Proximal and Federated Random Reshuffling

Konstantin Mishchenko, Ahmed Khaled, Peter Richtárik

Collaborators

Konstantin Mishchenko

Postdoctoral Researcher

Inria Sierra

Peter Richtárik

Professor, KAUST

Problem definition

Number of data points

$$\min_{x \in \mathbb{R}^d} \left[\frac{1}{n} \sum_{i=1}^{n} f_i(x) + \psi(x) \right]$$

Model

Regularizer

Proximal mapping

$$\min_{\boldsymbol{x} \in \mathbb{R}^{d \cdot M}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}) + \psi(\boldsymbol{x})$$

Proximal mapping

$$\min_{\boldsymbol{x} \in \mathbb{R}^{d \cdot M}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}) + \psi(\boldsymbol{x})$$

$$\operatorname{prox}_{\gamma\psi}(x) \stackrel{\text{def}}{=} \arg\min_{z} \left\{ \gamma\psi(z) + \frac{1}{2}\|z - x\|^2 \right\}$$

Proximal mapping

$$\min_{\boldsymbol{x} \in \mathbb{R}^{d \cdot M}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}) + \psi(\boldsymbol{x})$$

$$\mathrm{prox}_{\gamma\psi}(x) \stackrel{\mathrm{def}}{=} \arg\min_{z} \left\{ \gamma\psi(z) + \frac{1}{2}\|z - x\|^2 \right\}$$

We assume the proximal operator is **expensive**

Examples of expensive proximal operators

- **Communication in federated and distributed optimization (via the consensus regularizer)**

- **Imaging inverse problems**

Proximal SGD

Algorithm 2 Proximal SGD

Require: Stepsizes $\gamma_k > 0$, initial vector $x_0 \in \mathbb{R}^d$, number of steps K

1: **for** steps $k = 0, 1, \ldots, K-1$ **do**

2: Sample i_k uniformly at random from $[n]$

3: $x_{k+1} = \text{prox}_{\gamma_k \psi}(x_k - \gamma_k \nabla f_{i_k}(x_k))$

Prox evaluation every iteration
(too expensive!)

Proximal GD

Algorithm 3 Proximal GD

Input: Stepsizes $\gamma_k > 0$, initial vector $x_0 \in \mathbb{R}^d$, number of steps K
for steps $k = 0, 1, \ldots, K-1$ **do**
    $x_{k+1} = \mathrm{prox}_{\gamma_k \psi}(x_k - \gamma_k \nabla f(x_k))$
end for

Full gradient evaluation every iteration (too expensive!)

Prox-RR

Algorithm 1 Proximal Random Reshuffling (ProxRR)

Input: Stepsizes $\gamma_t > 0$, initial vector $x_0 \in \mathbb{R}^d$, number of epochs T
for epochs $t = 0, 1, \ldots, T-1$ **do**
  Sample a permutation $\pi = (\pi_0, \pi_1, \ldots, \pi_{n-1})$ of $[n]$
  $x_t^0 = x_t$
  for $i = 0, 1, \ldots, n-1$ **do**
    $x_t^{i+1} = x_t^i - \gamma_t \nabla f_{\pi_i}(x_t^i)$
  end for
  $x_{t+1} = \mathrm{prox}_{\gamma_t n \psi}(x_t^n)$
end for

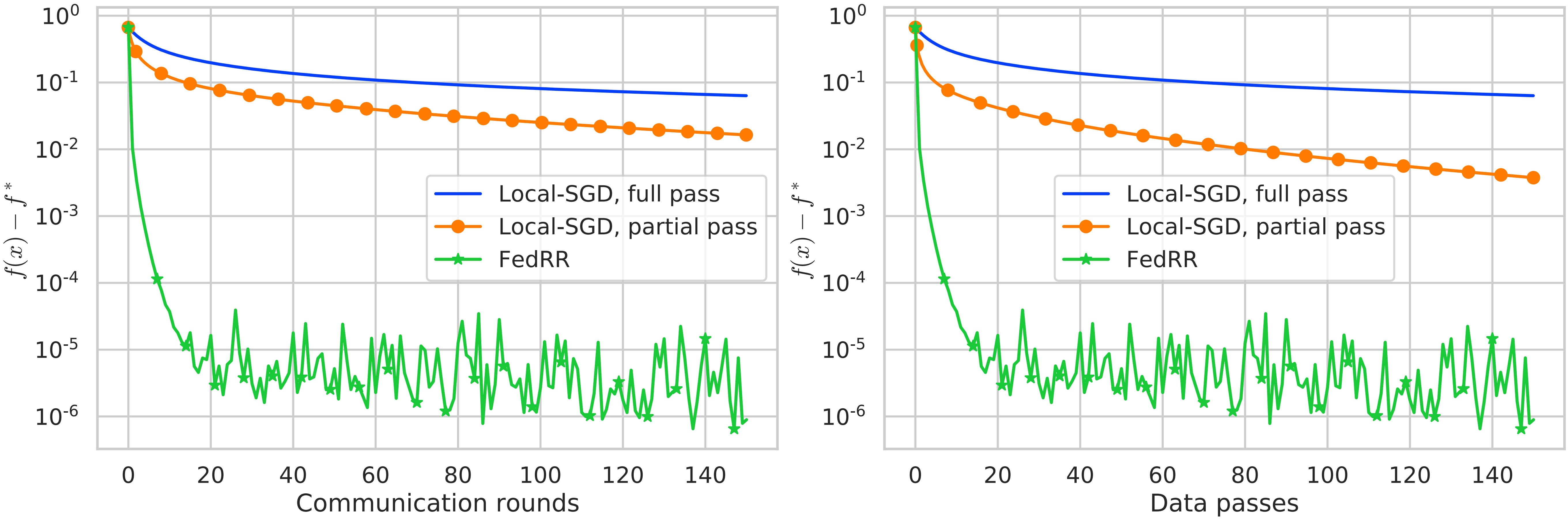
Stochastic gradients used during epoch

Proximal only once per epoch

ProxRR: Theory Results

- **Proximal complexity (# of proximal step calls)** is significantly faster than SGD for strongly convex objectives
- Compares favorably to GD and SVRG in some settings
- Nonconvex case also covered!

Experiments: federated learning (logistic regression w/ l2 regularization)



More things in the paper

1. **Importance sampling**
2. **Decreasing stepsizes**
3. **Details for federated learning**
4. **Bounds for iid and non-iid data**