The Power of Extrapolation in Federated Learning

Hanmin Li, Kirill Acharya, Peter Richtárik

King Abdullah University of Science and Technology

October 16, 2024

King Abdullah University
of Science and Technology

NEURAL INFORMATION
PROCESSING SYSTEMS

Introduction

- Federated learning (FL) is a distributed training approach for machine learning models, where multiple clients collaborate under the guidance of a central server to optimize a loss function [1, 3].
- In this paper, We consider the following federated optimization problem,

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}, \tag{1}$$

where each $f_i : \mathbb{R}^d \mapsto \mathbb{R}$ is a differentiable function, n is the number of clients.

Introduction

- The most commonly used algorithm to address this problem is the federated average (`FedAvg`) [2, 3] algorithm. However, it suffers from client drift when the data is heterogeneous.
- In an attempt to tackle with this, `FedProx` was introduced li2020federated, which can be formulated as

$$x_{k+1} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{prox}_{\gamma f_i} (x_k) . \tag{FedProx}$$

- Compared with gradient based algorithms, proximal algorithms are more stable.

Introduction

- Proximal operators of a convex function can be viewed as projection to a certain level set of the function.
- It is known that the parallel projection methods for solving the convex feasibility problem is accelerated by a practice called extrapolation.

$$x_{k+1} = x_k + \alpha_k \left(\frac{1}{n} \sum_{i=1}^{n} \Pi_{\mathcal{X}_i}(x_k) - x_k \right). \tag{2}$$

  Here $\alpha_k > 1$ is the extrapolation parameter, the intersection of convex sets $\mathcal{X}_i$ is assumed to be non-empty.
- This means that we move further along the line connecting the current iterate x_k and the average projection point $\frac{1}{n} \sum_{i=1}^{n} \Pi_{\mathcal{X}_i}(x_k)$.

Motivation

- In this paper, we assume that the proximal operators are solved exactly with no inaccuracies.
- Given the similarity between the proximal operator and the projection operator, we propose to use extrapolation with `FedProx`.

Assumptions

- (Interpolation) There exists $x_\star \in \mathbb{R}^d$ such that $\nabla f_i(x_\star) = 0$ for all $i \in [n]$.
- (Individual convexity) The function $f_i : \mathbb{R}^d \mapsto \mathbb{R}$ satisfies $0 \leq f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y \rangle$, for all $x, y \in \mathbb{R}^d$.
- (Smoothness) The function $f_i : \mathbb{R}^d \mapsto \mathbb{R}$ satisfies $f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y \rangle \leq \frac{L_i}{2} \|x - y\|^2$, for all $x, y \in \mathbb{R}^d$.

The interpolation assumption comes from the non-emptiness of convex feasibility.

Constant extrapolation

- For $\operatorname{prox}_{\gamma f_i}(x_k)$, we know that the following identity holds,

$$\nabla M_{f_i}^{\gamma}(x_k) = \frac{1}{\gamma}\left(x_k - \operatorname{prox}_{\gamma f_i}(x_k)\right),$$

where $M_{f_i}^{\gamma}(x)$ is the Moreau envelope of f_i. This allows us to formulate the algorithm as

$$x_{k+1} = x_k - \alpha_k \gamma \cdot \frac{1}{n}\sum_{i=1}^{n} \nabla M_{f_i}^{\gamma}(x_k),$$

which is running `SGD` towards $M^{\gamma}(x) := \frac{1}{n}\sum_{i=1}^{n} M_{f_i}^{\gamma}(x)$.

- The interpolation assumption guarantees that minimizers of f and M^{γ} coincide.

Constant extrapolation

Assume Assumption 1, 2 and 3 holds, a fixed $\alpha_k = \alpha \in (0, 2/\gamma L_{\gamma,\tau})$, minibatch of size τ, local stepsize γ, we have

$$\mathbb{E}\left[f(x_K)\right] - \inf f \le C(\gamma, \tau, \alpha) \cdot \frac{\|x_0 - x_\star\|^2}{K}, \tag{3}$$

where

$$C(\gamma, \tau, \alpha) := \frac{1 + \gamma L_{\max}}{\alpha\gamma\left(2 - \alpha\gamma L_{\gamma,\tau}\right)}$$
$$L_{\gamma,\tau} := \frac{n-\tau}{\tau(n-1)} \frac{L_{\max}}{1 + \gamma L_{\max}} + \frac{n(\tau-1)}{\tau(n-1)} L_\gamma.$$

Here L_γ is the smoothness constant of $M^\gamma(x)$.

Remarks

- The optimal constant extrapolation parameter is $1/\gamma L_{\gamma,\tau} > 1$, resulting in $C(\gamma, \tau, \alpha_{\gamma,\tau}) = L_{\gamma,\tau}(1 + \gamma L_{\max}) \leq L_{\max}$, which indicates convergence.
- If we assume in addition that f is μ-strongly convex, we obtain linear convergence.
- The convergence rate of `FedProx` is given by $C(\gamma, \tau, 1)$, and we have

$$\frac{C(\gamma, \tau, 1)}{C(\gamma, \tau, \alpha_{\gamma,\tau})} \geq 2 + \gamma L_{\max} + \frac{1}{\gamma L_{\max}},$$

indicating the superiority of our algorithm compared to `FedProx`.

Adaptive extrapolation

Since the extrapolation parameter α_k is naturally connected to stepsize of SGD, we can use adaptive rules to determine it.

$$\alpha_{k,G} := \frac{\frac{1}{n}\sum_{i=1}^{n}\left\|x_k - \mathrm{prox}_{\gamma f_i}(x_k)\right\|^2}{\left\|\frac{1}{n}\sum_{i=1}^{n}\left(x_k - \mathrm{prox}_{\gamma f_i}(x_k)\right)\right\|^2} \geq 1. \tag{GraDS}$$

$$\alpha_{k,S} := \frac{\frac{1}{n}\sum_{i=1}^{n}\left(M_{f_i}^{\gamma}(x_k) - \inf M_{f_i}^{\gamma}\right)}{\gamma\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla M_{f_i}^{\gamma}(x_k)\right\|^2} \geq \frac{1}{2\gamma L_\gamma}. \tag{StoPS}$$

Adaptive extrapolation

Assume assumption 1, 2 and 3 holds, if we are using $\alpha_k = \alpha_{k,G}$ in the full batch case, we have

$$\mathbb{E}\left[f(\bar{x}_K)\right] - \inf f \leq \frac{1+\gamma L_{\max}}{2+\gamma L_{\max}} \cdot \left(\frac{1}{\gamma} + L_{\max}\right) \cdot \frac{\left\|x_0 - x_\star\right\|^2}{\sum_{k=0}^{K-1} \alpha_{k,G}}, \quad (4)$$

where $\bar{x}_K$ is chosen randomly from the first K iterates $\{x_0, x_1, ..., x_{K-1}\}$ with probabilities $p_k = {}^{\alpha_{k,G}}/_{\sum_{k=0}^{K-1} \alpha_{k,G}}$. Similarly, if we are using $\alpha_k = \alpha_{k,S}$, we have

$$\mathbb{E}\left[f(\bar{x}_K)\right] - \inf f \leq \left(\frac{1}{\gamma} + L_{\max}\right) \cdot \frac{\left\|x_0 - x_\star\right\|^2}{\sum_{k=0}^{K-1} \alpha_{k,S}}, \quad (5)$$

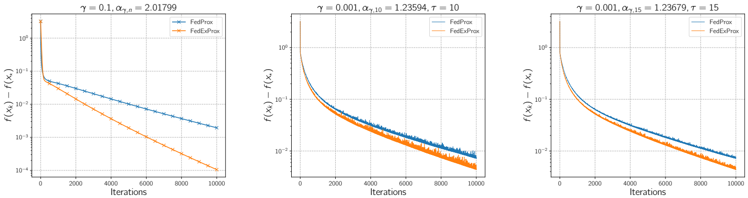
where $\bar{x}_K$ is chosen randomly from the first K iterates $\{x_0, x_1, ..., x_{K-1}\}$ with probabilities $p_k = {}^{\alpha_{k,S}}/_{\sum_{k=0}^{K-1} \alpha_{k,S}}$.

Remarks

- We can extend the theorem into stochastic setting, using a stochastic version of the two adaptive stepsizes.
- Both `FedExProx-GraDS` and `FedExProx-StoPS` exhibits "semi-adaptivit". A small γ hinders convergence, however, setting it to $\frac{1}{L_{\max}}$ limits the worsening of the convergence to a factor of 2.

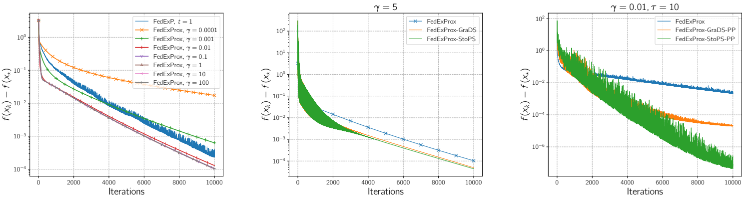
Experiments

- Comparison of `FedProx` and `FedExProx` in the full batch or minibatch setting.



Experiments

- Comparison of `FedExP`, `FedExProx`, `FedExProx-GraDS` and `FedExProx-StoPS` in terms of iteration complexity in the full batch or minibatch setting.



Bibiliography I

Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon.
Federated learning: Strategies for improving communication efficiency.
arXiv preprint arXiv:1610.05492, 8, 2016.

Olvi L Mangasarian and Mikhail V Solodov.
Backpropagation convergence via deterministic nonmonotone perturbed minimization.
Advances in Neural Information Processing Systems, 6, 1993.

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