

# Variance reduction for stochastic gradient methods

Axel Böhm

October 3, 2021



# The finite sum problem

A common Task in (supervised) machine learning:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^n \underbrace{f_i(x)}_{\text{loss for } i\text{-th sample}} + \underbrace{\psi(x)}_{\text{regularizer}}$$

where the  $i$ -th sample is  $(a_i, y_i)$ .

**Examples:**

- ◇ linear regression:  $f_i(x) = (a_i^T x - y_i)^2$ , and  $\psi = 0$
- ◇ logistic regression:  $f_i(x) = \log(1 + e^{-y_i a_i^T x})$ , and  $\psi = 0$   
“sigmoid function” and logistic loss.
- ◇ Lasso:  $f_i$  as for linear regression but  $\psi(x) = \|x\|_1$
- ◇ SVM:  $f_i(x) = \max\{0, 1 - y_i a_i^T x\}$  and  $\psi(x) = \|x\|^2$

# Gradient descent

---

## Algorithm (batch) GD

---

- 1: **for**  $k = 1, 2, \dots$  **do**
  - 2:      $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- 

- ◇ gradient can be computed via

$$\nabla f(x) = \nabla \left( \sum_{i=1}^n f_i(x) \right) = \sum_{i=1}^n \nabla f_i(x_k)$$

- ◇ good convergence properties
- ◇ can be **expensive** if  $n$  is large!

# Stochastic gradient descent

---

## Algorithm SGD

---

- 1: **for**  $k = 1, 2, \dots$  **do**
  - 2:     pick  $i_k$  uniform at random in  $[n]$
  - 3:      $x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k)$
- 

We already noticed that:

- ◇ unbiased:  $\mathbb{E}[\nabla f_{i_k}(x)] = \sum_{i=1}^n \mathbb{P}[i = i_k] \nabla f_i(x) = \sum_{i=1}^n \frac{1}{n} \nabla f_i(x)$
- ◇ large stepsizes fail to suppress variability in the stoch. gradients  $\rightarrow$  leads to oscillations
- ◇ decreasing stepsizes mitigate this problem but **slows down** convergence (too *conservative*)

# Recall SGD (template)

$$x_{k+1} = x_k - \alpha_k g_k$$

- ◇  $g_k$  is an unbiased estimator of the true gradient  $\nabla F(x_k)$
- ◇ convergence depends on **variance**  $\mathbb{E}[\|g_k - \nabla F(x_k)\|] \leq \sigma_g$   
(not strictly necessary)
- ◇ vanilla SGD uses  $g_k = \nabla f_{i_k}(x_k)$   
**issue:**  $\sigma_g$  is non-negligible even close to the solution
- ◇ **Q:** can we choose  $g_k$  in a different way to reduce variability?

# Minibatching

---

## Algorithm minibatch SGD

---

- 1: **for**  $k = 1, 2, \dots$  **do**
  - 2:     pick  $I_k$  random subset of  $[n]$  with  $|I_k| = b$
  - 3:      $x_{k+1} = x_k - \alpha_k \sum_{i \in I_k} \nabla f_i(x_k)$
- 

- ◇ typically we make a (uniform) *random* choice  
 $i_k \in [n] = \{1, \dots, n\}$   
(or random reshuffling)
- ◇ by increasing the size to a **random subset**  $I_k \subset [n]$  of size  $b \ll n$  we can
  - ▶ decrease variance
  - ▶ increase cost only moderately,
  - ▶ but cannot improve theoretical guarantees

# A simple idea

Consider

- ◇ estimator  $X$  for parameter  $\mu$  ( $\mathbb{E}[X] = \mu$  and  $\mathbb{V}[X] = \sigma^2$ )
- ◇ want to keep unbiased but reduce variance
- ◇ find  $Y$  such that  $\mathbb{E}[Y] = 0$  but  $\text{Cov}(X, Y)$  is large and  $\tilde{X} = X - Y$
- ◇ remains unbiased

$$\mathbb{V}[\tilde{X}] = \mathbb{V}[X] + \mathbb{V}[Y] - 2 \text{Cov}[X, Y]$$

- ◇ can be much smaller than  $\mathbb{V}[X]$  if  $X, Y$  are highly correlated



# Stochastic average gradient (SAG), 2013

- ◇ maintain table containing gradients  $g_i$  of  $f_i$
- ◇ at step  $k = 1, 2, \dots$  pick random  $i_k \in [n]$  and

$$g_{i_k}^k := \nabla f_{i_k}(x^k)$$

set  $g_i^k = g_i^{k-1}$  for all  $i \neq i_k$  (remain the same)

- ◇ Update

$$x^{k+1} = x^k - \alpha_k \frac{1}{n} \sum_{i=1}^n g_i^k$$

- ◇ assuming gradients do not change too much along trajectory
- ◇ gradient estimator **no longer unbiased**
- ◇ Isn't it expensive to average these gradients?

$$x^{k+1} = x^k - u\alpha_k \left( \frac{g_{i_k}^k}{n} - \frac{g_{i_k}^{k-1}}{n} + \underbrace{\frac{1}{n} \sum_{i=1}^n g_i^k}_{\text{old table average}} \right)$$

# SAG variance reduction

Gradient estimator in SAG:

$$x^{k+1} = x^k - \alpha_k \frac{1}{n} \left( \underbrace{g_{i_k}^k}_X - \underbrace{g_{i_k}^{k-1} - \sum_{i=1}^n g_i^k}_Y \right)$$

- ◇ While  $\mathbb{E}[X] = \nabla f(x^k)$ , but  $\mathbb{E}[Y] \neq 0 \rightarrow$  biased estimator
- ◇  $X$  and  $Y$  are correlated as  $X - Y \rightarrow 0$  as
- ◇  $x^k$  and  $x_{k-1}$  both converge to  $x^*$  and
- ◇ the last term converges to  $\nabla f(x^*) = 0$

# Convergence

As always, initialization plays a role:  $D^2 := \|x^0 - x^*\|^2$ .

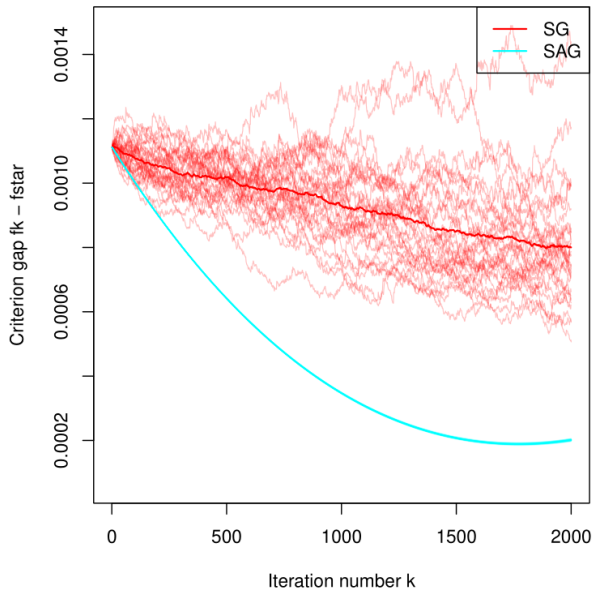
$$\text{SAG: } \frac{n}{k}(f(x^0) - f^*) + \frac{L}{k}D^2$$

$$\text{GD: } \frac{L}{k}D^2$$

$$\text{SAG: } \frac{L}{\sqrt{k}}D^2$$

- ◇ Achieves linear convergence in the *strongly* convex setting.
- ◇ proofs are difficult (and computer-aided)

Same gradient oracle cost as SGD, but same converge rate as GD.



## SAGA, 2014

Very similar:

- ◇ maintain table containing gradients  $g_i$  of  $f_i$
- ◇ at step  $k = 1, 2, \dots$  pick random  $i_k \in [n]$  and

$$g_{i_k}^k := \nabla f_{i_k}(x^k)$$

set  $g_i^k = g_i^{k-1}$  for all  $i \neq i_k$  (remain the same)

- ◇ Update

$$x^{k+1} = x^k - u\alpha_k \left( g_{i_k}^k - g_{i_k}^{k-1} + \frac{1}{n} \sum_{i=1}^n g_i^k \right)$$

- ◇ estimator now **unbiased**!

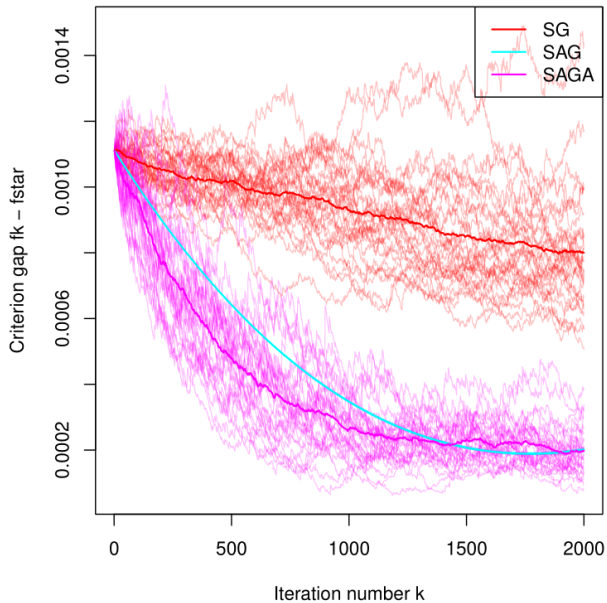
# For Comparison

SAGA gradient estimate:

$$g_{i_k}^k - g_{i_k}^{k-1} + \frac{1}{n} \sum_{i=1}^n g_i^k$$

SAGA gradient estimate:

$$\frac{1}{n} g_{i_k}^k - \frac{1}{n} g_{i_k}^{k-1} + \frac{1}{n} \sum_{i=1}^n g_i^k$$



# Stochastic Variance Reduced Gradient (SVRG), 2013

---

## Algorithm SVRG

---

```
1: for  $k = 1, 2, \dots$  do
2:   Set  $x^1 = \tilde{x} = \tilde{x}^k$ 
3:   Compute  $\tilde{\mu} := \nabla f(\tilde{x})$                                      //update snapshot
4:   for  $l = 1, 2, \dots, m$  do                                     //m iterations per epoch
5:     pick  $i_l$  uniform at random in  $[n]$ 
6:     Set  $x^{l+1} = x^l - \alpha(\nabla f_{i_l}(x^l) - \nabla f_{i_l}(\tilde{x}) + \tilde{\mu})$ 
7:    $\tilde{x}^{k+1} = x^{m+1}$ 
```

---

- ◇ Does **not** need to **store** full table of gradients.
- ◇ requires *batch* gradient computation every *epoch*
- ◇ per iteration cost is comparable to that of SGD if  $m \geq n$
- ◇ convergence rates similar to SAGA, but simpler analysis.



## SVRG

key idea: by storing old point we can

$$\underbrace{\nabla f_{i_k}(x^k) - \nabla f_{i_k}(x^{\text{old}})}_{\rightarrow 0 \text{ if } x \approx x^{\text{old}}} + \underbrace{\nabla f(x^{\text{old}})}_{\rightarrow 0 \text{ if } x^{\text{old}}}$$

- ◇ is an unbiased estimate of  $\nabla f(x^k)$
- ◇ converges to 0 (meaning reduced variability) if  $x^k \approx x^{\text{old}} \approx x^*$