When do distributed optimization algorithms meet centralized counterparts and beyond?

Jinming Xu

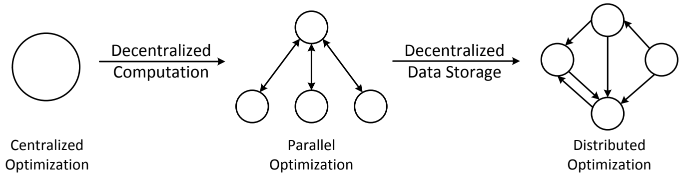
Zhejiang University

jimmyxu@zju.edu.cn

Online, Apr-25-2020

A Big Picture

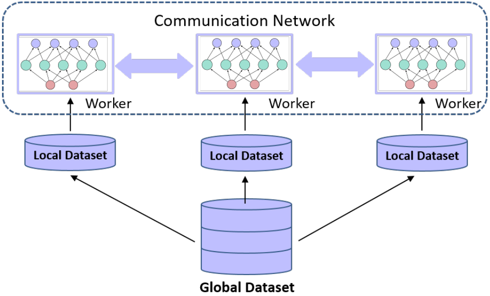
- A new paradigm for large-scale optimization



- What distributed structure can bring to us?
  - Robust and scablable,
  - Amenable to asynchronous running,
  - Privacy-preserving,
  - Speedup in overall running time

An Example from Distributed Learning

- Data-parallel training beyond the Datacenter.

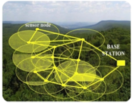


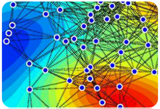
- Other parallel structure
  - Model parallel: dealing with large-scale model parameters.
  - Hybrid parallel: combining data-parallel and model-parallel.

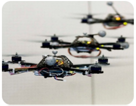
Other Examples

Distributed Estimation

- Source Localization
- Field Monitoring
- Distributed Learning







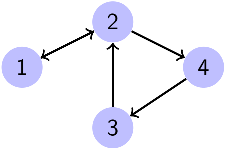
Distributed Control

- Wind Farm
- Smart/Micro Grid
- Formation Flying

Outline

Some Preliminaries for The Talk

- Graph
  - **connectivity**: connected if there is a path between every pair of nodes
  - **spanning tree**: a subgraph that is a tree covering all nodes with minimum possible number of edges
  - **root**: a subset of nodes that are able to reach all other nodes
- Weight Matrix[1] $\mathbf{W} := [w_{ij}]$
  - row-stochastic: $\mathbf{W1} = \mathbf{1}$
  - column-stochastic: $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$
  - doubly-stochastic: $\mathbf{W1} = \mathbf{1}, \mathbf{1}^T\mathbf{W} = \mathbf{1}^T$
- Matrix Induced Graph: $\mathbf{W} \to \mathcal{G}_W$



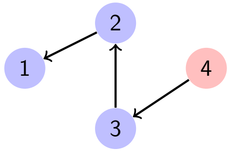
A Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

- $v_i \in \mathcal{V}$: an agent
- $e_{ij} \in \mathcal{E}$: the link
- w_{ij}: the weight to e_{ij}
- $\mathcal{N}_i := \{j | e_{ij} \in \mathcal{E}\}$: the neighbors of agent i

[1] $\mathbf{W}$ is non-negative; $\mathbf{1}$: all-one vector.

Some Preliminaries for The Talk

- Graph
  - **connectivity**: connected if there is a path between every pair of nodes
  - **spanning tree**: a subgraph that is a tree covering all nodes with minimum possible number of edges
  - **root**: a subset of nodes that are able to reach all other nodes



A Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

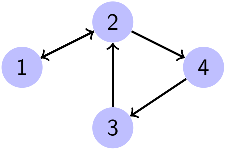
- $v_i \in \mathcal{V}$: an agent
- $e_{ij} \in \mathcal{E}$: the link
- w_{ij}: the weight to e_{ij}
- $\mathcal{N}_i := \{j | e_{ij} \in \mathcal{E}\}$: the neighbors of agent i

- Weight Matrix[1] $\mathbf{W} := [w_{ij}]$
  - row-stochastic: $\mathbf{W}\mathbf{1} = \mathbf{1}$
  - column-stochastic: $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$
  - doubly-stochastic: $\mathbf{W}\mathbf{1} = \mathbf{1}, \mathbf{1}^T\mathbf{W} = \mathbf{1}^T$
- Matrix Induced Graph: $\mathbf{W} \rightarrow \mathcal{G}_W$

[1] $\mathbf{W}$ is non-negative; $\mathbf{1}$: all-one vector.

Some Preliminaries for The Talk

- Graph
  - **connectivity**: connected if there is a path between every pair of nodes
  - **spanning tree**: a subgraph that is a tree covering all nodes with minimum possible number of edges
  - **root**: a subset of nodes that are able to reach all other nodes



A Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

- Weight Matrix[1] $\mathbf{W} := [w_{ij}]$
  - row-stochastic: $\mathbf{W1} = \mathbf{1}$
  - column-stochastic: $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$
  - doubly-stochastic: $\mathbf{W1} = \mathbf{1}, \mathbf{1}^T\mathbf{W} = \mathbf{1}^T$
- Matrix Induced Graph: $\mathbf{W} \rightarrow \mathcal{G}_W$

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & 0 & 0 \\ w_{21} & w_{22} & 0 & w_{24} \\ 0 & w_{32} & 0 & 0 \\ 0 & 0 & w_{43} & w_{44} \end{bmatrix}$$

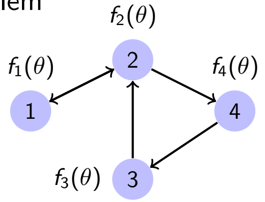
[1] $\mathbf{W}$ is non-negative; $\mathbf{1}$: all-one vector.

Distributed Optimization Problem

- Want to solve the following original problem[2]

$$\min_{\theta \in \mathcal{R}} F(\theta) = \sum_{i=1}^{m} f_i(\theta) \qquad \text{(DOP)}$$

  - $\theta \in \mathcal{R}$: the global decision variable
  - $f_i : \mathcal{H} \to \mathcal{R}$: the cost funciton **known only** by the associated agent i.



A Network Model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

- Equivalent to solve the problem as follows

$$\min_{\mathbf{x} \in \mathcal{R}^m} f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i) \qquad s.t.\ x_i = x_j,\ \forall i, j \in \mathcal{V}$$

  - $\mathbf{x} = [x_1, x_2, \ldots x_m]^T$: local estimates of agents for global optimum $\theta^\star$.

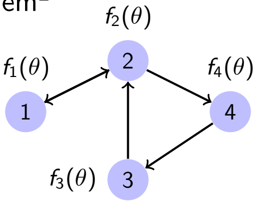
[2] We consider scalar cases only for simplicity.

Distributed Optimization Problem

- Want to solve the following original problem[2]

$$\min_{\theta \in \mathcal{R}} F(\theta) = \sum_{i=1}^{m} f_i(\theta) \qquad \text{(DOP)}$$

  - $\theta \in \mathcal{R}$: the global decision variable
  - $f_i : \mathcal{H} \to \mathcal{R}$: the cost funciton **known only** by the associated agent i.



A Network Model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

- Equivalent to solve the problem as follows

$$\min_{\mathbf{x} \in \mathcal{R}^m} f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i) \qquad s.t.\ x_i = x_j,\ \forall i, j \in \mathcal{V}$$

  - $\mathbf{x} = [x_1, x_2, ...x_m]^T$: local estimates of agents for global optimum $\theta^\star$.

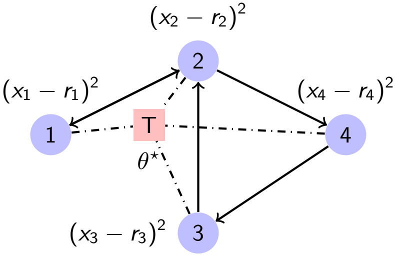
[2]We consider scalar cases only for simplicity.

A Canonical Example: Average Consensus

- A Canonical Example

$$\min_{\mathbf{x}} \sum_{i=1}^{m} (x_i - r_i)^2,$$
$$s.t.\ x_i = x_j, \forall i, j \in \mathcal{V},$$

  - r_i: local measurement to the position of a target,
  - x_i: local estimate of sensor i.



$\theta^\star = \frac{1}{m}\sum_i r_i$: position of target

Task: $x_1 = x_2 = x_3 = x_4 = \theta^\star$

- Average Consensus[3]

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_{j,k}$$

[3] Refer to (Olfati-Saber and Murray, 2004)

A Canonical Example: Average Consensus

- A Canonical Example

$$\min_{\mathbf{x}} \sum_{i=1}^{m} (x_i - r_i)^2,$$
$$s.t.\ x_i = x_j, \forall i, j \in \mathcal{V},$$

  - r_i: local measurement to the position of a target,
  - x_i: local estimate of sensor i.

- Average Consensus[3]

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_{j,k}$$

Lemma (Average Seeking)

If $\mathbf{W}$ *is doubly stochastic, then with* $\mathbf{x}_0 = \mathbf{r}$ *we have*

$$\sum_i x_{i,k} = \sum_i r_i, \forall k \geq 0$$

and, if the graph is connected,

$$x_i \rightarrow \theta^\star = \frac{1}{m} \sum_i r_i,\ \forall i \in \mathcal{V}$$

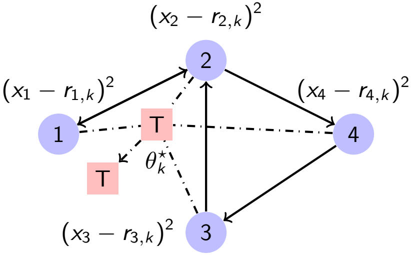
[3] Refer to (Olfati-Saber and Murray, 2004)

A Canonical Example: Dynamic Average Consensus

- A Canonical Example

$$\min_{\mathbf{x}} \sum_{i=1}^{m} (x_i - r_{i,k})^2,$$
$$s.t.\ x_i = x_j, \forall i, j \in \mathcal{V},$$

  - $r_{i,k}$: the local measurement which is **time-varying**.



$\theta_k^\star$: position of target

- Dynamic Average Consensus[4]

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_{j,k} + r_{i,k+1} - r_{i,k}$$

Task: $x_1 = x_2 = x_3 = x_4 \to \theta_k^\star$

[4] Refer to (Zhu and MartÃnez, 2010)

A Canonical Example: Dynamic Average Consensus

- A Canonical Example

$$\min_{\mathbf{x}} \sum_{i=1}^{m} (x_i - r_{i,k})^2,$$
$$s.t.\ x_i = x_j, \forall i, j \in \mathcal{V},$$

  - $r_{i,k}$: the local measurement which is **time-varying**.

- Dynamic Average Consensus[4]

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_{j,k} + r_{i,k+1} - r_{i,k}$$

Lemma (Average Tracking)

If $\mathbf{W}$ is doubly stochastic, then with $\mathbf{x}_0 = \mathbf{r}_0$ we have

$$\sum_i x_{i,k} = \sum_i r_{i,k}, \forall k \geq 0$$

and, if the graph is connected

$$x_{i,\infty} \to \theta^\star_\infty = \frac{1}{m} \sum_i r_{i,\infty},\ \forall i \in \mathcal{V}$$

[4] Refer to (Zhu and MartÃnez, 2010)

Distributed Subgradient Methods: A seminal work

- DSM Algorithm (Nedic and Ozdaglar, 2009)

$$\underbrace{x_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_{j,k}}_{\text{average consensus}} - \underbrace{\gamma_k \cdot s_{i,k}}_{\text{gradient search}}$$

  - γ_k is the stepsize chosen by agents at time k,
  - $s_{i,k} \in \partial f_i(x_{i,k})$ is the subgradient of f_i evaluated at $x_{i,k}$,

- Convergence Result for $\gamma_k \equiv \gamma$ (Yuan et al., 2013)

$$\max\{\text{Disagreement, Optimality Gap}\} \leq \mathcal{O}(1/k) + \mathcal{O}(\gamma)$$

  - steady state error[5] $\mathcal{O}(\gamma)$,
  - decaying stepsize for exact optimum seeking,
  - bounded (sub)gradient (even for smooth f_i: $\|\nabla f_i\| < C$).

[5] $\mathcal{O}(\cdot)$ denotes the order of magnitude

Distributed Subgradient Methods: A seminal work

- DSM Algorithm (Nedic and Ozdaglar, 2009)

$$\underbrace{x_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_{j,k}}_{\text{average consensus}} - \underbrace{\gamma_k \cdot s_{i,k}}_{\text{gradient search}}$$

  - γ_k is the stepsize chosen by agents at time k,
  - $s_{i,k} \in \partial f_i(x_{i,k})$ is the subgradient of f_i evaluated at $x_{i,k}$,

- Convergence Result for $\gamma_k \equiv \gamma$ (Yuan et al., 2013)

$$\max\{\text{Disagreement}, \text{Optimality Gap}\} \leq \mathcal{O}(1/k) + \mathcal{O}(\gamma)$$

  - steady state error[5] $\mathcal{O}(\gamma)$,
  - decaying stepsize for exact optimum seeking,
  - bounded (sub)gradient (even for smooth f_i: $\|\nabla f_i\| < C$).

[5] $\mathcal{O}(\cdot)$ denotes the order of magnitude

Other distributed algorithms[6]

- Consensus-based (*Primal-only*)
  - Dual Averaging (Duchi et al., 2012)
  - Diffusion Strategy (Chen and Sayed, 2012)
  - Newton-Raphson Consensus (Zanella et al., 2011)
  - Fast Distributed Gradient (Jakovetic et al., 2014)
  - Stochastic Gradient Push (Nedic and Olshevsky, 2014)

  pros: easy to analyze even for dynamic networks
  cons: steady-sate error; decaying stepsize $\Rightarrow \mathcal{O}(\frac{\ln k}{k})$

- Dual-decomposition-based (*Primal-Dual*)
  - D-ADMM (Wei and Ozdaglar, 2012; Mota et al., 2013; Shi et al., 2014); IC-ADMM (Chang et al., 2015), ADMM$^+$ (Bianchi and Hachem, 2014), DLM (Ling et al., 2015)
  - Augmented Lagrangian Method (Wang and Elia, 2011; Gharesifard and Cortes, 2014)
  - Primal-Dual Method: EXTRA, PG-EXTRA (Shi et al., 2015a,b)

  pros: no steady-state error; constant stepsize $\Rightarrow \mathcal{O}(\frac{1}{k})$ or even $\mathcal{O}(\lambda^k)$
  cons: difficult to analyze for dynamic networks

[6] Refer to (Nedić et al., 2018) for a recent comprehensive survey

Other distributed algorithms[6]

- Consensus-based (*Primal-only*)
  - Dual Averaging (Duchi et al., 2012)
  - Diffusion Strategy (Chen and Sayed, 2012)
  - Newton-Raphson Consensus (Zanella et al., 2011)
  - Fast Distributed Gradient (Jakovetic et al., 2014)
  - Stochastic Gradient Push (Nedic and Olshevsky, 2014)

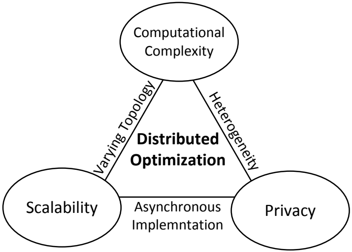
  pros: easy to analyze even for dynamic networks
  cons: steady-sate error; decaying stepsize $\Rightarrow \mathcal{O}(\frac{\ln k}{k})$
- Dual-decomposition-based (*Primal-Dual*)
  - D-ADMM (Wei and Ozdaglar, 2012; Mota et al., 2013; Shi et al., 2014); IC-ADMM (Chang et al., 2015), ADMM$^+$ (Bianchi and Hachem, 2014), DLM (Ling et al., 2015)
  - Augmented Lagrangian Method (Wang and Elia, 2011; Gharesifard and Cortes, 2014)
  - Primal-Dual Method: EXTRA, PG-EXTRA (Shi et al., 2015a,b)

  pros: no steady-state error; constant stepsize $\Rightarrow \mathcal{O}(\frac{1}{k})$ or even $\mathcal{O}(\lambda^k)$
  cons: difficult to analyze for dynamic networks

[6] Refer to (Nedić et al., 2018) for a recent comprehensive survey

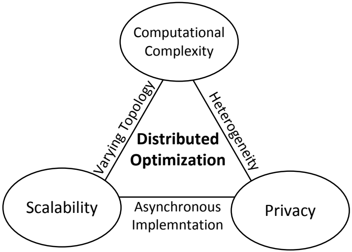
Objectives and Challenges

- Objectives
  - exact optimal solution
  - fast convergence rates
  - general networks
    - asynchronous
    - directed
    - ...
- Challenges
  - varying topology
  - asynchrony
  - heterogeneity
    - uncoordinated stepsize
    - directed graph



Objectives and Challenges

- Objectives
  - exact optimal solution
  - fast convergence rates
  - general networks
    - asynchronous
    - directed
    - ...
- Challenges
  - varying topology
  - asynchrony
  - heterogeneity
    - uncoordinated stepsize
    - directed graph



Outline

Algorithm Development

- Recalling the DSM algorithm for smooth functions[7]

$$\underbrace{\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k}_{\text{average consensus}} - \underbrace{\gamma \mathbf{g}(\mathbf{x}_k)}_{\text{gradient search}},$$

  - where $\mathbf{g}(\mathbf{x}_k) := [\nabla f_1(x_1), \nabla f_2(x_2), ..., \nabla f_m(x_m)]^T$
- Limit point analysis ($\mathbf{W}$ doubly stochastic $\Rightarrow (\mathbf{I} - \mathbf{W})\mathbf{1} = 0$):

$$\mathbf{x}_\infty \to \theta \mathbf{1} \;\Rightarrow\; 0 = (\mathbf{W} - \mathbf{I})\mathbf{x}_\infty = \gamma \mathbf{g}(\mathbf{x}_\infty) \neq 0$$

  - otherwise we have $\nabla f_i(\theta) = 0, \forall i = 1, 2, ..., m$
- essentially not able to reach consensus!

[7] Here, we consider a constant stepsize for simplicity.

Algorithm Development

- Recalling the DSM algorithm for smooth functions[7]

$$\underbrace{\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k}_{\text{average consensus}} - \underbrace{\gamma \mathbf{g}(\mathbf{x}_k)}_{\text{gradient search}},$$

  - where $\mathbf{g}(\mathbf{x}_k) := [\nabla f_1(x_1), \nabla f_2(x_2), ..., \nabla f_m(x_m)]^T$

- Limit point analysis ($\mathbf{W}$ doubly stochastic $\Rightarrow (\mathbf{I} - \mathbf{W})\mathbf{1} = 0$):

$$\mathbf{x}_\infty \to \theta\mathbf{1} \;\Rightarrow\; 0 = (\mathbf{W} - \mathbf{I})\mathbf{x}_\infty = \gamma \mathbf{g}(\mathbf{x}_\infty) \neq 0$$

  - otherwise we have $\nabla f_i(\theta) = 0, \forall i = 1, 2, ..., m$

- essentially not able to reach consensus!

[7] Here, we consider a constant stepsize for simplicity.

Algorithm Development

- Replacing with the *ideal* average of gradients

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma \cancel{\mathbf{g}(\mathbf{x}_k)} \rightarrow \frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{g}(\mathbf{x}_k) = \mathbf{1}\frac{1}{m}\sum_i \nabla f_i(x_{i,k}),$$

  - where $\frac{\mathbf{1}\mathbf{1}^T}{m} := [\frac{1}{m}]$ is the average matrix.
- Limit point analysis ($\mathbf{W}$ doubly stochastic $\Rightarrow (\mathbf{I}-\mathbf{W})\mathbf{1} = 0$):

$$\mathbf{x}_\infty \rightarrow \theta\mathbf{1} \;\Rightarrow\; 0 = (\mathbf{W}-\mathbf{I})\mathbf{x}_\infty = \gamma\frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{g}(\mathbf{x}_\infty) \;\Rightarrow\; \nabla F(\theta) = 0$$

- We are now able to reach consensus!
- But $\frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{g}(\mathbf{x}_k)$ not immediately available, how to obtain?

Algorithm Development

- Replacing with the *ideal* average of gradients

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma \cancelto{\frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{g}(\mathbf{x}_k) = \mathbf{1}\frac{1}{m}\sum_i \nabla f_i(x_{i,k})}{\mathbf{g}(\mathbf{x}_k)},$$

  - where $\frac{\mathbf{1}\mathbf{1}^T}{m} := [\frac{1}{m}]$ is the average matrix.

- Limit point analysis ($\mathbf{W}$ doubly stochastic $\Rightarrow (\mathbf{I} - \mathbf{W})\mathbf{1} = 0$):

$$\mathbf{x}_\infty \to \theta\mathbf{1} \;\Rightarrow\; 0 = (\mathbf{W} - \mathbf{I})\mathbf{x}_\infty = \gamma\frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{g}(\mathbf{x}_\infty) \;\Rightarrow\; \nabla F(\theta) = 0$$

- We are now able to reach consensus!
- But $\frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{g}(\mathbf{x}_k)$ not immediately available, how to obtain?

Algorithm Development

- Replacing with the *pseudo* average of gradients

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma \cancelto{\mathbf{y}_k}{\frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{g}(\mathbf{x}_k)} \quad ,$$

  – where $\mathbf{y}_k$ is the **surrogate** of the average of gradients.

- Resorting to Dynamic Average Consensus:

$$\mathbf{y}_{k+1} = \mathbf{W}\mathbf{y}_k + \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k)$$

- Average gradient tracking:

$$\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0) \quad \Rightarrow \quad y_{i,\infty} = \frac{1}{m}\mathbf{1}^T\mathbf{g}(\mathbf{x}_\infty), \ i \in \mathcal{V}$$

  – when $k \to \infty$, $\mathbf{y}_k$ essentially **tracks the average** of gradients.

Algorithm Development

- Replacing with the *pseudo* average of gradients

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma \underbrace{\frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{g}(\mathbf{x}_k)}_{\mathbf{y}_k},$$

  - where $\mathbf{y}_k$ is the **surrogate** of the average of gradients.

- Resorting to Dynamic Average Consensus:

$$\mathbf{y}_{k+1} = \mathbf{W}\mathbf{y}_k + \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k)$$

- Average gradient tracking:

$$\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0) \quad \Rightarrow \quad y_{i,\infty} = \frac{1}{m}\mathbf{1}^T\mathbf{g}(\mathbf{x}_\infty),\ i \in \mathcal{V}$$

  - when $k \to \infty$, $\mathbf{y}_k$ essentially **tracks the average** of gradients.

Algorithm and Implementation

AugDGM Algorithm[7]

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma \mathbf{y}_k$$
$$\mathbf{y}_{k+1} = \mathbf{W}\mathbf{y}_k + \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k),$$

- $\mathbf{y}_k$ is the auxiliary variable **tracking the average** of the gradients.

1. **Initialization**: $\forall$ agent $i \in \mathcal{V}$: $x_{i,0}$ randomly assigned; $y_{i,0} = \nabla f_i(x_{i,0})$.
2. **Local Optimization**: $\forall$ agent $i \in \mathcal{V}$, computes:
$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} x_{j,k} - \gamma \cdot y_{i,k}$$
3. **Dynamic Average Consensus**: $\forall$ agent $i \in \mathcal{V}$, computes:
$$y_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij} y_{j,k} + \nabla f_i(x_{i,k+1}) - \nabla f_i(x_{i,k})$$
4. Set $k \to k+1$ and go to Step 2.

[7] More general form can be found in (Xu et al., 2015)

Convergence Analysis: Preliminary

- Notations
  - $\mathbf{x}^\star$: the optimal solution
  - $\bar{\mathbf{x}} = \frac{\mathbf{1}\mathbf{1}^T}{m}\mathbf{x}$ (average), $\tilde{\mathbf{x}} = (\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{m})\mathbf{x}$ (disagreement)
  - ρ: the spectral radius of a given matrix
- Properties of cost functions
  - μ-strongly convex:

$$\left\|\psi(\mathbf{v}) - \psi(\mathbf{v}')\right\| \geq \mu \left\|\mathbf{v} - \mathbf{v}'\right\|, \forall \mathbf{v}, \mathbf{v}' \in \mathcal{R}^m$$

  - L-smooth:

$$\left\|\nabla\psi(\mathbf{v}) - \nabla\psi(\mathbf{v}')\right\| \leq L \left\|\mathbf{v} - \mathbf{v}'\right\|, \forall \mathbf{v}, \mathbf{v}' \in \mathcal{R}^m$$

Convergence Analysis[8]

- Assumptions
  - Cost functions $\{f_i\}$: μ-strongly convex, L-smooth
  - Weight Matrix:
    $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$, $\mathbf{W}\mathbf{1} = \mathbf{1}$ and $\rho_W := \rho\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) < 1$
  - There exists a solution to the problem

Theorem (Linear Rate for AugDGM)

Let $\{\mathbf{x}_k, \mathbf{y}_k\}_{k\geq 0}$ be the iterates generated by AugDGM with $\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0)$. Let $\kappa = L/\mu$ and suppose the above Assumptions hold. Then, if

$$\gamma < \frac{(1-\rho_W)^2}{(1+\sqrt{\kappa+3})L},$$

the residuals $\|\bar{\mathbf{x}}_k - \mathbf{x}^\star\|$ and $\|\tilde{\mathbf{x}}_k\|$ converge **linearly** *to zero.*

[8] Refer to (Xu et al., 2015, 2018b) for more details.

Convergence Analysis[8]

- Assumptions
  - Cost functions $\{f_i\}$: μ-strongly convex, L-smooth
  - Weight Matrix:
    $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$, $\mathbf{W}\mathbf{1} = \mathbf{1}$ and $\rho_W := \rho\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) < 1$
  - There exists a solution to the problem

Theorem (Linear Rate for AugDGM)

Let $\{\mathbf{x}_k, \mathbf{y}_k\}_{k\geq 0}$ be the iterates generated by AugDGM with $\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0)$. Let $\kappa = L/\mu$ and suppose the above Assumptions hold. Then, if

$$\gamma < \frac{(1-\rho_W)^2}{(1+\sqrt{\kappa+3})L},$$

the residuals $\|\bar{\mathbf{x}}_k - \mathbf{x}^\star\|$ and $\|\tilde{\mathbf{x}}_k\|$ converge **linearly** *to zero.*

[8] Refer to (Xu et al., 2015, 2018b) for more details.

A Sensor Fusion Example

- Overall loss function

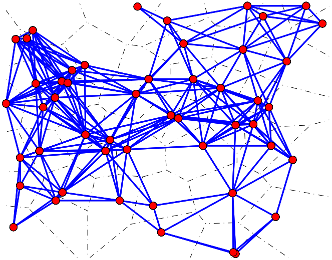
$$F(\theta) = \sum_{i=1}^{m} \|\mathbf{z}_i - \mathbf{M}_i\theta\|^2$$

  - $\theta \in \mathcal{R}^d$: the unknown parameter
  - $\mathbf{M}_i \in \mathcal{R}^{s\times d}$: measurement matrix
  - $\mathbf{z}_i \in \mathcal{R}^s$: the observation of sensor i

- Metropolis-Hastings protocol

$$w_{ij} = \begin{cases} \frac{1}{\max\{d_i, d_j\}}, & \text{if } (i,j) \in \mathcal{E} \\ 1 - \sum_{j\in\mathcal{N}_i} w_{ij}, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases}$$

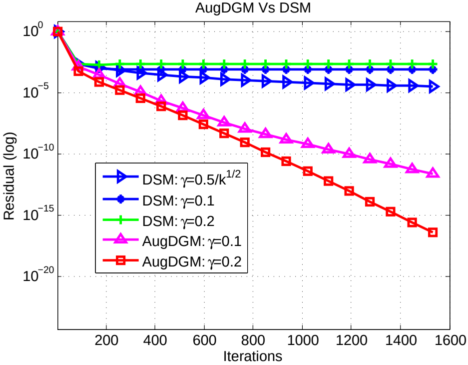
  - d_i: the degree of node i.



A random network of 50 nodes

Performance Evaluation

Parameter Setting: $d = 4, s = 1$; M_i: a unit uniform distribution; Gaussian Noise: $\mathcal{N}(0, 0.1)$

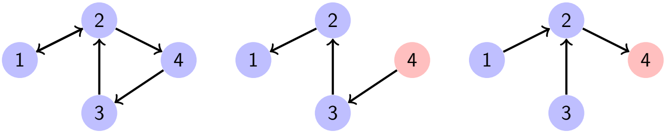


Residual ($res = \frac{\|\mathbf{x}_k - \mathbf{x}^\star\|^2}{\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}$) Vs. Iterations

Extension to Directed Networks

- Extended to directed networks by graph splitting

$$\mathcal{G}_W \;\Rightarrow\; \mathcal{G}_R \;\oplus\; \mathcal{G}_C$$



(a) $\mathcal{G}_W$ (b) $\mathcal{G}_R$ (c) $\mathcal{G}_C$

  - **R**: Row-stochastic matrix; **C**: Column-stochastic matrix

Push-Pull Algorithm

$$\mathbf{x}_{k+1} = \mathbf{R}\mathbf{x}_k - \gamma \mathbf{y}_k$$
$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{y}_k + \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k),$$

- $\mathbf{x}_k$: the decision **pulled** from the neighbors for average consensus
- $\mathbf{y}_k$: the gradient **pushed** to the neighbors for gradient tracking.

Convergence Analysis

- Assumptions
  - Cost functions $\{f_i\}$: μ-strongly convex, L-smooth
  - The subgraph $\mathcal{G}_R$ and $\mathcal{G}_{C^T}$[9]
    - each contains at least a spanning tree, i.e., $\rho_R = \rho(\mathbf{R} - \frac{\mathbf{1}\mathbf{1}^T}{m}) < 1,\ \rho_C = \rho(\mathbf{C} - \frac{\mathbf{1}\mathbf{1}^T}{m}) < 1$
    - have a common root, i.e., information flow is not blocked
  - There exists a solution to the problem

Theorem (Linear Rate for Push-Pull)

Let $\{\mathbf{x}_k, \mathbf{y}_k\}_{k\geq 0}$ be the iterates generated by Push-Pull with $\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0)$. Let $\kappa = L/\mu$ and suppose the above Assumptions hold. Then, if

$$\gamma < \frac{(1-\rho_R)(1-\rho_C)}{\phi(\kappa, \mathbf{R}, \mathbf{C})L},$$

the residuals $\|\bar{\mathbf{x}}_k - \mathbf{x}^\star\|$ and $\|\tilde{\mathbf{x}}_k\|$ converge **linearly** *to zero.*

[9] $\mathcal{G}_{C^T} = \mathcal{G}_C$ with edges reversed; Refer to (Pu et al., 2020) for more detials.

Convergence Analysis

- Assumptions
  - Cost functions $\{f_i\}$: μ-strongly convex, L-smooth
  - The subgraph $\mathcal{G}_R$ and $\mathcal{G}_{C^T}$[9]
    - each contains at least a spanning tree, i.e., $\rho_R = \rho(\mathbf{R} - \frac{\mathbf{1}\mathbf{1}^T}{m}) < 1, \; \rho_C = \rho(\mathbf{C} - \frac{\mathbf{1}\mathbf{1}^T}{m}) < 1$
    - have a common root, i.e., information flow is not blocked
  - There exists a solution to the problem

Theorem (Linear Rate for Push-Pull)

Let $\{\mathbf{x}_k, \mathbf{y}_k\}_{k\geq 0}$ be the iterates generated by Push-Pull with $\mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0)$. Let $\kappa = L/\mu$ and suppose the above Assumptions hold. Then, if

$$\gamma < \frac{(1-\rho_R)(1-\rho_C)}{\phi(\kappa, \mathbf{R}, \mathbf{C})L},$$

the residuals $\|\bar{\mathbf{x}}_k - \mathbf{x}^\star\|$ and $\|\tilde{\mathbf{x}}_k\|$ converge **linearly** *to zero.*

[9] $\mathcal{G}_{C^T} = \mathcal{G}_C$ with edges reversed; Refer to (Pu et al., 2020) for more detials.

Outline

Equivalent Primal-Dual Problems

- Recalling the orginal DOP problem as follows

$$\min_{\mathbf{x}\in\mathcal{R}^m} f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i)$$
$$s.t.\ x_i = x_j,\ \forall i,j \in \mathcal{V}$$

  - $\mathbf{x} = [x_1, x_2, ...x_m]^T$: the local estimates of agents.

- Equivalent [10] to the (primal) optimal consensus problem

$$\min_{\mathbf{x}\in\mathcal{R}^m} f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i) \qquad \text{(OCP)}$$
$$s.t.\ \ (\mathbf{I}-\mathbf{W})\mathbf{x} = 0,$$

  - $\mathbf{W}$: the weight matrix associated with the network

[10] If the graph is strongly connected such that $\mathbf{null}(\mathbf{I}-\mathbf{W}) = \mathbf{span}(\mathbf{1})$.

Equivalent Primal-Dual Problems

- The Lagrange dual problem is depicted as follows

$$\max_{\mathbf{y}' \in \mathcal{R}^m} \min_{\mathbf{x} \in \mathcal{R}^m} \left\{ f(\mathbf{x}) + \mathbf{y}'^T (\mathbf{I} - \mathbf{W}) \mathbf{x} \right\}$$

  - $\mathbf{y}' = [y'_1, y'_2, ..., y'_m]^T$: the Lagrange multiplier or dual variables.
- Let $\mathbf{y} = (\mathbf{I} - \mathbf{W})\mathbf{y}'$. The above problem becomes

$$\max_{\mathbf{y} \in \mathcal{R}^m} \min_{\mathbf{x} \in \mathcal{R}^m} \{ f(\mathbf{x}) + \langle \mathbf{y}, \mathbf{x} \rangle \} = \max_{\mathbf{y} \in \mathcal{R}^m} -f^*(-\mathbf{y})$$

  - f^*: the convex conjugate of the function f.
- $\mathbf{1}^T \mathbf{y} = \mathbf{1}^T (\mathbf{I} - \mathbf{W}) \mathbf{y}' = 0 \Rightarrow$ the (dual) optimal exchange problem

$$\min_{\mathbf{y} \in \mathcal{R}^m} f^*(\mathbf{y}) = \sum_{i=1}^{m} f_i^*(y_i) \quad \text{(OEP[11])}$$
$$s.t. \quad \mathbf{1}^T \mathbf{y} = 0,$$

  - $\mathbf{y} = [y_1, y_2, ..., y_m]^T$: the introduced dual variables.

[11] Refer to (Xu et al., 2018c) for more details.

Equivalent Primal-Dual Problems

- The Lagrange dual problem is depicted as follows

$$\max_{\mathbf{y}'\in\mathcal{R}^m}\min_{\mathbf{x}\in\mathcal{R}^m}\left\{f(\mathbf{x})+\mathbf{y}'^T(\mathbf{I}-\mathbf{W})\mathbf{x}\right\}$$

  - $\mathbf{y}'=[y_1',y_2',...,y_m']^T$: the Lagrange multiplier or dual variables.
- Let $\mathbf{y}=(\mathbf{I}-\mathbf{W})\mathbf{y}'$. The above problem becomes

$$\max_{\mathbf{y}\in\mathcal{R}^m}\min_{\mathbf{x}\in\mathcal{R}^m}\{f(\mathbf{x})+\langle\mathbf{y},\mathbf{x}\rangle\}=\max_{\mathbf{y}\in\mathcal{R}^m}-f^*(-\mathbf{y})$$

  - f^*: the convex conjugate of the function f.
- $\mathbf{1}^T\mathbf{y}=\mathbf{1}^T(\mathbf{I}-\mathbf{W})\mathbf{y}'=0\Rightarrow$ the (dual) optimal exchange problem

$$\min_{\mathbf{y}\in\mathcal{R}^m} f^*(\mathbf{y})=\sum_{i=1}^m f_i^*(y_i) \quad \text{s.t.}\ \ \mathbf{1}^T\mathbf{y}=0, \tag{OEP[11]}$$

  - $\mathbf{y}=[y_1,y_2,...,y_m]^T$: the introduced dual variables.

[11] Refer to (Xu et al., 2018c) for more details.

Equivalent Primal-Dual Problems

- The Lagrange dual problem is depicted as follows

$$\max_{\mathbf{y}' \in \mathcal{R}^m} \min_{\mathbf{x} \in \mathcal{R}^m} \left\{ f(\mathbf{x}) + \mathbf{y}'^T (\mathbf{I} - \mathbf{W}) \mathbf{x} \right\}$$

  - $\mathbf{y}' = [y'_1, y'_2, ..., y'_m]^T$: the Lagrange multiplier or dual variables.
- Let $\mathbf{y} = (\mathbf{I} - \mathbf{W})\mathbf{y}'$. The above problem becomes

$$\max_{\mathbf{y} \in \mathcal{R}^m} \min_{\mathbf{x} \in \mathcal{R}^m} \{ f(\mathbf{x}) + \langle \mathbf{y}, \mathbf{x} \rangle \} = \max_{\mathbf{y} \in \mathcal{R}^m} -f^*(-\mathbf{y})$$

  - f^*: the convex conjugate of the function f.
- $\mathbf{1}^T\mathbf{y} = \mathbf{1}^T(\mathbf{I} - \mathbf{W})\mathbf{y}' = 0 \Rightarrow$ the (dual) optimal exchange problem

$$\min_{\mathbf{y} \in \mathcal{R}^m} f^*(\mathbf{y}) = \sum_{i=1}^{m} f_i^*(y_i) \qquad \text{(OEP[11])}$$
$$s.t. \ \ \mathbf{1}^T\mathbf{y} = 0,$$

  - $\mathbf{y} = [y_1, y_2, ..., y_m]^T$: the introduced dual variables.

[11] Refer to (Xu et al., 2018c) for more details.

Algorithm and Implementation

ID-FBBS Algorithm[12]

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma(\mathbf{g}(\mathbf{x}_k) + \mathbf{y}_k)$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{1}{\gamma}(\mathbf{I} - \mathbf{W})\mathbf{x}_{k+1},$$

– $\mathbf{y}_k$ is the dual variable whose sum is **maintained at zero**.

1. **Initialization**: $\forall$ agent $i \in \mathcal{V}$: $x_{i,0}$ randomly assigned; $\sum_{i\in\mathcal{V}} y_{i,0} = 0$.
2. **Primal Update**: $\forall$ agent $i \in \mathcal{V}$, computes:

$$x_{i,k+1} = \sum_{j\in\mathcal{N}_i} w_{ij} x_{j,k} - \gamma\big(g_i(x_{i,k}) + y_{i,k}\big)$$

3. **Dual Update**: $\forall$ agent $i \in \mathcal{V}$, computes:

$$y_{i,k+1} = y_{j,k} + \frac{1}{\gamma}\sum_{j\in\mathcal{N}_i} w_{ij}\big(x_{i,k+1} - x_{j,k+1}\big)$$

4. Set $k \to k+1$ and go to Step 2.

[12] More general form can be found in (Xu et al., 2016)

Convergence Analysis

- Assumptions
  - Cost functions $\{f_i\}$: L-smooth
  - Weight Matrix: $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T, \mathbf{W}\mathbf{1} = \mathbf{1}$, $\rho\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) < 1$, and $\mathbf{W} = \mathbf{W}^T, \mathbf{W} > 0$
  - There exists a saddle point to the OCP-OEP problem

Theorem (Sublinear rate for ID-FBBS)

Let $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k\geq 0}$ be the iterates generated by ID-FBBS with $\mathbf{1}^T\mathbf{y}_0 = 0$. Suppose the above Assumptions hold. Then, if

$$\gamma < \frac{\lambda_{\min}(\mathbf{W})}{L},$$

- *it will converge to an optimal solution pair $(\mathbf{x}^\star, \mathbf{y}^\star)$ where $\mathbf{x}^\star$ solves the OCP problem while $\mathbf{y}^\star$ solves the OEP problem.*
- *and converge at a rate[13] of $\mathcal{O}(\frac{1}{k})$.*

[13] holds for *non-smooth* functions (Xu et al., 2018a); Linear rate (Shi et al., 2015a)

Convergence Analysis

- Assumptions
  - Cost functions $\{f_i\}$: L-smooth
  - Weight Matrix: $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T, \mathbf{W}\mathbf{1} = \mathbf{1}$, $\rho\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) < 1$, and $\mathbf{W} = \mathbf{W}^T, \mathbf{W} > 0$
  - There exists a saddle point to the OCP-OEP problem

Theorem (Sublinear rate for ID-FBBS)

Let $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k\geq 0}$ be the iterates generated by ID-FBBS with $\mathbf{1}^T\mathbf{y}_0 = 0$. Suppose the above Assumptions hold. Then, if

$$\gamma < \frac{\lambda_{\min}(\mathbf{W})}{L},$$

- *it will converge to an optimal solution pair $(\mathbf{x}^\star, \mathbf{y}^\star)$ where $\mathbf{x}^\star$ solves the OCP problem while $\mathbf{y}^\star$ solves the OEP problem.*
- *and converge at a rate*[13] *of $\mathcal{O}(\frac{1}{k})$.*

[13] holds for *non-smooth* functions (Xu et al., 2018a); Linear rate (Shi et al., 2015a)

Connections to Existing Algorithms

- Recalling the ID-FBBS Algorithm

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma(\mathbf{g}(\mathbf{x}_k) + \mathbf{y}_k) \quad \text{(a)}$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{1}{\gamma}(\mathbf{I} - \mathbf{W})\mathbf{x}_{k+1}, \quad \text{(b)}$$

- Setting $y_0 = 0$, summing (b) and substituting into (a) yields

$$\underbrace{\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma\mathbf{g}(\mathbf{x}_k)}_{DSM} - \underbrace{\sum_{i=0}^{k}(\mathbf{I} - \mathbf{W})\mathbf{x}_i}_{Correction},$$

  - equivalent[16] to EXTRA with $\mathbf{W} = \tilde{\mathbf{W}} = \frac{\mathbf{I}+\mathbf{W}'}{2}$ in $\mathbf{x}$-update,
  - $\tilde{\mathbf{W}}, \mathbf{W}'$ are two weight matrices of EXTRA (Shi et al., 2015a).

[16] Refer to (Xu et al., 2016, 2018a) for more details

Connections to Existing Algorithms

- Recalling the ID-FBBS Algorithm

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma(\mathbf{g}(\mathbf{x}_k) + \mathbf{y}_k) \tag{a}$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{1}{\gamma}(\mathbf{I} - \mathbf{W})\mathbf{x}_{k+1}, \tag{b}$$

- Let $\gamma\mathbf{y}_k = \sqrt{\mathbf{I} - \mathbf{W}}\mathbf{y}'_k$, the above algorithm can be rewritten as

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma\mathbf{g}(\mathbf{x}_k) - \sqrt{\mathbf{I} - \mathbf{W}}\mathbf{y}'_k$$
$$\mathbf{y}'_{k+1} = \mathbf{y}'_k + \sqrt{\mathbf{I} - \mathbf{W}}\mathbf{x}_{k+1}$$

- Equivalent to applying the Arrow-Hurwicz-Uzawa Method[17]

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}'_k) \\ \mathbf{y}'_{k+1} = \mathbf{y}'_k + \gamma\nabla_{\mathbf{y}'} L(\mathbf{x}_{k+1}, \mathbf{y}') \end{cases}$$

  - where $L(\mathbf{x}, \mathbf{y}') = f(\mathbf{x}) + \frac{1}{\gamma}\mathbf{x}^T\sqrt{\mathbf{I} - \mathbf{W}}\mathbf{y}' + \frac{1}{2\gamma}\mathbf{x}^T(\mathbf{I} - \mathbf{W})\mathbf{x}$

[17] Refer to (Xu et al., 2016, 2018a) for more details

Connections to Existing Algorithms

- Taking the augmented Lagrangian as follows:

$$L(\mathbf{x}, \mathbf{y}') = f(\mathbf{x}) + \frac{1}{\gamma}\mathbf{x}^T(\mathbf{I} - \mathbf{W})\mathbf{y}' + \frac{1}{2\gamma}\mathbf{x}^T(\mathbf{I} - \mathbf{W}^2)\mathbf{x},$$

  Applying the Arrow-Hurwicz-Uzawa Method leads to

$$\mathbf{x}_{k+1} = \mathbf{W}^2\mathbf{x}_k - \gamma\mathbf{g}(\mathbf{x}_k) - (\mathbf{I} - \mathbf{W})\mathbf{y}'_k \quad \text{(c)}$$
$$\mathbf{y}'_{k+1} = \mathbf{y}'_k + (\mathbf{I} - \mathbf{W})\mathbf{x}_{k+1} \quad \text{(d)}$$

- Evaluating (c) at $k+1$ and k, respectively and eliminating $\mathbf{y}'$ using (d), simple calculation gives

$$\mathbf{x}_{k+2} - \mathbf{W}\mathbf{x}_{k+1} = \mathbf{W}(\mathbf{x}_{k+1} - \mathbf{W}\mathbf{x}_k) + \gamma(\mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k))$$

  Let $\gamma\mathbf{y}_{k+1} = \mathbf{x}_{k+2} - \mathbf{W}\mathbf{x}_{k+1}$. Then, we recover

$$\text{the original AugDGM} \begin{cases} \mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma\mathbf{y}_k \\ \mathbf{y}_{k+1} = \mathbf{W}\mathbf{y}_k + \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k). \end{cases}$$

Outline

A Unified Algorithm

A unified algorithm[18]

$$\mathbf{x}^{k+1} = \mathbf{A}\mathbf{x}^k - \gamma \mathbf{B}\mathbf{g}(\mathbf{x}^k) - \mathbf{y}^k,$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{C}\mathbf{x}^{k+1},$$

– where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are three weight matrices to be properly defined.

The above unified algorithm subsumes many existing algorithms.

Algorithm	A	B	C
ID-FBBS/EXTRA	$\frac{1}{2}(\mathbf{I}+\mathbf{W})$	$\mathbf{I}$	$\frac{1}{2}(\mathbf{I}-\mathbf{W})$
NIDS/Exact Diffusion	$\frac{1}{2}(\mathbf{I}+\mathbf{W})$	$\frac{1}{2}(\mathbf{I}+\mathbf{W})$	$\frac{1}{2}(\mathbf{I}-\mathbf{W})$
AugDGM/NEXT	$\mathbf{W}^2$	$\mathbf{W}^2$	$(\mathbf{I}-\mathbf{W})^2$
DIGing/Harnessing	$\mathbf{W}^2$	$\mathbf{I}$	$(\mathbf{I}-\mathbf{W})^2$

[18] More general form can be found in (Xu et al., 2020b)

Sublinear Convergence Rate

Let $\mathbb{S}^m$ be the set of $m \times m$ symmetric matrices.

- Assumptions
  - Cost function $\{f_i\}$: L-smooth;
  - Weight Matrix:
    - i) $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{S}^m$ and $\mathbf{C} \succeq 0$,
    - ii) $\mathbf{A} = \mathbf{B}$, $\mathbf{BC} = \mathbf{CB}$, $0 \preceq \mathbf{A} \preceq \mathbf{I} - \mathbf{C}$,
    - iii) $\mathbf{span}(\mathbf{1}) = \mathbf{null}(\mathbf{C}) \subseteq \mathbf{null}(\mathbf{I} - \mathbf{A})$.

Theorem (Sublinear rate for the unified algorithm)

Let $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k \geq 0}$ be the iterates generated by the above algorithm with $\mathbf{1}^T \mathbf{y}_0 = 0$. Suppose the above hold. Then, if $\gamma = \min\{\frac{1}{L}, \mathcal{O}(\sqrt{\eta})\}$, the algorithm converges at a sublinear rate of

$$\max\left\{\frac{L\left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2}{k+1}, \frac{1}{\sqrt{\eta(\mathbf{C})}} \frac{\left\|\mathbf{x}^0 - \mathbf{x}^\star\right\| \left\|\mathbf{g}(\mathbf{x}^\star)\right\|}{k+1}\right\},$$

where $\eta(\mathbf{C}) := \frac{\lambda_{\min}(\mathbf{C})}{\lambda_{\max}(\mathbf{C})}$ denotes the eigengap of the matrix $\mathbf{C}$.

Sublinear Convergence Rate

Let $\mathbb{S}^m$ be the set of $m \times m$ symmetric matrices.

- Assumptions
  - Cost function $\{f_i\}$: L-smooth;
  - Weight Matrix:
    - i) $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{S}^m$ and $\mathbf{C} \succeq 0$,
    - ii) $\mathbf{A} = \mathbf{B}$, $\mathbf{BC} = \mathbf{CB}$, $0 \preceq \mathbf{A} \preceq \mathbf{I} - \mathbf{C}$,
    - iii) $\mathbf{span}(\mathbf{1}) = \mathbf{null}(\mathbf{C}) \subseteq \mathbf{null}(\mathbf{I} - \mathbf{A})$.

Theorem (Sublinear rate for the unified algorithm)

Let $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k\geq 0}$ be the iterates generated by the above algorithm with $\mathbf{1}^T\mathbf{y}_0 = 0$. Suppose the above hold. Then, if $\gamma = \min\{\frac{1}{L}, \mathcal{O}(\sqrt{\eta})\}$, the algorithm converges at a sublinear rate of

$$\max\left\{\frac{L\left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2}{k+1}, \frac{1}{\sqrt{\eta(\mathbf{C})}}\frac{\left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|\left\|\mathbf{g}(\mathbf{x}^\star)\right\|}{k+1}\right\},$$

where $\eta(\mathbf{C}) := \frac{\lambda_{\min}(\mathbf{C})}{\lambda_{\max}(\mathbf{C})}$ denotes the eigengap of the matrix $\mathbf{C}$.

Some Observations

The convergence rate has the following structure[19]

$$\max\left\{\underbrace{\frac{L\left\|\mathbf{x}^0-\mathbf{x}^\star\right\|^2}{k+1}}_{\text{computation}},\ \underbrace{\frac{1}{\sqrt{\eta(\mathbf{C})}}\frac{\left\|\mathbf{x}^0-\mathbf{x}^\star\right\|\left\|\mathbf{g}(\mathbf{x}^\star)\right\|}{k+1}}_{\text{communication}}\right\}\overset{\mathbf{g}(\mathbf{x}^\star)=0}{\Longrightarrow}\underbrace{\mathcal{O}\left(\frac{L\left\|\mathbf{x}^0-\mathbf{x}^\star\right\|^2}{k+1}\right)}_{\text{centralized rate}}.$$

- $1/\sqrt{\eta}\approx$ the diameter of the network for simple networks, e.g., line graphs
- $\|\mathbf{g}(\mathbf{x}^\star)\|$ encodes the "heterogeneity" of functions; $\mathbf{g}(\mathbf{x}^\star)=0$ implies
  - **Case 1**: When all agents share common solution, e.g., the distribution of all local data sets are similar.
  - **Case 2**: When a spanning tree algorithm is employed, e.g, exact average of local data, e.g., local gradients.
- The algorithm reduces to the centralized one!

[19] Refer to (Xu et al., 2020a,b) for more details.

Linear Convergence Rate

Let $\mathbb{S}^m$ be the set of $m \times m$ symmetric matrices.

- Assumptions
  - Cost function $\{f_i\}$: L-smooth and μ-strongly convex;
  - Weight Matrix:
    - i) $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{S}^m$ and $\mathbf{C} \succeq 0$,
    - ii) $\mathbf{A} = \mathbf{B}$, $\mathbf{BC} = \mathbf{CB}$, $\mathbf{B}^2 \preceq \mathbf{I} - \mathbf{C}$,
    - iii) $\mathbf{span}(\mathbf{1}) = \mathbf{null}(\mathbf{C}) \subseteq \mathbf{null}(\mathbf{I} - \mathbf{A})$.

Theorem (Linear rate for the unified algorithm)

Let $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k\geq 0}$ be the iterates generated by the above algorithm with $\mathbf{1}^T\mathbf{y}_0 = 0$. Suppose the above Assumptions hold. Then, if $\gamma = \frac{2}{L+\mu}$, the algorithm converges at a linear rate of $\mathcal{O}(\sigma^k)$ with

$$\sigma = \max\left\{\left(\frac{\kappa-1}{\kappa+1}\right)^2, 1 - \lambda_{\min}(\mathbf{C})\right\},$$

where $\lambda_{\min}(\mathbf{C})$ denotes the connectivity of the graph.

Linear Convergence Rate

Let $\mathbb{S}^m$ be the set of $m \times m$ symmetric matrices.

- Assumptions
  - Cost function $\{f_i\}$: L-smooth and μ-strongly convex;
  - Weight Matrix:
    1. i) $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{S}^m$ and $\mathbf{C} \succeq 0$,
    2. ii) $\mathbf{A} = \mathbf{B}$, $\mathbf{BC} = \mathbf{CB}$, $\mathbf{B}^2 \preceq \mathbf{I} - \mathbf{C}$,
    3. iii) $\mathbf{span}(\mathbf{1}) = \mathbf{null}(\mathbf{C}) \subseteq \mathbf{null}(\mathbf{I} - \mathbf{A})$.

Theorem (Linear rate for the unified algorithm)

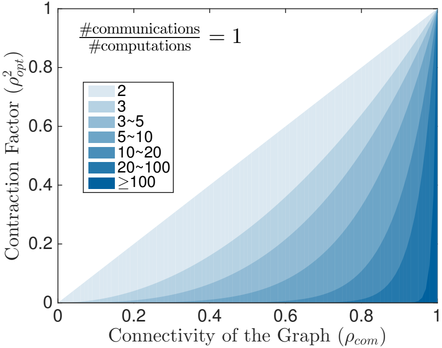
Let $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k\geq 0}$ be the iterates generated by the above algorithm with $\mathbf{1}^T\mathbf{y}_0 = 0$. Suppose the above Assumptions hold. Then, if $\gamma = \frac{2}{L+\mu}$, the algorithm converges at a linear rate of $\mathcal{O}(\sigma^k)$ with

$$\sigma = \max\left\{\left(\frac{\kappa-1}{\kappa+1}\right)^2, 1 - \lambda_{\min}(\mathbf{C})\right\},$$

where $\lambda_{\min}(\mathbf{C})$ denotes the connectivity of the graph.

Balancing Communication and Computation[20]

Set $\mathbf{A} = \mathbf{B} = \mathbf{I} - \mathbf{C} = \mathbf{W}^k$ and $\rho_{opt} = \frac{\kappa-1}{\kappa+1}$, $\rho_{com} = \rho(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{m})$



Perform similar as centralized counterparts

with finite number of inner consensus steps

[20] More details can be found in (Xu et al., 2020b)

Conclusion and Future Work

Summary

- Proposed a class of distributed algorithms that work for fixed, undirected or directed networks,
- Showed their basic convergence and the relationships between primal-only methods and primal-dual methods,
- Provided a unified algorithmic framework and showed the condition to achieving the "centralized" performance.

Conclusion and Future Work

Recommendation for future work

- Communication and Computation Trade-offs

$$\mathcal{O}(comm.)\ Vs.\ \mathcal{O}(comp.)$$

  - complexity, optimality, fundamental limits
- Extension and Generalization
  - constraints, general graphs, total asynchrony...
- Security and Privacy
  - robust and secure against malicious attacks
  - protect the data in optimization process
- Applications
  - UAVs, Internet of Things, Artificial Intelligence...

References I

Bianchi, P. and Hachem, W. (2014). A primal-dual algorithm for distributed optimization. In *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 4240–4245.

Chang, T.-H., Hong, M., and Wang, X. (2015). Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Trans. Signal Process*, 63(2):482–497.

Chen, J. and Sayed, A. H. (2012). Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process*, 60(8):4289–4305.

Duchi, J., Agarwal, A., and Wainwright, M. (2012). Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Autom. Control*, 57(3):592–606.

Gharesifard, B. and Cortes, J. (2014). Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Trans. Autom. Control*, 59(3):781–786.

Jakovetic, D., Xavier, J., and Moura, J. (2014). Fast distributed gradient methods. *IEEE Trans. Autom. Control*, 59(5):1131–1146.

Ling, Q., Shi, W., Wu, G., and Ribeiro, A. (2015). DLM: Decentralized linearized alternating direction method of multipliers. *IEEE Trans. Signal Process*, 63(15):4051–4064.

Mota, J., Xavier, J., Aguiar, P., and Puschel, M. (2013). D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Trans. Signal Process*, 61(10):2718–2723.

Nedic, A. and Olshevsky, A. (2014). Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *arXiv:1406.2075*.

Nedić, A., Olshevsky, A., and Rabbat, M. G. (2018). Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976.

Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48–61.

Olfati-Saber, R. and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 49(9):1520–1533.

References II

Pu, S., Shi, W., Xu, J., and Nedic, A. (2020). Push-pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, pages 1–1.

Shi, W., Ling, Q., Wu, G., and Yin, W. (2015a). EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966.

Shi, W., Ling, Q., Wu, G., and Yin, W. (2015b). A proximal gradient algorithm for decentralized composite optimization. *IEEE Trans. Signal Process*, 63(22):6013–6023.

Shi, W., Ling, Q., Yuan, K., Wu, G., and Yin, W. (2014). On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Trans. Signal Process*, 62(7):1750–1761.

Wang, J. and Elia, N. (2011). A control perspective for centralized and distributed convex optimization. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 3800–3805.

Wei, E. and Ozdaglar, A. E. (2012). Distributed alternating direction method of multipliers. In *Proceedings of IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 5445–5450.

Xu, J., Tian, Y., Sun, Y., and Scutari, G. (2020a). Accelerated primal-dual algorithms for distributed smooth convex optimization over networks. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Xu, J., Tian, Y., Sun, Y., and Scutari, G. (2020b). Distributed algorithms for composite optimization: Unified and tight convergence analysis. *arXiv preprint arXiv:2002.11534*.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. (2015). Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *Proceedings of 54th IEEE Conference on Decision and Control (CDC)*, pages 2055–2060.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. (2016). A forward-backward Bregman splitting scheme for regularized distributed optimization problems. In *Proceedings of 55th IEEE Conference on Decision and Control (CDC)*, pages 1093–1098.

References III

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. (2018a). A Bregman splitting scheme for distributed optimization over networks. *IEEE Transactions on Automatic Control, online available: https://arxiv.org/pdf/1608.08031.pdf*.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. (2018b). Convergence of asynchronous distributed gradient methods over stochastic networks. *IEEE Transactions on Automatic Control*, 63(2):434–448.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. (2018c). A dual splitting approach for distributed resource allocation with regularization. *IEEE Transactions on Control of Network Systems, accepted*.

Yuan, K., Ling, Q., and Yin, W. (2013). On the convergence of decentralized gradient descent. *arXiv preprint arXiv:1310.7063*.

Zanella, F., Varagnolo, D., Cenedese, A., Pillonetto, G., and Schenato, L. (2011). Newton-Raphson consensus for distributed convex optimization. In *Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5917–5922.

Zhu, M. and MartÃnez, S. (2010). Discrete-time dynamic average consensus. *Automatica*, 46(2):322 – 329.

Acknowledgment

This talk would not have been possible without the following excellent collaborators:

- Prof. Yeng Chai Soh, Prof. Lihua Xie from NTU;
- Prof. Shanying Zhu from SJTU;
- Prof. Angelia Nedich from ASU;
- Dr. Shi Pu from CUHK (SZ);
- Dr. Wei Shi from Princeton;
- Prof. Gesualdo Scutari, Ye Tian, Dr. Ying Sun from Purdue.

Q & A

http://jinmingxu.github.io

Master, PhD and Postdoc positions!