An Inexact Fenchel Dual Gradient Algorithm for Distributed Optimization

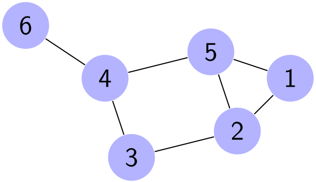
He Wang and Jie Lu

ShanghaiTech University

ICCA 2020

Distributed Optimization

- Undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:
  - Node set: $\mathcal{V} = \{1, \ldots, N\}$.
  - Link set: $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.



- Problem:

$$\begin{aligned} &\text{minimize}_{x \in \mathbb{R}^d} && \sum_{i \in \mathcal{V}} f_i(x) \\ &\text{subject to} && x \in \bigcap_{i \in \mathcal{V}} X_i \end{aligned}$$

  - Each node has local objective f_i and local constraint X_i.
  - Only local communications.

- Applications:
  - Wireless sensor network.
  - Cognitive radio network.
  - Large-scale machine learning.

Literature Review

- Unconstained optimization methods: DGD (Yuan, *et al.*, 2016), EXTRA (Shi, *et al.*, 2015) and DIGing (Nedić, *et al.*, 2017), etc.
- Constrained optimization methods: the projected subgradient algorithm (Nedić, *et al.*, 2010), PG-EXTRA (Shi, *et al.*, 2015), and
  - **Fenchel dual gradient (FDG) method** (Wu & Lu, 2019)
    - Apply a weighted gradient method to the Fenchel dual.
    - Highly scalable in terms of the network size.
    - Require to solve a constrained convex optimization problem per iteration.
- **Goal: Reduce the computational costs of FDG.**

Contribution

- An Inexact Fenchel Dual Gradient (IFDG) algorithm.
- A significant reduction in computational costs of FDG.
- An $O(1/k)$ convergence rate for strongly convex and smooth local objectives.
- A linear rate if the problem is unconstrained.
- Numerical simulations demonstrate the convergence performance of IFDG.

Problem Formulation

- Equivalent problem:

$$\text{minimize}_{\mathbf{x}\in\mathbb{R}^{Nd}} \quad F(\mathbf{x}) := \sum_{i\in\mathcal{V}} f_i(x_i) \qquad \textbf{(P1)}$$

$$\text{subject to} \quad x_i \in X_i,\ \forall i \in \mathcal{V} \text{ and } \mathbf{x} \in S,$$

  where $\mathbf{x} = (x_1^T, \ldots, x_N^T)^T \in \mathbb{R}^{Nd}$ and $S = \{\mathbf{x} \in \mathbb{R}^{Nd} : x_1 = \cdots = x_N\}$.

- Assumption 1: Each f_i, $i \in \mathcal{V}$ is strongly convex and smooth on X_i with convexity parameter $\mu_{f_i} > 0$ and smoothness parameter $L_{f_i} > 0$, i.e., $\forall x, y \in X_i$,

$$\mu_{f_i}\|y - x\|^2 \leq (\nabla f_i(y) - \nabla f_i(x))^T (y - x) \leq L_{f_i}\|y - x\|^2.$$

- Assumption 2: Each X_i, $i \in \mathcal{V}$ is a closed and convex set. In addition, rel int $\bigcap_{i\in\mathcal{V}} X_i \neq \emptyset$.

- Assumption 1 + Assumption 2 $\Longrightarrow$ A unique optimal solution.

Equivalent Fenchel Dual Problem

- The Fenchel dual problem is:

$$\begin{aligned} &\text{minimize}_{\mathbf{w}\in\mathbb{R}^{Nd}} \quad && D(\mathbf{w}) = \sum_{i\in\mathcal{V}} d_i(w_i) \qquad \textbf{(P2)} \\ &\text{subject to} && \mathbf{w} \in S^{\perp}, \end{aligned}$$

  where $d_i(w_i) := \sup_{x_i\in X_i} w_i^T x_i - f_i(x_i)$ and $S^{\perp} = \{\mathbf{w} \in \mathbb{R}^{Nd} : w_1 + \cdots + w_N = \mathbf{0}_d\}$.

- For each $i \in \mathcal{V}$, the gradient of d_i is $1/\mu_i$-Lipschitz continuous. Furthermore,

$$\nabla d_i(w_i) = \arg\max_{x\in X_i} w_i^T x - f_i(x), \ \forall i \in \mathcal{V}.$$

Fenchel Dual Gradient (FDG) Method

Initialization:

$$\mathbf{w}^0 \in S^\perp, \text{ or simply } \mathbf{w}^0 = \mathbf{0}_{Nd}.$$

Update: for any $k \geq 0$,

Primal update: $$\mathbf{x}^k = \arg\max_{\mathbf{x}\in X} (\mathbf{w}^k)^T\mathbf{x} - F(\mathbf{x}),$$

Dual update: $$\mathbf{w}^{k+1} = \mathbf{w}^k - \beta(H_{\mathcal{G}} \otimes I_d)\mathbf{x}^k,$$

where $X = X_1 \times X_2 \times \cdots \times X_N$ and $H_{\mathcal{G}} \in \mathbb{R}^{N\times N}$ is a symmetric and positive semidefinite matrix in the form of

$$[H_{\mathcal{G}}]_{ij} = \begin{cases} \sum_{s\in\mathcal{N}_i} h_{is}, & \text{if } i = j, \\ -h_{ij}, & \text{if } \{i,j\} \in \mathcal{E}, \\ 0, & \text{otherwise,} \end{cases}$$

where $h_{ij} = h_{ji} > 0 \ \forall \{i,j\} \in \mathcal{E}$.

Drawbacks: Each iteration of FDG is computational expensive.

Inexact Fenchel Dual Gradient (IFDG) Method

Initialization:

$$\mathbf{w}^0 \in S^{\perp}, \text{ or simply } \mathbf{w}^0 = \mathbf{0}_{Nd}. \text{ Also, arbitrarily choose } \mathbf{x}^{-1} \in \mathbb{R}^{Nd}.$$

Update: for any $k \geq 0$,

Primal update: $\mathbf{x}^k = \arg\max_{\mathbf{x} \in X} (\mathbf{w}^k)^T \mathbf{x} - F(\mathbf{x}).$

$$\Downarrow$$

$$\mathbf{x}^k = \text{Proj}_X\{\mathbf{x}^{k-1} - \alpha \nabla \phi^k(\mathbf{x}^{k-1})\}, \text{ where } \phi^k(\mathbf{x}) = -(\mathbf{w}^k)^T \mathbf{x} + F(\mathbf{x}).$$

Dual update: $\mathbf{w}^{k+1} = \mathbf{w}^k - \beta(H_{\mathcal{G}} \otimes I_d)\mathbf{x}^k.$

Distributed Implementation

Initialization: Each node $i \in \mathcal{V}$ sets $w_i^0 = \mathbf{0}_d$ and arbitrarily chooses $x_i^{-1} \in \mathbb{R}^d$.

Update: for any $k \geq 0$,

- Each node $i \in \mathcal{V}$ updates $x_i^k = \mathsf{Proj}_{X_i}\{x_i^{k-1} - \alpha \nabla \phi_i^k(x_i^{k-1})\}$, where ϕ_i^k is the convex objective function given by $\phi_i^k(x_i) := -(w_i^k)^T x_i + f_i(x_i)$.
- Each node $i \in \mathcal{V}$ sends x_i^k to every neighbor $j \in \mathcal{N}_i$.
- Upon receiving $x_j^k \ \forall j \in \mathcal{N}_i$, each node $i \in \mathcal{V}$ updates $w_i^{k+1} = w_i^k - \beta \sum_{j \in \mathcal{N}_i} h_{ij}(x_i^k - x_j^k)$.

Convergence Analysis: Unconstrained Case

- Connection with EXTRA
  - The updates of IFDG in two successive iterations as follows:

$$\begin{aligned}\mathbf{w}^{k+1} &= \mathbf{w}^{k} - \beta(H_{\mathcal{G}} \otimes I_d)\mathbf{x}^{k},\\ \mathbf{x}^{k+1} &= \mathbf{x}^{k} - \alpha\nabla F(\mathbf{x}^{k}) + \alpha\mathbf{w}^{k+1},\\ \mathbf{w}^{k+2} &= \mathbf{w}^{k+1} - \beta(H_{\mathcal{G}} \otimes I_d)\mathbf{x}^{k+1},\\ \mathbf{x}^{k+2} &= \mathbf{x}^{k+1} - \alpha\nabla F(\mathbf{x}^{k+1}) + \alpha\mathbf{w}^{k+2}.\end{aligned}$$

  - By subtraction and substitution,

$$\mathbf{x}^{k+2} - \mathbf{x}^{k+1} = \boxed{(H + I_{Nd})}\mathbf{x}^{k+1} - \boxed{I_{Nd}}\mathbf{x}^{k} - \alpha[\nabla F(\mathbf{x}^{k+1}) - \nabla F(\mathbf{x}^{k})],$$

    where $H = -\alpha\beta H_{\mathcal{G}} \otimes I_d \in \mathbb{R}^{Nd\times Nd}$ is a negative semidefinite matrix.

  - The update of EXTRA:

$$\mathbf{x}^{k+2} - \mathbf{x}^{k+1} = \boxed{W}\mathbf{x}^{k+1} - \boxed{\tilde{W}}\mathbf{x}^{k} - \alpha[\nabla F(\mathbf{x}^{k+1}) - \nabla F(\mathbf{x}^{k})]$$

    where $\frac{I+W}{2} \succeq \tilde{W}$.

Convergence Analysis: Unconstrained Case

- **Theorem**: If $X_i = \mathbb{R}^d \ \forall i \in \mathcal{V}$, then $\mathbf{x}^k$ linearly converges to $\mathbf{x}^\star$ with proper algorithm parameters.

Convergence Analysis: Constrained Case

- Let $V = H_{\mathcal{G}} \otimes I_d$ and $\mathbf{r}^k = (V^{\frac{1}{2}})^\dagger \mathbf{w}^k/\beta$. Then, the updates of IFDG are:

$$\mathbf{x}^k = \arg\min_{\mathbf{x}\in\mathbb{R}^{Nd}} \{u^{k-1}(\mathbf{x}) + I_X(\mathbf{x}) - \beta\langle V^{\frac{1}{2}}\mathbf{r}^k, \mathbf{x}\rangle\},$$
$$\mathbf{r}^{k+1} = \mathbf{r}^k - V^{\frac{1}{2}}\mathbf{x}^k,$$

  where $u^k(\mathbf{x}) = \langle \nabla F(\mathbf{x}^k), \mathbf{x}\rangle + \frac{1}{2\alpha}\|\mathbf{x}\|^2 - \frac{1}{\alpha}\langle \mathbf{x}, \mathbf{x}^k\rangle$.

  - u^k is a $\frac{1}{\alpha}$-smooth and $\frac{1}{\alpha}$-strongly convex function.
  - $\nabla u^k(\mathbf{x}) = \nabla F(\mathbf{x}^k) + \frac{1}{\alpha}(\mathbf{x} - \mathbf{x}^k)$ and $\nabla u^k(\mathbf{x}^k) = \nabla F(\mathbf{x}^k)$.

- **Theorem**: For each $K \geq 1$, let $\bar{\mathbf{x}}^K = \frac{1}{K}\sum_{k=0}^{K-1} \mathbf{x}^k$. Then, the following hold with proper algorithm parameters:
  - Objective value: $F(\bar{\mathbf{x}}^K) - F(\mathbf{x}^*) \leq O(1/K)$.
  - Primal feasibility: $\|V^{\frac{1}{2}}\bar{\mathbf{x}}^K\| \leq O(1/K)$.

Numerical Example: Unconstrained Problem

- Consider a logistic regression problem that often arises in machine learning.

$$\text{minimize}_{x\in\mathbb{R}^5} \sum_{i\in\mathcal{V}} \sum_{j=1}^{6} \log(1+e^{-(u_{ij}^T x)v_{ij}}) + \frac{\lambda}{2}\|x\|^2$$

where $\lambda = 5$, $(u_{ij}, v_{ij}) \in \mathbb{R}^5 \times \{-1, 1\}$ $\forall j = 1, \ldots, 6$, $\forall i \in \mathcal{V}$, and $\mathcal{V} = \{1, \ldots, 20\}$.

Numerical Example: Unconstrained Problem

Figure: Convergence performance of IFDG, FDG, and EXTRA in solving the logistic problem.

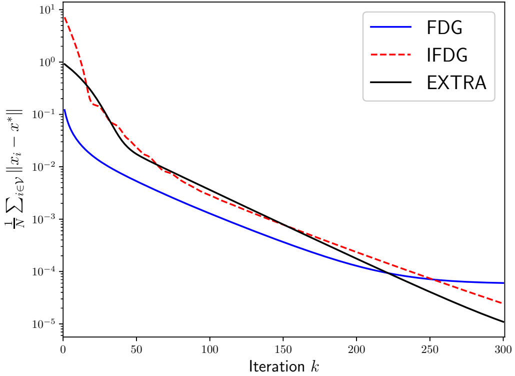


Table: Running time and accuracy after 300 iterations of IFDG, FDG and EXTRA in solving the logistic problem.

Algorithm	Running Time	Accuracy
IFDG	1.52s	2.44×10^{-5}
FDG	517.91s	6.04×10^{-5}
EXTRA	6.26s	1.10×10^{-5}

Numerical Example: Constrained Problem

- Consider a constrained quadratic programming problem:

$$\begin{array}{ll} \text{minimize}_{x \in \mathbb{R}^5} & \displaystyle\sum_{i \in \mathcal{V}} x^T A_i x + b_i^T x \\ \text{subject to} & x \in \displaystyle\bigcap_{i \in \mathcal{V}} \{x \in \mathbb{R}^5 : p_i \leq x \leq q_i\} \end{array}$$

where each $A_i \in \mathbb{R}^{5\times 5}$ is symmetric positive definite, $b_i \in \mathbb{R}^5$, $p_i, q_i \in \mathbb{R}^5$, $i \in \mathcal{V}$ and $\mathcal{V} = \{1, \ldots, 20\}$.

Numerical Example: Constrained Problem

Figure: Convergence performance of IFDG, FDG, and PG-EXTRA in solving the constrained problem.

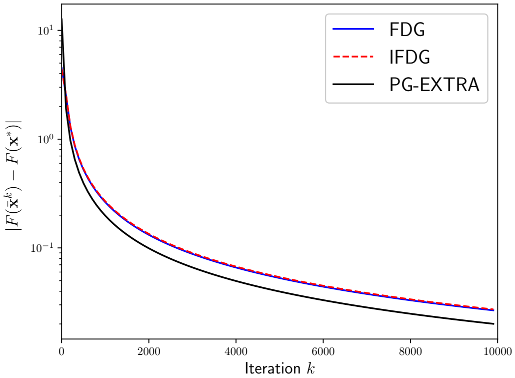


Table: Running time and accuracy after 10000 iterations of IFDG, FDG, and PG-EXTRA in solving the constrained problem.

Algorithm	Running Time	Accuracy
IFDG	13.65s	2.70×10^{-2}
FDG	2069.87s	2.64×10^{-2}
PG-EXTRA	172.09s	2.00×10^{-2}

Conclusion

- Develop the Inexact Fenchel Dual Gradient (IFDG) method for solving distributed optimization problems.
- Provide rates of convergence to the optimal solution for IFDG.
  - Linear rate for unconstrained problems.
  - Sublinear rate for constrained problems.
- Comparable accuracy with FDG, but significant reduction in computational costs.
- Simulations validate the convergence performance.

Thanks!