Optimization Theory

Lecture 14

Fudan University

luoluo@fudan.edu.cn

Outline

Outline

Stochastic Variance Reduced Gradient (SVRG)

Algorithm 1 Stochastic Variance Reduced Gradient

Input: $\mathbf{x}_0$, η, m, S
$\tilde{\mathbf{x}}^{(0)} = \mathbf{x}_0$
for $s = 0, \dots, S-1$
    $\tilde{\boldsymbol{\mu}} = \nabla f(\tilde{\mathbf{x}}^{(s)})$
    $\mathbf{x}_0 = \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(s)}$
    for $t = 0, \dots, m-1$
        draw t_i from $\{1, \dots, n\}$ uniformly
        $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta(\nabla f_{t_i}(\mathbf{x}_t) - \nabla f_{t_i}(\tilde{\mathbf{x}}) + \tilde{\boldsymbol{\mu}})$,
    end for
    Option I: $\tilde{\mathbf{x}}^{(s+1)} = \mathbf{x}_m$
    Option II: $\tilde{\mathbf{x}}^{(s+1)} = \mathbf{x}_t$ for randomly chosen $t \in \{0, \dots, m-1\}$
end for
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

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

The incremental first-order oracle complexity to achieve

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

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

Loopless SVRG

Algorithm 2 L-SVRG

Input: η, T and p.

$\mathbf{x}_0 = \mathbf{w}_0$

for $t = 0, 1, \dots, T$ **do**

  $\mathbf{v}_t = \nabla f_{t_i}(\mathbf{x}_t) - \nabla f_{t_i}(\mathbf{w}_t) + \nabla f(\mathbf{w}_t)$

  $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{v}_t$

  $$\mathbf{w}_{t+1} = \begin{cases} \mathbf{x}_t & \text{with probability } p \\ \mathbf{w}_t & \text{with probability } 1-p \end{cases}$$

end for

Outline

Variance Reduction

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?

Catalyst Acceleration

Consider the inexact proximal point iteration

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

How design the algorithm?

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

Catalyst Acceleration

Algorithm 3 Catalyst Acceleration

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 $\mathcal{A}$.

$q = \mu/(\mu + \gamma)$, $\mathbf{y}_0 = \mathbf{x}_0$

for $t = 0, 1, \ldots, T$ do

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 $G_t(\mathbf{x}_{t+1}) - G_t^* \le \epsilon_t$

Compute $\alpha_t \in (0, 1)$ from equation $\alpha_{t+1}^2 = (1 - \alpha_{t+1})\alpha_t^2 + q\alpha_{t+1}$

Compute $\mathbf{y}_{t+1} = \mathbf{x}_{t+1} + \beta_t(\mathbf{x}_{t+1} - \mathbf{x}_t)$, where $\beta_t = \dfrac{\alpha_t(1 - \alpha_t)}{\alpha_t^2 + \alpha_{t+1}}$

end for

Output: $\mathbf{x}_T$

Catalyst Acceleration

Theorem

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

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

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

Direct Acceleration: Katyusha

Algorithm 4 Katyusha

Input: $\mathbf{x}_0$, η, m, S, τ_1, τ_2

$\mathbf{y}_0 = \mathbf{z}_0 = \tilde{\mathbf{x}}^{(0)} = \mathbf{x}_0$

for $s = 0, \dots, S-1$

  $\tilde{\boldsymbol{\mu}}^{(s)} = \nabla f(\tilde{\mathbf{x}}^{(s)})$

  for $t = 0, \dots, m-1$

    $k = sm + t$

    $\mathbf{x}_{k+1} = \tau_1 \mathbf{z}_k + \tau_2 \tilde{\mathbf{x}}^{(s)} + (1 - \tau_1 - \tau_2)\mathbf{y}_k$

    draw i_k from $\{1, \dots, n\}$ uniformly

    $\mathbf{z}_{k+1} = \mathbf{z}_k - \eta(\nabla f_{i_k}(\mathbf{x}_t) - \nabla f_{i_k}(\tilde{\mathbf{x}}) + \tilde{\boldsymbol{\mu}}^{(s)})$,

    $\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \tau_1(\mathbf{z}_{k+1} - \mathbf{z}_k)$,

  end for

  $\tilde{\mathbf{x}}^{(s+1)} = \left(\sum_{j=0}^{m-1}(1+\eta\mu)^j\right)^{-1} \sum_{j=0}^{m-1}(1+\eta\mu)^j \mathbf{y}_{sm+j+1}$

end for

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

The above results achieve the near optimal IFO complexities.

Outline

Stochastic Recursive Gradient Algorithm

Stochastic Recursive Gradient Algorithm (SARAH)

Algorithm 5 Stochastic Variance Reduced Gradient

Input: $\mathbf{x}_0$, η, m, S
$\tilde{\mathbf{x}}^{(0)} = \mathbf{x}_0$
for $s = 0, \dots, S-1$
    $\mathbf{v}_0 = \nabla f(\tilde{\mathbf{x}}^{(s)})$
    $\mathbf{x}_0 = \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{(s)}$
    for $t = 0, \dots, m-1$
        draw t_i from $\{1, \dots, n\}$ uniformly
        $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{v}_t$
        $\mathbf{v}_{t+1} = \nabla f_{t_i}(\mathbf{x}_{t+1}) - \nabla f_{t_i}(\mathbf{x}_t) + \mathbf{v}_t$
    end for
    $\tilde{\mathbf{x}}^{(s+1)} = \mathbf{x}_t$ for randomly chosen $t \in \{0, \dots, m-1\}$
end for
Output: $\tilde{\mathbf{x}}^{(S)}$

Stochastic Recursive Gradient Algorithm (SARAH)

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

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

The more interesting result is in the nonconvex optimization.

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

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

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 $\xi_{t_i} \overset{\text{i.i.d}}{\sim} \mathcal{D}$ 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 for Nonconvex Optimization

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

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

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 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 $\xi_i \overset{\text{i.i.d}}{\sim}$ sampled from $\mathcal{D}$

3: $\mathbf{v}_0 = \dfrac{1}{b_0} \sum_{\xi \in \mathcal{S}_0} \nabla F(\mathbf{x}_0; \xi)$

4: **for** $t = 0, 1, \dots, T$ do

5: $\quad \mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{v}_t$

6: $\quad \mathcal{S}_{t+1} = \{\xi_1, \dots, \xi_0\}$, where $\xi_i \overset{\text{i.i.d}}{\sim}$ sampled from $\mathcal{D}$

7:
$$\mathbf{v}_{t+1} = \begin{cases} \dfrac{1}{b_0} \displaystyle\sum_{\xi \in \mathcal{S}_{t+1}} \nabla F(\mathbf{x}_{t+1}; \xi) & \text{with probability } 1-p \\ \mathbf{v}_t + \dfrac{1}{b} \displaystyle\sum_{\xi \in \mathcal{S}_{t+1}} (\nabla F(\mathbf{x}_{t+1}; \xi) - \nabla F(\mathbf{x}_t; \xi)) & \text{with probability } p \end{cases}$$

8: **end for**

9: $\mathbf{x}_{\text{out}} = \mathbf{x}_t$ for randomly chosen $t \in \{0, \dots, T-1\}$