Stochastically Controlled Stochastic Gradient (SCSG) Method

Lihua Lei

joint works with Cheng Ju, Jianbo Chen and Michael Jordan

March 14, UC Davis

Table of Contents

Table of Contents

Working Horse of Modern Machine Learning

Working Horse of Modern Machine Learning

Working Horse of Modern Machine Learning

Herbert Robbins (1915-2001)

Working Horse of Modern Machine Learning

Herbert Robbins (1915-2001)

A STOCHASTIC APPROXIMATION METHOD[1]

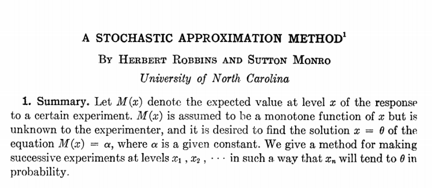
By Herbert Robbins and Sutton Monro

University of North Carolina

1. Summary. Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant. We give a method for making successive experiments at levels $x_1, x_2, \cdots$ in such a way that x_n will tend to θ in probability.

Working Horse of Modern Machine Learning

Herbert Robbins (1915-2001)

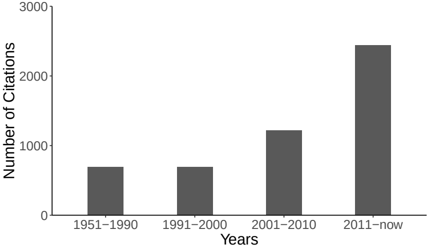


A STOCHASTIC APPROXIMATION METHOD[1]

By Herbert Robbins and Sutton Monro

University of North Carolina

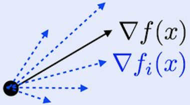
1. Summary. Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant. We give a method for making successive experiments at levels $x_1, x_2, \cdots$ in such a way that x_n will tend to θ in probability.



Robbins-Monro Algorithm/ Stochastic Gradient Descent

Finite sums

$$f(x) \stackrel{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$
$$\nabla f(x) = \frac{1}{n}\sum_i \nabla f_i(x)$$

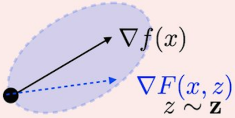


Draw $i \in \{1, \dots, n\}$ uniformly.

$$x_{k+1} = x_k - \tau_k \nabla f_i(x_k)$$

Expectation

$$f(x) \stackrel{\text{def.}}{=} \mathbb{E}_{\mathbf{z}}(f(x, \mathbf{z}))$$
$$\nabla f(x) = \mathbb{E}_{\mathbf{z}}(\nabla F(x, \mathbf{z}))$$



Draw $z \sim \mathbf{z}$

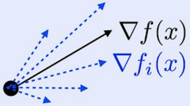
$$x_{k+1} = x_k - \tau_k \nabla F(x, z)$$

Robbins-Monro Algorithm/ Stochastic Gradient Descent

Finite sums

$$f(x) \stackrel{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

$$\nabla f(x) = \frac{1}{n}\sum_i \nabla f_i(x)$$



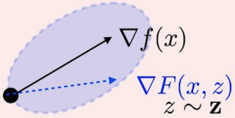
Draw $i \in \{1, \ldots, n\}$ uniformly.

$$x_{k+1} = x_k - \tau_k \nabla f_i(x_k)$$

Expectation

$$f(x) \stackrel{\text{def.}}{=} \mathbb{E}_{\mathbf{z}}(f(x, \mathbf{z}))$$

$$\nabla f(x) = \mathbb{E}_{\mathbf{z}}(\nabla F(x, \mathbf{z}))$$



Draw $z \sim \mathbf{z}$

$$x_{k+1} = x_k - \tau_k \nabla F(x, z)$$

Theorem 1 (Robbins and Monro, 1951).

Let $\sum_k \tau_k = \infty, \sum_k \tau_k^2 < \infty$. *Then under technical conditions,*

$$x_k \stackrel{a.s.}{\rightarrow} \arg\min f(x)$$

Optimization in Machine Learning

Assume $(y_i, z_i) \overset{i.i.d.}{\sim} G$. The goal is to learn a map $h(\cdot; x)$ from a function class parametrized by $x \in \mathbb{R}^d$, such that $h(z; x)$ is a good "guess" of y.

Optimization in Machine Learning

Assume $(y_i, z_i) \overset{i.i.d.}{\sim} G$. The goal is to learn a map $h(\cdot; x)$ from a function class parametrized by $x \in \mathbb{R}^d$, such that $h(z; x)$ is a good "guess" of y.

Empirical Risk Minimization

$$\min_x \hat{f}(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(z_i; x))$$

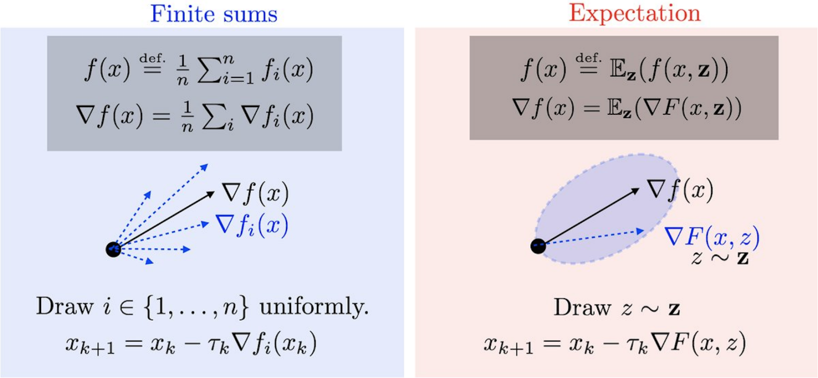
- batch learning;
- observed objective;
- training loss.

Stochastic Optimization

$$\min_x f(x) \triangleq \mathbb{E}_G \, \ell(Y, h(Z; x)).$$

- online/streaming learning;
- unobserved objective;
- testing loss.

Finite-Sum Optimization V.S. Stochastic Optimization

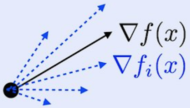


Finite-Sum Optimization V.S. Stochastic Optimization

Finite sums

$$f(x) \stackrel{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

$$\nabla f(x) = \frac{1}{n}\sum_i \nabla f_i(x)$$



Draw $i \in \{1, \dots, n\}$ uniformly.

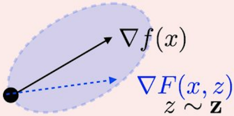
$$x_{k+1} = x_k - \tau_k \nabla f_i(x_k)$$

- can access each data for multiple times;
- full gradients can be computed with finite cost

Expectation

$$f(x) \stackrel{\text{def.}}{=} \mathbb{E}_{\mathbf{z}}(f(x, \mathbf{z}))$$

$$\nabla f(x) = \mathbb{E}_{\mathbf{z}}(\nabla F(x, \mathbf{z}))$$



Draw $z \sim \mathbf{z}$

$$x_{k+1} = x_k - \tau_k \nabla F(x, z)$$

- must access a "fresh" sample at each step;
- full gradients cannot be computed with finite cost

Finite-Sum Optimization V.S. Stochastic Optimization

- Finite-sum optimization can be regarded as a special case of stochastic optimization:

$$\frac{1}{n}\sum_{i=1}^{n} f_i(x) = \mathbb{E}_{z\sim U([n])} f_z(x);$$

- Any algorithm that works for stochastic optimization also works for finite-sum optimization, with same complexity.

Finite-Sum Optimization V.S. Stochastic Optimization

- Finite-sum optimization has more structure and more applications than stochastic optimization;
- (y_i, z_i) are not i.i.d. or even not random:
  - ubiquitous in statistical inference for fixed designs;
  - stochastic optimization even not defined
- objective involving pairwise comparison:
  - $$\begin{aligned} f(x) &= \mathbb{E}F(x; (y_1, z_1), (y_2, z_2)) \\ &\approx \frac{1}{n(n-1)} \sum_{i \neq j} F(x; (y_i, z_i), (y_j, z_j)) \end{aligned}$$
  - metric learning, preference elicitation, sport analysis...

SGD: A Brief Overview

Algorithm (for finite-sum optimization and stochastic optimization):

$$\mathrm{SGD}: \; x_{t+1} = x_t - \eta_t g_t, \quad \mathbb{E} g_t = \nabla f(x_t)$$

Main assumption (smoothness):

$$\mu I \preceq \nabla^2 f(x) \preceq LI, \quad (L > 0)$$

SGD: A Brief Overview

Algorithm (for finite-sum optimization and stochastic optimization):

$$\text{SGD}: \quad x_{t+1} = x_t - \eta_t g_t, \quad \mathbb{E} g_t = \nabla f(x_t)$$

Main assumption (smoothness):

$$\mu I \preceq \nabla^2 f(x) \preceq LI, \quad (L > 0)$$

- strongly convex $(\mu > 0, \kappa = L/\mu)$;
- non-strongly convex $(\mu = 0)$;
- non-convex: $(\mu = -L)$.

type of objective	η_t	goal	complexity
strongly convex	$O\left(\frac{1}{\mu t}\right)$	$\mathbb{E}(f(x) - f(x^*)) \leq \epsilon$	$O\left(\frac{1}{\mu \epsilon}\right)$
convex	$O\left(\frac{1}{\sqrt{t}}\right)$	$\mathbb{E}(f(x) - f(x^*)) \leq \epsilon$	$O\left(\frac{1}{\epsilon^2}\right)$
non-convex	$O\left(\frac{1}{\sqrt{t}}\right)$	$\mathbb{E}\|\nabla f(x)\|^2 \leq \epsilon$	$O\left(\frac{1}{\epsilon^2}\right)$

SVRG: A Brief Overview

Algorithms (for finite-sum optimization):

$\mathrm{SAG}, \mathrm{SAGA}, \mathrm{SVRG}, \mathrm{SDCA}, \mathrm{APCG}, \mathrm{SPDC}, \mathrm{Katyusha}, \mathrm{Natasha} \ldots$

SVRG: A Brief Overview

Algorithms (for finite-sum optimization):

$\mathrm{SAG}, \mathrm{SAGA}, \mathrm{SVRG}, \mathrm{SDCA}, \mathrm{APCG}, \mathrm{SPDC}, \mathrm{Katyusha}, \mathrm{Natasha}\ldots$

type of objective	algorithm	complexity
strongly convex	[JZ13]	$O\left((n+\kappa)\log\left(\frac{1}{\epsilon}\right)\right)$
convex	[AZY15]	$O\left(n\log\left(\frac{1}{\epsilon}\right)+\frac{1}{\epsilon}\right)$
non-convex	[RHS+16]	$O\left(n+\frac{n^{2/3}}{\epsilon}\right)$

SGD v.s. SVRG: An Brief Comparison

type of objective	complexity (SGD)	complexity (SVRG)
strongly convex	$O\left(\frac{1}{\mu\epsilon}\right)$	$O\left((n+\kappa)\log(\frac{1}{\epsilon})\right)$
convex	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n\log(\frac{1}{\epsilon})+\frac{1}{\epsilon}\right)$
non-convex	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n+\frac{n^{2/3}}{\epsilon}\right)$

SGD v.s. SVRG: An Brief Comparison

type of objective	complexity (SGD)	complexity (SVRG)
strongly convex	$O\left(\frac{1}{\mu\epsilon}\right)$	$O\left((n+\kappa)\log(\frac{1}{\epsilon})\right)$
convex	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n\log(\frac{1}{\epsilon})+\frac{1}{\epsilon}\right)$
non-convex	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n+\frac{n^{2/3}}{\epsilon}\right)$

- SVRG only works for finite-sums while SGD works for both;

SGD v.s. SVRG: An Brief Comparison

type of objective	complexity (SGD)	complexity (SVRG)
strongly convex	$O\left(\frac{1}{\mu\epsilon}\right)$	$O\left((n+\kappa)\log(\frac{1}{\epsilon})\right)$
convex	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n\log(\frac{1}{\epsilon})+\frac{1}{\epsilon}\right)$
non-convex	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n+\frac{n^{2/3}}{\epsilon}\right)$

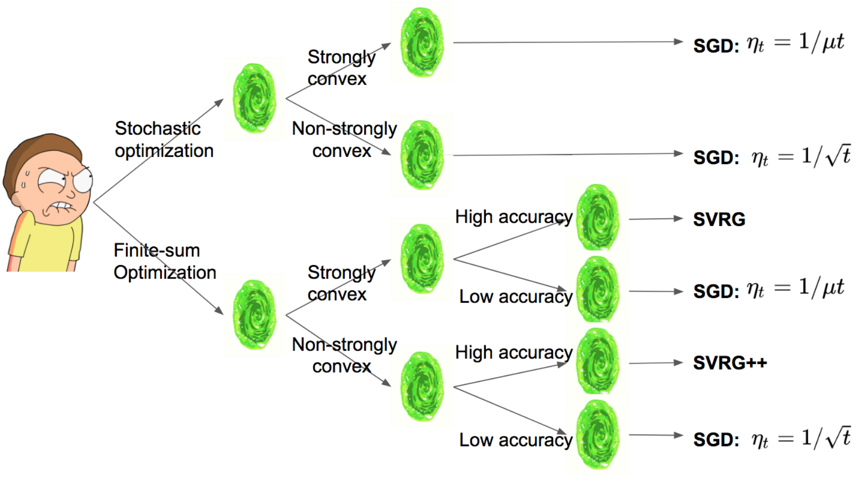
- SVRG only works for finite-sums while SGD works for both;
- Both SGD and SVRG need different settings for strongly/non-strongly/non-convex objectives;

SGD v.s. SVRG: An Brief Comparison

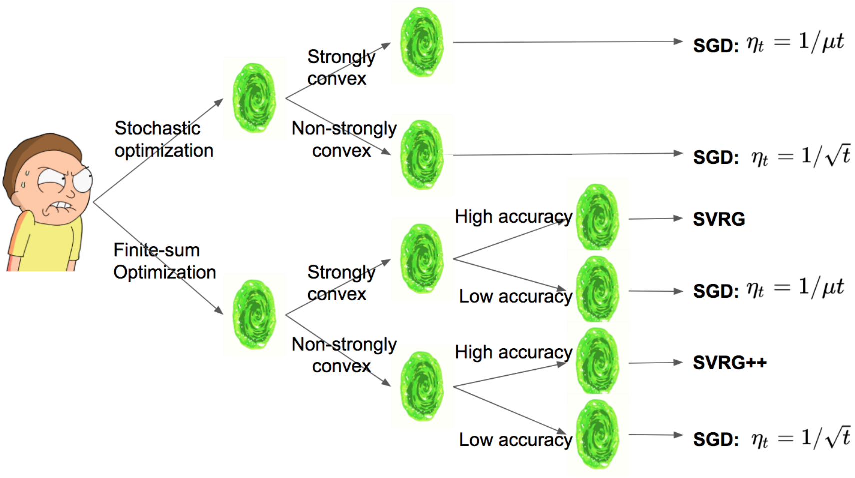
type of objective	complexity (SGD)	complexity (SVRG)
strongly convex	$O\left(\frac{1}{\mu\epsilon}\right)$	$O\left((n+\kappa)\log(\frac{1}{\epsilon})\right)$
convex	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n\log(\frac{1}{\epsilon})+\frac{1}{\epsilon}\right)$
non-convex	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n+\frac{n^{2/3}}{\epsilon}\right)$

- SVRG only works for finite-sums while SGD works for both;
- Both SGD and SVRG need different settings for strongly/non-strongly/non-convex objectives;
- SVRG has better dependence on ϵ but may be worse than SGD for low accuracy computation where $\frac{1}{\epsilon} \ll n$.

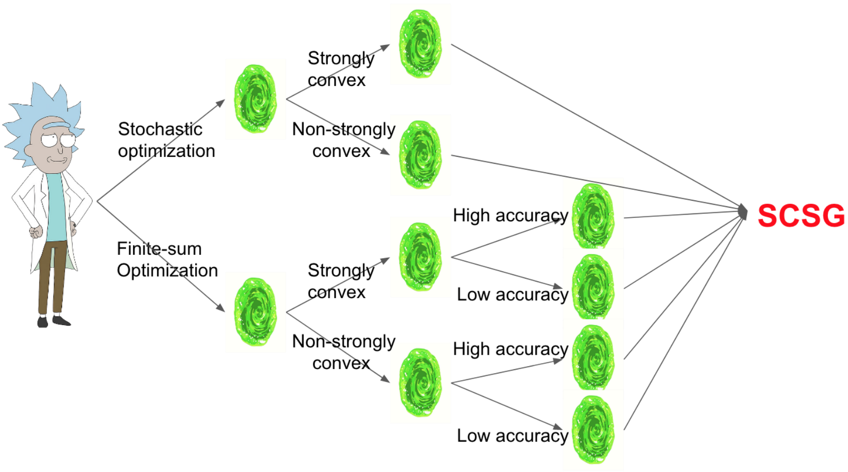

Finite-Sum optimization
Stochastic optimization
Strongly Convex
Non-strongly Convex
High Accuracy
Low Accuracy

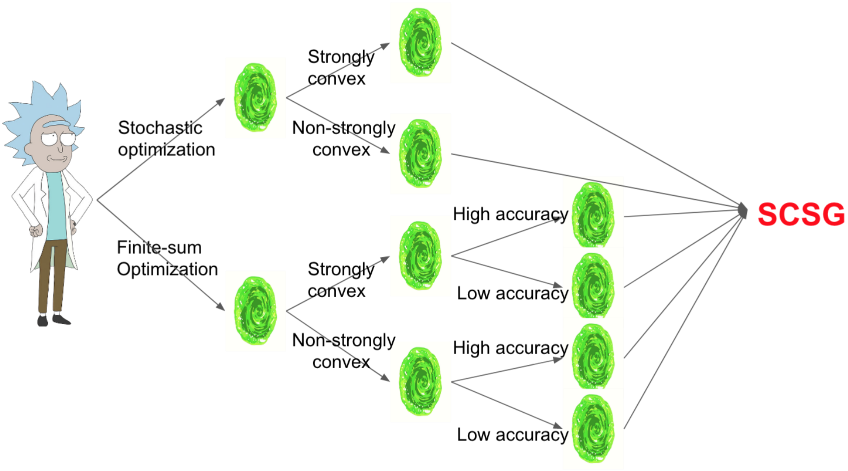


Stochastic optimization
Finite-sum Optimization
Strongly convex
Non-strongly convex
Strongly convex
Non-strongly convex
High accuracy
Low accuracy
High accuracy
Low accuracy
SGD: $\eta_t = 1/\mu t$
SGD: $\eta_t = 1/\sqrt{t}$
SVRG
SGD: $\eta_t = 1/\mu t$
SVRG++
SGD: $\eta_t = 1/\sqrt{t}$



Oh Geez, why is life so complicated ?


Stochastic optimization
Finite-sum Optimization
Strongly convex
Non-strongly convex
Strongly convex
Non-strongly convex
High accuracy
Low accuracy
High accuracy
Low accuracy
SCSG



Hey Morty, let's adventure in the new world!

Table of Contents

Stochastic Variance Reduced Gradient (SVRG) Method

SGD with constant stepsize:

$$x_{t+1} = x_t - \eta g_t, \quad \mathbb{E} g_t = \nabla f(x_t).$$

It does not converge because $\mathrm{Var}(x_{t+1} - x_t) = \eta^2 \, \mathrm{Var}(g_t) \not\to 0$.

Stochastic Variance Reduced Gradient (SVRG) Method

SGD with constant stepsize:

$$x_{t+1} = x_t - \eta g_t, \quad \mathbb{E} g_t = \nabla f(x_t).$$

It does not converge because $\mathrm{Var}(x_{t+1} - x_t) = \eta^2 \,\mathrm{Var}(g_t) \not\to 0$.

Idea: find an extra term h_t with

$$x_{t+1} = x_t - \eta(g_t - h_t), \quad \mathbb{E} h_t = 0, \quad \mathrm{Var}(g_t - h_t) \to 0.$$

Stochastic Variance Reduced Gradient (SVRG) Method

SGD with constant stepsize:

$$x_{t+1} = x_t - \eta g_t, \quad \mathbb{E} g_t = \nabla f(x_t).$$

It does not converge because $\mathrm{Var}(x_{t+1} - x_t) = \eta^2 \mathrm{Var}(g_t) \not\to 0$.

Idea: find an extra term h_t with

$$x_{t+1} = x_t - \eta(g_t - h_t), \quad \mathbb{E} h_t = 0, \quad \mathrm{Var}(g_t - h_t) \to 0.$$

SVRG: $h_t = g_{t'} - \mathbb{E} g_{t'}$ for some $t' \leq t$. Then

$$g_t - g_{t'} \to 0, \quad t, t' \to \infty.$$

SVRG

Consider finite-sum optimization:

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

SVRG (Outer Loop)

Inputs: $\tilde{x}_0, \{\eta_j\}, \{m_j\}, T$

for $j = 1, 2, \cdots, T$ **do**
  $\tilde{x}_j \leftarrow$ **SVRGEpoch**$(\tilde{x}_{j-1}, \eta_j, m_j)$
end for

Output: $\tilde{x}_T$

SVRGEpoch (Inner Loop)

Inputs: x_0, η, m

$g \leftarrow \frac{1}{n} \sum_{i \in [n]} f_i'(x_0)$
Generate $N \sim U([m])$
for $k = 1, 2, \cdots, N$ **do**
  Randomly pick $i \in [n]$
  $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$
  $x \leftarrow x - \eta\nu$
end for

Output: x

SVRG and Its Variants

type	algorithm	η_j	m_j	complexity
strongly convex	[JZ13]	$O\left(\frac{1}{L}\right)$	$O(\kappa)$	$O\left((n+\kappa)\log\left(\frac{1}{\epsilon}\right)\right)$
convex	[AZY15]	$O\left(\frac{1}{L}\right)$	2^j	$O\left(n\log\left(\frac{1}{\epsilon}\right)+\frac{1}{\epsilon}\right)$
non-convex	[RHS+16]	$O\left(\frac{1}{Ln^{2/3}}\right)$	$O(n)$	$O\left(n+\frac{n^{2/3}}{\epsilon}\right)$

SVRG and Its Variants

type	algorithm	η_j	m_j	complexity
strongly convex	[JZ13]	$O\left(\frac{1}{L}\right)$	$O(\kappa)$	$O\left((n+\kappa)\log\left(\frac{1}{\epsilon}\right)\right)$
convex	[AZY15]	$O\left(\frac{1}{L}\right)$	2^j	$O\left(n\log\left(\frac{1}{\epsilon}\right)+\frac{1}{\epsilon}\right)$
non-convex	[RHS$^+$16]	$O\left(\frac{1}{Ln^{2/3}}\right)$	$O(n)$	$O\left(n+\frac{n^{2/3}}{\epsilon}\right)$

Theoretical concerns:

- SVRG does not work for stochastic optimization, in which the full gradient is inaccessible;
- SVRG outperforms SGD only if ϵ is small;
- SVRG requires the knowledge of κ to achieve the fast rate for strongly-convex objectives.

Practical Concern of SVRG

Computing full gradient is too costly!

SCSG in Finite-Sum Optimization

SVRGEpoch

Inputs: x_0, η, m

$\mathcal{I} \leftarrow [n]$

$g \leftarrow \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} f_i'(x_0)$

Gen. $N \sim U([m])$

for $k = 1, 2, \cdots, N$ **do**

  Randomly pick $i \in [n]$

  $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$

  $x \leftarrow x - \eta\nu$

end for

SCSG in Finite-Sum Optimization

SVRGEpoch

Inputs: x_0, η, m

$\mathcal{I} \leftarrow [n]$
$g \leftarrow \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} f_i'(x_0)$
Gen. $N \sim U([m])$
for $k = 1, 2, \cdots, N$ **do**
    Randomly pick $i \in [n]$
    $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$
    $x \leftarrow x - \eta \nu$
end for

SCSGEpoch

Inputs: x_0, η, B, m

SCSG in Finite-Sum Optimization

SVRGEpoch

Inputs: x_0, η, m

1: $\mathcal{I} \leftarrow [n]$

2: $g \leftarrow \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} f_i'(x_0)$

3: Gen. $N \sim U([m])$

4: **for** $k = 1, 2, \cdots, N$ **do**

5: Randomly pick $i \in [n]$

6: $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$

7: $x \leftarrow x - \eta\nu$

8: **end for**

SCSGEpoch

Inputs: x_0, η, B, m

1: Randomly pick $\mathcal{I}$ with size B

2: $g \leftarrow \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} f_i'(x_0)$

SCSG in Finite-Sum Optimization

SVRGEpoch

Inputs: x_0, η, m

$\mathcal{I} \leftarrow [n]$
$g \leftarrow \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} f_i'(x_0)$
Gen. $N \sim U([m])$
for $k = 1, 2, \cdots, N$ **do**
    Randomly pick $i \in [n]$
    $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$
    $x \leftarrow x - \eta\nu$
end for

SCSGEpoch

Inputs: x_0, η, B, m

Randomly pick $\mathcal{I}$ with size B
$g \leftarrow \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} f_i'(x_0)$
Gen. $N \sim \mathrm{Geo}$ with mean m

$$N \sim \mathrm{Geo}(\gamma) \text{ iff } P(N = k) = (1-\gamma)\gamma^k \ (k \geq 0) \Longrightarrow \mathbb{E}N = \frac{\gamma}{1-\gamma}$$

SCSG in Finite-Sum Optimization

SVRGEpoch

Inputs: x_0, η, m

$\mathcal{I} \leftarrow [n]$
$g \leftarrow \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} f_i'(x_0)$
Gen. $N \sim U([m])$
for $k = 1, 2, \cdots, N$ **do**
    Randomly pick $i \in [n]$
    $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$
    $x \leftarrow x - \eta\nu$
end for

SCSGEpoch

Inputs: x_0, η, B, m

Randomly pick $\mathcal{I}$ with size B
$g \leftarrow \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} f_i'(x_0)$
Gen. $N \sim \mathrm{Geo}$ with mean m
for $k = 1, 2, \cdots, N$ **do**
    Randomly pick $i \in [n]$
    $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$
    $x \leftarrow x - \eta\nu$
end for

$$N \sim \mathrm{Geo}(\gamma) \text{ iff } P(N = k) = (1-\gamma)\gamma^k \ (k \geq 0) \Longrightarrow \mathbb{E}N = \frac{\gamma}{1-\gamma}$$

SCSG in Stochastic Optimization

SCSGEpoch (finite-sum)

Obj.: $f(x) = \frac{1}{n}\sum_{i\in[n]} f_i(x)$

Inputs: x_0, η, B, m

Randomly pick $\mathcal{I}$ with size B

$g \leftarrow \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} f_i'(x_0)$

Gen. $N \sim \mathrm{Geo}$ with mean m

for $k = 1, 2, \cdots, N$ **do**

  Randomly pick $i \in [n]$

  $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$

  $x \leftarrow x - \eta\nu$

end for

SCSGEpoch (expectation)

Obj.: $f(x) = \mathbb{E}_{\xi\sim G} F_\xi(x)$

Inputs: x_0, η, B, m

Gen. $\{\xi_i\}_{i=1}^B \overset{i.i.d.}{\sim} G$

$g \leftarrow \frac{1}{|\mathcal{I}|}\sum_{i=1}^B F_{\xi_i}'(x_0)$

Gen. $N \sim \mathrm{Geo}$ with mean m

for $k = 1, 2, \cdots, N$ **do**

  Gen. $\xi \sim G$

  $\nu \leftarrow F_\xi'(x) - F_\xi'(x_0) + g$

  $x \leftarrow x - \eta\nu$

end for

SCSG: A Brief Summary

In non-convex optimization problems,

- SCSG strictly outperforms SGD in both finite-sum and stochastic optimization, for all accuracy levels;
- SCSG is never worse than SVRG in finite-sum optimization, for all accuracy levels.

SCSG: A Brief Summary

In non-convex optimization problems,

- SCSG strictly outperforms SGD in both finite-sum and stochastic optimization, for all accuracy levels;
- SCSG is never worse than SVRG in finite-sum optimization, for all accuracy levels.

In convex optimization problems,

- SCSG is never worse than SGD and SVRG for all accuracy levels and for both finite-sum and stochastic optimization;
- SCSG does not need the knowledge of μ to achieve the same complexity for strongly convex objectives as SVRG.

Two Techniques

SCSGEpoch

Inputs: x_0, η, B, m

Randomly pick $\mathcal{I}$ with size B — Batching-VR

$g \leftarrow \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} f_i'(x_0)$

Gen. $N \sim \mathrm{Geo}$ with mean m — Geometrization

for $k = 1, 2, \cdots, N$ **do**

  Randomly pick $i \in [n]$

  $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$

  $x \leftarrow x - \eta\nu$

end for

Two Techniques

Batching-VR

- First considered by [HAV+15]. However the analysis requires $B = O(n)$ and unrealistic assumptions (e.g. bounded domain).
- [HAV+15] only holds for strongly-convex objectives and requires the knowledge of μ;
- Also considered by [FGKS15]. However the analysis relies on stringent assumptions and the algorithm has extremely unrealistic settings.

Geometrization

- Implicitly considered by [HLLJM15] in a special setting. However, the analysis still relies on the strong convexity and does not show the gain.

Batching-VR + Geometrization work!

Table of Contents

Smooth Non-convex Optimization

Finite-Sum Optimization

$$\min_x f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

$$\text{Goal}: \mathbb{E}\|\nabla f(x)\|^2 \leq \epsilon$$

Assumptions:

A1 $-LI \preceq \nabla^2 f_i(x) \preceq LI$;

A2 $\sup \|\nabla f_i(x)\|^2 = O(1)$.

Complexity Results:

- SGD: $O\left(\frac{1}{\epsilon^2}\right)$;
- SVRG: $O\left(n + \frac{n^{2/3}}{\epsilon}\right)$;
- SCSG: $\tilde{O}\left(\frac{1}{\epsilon^{5/3}} \wedge \frac{n^{2/3}}{\epsilon}\right)$.

Stochastic Optimization

$$\min_x f(x) \triangleq \mathbb{E}_{\xi\sim G}\, F(x;\xi).$$

$$\text{Goal}: \mathbb{E}\|\nabla f(x)\|^2 \leq \epsilon$$

Assumptions:

A1 $-LI \preceq \nabla^2 F(x;\xi) \preceq LI$;

A2 $\sup \|\nabla F(x;\xi)\|^2 = O(1)$.

Complexity Results:

- SGD: $O\left(\frac{1}{\epsilon^2}\right)$;
- SVRG: not available;
- SCSG: $\tilde{O}\left(\frac{1}{\epsilon^{5/3}}\right)$.

Comparison in Finite-Sum Optimization

	General	$\epsilon \sim n^{-1/2}$	$\epsilon \sim n^{-1}$
Gradient Methods			
GD	$O\left(\frac{n}{\epsilon}\right)$	$O\left(n^{3/2}\right)$	$O\left(n^{2}\right)$
Best available	$\tilde{O}\left(\frac{n}{\epsilon^{5/6}}\right)$	$\tilde{O}\left(n^{17/12}\right)$	$\tilde{O}\left(n^{11/6}\right)$
Stochastic Gradient Methods			
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(n\right)$	$O\left(n^{2}\right)$
Best available	$O\left(n + \frac{n^{2/3}}{\epsilon}\right)$	$O\left(n^{7/6}\right)$	$O\left(n^{5/3}\right)$
SCSG	$\tilde{O}\left(\frac{1}{\epsilon^{5/3}} \wedge \frac{n^{2/3}}{\epsilon}\right)$	$\tilde{O}\left(n^{5/6}\right)$	$\tilde{O}\left(n^{5/3}\right)$

Parameter Settings in SCSG

SCSG (Outer Loop)

Inputs:
$\tilde{x}_0, \{\eta_j\}, \{B_j\}, \{m_j\}, T$

for $j = 1, 2, \cdots, T$ **do**
    $\tilde{x}_j \leftarrow$ **SCSGEpoch**$(\tilde{x}_{j-1}, \eta_j, B_j, m_j)$
end for

Output: $\tilde{x}_T$

SCSGEpoch (Inner Loop)

Inputs: x_0, η, B, m

Randomly pick $\mathcal{I}$ with size B
$g \leftarrow \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} f_i'(x_0)$
Gen. $N \sim \mathrm{Geo}$ with mean m
for $k = 1, 2, \cdots, N$ **do**
    Randomly pick $i \in [n]$
    $\nu \leftarrow f_i'(x) - f_i'(x_0) + g$
    $x \leftarrow x - \eta\nu$
end for

Ouput: x

Parameter Settings in SCSG

SCSG (Outer Loop)

Inputs:
$\tilde{x}_0, \{\eta_j\}, \{B_j\}, \{m_j\}, T$

1: **for** $j = 1, 2, \cdots, T$ **do**
2: $\quad\tilde{x}_j \leftarrow$ **SCSGEpoch**$(\tilde{x}_{j-1}, \eta_j, B_j, m_j)$
3: **end for**

Output: $\tilde{x}_T$

Parameters:

	option 1	option 2
B_j	$O\left(\frac{1}{\epsilon} \wedge n\right)$	$j^{3/2} \wedge n$
m_j	B_j	B_j
η_j	$\frac{1}{2LB_j^{2/3}}$	$\frac{1}{2LB_j^{2/3}}$

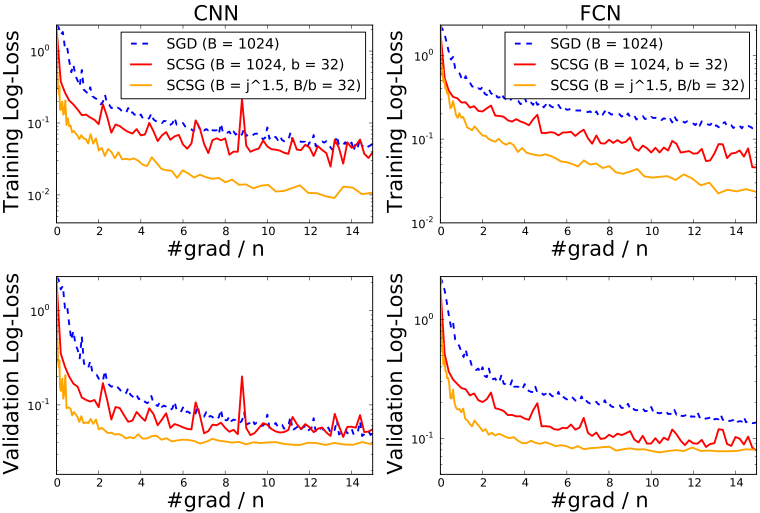
SCSGEpoch (Inner Loop)

Inputs: x_0, η, B, m

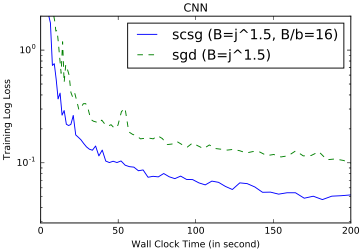
1: Randomly pick $\mathcal{I}$ with size B
2: $g \leftarrow \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} f_i'(x_0)$
3: Gen. $N \sim \mathrm{Geo}$ with mean m
4: **for** $k = 1, 2, \cdots, N$ **do**
5: $\quad$Randomly pick $i \in [n]$
6: $\quad\nu \leftarrow f_i'(x) - f_i'(x_0) + g$
7: $\quad x \leftarrow x - \eta\nu$
8: **end for**

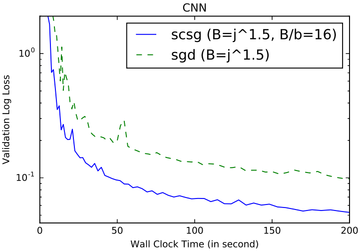
Ouput: x

SCSG for Training Neural Networks



SCSG for Training Neural Networks





Discussion

- Existing acceleration techniques include: **Variance Reduction**, **Momentum**, **Adaptive Gradient**:
  - **Momentum**: Momentum SGD;
  - **Adaptive Gradient**: AdaGrad;
  - **Momentum** + **Adaptive Gradient**: Adam;
  - **Variance Reduction**: SVRG/SAGA, but not in practice!
- The mechanisms of three techniques are different and might be "orthogonal"! Potential gain by combining all:

Variance Reduction + **Momentum** + **Adaptive Gradient**

Table of Contents

Smooth Convex Optimization

Finite-Sum Optimization

$$\min_x f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

$$\text{Goal} : \mathbb{E}(f(x) - f(x^*)) \le \epsilon$$

Assumption:

$$\mu I \preceq \nabla^2 f_i(x) \preceq LI,\ \ \mu \ge 0$$

Stochastic Optimization

$$\min_x f(x) \triangleq \mathbb{E}_{\xi\sim G}\ F(x;\xi).$$

$$\text{Goal} : \mathbb{E}(f(x) - f(x^*)) \le \epsilon$$

Assumption:

$$\mu I \preceq \nabla^2 F(x;\xi) \preceq LI,\ \ \mu \ge 0;$$

Convex Optimization Theory is Weird

SGD (in convex stochastic optimization):

- Always different settings (of stepsizes) for strongly and non-strongly convex objectives:
- $\eta_t = O\left(\frac{1}{\sqrt{t}}\right)$ for non-strongly convex case; complexity $O\left(\frac{1}{\epsilon^2}\right)$;
- $\eta_t = O\left(\frac{1}{\mu t}\right)$ for strongly convex case; complexity $O\left(\frac{1}{\mu\epsilon}\right)$;

Convex Optimization Theory is Weird

SGD (in convex stochastic optimization):

- Always different settings (of stepsizes) for strongly and non-strongly convex objectives:
- $\eta_t = O\left(\frac{1}{\sqrt{t}}\right)$ for non-strongly convex case; complexity $O\left(\frac{1}{\epsilon^2}\right)$;
- $\eta_t = O\left(\frac{1}{\mu t}\right)$ for strongly convex case; complexity $O\left(\frac{1}{\mu\epsilon}\right)$;
- Use $\eta_t = O\left(\frac{1}{\sqrt{t}}\right)$ for strongly convex case does not yield the better complexity $O\left(\frac{1}{\mu\epsilon}\right)$ (in general);
- Use $\eta_t = O\left(\frac{1}{\mu t}\right)$ for non-strongly convex case (with a wrong guess of μ) could yield a complexity as bad as $O\left(e^{\frac{1}{\epsilon}}\right)$;
- Users must know the property of the objective and must know μ to take advantage of strong convexity!

Convex Optimization Theory is Weird

SVRG (in convex finite-sum optimization):

- Also different settings for strongly and non-strongly convex objectives:
- Original SVRG only for strongly convex objectives with $m_j \equiv O(\kappa), \eta_j \equiv O\left(\frac{1}{L}\right)$; complexity $\tilde{O}(n+\kappa)$;
- In order to extend SVRG to non-strongly convex objectives,
  - [AZY15]: $m_j = 2^j$; complexity $\tilde{O}\left(n + \frac{1}{\epsilon}\right)$;
  - [RHS$^+$16]: $m_j = O(n), \eta_j = O\left(\frac{1}{L\sqrt{n}}\right)$; complexity $O\left(n + \frac{\sqrt{n}}{\epsilon}\right)$.
- Again, separate analyses for different settings.

Adaptivity Matters!

An popular hand-waving argument of strong convexity:

μ is always known in practice because an L_2 regularizer,
in the form of $\frac{\lambda}{2}\|x\|^2$, is always added so one can set $\mu = \lambda$.

Adaptivity Matters!

An popular hand-waving argument of strong convexity:

μ is always known in practice because an L_2 regularizer,
in the form of $\frac{\lambda}{2}\|x\|^2$, is always added so one can set $\mu = \lambda$.

No!

- λ is usually small, e.g. $\lambda \sim 10^{-6}$, in which case the condition number κ is too large to justify the gain of strong convexity: compare $O\left(\frac{1}{\epsilon^2}\right)$ with $O\left(\frac{10^6}{\epsilon}\right)$;
- λ is too conservative: the global strong convexity parameter might be way larger than λ and the local strong convexity parameter, around the optimum, could be even larger.

Adaptivity Matters!

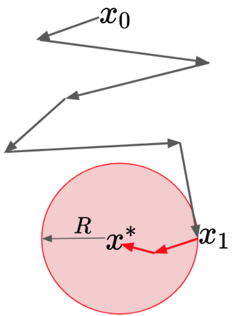
The degree of strong convexity forms a continuum.
An algorithm should depend on μ continuously without knowing it!

Adaptivity Matters!

The degree of strong convexity forms a continuum.
An algorithm should depend on μ continuously without knowing it!

Advantages of adaptive algorithms:

- Unified algorithm for both cases;
- Global adaptivity $\Longrightarrow$ local adaptivity.



Adaptivity Matters!

Knowing μ makes a difference in terms of oracle lower bounds:

- [AS16] proves the lower bound $\Omega\left((n+\sqrt{n\kappa})\log\left(\frac{1}{\epsilon}\right)\right)$, in terms of ϵ, for CLI algorithms in finite-sum optimization, if μ is known;
- [Arj17b] shows that the above bound is not achievable without knowing μ, in which case the lower bound is $O\left((n+\kappa)\log\left(\frac{1}{\epsilon}\right)\right)$, in terms of ϵ.

Existing Works on Adaptivity

- Deterministic gradient method [N+07]:
  - doubling/halving technique;
  - need to check conditions on the norm of gradients at each step, thus not applicable in stochastic algorithms.
- Adaptive SVRG [XLY17]:
  - doubling/halving technique;
  - achieves the complexity $\tilde{O}((n+\kappa)\log\left(\frac{1}{\epsilon}\right))$;
  - need a lower bound for μ and hence no guarantee for non-strongly convex objective;
  - parameters depend on ϵ.
- Hand-waving algorithms:
  - Ad-hoc approaches to adaptively estimate μ; extra overhead may dominate;
  - Restarting schemes; need the knowledge of μ to obtain theoretical guarantee.

Achieving Adaptivity Via SCSG

Randomized SVRG [LJ17]:

- A special case of SCSG;
- $B_j = m_j \equiv n, \eta_j = \frac{1}{3L}$ with complexity
$$\tilde{O}\left(\frac{n}{\epsilon} \wedge (n + \kappa)\right);$$
- need to record both the average (for the former) and the last iterate (for the latter);
- compared to SVRG: $B_j \equiv n, m_j \equiv m = O(\kappa), \eta_j \equiv \eta < \frac{1}{2L}$ with complexity
$$\tilde{O}\left(n + \kappa\right).$$

Achieving Adaptivity Via SCSG

SCSG+ (to appear soon):

- $B_j = B_0 \cdot 1.05^{2j} \wedge n, m_j = m_0 \cdot 1.05^j, \eta_j \equiv \eta = \frac{1}{4L}$ with complexity

$$\tilde{O}\left(\frac{1}{\epsilon^2} \wedge \left(n + \frac{1}{\epsilon}\right) \wedge \left(n + \kappa \left(\frac{1}{\epsilon\kappa}\right)^{0.05}\right)\right);$$

- The extra term $\left(\frac{1}{\epsilon\kappa}\right)^{0.05}$ is almost negligible. In addition, the exponent 0.05 can be made arbitrarily small by shrinking η; roughly $\log(1.05)/\log(1/\eta L)$;
- SGD with $\eta_t = \frac{1}{\sqrt{t}}$ achieves $\tilde{O}\left(\frac{1}{\epsilon^2}\right)$;
  SVRG^{++} achieves $\tilde{O}\left(n + \frac{1}{\epsilon}\right)$;
  SVRG achieves $\tilde{O}(n + \kappa)$ with known μ;
  SCSG almost achieves the best of them, without knowing μ!
- Adaptivity to both strong convexity and required accuracy.

Other Remarks on SCSG

- SGD relies on bounded gradient condition

$$\mathcal{H}^* \triangleq \sup_{i,x} \|\nabla f_i(x)\|^2 = O(1) \text{ (for finite-sum optimization)}$$

or $\mathcal{H}^* \triangleq \sup_{\xi,x} \|\nabla F(x;\xi)\|^2 = O(1)$ (for stochastic optimization).

- Unfortunately this even does not hold for least squares unless the domain is bounded and projection step is performed every step. But nobody uses that in practice!
- SCSG relies on a much weaker condition ($x^* = \arg\min f(x)$)

$$\mathcal{H} \triangleq \sup_{i} \|\nabla f_i(x^*)\|^2 = O(1) \text{ (for finite-sum optimization)}$$

or $\mathcal{H} \triangleq \sup_{\xi} \|\nabla F(x^*;\xi)\|^2 = O(1)$ (for stochastic optimization).

- Extensive discussion of $\mathcal{H}$ in [LJ16].

Other Remarks on SCSG

Refined rate of SCSG+

$$\tilde{O}\Bigg(\left(\frac{D}{\epsilon}\right)^2 \wedge \Bigg(\left(\frac{D_H}{\epsilon}\right)^2 + \kappa^2 \left(\frac{D_x}{\epsilon\kappa}\right)^{0.09} \Bigg) \wedge \left(n + \frac{D}{\epsilon}\right) \wedge \Bigg(n + \kappa \left(\frac{D_H}{\epsilon\kappa}\right)^{0.05}\Bigg) \Bigg);$$

where $\quad D_x = L \cdot \mathbb{E}\|\tilde{x}_0 - x^*\|^2, \quad D_H = \frac{\mathcal{H}}{L}, \quad D = \max\{D_x, D_H\}.$

- D_x measures the quality of initialization; D_H measures heterogeneity of the components;
- D_x is algorithm/user driven while D_H is intrinsic;
- SCSG+ shows adaptivity for large ϵ with more tolerance to bad initialization when $\kappa \ll \frac{1}{\epsilon}$ (same condition for SGD to take advantage of strong convexity).

Optimality?

- [AB14] proves the lower bound $\Omega\left(\frac{1}{\epsilon^2}\right)$;
- [Arj17a] proves the lower bound $\tilde{\Omega}\left(n+\kappa\right)$ for strongly-convex objectives;
- [Arj17a] proves the lower bound $\tilde{\Omega}\left(n+\sqrt{\frac{n}{\epsilon}}\right)$, achieved by Accelerate SDCA on Generalized Linear Models;
- [WS16] proves the lower bound $\Omega\left(\frac{1}{\mu\epsilon}\right)$ for strongly-convex objectives when μ is known.
- My conjecture: $\Omega\left(\frac{1}{\mu\epsilon}\right)$ is not achievable when μ is unknown.

Optimality?

The above results give a (possibly loose) lower bound as

$$\tilde{\Omega}\left(\frac{1}{\epsilon^2} \wedge \frac{1}{\mu\epsilon} \wedge \left(n + \sqrt{\frac{n}{\epsilon}}\right) \wedge (n+\kappa)\right)$$

Recall the bound of SCSG:

$$\tilde{O}\left(\frac{1}{\epsilon^2} \wedge \left(n + \frac{1}{\epsilon}\right) \wedge (n+\kappa)\right)$$

Summary

In non-convex optimization problems,

- SCSG has complexity $\tilde{O}\left(\frac{1}{\epsilon^{5/3}} \wedge \frac{n^{2/3}}{\epsilon}\right)$ to reach an ϵ-approximated first-order stationary point;
- SCSG strictly outperforms SGD, with complexity $O\left(\frac{1}{\epsilon^2}\right)$, in both finite-sum and stochastic optimization, for all accuracy;
- SCSG is never worse than SVRG, with complexity $O\left(n + \frac{n^{2/3}}{\epsilon}\right)$, in stochastic optimization, for all accuracy.

Summary

In convex optimization problems,

- SCSG has complexity $\tilde{O}\left(\frac{1}{\epsilon^2} \wedge \left(n + \frac{1}{\epsilon}\right) \wedge (n + \kappa)\right)$ to reach an ϵ-approximated solution;
- SCSG is never worse than SGD, with complexity and SVRG (SVRG^{++}, ...), for all accuracy and for both finite-sum and stochastic optimization;
- SCSG does not need the knowledge of μ to achieve the same complexity for strongly convex objectives as SVRG.

References

[AB14] Alekh Agarwal and Leon Bottou. A lower bound for the optimization of finite sums. ArXiv e-prints abs/1410.0723, 2014.

[Arj17a] Yossi Arjevani. Limitations on variance-reduction and acceleration schemes for finite sum optimization. arXiv preprint arXiv:1706.01686, 2017.

[Arj17b] Yossi Arjevani. Limitations on variance-reduction and acceleration schemes for finite sums optimization. In Advances in Neural Information Processing Systems, pages 3543–3552, 2017.

[AS16] Yossi Arjevani and Ohad Shamir. Dimension-free iteration complexity of finite sum optimization problems. In Advances in Neural Information Processing Systems, pages 3540–3548, 2016.

[AZY15] Zeyuan Allen-Zhu and Yang Yuan. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. ArXiv e-prints, abs/1506.01972, 2015.

[FGKS15] Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In Conference on learning theory, pages 728–763, 2015.

[HAV+15] Reza Harikandeh, Mohamed Osama Ahmed, Alim Virani, Mark Schmidt, Jakub Konečnỳ, and Scott Sallinen. Stop wasting my gradients: Practical SVRG. In Advances in Neural Information Processing Systems, pages 2242–2250, 2015.

[HLLJM15] Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance reduced stochastic gradient descent with neighbors. In Advances in Neural Information Processing Systems, pages 2305–2313, 2015.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Advances in Neural Information Processing Systems, pages 315–323, 2013.

[LJ16] Lihua Lei and Michael I Jordan. Less than a single pass: Stochastically controlled stochastic gradient method. arXiv preprint arXiv:1609.03261, 2016.

[LJ17] Lihua Lei and Michael I Jordan. Less than a single pass: Stochastically controlled stochastic gradient method. arXiv preprint arXiv:1609.03261, 2017.

[N+07] Yurii Nesterov et al. Gradient methods for minimizing composite objective function, 2007.

[RHS+16] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. arXiv preprint arXiv:1603.06160, 2016.

[WS16] Blake Woodworth and Nathan Srebro. Tight complexity bounds for optimizing composite objectives. ArXiv e-prints abs/1605.08003, 2016.

[XLY17] Yi Xu, Qihang Lin, and Tianbao Yang. Adaptive svrg methods under error bound conditions with unknown growth parameter. In Advances in Neural Information Processing Systems, pages 3279–3289, 2017.

THANKS!

A bit about myself

- With Peter Bickel and Noureddine El Karoui
  - exact and asymptotic inference on high-dimensional non-sparse linear models;
- With Michael Jordan
  - convex and non-convex optimization;
  - higher-order accuracy of bootstrap and its variant;
- With William Fithian
  - interactive multiple testing with side information;
  - knockoffs-based inference;
- With Alex D'amour, Peng Ding, Avi Feller and Jasjeet Sekhon
  - debiasing regression-adjustment in randomized experiments;
  - robust randomized designs;
  - justifying overlap condition in observational studies.