An Introduction to Federated Computation

Akash Bharadwaj
Graham Cormode
Meta AI, USA, UK
{akashb,gcormode}@fb.com

ABSTRACT

Federated Computation is an emerging area that seeks to provide stronger privacy for user data, by performing large scale, distributed computations where the data remains in the hands of users. Only the necessary summary information is shared, and additional security and privacy tools can be employed to provide strong guarantees of secrecy. The most prominent application of federated computation is in training machine learning models (federated learning), but many additional applications are emerging, more broadly relevant to data management and querying data. This tutorial gives an overview of federated computation models and algorithms. It includes an introduction to security and privacy techniques and guarantees, and shows how they can be applied to solve a variety of distributed computations providing statistics and insights to distributed data. It also discusses the issues that arise when implementing systems to support federated computation, and open problems for future research.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; **Distributed computing methodologies**; • **Security and privacy**;

KEYWORDS

federated computation; federated analytics; federated learning; distributed computation; privacy

ACM Reference Format:
Akash Bharadwaj and Graham Cormode. 2022. An Introduction to Federated Computation. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3514221.3522561

1 MOTIVATION

There are increasing needs for privacy-enhancing technologies to be deployed at scale in order to provide users with useful products and services while protecting their private information. In particular, recent legislation (e.g., GDPR, CCPA) imposes a greater requirement on service providers to protect data that is entrusted to them. Meanwhile, new restrictions on what data is available (such as Apple's opt-in tracking and the planned deprecation of third-party cookies) limit the opportunity for data collection on which many data-intensive organizations have relied.

In response to these constraints, there has been a growth of interest in the notion of Federated Computation as a means for working with private data at very large scale. In this setting, federation refers to the large collection of heterogenous user devices that interact with an online service. The core concept of federated computation is to move the bulk of data processing to the edge of the network, so that user data is processed on user devices, and only a small amount of summary information is provided back to the server for further aggregation. This stands in contrast to the centralized approach of gathering all data in one place for analysis. It echoes the notion of "move the compute to the data, rather than vice-versa" that has informed large-scale distributed data processing such as MapReduce, but on a different set of scales: here, the (heterogenous) computational entities are typically quite weak (user devices such as phones or browsers), and the number of such entities can range from hundreds to millions. It embodies the notions of "data minimization" and "purpose limitation": that the information shared by the users is intended to be as little as possible, and only enough to support the intended application. In addition, more mathematical notions of privacy, such as differential privacy and (partially) homomorphic encryption, can be adopted in order to provide a strong guarantee of protection on the data that is shared to the server.

The most prominent notion of federated computation is Federated Learning: the process of training a machine learning model over data held by distributed clients [8]. Federated learning has been adopted by several large technology companies (most prominently, Google, but also Apple and Meta) in order to train their machine learning to a high level of accuracy while promising that the training data never leaves the client device. Typically, federated learning is based on a distributed notion of (stochastic) gradient descent: the server provides each user with a current model, and the user computes a gradient with which to update the model based on how their examples are classified by it. The server will then aggregate the returned gradients and propose a next iteration of the model.

While federated learning is the most prominent notion of federated computation, it is by no means the only one. Modern data processing relies on a wide range of different tasks informed by user data. These include gathering statistics, performing statistical tests, identifying trends and changes in data, and other complex calculations and analyses over distributed data. These too benefit from maintaining the data on user devices and only sharing the minimal amount of derived data to complete the computation. Techniques from privacy and security are additionally applied to protect the sensitive information. Algorithmic techniques, such as data summarization, dimensionality reduction and aggregation are used

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

ACM ISBN 978-1-4503-9249-5/22/06…$15.00
https://doi.org/10.1145/3514221.3522561

to enable complex functions to be evaluated in the distributed setting. If anything, these applications are even more important than federated learning: while many trained models are not ultimately deployed in production, statistics and insights are more ubiquitously employed in operations and decision making. Such analytics can be both reactive (helping to identify the root cause of changes or debug operations) and proactive (ongoing monitoring and adjusting to collective behavior, identifying new market segments to launch a new product). Current approaches are mostly bespoke: specific algorithms deployed to address a single query type. The long term prospect is to move more in the direction of high-level query languages, where target analytics can be defined descriptively, and compiled automatically into tasks to be performed on client devices, coupled with aggregate analysis centrally.

In this tutorial, we will give an overview of the state-of-the-art in federated computation, and provide particular emphasis on federated analytics (FA), broadly understood as the aspects of federated computation outside of federated learning (FL). Crudely, we can think of FL as techniques based around vector aggregation and gradient descent, and FA as techniques beyond those. We will introduce the models and techniques that have been used to build federated systems, and explain the algorithmic approaches to provide private collection of actionable analytics. Throughout, we will comment on open problems and directions for future work in this rapidly expanding area.

2 OUTLINE OF THE TUTORIAL

2.1 Introduction and Motivation

We begin with a motivation of federated computation as a growing area of study, with examples from existing deployments in practice. We explain the motivating factors, primarily around the need for privacy, and to scale with a large number of participating users. We outline the two concepts of federated learning and federated analytics, and highlight the key differences in terms of their focus. We also point out similarities and differences with related models of distributed computation, such as MapReduce [16] and sketching [6]. Then, we address social aspects of federated computation, such as ensuring diversity and fairness for participants.

There are several different models to capture different styles of federated computation. These include the distinction between horizontal and vertical federation, and the scale of the setting. Scales range from co-operating silos (e.g., a few hospitals), where each participant represents many individuals, to massively distributed scenarios (e.g., millions of mobile clients), where each participant represents a single individual. Our main focus in the tutorial is on the latter scenario.

We introduce the different notions of data privacy that are compatible with federated computation. These include privacy by data minimization, which may be additionally protected by secure aggregation [3]. Secure multi-party computation is a key tool to provide protection of data when shared between multiple parties [10]. Differential privacy (DP) is widely adopted to provide privacy, and can be achieved under different models – central, local or distributed/shuffle DP provide different trade-offs between trust and accuracy [7]. Differential privacy can be applied at different levels – per event, per device, or per user, to provided different protection at different levels of sensitivity to change.

2.2 Core Federated Computation Algorithms

We give examples of different algorithmic approaches to federated computations under different models of privacy, and with increasing complexity.

- We start with the foundational question of computing sums and counts of numeric values. If each user is content to reveal their value to an aggregator, it is straightforward to add up all responses from all users. Central differential privacy can be achieved by adding appropriate random noise to the result, with minimal impact on accuracy. For a stronger local differential privacy guarantee, each user can add their own noise independently, or via mechanisms such as randomized response. This removes the need to place trust in the aggregator, but increases the magnitude of the noise.
- Computing the sum (or average) of vectors, where each user submits a single vector, is a similarly fundamental task. Techniques from multiparty computation (MPC), such as secret sharing, are suitable to perform this sum while only revealing the aggregate sum and not any of the intermediate values. However, additional steps are needed to handle situations which arise in reality, such as when users may fail to complete the protocol (i.e., by going off-line midway through the data collection). Full MPC can also be challenging to operate at federated scale, due to reliance on public key infrastructure and full peer-to-peer connectivity. This leads to the notion of lightweight secure aggregation protocols that can handle dropouts and make minimal assumptions about infrastructure. These form the basis of FedAVG [11] and FEDSGD [14], two very popular approaches to training ML models under federated learning.
- Mean and variance estimation are two related statistical tasks that underpin many more sophisticated analytics. Several approaches have been proposed recently, which use randomization to extract a single bit from each user [2, 5, 17]. Reducing the communication to a single bit makes the privacy-preserving nature of the protocol more apparent, but requires a little more care to ensure that the result provides meaningful accuracy, particularly for variance and other non-linear aggregates.
- The problem of computing a histogram (set of counts) from a data set is one of the most heavily studied under the model of differential privacy. Recent efforts have aimed to translate these results to the distributed setting, where techniques based on sampling and sketching are used to handle high-dimensional data [1, 4, 18].
- For a discrete empirical data distribution, it is often useful to characterize it based on the heavy hitters (i.e., peaks in the PDF) and quantiles (i.e., an approximation of the CDF). Hierarchical approaches build on histogram computation and allow us to find heavy hitters and quantiles of data distributions, with tradeoffs between interactive and non-interactive approaches [18].

2.3 Federated Computation Systems

We go on to discuss the challenges that emerge in building systems to support federated computation at scale. In reality, user devices show extreme heterogeneity, which manifests in highly variable response times (straggglers) and limited computational capabilities (i.e., small space and limited processing power) [12]. A practical federated stack will need to take account of these variations, while ensuring that weaker devices are not excluded from participating. A useful implementation must support a wide range of use-cases, and so allow a range of tasks to be registered, while ensuring that no user is overwhelmed with demands. So far, open source toolkits for general purpose federated computation have been limited, but we will mention Tensor Flow Federated [15] for learning-like workloads, and Crypten for multi-party computation [9].

2.4 Advanced Topics and Open Problems

We conclude the tutorial with coverage of some more advanced computations, and point to areas ripe for further research. A key area for more study is the borderline between federated analytics and federated learning. Most FL work concentrates on the core tasks of training the parameters of a specified machine learning model, while FA research to date has mostly focused on gathering basic statistics. There is vast potential to expand the scope of the capabilities of federated computation in the direction of more general purpose computation. In particular, there are many other tasks relevant to deploying machine learning that require a federated solution, such as drawing statistics on features to inform feature scaling, feature selection and model architecture choice. It is important to also be able to compute properties of learned models, such as their accuracy and performance on different inputs – this too can require a federated solution to protect the user data used to test the model.

The bulk of the examples discussed so far apply to data that can be considered structured in a precise mathematical way, i.e., represented as scalars, vectors, matrices or tensors. It is open to understand different types of data which bring their own semantics, such as text or graph structured data. These in turn bring questions about privacy – what is the right notion of privacy to protect individuals that may be represented in text or graphs, where an individual may be represented in many different places? Related problems concern other data mining tasks such as clustering, for example in the FLoC effort to identify interest-based clusters of users in a distributed way [13].

Last, the approaches discussed so far tend to assume that the data is static. But in reality, data evolves over time, and federated approaches need to be defined that can handle time-series data, which arrives at the client device as a stream of updates, and allow streaming versions of federated computation.

3 INTENDED AUDIENCE AND BACKGROUND KNOWLEDGE

The tutorial is intended to be accessible to all participants at SIGMOD, and so makes minimal assumptions on prior knowledge. While some background on security and privacy would be useful, all the technical ideas can be understood without requiring formal security proofs or statistical notions of privacy. Instead, they are presented assuming properties of existing security and privacy tools, which are presented in more detail in the literature. To this end, the tutorial does not include any formal proofs, but instead provides an overview and intuition of the key concepts. As a result the tutorial is intended to be suitable for starting researchers, of for those with expertise in other areas seeking to understand the emerging topic of federated computation.We will make the tutorial accessible to all participants at SIGMOD, and so will minimize the assumptions on prior knowledge. While some background on security and privacy would be useful, we can explain all the technical ideas to a suitable level of detail without requiring formal security proofs or statistical notions of privacy. Instead, they can be presented assuming properties of existing security and privacy tools, which are presented in more detail in the literature. To this end, we will not present any formal proofs, but instead provide an overview and intuition of the key concepts. As a result, we intend that the tutorial will be suitable for starting researchers, or for those with expertise in other areas seeking to understand the emerging topic of federated computation.

The intended learning outcomes are:

- Understand the motivation for and benefits of federated computation;
- Compare the different models of security and privacy that federated computation can be performed in;
- Understand existing algorithms that operate in the federated model, and design new algorithms;
- Describe the challenges that arise in moving federated algorithms from theory to practice;
- Identify novel open problems and research directions for federated computation.

4 ABOUT THE PRESENTERS

The presenters are based at Meta AI, working on privacy-preserving machine learning. Their main focus is in designing tools and solutions for federated computation at scale as part of an applied research group. The presenting team has extensive experience of delivering technical material to research audiences, including several prior tutorials at VLDB, SIGMOD, and KDD.

Akash Bharadwaj is a technical lead (research) at Meta in the USA, working on applications of federated analytics and other privacy enhancing technologies. Previously, he was a senior research at Elemental Cognition, working on ML and NLP to build conversational AI. He was a research assistant Carnegie Mellon University, working on NLP and machine learning, particularly named entity recognition and machine translation.

Graham Cormode is a research scientist at Meta in the UK. Since 2013, he was a professor at the University of Warwick, UK; prior to this, he held research positions at AT&T Labs and Bell Labs in NJ, USA. He has worked extensively on privacy and data summarization, and has been recognized as a Fellow of the ACM for his work on data summarization and privacy.

REFERENCES

[1] E. Bagdasaryan, P. Kairouz, S. Mellem, A. Gascón, K. A. Bonawitz, D. Estrin, and M. Gruteser. Towards sparse federated analytics: Location heatmaps under distributed differential privacy with secure aggregation. *CoRR*, abs/2111.02356, 2021.

[2] R. B. Basat, M. Mitzenmacher, and S. Vargaftik. How to send a real number using a single bit (and some shared randomness). In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 25:1–25:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[3] 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*, CCS '17, page 1175–1191, 2017.

[4] D. Boneh, E. Boyle, H. Corrigan-Gibbs, N. Gilboa, and Y. Ishai. Lightweight techniques for private heavy hitters. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 762–776. IEEE, 2021.

[5] G. Cormode and I. L. Markov. Bit-efficient numerical aggregation and stronger privacy for trust in federated analytics. *CoRR*, abs/2108.01521, 2021.

[6] G. Cormode and K. Yi. *Small Summaries for Big Data*. CUP, 2020.

[7] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[8] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. A. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021.

[9] B. Knott, S. Venkataraman, A. Y. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten. Crypten: Secure multi-party computation meets machine learning. *CoRR*, abs/2109.00984, 2021.

[10] Y. Lindell. Secure multiparty computation. *Commun. ACM*, 64(1):86–96, Dec. 2020.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

[12] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. M. Esmaeili, and D. Huba. Federated learning with buffered asynchronous aggregation. *CoRR*, abs/2106.06639, 2021.

[13] D. Ravichandran and S. Vasilvitskii. Evaluation of cohort algorithms for the FLoC API. https://github.com/google/ads-privacy/blob/master/proposals/FLoC/FLOC-Whitepaper-Google.pdf, 2021.

[14] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 1310–1321. ACM, 2015.

[15] Tensorflow federated. https://www.tensorflow.org/federated, 2021.

[16] S. Vassilvitskii. Mapreduce algorithmics. In *Algorithms and Data Structures - 13th International Symposium, WADS*, volume 8037 of *Lecture Notes in Computer Science*, page 524. Springer, 2013.

[17] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu. Collecting and analyzing multidimensional data with local differential privacy. In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 638–649. IEEE, 2019.

[18] W. Zhu, P. Kairouz, B. McMahan, H. Sun, and W. Li. Federated heavy hitters discovery with differential privacy. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 3837–3847. PMLR, 2020.