A Newton Tracking Algorithm with Exact Linear Convergence Rate for Decentralized Consensus Optimization

Jiaojiao Zhang

Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong

Joint work with
Qing Ling (SYSU) and Anthony Man-Cho So (CUHK)

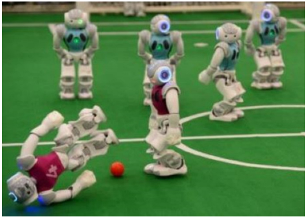
IEEE-CDC 2020

Outline

Outline

Background

Decentralized optimization




Background

Consider the decentralized convex optimization problem

$$x^* = \arg\min_{x \in \mathbb{R}^p} \sum_{i=1}^{n} f_i(x)$$

- $f_i(x)$ is a convex and twice continuously differentiable function.

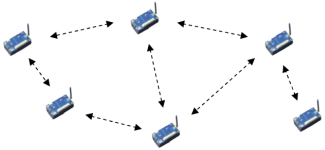


Fig 1. Decentralized network

- Data are distributed across a connected network of n nodes.
- Each node is only allowed to send/receive information to/from its neighboring nodes.
- All the nodes cooperate to get a common variable.

Background

An equivalent decentralized formulation is

$$\{x_i^*\}_{i=1}^n := \arg\min_{\{x_i\}_{i=1}^n} \sum_{i=1}^n f_i(x_i)$$
$$\text{s.t. } x_i = x_j, \forall j \in \mathcal{N}_i, \forall i$$

Two key components for decentralized optimization

- consensus - all nodes must agree on the same state, i.e., $x_1^* = \cdots = x_n^*$.
- optimality - the same state should be the minimizer of the original problem, i.e., $x_1^* = \cdots = x_n^* = x^*$.

Related First-order Work

Primal method

- Gradient methods [Nedic 2009, Yuan 2016]
- Gradient Tracking [Lorenzo 2015, Qu 2017, Nedic 2017,Sun 2019]

Primal-Dual method

- Decentralized Alternating Direction Method of Multipliers (DADMM) [Shi 2014,Chang 2015]
- Decentralized linearized ADMM [Ling 2015]
- Dual Ascent [Maros 2018]

Other method

- EXTRA [Shi 2015]
- NIDS [Li 2019]

Related Second-order Work

Penalized second-order algorithms converge to a neighborhood of an optimal solution

- Network Newton [Mokhtari 2016]
- Distributed asynchronous Newton-based algorithm [Mansoori 2019]

Primal-dual second-order methods achieve exact convergence with linear rates

- DQM [Mokhtari 2016]
- ESOM [Mokhtari 2016]

Second-order methods with superlinear convergence rates under stricter conditions

- Distributed averaged quasi-Newton method for a master-slave network [Soori 2019]
- Polyak's adaptive Newton method running a finite-time set-consensus inner loop [Zhang 2020]

Outline

We make the following assumptions

Assumption 1

Introduce the mixing matrix W with elements $w_{ij} \geq 0$. $w_{ij} = 0$ if and only if $j \notin \mathcal{N}_i \cup \{i\}$. Further, $W^T = W$, $W 1_{n\times 1} = 1_{n\times 1}$ and $null(I - W) = span(1_{n\times 1})$.

Assumption 2

The local objective functions $f_i(x_i)$ are twice differentiable. Hessians $\nabla^2 f_i(x_i)$ are bounded by

$$\mu_f I_p \preceq \nabla^2 f_i(x_i) \preceq L_f I_p.$$

Consider the decentralized optimization problem

$$\{x_i^*\}_{i=1}^n := \arg\min_{\{x_i\}_{i=1}^n} \sum_{i=1}^n f_i(x_i)$$
$$\text{s.t. } x_i = x_j, \forall j \in \mathcal{N}_i, \forall i$$

Global negative Newton direction at $\bar{x}^t \triangleq \frac{1}{n}\sum_{i=1}^n x_i^t$ is

$$u^t \triangleq \left(\frac{1}{n}\sum_{i=1}^n \nabla^2 f_i\left(\bar{x}^t\right)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \nabla f_i\left(\bar{x}^t\right)\right)$$

Our idea: Use a local u_i^t to track the negative global Newton direction such that $u_i^t \approx u^t$.

Algorithm Development

The proposed **Newton tracking**

$$
\begin{aligned}
x_i^{t+1} =& x_i^t - u_i^t \\
u_i^{t+1} =& \left(\nabla^2 f_i\left(x_i^{t+1}\right) + \epsilon I_p\right)^{-1} \left[\left(\nabla^2 f_i\left(x_i^t\right) + \epsilon I_p\right) u_i^t + \nabla f_i\left(x_i^{t+1}\right) - \nabla f_i\left(x_i^t\right)\right. \\
& \left. + 2\alpha\left(x_i^{t+1} - \sum_{j \in \mathcal{N}_i} w_{ij} x_j^{t+1}\right) - \alpha\left(x_i^t - \sum_{j \in \mathcal{N}_i} w_{ij} x_j^t\right)\right]
\end{aligned}
$$

with initialization $x_i^0 = 0_p$ and $u_i^0 = \left(\nabla^2 f_i\left(0_p\right) + \epsilon I_p\right)^{-1} \nabla f_i\left(0_p\right)$.

- The global Hessian $\frac{1}{n}\sum_{i=1}^n \nabla^2 f_i\left(\bar{x}^{t+1}\right)$ is replaced by the regularized local Hessian $\nabla^2 f_i\left(x_i^{t+1}\right) + \epsilon I_p$.
- The global gradient $\frac{1}{n}\sum_{i=1}^n \nabla f_i\left(\bar{x}^t\right)$ is replaced by three terms that are locally computable.

Why $u_i^t \approx u^t$?

Rewrite Newton tracking

$$\left(\nabla^2 f_i\left(x_i^{t+1}\right)+\epsilon I_p\right) u_i^{t+1} = \Big[\left(\nabla^2 f_i\left(x_i^t\right)+\epsilon I_p\right) u_i^t + \nabla f_i\left(x_i^{t+1}\right)-\nabla f_i\left(x_i^t\right) + 2\alpha\Big(x_i^{t+1}-\sum_{j\in\mathcal{N}_i} w_{ij}x_j^{t+1}\Big)-\alpha\Big(x_i^t-\sum_{j\in\mathcal{N}_i} w_{ij}x_j^t\Big)\Big]$$

Sum up over $i = 1 \ldots n$ and invoke the double stochasticity of W

$$\sum_{i=1}^{n}\left(\nabla^2 f_i\left(x_i^{t+1}\right)+\epsilon I_p\right) u_i^{t+1} = \sum_{i=1}^{n}\Big[\left(\nabla^2 f_i\left(x_i^t\right)+\epsilon I_p\right) u_i^t + \nabla f_i\left(x_i^{t+1}\right)-\nabla f_i\left(x_i^t\right)\Big]$$

With $\sum_{i=1}^n \nabla f_i\left(x_i^0\right) = \sum_{i=1}^n \left(\nabla^2 f_i\left(x_i^0\right)+\epsilon I_p\right) u_i^0$, sum up over time t

$$\sum_{i=1}^{n}\left(\nabla^2 f_i\left(x_i^t\right)+\epsilon I_p\right) u_i^t = \sum_{i=1}^{n}\nabla f_i\left(x_i^t\right)$$

Thus, when x_i^t is close to $\bar{x}^t$, u_i^t tracks a regularized Newton direction.

Newton Tracking

Newton tracking can be written in a compact form

$$
\begin{aligned}
\mathrm{x}^{t+1} =& \mathrm{x}^{t} - \mathrm{u}^{t} \\
\mathrm{u}^{t+1} =& \left(\mathrm{H}^{t+1}\right)^{-1}\Big[\mathrm{H}^{t}\mathrm{u}^{t} + \nabla f\left(\mathrm{x}^{t+1}\right) - \nabla f\left(\mathrm{x}^{t}\right) \\
& +\alpha(\mathrm{I}-\mathrm{W})\left(2\mathrm{x}^{t+1}-\mathrm{x}^{t}\right)\Big]
\end{aligned}
$$

- x (or u) stacks local variables such as $\mathrm{x} \triangleq [x_1; \ldots; x_n] \in \mathbb{R}^{np}$
- $\mathrm{W} \triangleq W \otimes I_p \in \mathbb{R}^{np \times np}$
- $f(\mathrm{x}) = f(x_1, \cdots, x_n) = \sum_{i=1}^{n} f_i(x_i)$.
- $\nabla f(\mathrm{x}) = [\nabla f_1(x_1); \ldots; \nabla f_n(x_n)] \in \mathbb{R}^{np}$
- $\mathrm{H} \in \mathbb{R}^{np \times np}$ is a block diagonal matrix whose i-th diagonal block is $\nabla^2 f_i(x) + I_p$

Connection with Gradient Tracking

Gradient tracking proceeds as

$$\begin{aligned}\mathrm{x}^{t+1} &= \mathrm{W}\mathrm{x}^{t} - \alpha \mathrm{y}^{t} \\ \mathrm{y}^{t+1} &= \mathrm{W}\mathrm{y}^{t} + \nabla f\left(\mathrm{x}^{t+1}\right) - \nabla f\left(\mathrm{x}^{t}\right)\end{aligned}$$

with initialization $\mathrm{y}^{0} = \nabla f\left(\mathrm{x}^{0}\right)$.
To see the connection with Newton tracking, we rewrite

$$\begin{aligned}\mathrm{x}^{t+1} =& \mathrm{x}^{t} - \mathrm{r}^{t} \\ \mathrm{r}^{t+1} =& \color{red}{\mathrm{W}\mathrm{r}^{t}} + \color{blue}{\alpha\left[\nabla f\left(\mathrm{x}^{t+1}\right) - \nabla f\left(\mathrm{x}^{t}\right)\right]} + \color{teal}{(\mathrm{I}-\mathrm{W})\left(\mathrm{x}^{t+1} - \mathrm{W}\mathrm{x}^{t}\right)}\end{aligned}$$

with $\mathrm{r}^{t} = (\mathrm{I}-\mathrm{W})\mathrm{x}^{t} + \alpha \mathrm{y}^{t} \in \mathbb{R}^{np}$ and $\sum_{i=1}^{n} r_i^0 = \alpha \sum_{i=1}^{n} \nabla f_i\left(x_i^0\right)$. Sum up over node i and time t

$$\sum_{i=1}^{n} r_i^t = \alpha \sum_{i=1}^{n} \nabla f_i\left(x_i^t\right)$$

Thus, when x_i^t is close to $\bar{x}^t$, r_i^t tracks a scaled gradient direction.

Connection with Primal-dual Algorithms

Under Assumption 1, we have $(\mathrm{I}-\mathrm{W})^{\frac{1}{2}}\mathrm{x}=0$ if and only if $x_1=\cdots=x_n$. Thus, the original problem is equivalent to

$$\mathrm{x}^* \triangleq \arg\min_{\mathrm{x}} f(\mathrm{x}) \quad \text{s.t. } (\mathrm{I}-\mathrm{W})^{\frac{1}{2}}\mathrm{x}=0.$$

The augmented Lagrangian is

$$L(\mathrm{x},\mathrm{v})=f(\mathrm{x})+\left\langle \mathrm{v},(\mathrm{I}-\mathrm{W})^{\frac{1}{2}}\mathrm{x}\right\rangle+\frac{\alpha}{2}\mathrm{x}^T(\mathrm{I}-\mathrm{W})\mathrm{x}.$$

We use a quadratic approximation of f and a linear approximation of $\mathrm{x}\mapsto\frac{\alpha}{2}\mathrm{x}^T(\mathrm{I}-\mathrm{W})\mathrm{x}$. The update of x^{t+1} is given by the solution of

$$\begin{aligned}\min_{\mathrm{x}}&\left\langle \nabla f\left(\mathrm{x}^t\right)+(\mathrm{I}-\mathrm{W})^{\frac{1}{2}}\mathrm{v}^t+\alpha(\mathrm{I}-\mathrm{W})\mathrm{x}^t,\mathrm{x}-\mathrm{x}^t\right\rangle\\&+\frac{1}{2}\left(\mathrm{x}-\mathrm{x}^t\right)^T\nabla^2 f\left(\mathrm{x}^t\right)\left(\mathrm{x}-\mathrm{x}^t\right)+\frac{\epsilon}{2}\left\|\mathrm{x}-\mathrm{x}^t\right\|^2.\end{aligned}$$

Connection with Primal-dual Algorithms

Thus, the updates of x^{t+1} and v^{t+1} are

$$\mathsf{x}^{t+1} = \mathsf{x}^t - \left(\mathsf{H}^t\right)^{-1}\left[\nabla f\left(\mathsf{x}^t\right) + (\mathsf{I}-\mathsf{W})^{\frac{1}{2}}\mathsf{v}^t + \alpha(\mathsf{I}-\mathsf{W})\mathsf{x}^t\right]$$
$$\mathsf{v}^{t+1} = \mathsf{v}^t + \alpha(\mathsf{I}-\mathsf{W})^{\frac{1}{2}}\mathsf{x}^{t+1}$$

where we set $x^0 = 0$ and $v^0 = 0$. By manipulation, we get

$$\begin{aligned}\mathsf{x}^{t+1} =& \mathsf{x}^t - \left(\mathsf{H}^t\right)^{-1}\mathsf{q}^t\\ \mathsf{q}^{t+1} =& \mathsf{q}^t + \nabla f\left(\mathsf{x}^{t+1}\right) - \nabla f\left(\mathsf{x}^t\right) + \alpha(\mathsf{I}-\mathsf{W})\left(2\mathsf{x}^{t+1} - \mathsf{x}^t\right),\end{aligned}$$

which is equivalent to Newton tracking in the sense that $\mathsf{u}^t = \left(\mathsf{H}^t\right)^{-1}\mathsf{q}^t$.

Outline

Theorem 1

Under Assumptions 1 and 2, suppose that the parameters ϵ and α satisfy $\epsilon - \alpha\lambda_{\max}(\mathrm{I}-\mathrm{W}) > \frac{4L_f^2}{\mu_f}$. Then Newton tracking starting with $x_i^0 = 0_p$ and $u_i^0 = \left(\nabla^2 f_i(0_p) + \epsilon I_p\right)^{-1} \nabla f_i(0_p)$ satisfies

$$\left\|\zeta^{t+1} - \zeta^*\right\|_{\mathrm{G}}^2 \leq \frac{1}{1+\delta'} \left\|\zeta^t - \zeta^*\right\|_{\mathrm{G}}^2,$$

where $\delta' > 0$.

- Define $\zeta^t = \begin{bmatrix} \mathrm{x}^t \\ \mathrm{v}^t \end{bmatrix}, \zeta^* = \begin{bmatrix} \mathrm{x}^* \\ \mathrm{v}^* \end{bmatrix}, \mathrm{G} = \begin{bmatrix} \epsilon\mathrm{I} - \alpha(\mathrm{I}-\mathrm{W}) & 0 \\ 0 & \frac{1}{\alpha}\mathrm{I} \end{bmatrix}$.
- Theorem 1 shows that the sequence $\{\left\|\zeta^{t+1} - \zeta^*\right\|_{\mathrm{G}}^2\}_t$ converges linearly with the factor $\frac{1}{1+\delta'}$.

Outline

Decentralized logistic regression problem

$$\min_{x \in \mathbb{R}^p} \frac{\rho}{2}\|x\|^2 + \sum_{i=1}^{n} \sum_{j=1}^{m_i} \ln\left(1 + \exp\left(-\left(\mathrm{o}_{ij}^T x\right) \mathrm{p}_{ij}\right)\right)$$

- Node i has access to m_i training samples, $(\mathrm{o}_{ij}, \mathrm{p}_{ij}) \in \mathbb{R}^p \times \{-1, +1\}$.
- Relative error $\|\mathrm{x}^t - \hat{\mathrm{x}}^*\| \,/\, \left\|\mathrm{x}^0 - \hat{\mathrm{x}}^*\right\|$.

Comparison with Related Methods

- $n = 10$, connectivity ratio=0.5, $m_i = 12$, $p = 8$, $\rho = 0.001$

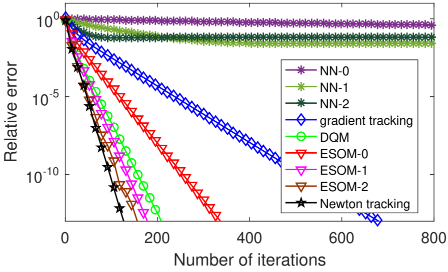


Fig 2. Relative errors of Newton tracking, gradient tracking, NN-K, ESOM-K, and DQM versus number of iterations.

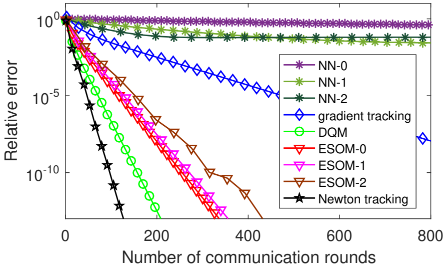


Fig 3. Relative errors of Newton tracking, gradient tracking, NN-K, ESOM-K, and DQM versus number of communication rounds.

Effect of Network Topology

- Four different topologies including line graph, cycle graph, random graphs, and complete graph
- $n = 10$, $m_i = 12$, $p = 8$, connectivity ratio=0.5, $\rho = 0.001$

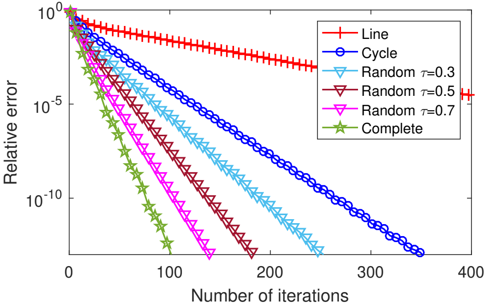


Fig 4. Relative errors of Newton tracking for different topologies

Outline

Conclusions

To summarize

- We propose Newton tracking, in which each node updates its local variable along a modified local Newton direction.
- Newton tracking employs a fixed step size and yet can still be proven to converge to an exact solution.
- We give the connections between Newton tracking and several existing methods, including gradient tracking and primal-dual algorithms.
- Newton tracking is linearly convergent under the strong convexity assumption.

Thank you !