Optimization for data science

Stochastic Gradient Descent

R. Flamary

Master Data Science, Institut Polytechnique de Paris

October 29, 2024

Full course overview

1. **Introduction to optimization for data science**
   1.1 ML optimization problems and linear algebra recap
   1.2 Optimization problems and their properties (Convexity, smoothness)
2. **Smooth optimization : Gradient descent**
   2.1 First order algorithms, convergence for smooth and strongly convex functions
3. **Smooth Optimization : Quadratic problems**
   3.1 Solvers for quadratic problems, conjugate gradient
   3.2 Linesearch methods
4. **Non-smooth Optimization : Proximal methods**
   4.1 Proximal operator and proximal algorithms
   4.2 Lab 1: Lasso and group Lasso
5. **Stochastic Gradient Descent**
   5.1 SGD and variance reduction techniques
   5.2 Lab 2: SGD for Logistic regression
6. **Standard formulation of constrained optimization problems**
   6.1 LP, QP and Mixed Integer Programming
7. **Coordinate descent**
   7.1 Algorithms and Labs
8. **Newton and quasi-newton methods**
   8.1 Second order methods and Labs
9. **Beyond convex optimization**
   9.1 Nonconvex reg., Frank-Wolfe, DC programming, autodiff

Current course overview

Machine learning a.k.a minimizing a finite sum

Optimization problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \qquad F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}) \tag{1}$$

- Standard ML problem (supervised or unsupervised learning).
- d is the number of parameter in the model, n the number of training samples.
- Can handle both ERM and regularized learning:
  - Empirical Risk Minimization : $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2$
  - Regularization : $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$
- Gradient of F is: $\nabla_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\mathbf{w}} f_i(\mathbf{w})$

Large sale optimization

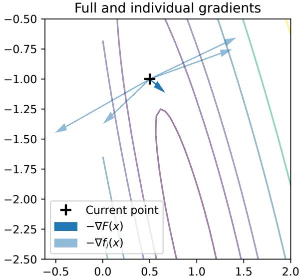
- Both n and d can be very large.
- Computation of F and ∇F is $O(nd)$.
- Dataset may not fit in memory.

$\Rightarrow$ Approximate the gradient: Stochastic Gradient Descent.

Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) algorithm

1: Initialize $\mathbf{x}^{(0)}$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad i^{(k)} \leftarrow$ randomly pick an index $i \in \{1, \ldots, n\}$
4: $\quad \mathbf{d}^{(k)} \leftarrow -\nabla_{\mathbf{x}} f_{i^{(k)}}(\mathbf{x}^{(k)})$
5: $\quad \mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \rho^{(k)} \mathbf{d}^{(k)}$
6: **end for**



- $\mathbf{d}^{(k)} \in \mathbb{R}^n$ is an approximation of the full gradient on one sample.
- Iteration complexity is $O(d)$ VS $O(nd)$ for GD.
- With very small step size, SGD (over an epoch) is very close to GD.
- Step size strategies:
  - Fixed step size : $\rho^{(k)} = \rho$
  - Decreasing step size : $\rho^{(k)} = \frac{1}{\sqrt{k}}$

Convergence of SGD with fixed step size (1)

Assumptions

- F is μ-strongly convex.
- $F = \frac{1}{n}\sum_i f_i$ has Expected Bounded Stochastic Gradients (EBSG):

$$\mathbb{E}_{i\sim\frac{1}{n}}[\|\nabla f_i(\mathbf{x}^{(k)})\|^2] \leq B^2, \quad \forall k \tag{2}$$

Convergence of fixed step SGD on strongly convex functions

If F is μ-strongly convex and $F = \frac{1}{n}\sum_i f_i$ has Expected Bounded Stochastic Gradients, then for $\rho < \frac{1}{\mu}$ we have for fixed step SGD:

$$\mathbb{E}[\|\mathbf{x}^{(k)} - \mathbf{x}^\star\|^2] \leq (1-\rho\mu)^k\|\mathbf{x}^{(0)} - \mathbf{x}^\star\|^2 + \frac{\rho}{\mu}B^2 \tag{3}$$

- Fast (exponential) convergence of the first term.
- Bias term $\frac{\rho}{\mu}B^2$ proportional to the step size!

Proof of convergence of fixed step SGD (1)

$$
\begin{aligned}
\|\mathbf{x}^{(k+1)} - \mathbf{x}^\star\|^2 &= \|\mathbf{x}^{(k)} - \rho\nabla f_{i^{(k)}}(\mathbf{x}^{(k)}) - \mathbf{x}^\star\|^2 \\
&\leq \|\mathbf{x}^{(k)} - \mathbf{x}^\star\|^2 - 2\rho\nabla f_{i^{(k)}}^\top(\mathbf{x}^{(k)} - \mathbf{x}^\star) + \rho^2\|\nabla f_{i^{(k)}}(\mathbf{x}^{(k)})\|^2
\end{aligned}
$$

By taking the expectation w.r.t. $i^{(k)}$ we get:

$$
\begin{aligned}
\mathbb{E}_{i^{(k)}\sim\frac{1}{n}}[\|\mathbf{x}^{(k+1)} - \mathbf{x}^\star\|^2] &\underset{1}{\leq} \|\mathbf{x}^{(k)} - \mathbf{x}^\star\|^2 - 2\rho\nabla F(\mathbf{x}^{(k)})^\top(\mathbf{x}^{(k)} - \mathbf{x}^\star) + \rho^2 B^2 \\
&\underset{2}{\leq} (1-\rho\mu)\|\mathbf{x}^{(k)} - \mathbf{x}^\star\|^2 + \rho^2 B^2
\end{aligned}
$$

Now taking the total expectation w.r.t. all steps

$$
\begin{aligned}
\mathbb{E}[\|\mathbf{x}^{(k+1)} - \mathbf{x}^\star\|^2] &\leq (1-\rho\mu)\mathbb{E}[\|\mathbf{x}^{(k)} - \mathbf{x}^\star\|^2] + \rho^2 B^2 \\
&\leq (1-\rho\mu)^k\|\mathbf{x}^{(0)} - \mathbf{x}^\star\|^2 + \rho^2 B^2\sum_{i=0}^{k}(1-\rho\mu)^i \\
&\leq (1-\rho\mu)^k\|\mathbf{x}^{(0)} - \mathbf{x}^\star\|^2 + \rho^2 B^2\frac{1-(1-\rho\mu)^{i+1}}{1-(1-\rho\mu)} \\
&\leq (1-\rho\mu)^k\|\mathbf{x}^{(0)} - \mathbf{x}^\star\|^2 + \frac{\rho}{\nu}B^2
\end{aligned}
$$

---

[1]Unbiased gradient $\nabla F(\mathbf{x}^{(k)}) = \mathbb{E}_{i\sim\frac{1}{n}}\nabla f_i(\mathbf{x}^{(k)})$ and $\mathbb{E}_{i\sim\frac{1}{n}}[\|\nabla f_i(\mathbf{x}^{(k)})\|^2] \leq B^2$

[2]Strong convexity $\nabla F(\mathbf{x}^{(k)})^\top(\mathbf{x}^{(k)} - \mathbf{x}^\star) \geq \mu\|\mathbf{x}^{(k)} - \mathbf{x}^\star\|^2$

Assumptions for convergence of SGD

Expected Bounded Stochastic Gradients (EBSG)

$$\mathbb{E}_{i\sim\frac{1}{n}}[\|\nabla f_i(\mathbf{x}^{(k)})\|^2] \leq B^2, \quad \forall k$$

Exercise 1: Linear regression

1. $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2$.
2. Compute $\nabla f_i(\mathbf{w})$

$$\nabla f_i(\mathbf{w}) =$$

3. Compute $\mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2]$

$$\mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2] =$$

4. What is $\max_{\mathbf{w}} \mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2]$?
5. Is Quadratic loss EBSG?

Assumptions for convergence of SGD

Expected Bounded Stochastic Gradients (EBSG)

$$\mathbb{E}_{i\sim\frac{1}{n}}[\|\nabla f_i(\mathbf{x}^{(k)})\|^2] \leq B^2, \quad \forall k$$

Exercise 1: Linear regression

1. $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2$.
2. Compute $\nabla f_i(\mathbf{w})$
$$\nabla f_i(\mathbf{w}) = -2(y_i - \mathbf{x}_i^T\mathbf{w})\mathbf{x}_i$$
3. Compute $\mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2]$
$$\mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2] =$$
4. What is $\max_{\mathbf{w}} \mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2]$?
5. Is Quadratic loss EBSG?

Assumptions for convergence of SGD

Expected Bounded Stochastic Gradients (EBSG)

$$\mathbb{E}_{i\sim\frac{1}{n}}[\|\nabla f_i(\mathbf{x}^{(k)})\|^2] \leq B^2, \quad \forall k$$

Exercise 1: Linear regression

1. $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2$.
2. Compute $\nabla f_i(\mathbf{w})$
$$\nabla f_i(\mathbf{w}) = -2(y_i - \mathbf{x}_i^T\mathbf{w})\mathbf{x}_i$$
3. Compute $\mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2]$
$$\begin{aligned}\mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2] &= \frac{4}{n}\sum_i \|\mathbf{x}_i(y_i - \mathbf{x}^\top\mathbf{w})\|^2 \\ &= \frac{4}{n}\sum_i \|\mathbf{x}_i\|^2(y_i - \mathbf{x}_i^\top\mathbf{w})^2 \\ &= \frac{4}{n}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2_{\mathsf{diag}(\|\mathbf{x}_i\|)^{-1}}\end{aligned}$$
4. What is $\max_{\mathbf{w}} \mathbb{E}[\|\nabla f_i(\mathbf{w})\|^2]$?
5. Is Quadratic loss EBSG?

Convergence of SGD with fixed step size (2)

Assumptions

- F is μ-strongly convex.
- $F = \frac{1}{n}\sum_i f_i$ and each f_i is L_i-smooth.
- Definition: **Gradient noise**

$$\sigma^2 = \mathbb{E}_{i\sim\frac{1}{n}}[\|\nabla f_i(\mathbf{x}^\star)\|^2] \tag{4}$$

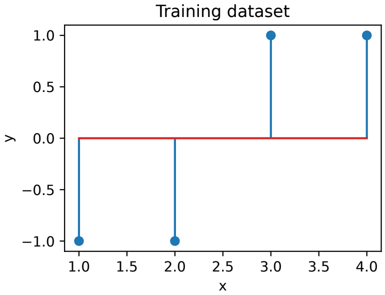
Convergence of fixed step SGD on strongly convex and smooth functions

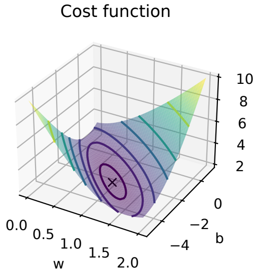
If F is μ-strongly convex and $F = \frac{1}{n}\sum_i f_i$ with $\forall i,\ f_i$ is L_i-smooth and $L_{max} = \max_i L_i$, then for $\rho \leq \frac{1}{2L_{max}}$ we have for fixed step SGD:

$$\mathbb{E}[\|\mathbf{x}^{(k)} - \mathbf{x}^\star\|^2] \leq (1-\rho\mu)^k\|\mathbf{x}^{(0)} - \mathbf{x}^\star\|^2 + \frac{2\rho}{\mu}\sigma^2 \tag{5}$$

- Fast (exponential) convergence of the first term.
- Bias term $\frac{\rho}{\mu}\sigma^2$ proportional to the step size but now only on solution.
- Homework exercise on moodle, proof available in [Gower et al., 2019].

Example optimization problem



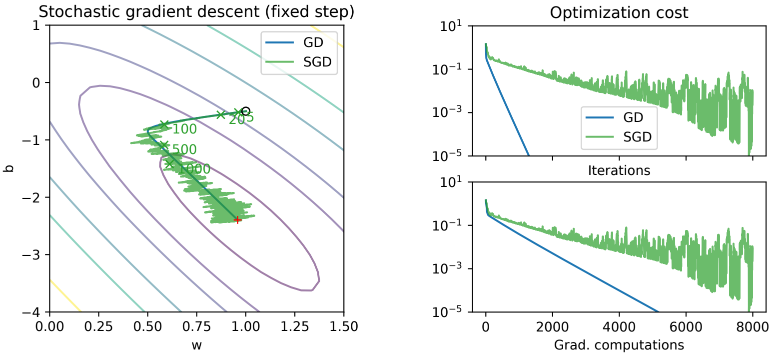


1D Logistic regression

$$\min_{w,b} \quad \sum_{i=1}^{n} \log(1 + \exp(-y_i(wx_i + b))) + \lambda \frac{w^2}{2}$$

- Linear prediction model : $f(x) = wx + b$
- Training data (x_i, y_i) : $(1, -1), (2, -1), (3, 1), (4, 1)$.
- Problem solution for $\lambda = 1$: $\mathbf{x}^* = [w^\star, b^\star] = [0.96, -2.40]$
- Initialization : $\mathbf{x}^{(0)} = [1, -0.5]$.

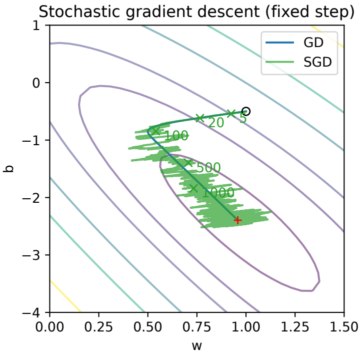
Example of constant step SGD

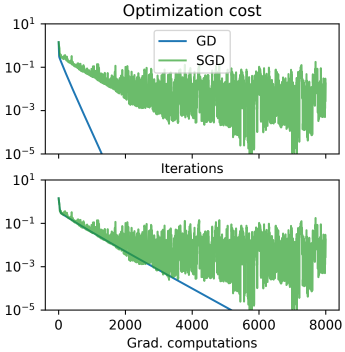


Discussion

- SGD VS GD (as a function of iterations and nb of grad. computation).
- Fixed step size : $\rho^{(k)} = 0.01$ and $\rho^{(k)} = 0.02$
- One GD iter $\equiv 4$ SGD iter (since $n = 4$).
- Complexity $\mathcal{O}(d)$ per iteration but not convergence (bias).

Example of constant step SGD





Discussion

- SGD VS GD (as a function of iterations and nb of grad. computation).
- Fixed step size : $\rho^{(k)} = 0.01$ and $\rho^{(k)} = 0.02$
- One GD iter $\equiv 4$ SGD iter (since $n = 4$).
- Complexity $\mathcal{O}(d)$ per iteration but not convergence (bias).

Exercise 2: Calculating smoothness constants (1)

Ridge regression

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda\|\mathbf{w}\|^2$$

Compute the smoothness constant L_i and L_{max}.

1. $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda\|\mathbf{w}\|^2$.
2. Compute $\nabla f_i(\mathbf{w})$.

$$\nabla f_i(\mathbf{w}) =$$

3. Compute $\nabla^2 f_i(\mathbf{w})$.

$$\nabla^2 f_i(\mathbf{w}) =$$

4. Find L_i.

$$\|\nabla^2 f_i(\mathbf{w})\| =$$

5. Fin $L_{max} =$.

Exercise 2: Calculating smoothness constants (1)

Ridge regression

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda\|\mathbf{w}\|^2$$

Compute the smoothness constant L_i and L_{max}.

1. $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda\|\mathbf{w}\|^2$.
2. Compute $\nabla f_i(\mathbf{w})$.

$$\nabla f_i(\mathbf{w}) = -2(y_i - \mathbf{x}_i^T\mathbf{w})\mathbf{x}_i + 2\lambda\mathbf{w}$$

3. Compute $\nabla^2 f_i(\mathbf{w})$.

$$\nabla^2 f_i(\mathbf{w}) =$$

4. Find L_i.

$$\|\nabla^2 f_i(\mathbf{w})\| =$$

5. Fin $L_{max} =$.

Exercise 2: Calculating smoothness constants (1)

Ridge regression

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda\|\mathbf{w}\|^2$$

Compute the smoothness constant L_i and L_{max}.

1. $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda\|\mathbf{w}\|^2$.
2. Compute $\nabla f_i(\mathbf{w})$.

$$\nabla f_i(\mathbf{w}) = -2(y_i - \mathbf{x}_i^T\mathbf{w})\mathbf{x}_i + 2\lambda\mathbf{w}$$

3. Compute $\nabla^2 f_i(\mathbf{w})$.

$$\nabla^2 f_i(\mathbf{w}) = 2\mathbf{x}_i\mathbf{x}_i^\top + 2\lambda\mathbf{I}$$

4. Find L_i.

$$\|\nabla^2 f_i(\mathbf{w})\| =$$

5. Fin $L_{max} =$.

Exercise 2: Calculating smoothness constants (1)

Ridge regression

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda\|\mathbf{w}\|^2$$

Compute the smoothness constant L_i and L_{max}.

1. $f_i(\mathbf{w}) = (y_i - \mathbf{x}_i^T\mathbf{w})^2 + \lambda\|\mathbf{w}\|^2$.
2. Compute $\nabla f_i(\mathbf{w})$.

$$\nabla f_i(\mathbf{w}) = -2(y_i - \mathbf{x}_i^T\mathbf{w})\mathbf{x}_i + 2\lambda\mathbf{w}$$

3. Compute $\nabla^2 f_i(\mathbf{w})$.

$$\nabla^2 f_i(\mathbf{w}) = 2\mathbf{x}_i\mathbf{x}_i^\top + 2\lambda\mathbf{I}$$

4. Find L_i.

$$\|\nabla^2 f_i(\mathbf{w})\| =\leq 2\|\mathbf{x}_i\|^2 + 2\lambda = L_i$$

5. Fin $L_{max} = 2(\lambda + \max_i \|\mathbf{x}_i\|^2)$.

Exercise 3: Calculating smoothness constants (2)

Logistic regression

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log(1+\exp(-y_i \mathbf{x}_i^\top \mathbf{w})) + \lambda\|\mathbf{w}\|^2$$

Compute the smoothness constant L_i and L_{max}.

1. $f_i(\mathbf{w}) = \log(1+\exp(-y_i \mathbf{x}_i^\top \mathbf{w})) + \lambda\|\mathbf{w}\|^2$.
2. Compute $\nabla f_i(\mathbf{w}) =$
3. Compute $\nabla^2 f_i(\mathbf{w})$

$$\nabla^2 f_i(\mathbf{w}) =$$

4. Find L_i.

$$\nabla^2 f_i(\mathbf{w}) \preceq \qquad\qquad \text{(hint } e^t/(1+e^t)^2 \leq \frac{1}{4}\text{)}$$

5. Find $L_{max} =$.

Exercise 3: Calculating smoothness constants (2)

Logistic regression

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log(1+\exp(-y_i \mathbf{x}_i^\top \mathbf{w})) + \lambda\|\mathbf{w}\|^2$$

Compute the smoothness constant L_i and L_{max}.

1. $f_i(\mathbf{w}) = \log(1+\exp(-y_i \mathbf{x}_i^\top \mathbf{w})) + \lambda\|\mathbf{w}\|^2$.
2. Compute $\nabla f_i(\mathbf{w}) = \frac{-y_i \mathbf{x}_i}{1+\exp(y_i \mathbf{x}_i^\top \mathbf{w})} + 2\lambda \mathbf{w}$
3. Compute $\nabla^2 f_i(\mathbf{w})$

$$\nabla^2 f_i(\mathbf{w}) =$$

4. Find L_i.

$$\nabla^2 f_i(\mathbf{w}) \preceq \qquad \qquad \text{(hint } e^t/(1+e^t)^2 \leq \frac{1}{4}\text{)}$$

5. Find $L_{max} =$.

Exercise 3: Calculating smoothness constants (2)

Logistic regression

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log(1+\exp(-y_i\mathbf{x}_i^\top \mathbf{w})) + \lambda\|\mathbf{w}\|^2$$

Compute the smoothness constant L_i and L_{max}.

1. $f_i(\mathbf{w}) = \log(1+\exp(-y_i\mathbf{x}_i^\top \mathbf{w})) + \lambda\|\mathbf{w}\|^2$.
2. Compute $\nabla f_i(\mathbf{w}) = \frac{-y_i\mathbf{x}_i}{1+\exp(y_i\mathbf{x}_i^\top \mathbf{w})} + 2\lambda\mathbf{w}$
3. Compute $\nabla^2 f_i(\mathbf{w})$

$$\nabla^2 f_i(\mathbf{w}) = \frac{\mathbf{x}_i\mathbf{x}_i^\top \exp(y_i\mathbf{x}_i^\top \mathbf{w})}{(1+\exp(y_i\mathbf{x}_i^\top \mathbf{w}))^2} + 2\lambda\mathbf{I}$$

4. Find L_i.

$$\nabla^2 f_i(\mathbf{w}) \preceq \frac{\|\mathbf{x}_i\|^2}{4}\mathbf{I} + 2\lambda\mathbf{I} = L_i\mathbf{I} \qquad \text{(hint } e^t/(1+e^t)^2 \leq \frac{1}{4}\text{)}$$

5. Find $L_{max} = \frac{\max_i \|\mathbf{x}_i\|^2}{4} + 2\lambda$.

SGD with decreasing step size

Convergence for strongly convex and smooth function with $\rho^{(k)} = O(\frac{1}{k})$

If $F = \frac{1}{n}\sum_i f_i$ μ-strongly convex with $\forall i,\ f_i$ is L_i-smooth with $\mathcal{K} = \frac{L_{max}}{\mu}$ and the step size is

$$\rho^{(k)} = \begin{cases} \frac{1}{2L_{max}} & \text{if } k \leq 4\lceil\mathcal{K}\rceil \\ \frac{2k+1}{(k+1)^2\mu} & \text{else} \end{cases}$$

for $k > 4\lceil\mathcal{K}\rceil$ we have for SGD:

$$\mathbb{E}[\|\mathbf{x}^{(k)} - \mathbf{x}^\star\|^2] \leq \frac{8\sigma^2}{\mu^2 k} + \frac{16\lceil\mathcal{K}\rceil^2\|\mathbf{x}^{(0)} - \mathbf{x}^\star\|^2}{e^2k^2} \tag{6}$$

Convergence for smooth function with $\rho^{(k)} = O(\frac{1}{\sqrt{k}})$

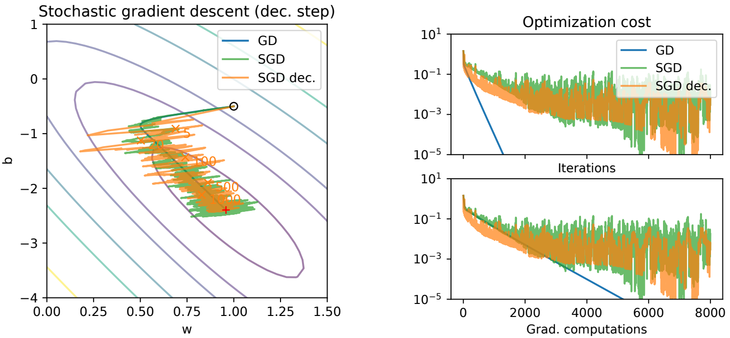
If $F = \frac{1}{n}\sum_i f_i$ with $\forall i,\ f_i$ is L_i-smooth and $\rho^{(k)} = \frac{\rho}{\sqrt{1+k}}$ and $\rho \leq \frac{1}{4L_{max}}$ we have for SGD:

$$\mathbb{E}[F(\bar{\mathbf{x}}^{(k)}) - F(\mathbf{x}^\star)] \leq \frac{\|\mathbf{x}^{(0)} - \mathbf{x}^\star\|^2 + 2\rho(F(\bar{\mathbf{x}}^{(0)}) - F(\mathbf{x}^\star))}{2\rho\sqrt{k-1}} + \frac{2\sigma^2(\log(k)+1)}{\sqrt{k-1}} \tag{7}$$

with $\bar{\mathbf{x}}^{(k)} = \frac{1}{k+1}\sum_{i=0}^k \mathbf{x}^{(i)}$.

See details in [Garrigos and Gower, 2023]

Example of decreasing step SGD



Discussion

- Decreasing step size : $\rho^{(k)} = \frac{1}{\sqrt{k}}$
- Slow convergence but less noise for large number of iterations.
- Complexity $\mathcal{O}(d)$ per iteration.

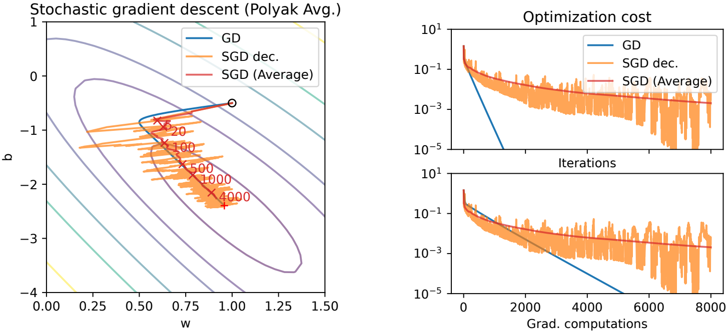
SGD with averaging (SGDA)

SGD with late start averaging

Initialize $\mathbf{x}^{(0)}$ set $s_0 \geq 0$
for $k = 0, 1, 2, \ldots$ **do**
  $i^{(k)} \leftarrow$ randomly pick an index $i \in \{1, \ldots, n\}$
  $\mathbf{d}^{(k)} \leftarrow -\nabla_{\mathbf{x}} f_{i^{(k)}}(\mathbf{x}^{(k)})$
  $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \rho^{(k)} \mathbf{d}^{(k)}$
  if $k \geq s_0$ **then**
    $\bar{\mathbf{x}}^{(k)} = \frac{1}{k - s_0} \sum_{i=s_0}^{k} \mathbf{x}^{(i)}$
  else
    $\bar{\mathbf{x}}^{(k)} = \mathbf{x}^{(k)}$
  end if
end for

- Principle : Averaging of the iterates after a certain number of steps to compensate oscillations around optimality.
- Convergence of the average $\bar{\mathbf{x}}^{(k)}$ to the optimality in $O(\frac{1}{\sqrt{k}})$ for L_i smooth and convex functions f_i [Polyak and Juditsky, 1992].
- Convergence remains slow because averaging slows changes.

Example of SGD with averaging



Discussion

- Decreasing step size : $\rho^{(k)} = \frac{1}{\sqrt{k}}$
- Slow convergence of $\bar{\mathbf{x}}^{(k)}$ but less noise that SGD.
- Complexity $\mathcal{O}(d)$ per iteration (how is that implemented?).

Convergence of SGD VS GD

Iteration complexity for a linear model is with d parameters and n samples and k iterations.

On strongly convex and smooth functions

Method	Cost 1 iter.	Convergence	Nb. iter.	Running time
GD	nd	$\exp(-k/\kappa)$	$\kappa \log(1/\epsilon)$	$nd\kappa \log(1/\epsilon)$
SGD ($O(\frac{1}{k})$ step)	d	κ/k	κ/ϵ	$d\kappa/\epsilon$

- Conditioning of the problem is $\kappa = \frac{L_{max}}{\mu}$.
- SGD more efficient when $n \gg \frac{1}{\epsilon \log(\epsilon)}$ is very large.

On smooth functions

Method	Cost 1 iter.	Convergence	Nb. iter.	Running time
GD	nd	$1/k$	$1/\epsilon$	dn/ϵ
AGD	nd	$1/k^2$	$1/\sqrt{\epsilon}$	$dn/\sqrt{\epsilon}$
SGDA ($O(\frac{1}{\sqrt{k}})$ step)	d	$1/\sqrt{k}$	$1/\epsilon^2$	d/ϵ^2

- SGD more efficient than GD when $n \gg \frac{1}{\epsilon}$ is very large.

Limits of SGD

- Convergence remains slow in practice because of gradient noise.
- Better estimation of the gradient can be done with variance reduction methods.

Stochastic Variance Reduced methods

Principle

- Keep iteration cost of SVG (compute only one gradient $\nabla f_{i^{(k)}}$).
- Use and estimate $\mathbf{g}^{(k)} \approx \nabla F(\mathbf{x}^{(k)})$ with low variance updated (for cheap) at each step.
- Use $\mathbf{g}^{(k)}$ to compute the descent update.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \rho^{(k)}\mathbf{g}^{(k)}$$

What we want for $\mathbf{g}^{(k)}$

- Unbiased estimator of the gradient $\nabla F(\mathbf{x}^{(k)})$:

$$\mathbb{E}_{i\sim\frac{1}{n}}[\mathbf{g}^{(k)}] = \nabla F(\mathbf{x}^{(k)})$$

- Low variance $\mathbb{VAR}[\mathbf{g}^{(k)}] = \mathbb{E}[\|\mathbf{g}^{(k)} - \nabla F(\mathbf{x}^{(k)})\|^2]$ for faster convergence.
- Convergence in L2 to 0 at solution (no need for decreasing step size):

$$\lim_{\mathbf{x}^{(k)}\to\mathbf{x}^\star} \mathbb{E}[\|\mathbf{g}^{(k)}\|^2] = 0$$

Controling the variance with covariates

Controlled Stochastic Reformulation

- **Covariate function** : $\mathbf{z}_i$ is a function of the sample i, $\forall i \in 1, \dots, n$.
- Reformulation of original problem:

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) = \mathbb{E}_{i\sim\frac{1}{n}}[f_i(\mathbf{x})] &= \mathbb{E}_{i\sim\frac{1}{n}}[f_i(\mathbf{x}) - z_i(\mathbf{x}) + z_i(\mathbf{x})] \\ &= \mathbb{E}_{i\sim\frac{1}{n}}[f_i(\mathbf{x}) - z_i(\mathbf{x}) + \mathbb{E}_{i\sim\frac{1}{n}}[z_i(\mathbf{x})]]\end{aligned}$$

- Equivalent optimization problem but one can use the gradient estimation for sample i:

$$\mathbf{g}_i = \nabla f_i(\mathbf{x}) - \nabla z_i(\mathbf{x}) + \mathbb{E}_{i\sim\frac{1}{n}}[\nabla z_i(\mathbf{x})]$$

- How to choose z_i to control the variance?

Covariates

Let x and z two random variables, we say that x and z are covariates if:

$$\mathrm{cov}(x, z) = \mathbb{E}[(x - \mathbb{E}[x])(z - \mathbb{E}[z])] \geq 0$$

Covariates and variance reduction

Variance reduced estimate

When x and z are covariates one can define the variance reduced estimate:

$$x_z = x - z + \mathbb{E}[z]$$

Exercise 4: Properties of variance reduction

1. Compute $\mathbb{E}[x_z] = \mathbb{E}[x]$
2. Compute $\mathbb{VAR}[x_z] = \mathbb{E}[(x_z - \mathbb{E}[x_z])^2]$

$$\begin{aligned} \mathbb{VAR}[x_z] &= \mathbb{E}[(x_z - \mathbb{E}[x_z])^2] \\ &= \end{aligned}$$

3. Under which condition is $\mathbb{VAR}[x_z] \leq \mathbb{VAR}[x]$?

Covariates and variance reduction

Variance reduced estimate

When x and z are covariates one can define the variance reduced estimate:

$$x_z = x - z + \mathbb{E}[z]$$

Exercise 4: Properties of variance reduction

1. Compute $\mathbb{E}[x_z] = \mathbb{E}[x]$
2. Compute $\mathbb{VAR}[x_z] = \mathbb{E}[(x_z - \mathbb{E}[x_z])^2]$

$$\begin{aligned} \mathbb{VAR}[x_z] &= \mathbb{E}[(x_z - \mathbb{E}[x_z])^2] \\ &= \mathbb{E}[(x - \mathbb{E}[x] - (z - \mathbb{E}[z]))^2] \\ &= \mathbb{E}[(x - \mathbb{E}[x])^2] + \mathbb{E}[(z - \mathbb{E}[z])^2] - 2\mathbb{E}[(x - \mathbb{E}[x])(z - \mathbb{E}[z])] \\ &= \mathbb{VAR}[x] + \mathbb{VAR}[z] - 2\text{cov}(x, z) \end{aligned}$$

3. Under which condition is $\mathbb{VAR}[x_z] \leq \mathbb{VAR}[x]$?

Covariates and variance reduction

Variance reduced estimate

When x and z are covariates one can define the variance reduced estimate:

$$x_z = x - z + \mathbb{E}[z]$$

Exercise 4: Properties of variance reduction

1. Compute $\mathbb{E}[x_z] = \mathbb{E}[x]$
2. Compute $\mathbb{VAR}[x_z] = \mathbb{E}[(x_z - \mathbb{E}[x_z])^2]$

$$\begin{aligned}
\mathbb{VAR}[x_z] &= \mathbb{E}[(x_z - \mathbb{E}[x_z])^2] \\
&= \mathbb{E}[(x - \mathbb{E}[x] - (z - \mathbb{E}[z]))^2] \\
&= \mathbb{E}[(x - \mathbb{E}[x])^2] + \mathbb{E}[(z - \mathbb{E}[z])^2] - 2\mathbb{E}[(x - \mathbb{E}[x])(z - \mathbb{E}[z])] \\
&= \mathbb{VAR}[x] + \mathbb{VAR}[z] - 2\text{cov}(x, z)
\end{aligned}$$

3. Under which condition is $\mathbb{VAR}[x_z] \leq \mathbb{VAR}[x]$?

$$\text{cov}(x, z) \geq \frac{1}{2}\mathbb{VAR}[z]$$

the larger the correlation the better the variance reduction.

Stochastic Variance Reduced Gradient (SVRG)

Principle of SVRG [Johnson and Zhang, 2013]

- Use covariate function z_i that is a linear approximation of f_i:

$$z_i(\mathbf{x}) = f_i(\tilde{\mathbf{x}}) + \nabla f_i(\tilde{\mathbf{x}})^\top (\mathbf{x} - \tilde{\mathbf{x}}) \tag{8}$$

  where $\tilde{\mathbf{x}}$ is a reference (anchor) point.

- The gradient $\mathbf{g}_i$ with the variance reduced estimate:

$$\mathbf{g}_i = \nabla f_i(\mathbf{x}) - \nabla f_i(\tilde{\mathbf{x}}) + \nabla F(\tilde{\mathbf{x}})$$

- The variance of the gradient estimation is:

$$\begin{aligned}
\mathbb{VAR}[\mathbf{g}_i] = &= \mathbb{E}[\|\nabla f_i(\mathbf{x}) - \nabla f_i(\tilde{\mathbf{x}}) - \nabla F(\mathbf{x}) + \nabla F(\tilde{\mathbf{x}})\|^2] \\
&\underset{3}{\leq} 2\mathbb{E}[\|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2] + 2\mathbb{E}[\|\nabla f_i(\tilde{\mathbf{x}}) - \nabla F(\tilde{\mathbf{x}})\|^2] \\
&\leq 2(L_{max}^2 + L^2)\|\mathbf{x} - \tilde{\mathbf{x}}\|^2
\end{aligned}$$

  Smaller variance when $\mathbf{x}$ is close to $\tilde{\mathbf{x}}$.

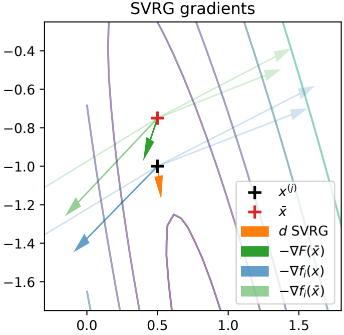
---

[3] Use $\|\mathbf{x} + \mathbf{y}\|^2 \leq 2\|\mathbf{x}\|^2 + 2\|\mathbf{y}\|^2$

Algorithm of SVRG

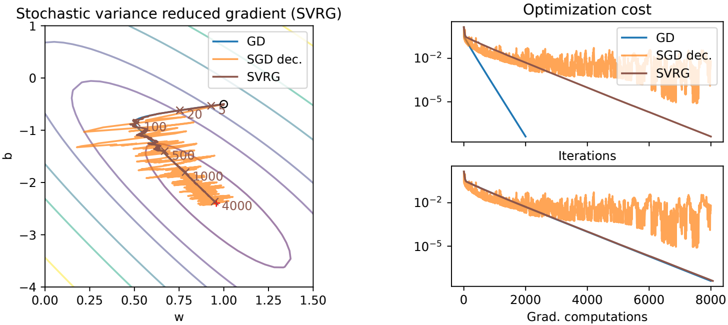
Algorithm of SVRG [Johnson and Zhang, 2013]

Initialize $\mathbf{x}^{(0)}$, $\tilde{\mathbf{x}}^{(0)} = \mathbf{x}^{(0)}$
for $k = 0, 1, 2, \ldots$ **do**
  $\mathbf{x}^{(0)} \leftarrow \tilde{\mathbf{x}}^{(k)}$
  for $j = 0, \ldots, M-1$ **do**
    $i \leftarrow$ randomly pick an index $i \in \{1, \ldots, n\}$
    $\mathbf{g} = \nabla f_i(\mathbf{x}^{(j)}) - \nabla f_i(\tilde{\mathbf{x}}^{(k)}) + \nabla F(\tilde{\mathbf{x}}^{(k)})$
    $\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} - \rho \mathbf{g}$
  end for
  $\tilde{\mathbf{x}}^{(k+1)} = \mathbf{x}^{(M)}$
end for



- The gradient g is the variance reduced estimate of the gradient.
- The anchor point $\tilde{\mathbf{x}}^{(k)}$ is updated every M steps.
- The full gradient $\nabla F(\tilde{\mathbf{x}}^{(k)})$ is computed when anchor point is updated.
- Need to choose the parameter M.
- Convergence in $O(e^{-Ck})$ for strongly convex and smooth functions and M sufficiently large (same as GD because full gradient...).

Example of SVRG



Discussion

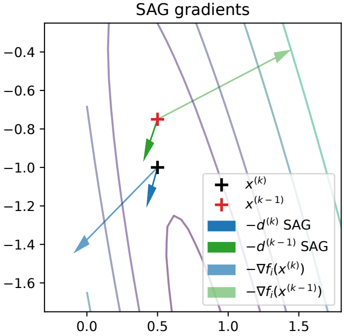
- Fixed step : $\rho^{(k)} = 0.02$ (same as GD)
- $M = 500 = 125 * n$
- Convergence in $O(e^{-Ck})$ similar to GD for strongly convex and smooth functions.
- Similar speed as GD in term of gradient computation (full gradient every M iter.).

Stochastic Average Gradient (SAG)

Stochastic Average Gradient (SAG) [Roux et al., 2012]

1: Initialize $\mathbf{x}^{(0)}, \mathbf{g}_i = \nabla f_i(\mathbf{x}^{(0)})\ \forall i$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad i^{(k)} \leftarrow$ randomly pick an index $i \in \{1, \ldots, n\}$
4: $\quad \mathbf{g}_{i^{(k)}} \leftarrow \nabla_{\mathbf{x}} f_{i^{(k)}}(\mathbf{x})$
5: $\quad \mathbf{d}^{(k)} \leftarrow -\frac{1}{n}\sum_i \mathbf{g}_i$
6: $\quad \mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \rho \mathbf{d}^{(k)}$
7: **end for**

- Keep in memory all previous computed gradients $\mathbf{g}_i$, update only for sample $i^{(k)}$.
- Iteration is $O(d)$, memory is $O(nd)$.
- **Convergence speed** [Roux et al., 2012]

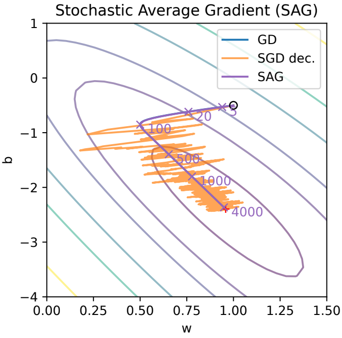


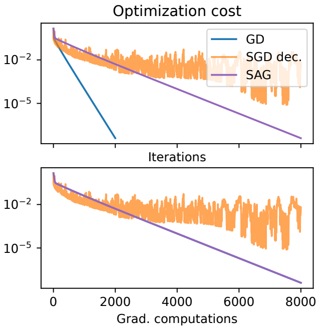
$$E[F(\bar{\mathbf{x}}^{(k)}) - F(\mathbf{x}^\star)] = \begin{cases} O(\frac{1}{k}) & \text{for } F \text{ convex} \\ O(e^{-Ck}) & \text{for } F \text{ strongly convex} \end{cases}$$

Exercise 5: Efficient implementation of SAG

- How to implement (reformulate) line 5 to avoid $O(n)$ complexity?
- For a linear model with $f_i(\mathbf{x}) = l_i(\mathbf{a}_i^\top \mathbf{x})$, do we weed to store all gradients $\mathbf{g}_i$?

Example of Stochastic Average Gradient (SAG)





Discussion

- Constant step size : $\rho^{(k)} = 0.02$
- Fast convergence because the problem is strongly convex..
- One GD iter $\equiv 4$ SGD iter (since $n = 4$).
- SAG complexity $O(d)$ per iteration (but $O(nd)$ in memory).

SAGA: Stochastic Average Gradient Accelerated

SAGA [Defazio et al., 2014]

1: Initialize $\mathbf{x}^{(0)}, \mathbf{g}_i = \nabla f_i(\mathbf{x}^{(0)})\ \forall i$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad i^{(k)} \leftarrow$ randomly pick an index $i \in \{1, \ldots, n\}$
4: $\quad \mathbf{d}^{(k)} \leftarrow -\left(\nabla_{\mathbf{x}} f_{i^{(k)}}(\mathbf{x}^{(k)}) - \mathbf{g}_{i^{(k)}} + \frac{1}{n}\sum_i \mathbf{g}_i\right)$
5: $\quad \mathbf{g}_{i^{(k)}} \leftarrow \nabla_{\mathbf{x}} f_{i^{(k)}}(\mathbf{x}^{(k)})$
6: $\quad \mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \rho \mathbf{d}^{(k)}$
7: $\quad \mathbf{x}^{(k+1)} \leftarrow \mathbf{prox}_{\rho h}(\mathbf{x}^{(k+1)})$
8: **end for**

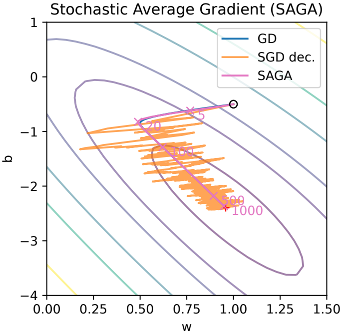
- Minimizes the following problem:

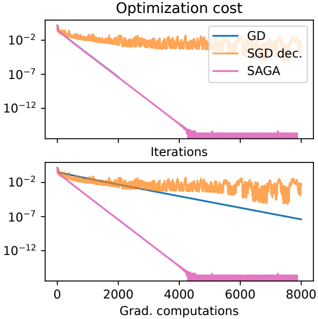
$$\min_{\mathbf{x}} \quad F(\mathbf{x}) + h(x) = \frac{1}{n}\sum_i f_i(\mathbf{x}) + h(\mathbf{x})$$

- SAGA is a variant of SAG that can handle proximal operators.
- **Convergence speed** is same as SAG but better constant [Defazio et al., 2014]

$$E[F(\bar{\mathbf{x}}^{(k)}) - F(\mathbf{x}^\star)] = \begin{cases} O(\frac{1}{k}) & \text{for } F \text{ convex} \\ O(e^{-Ck}) & \text{for } F \text{ strongly convex} \end{cases}$$

Example of SAGA

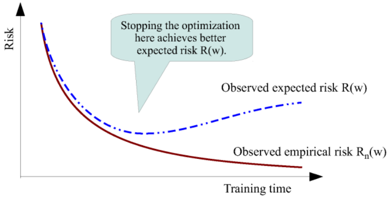




Discussion

- Constant step size : $\rho^{(k)} = 0.02$
- Fast convergence because the problem is strongly convex..
- One GD iter $\equiv 4$ SGD iter (since $n = 4$).
- SAGA complexity $\mathcal{O}(d)$ per iteration (but $O(n)$ in memory for linear models).

SGD in machine learning



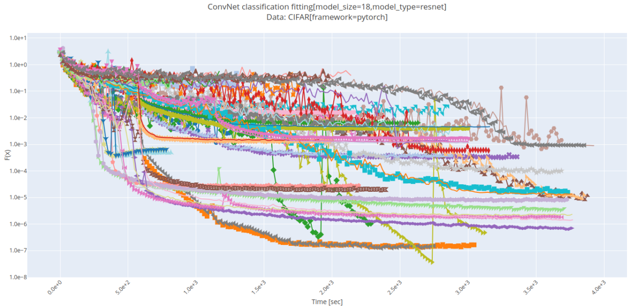
Large scale optimization [Bottou, 2010, Bottou et al., 2018]

- Used for training linear and non-linear models on very large datasets.
- State of the art algorithm for linear SVM, logistic regression, least square.
- Classification (SVM,Logistic) : `sklearn.linear_model.SGDClassifier`.
- Regression (least square, huber) : `sklearn.linear_model.SGDRegressor`.

Efficient implementation

- Minibatches (compute stochastic gradient on multiple samples).
- Sparse implementation for sparse data.
- Parallel implementation on CPU/GPU.
- Early stopping can be used as regularization.

SGD in deep learning



Training Neural Networks with SGD

- Usually use fixed step or scheduling of the step decrease.
- Use early stopping as regularization (but not always : double descent).
- Works very well on continuous, nonconvex problems but not very well understood.
- Several momentum averaging and adaptive step size strategies:
  - Momentum and Accelerated gradients [Nesterov, 1983]
  - RMSPROP [Tieleman and Hinton, 2012].
  - Adaptive gradient step ADAGRAD [Duchi et al., 2011].
  - Adaptive Moment estimation ADAM [Kingma and Ba, 2014].

Complexity of GD methods

- Iteration complexity for a linear model is with d parameters and n samples.
- Conditioning of the problem is $\kappa = \frac{L}{\mu}$ or $\kappa = \frac{L_{max}}{\mu}$ for SGD.

On strongly convex and smooth functions

Method	1 iter.	Convergence	Nb. iter.	Running time
GD	nd	$\exp(-k/\kappa)$	$\kappa \log(1/\epsilon)$	$nd\kappa \log(1/\epsilon)$
SGD ($O(\frac{1}{k})$ step)	d	κ/k	κ/ϵ	$d\kappa/\epsilon$
SAG(A)/SVRG	d	$1/k$	$(n+\kappa)\log(1/\epsilon)$	$d(n+\kappa)\log(1/\epsilon)$

On smooth functions

Method	Cost 1 iter.	Convergence	Nb. iter.	Running time
GD	nd	$1/k$	$1/\epsilon$	dn/ϵ
AGD	nd	$1/k^2$	$1/\sqrt{\epsilon}$	$dn/\sqrt{\epsilon}$
SGDA ($O(\frac{1}{\sqrt{k}})$ step)	d	$1/\sqrt{k}$	$1/\epsilon^2$	d/ϵ^2
SAG(A)/SVRG	d	$\sqrt{n}/k$	$\sqrt{n}/\epsilon$	$d\sqrt{n}/\epsilon$

- SGD and variance reduction methods are more efficient for large n.
- SAGA only needs smoothness params but require to store gradients.
- SVRG is $O(d)$ in memory but require full regular full gradienst (+ param M).
- Accelerated version of SAGA and SVRG are also available [Lin et al., 2018].

References I

Bottou, L. (2010).
Large-scale machine learning with stochastic gradient descent.
In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

Bottou, L., Curtis, F. E., and Nocedal, J. (2018).
Optimization methods for large-scale machine learning.
SIAM review, 60(2):223–311.

Defazio, A., Bach, F., and Lacoste-Julien, S. (2014).
Saga: A fast incremental gradient method with support for non-strongly convex composite objectives.
In *Advances in neural information processing systems*, pages 1646–1654.

Duchi, J., Hazan, E., and Singer, Y. (2011).
Adaptive subgradient methods for online learning and stochastic optimization.
Journal of machine learning research, 12(Jul):2121–2159.

Garrigos, G. and Gower, R. M. (2023).
Handbook of convergence theorems for (stochastic) gradient methods.
arXiv preprint arXiv:2301.11235.

References II

Gower, R. M., Loizou, N., Qian, X., Sailanbayev, A., Shulgin, E., and Richtárik, P. (2019).
Sgd: General analysis and improved rates.
In *International conference on machine learning*, pages 5200–5209. PMLR.

Johnson, R. and Zhang, T. (2013).
Accelerating stochastic gradient descent using predictive variance reduction.
In *Advances in neural information processing systems*, pages 315–323.

Kingma, D. P. and Ba, J. (2014).
Adam: A method for stochastic optimization.
arXiv preprint arXiv:1412.6980.

Lin, H., Mairal, J., and Harchaoui, Z. (2018).
Catalyst acceleration for first-order convex optimization: from theory to practice.
Journal of Machine Learning Research, 18(212):1–54.

Nesterov, Y. E. (1983).
A method for solving the convex programming problem with convergence rate o (1/k^ 2).
In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547.

References III

Polyak, B. T. and Juditsky, A. B. (1992).
Acceleration of stochastic approximation by averaging.
SIAM journal on control and optimization, 30(4):838–855.

Roux, N. L., Schmidt, M., and Bach, F. R. (2012).
A stochastic gradient method with an exponential convergence _rate for finite training sets.
In *Advances in neural information processing systems*, pages 2663–2671.

Tieleman, T. and Hinton, G. (2012).
Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude.
COURSERA: Neural networks for machine learning, 4(2):26–31.