High-order Tensor Method for Distributed Convex Optimization

Kamzolov Dmitry
Moscow Institute of Physics and Technology

Optimization without borders (Sirius, Russia)

https://arxiv.org/pdf/2102.08246
Joint work with: P. Dvurechensky, A. Gasnikov, A. Lukashevich
S. Lee, E. Ordentlich, C. Uribe
Supported by RFBR projects no. 19-31-27001

July 17, 2021

Plan

- High-order methods
- Experiments
- Statistically Preconditioned Distributed Method

Problem formulation

High-order convex problem

$$\min f(x),$$

$f(x)$ is convex function with Lipschitz p-th derivative with constant L_p

Problem formulation

Lipschitz derivative

$$\|D^p f(x) - D^p f(y)\| \leq L_p \|x - y\|$$

Taylor approximation

$$\Omega_p(f, x; y) = f(x) + \sum_{k=1}^{p} \frac{1}{k!} D^k f(x) \left[y - x\right]^k, y \in E$$

Corrolary

$$|f(y) - \Omega_p(f, x; y)| \leq \frac{L_p}{(p+1)!} \|y - x\|^{p+1}$$

Basic Method

Basic step [Nesterov, 2018]

$$T_{H_p}(x) = \operatorname*{argmin}_y \left\{ \tilde{\Omega}_{p,H_p}(f, x; y) \right\},$$

where

$$\tilde{\Omega}_{p,H_p}(f, x; y) = \Omega_p(f, x; y) + \frac{H_p}{p!} \|y - x\|^{p+1}.$$

For $H_p \geq L_p$ this subproblem is convex and hence implementable.

Basic Method

Basic step [Nesterov, 2018]

$$T_{H_p}(x) = \operatorname*{argmin}_y \left\{ \tilde{\Omega}_{p,H_p}(f, x; y) \right\},$$

where

$$\tilde{\Omega}_{p,H_p}(f, x; y) = \Omega_p(f, x; y) + \frac{H_p}{p!} \|y - x\|^{p+1}.$$

For $H_p \geq L_p$ this subproblem is convex and hence implementable.

Convergence

$$f(x_N) - f(x_*) \leq O\left(\frac{H_p R^{p+1}}{N^p} \right)$$

Superfast Second-Order Method
Nesterov 2018/ Nesterov 2020

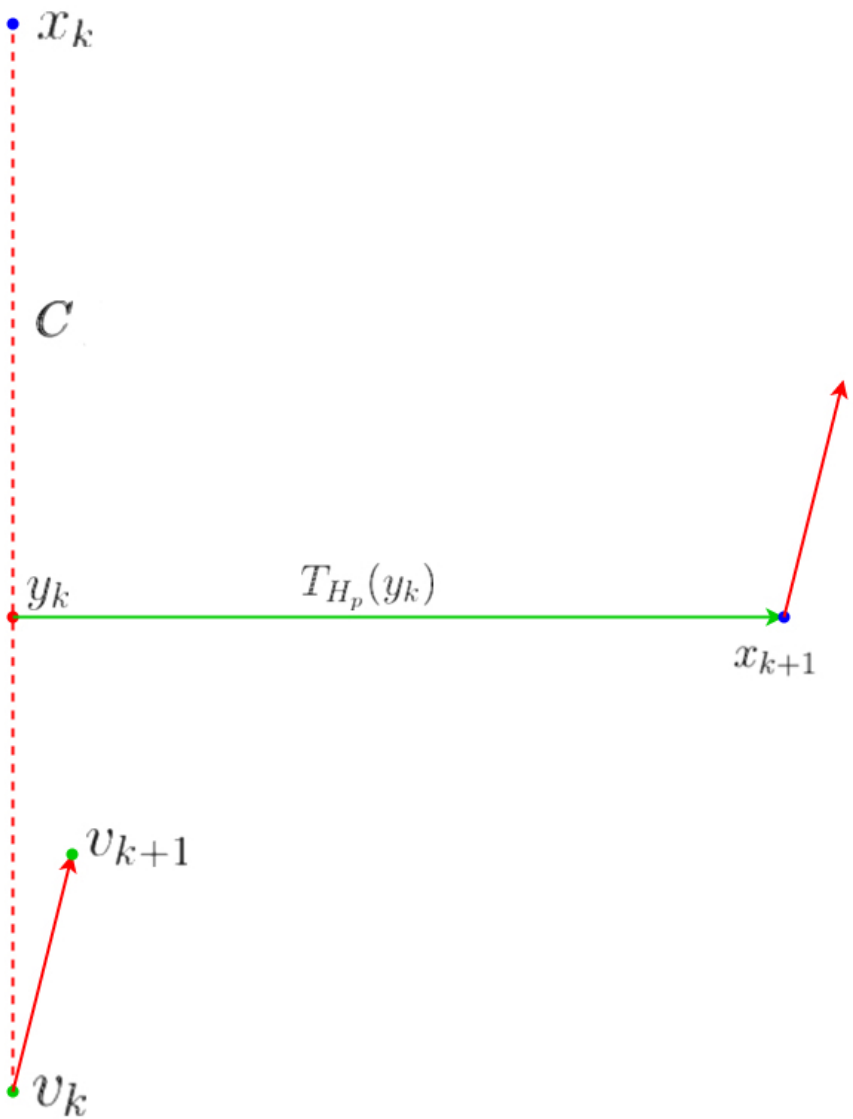
1: Choose $x_0 \in E$ and define $A_k = O(k^{p+1})$
2: Define $\psi_0(x) = \frac{1}{p+1}\|x - x_0\|^{p+1}$
3: **for** $k = 1$ **to** $k = N$ **do**
4: Compute $v_k = \underset{x \in E}{\operatorname{argmin}} \psi_k(x)$ and $y_k = \frac{A_k}{A_{k+1}} x_k + \frac{a_{k+1}}{A_{k+1}} v_k$, where

$$a_{k+1} = A_{k+1} - A_k$$

5: Compute $x_{k+1} = T_{H_p}(y_k)$ and update

$$\psi_{k+1} = \psi_k(x) + a_{k+1}[f(x_{k+1}) + \langle \nabla f(x_{k+1}), x - x_{k+1} \rangle]$$

Superfast Second-Order Method
Nesterov 2018/ Nesterov 2020



Superfast Second-Order Method
Nesterov 2018/ Nesterov 2020

Convergence rate

$$f(x_N) - f(x_*) \leq O\left(\frac{H_p R^{p+1}}{N^{p+1}}\right)$$

Hyperfast Second-Order Method
Gasnikov, Bubeck et.al. 2019/ Kamzolov, Gasnikov 2020

1: Define $A_0 = 0, x_0 = y_0 = 0$
2: **for** $k = 0$ **to** $k = N - 1$ **do**
3: Compute a pair $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ such that

$$\frac{1}{2} \leq \lambda_{k+1} \frac{L_p \cdot \|x_{k+1} - y_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1},$$

where

$$x_{k+1} = T_{L_p}(y_k)$$

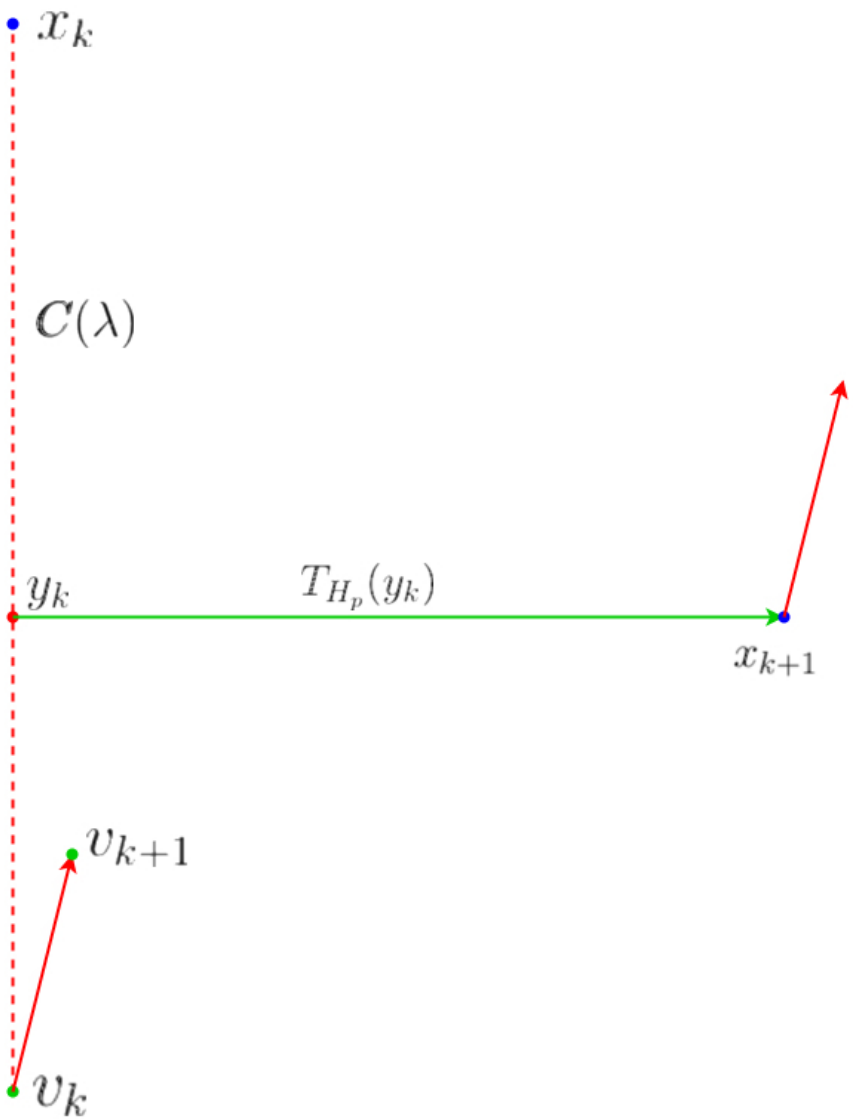
and

$$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}, y_k = \frac{A_k}{A_{k+1}} x_k + \frac{a_{k+1}}{A_{k+1}} v_k,$$

$$A_{k+1} = A_k + a_{k+1}$$

4: Update $v_{k+1} = v_k - a_{k+1} \nabla f(x_{k+1})$
5: **return** x_N

Hyperfast Second-Order Method
Gasnikov, Bubeck et.al. 2019/ Kamzolov, Gasnikov 2020



Hyperfast Second-Order Method
Gasnikov, Bubeck et.al. 2019/ Kamzolov, Gasnikov 2020

Convergence rate

$$f(x_N) - f(x_*) \leq \tilde{O}\left(\frac{H_p R^{p+1}}{N^{\frac{3p+1}{2}}}\right)$$

where $\tilde{O}(\cdot)$ means accuracy up to a logarithmic factor

Theoretical Line-search complexity

$$k \leq 30p \log p + \log\left(\frac{H_p \|x^*\|^{p+1}}{\varepsilon}\right)$$

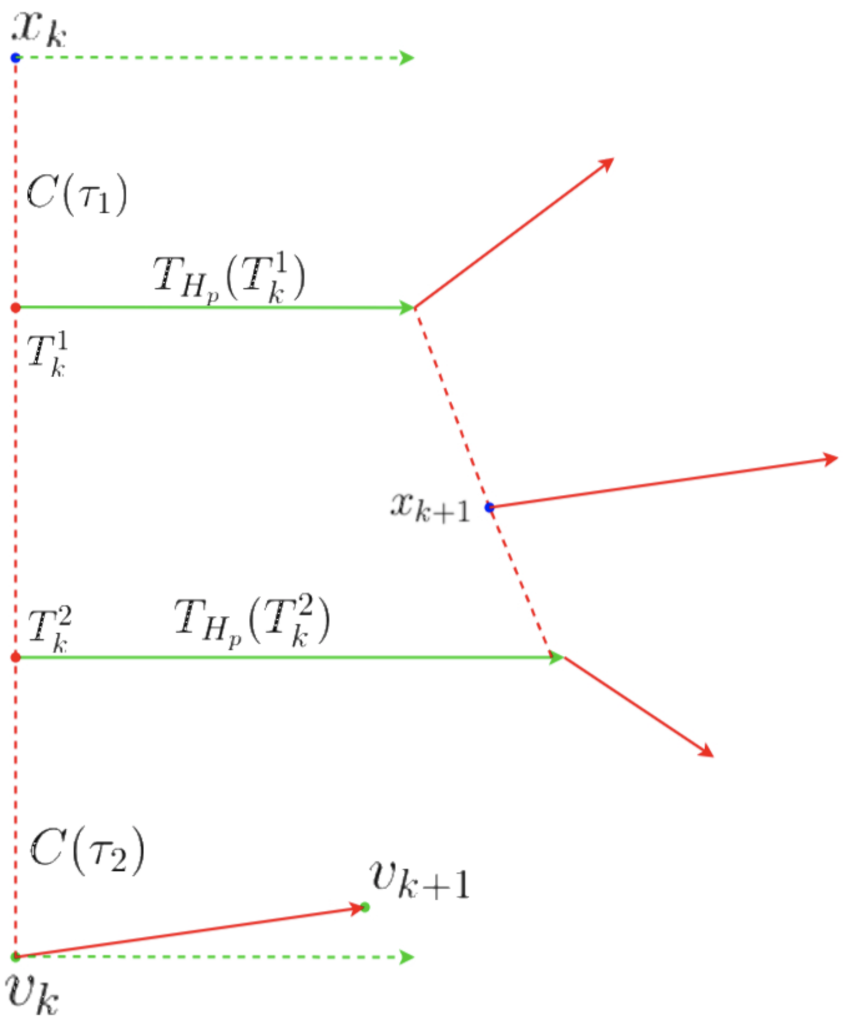
Proximal-Point Method With Segment Search Nesterov 2020

1: Set $v_0 = x_0 \in \mathbb{E}$, $H_p > 0$ и $A_0 = 0$
2: **for** $k = 0$ **to** $k = N - 1$ **do**
3: Compute $x_k^0 \in T_{H_p}(x_k)$. **If** $\langle \nabla f(x_k^0), u_k \rangle \geq 0$, **then** $g_k = \nabla f(x_k^0)$,
4: **Else**, $x_k^1 \in T_{H_p}(v_k)$. **If** $\langle \nabla f(x_k^1), u_k \rangle \leq 0$, **then** $g_k = \nabla f(x_k^1)$
5: **Else**, find $0 \leq \tau_k^1 \leq \tau_k^2 \leq 1$, $T_k^1 \in T_{H_p}(x_k + \tau_k^1 u_k)$ и $T_k^2 \in T_{H_p}(x_k + \tau_k^2 u_k)$ such that:

$$\textbf{a)} \alpha_k \leq 0 \leq \beta_k \quad \textbf{b)} \gamma_k \alpha_k (\tau_k^1 - \tau_k^2) \leq h(T_k^2),$$

where $\alpha_k = \langle \nabla f(T_k^1), u_k \rangle$, $\beta_k = \langle \nabla f(T_k^2), u_k \rangle$, $\gamma_k = \frac{\beta_k}{\beta_k - \alpha_k}$,
$h(x)$ - special function for step-size computation.
Compute $g_k = \gamma_k \nabla f(T_k^1) + (1 - \gamma_k) \nabla f(T_k^2)$ и
$x_{k+1} = \gamma_k T_k^1 + (1 - \gamma_k) T_k^2$.
6: Compute $a_{k+1} > 0$ from $\frac{a_{k+1}^2}{A_k + a_{k+1}} = h(x_{k+1})$ and $A_{k+1} = A_k + a_{k+1}$
7: Compute $v_{k+1} = v_k - a_{k+1} g_k$ and $u_{k+1} = v_{k+1} - x_{k+1}$.

Proximal-Point Method With Segment Search
Nesterov 2020



Proximal-Point Method With Segment Search

Convergence rate

$$f(x_N) - f(x_*) \leq \tilde{O}\left(\frac{H_p R^{p+1}}{N^{\frac{3p+1}{2}}}\right)$$

Theoretical Line-search complexity

$$k \leq 2 + \frac{1}{p} \log\left(\frac{3H_p \|x^*\|^{p+1}}{2\varepsilon}\right)$$

Experiments

MNIST Logistic regression

Problem

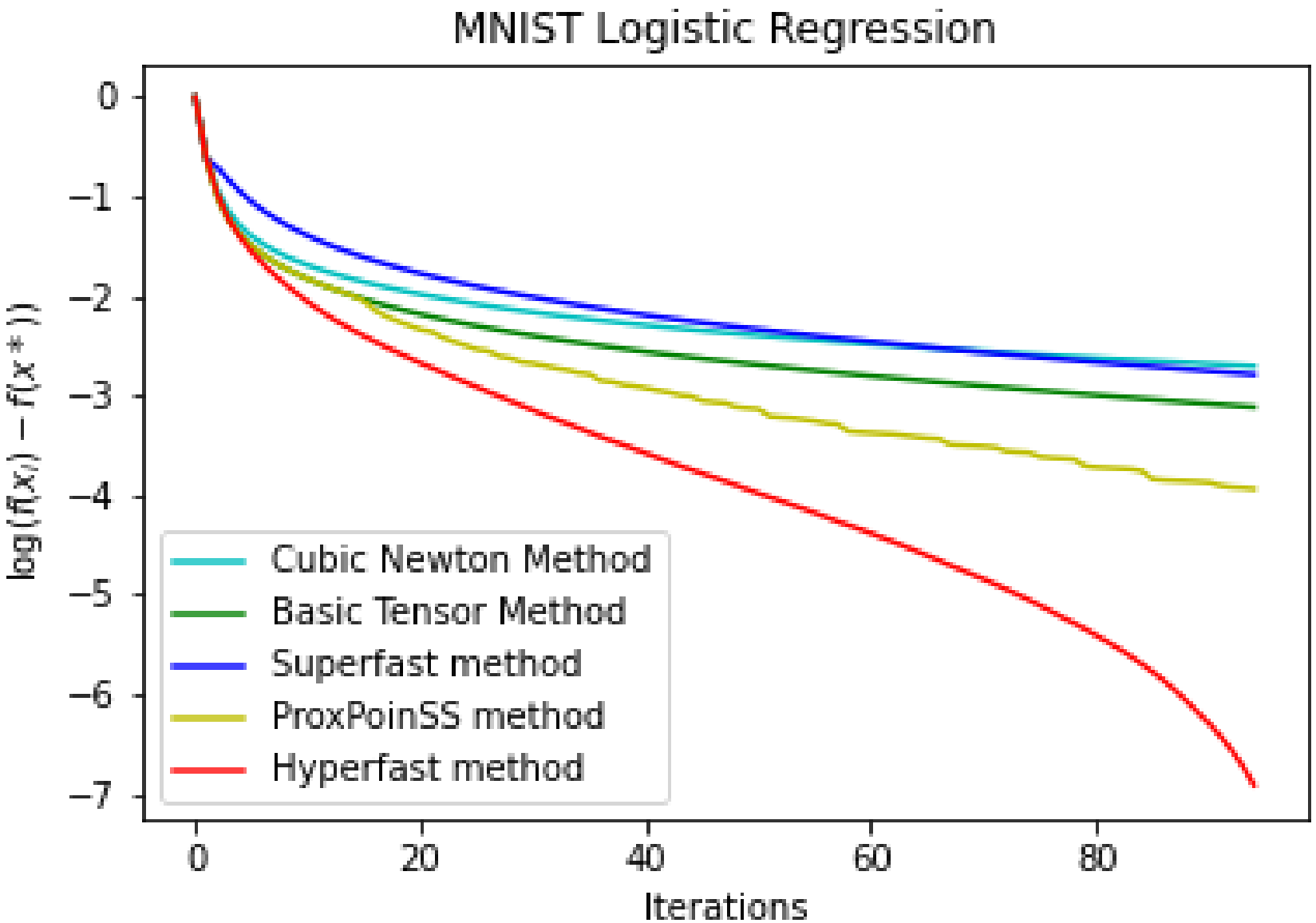
$$f(x) = \frac{1}{M}\sum_{i=1}^{M}\ell(x,\zeta_i) = \frac{1}{M}\sum_{i=1}^{M}\log\left(1+\exp(-\eta_i x^\top \xi_i)\right),$$

where $M = 5000$, $d = 784$ for MNIST

Implementation Time

Method	Time	Time/iteration
Cubic Newton Method	1073 сек.	10.73 сек./итер.
Basic Tensor Method	1079 сек.	10.79 сек./итер.
Superfast Method	1102 сек.	11.02 сек./итер.
ProxPointSS Method	1746 сек.	17.46 сек./итер.
Hyperfast Method	1579 сек.	15.79 сек./итер.

Experiments



Experiments

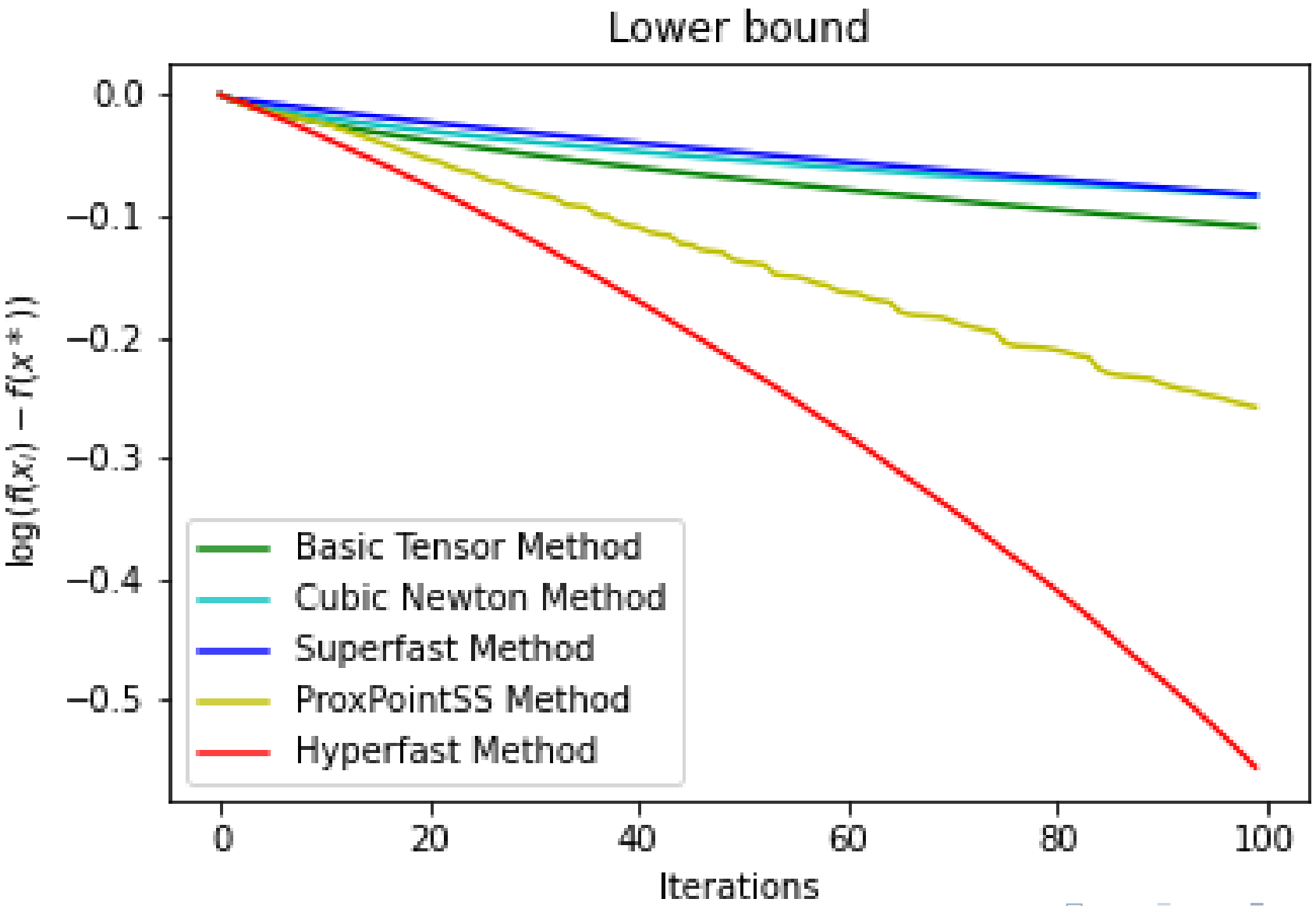
Lower bound Nesterov function

$$f_4(x) = \frac{1}{4}\sum_{i=1}^{M}|x_i - x_{i+1}|^4 - x_1$$

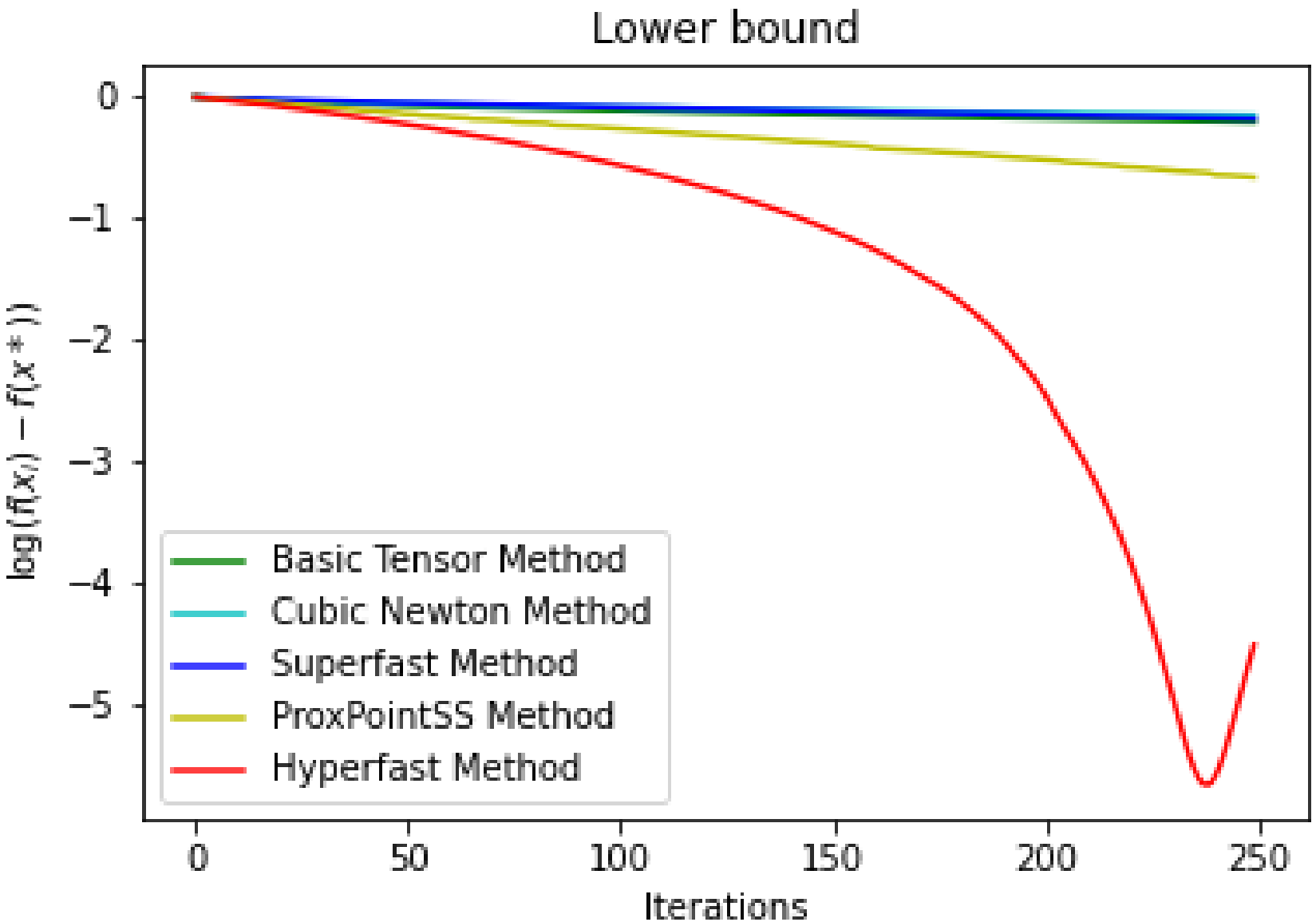
Implementation Time

Method	Iteration	Time	Time/iteration
Cubic Newton Method	1000	21 сек.	0.021 сек./итер.
Basic Tensor Method	1000	65 сек.	0.065 сек./итер.
Superfast Method	1000	63 сек.	0.063 сек./итер.
ProxPointSS Method	250	41 сек.	0.164 сек./итер.
Hyperfast Method	250	24 сек.	0.096 сек./итер.

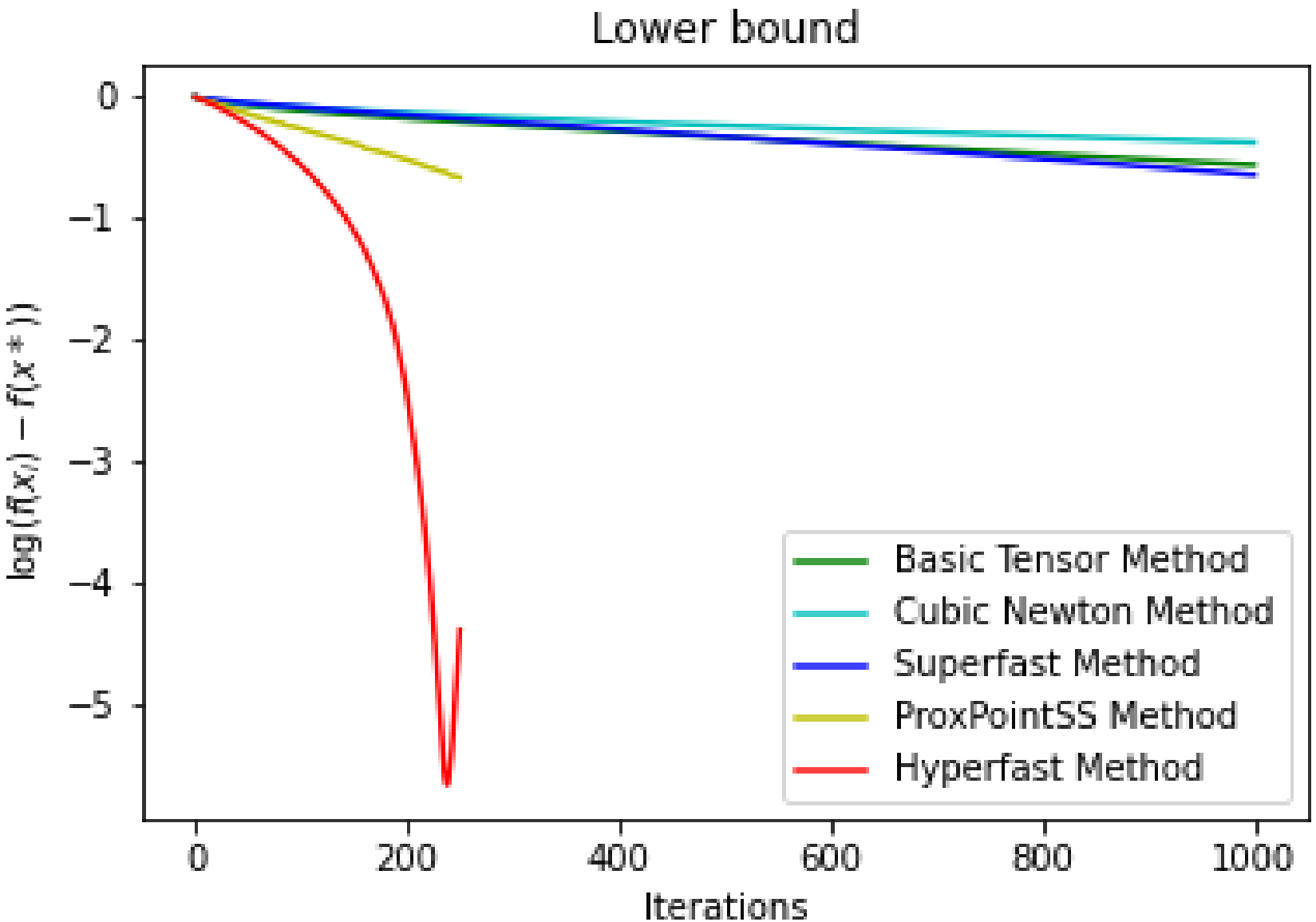
Experiments



Experiments



Experiments



Problem formulation

Empirical Risk Minimization(ERM)

$$\min_{x\in\mathbb{R}^d} F(x) \triangleq \frac{1}{M}\sum_{i=1}^{M} \ell(x;\xi_i,\eta_i), \tag{1}$$

where $\{\zeta_i = (\xi_i,\eta_i)\}_{i=1}^M$ are training samples, and ℓ is a convex loss function with respect to x.

Convexity

Furthermore, we assume F is L_F-smooth and σ_F-strongly convex, i.e.,

$$\sigma_F I_d \preceq \nabla^2 F(x) \preceq L_F I_d,$$

where I_d is the d-dimensional identity matrix. The condition number of F is denoted as $\kappa_F = L_F/\sigma_F$, and the solution to (1) as x_*.

Distributed setup

Distributed configuration

Data is distributed uniformly among m computing units/nodes/agents such that $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$ and $M = mn$. That is, each machine $j \in \{1, \ldots, m\}$ locally stores n samples $\mathcal{D}_j = \{\xi_i^{(j)}, \eta_i^{(j)}\}_{i=1}^n$. Moreover, there is a central node, that is able to communicate with all the worker nodes.

Distributed ERM

Each agent j has a local empirical risk, denoted as $F_j(x) \triangleq (1/n) \sum_{i=1}^n \ell(x; \xi_i^{(j)}, \eta_i^{(j)})$, where $F_1(x)$ is a central node. Thus,

$$F(x) = \frac{1}{M} \sum_{j=1}^{m} F_j(x) = \frac{1}{nm} \sum_{j=1}^{m} \sum_{i=1}^{n} \ell(x; \xi_i^{(j)}, \eta_i^{(j)}).$$

Statistically preconditioning

Reference function

Following the same algorithmic structure as DANE and SPAG, we define a reference function

$$\phi(x) = \frac{1}{n}\sum_{k=1}^{n} \ell(x, \zeta_k) + \frac{\mu}{2}\|x\|_2^2,$$

where the examples ζ_k are taken from the node which is chosen to be central. It follows that $\phi(x)$ is L_ϕ-smooth, and σ_ϕ-strongly convex.

Statistical similarity

The value of the parameter μ is set to be a upper bound that quantifies how statistically similar the function F_1 is from F, i.e., we assume that with high probability, it holds that

$$\|\nabla^2 F(x) - \nabla^2 F_1(x)\|_2 \leq \mu.$$

Relative smoothness

Relative condition number

$F(x)$ is $L_{F/\phi}$-relative smooth and $\sigma_{F/\phi}$-relative strongly convex with respect to $\phi(x)$, i.e.,

$$\sigma_{F/\phi}\nabla^2\phi(x) \leq \nabla^2 F(x) \leq L_{F/\phi}\nabla^2\phi(x),$$

with $L_{F/\phi} = 1$, $\sigma_{F/\phi} = \sigma_F/(\sigma_F + 2\mu)$, and $\kappa_{F/\phi} = L_{F/\phi}/\sigma_{F/\phi}$

Bregman divergence

The Bregman divergence is defined as

$$D_\phi(x, y) \triangleq \phi(x) - \phi(y) - \nabla\phi(y)^\top(x - y).$$

Bregman proximal method

$$x_{k+1} \underset{x\in\mathbb{R}^d}{\operatorname{argmin}} \left\{\langle \nabla F(x_k), x - x_k\rangle + L_{F/\phi} D_\phi(x, x_k)\right\}.$$

Inexact SPAG

Algorithm 1 InSPAG $(L_{F/\phi}, \sigma_{F/\phi}, x_0, D)$

1: **Input:** D s.t. $x_* \in B_2(0, D)$, $\hat{D}_\phi^2 = 2L_\phi D^2$, $L_{F/\phi}$, $\sigma_{F/\phi}$, G_0.
2: Set $y_0 = u_0 = x_0 \in B_2(0, D)$, $\alpha_0 \triangleq 0$, $A_0 \triangleq \alpha_0$.
3: **for** $t \geq 0$ **do**
4: **At the central node**
5: Find the smallest integer $i_t \geq 0$ such that

$$D_\phi(x_{t+1}, y_{t+1}) \leq \frac{G_{t+1}\alpha_{t+1}^2}{A_{t+1}^2} D_\phi(u_{t+1}, u_t), \quad (9)$$

where $G_{t+1}=2^{i_t-1}G_t$, $A_{t+1} \triangleq A_t + \alpha_{t+1}$, and α_{t+1} is the largest root of

$$A_{t+1}(1 + A_t\sigma_{F/\phi}) = L_{F/\phi}G_{t+1}\alpha_{t+1}^2. \quad (10)$$

6: Send $y_{t+1} \triangleq (\alpha_{t+1}u_t + A_t x_t)/A_{t+1}$ to workers.

Inexact SPAG

7: **At every worker node**

8: Compute $\frac{1}{n}\sum_{i=1}^{n}\nabla\ell\big(y_{t+1},\zeta_i^{(j)}\big)$ and send it to the central node.

9: **At the central node**

10: Compute

$$\nabla F(y_{t+1}) = \frac{1}{nm}\sum_{j=1}^{m}\sum_{i=1}^{n}\nabla\ell\big(y_{t+1},\zeta_i^{(j)}\big).$$

11:

$$\text{Solve } u_{t+1} \triangleq \operatorname*{arg\,min}_{x\in B_2(0,D)} \hat{D}_\phi^2/t V_t(x), \tag{11}$$

$$\begin{aligned}\text{where } V_t(x) \triangleq\ & \alpha_{t+1}\langle\nabla F(y_{t+1}), x - y_{t+1}\rangle + \\ & + (1 + A_t\sigma_{F/\phi})D_\phi(x,u_t) + \\ & + \alpha_{t+1}\sigma_{F/\phi}D_\phi(x,y_{t+1}),\end{aligned} \tag{12}$$

12:

$$\text{Set } x_{t+1} \triangleq \frac{\alpha_{t+1}u_{t+1} + A_t x_t}{A_{t+1}}. \tag{13}$$

13: **end for**

Complexity

Total complexity

To obtain an ε accurate minimizer of $F(x)$ total number of subiteration is

$$\tilde{O}\left(\frac{\sqrt{\kappa_F D}}{n^{1/4}}\left(\frac{\|A^\top A\|_2^2 D^2}{\min\{\lambda_1, \lambda_2\} + \mu}\right)^{\frac{1}{5}}\right).$$

Communication complexity

If ϕ is a quadratic function, then $G_t = 1$, and the communication complexity will be $O(\sqrt{\kappa_{F/\phi}})$. In the general case, where ϕ is not quadratic, $G_t \to 1$ linearly with rate $\tilde{O}(\sqrt{\kappa_F})$.

Experiments

Communication complexity

Statistics of the datasets. M is the number of samples, d is the number of features, Feat. is the average number of non-zero features, and Size is the data size in MB.

Dataset	M	d	Feat.	Size
RCV1	20k	47k	74.05	13.7
In-house	710M	3,246k	109.86	650.8k

Experiments

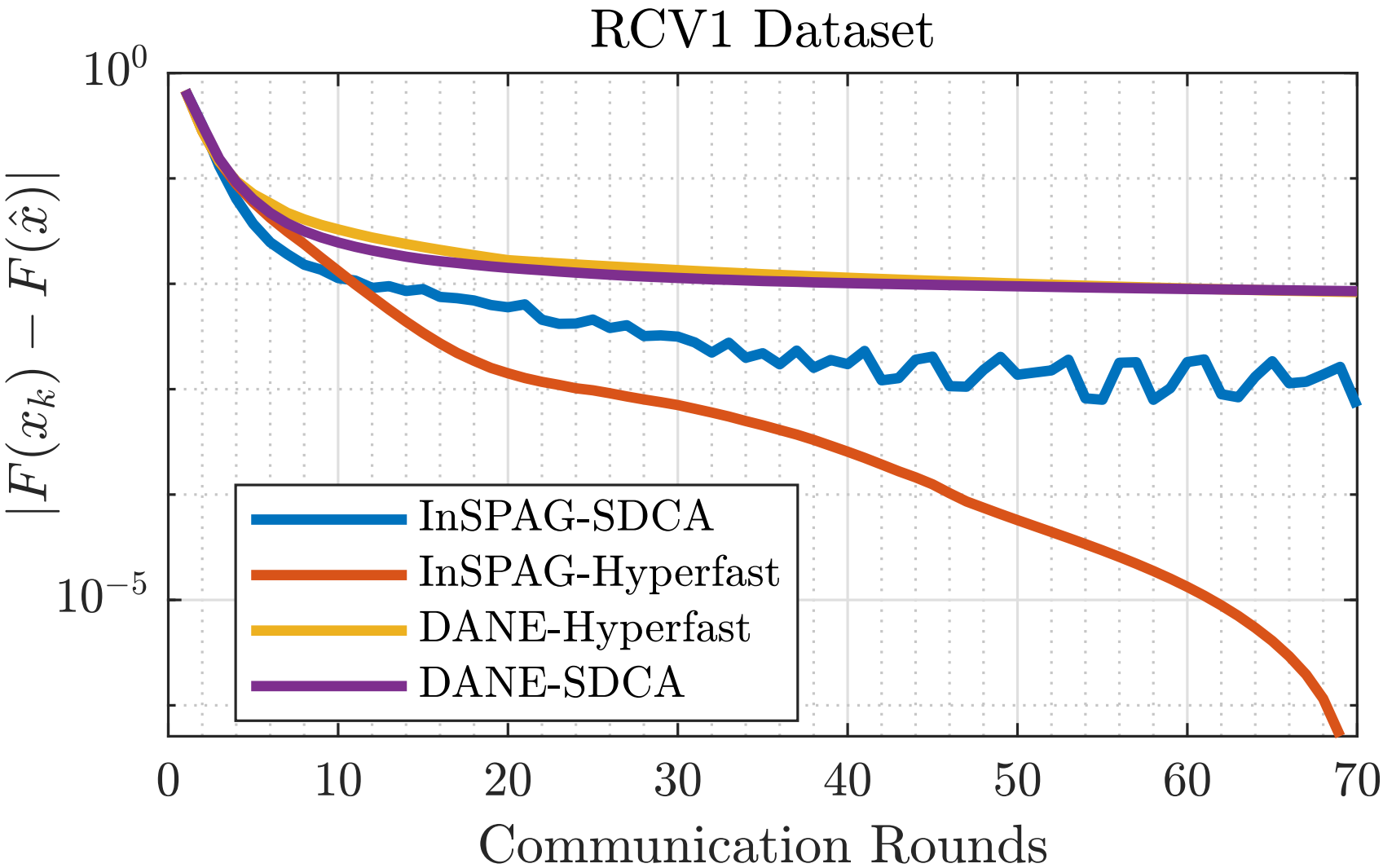


Figure 5: Comparison of the communication rounds for the data set RCV1.

Experiments

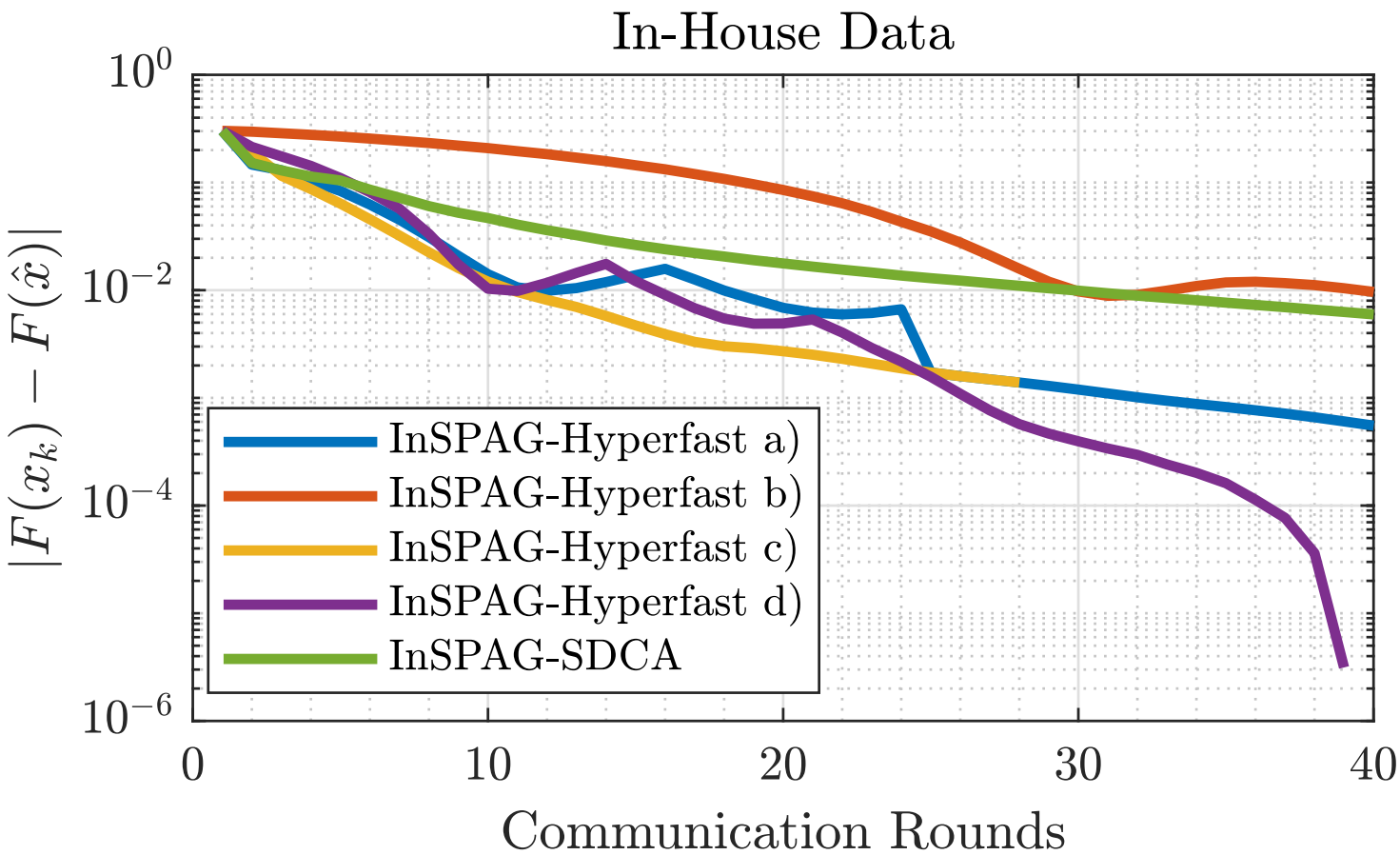


Figure 6: Comparison of the communication rounds for the in house dataset. a) $L_3 = 10$, ADAM learning rate 0.01, $n = 10000$; b) $L_3 = 100$, ADAM learning rate 0.1, $n = 10000$; c) $L_3 = 10$, ADAM learning rate 0.1, $n = 10000$; d) $L_3 = 15$, ADAM learning rate 0.01, $n = 1000$.

Thank you!