

Optimization Theory

Lecture 14

Fudan University

luoluo@fudan.edu.cn

- 1 Stochastic Variance Reduced Gradient
- 2 Catalyst Acceleration and Direct Acceleration
- 3 Stochastic Recursive Gradient Algorithm

- 1 Stochastic Variance Reduced Gradient
- 2 Catalyst Acceleration and Direct Acceleration
- 3 Stochastic Recursive Gradient Algorithm

Stochastic Variance Reduced Gradient (SVRG)

Algorithm 1 Stochastic Variance Reduced Gradient

```
1: Input:  $\mathbf{x}_0, \eta, m, S$ 
2:  $\tilde{\mathbf{x}}^{(0)} = \mathbf{x}_0$ 
3: for  $s = 0, \dots, S - 1$ 
4:    $\tilde{\mu} = \nabla f(\tilde{\mathbf{x}}^{(s)})$ 
5:    $\mathbf{x}_0 = \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(s)}$ 
6:   for  $t = 0, \dots, m - 1$ 
7:     draw  $t_i$  from  $\{1, \dots, n\}$  uniformly
8:      $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta(\nabla f_{t_i}(\mathbf{x}_t) - \nabla f_{t_i}(\tilde{\mathbf{x}}) + \tilde{\mu}),$ 
9:   end for
10:  Option I:  $\tilde{\mathbf{x}}^{(s+1)} = \mathbf{x}_m$ 
11:  Option II:  $\tilde{\mathbf{x}}^{(s+1)} = \mathbf{x}_t$  for randomly chosen  $t \in \{0, \dots, m - 1\}$ 
12: end for
13: Output:  $\tilde{\mathbf{x}}^{(S)}$ 
```

Stochastic Variance Reduced Gradient (SVRG)

Assume $\eta = \Theta(1/L)$ and m is sufficient large so that

$$\rho = \frac{1}{\mu\eta(1 - 2L\eta)m} + \frac{2L\eta}{1 - 2L\eta} < 1,$$

then SVRG holds that

$$\mathbb{E}[f(\tilde{\mathbf{x}}^{(s)}) - f(\mathbf{x}^*)] \leq \rho^s(f(\tilde{\mathbf{x}}_0) - f(\mathbf{x}^*)).$$

The incremental first-order oracle complexity to achieve

$$\mathbb{E}[f(\tilde{\mathbf{x}}^{(s)}) - f(\mathbf{x}^*)] \leq \epsilon$$

is at most $\mathcal{O}((\kappa + n) \log(1/\epsilon))$.

Algorithm 2 L-SVRG

- 1: **Input:** η , T and p .
 - 2: $\mathbf{x}_0 = \mathbf{w}_0$
 - 3: **for** $t = 0, 1, \dots, T$ **do**
 - 4: $\mathbf{v}_t = \nabla f_{t_i}(\mathbf{x}_t) - \nabla f_{t_i}(\mathbf{w}_t) + \nabla f(\mathbf{w}_t)$
 - 5: $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{v}_t$
 - 6: $\mathbf{w}_{t+1} = \begin{cases} \mathbf{x}_t & \text{with probability } p \\ \mathbf{w}_t & \text{with probability } 1 - p \end{cases}$
 - 7: **end for**
-

- 1 Stochastic Variance Reduced Gradient
- 2 Catalyst Acceleration and Direct Acceleration
- 3 Stochastic Recursive Gradient Algorithm

Comparisons on IFO complexities:

- ① SAG/SVRG/SAGA is better than GD.
- ② SAG/SVRG/SAGA is worse than AGD when $\kappa \geq \Omega(n^2)$.
- ③ The optimal dependency on condition number should be $\sqrt{\kappa}$.

How to accelerate variance reduced methods?

Consider the inexact proximal point iteration

$$\begin{aligned}\mathbf{x}_{t+1} &\approx \text{prox}_{f/\gamma}(\mathbf{x}_t) \\ &= \arg \min_{\mathbf{x} \in \mathbb{R}^d} \left(f(\mathbf{x}) + \frac{\gamma}{2} \|\mathbf{x} - \mathbf{x}_t\|_2^2 \right).\end{aligned}$$

How design the algorithm?

- 1 Select appropriate value of γ .
- 2 Introduce the step of acceleration.

Algorithm 3 Catalyst Acceleration

- 1: **Input:** initial point $\mathbf{x}_0 \in \mathbb{R}^d$, iterations number T , parameters γ and $\alpha_0 > 0$, sequence $\{\epsilon_t\}$, sub-problem solver \mathcal{A} .
 - 2: $q = \mu/(\mu + \gamma)$, $\mathbf{y}_0 = \mathbf{x}_0$
 - 3: **for** $t = 0, 1, \dots, T$ **do**
 - 4: Apply \mathcal{A} to find
$$\mathbf{x}_{t+1} \approx \arg \min_{\mathbf{x} \in \mathbb{R}^d} \left(G_t(\mathbf{x}) \triangleq f(\mathbf{x}) + \frac{\gamma}{2} \|\mathbf{x} - \mathbf{y}_t\|_2^2 \right)$$
such that $G_t(\mathbf{x}_{t+1}) - G_t^* \leq \epsilon_t$
 - 5: Compute $\alpha_t \in (0, 1)$ from equation $\alpha_{t+1}^2 = (1 - \alpha_{t+1})\alpha_t^2 + q\alpha_{t+1}$
 - 6: Compute $\mathbf{y}_{t+1} = \mathbf{x}_{t+1} + \beta_t(\mathbf{x}_{t+1} - \mathbf{x}_t)$, where $\beta_t = \frac{\alpha_t(1 - \alpha_t)}{\alpha_t^2 + \alpha_{t+1}}$
 - 7: **end for**
 - 8: **Output:** \mathbf{x}_T
-

Theorem

Let $\alpha_0 = \sqrt{q}$ with $q = \mu/(\mu + \beta)$ and

$$\epsilon_t = \frac{2}{9}(f(\mathbf{x}_0) - f^*)(1 - \rho)^{t+1} \quad \text{with} \quad \rho < \sqrt{q}.$$

Then Algorithm 3 generates $\{\mathbf{x}_t\}$ such that

$$f(\mathbf{x}_t) - f^* \leq \frac{8(1 - \rho)^t}{(\sqrt{q} - \rho)^2}(f(\mathbf{x}_0) - f^*).$$

A generic framework for acceleration:

- ① Let \mathcal{A} be GD and $\beta = \Theta(L)$, then total FO complexity is

$$\tilde{\mathcal{O}}(\sqrt{\kappa} \log(1/\epsilon)).$$

- ② Let \mathcal{A} be SVRG and $\beta = \Theta(L/n)$, then total IFO complexity is

$$\tilde{\mathcal{O}}(\sqrt{\kappa n} \log(1/\epsilon)), \quad \text{where} \quad \kappa \geq \Omega(n).$$

Algorithm 4 Katyusha

```
1: Input:  $\mathbf{x}_0, \eta, m, S, \tau_1, \tau_2$ 
2:  $\mathbf{y}_0 = \mathbf{z}_0 = \tilde{\mathbf{x}}^{(0)} = \mathbf{x}_0$ 
3: for  $s = 0, \dots, S - 1$ 
4:    $\tilde{\boldsymbol{\mu}}^{(s)} = \nabla f(\tilde{\mathbf{x}}^{(s)})$ 
5:   for  $t = 0, \dots, m - 1$ 
6:      $k = sm + t$ 
7:      $\mathbf{x}_{k+1} = \tau_1 \mathbf{z}_k + \tau_2 \tilde{\mathbf{x}}^{(s)} + (1 - \tau_1 - \tau_2) \mathbf{y}_k$ 
8:     draw  $i_k$  from  $\{1, \dots, n\}$  uniformly
9:      $\mathbf{z}_{k+1} = \mathbf{z}_k - \eta(\nabla f_{i_k}(\mathbf{x}_t) - \nabla f_{i_k}(\tilde{\mathbf{x}}) + \tilde{\boldsymbol{\mu}}^{(s)})$ ,
10:     $\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \tau_1(\mathbf{z}_{k+1} - \mathbf{z}_k)$ ,
11:   end for
12:    $\tilde{\mathbf{x}}^{(s+1)} = \left( \sum_{j=0}^{m-1} (1 + \eta \mu)^j \right)^{-1} \sum_{j=0}^{m-1} (1 + \eta \mu)^j \mathbf{y}_{sm+j+1}$ 
13: end for
14: Output:  $\tilde{\mathbf{x}}^{(S)}$ 
```

Direct Acceleration: Katyusha

Katyusha outputs $\tilde{\mathbf{x}}^{(S)}$ satisfying $\mathbb{E}[f(\tilde{\mathbf{x}}^{(S)})] - f^* \leq \epsilon$ within

- ① $\mathcal{O}((n + \sqrt{\kappa n}) \log(1/\epsilon))$ IFO complexity for strongly convex objective;
- ② $\mathcal{O}(n \log(1/\epsilon) + \sqrt{nL/\epsilon})$ IFO complexity for convex objective.

The above results achieve the near optimal IFO complexities.

- 1 Stochastic Variance Reduced Gradient
- 2 Catalyst Acceleration and Direct Acceleration
- 3 Stochastic Recursive Gradient Algorithm

Stochastic Recursive Gradient Algorithm

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH GROUP
LEHIGH UNIVERSITY

ISE
Industrial and Systems Engineering

Lam Nguyen
(Lehigh)



SARAH.

**A Novel Method for Machine Learning Problems Using
StochAstic Recursive GrAdient AlgoritHm**

Martin Takáč

 **LEHIGH**
UNIVERSITY.

August 8, 2017

Poster: Tue Aug 8th @ Gallery #48

 Jie Liu
(Lehigh)

 Katya Scheinberg
(Lehigh)

Stochastic Recursive Gradient Algorithm (SARAH)

Algorithm 5 Stochastic Variance Reduced Gradient

```
1: Input:  $\mathbf{x}_0, \eta, m, S$ 
2:  $\tilde{\mathbf{x}}^{(0)} = \mathbf{x}_0$ 
3: for  $s = 0, \dots, S - 1$ 
4:    $\mathbf{v}_0 = \nabla f(\tilde{\mathbf{x}}^{(s)})$ 
5:    $\mathbf{x}_0 = \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(s)}$ 
6:   for  $t = 0, \dots, m - 1$ 
7:     draw  $t_i$  from  $\{1, \dots, n\}$  uniformly
8:      $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{v}_t$ 
9:      $\mathbf{v}_{t+1} = \nabla f_{t_i}(\mathbf{x}_{t+1}) - \nabla f_{t_i}(\mathbf{x}_t) + \mathbf{v}_t$ 
10:  end for
11:   $\tilde{\mathbf{x}}^{(s+1)} = \mathbf{x}_t$  for randomly chosen  $t \in \{0, \dots, m - 1\}$ 
12: end for
13: Output:  $\tilde{\mathbf{x}}^{(S)}$ 
```

Stochastic Recursive Gradient Algorithm (SARAH)

SARAH outputs $\tilde{\mathbf{x}}^{(S)}$ satisfying $\mathbb{E} \|\nabla f(\tilde{\mathbf{x}}^{(S)})\|_2 \leq \epsilon$ within

- ① $\mathcal{O}((n + \kappa) \log(1/\epsilon))$ IFO complexity for strongly convex objective;
- ② $\mathcal{O}((n + L/\epsilon^2) \log(1/\epsilon))$ IFO complexity for convex objective.

The more interesting result is in the nonconvex optimization:

- ① Cong Fang, Chris Junchi Li, Zhouchen Lin, Tong Zhang.
SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *NeurIPS* 2018.

SGD for Nonconvex Optimization

We consider the stochastic optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) \triangleq \mathbb{E}_{\xi}[F(\mathbf{x}; \xi)],$$

where $f(\mathbf{x})$ is L -smooth and lower bounded, and each $F(\mathbf{x}; \xi)$ is differentiable.

We suppose there exists $\sigma > 0$ such that

$$\mathbb{E} \|\nabla F(\mathbf{x}; \xi) - \nabla f(\mathbf{x})\|_2^2 \leq \sigma^2$$

for any $\mathbf{x} \in \mathbb{R}^d$. The SGD iteration

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \cdot \frac{1}{b} \sum_{i=1}^b \nabla F(\mathbf{x}_t; \xi_{t_i})$$

with $\xi_{t_i} \stackrel{\text{i.i.d.}}{\sim} \mathcal{D}$ can find an ϵ -stationary point of $f(\cdot)$ within

$$\mathcal{O}(L\sigma^2\epsilon^{-4})$$

stochastic first-order oracle (SFO) complexity in expectation.

SARAH/SPIDER for Nonconvex Optimization

We consider the L -average smooth function, i.e. there exists $L > 0$ such that

$$\mathbb{E} \|\nabla F(\mathbf{x}; \xi) - \nabla F(\mathbf{y}; \xi)\|_2^2 \leq L^2 \|\mathbf{x} - \mathbf{y}\|_2^2$$

for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

The algorithms with stochastic recursive gradient require

$$\mathcal{O}(\sigma^2 \epsilon^{-2} + L \sigma^2 \epsilon^{-3})$$

SFO complexity to find an ϵ -stationary point.

SARAH/SPIDER for Nonconvex Optimization

We consider the finite-sum problem

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n} \sum_{i=1}^n f_i(\mathbf{x}).$$

Under the L -average smooth assumption, the algorithms with stochastic recursive gradient require

$$\mathcal{O}(n + L\sqrt{n}\epsilon^{-2})$$

SFO complexity to find an ϵ -stationary point.

Algorithm 6 ProbAbilistic Gradient Estimator (PAGE)

- 1: **Input:** η , T , b_0 , b and p .
 - 2: $\mathcal{S}_0 = \{\xi_1, \dots, \xi_0\}$, where $\xi_i \stackrel{\text{i.i.d}}{\sim}$ sampled from \mathcal{D}
 - 3: $\mathbf{v}_0 = \frac{1}{b_0} \sum_{\xi \in \mathcal{S}_0} \nabla F(\mathbf{x}_0; \xi)$
 - 4: **for** $t = 0, 1, \dots, T$ **do**
 - 5: $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{v}_t$
 - 6: $\mathcal{S}_{t+1} = \{\xi_1, \dots, \xi_0\}$, where $\xi_i \stackrel{\text{i.i.d}}{\sim}$ sampled from \mathcal{D}
 - 7:
$$\mathbf{v}_{t+1} = \begin{cases} \frac{1}{b_0} \sum_{\xi \in \mathcal{S}_{t+1}} \nabla F(\mathbf{x}_{t+1}; \xi) & \text{with probability } 1 - p \\ \mathbf{v}_t + \frac{1}{b} \sum_{\xi \in \mathcal{S}_{t+1}} (\nabla F(\mathbf{x}_{t+1}; \xi) - \nabla F(\mathbf{x}_t; \xi)) & \text{with probability } p \end{cases}$$
 - 8: **end for**
 - 9: $\mathbf{x}_{\text{out}} = \mathbf{x}_t$ for randomly chosen $t \in \{0, \dots, T-1\}$
-