Stochastic Optimization for Learning over Networks

Guanghui (George) Lan

School of Industrial and Systems Engineering
Georgia Institute of Technology

East Coast Optimization Meeting 2019,
April 4-5, 2019
Department of Mathematical Sciences
George Mason University
Fairfax, Virginia (USA)

Machine Learning

Given a set of observed data $S = \{(u_i, v_i)\}_{i=1}^m$, drawn from a certain unknown distribution $\mathcal{D}$ on $U \times V$.

- Goal: to describe the relation between u_i and v_i's for prediction.
- Applications: predicting strokes and seizures, identifying heart failure, stopping credit card fraud, predicting machine failure, identifying spam, ...
- Classic models:
  - Lasso regression: $\min_x \mathbb{E}[(\langle x, u\rangle - v)^2] + \rho\|x\|_1$.
  - Support vector machine: $\min \mathbb{E}_{u,v}[\max\{0, v\langle x, u\rangle] + \rho\|x\|_2^2$.
  - Deep learning: $\min_x \mathbb{E}_{u,v}(F(u, x) - v)^2 + \rho\|Ux\|_1$

Machine learning and stochastic optimization

Generic stochastic optimization model:

$$\min_{x \in X} \{f(x) := \mathbb{E}_\xi[F(x, \xi)]\}.$$

- In ML, F is the regularized loss function and $\xi = (u, v)$:
  $F(x, \xi) = (\langle x, u \rangle - v)^2 + \rho \|x\|_1$
  $F(x, \xi) = \max\{0, v\langle x, u \rangle\} + \rho \|x\|_2^2$.
- To compute the gradient of f is expensive or impossible.
- Stochastic first-order methods: iterative methods which operate with the stochastic gradients (subgradients) of f.

Learning over networks

How about data are distributed over a multi-agent network?

$$\begin{aligned} \min_x \quad & f(x) := \sum_{i=1}^m f_i(x) \\ \text{s.t.} \quad & x \in X, \quad X := \cap_{i=1}^m X_i, \end{aligned}$$

where $f_i(x) = \mathbb{E}[F_i(x, \zeta_i)]$ is given in the form of expectation.

- Optimization defined over complex multi-agent network.
- Each agent has its own data (observations of ζ_i).
- Data usually are private - no sharing.
- Can share knowledge learned from data.
- Communication among agents can be expensive.
- Data can be captured on-line.

Example: SVM over networks

Three agents: $\min_x \frac{1}{3}[f_1(x) + f_2(x) + f_3(x)]$

- $f_1(x) = \frac{1}{N_1}\sum_{i=1}^{N_1}\left[\max\{0, v_i^1\langle x, u_i^1\rangle\right] + \rho\|x\|_2^2$.
- $f_2(x) = \frac{1}{N_2}\sum_{i=1}^{N_2}\left[\max\{0, v_i^2\langle x, u_i^2\rangle\right] + \rho\|x\|_2^2$.
- $f_3(x) = \frac{1}{N_3}\sum_{i=1}^{N_3}\left[\max\{0, v_i^3\langle x, u_i^3\rangle\right] + \rho\|x\|_2^2$.
- Dataset for agent 1: $\{(u_i^1, v_i^1)\}_{i=1}^{N_1}$.
- Dataset for agent 2: $\{(u_i^2, v_i^2)\}_{i=1}^{N_2}$.
- Dataset for agent 3: $\{(u_i^2, v_i^2)\}_{i=1}^{N_3}$.
- Each agent accesses its own dataset.

Example: SVM over networks with streaming data

$\min_x \frac{1}{3}[f_1(x) + f_2(x) + f_3(x)]$,
where $f_j(x) = \mathbb{E}\left[\max\{0, v^j\langle x, u^j\rangle\right] + \rho\|x\|_2^2$, $j = 1, 2, 3$.

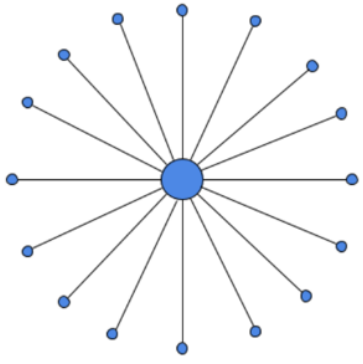
- Dataset for agent i can be viewed as samples of random vector (u^j, v^j), $j = 1, 2, 3$.
- (u^j, v^j) can satisfy different distribution.
- Samples can be collected in an online fashion.
- Agents can possibly share solutions, but not the samples.
- Need to minimize the communication costs.

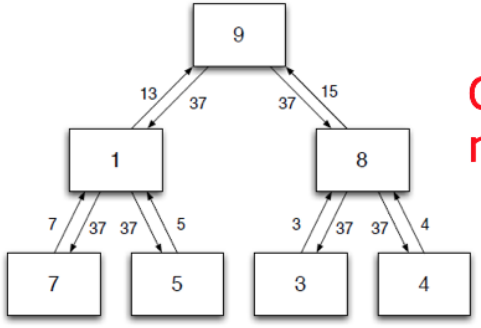
Key questions

- # samples - sampling complexity
- # communication rounds - communication complexity

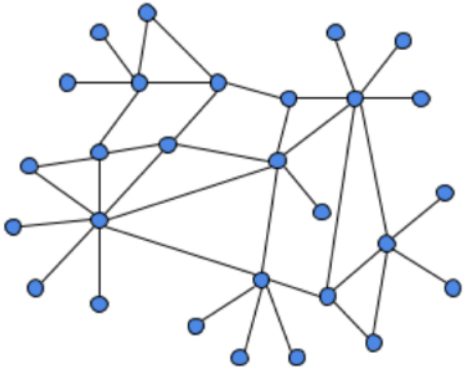
Network topology?

$$\min_x f(x) := \sum_{i=1}^m f_i(x) \quad \text{s.t. } x \in \bigcap_{i=1}^m X_i$$





Only the central node maintains x



Everybody maintains a local copy of x

- Centralized stochastic gradient descent
  - Sampling complexity
- Distributed SGD and federated learning
  - Sampling complexity
  - Communication complexity
- Decentralized SGD
  - How to communicate
  - Sampling complexity
  - Communication complexity

Stochastic (sub)gradients

The Problem: $\min_{x \in X}\{f(x) := \mathbb{E}_\xi[F(x,\xi)]\}$.

Stochastic (sub)gradients

At iteration t, $x_t \in X$ being the input, we have access to a vector $G(x_t, \xi_t)$, where $\{\xi_t\}_{t\geq 1}$ are i.i.d. random variables s.t.

$$\mathbb{E}[G(x_t,\xi_t)] \equiv g(x_t) \in \partial\Psi(x_t), \mathbb{E}[\|G(x,\xi)\|^2] \leq M^2.$$

Examples:

- Regression with batch data:
  $\min_x f(x) = \frac{1}{N}\sum_{i=1}^N (\langle x, u_i\rangle - v_i)^2$
  - Stochastic gradient: $2(\langle x, u_{i_t}\rangle - v_{i_t})u_{i_t}$
- Regression with streaming data:
  $\min_x f(x) = \mathbb{E}\left[(\langle x, u\rangle - v)^2\right]$
  - Stochastic gradient: $2(\langle x, u\rangle - v)u$

Stochastic (sub)grdient descent

The algorithm

$$x_{t+1} = \text{argmin}_{x \in X} \|x - (x_t - \gamma_t G_t)\|_2^2, t = 1, 2, \ldots$$

Theorem (Nemirovski, Juditsky, Lan and Shapiro 07 (09))

Let $D_X \geq \max_{x_1, x_2 \in X} \|x_1 - x_2\|_2$. *If* $\gamma_t = \sqrt{\Omega^2/(kM^2)}$, $t = 1, \ldots, k$, *and* $\bar{x}^k = \sum_{t=1}^k x_t/k$,

$$\mathbb{E}[f(\bar{x}^k) - f^*] \leq \frac{MD_X}{2\sqrt{k}}, \quad \forall k \geq 1.$$

Sampling complexity

samples = # iterations = $\mathcal{O}(1)(\frac{M^2 D_X^2}{\epsilon^2})$,
to find a solution $\bar{x} \in X$ such that $\mathbb{E}[f(\bar{x}) - f^*] \leq \epsilon$.

Recent developments

- Accelerated SGD (Lan 08 (12))
  - Stochastic version of Nesterov's accelerated gradient method
  - A universally optimal method for smooth, nonsmooth and stochastic optimization
  - The impact of Lipschitz constants vanishes for stochastic problems
  - Popular in training deep neural networks (Sutskever, Martens, Dahl, Hinton 13)
- Adaptive stochastic subgradient (Duchi, Hazan, Singer 11)
- Nonconvex SGD and its acceleration (Ghadimi and Lan 12)
- Adaptive sample sizes (Byrd, Chin, Nocedal and Wu 12)
- SGD for finite-sum problems (Schmidt, Roux and Bach 13)
- Optimal incremental gradient methods (Lan and Zhou 15)

The distributed structure - star topology

Figure: A cloud-device based distributed learning system

- Data sets are distributed over individual agents (devices) in the network.
- All devices are connected to a parameter server (or central cloud), which controls the learning process and updates solutions.
- One example: *federated learning*.

Stochastic finite sum optimization

Consider the convex programming (CP) problem given by

$$\min_{x \in X} \Psi(x) := \frac{1}{m}\sum_{i=1}^{m} f_i(x) + \mu\, \omega(x).$$

- $X \subseteq \mathbb{R}^n$ is a closed convex set.
- $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, are smooth convex with Lipschitz constants $L_i \geq 0$. $f(x) := \frac{1}{m}\sum_{i=1}^{m} f_i(x)$ is smooth convex with Lipschitz constant $L_f \leq L = \frac{1}{m}\sum_{i=1}^{m} L_i$.
- $\omega : X \to \mathbb{R}$ is a strongly convex function with modulus 1 w.r.t. an arbitrary norm $\|\cdot\|$.
- $\mu \geq 0$ is a given constant.
- $f_i(x) = \mathbb{E}[F_i(x, \xi_i)]$ can be represented by a stochastic oracle, providing stochastic (sub)gradients upon request.

Randomized incremental gradient (RIG) methods

Randomized incremental gradient (RIG) methods solve $\min_{x \in X} \frac{1}{m} \sum_{i=1}^{m} f_i(x)$ iteratively, at k-th iteration,

1) Randomly select a component index i_k from $1, \ldots, m$ (server).
2) Compute the gradient of the component function $f_{i_k}(x_k)$ (agents).
3) Set $x_{k+1} = P_X(x_k - \alpha_k \nabla f_{i_k}(x_k))$, where α_k is a positive step-size, $P_X(\cdot)$ denotes projection on X (server).

- Potentially save the total number of gradient computations.
- Save communication costs in distributed setting .

Existing RIG methods

- **SAG/SAGA** in Schmidt et al, 13 and Defazio et al, 14, and **SVRG** in Johnson and Zhang, 13 obtained $\mathcal{O}\left((m + L/\mu)\log\frac{1}{\epsilon}\right)$ rate of convergence $\rightsquigarrow$ rate of convergence not optimal
- **RPDG** in Lan and Zhou, 15 (precursor: Zhang and Xiao 14, Dang and Lan 14) $\rightsquigarrow$ require exact gradient evaluation at the initial point, and differentiability over $\mathbb{R}^n$
- **Catalyst scheme** in Lin et al, 15 and Katyusha in Allen-Zhu, 16 require re-evaluating exact gradients from time to time $\rightsquigarrow$ synchronous delays
- No existing studies on stochastic finite-sum: each f_i is represented by a stochastic oracle, or each agent only has access to noisy first-order information

Road map

Goals:

- Fully-distributed (no exact gradient evaluations)
- Direct acceleration with optimal communication costs
- Applicable to solve stochastic finite-sum problems - optimal sampling complexity

Outline:

- Gradient Extrapolation Method - GEM
- Interpretation on GEM
- Randomized Gradient Extrapolation Method - RGEM

Prox-function and prox-mapping

We define a *prox-function* associated with ω as

$$P(x^0, x) := \omega(x) - [\omega(x^0) + \langle \omega'(x^0), x - x^0 \rangle],$$

and the prox-mapping associated with X and ω is given by

$$\mathcal{M}_X(g, x^0, \eta) := \arg\min_{x \in X} \left\{ \langle g, x \rangle + \mu\, \omega(x) + \eta P(x^0, x) \right\}.$$

- $P(\cdot, \cdot)$ is a generalization of Bregman's distance, since ω is not necessarily differentiable.
- $P(\cdot, \cdot)$ is strongly convex w.r.t. an arbitrary norm because of the strong convexity of ω.
- Reasonable to assume the above prox-mapping problem is easy to solve.

The Problem: $\min_{\mathbf{x}\in X}\{f(x)+\mu\omega(x) := \frac{1}{m}\sum_{i=1}^{m} f_i(x)+\mu\omega(x)\}$

Gradient Extrapolation Method (GEM)

Initialization:

$\underline{x}^0 = x^0 \in X$ and $g^{-1} = g^0 = \nabla f(x^0)$. $\rightsquigarrow$ exact gradient evaluation

for $t = 1, 2, \ldots, k$ **do**

$$\tilde{g}^t = \alpha_t(g^{t-1} - g^{t-2}) + g^{t-1}.$$
$$x^t = \mathcal{M}_X(\tilde{g}^t, x^{t-1}, \eta_t).$$
$$\underline{x}^t = \left(x^t + \tau_t \underline{x}^{t-1}\right)/(1+\tau_t).$$
$$g^t = \nabla f(\underline{x}^t). \rightsquigarrow \text{one exact gradient evaluation}$$

end for

Output: $\underline{x}^k$.

Intuition: Game interpretation of GEM

- A game iteratively performed by a primal player (x) and a dual player (g).

$$\begin{aligned}&\min_{x\in X}\{f(x)+\mu\,\omega(x)\}\\&=\min_{x\in X}\left\{\max_{g\in\mathcal{G}}\{\langle x,g\rangle-J_f(g)\}+\mu\,\omega(x)\right\}.\end{aligned}$$

- The primal player predicts the dual player's action based on historical information, and determines her/his corresponding action by minimizing predicted cost.

$$\begin{aligned}\tilde{g}^t&=\alpha_t(g^{t-1}-g^{t-2})+g^{t-1},\\x^t&=\arg\min_{x\in X}\left\{\langle\tilde{g}^t,x\rangle+\mu\,\omega(x)+\eta_t P(x^{t-1},x)\right\}.\end{aligned}$$

- The dual player determines her/his action g^t by maximizing the profits.

$$g^t=\arg\min_{g\in\mathcal{G}}\left\{\langle -x^t,g\rangle+J_f(g)+\tau_t D_g(g^{t-1},g)\right\}\Leftrightarrow$$

$$\begin{cases}\underline{x}^t&=\left(x^t+\tau_t\underline{x}^{t-1}\right)/(1+\tau_t),\\g^t&=\nabla f(\underline{x}^t).\end{cases}$$

GEM: the dual of Nesterov's accelerated gradient method

GEM:

$$\tilde{g}^t = \alpha_t(g^{t-1} - g^{t-2}) + g^{t-1}$$
$$x^t = \mathcal{M}_X(\tilde{g}^t, x^{t-1}, \eta_t)$$
$$g^t = \mathcal{M}_{\mathcal{G}}(-x^t, g^{t-1}, \tau_t)$$

NEST:

$$\tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}$$
$$g^t = \mathcal{M}_{\mathcal{G}}(-\tilde{x}^t, g^{t-1}, \tau_t)$$
$$x^t = \mathcal{M}_X(g^t, x^{t-1}, \eta_t)$$

Adding randomization...

The Problem: $\min_{\mathbf{x}\in X}\{f(x)+\mu\omega(x) := \frac{1}{m}\sum_{i=1}^{m} f_i(x)+\mu\omega(x)\}$

GEM

Initialization:
$\underline{x}^0 = x^0 \in X$ and $g^{-1} = g^0 = \nabla f(x^0)$.
for $t = 1, 2, \ldots, k$ **do**

$$\tilde{g}^t = \alpha_t(g^{t-1} - g^{t-2}) + g^{t-1}.$$
$$x^t = \mathcal{M}_X(\tilde{g}^t, x^{t-1}, \eta_t).$$
$$\underline{x}^t = \left(x^t + \tau_t \underline{x}^{t-1}\right)/(1+\tau_t).$$
$$g^t = \nabla f(\underline{x}^t).$$

end for
Output: $\underline{x}^k$.

The Problem: $\min_{\mathbf{x}\in X} \psi(x) := \frac{1}{m}\sum_{i=1}^{m} f_i(x) + \mu\omega(x)$

RGEM

Initialization:

$\underline{x}_i^0 = x^0 \in X,\ \forall i,\ y^{-1} = y^0 = \mathbf{0}$. $\rightsquigarrow$ No exact gradient evaluation

for $t = 1, \ldots, k$ **do**

Choose i_t uniformly from $\{1, \ldots, m\}$,

$$\tilde{y}^t \leftarrow y^{t-1} + \alpha_t(y^{t-1} - y^{t-2}),$$

$$x^t \leftarrow \mathcal{M}_X(\tfrac{1}{m}\textstyle\sum_{i=1}^{m} \tilde{y}_i^t, x^{t-1}, \eta_t),$$

$$\underline{x}_i^t \leftarrow \begin{cases} (1+\tau_t)^{-1}(x^t + \tau_t \underline{x}_i^{t-1}), & i = i_t, \\ \underline{x}_i^{t-1}, & i \neq i_t. \end{cases}$$

$$y_i^t \leftarrow \begin{cases} \nabla f_i(\underline{x}_i^t), & i = i_t, \rightsquigarrow \text{one gradient evaluation} \\ y_i^{t-1}, & i \neq i_t. \end{cases}$$

end for

Output: For some $\theta_t > 0$, set $\underline{x}^k := (\sum_{t=1}^{k}\theta_t)^{-1}\sum_{t=1}^{k}\theta_t x^t$.

RGEM from the server and activated agent perspective

RGEM The server's perspective

1: **while** $t \leq k$ **do**
2: $x^t \leftarrow \mathcal{M}_X(g^{t-1} + \frac{\alpha_t}{m}\Delta y, x^{t-1}, \eta_t)$
3: $\text{sumx} \leftarrow \text{sumx} + \theta_t x^t$
4: $\text{sum}\theta \leftarrow \text{sum}\theta + \theta_t$
5: Send signal to the i_t-th agent where i_t is selected uniformly from $\{1, \ldots, m\}$
6: **if** i_t-th agent is responsive **then**
7: Send current iterate x^t to i_t-th agent
8: **if** Receive feedback Δy **then**
9: $g^t \leftarrow g^{t-1} + \Delta y$
10: $t \leftarrow t + 1$
11: **else goto** *Line* 5
12: **end if**
13: **else goto** *Line* 5
14: **end if**
15: **end while**

RGEM The activated i_t-th agent's perspective

1: Download the current iterate x^t from the server
2: **if** $t = 1$ **then**
3: $y_i^{t-1} \leftarrow \mathbf{0}$
4: **else**
5: $y_i^{t-1} \leftarrow \nabla f_i(\underline{x}_i^{t-1})$ ▷ Optional
6: **end if**
7: $\underline{x}_i^t \leftarrow (1 + \tau_t)^{-1}(x^t + \tau_t \underline{x}_i^{t-1})$
8: $y_i^t \leftarrow \nabla f_i(\underline{x}_i^t)$
9: Upload the local changes to the server, i.e., $\Delta y_i = y_i^t - y_i^{t-1}$

- The server updates iterates x^t and calculates the output solution $\underline{x}^k$ given by $\text{sumx}/\text{sum}\theta$.
- One agent is activated at a time, updating local variables

RGEM for deterministic finite-sum optimization

Theorem

Let x^* *be an optimal solution,* $\hat{L} = \max_{i=1,\ldots,m} L_i$,

$$\tau_t = \frac{1}{m(1-\alpha)} - 1,\ \eta_t = \frac{\alpha}{1-\alpha}\mu,\ \alpha_t \equiv m\alpha,\ \alpha = 1 - \frac{1}{m+\sqrt{m^2+16m\hat{L}/\mu}}.$$

$$\begin{aligned} \mathbb{E}[P(x^k, x^*)] &\leq \frac{2\Delta_{0,\sigma_0}\alpha^k}{\mu}, \\ \mathbb{E}[\psi(x^k) - \psi(x^*)] &\leq 16\max\left\{m, \frac{\hat{L}}{\mu}\right\}\Delta_{0,\sigma_0}\alpha^{k/2}, \end{aligned}$$

where $\Delta_{0,\sigma_0} := \mu P(x^0, x^*) + \psi(x^0) - \psi(x^*) + \frac{\sigma_0^2}{m\mu}$, *and* σ_0 *satisfies* $\frac{1}{m}\sum_{i=1}^m \|\nabla f_i(x^0)\|_*^2 \leq \sigma_0^2$.

To obtain a stochastic ϵ-solution $\Rightarrow$
of **gradient evaluations of** f_i / **communication rounds**:

$\mathcal{O}\{\left(m + \sqrt{\frac{m\hat{L}}{\mu}}\right)\log\frac{1}{\epsilon}\}$ (not improvable, Lan and Zhou 17).

The Problem: $\min_{\mathbf{x}\in X} \psi(x) := \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{\xi_i}[F_i(x,\xi_i)] + \mu\omega(x)$

Assumption

At iteration t, for a given point $\underline{x}_i^t \in X$, a stochastic first-order ($\mathcal{SO}$) oracle outputs a vector $G_i(\underline{x}_i^t, \xi_i^t)$ s.t.

$$\mathbb{E}_\xi[G_i(\underline{x}_i^t, \xi_i^t)] = \nabla f_i(\underline{x}_i^t),\ i = 1,\dots,m,$$
$$\mathbb{E}_\xi[\|G_i(\underline{x}_i^t, \xi_i^t) - \nabla f_i(\underline{x}_i^t)\|_*^2] \le \sigma^2,\ i = 1,\dots,m.$$

RGEM for stochastic finite-sum optimization

The same as RGEM except that the gradient update step is replaced by

$$y_i^t \leftarrow \begin{cases} \frac{1}{B_t}\sum_{j=1}^{B_t} G_i(\underline{x}_i^t, \xi_{i,j}^t), & i = i_t, \rightsquigarrow \text{stochastic gradients of } f_i \text{ given by } \mathcal{SO} \\ y_i^{t-1}, & i \neq i_t. \end{cases}$$

RGEM for stochastic finite-sum optimization

Theorem

Let τ_t, η_t *and* α_t *be the same as before and let*

$$B_t = \lceil k(1-\alpha)^2\alpha^{-t} \rceil, \ t = 1, \ldots, k,$$

$$\begin{aligned} \mathbb{E}[P(x^k, x^*)] &\leq \frac{2\alpha^k \Delta_{0,\sigma_0,\sigma}}{\mu}, \\ \mathbb{E}[\psi(\underline{x}^k) - \psi(x^*)] &\leq 16 \max\left\{m, \frac{\hat{L}}{\mu}\right\} \Delta_{0,\sigma_0,\sigma} \alpha^{k/2}, \end{aligned}$$

where $\Delta_{0,\sigma_0,\sigma} := \mu P(x^0, x^*) + \psi(x^0) - \psi(x^*) + \frac{\sigma_0^2/m + 5\sigma^2}{\mu}$.

communication rounds: $\mathcal{O}\{\left(m + \sqrt{m\hat{L}/\mu}\right) \log \frac{1}{\epsilon}\}$,

stochastic gradient evaluations: $\tilde{\mathcal{O}}\{(\frac{\Delta_{0,\sigma_0,\sigma}}{\mu\epsilon} + m + \sqrt{m\hat{L}/\mu})\}$.

Note: Sampling complexity independent of m asymptotically.

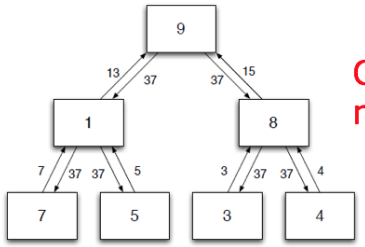
Advantages of RGEM for distributed learning

RGEM is an enhanced RIG method for distributed learning:

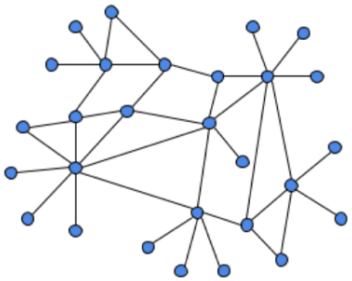
- Require no exact gradient evaluations of f
- Involve communication only between the server and the activated agent iteratively, and tolerate communication failures
- Possess direct accelerated algorithmic scheme
- Stochastic/online optimization - minimization of generalization risk
- Optimal $\mathcal{O}\{(m+\sqrt{m\hat{L}/\mu})\log\frac{1}{\epsilon}\}$ communication complexity
- Nearly optimal $\tilde{\mathcal{O}}\{\frac{\Delta_{0,\sigma_0,\sigma}}{\mu\epsilon}\}$ sampling complexity.

Network topology?

$$\min_x f(x) := \sum_{i=1}^m f_i(x) \quad \text{s.t. } x \in \bigcap_{i=1}^m X_i$$



Only the central node maintains x



Everybody maintains a local copy of x

Example: Policy evaluation for multi-agent reinforcement learning (Wai, Yang, Wang, Hong 18).

Decentralized optimization techniques

- Most studies focus on deterministic optimization (e.g., Nedic and Ozdaglar 09; Shi, Ling, Wu, and Yin 15).
  - $\mathcal{O}(1/\epsilon)$ communication rounds and gradient computation.
  - $\mathcal{O}(\log(1/\epsilon))$ communication rounds for unconstrained smooth and strongly convex problems.
- For stochastic optimization problems
  - Direct extension of SGD type methods (e.g. Duchi, Agarwal, and Wainwright 12).
  - $\mathcal{O}(1/\epsilon^2)$ communication rounds and stochastic (sub)gradient computations.

Question

Is SGD still a good algorithm for decentralized stochastic optimization and machine learning?

How to handle decentralized structure?

- Dual decomposition (explicit)

  $\min_{\mathbf{x}} \quad F(\mathbf{x}) := \sum_{i=1}^{m} f_i(x_i)$

  s.t. $\quad x_1 = x_2 = \ldots = x_m, \quad x_i \in X_i, \forall i = 1, \ldots, m.$

  where $x = [x_1^T, \ldots, x_m^T]$.

Background: Laplacian L

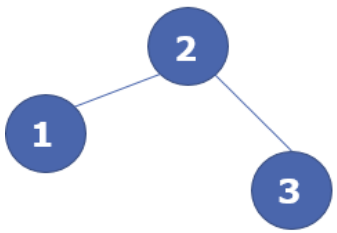
- Let N_i denote the set of neighbors of agent i:

$$N_i = \{j \in V \mid (i,j) \in \mathcal{E}\} \cup \{i\}$$

- Then, the Laplacian $L \in \mathbb{R}^{m \times m}$ of a graph $G = (V, \mathcal{E})$ is defined as:

$$L_{ij} = \begin{cases} |N_i| - 1 & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

- For example:



$$L = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$L\mathbf{1} = \mathbf{0}$
"Agreement Subspace"

Problem Formulation

$$\min_{\mathbf{x}} F(\mathbf{x}) := \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t. } x_1 = \cdots = x_m$$
$$x_i \in X_i,\ i = 1, \ldots, m$$

(=)

$$\min_{\mathbf{x}} F(\mathbf{x}) := \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t. } x_i = x_j,\ \forall (i,j) \in \mathcal{E}$$
$$x_i \in X_i,\ i = 1, \ldots, m$$

If G is connected

(=)

$$\min_{\mathbf{x}} F(\mathbf{x}) := \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t. } \mathbf{Lx} = \mathbf{0}$$
$$x_i \in X_i,\ i = 1, \ldots, m$$

Using Laplacian L, consistency constraints can be compactly rewritten $\mathbf{L} := L \otimes I_d$

(=)

$$\min_{\mathbf{x} \in X^m} F(\mathbf{x}) + \max_{\mathbf{y} \in \mathbb{R}^{md}} \langle \mathbf{Lx}, \mathbf{y} \rangle$$

Equivalent Saddle Point form

Decentralized Primal-Dual (DPD): Vector Form

$$\min_{\mathbf{x}\in X^m} F(\mathbf{x}) + \max_{\mathbf{y}\in\mathbb{R}^{md}} \langle \mathbf{L}\mathbf{x}, \mathbf{y}\rangle$$

$$\mathbf{x} := [x_1^\top \cdots x_m^\top]^\top$$
$$\mathbf{y} := [y_1^\top \cdots y_m^\top]^\top$$

Let $\mathbf{x}^0 = \mathbf{x}^{-1} \in X^m$, $\mathbf{y}^0 \in \mathbb{R}^{md}$, $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$, and $\{\theta_k\}$ be given.

For $k = 1, \ldots, N$, update $\mathbf{z}^k = (\mathbf{x}^k, \mathbf{y}^k)$

$$\tilde{\mathbf{x}}^k = \alpha_k(\mathbf{x}^{k-1} - \mathbf{x}^{k-2}) + \mathbf{x}^{k-1}$$
$$\mathbf{y}^k = \operatorname{argmin}_{\mathbf{y}\in\mathbb{R}^{md}} \langle -\mathbf{L}\tilde{\mathbf{x}}^k, \mathbf{y}\rangle + \tfrac{\tau_k}{2}\|\mathbf{y} - \mathbf{y}^{k-1}\|^2$$
$$\mathbf{x}^k = \operatorname{argmin}_{\mathbf{x}\in X^m} \langle \mathbf{L}\mathbf{y}^k, \mathbf{x}\rangle + F(\mathbf{x}) + \tfrac{\eta_k}{2}\|\mathbf{x}^{k-1} - \mathbf{x}\|^2$$

Return $\bar{\mathbf{z}}^N = (\sum_{k=1}^N \theta_k)^{-1} \sum_{k=1}^N \theta_k \mathbf{z}^k$.

DPD: Agent i's point of view

Let $\mathbf{x}^0 = \mathbf{x}^{-1} \in X^m$, $\mathbf{y}^0 \in \mathbb{R}^{md}$, $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$, and $\{\theta_k\}$ be given.

For $k = 1, \ldots, N$, update $z_i^k = (x_i^k, y_i^k)$

$$\tilde{x}_i^k = \alpha_k(x_i^{k-1} - x_i^{k-2}) + x_i^{k-1}$$
$$v_i^k = \sum_{j \in N_i} L_{ij} \tilde{x}_j^k$$
$$y_i^k = y_i^{k-1} + \frac{1}{\tau_k} v_i^k$$
$$w_i^k = \sum_{j \in N_i} L_{ij} y_j^k$$
$$x_i^k = \operatorname{argmin}_{x_i \in X_i} \langle w_i^k, x_i \rangle + f_i(x_i) + \frac{\eta_k}{2} \|x_i^{k-1} - x_i\|^2$$

Return $\bar{\mathbf{z}}^N = (\sum_{k=1}^N \theta_k)^{-1} \sum_{k=1}^N \theta_k \mathbf{z}^k$

Decentralized Communication Sliding (DCS)

Q: Is the subproblem always easy to solve?

$$x_i^k = \text{argmin}_{x_i \in X_i} \langle w_i^k, x_i \rangle + f_i(x_i) + \frac{\eta_k}{2} \|x_i^{k-1} - x_i\|^2$$

A: No, solve this iteratively using linearization of $f_i(x_i)$

Let $u^0 = \hat{u}^0 = x_i^{k-1}$, $\{\beta_t\}$, and $\{\lambda_t\}$ be given.
For $t = 1, \ldots, T_k$,

$$h^{t-1} \in \partial f_i(u^{t-1})$$

$$u^t = \text{argmin}_{u \in X_i} \langle h^{t-1} + w_i^k, u \rangle + \frac{\eta_k}{2} \|x_i^{k-1} - u\|^2 + \frac{\eta_k \beta_t}{2} \|u^{t-1} - u\|^2$$

Return $x_i^k = u^T$ and $\hat{x}_i^k = \left(\sum_{t=1}^t \lambda_t\right)^{-1} \sum_{t=1}^t \lambda_t u^t$

The same w_i^k is used, communication is skipped! There are two output points x_i^k and $\hat{x}_i^k$.

Decentralized Communication Sliding (DCS)

Let $\mathbf{x}^0 = \mathbf{x}^{-1} \in X^m$, $\mathbf{y}^0 \in \mathbb{R}^{md}$, $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$, $\{\theta_k\}$, and $\{T_k\}$ be given.

For $k = 1, \ldots, N$, update $z_i^k = (\hat{x}_i^k, y_i^k)$

$$\begin{aligned}
\tilde{x}_i^k &= \alpha_k(\hat{x}_i^{k-1} - x_i^{k-2}) + x_i^{k-1} \\
v_i^k &= \textstyle\sum_{j \in N_i} L_{ij} \tilde{x}_j^k \\
y_i^k &= \operatorname{argmin}_{y_i \in \mathbb{R}^d} \langle -v_i^k, y_i \rangle + \tfrac{\tau_k}{2} \| y_i - y_i^{k-1} \|^2 \\
w_i^k &= \textstyle\sum_{j \in N_i} L_{ij} y_j^k \\
(x_i^k, \hat{x}_i^k) &= \text{Inner loop for } T_k \text{ times}
\end{aligned}$$

Convergence of DCS

Theorem

Let $\hat{\mathbf{x}}^N = \frac{1}{N}\sum_{k=1}^{N}\hat{\mathbf{x}}^k$ *and* $\mathbf{x}^*$ *be an optimal solution. If*

$$\alpha_k = \theta_k = 1,\ \eta_k = 2\|\mathbf{L}\|,\ \tau_k = \|\mathbf{L}\|,\ \text{and } T_k = \left\lceil \frac{m(M^2+\sigma^2)N}{\|\mathbf{L}\|^2\tilde{D}} \right\rceil,$$

then

$$F(\hat{\mathbf{x}}^N) - F(\mathbf{x}^*) \le \frac{\|\mathbf{L}\|}{N}\left[\frac{3}{2}\|\mathbf{x}^0 - \mathbf{x}^*\|^2 + \frac{1}{2}\|\mathbf{y}^0\|^2 + 4\tilde{D}\right],$$

$$\|\mathbf{L}\hat{\mathbf{x}}^N\| \le \frac{\|\mathbf{L}\|}{N}\left[3\sqrt{3\|\mathbf{x}^0 - \mathbf{x}^*\|^2 + 8\tilde{D}} + 4\|\mathbf{y}^* - \mathbf{y}^0\|\right].$$

- $\mathcal{O}(1/\epsilon)$ iterations for ϵ-optimal and ϵ-feasible solution.
- # of required communications is also $\mathcal{O}(1/\epsilon)$.
- # of subgradient evaluations is $\mathcal{O}(1/\epsilon^2)$.

- Decentralized stochastic optimization

$$f_i(x) := \mathbb{E}_{\xi_i}[F_i(x; \xi_i)],$$

where ξ_i models agent i's uncertainty and $\mathbb{P}(\xi_i)$ not known.

- Only noisy first-order information $G_i(\cdot, \xi_i^t)$ is available:

$$\mathbb{E}[G_i(u^t, \xi_i^t)] = f_i'(u^t) \in \partial f_i(u^t),$$
$$\mathbb{E}[\|G_i(u^t, \xi_i^t) - f_i'(u^t)\|_*^2] \leq \sigma^2.$$

The primal subproblem is solved with **noisy subgradients**

$$x_i^k = \text{argmin}_{x_i \in X_i} \langle w_i^k, x_i \rangle + f_i(x_i) + \frac{\eta_k}{2} \|x_i^{k-1} - x_i\|^2.$$

Let $u^0 = \hat{u}^0 = x_i^{k-1}$, $\{\beta_t\}$, and $\{\lambda_t\}$ be given.
For $t = 1, \ldots, T_k$,

$$h^{t-1} = G_i(u^{t-1}, \xi_i^{t-1})$$
$$u^t = \text{argmin}_{u \in X_i} \langle h^{t-1} + w_i^k, u \rangle + \frac{\eta_k}{2} \|x_i^{k-1} - u\|^2 + \frac{\eta_k \beta_t}{2} \|u^{t-1} - u\|^2$$

Return $x_i^k = u^T$ and $\hat{x}_i^k = \left(\sum_{t=1}^t \lambda_t\right)^{-1} \sum_{t=1}^t \lambda_t u^t$

Observations: The same w_i^k is used, communication is skipped! There are two output points x_i^k and $\hat{x}_i^k$.

Convergence of SCDS

Theorem

Let $\hat{\mathbf{x}}^N = \frac{1}{N}\sum_{k=1}^{N}\hat{\mathbf{x}}^k$. *If* $\beta_t = \frac{t}{2}$, $\lambda_t = t+1$, $\alpha_k = \theta_k = 1$, $\eta_k = 2\|\mathbf{L}\|$, $\tau_k = \|\mathbf{L}\|$ *and* $T_k = \left\lceil \frac{m(M^2+\sigma^2)N}{\|\mathbf{L}\|^2\tilde{D}} \right\rceil$ *for some* $\tilde{D} > 0$, *then*

$$\mathbb{E}[F(\hat{\mathbf{x}}^N) - F(\mathbf{x}^*)] \leq \frac{\|\mathbf{L}\|}{N}\left[\frac{3}{2}\|\mathbf{x}^0 - \mathbf{x}^*\|^2 + \frac{1}{2}\|\mathbf{y}^0\|^2 + 4\tilde{D}\right],$$

$$\mathbb{E}[\|\mathbf{L}\hat{\mathbf{x}}^N\|] \leq \frac{\|\mathbf{L}\|}{N}\left[3\sqrt{3\|\mathbf{x}^0 - \mathbf{x}^*\|^2 + 8\tilde{D}} + 4\|\mathbf{y}^* - \mathbf{y}^0\|\right].$$

Conclusion:

- $\mathcal{O}(1/\epsilon)$ iterations for ϵ-optimal and ϵ-feasible solution.
- # of required communications is also $\mathcal{O}(1/\epsilon)$.
- # of stochastic subgradient evaluations is $\mathcal{O}(1/\epsilon^2)$.

Summary of convergence results

Table: Complexity for obtaining an ϵ-optimal and ϵ-feasible solution

Algorithm (problem type)	# of communications	# of subgradient evaluations
DCS: Convex	$\mathcal{O}\left\{\frac{\|\mathbf{L}\|\mathcal{D}_{X^m}^2}{\epsilon}\right\}$	$\mathcal{O}\left\{\frac{mM^2\mathcal{D}_{X^m}^2}{\epsilon^2}\right\}$
SDCS: Convex	$\mathcal{O}\left\{\frac{\|\mathbf{L}\|\mathcal{D}_{X^m}^2}{\epsilon}\right\}$	$\mathcal{O}\left\{\frac{m(M^2+\sigma^2)\mathcal{D}_{X^m}^2}{\epsilon^2}\right\}$
DCS: Strongly convex	$\mathcal{O}\left\{\sqrt{\frac{\mu\mathcal{D}_{X^m}^2}{\epsilon}}\right\}$	$\mathcal{O}\left\{\frac{mM^2}{\mu\epsilon}\right\}$
SDCS: Strongly convex	$\mathcal{O}\left\{\sqrt{\frac{\mu\mathcal{D}_{X^m}^2}{\epsilon}}\right\}$	$\mathcal{O}\left\{\frac{m(M^2+\sigma^2)}{\mu\epsilon}\right\}$

Assumptions: In the worse case, $M_f \leq mM$, $\mu_f \geq m\mu$, $\mathcal{D}_X^2/\mathcal{D}_{X^m}^2 = \mathcal{O}(1/m)$ and $\tilde{\sigma}^2 \leq m\sigma^2$.

Table: # of stochastic subgradient evaluations

Problem type	SDCS (individual agent)	SGD
Convex	$\mathcal{O}\left\{\frac{m(M^2+\sigma^2)\mathcal{D}_{X^m}^2}{\epsilon^2}\right\}$	$\mathcal{O}\left\{\frac{m(M_f^2+\tilde{\sigma}^2)\mathcal{D}_X^2}{\epsilon^2}\right\}$
Strongly convex	$\mathcal{O}\left\{\frac{m(M^2+\sigma^2)}{\mu\epsilon}\right\}$	$\mathcal{O}\left\{\frac{m(M_f^2+\tilde{\sigma}^2)}{\mu_f\epsilon}\right\}$

Conclusion: Sampling complexity comparable to centralized SGD under reasonable assumptions and hence not improvable in general.

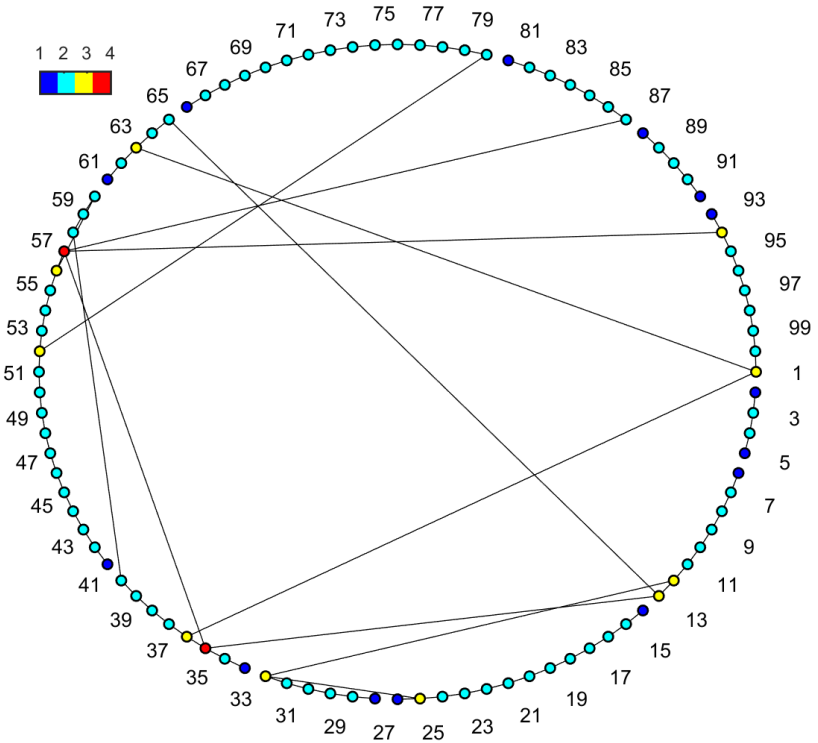
Numerical example

- Test problem: decentralized linear SVM model

$$\min_{\mathbf{x}} \sum_{i=1}^{m} \mathbb{E}_{(u_i,v_i)}[\max\{0, 1 - v_i\langle x_i, u_i\rangle\}]$$

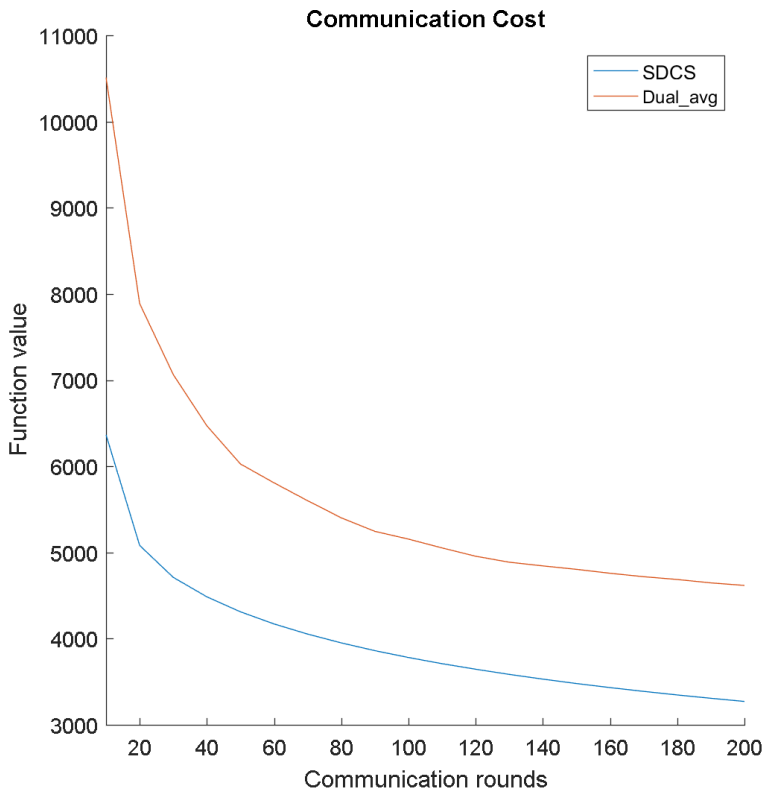
$$\text{s.t.} \quad \mathbf{Lx} = \mathbf{0}.$$

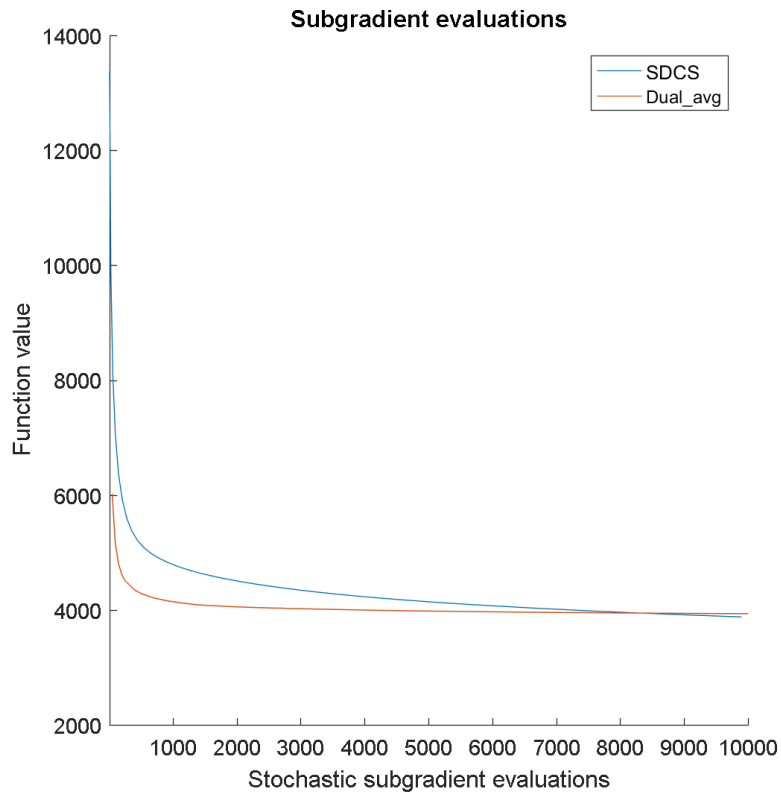
- Network structure: connected graph with 100 nodes.
- Data set: real data set “ijcnn1” from LIBSVM.



The underlying decentralized network

Comparing with distributed dual averaging





Conclusion: SDCS saves inter-node communication rounds while preserving the same order of sampling complexity.

- Centralized SGD
- Random gradient extroplation for federated learning over networks
  - Optimal $\mathcal{O}(\sqrt{mL/\mu}\log(1/\epsilon))$ communication complexity
  - Nearly $\mathcal{O}(1/\epsilon)$ optimal sampling complexity
- Stochastic communication sliding for decentralized learning over networks
  - # stochastic subgradient evaluation is comparable to centralized SGD.
  - # communication rounds is negligible in comparison with # stochastic subgradient evaluation.

Thanks!

- G. Lan and Y. Zhou, “Random gradient extrapolation for distributed and stochastic optimization”, *SIAM Journal on Optimization*, 28(4), 2753-2782, 2018.
- G. Lan, S. Lee and Y. Zhou, “Communication-efficient Algorithms for Decentralized and Stochastic Optimization”, *Mathematical Programming* to appear.