commonly used: each agent i in each iteration t randomly samples a set z_i^t of s data points, and computes a stochastic gradient

$$g_i^t = \tfrac{1}{s}\textstyle\sum_{z\in z_i^t} \nabla\ell(z; x^t), \tag{6}$$

as it is a noisy estimator of the actual gradient $\nabla Q_i(w^t)$.

In [4] we analyzed a D-SGD-based algorithm in a very special case of $2f$-redundancy, where all agents has the same data generation distribution $\mathcal{D}$, i.e, the expected local cost function of every non-faulty agent is the same. We showed that the algorithm is expected to converge to the true minimizer within a margin of error. A similar analysis can also be applied to more general cases, i.e., when $2f$- or $(2f, \epsilon)$-redundancy is observed. We would like to point out that the case of $(2f, \epsilon)$-redundancy should be more suitable to the topic of federated learning, where each agent can have a different data generation distribution, while certain *useful* redundancy still exists – such redundancy is rooted in the learning problem itself.

Although we are still working on the theoretical analysis, some empirical experiments on benchmark machine learning datasets are provided in [13]. Those experiments also indicate the redundancies that we can utilize widely exist in real-world problems.

4 SUMMARY AND FUTURE WORKS

We reviewed our work in Byzantine fault-tolerant distributed optimization with redundancy in cost functions, and discussed their applicability to resilient federated learning problems. $2f$-redundancy and $(2f, \epsilon)$-redundancy can be used to guarantee or bound the accuracy of Byzantine optimization algorithms, including Byzantine federated learning algorithms.

The next task following this line of work is to analyse the convergence of stochastic algorithms with redundancy properties. Other questions related to resilient federated learning include asynchronous learning. We explored in [14] the utilization of redundancy in cost functions to solve asynchronous optimization problems, and the combination of resilience against Byzantine agents and stragglers. These works can also be further adapted to stochastic algorithms that are more suitable for federated learning tasks.

ACKNOWLEDGMENTS

Research reported in this paper was supported in part by the Army Research Laboratory under Cooperative Agreement W911NF- 17-2-0196, and by the National Science Foundation award 1842198. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, National Science Foundation or the U.S. Government. Research reported in this paper is also supported in part by a Fritz Fellowship from Georgetown University.

REFERENCES

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How To Backdoor Federated Learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Vol. 108. PMLR, 2938–2948.

[2] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing Federated Learning through an Adversarial Lens. In *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97. PMLR, 634–643.

[3] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1175–1191.

[4] Nirupam Gupta, Shuo Liu, and Nitin Vaidya. 2021. Byzantine Fault-Tolerant Distributed Machine Learning with Norm-Based Comparative Gradient Elimination. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 175–181.

[5] Nirupam Gupta and Nitin H. Vaidya. 2020. Fault-Tolerance in Distributed Optimization: The Case of Redundancy. In *Proceedings of the 39th Symposium on Principles of Distributed Computing* (Virtual Event, Italy) *(PODC '20)*. Association for Computing Machinery, New York, NY, USA, 365–374.

[6] Meng Hao, Hongwei Li, Xizhao Luo, Guowen Xu, Haomiao Yang, and Sen Liu. 2019. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics* 16, 10 (2019), 6532–6542.

[7] Jiawen Kang, Zehui Xiong, Dusit Niyato, Yuze Zou, Yang Zhang, and Mohsen Guizani. 2020. Reliable federated learning for mobile networks. *IEEE Wireless Communications* 27, 2 (2020), 72–80.

[8] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. 2015. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575* (2015).

[9] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. 2020. A review of applications in federated learning. *Computers & Industrial Engineering* 149 (2020), 106854.

[10] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[11] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* (2018).

[12] Shuo Liu, Nirupam Gupta, and Nitin H. Vaidya. 2021. Approximate Byzantine Fault-Tolerance in Distributed Optimization. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing* (Virtual Event, Italy) *(PODC'21)*. Association for Computing Machinery, New York, NY, USA, 379–389.

[13] Shuo Liu, Nirupam Gupta, and Nitin H Vaidya. 2021. Approximate Byzantine Fault-Tolerance in Distributed Optimization. *arXiv preprint arXiv:2101.09337* (2021).

[14] Shuo Liu, Nirupam Gupta, and Nitin H Vaidya. 2021. Asynchronous Distributed Optimization with Redundancy in Cost Functions. *arXiv preprint arXiv:2106.03998* (2021).

[15] Chuan Ma, Jun Li, Ming Ding, Howard H Yang, Feng Shu, Tony QS Quek, and H Vincent Poor. 2020. On safeguarding privacy and security in the framework of federated learning. *IEEE network* 34, 4 (2020), 242–248.

[16] Solmaz Niknam, Harpreet S Dhillon, and Jeffrey H Reed. 2020. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine* 58, 6 (2020), 46–51.

[17] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. 2020. The future of digital health with federated learning. *NPJ digital medicine* 3, 1 (2020), 1–7.

[18] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated multi-task learning. *arXiv preprint arXiv:1705.10467* (2017).

[19] Lili Su and Nitin H. Vaidya. 2016. Fault-Tolerant Multi-Agent Optimization: Optimal Iterative Distributed Algorithms. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing* (Chicago, Illinois, USA) *(PODC '16)*. Association for Computing Machinery, New York, NY, USA, 425–434.

[20] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221.

[21] Wentai Wu, Ligang He, Weiwei Lin, Rui Mao, Carsten Maple, and Stephen Jarvis. 2020. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Trans. Comput.* 70, 5 (2020), 655–668.

[22] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, 5650–5659.