Distributed Second Order Methods with Fast Rates and Compressed Communication
Rustem Islamov\(^1\), Xun Qian\(^1\), Peter Richtárik\(^1\)
\(^{1}\)KAUST \(^{2}\)MPIT

The Problem

\[
\min \left\{ P(x) - f(x) + \frac{1}{2} \|x\|^2 \right\},
\]

Function \(f\) is convex, and has an "average of averages" structure:

\[
f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x), \quad f_i(x) = \frac{1}{m} \sum_{j=1}^{m} f_{ij}(x),
\]

and \(\lambda \geq 0\) is a regularization parameter. Each \(f_i\) is a function of the form: \(f_i(x) = \varphi_i(x_i)\). The Hessian of \(f_i\) at point \(x_i\) is

\[
H_{ij}(x_i) = \nabla^2 \varphi_i(x_i), \quad H_{ij}(x_i) = \nabla^2 \varphi_i(x_i). \tag{3}
\]

The Hessian \(H_i\) of local functions \(f_i\) and the Hessian \(H(x)\) of \(f\) can be represented as linear combination of one-rank matrices.

Assumptions

We assume that Problem (1) has at least one optimal solution \(x^*\). For all \(x_i\) and \(x_i^*\), \(\varphi_i\) is \(\gamma\)-smooth, twice differentiable, and its second derivative \(\varphi_i''\) is \(\alpha\)-Lipschitz continuous.

Main goal

Our goal is to develop a communication efficient Newton-type method for distributed optimization.

Naive distributed implementation of Newton's method

Newton's step: \(x^{k+1} = x^k - \frac{1}{\lambda} \nabla^2 P(x^k)^{-1} \nabla P(x^k)\)

Each node: computes the local Hessian \(H_i(x_i)\) and gradient \(\nabla f_i(x_i)\), then sends them to the server.

Server: averages the local Hessians and gradients to produce \(H(x)\) and \(\nabla f(x)\), respectively. Adds \(\lambda I\) to \(H(x)\) and \(\lambda \nabla f(x)\) to \(\nabla f(x)\), then performs Newton-Raphson step. Next, it sends \(x^{k+1}\) back to the workers.

Pros:
- Fast local quadratic convergence rate
- Rate is independent on the condition number
- Requires \(O(d)\) bits to be communicated by each worker to the server, where \(d\) is typically very large

Cons:
- Cannot be implemented in practice

Lemma 1 (Convergence of NS)

Assume that \(H(x_i) \succeq \mu I\) for some \(\mu \geq 0\) and that \(\mu + \gamma > 0\). Then for any starting point \(x^0 \in \mathbb{R}^d\), the iterates of Newton-STAR satisfy the following inequality:

\[
\|x^k - x^*\|^2 \leq \frac{\|x^0 - x^*\|^2}{k}, \tag{5}
\]

Pros:
- Fast local quadratic convergence rate
- Rate is independent on the condition number
- Communication cost is \(O(d)\) per iteration

Cons:
- Cannot be implemented in practice

NEWTON-LEARN (NL1)

How to address the communication bottleneck?
- Compressed communication
- Taking advantage of the structure of the problem

In NEWTON-LEARN we maintain a sequence of vectors

\[
h_i^{(k)} = (h_i^{(k)}), \ldots, h_i^{(k)} \in \mathbb{R}^n\]

for all \(s = 1, \ldots, n\) and compare it to the iterates \(k \geq 0\), with the goal of learning the values \(h_i(x^*)\) for all \(i, j\).

Using \(h_i(x^*) \approx h_i(x^{k+1})\) as \(k \to +\infty\), we can estimate the Hessian \(H(x^*)\) via

\[
H(x^*) \approx H^k = \frac{1}{n} \sum_{i=1}^{n} h_i^{(k)} h_i^{(k)^T}.
\]

CUBIC-NEWTON-LEARN

We also constructed a method (CNL) with global convergence guarantees using cubic regularization [3].

Pros:
- Local linear and superlinear rates
- Global linear rate in the strongly convex case and global sublinear rate in the convex case
- Rates are independent on the condition number
- \(O(d)\) bits are communicated per iteration

Experiments

Algorithm 1: NL1: NEWTON-LEARN (\(\lambda > 0\) case)

Parameters:
- learning rate \(\eta > 0\)
- Initialization: \(x^0 \in \mathbb{R}^d\), \(h_i^0 \in \mathbb{R}^n\)
- \(H^0 = \sum_{i=1}^{n} h_i^0 h_i^{0T}\in \mathbb{R}^{d \times d}\)

for \(k \geq 0, 1, \ldots, d\)

Broadcast \(x^k\) to all workers

for each node \(i = 1, \ldots, d\)

Compute local gradient \(\nabla f_i(x^k)\)

\[
h_i^{k+1} = h_i^k + \frac{\eta}{2} \nabla^2 f_i(x^k) h_i^{k+1} \nabla f_i(x^k) h_i^{k+1} - h_i^{k+1}
\]

and corresponding \(a_i\) to server

end

end

end

end

Convergence theory

The analysis relies on the Lyapunov function

\[
1 \leq \frac{1}{\lambda} \eta \|x^k - x^*\|^2 + \frac{1}{\lambda} \eta \|x^k - x^k - 1\|^2 + \parallel \nabla f(x^k) \parallel^2,
\]

\(R = \max \{a_i\}\).

Theorem 2 (convergence of NL1)

Theorem 2. Let each \(\varphi_i(x)\) be convex, \(\lambda > 0\), and \(\eta \leq \frac{1}{\lambda}\). Assume that \(\|x^k - x^k - 1\|^2 \leq \frac{1}{\lambda} \eta^2\) for all \(k \geq 0\). Then for Algorithm 1 we have the inequalities

\[
\frac{1}{\lambda} \eta \|x^k - x^k - 1\|^2 \leq \frac{1}{\lambda} \eta \|x^k - x^k - 1\|^2 + \frac{1}{\lambda} \eta \|x^k - x^k - 1\|^2,
\]

where \(\eta = 1 - \frac{\|x^k - x^k - 1\|^2}{\lambda^2} + \frac{\lambda}{\eta} \frac{1}{\lambda} \eta \|x^k - x^k - 1\|^2\), which is independent on the condition number.

Assumption on \(\|x^k - x^k - 1\|^2\) can be relaxed using the following lemma.

Lemma 1

Assume \(h_i^0\) is a convex combination of \(h_i(x_i^k), \ldots, h_i(x_i^k))\) for all \(s \leq 4\).

Main properties:
- \(h_i^k \geq 0\)
- \(h_i^k \geq 0\) for all \(i, j\)
- update is sparse: \(h_i^k l_i^k h_i^k \leq s, \) where \(s = O(1)\)
- \(H^0 \succeq 0\)

Pros:
- Fast local quadratic convergence rate
- Rate is independent on the condition number
- Communication cost is \(O(d)\) per iteration

Cons:
- Cannot be implemented in practice

References