A Low Computation Cost Cubic Regularized Quasi-Newton Method for Distributed Optimization: LC3RQN

2024 International Workshop on Modern Optimization and Applications

Wei Hu, Pengcheng Xie, Ya-xiang Yuan, Li Zhang
huwei@amss.ac.cn, xpc@lsec.cc.ac.cn, yyx@lsec.cc.ac.cn, zhangli2022@lsec.cc.ac.cn

Background

- **Introduction of distributed optimization:**
  We want to solve:
  $$\min_x \sum_{i=1}^{n} f_i(x)$$
  which is equivalent to:
  $$\begin{aligned} \min_x \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{s.t.} \quad & x_i = x_j, \quad \forall i, j \in \mathcal{N} \end{aligned}$$
  - Divide large problem into small problems.
  - Intergrate dispersive computing resources.
  - Fitting distributed data structure.
- **Distributed algorithms:**
  - Locally decent steps for reducing function values.
  - Neighboringly communication steos for reaching concensus.

  We should consider both computation and communication cost!.

Main contributions

- Firstly introduce cubic regularized quasi-Newton method into distributed optimization.
- Use closed-form approximately solver of cubic subproblem to efficiently reduce computing time.

We need faster algorithms in communcation-expensive environment!

Why We Need Second Order Algroithms?

- **Drawbacks of first-order methods**:
  - Relatively slow convergence leading to high communication cost.
  - Poor behavior in handling ill-conditioned problems.

We need faster algorithms in communcation-expensive environment!

Challenges in Developing Practical Second-Order-Algorithms

- Difficulties in implement second-order methods in distributed setting:
  - Computing and storing of Hessians.
  - Communication of second-order information.
- Inherent shortcomings of second-order methods:
  - Starting slowly and rely on initial values.
  - Sensitive to hyperparameters.
  - Not suitable for handling **non-convex** problems.
    - * Hessians may not be positive definite.
    - * May be trapped in saddle points.

Tool 1: Cubic Regularization

- Cubic regularization minimize a cubic surrogate function every update:
  $$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s + \frac{\sigma_k}{3}\|s\|^3,$$
  $$x_{k+1} = x_k + s.$$
- Advantages:
  - Global convergent under very mild assumptions.
  - Free of line-search routines and robust to initials.
  - Favorable first-order worst-case iteration complexity bound of $O(\epsilon^{-3/2})$.

Tool 2: Quasi-Newton Update

- Quasi-Newton update reduce the cost of computing and storing Hessian.
- In distributed setting, quasi-Newton update allows to communicate second-order information through **vectors** rather than **matrices**.

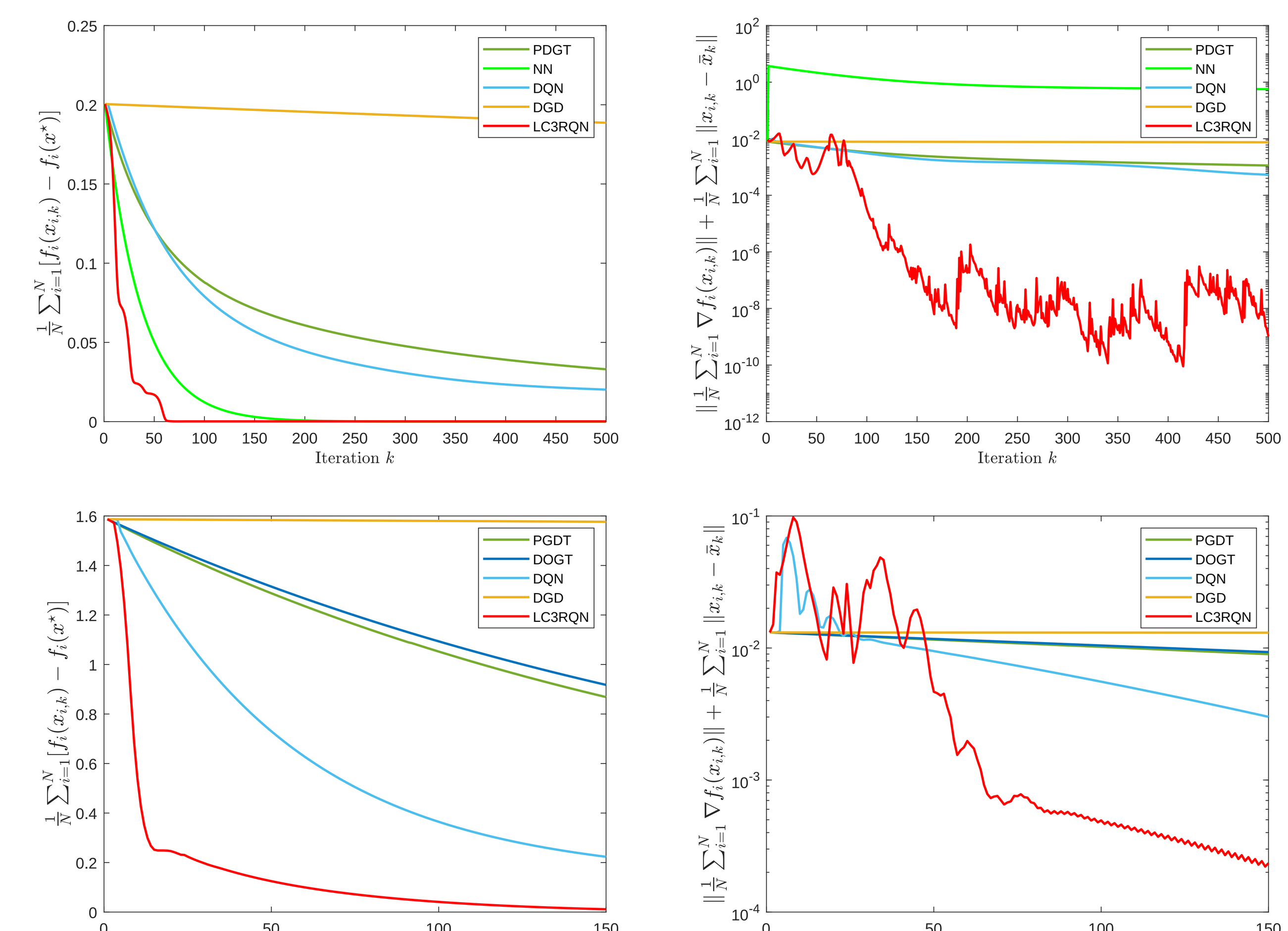
Tool 3: Close-Form Approximate Solver

- From the similarity of the updating form of Levenberg-Marquardt (LM) method and cubic regularization:
  $$x_{k+1} = x_k - \left(\nabla^2 f(x_k) + \lambda_k I\right)^{-1} \nabla f(x_k).$$
  $$x_{k+1} = x_k - \left(\nabla^2 f(x_k) + H\|x_{k+1} - x_k\| I\right)^{-1} \nabla f(x_k).$$
- We choose:
  $$\lambda_k = \sqrt{H\|\nabla f(x_k)\|}.$$

Complete Algorithm: LC3RQN

- **Updates of local variable:**
  $$x_i(t+1) = \sum_{j \in n_i} w_{ij} \text{Solve}(x_i(t), v_i(t), H_i(t), \overline{q}_i(t));$$
- **Gradient tracking:**
  $$v_i(t+1) = \sum_{j \in n_i} w_{ij} \left(v_j(t) + \nabla f_j(x_j(t+1)) - \nabla f_j(x_j(t))\right);$$
- **Quasi-Newton update for Hessian approximation:**
  $$H_i(t+1) = H_i(t) + \frac{s_i(t) s_i(t)^T}{s_i(t)^T y_i(t)} - \frac{H_i(t) y_i(t) y_i(t)^T H_i(t)}{y_i(t)^T H_i(t) y_i(t)};$$
- **Lipschitz constant estimation:**
  $$\overline{q}_i(t) = \frac{\|v_i(t) - v_i(t-1) - H_i(t-1)(x_i(t) - x_i(t-1))\|}{\|x_i(t) - x_i(t-1)\|^2};$$
- **The Solve function is:**
  $$\text{Solve}(x_i(t), v_i(t), H_i(t), \overline{q}_i(t)) = x_i(t) - (H_i(t) + \lambda_i(t) I)^{-1} v_i(t);$$
  where $\lambda_k = \sqrt{q_i(t)\|v_i(t)\|}$, and $q_i(t) = \max\left\{\overline{q}_i(t), \frac{q_i(t-1)}{2}\right\}$.

Numerical Results



References

[1] Y. Nesterov and B. Polyak, "Cubic Regularization of Newton Method," Mathematical Programming, vol. 108, no. 1, 2006.

[2] C. Cartis, N. Gould, and P. Toint, "Adaptive cubic regularization methods for unconstrained optimization. Part I: Motivation, convergence, and numerical results," Mathematical Programming, vol. 127, no. 2, 2011.

[3] K. Mishchenko, "Regularized Newton Method with Global Convergence," SIAM Journal on Optimization, vol. 33, no. 3, 2023.

[4] O. Shorinwa and M. Schwager, "Distributed Quasi-Newton Method for Multi-Agent Optimization," *arXiv:2201.03759*, 2024.