# Optimization for Machine Learning

## Lecture 9: Finite Sum Optimization
## Variance-reduced Stochastic Methods

**Sebastian Stich**

CISPA – https://cms.cispa.saarland/optml24/

June 18, 2024

# Course Outlook

- course format: mini-survey
- course project: on plan? questions?

# Group Project Timeline

- Group registration between May 28 – June 4 (register on CMS)
  - groups of 2–3
- before June 18: get in touch with the contact person and schedule a meeting!
  - read the related literature
  - prepare a list of **research goals and tasks**
- before June 25: meet with your contact person
  - zoom meeting, 30-60min, can also be in-person
  - **discuss your research plan**
  - **ask questions** about things you do not understand
- July 16: Poster presentation (& suggested report submission)
  - (note that the **poster printing deadline** is a bit earlier, TBA!)
- July 26: last possible date to submit the report

There will be no exercise sheets in the weeks of June 25/July 2 — you can also discuss the project ~~in the exercise session~~ with your contact person.

# Recap

- We formulated (& simplified) federated learning as a finite sum optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \tag{FS}$$

- We observed optimization difficulties in the heterogeneous setting: $f_i(\mathbf{x}^\star) \neq f_j(\mathbf{x}^\star)$.
- Specialized methods, like SCAFFOLD can mitigate "drift". In this lecture, we will see that drift correction can be seen as a special case of variance reduction.

# SGD vs. GD for Finite Sum Problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

Table: Complexity for smooth and strongly convex problems: $\kappa = L/\mu$

| | iteration complexity | per-iteration cost | total cost |
|---|---|---|---|
| GD | | | |
| SGD | | | |

- GD converges faster but with expensive iteration cost
- SGD converges slowly but with cheap iteration cost
- SGD is more appealing for large $n$ and moderate accuracy $\epsilon$.

# Can we achieve both worlds?

- GD: deterministic, linear rate, $O(n)$ iteration cost, fixed stepsize.
- SGD: stochastic, sublinear rate, $O(1)$ iteration cost, diminishing stepsize.

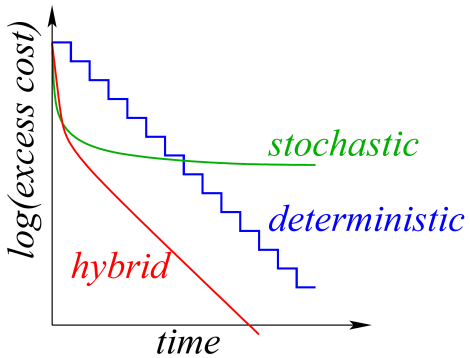



Figure from Bach's NeurIPS 2016 tutorial

# Observation: reducing variance is the key

$$\mathbb{E}[\|\nabla f_{i_t}(\mathbf{x}_t) - \nabla F(\mathbf{x}_t)\|_2^2] \leq \sigma^2$$

A high variance slows down the convergence when seeking high accuracy:

$$\mathcal{O}\left(\kappa \log \frac{1}{\epsilon} + \frac{\sigma^2}{\mu\epsilon}\right)$$

**Q:** Can we design gradient estimators with reduced variance?

# Stochastic Varianced-reduced Methods

Stochastic variance-reduced methods are as cheap to update as SGD, but have as fast convergence as full gradient descent.

**Popular algorithms**:

- SAG (stochastic average gradient) [Le Roux et al., 2012]
- SVRG (stochastic variance-reduced gradient) [Johnson and Zhang, 2013]
- SDCA (stochastic dual coordinate ascent) [Shalev-Shwartz and Zhang, 2013]
- SAGA (stochastic average gradient amélioré) [Defazio et al., 2014]
- Many many others: MISO, Finito, Catalyst-SVRG, S2GD, etc.
- Recent variants for nonconvex setting: SPIDER, SARAH, STORM, PAGE, etc.

# Preview of VR Methods

| Algorithm | # of Iterations | Per-iteration Cost |
|---|---|---|
| GD | $O\left(\kappa \log \frac{1}{\epsilon}\right)$ | $O(n)$ |
| SGD | $O\left(\frac{\kappa}{\epsilon}\right)$ | $O(1)$ |
| VR | $O\left((n+\kappa) \log \frac{1}{\epsilon}\right)$ | $O(1)$ |

Table: Complexity of strongly convex and smooth finite-sum optimization

# Preview of VR Methods

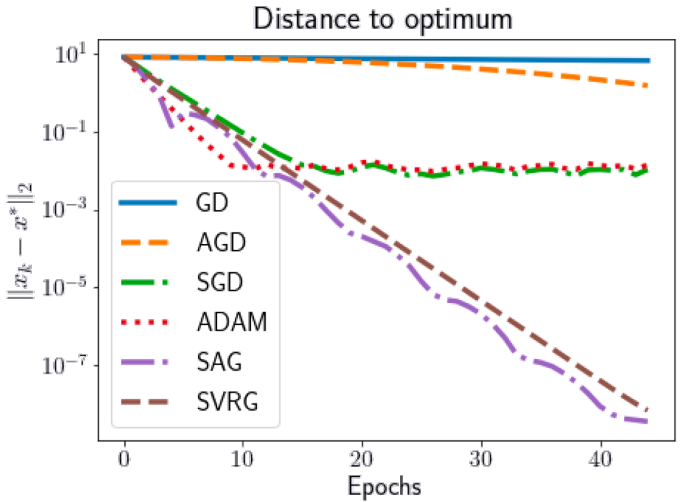



Figure: Logistic regression on `mushrooms` dataset with $n = 8124$ [Gow20]

# Lecture Outline

Variance Reduction Techniques

Stochastic Variance-reduced Methods

- SAG/SAGA
- SVRG

# Classical Variance Reduction Techniques

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

- Mini-batching: Use the average of gradients from a random subset

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \frac{1}{|B_t|}\sum_{i\in B_t} \nabla f_i(\mathbf{x}_t)$$

NB: Variance reduction comes at a computational cost.

- Momentum: add momentum to the gradient step

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \hat{\mathbf{m}}_t, \text{ where } \hat{\mathbf{m}}_t = c \cdot \sum_{\tau=1}^{t} \alpha^{t-\tau} \nabla f_{i_\tau}(\mathbf{x}_\tau)$$

NB: Here $\mathbf{m}_t$ is the weighted average of the past stochastic gradients.

# A Modern Variance Reduction Technique

Suppose we want to estimate $\theta = \mathbb{E}[X]$, $X$ is a random variable.
Consider the **point estimator** for $\theta$:

$$\hat{\Theta} := X - Y$$

- $\mathbb{E}[X - Y] = \theta$ if and only if $\mathbb{E}[Y] = 0$
- $\mathbb{V}[X - Y]$ is less than $\mathbb{V}[X]$ if $Y$ is highly positively correlated with $X$.

# A Modern Variance Reduction Technique

Suppose $X$ is positively correlated with $Y$ and we can compute $\mathbb{E}[Y]$.

**Point Estimator**:

$$\hat{\Theta}_\alpha = \alpha(X - Y) + \mathbb{E}[Y], \quad (0 \leq \alpha \leq 1).$$

$$\mathbb{E}[\hat{\Theta}_\alpha] = \alpha\mathbb{E}[X] + (1 - \alpha)\mathbb{E}[Y]$$
$$\mathbb{V}[\hat{\Theta}_\alpha] = \alpha^2(\mathbb{V}[X] + \mathbb{V}[Y] - 2\mathrm{Cov}[X, Y])$$

- If covariance is sufficiently large, then $\mathbb{V}[\hat{\Theta}_\alpha] \leq \mathbb{V}[X]$.

# Motivation

**Q:** Can we design cheap gradient estimators with reduced variance?

**Key Idea:** if $\mathbf{x}_t$ is not too far away from previous iterates, then we can leverage previous gradient information to construct positively correlated control variates.

- SGD: estimate $\nabla F(\mathbf{x}_t)$ by $\nabla f_{i_t}(\mathbf{x}_t)$
- VR: estimate $\nabla F(\mathbf{x}_t)$ by $\mathbf{g}_t := \alpha(\nabla f_{i_t}(\mathbf{x}_t) - Y) + \mathbb{E}[Y]$ such that

$$\mathbb{E}[\|\mathbf{g}_t - \nabla F(\mathbf{x}_t)\|^2] \to 0, \text{ as } t \to \infty. \qquad \text{(VR property)}$$

So how to design $Y$?

# Design Ideas

Goal: Construct $Y$ that is positively correlated to $X = \nabla f_{i_t}(\mathbf{x}_t)$:

Choice I: $Y = \nabla f_{i_t}(\mathbf{x}^\star)$, where $\mathbf{x}^\star$ is the optimal solution

- $\mathbb{E}[Y] = 0$, unrealistic but conceptually useful

Choice II: $Y = \nabla f_{i_t}(\bar{\mathbf{x}}_{i_t})$, where $\bar{\mathbf{x}}_i$ is the last point for which we evaluated $\nabla f_i(\bar{\mathbf{x}}_i)$

- $\mathbb{E}[Y] = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\bar{\mathbf{x}}_i)$, requires storage of $\{\bar{\mathbf{x}}_i\}_{i=1}^n$ or $\{\nabla f_i(\bar{\mathbf{x}}_i)\}_{i=1}^n$

Choice III: $Y = \nabla f_{i_t}(\tilde{\mathbf{x}})$, where $\tilde{\mathbf{x}}$ is some fixed reference point

- $\mathbb{E}[Y] = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{\mathbf{x}})$, requires computing the full gradient at $\tilde{\mathbf{x}}$

# Lecture Outline

Variance Reduction Techniques

Stochastic Variance-reduced Methods

- SAG/SAGA
- SVRG

# Variance Reduction Techniques for Finite Sum Problems

Goal: estimate $\theta = \nabla F(\mathbf{x}_t)$, $X = \nabla f_{i_t}(\mathbf{x}_t)$

- SGD: $\mathbf{g}_t = \nabla f_{i_t}(\mathbf{x}_t)$ $\quad [\alpha = 1, Y = 0]$
- SAG: $\mathbf{g}_t = \frac{1}{n}(\nabla f_{i_t}(\mathbf{x}_t) - \mathbf{v}_{i_t}) + \frac{1}{n}\sum_{i=1}^n \mathbf{v}_i$ $\quad [\alpha = \frac{1}{n}, Y = \mathbf{v}_{i_t}]$
- SAGA: $\mathbf{g}_t = (\nabla f_{i_t}(\mathbf{x}_t) - \mathbf{v}_{i_t}) + \frac{1}{n}\sum_{i=1}^n \mathbf{v}_i$ $\quad [\alpha = 1, Y = \mathbf{v}_{i_t}]$

  Here $\{\mathbf{v}_i, i = 1, \ldots, n\}$ are the past stored gradients for each component.
- SVRG: $\mathbf{g}_t = \nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\tilde{\mathbf{x}}) + \nabla F(\tilde{\mathbf{x}})$ $\quad [\alpha = 1, Y = \nabla f_{i_t}(\tilde{\mathbf{x}})]$

# Stochastic Average Gradient (SAG)

**Idea:** keep track of the average of $\mathbf{v}_i$ as an estimate of the full gradient

$$\mathbf{g}_t = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i^t \qquad \approx \qquad \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{x}_t) = \nabla F(\mathbf{x}_t)$$

- The past gradients are updated as:

$$\mathbf{v}_i^t = \begin{cases} \nabla f_{i_t}(\mathbf{x}_t), & \text{if } i = i_t, \\ \mathbf{v}_i^{t-1}, & \text{if } i \neq i_t. \end{cases}$$

- Equivalently, we have

$$\mathbf{g}_t = \mathbf{g}_{t-1} - \frac{1}{n}\mathbf{v}_{i_t}^{t-1} + \frac{1}{n}\nabla f_{i_t}(\mathbf{x}_t)$$

# Stochastic Average Gradient (SAG, continued)

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\gamma}{n}\sum_{i=1}^{n}\mathbf{v}_i^t, \text{ where } \mathbf{v}_i^t = \begin{cases} \nabla f_{i_t}(\mathbf{x}_t), & \text{if } i = i_t \\ \mathbf{v}_i^{t-1}, & \text{otherwise} \end{cases}$$

**Algorithm** SAG (Le Roux et al., 2012)

Initialize $\mathbf{v}_i = 0, i = 1, \ldots, n$
**for** $t = 1, 2, \ldots, T$ **do**
  Randomly pick $i_t \in \{1, 2, \ldots, n\}$
  $\mathbf{g}_t = \mathbf{g}_{t-1} - \frac{1}{n}\mathbf{v}_{i_t}$
  $\mathbf{v}_{i_t} = \nabla f_{i_t}(\mathbf{x}_t)$
  $\mathbf{g}_t = \mathbf{g}_t + \frac{1}{n}\mathbf{v}_{i_t}$
  $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma\mathbf{g}_t$
**end for**

- Biased gradient
- Cheap iteration cost
- $O(nd)$ memory cost
- Hard to analyze

# Stochastic Average Gradient (SAG, continued)

- **Linear convergence:** The first stochastic methods to enjoy linear rate using a constant stepsize for strongly-convex and smooth objectives.

  If $F$ is $\mu$-strongly convex and each $f_i$ is $L_i$-smooth and convex, setting $\gamma = 1/(16L_{\max})$, one can show that

  $$\mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}^\star)] \leq C \cdot \left(1 - \min\{\frac{1}{8n}, \frac{\mu}{16L_{\max}}\}\right)^t.$$

  Here $L_{\max} := \max\{L_1, \ldots, L_n\}$.

- **Memory cost:** $O(n)$ times higher than SGD/SVRG
- **Per-iteration cost**: one gradient evaluation
- Total complexity: $O\left((n + \kappa_{\max})\log(\frac{1}{\epsilon})\right)$.

# SAGA

**SAGA (Defazio, Bach, Lacoste-Julien, 2016):**

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \left[ (\nabla f_{i_t}(\mathbf{x}_t) - \mathbf{v}_{i_t}^{t-1}) + \frac{1}{n} \sum_{i=1}^{n} \mathbf{v}_i^{t-1} \right]$$

- Unbiased update, while SAG is biased
- Same $O(nd)$ memory cost as SAG
- Similar linear convergence rate as SAG, but has a much simpler proof

# Stochastic Variance Reduced Gradient (SVRG)

**Key idea:** Build covariates based on fixed reference point; balance the frequency of reference point update and the variance reduction.

**Algorithm** Stochastic Variance Reduced Gradient (Johnson & Zhang '13)

**for** $s = 1, 2, \ldots$ **do**
  Set $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{s-1}$ and compute $\nabla F(\tilde{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{\mathbf{x}})$ (update snapshot)
  Initialize $\mathbf{x}_0 = \tilde{\mathbf{x}}$
  **for** $t = 0, 1, \ldots, m-1$ **do**
    Randomly pick $i_t \in \{1, 2, \ldots, n\}$ and update
      $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \left( \nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\tilde{\mathbf{x}}) + \nabla F(\tilde{\mathbf{x}}) \right)$ (cheap cost)
  **end for**
  **Update** $\tilde{\mathbf{x}}^s = \frac{1}{m} \sum_{t=0}^{m-1} \mathbf{x}_t$
**end for**

# SVRG: Key Features

Intuition: the closer $\tilde{\mathbf{x}}$ is to $\mathbf{x}_t$, the smaller the variance of the gradient estimator

$$\mathbb{E}[\|\mathbf{g}_t - \nabla F(\mathbf{x}_t)\|^2] \leq \mathbb{E}[\|\nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\tilde{\mathbf{x}})\|^2] \leq L_{\max}^2 \|\mathbf{x}_t - \tilde{\mathbf{x}}\|^2$$

**Two-loop structure**:

- Outer loop: update reference point and compute its full gradient at $O(n)$ cost
- Inner loop: update iterates with variance-reduced gradient for $m$ steps
- Total of $O(n + 2m)$ component gradient evaluations at each epoch

**Compare to SAG/SAGA**

(+) Cheap memory cost, no need to store past gradients or past iterates

(−) More parameter tuning, two gradient computation per iteration

# Convergence of SVRG

## Theorem 9.1 (Johnson & Zhang, 2013)

*Assume each $f_i(\mathbf{x})$ is convex and $L_i$-smooth, $F(\mathbf{x})$ is $\mu$-strongly convex. Assume $m$ is sufficiently large and $\eta < \frac{1}{2L_{max}}$ such that $\rho = \frac{1}{\mu\eta(1-2\eta L_{max})m} + \frac{2\eta L_{max}}{1-2\eta L_{max}} < 1$, then*

$$\mathbb{E}[F(\tilde{\mathbf{x}}^s) - F(\mathbf{x}^*)] \leq \rho^s \, [F(\tilde{\mathbf{x}}^0) - F(\mathbf{x}^*)].$$

- **Linear convergence:** choose $m = O(\frac{L_{\max}}{\mu}), \eta = O(\frac{1}{L_{\max}})$ such that $\rho \in (0, \frac{1}{2})$.
- **Total complexity**:

$$O\left((2m+n)\log\frac{1}{\epsilon}\right) = O\left(\left(n + \frac{L_{\max}}{\mu}\right)\log\frac{1}{\epsilon}\right).$$

# SVRG vs. SAG/SAGA

Table: Comparisons between SVRG and SAG/SAGA

| | **SVRG** | **SAG/SAGA** |
|---|---|---|
| memory cost | $O(d)$ | $O(nd)$ |
| epoch-based | yes | no |
| # gradients per step | at least 2 | 1 |
| parameters | stepsize & epoch length | stepsize |
| unbiasedness | yes | yes/no |
| total complexity | $O\left((n+\kappa_{\max})\log\frac{1}{\epsilon}\right)$ | $O\left((n+\kappa_{\max})\log\frac{1}{\epsilon}\right)$ |

Loopless-SVRG: [Hofmann et al., 2015]

# Numerical Illustration

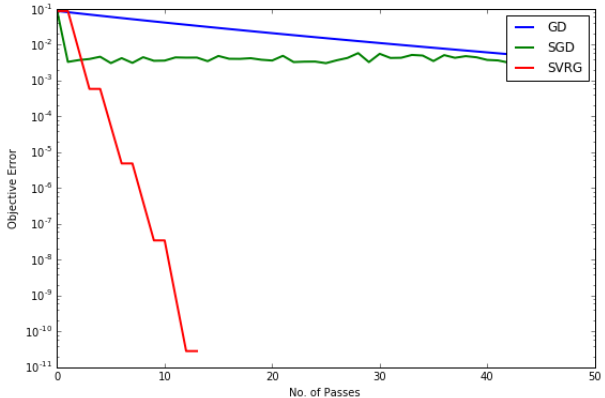



Figure: Numerical illustration among GD, SGD, SVRG on logistic regression.

# Convergence Analysis of SVRG: Key Lemma

## Lemma 9.2 (Exercise, property of smoothness)

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}^\star)\|_2^2 \leq 2L_{max}(F(\mathbf{x}) - F(\mathbf{x}^\star))$$

## Lemma 9.3 (Bound of variance)

*Denote* $\mathbf{g}_t = \nabla f_{i_t}(\mathbf{x}^t) - \nabla f_{i_t}(\tilde{\mathbf{x}}) + \nabla F(\tilde{\mathbf{x}})$. *We have*

$$\mathbb{E}[\|\mathbf{g}_t\|_2^2] \leq 4L_{max}[F(\mathbf{x}_t) - F(\mathbf{x}^\star) + F(\tilde{\mathbf{x}}) - F(\mathbf{x}^\star)].$$

# Convergence Analysis of SVRG: Proof

For notation simplicity, denote $L = L_{\max}$. From Lemma 9.3, we have

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|_2^2\right] \\
&= \|\mathbf{x}_t - \mathbf{x}^\star\|_2^2 - 2\eta(\mathbf{x}_t - \mathbf{x}^\star)^\top \mathbb{E}\left[\mathbf{g}_t\right] + \eta^2 \mathbb{E}\left[\|\mathbf{g}_t\|_2^2\right] \\
&\leq \|\mathbf{x}_t - \mathbf{x}^\star\|_2^2 - 2\eta(1 - 2L\eta)(F(\mathbf{x}_t) - F(\mathbf{x}^\star)) + 4L\eta^2\left[F(\tilde{\mathbf{x}}) - F(\mathbf{x}^\star)\right]
\end{aligned}$$

We can then establish the contraction after telescoping the sum and invoking the definition for $\tilde{\mathbf{x}}$.

# Convergence Analysis of SVRG: Proof (continued)

It follows that

$$
\begin{aligned}
&\mathbb{E}\left[\|\mathbf{x}_m - \mathbf{x}_\star\|^2\right] + 2\eta(1-2L\eta)m\mathbb{E}\left[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^\star)\right] \\
&\leq \mathbb{E}\left[\|\mathbf{x}_m - \mathbf{x}_\star\|^2\right] + 2\eta(1-2L\eta)\sum\nolimits_{t=0}^{m-1}\mathbb{E}\left[f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right] && \text{(by convexity)} \\
&\leq \mathbb{E}\left[\|\mathbf{x}_0 - \mathbf{x}^\star\|^2\right] + 4Lm\eta^2\mathbb{E}\left[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^\star)\right] && \text{(by telescoping)} \\
&\leq \mathbb{E}\left[\|\tilde{\mathbf{x}}^{s-1} - \mathbf{x}^\star\|^2\right] + 4Lm\eta^2\mathbb{E}\left[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^\star)\right] && \text{(by definition of } \mathbf{x}_0\text{)} \\
&\leq \frac{2}{\mu}\mathbb{E}\left[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^\star)\right] + 4Lm\eta^2\mathbb{E}\left[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^\star)\right] && \text{(by } \mu \text{ strongly convexity)}
\end{aligned}
$$

This further implies

$$
\mathbb{E}\left[f(\tilde{\mathbf{x}}^s) - f(\mathbf{x}^\star)\right] \leq \left[\frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta}\right]\mathbb{E}\left[f(\tilde{\mathbf{x}}^{s-1}) - f(\mathbf{x}^\star)\right].
$$

□

# Summary: Finite Sum Optimization

$$\min_{\mathbf{x}\in\mathbb{R}^d} \quad F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

($f_i$ is $L_i$-smooth and convex, $F$ is $L$-smooth and $\mu$-strongly convex)

| Algorithm | # of Iterations | Per-iteration Cost |
|---|---|---|
| GD | $O\left(\kappa \log \frac{1}{\epsilon}\right)$ | $O(n)$ |
| SGD | $O\left(\frac{\kappa_{\max}}{\epsilon}\right)$ | $O(1)$ |
| SAG/SAGA/SVRG | $O\left((n+\kappa_{\max}) \log \frac{1}{\epsilon}\right)$ | $O(1)$ |

Table: Complexity of finite-sum optimization, $\kappa = \frac{L}{\mu}$, $\kappa_{\max} = \frac{L_{\max}}{\mu}$

# Remarks

- Variance reduction technique is crucial for finite sum problems.
- In general, $L \leq L_{\max} \leq nL$. VR methods are always superior in terms of total runtime than GD.
- If $L_i = L, \forall i$, then $\kappa = \kappa_{\max}$, VR methods are much faster than GD especially when $\kappa = O(n)$.
- SGD has much worse dependency on $\epsilon$ than VR methods, which explain its poor performance when $\epsilon$ is small.

# Can we further improve the VR methods?

- Non-uniform sampling: improve to $O\left((n+\kappa_{\text{avg}})\log\frac{1}{\epsilon}\right)$

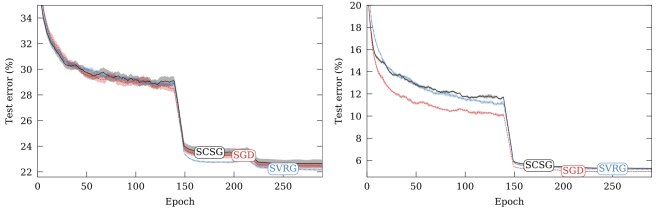
$$P(i_t = i) = \frac{L_i}{\sum_{i=1}^{n} L_i}$$

- Incorporating acceleration: can improve to $O\left((n+\sqrt{n\kappa_{\max}})\log\frac{1}{\epsilon}\right)$.

- Lower complexity bound: $O\left((n+\sqrt{n\kappa_{\max}})\log\frac{1}{\epsilon}\right)$ for the strongly-convex and smooth finite-sum problems considered
(Woodworth and Srebro, 2016; Lan and Zhou, 2018)

# Limitations?

- Challenges with practical implementations: learning rate and sampling
- Less advantage beyond smooth or strongly convex objectives or finite-sum setting
- VR may be ineffective for training neural networks [Defazio and Bottou, 2019].




(a) LeNet on CIFAR10 (b) DenseNet on CIFAR10

(c) ResNet-110 on CIFAR10 (d) ResNet-18 on ImageNet

# Lecture 9 Recap

- Finite-sum problems allow for variance reduction
  - reason: it is possible to query the stochastic oracle twice!
- Discussed several variance reduction techniques, with pros & cons

# Bibliography


R. M. Gower, M. Schmidt, F. Bach, P. Richtarik
*Variance-reduced methods for machine learning.*
Proceedings of the IEEE, 108(11), 1968-1983, 2022.

T. Hofmann, A. Lucchi, S. Lacoste-Julien, B. McWilliams
*Variance Reduced Stochastic Gradient Descent with Neighbors.*
NeurIPS, 2015.