Lecture 6: Stochastic gradient methods

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-556 (Fall 2019)
License Information for Mathematics of Data Slides

- This work is released under a [Creative Commons License](https://creativecommons.org/licenses/by-nc-sa/4.0/) with the following terms:
  - **Attribution**
    - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
  - **Non-Commercial**
    - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
  - **Share Alike**
    - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor’s work.
  - [Full Text of the License](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Mathematics of Data | Prof. Volkan Cevher, volkan.cevher@epfl.ch
Outline

- This class
  1. Stochastic programming
  2. Stochastic gradient descent
  3. Variance reduction technique

- Next class
  1. Non-convex optimization

2. A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming.

Recall: Gradient descent

**Problem (Unconstrained convex problem)**

Consider the following convex minimization problem:

\[ f^* = \min_{x \in \mathbb{R}^p} f(x) \]

\( f(x) \) is proper, closed, and convex (perhaps strongly-convex and/or smooth).

**Gradient descent**

Choose a starting point \( x^0 \) and iterate

\[ x^{k+1} = x^k - \gamma_k \nabla f(x^k) \]

where \( \gamma_k \) is a step-size to be chosen so that \( x^k \) converges to \( x^* \).
Recall: Gradient descent

Problem (Unconstrained convex problem)

Consider the following convex minimization problem:

$$f^* = \min_{x \in \mathbb{R}^p} f(x)$$

$f(x)$ is proper, closed, and convex (perhaps strongly-convex and/or smooth).

Gradient descent

Choose a starting point $x^0$ and iterate

$$x^{k+1} = x^k - \gamma_k \nabla f(x^k)$$

where $\gamma_k$ is a step-size to be chosen so that $x^k$ converges to $x^*$.  

<table>
<thead>
<tr>
<th></th>
<th>$f$ is $L$-smooth &amp; convex</th>
<th>$f$ is $L$-smooth &amp; non-convex</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD</td>
<td>$O(1/T)$ (fast)</td>
<td>$O(1/T)$ (optimal)</td>
</tr>
<tr>
<td>AGD</td>
<td>$O(1/T^2)$ (optimal)</td>
<td>$O(1/T)$ (optimal) [6]</td>
</tr>
</tbody>
</table>

Why should we study anything else?
Statistical learning

A basic statistical learning model [16]

A statistical learning model consists of the following three elements.

1. A sample of i.i.d. random variables \((a_j, b_j) \in \mathcal{A} \times \mathcal{B}, j = 1, \ldots, n\), following an *unknown* probability distribution \(P\).
2. A class (set) \(\mathcal{F}\) of functions \(f : \mathcal{A} \to \mathcal{B}\).
3. A loss function \(L : \mathcal{B} \times \mathcal{B} \to \mathbb{R}\).
A basic statistical learning model [16]

A statistical learning model consists of the following three elements.

1. A sample of i.i.d. random variables \((a_j, b_j) \in A \times B, j = 1, \ldots, n\), following an \textit{unknown} probability distribution \(P\).
2. A class (set) \(F\) of functions \(f : A \to B\).
3. A loss function \(L : B \times B \to \mathbb{R}\).

Definition (Risk)

Let \((a, b)\) follow the probability distribution \(P\) and be independent of \(\{(a_i, b_i)\}_{i=1}^n\). Then, the \textit{risk} corresponding to any \(f \in F\) is its expected loss:

\[
R(f) := \mathbb{E}_{(a, b)} [L(f(a), b)].
\]

Statistical learning seeks to find a \(f^* \in F\) that minimizes the risk, i.e., it solves

\[
f^* \in \arg\min_{f \in F} R(f).
\]

Many problems in machine learning cast into this formulation
Empirical risk minimization (ERM) I

- By the law of large numbers, we can expect that for any fixed $f \in \mathcal{F}$,

$$R(f) := \mathbb{E} [L(f(a), b)] \approx \frac{1}{n} \sum_{j=1}^{n} L(f(a_j), b_j)$$

when $n$ is large enough, with high probability.

Statistical learning with Empirical risk minimization (ERM) [16]

We approximate $f^*$ by minimizing the empirical average of the loss instead of the risk.

$$\arg \min_{f \in \mathcal{F}} \left\{ R_n(f) := \frac{1}{n} \sum_{j=1}^{n} L(f(a_j), b_j) \right\}.$$

Example: Least squares

Recall that the LS estimator is given by

$$\arg \min_{x \in \mathbb{R}^p} \left\{ \frac{1}{2n} \| b - Ax \|_2^2 \right\} = \arg \min_{x \in \mathbb{R}^p} \left\{ \frac{1}{2n} \sum_{j=1}^{n} (b_j - \langle a_j, x \rangle)^2 \right\},$$

where we define $b := (b_1, \ldots, b_n)^T$ and $a_j^T$ to be the $j$-th row of $A$. 
Empirical risk minimization (ERM) II

Example: Logistic regression

Recall the logistic regression formulation

\[
\arg\min_{x,\mu} \left\{ \frac{1}{n} \sum_{j=1}^{n} \log \left( 1 + e^{-b_j (\langle x, a_j \rangle + \mu)} \right) : x \in \mathbb{R}^p, \mu \in \mathbb{R} \right\}
\]

where \( b := (b_1, \ldots, b_n)^T \in \{-1, 1\}^n \).

Gradient descent for ERM

\[
f^{k+1} = f^k - \gamma_k \nabla R_n(f) = f^k - \gamma_k \frac{1}{n} \sum_{j=1}^{n} \nabla L(f(a_j), b_j).
\]

Computational cost per iteration is proportional to sample size \( n \), which is expensive when \( n \) is large.
Statistical learning with streaming data

Recall that statistical learning seeks to find a $f^* \in \mathcal{F}$ that minimizes the expected risk,

$$f^* \in \arg \min_{f \in \mathcal{F}} \left\{ R(f) := \mathbb{E}_{(a,b)} \left[ L(f(a), b) \right] \right\},$$

In practice, data can arrive in a streaming way.

Example: Markowitz portfolio optimization

$$f^* \equiv \min_{x \in \mathcal{X}} \left\{ \mathbb{E} \left[ |\rho - \langle x, \theta_t \rangle |^2 \right] \right\}$$

- $\rho \in \mathbb{R}$ is the desired return.
- $\mathcal{X}$ is intersection of the standard simplex and the constraint: $\langle x, \mathbb{E}[\theta_t] \rangle \geq \rho$.

Gradient method

$$f^{k+1} = f^k - \gamma_k \nabla R(f) = f^k - \gamma_k \mathbb{E}_{(a,b)}[\nabla L(f^k(a), b)].$$

This can not be implemented in practice as the distribution of $(a, b)$ is unknown.
Problem (Mathematical formulation)

Consider the following convex minimization problem:

\[
    f^* = \min_{x \in \mathbb{R}^p} \left\{ f(x) := \mathbb{E}[f(x, \theta)] \right\}
\]

- \( \theta \) is a random vector whose probability distribution is supported on set \( \Theta \).
- \( f(x) := \mathbb{E}[f(x, \theta)] \) is proper, closed, and convex.
- The solution set \( S^* := \{ x^* \in \text{dom}(f) : f(x^*) = f^* \} \) is nonempty.
Stochastic gradient descent (SGD)

1. Choose $x^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k \in \mathbb{N}} \in ]0, +\infty[^\mathbb{N}$.
2. For $k = 0, 1, \ldots$ perform:
   \[ x^{k+1} = x^k - \gamma_k G(x^k, \theta_k). \]

- $G(x^k, \theta_k)$ is an unbiased estimate of the full gradient:
  \[ \mathbb{E}[G(x^k, \theta_k)] = \nabla f(x^k). \]
Stochastic gradient descent (SGD)

1. Choose $x^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k \in \mathbb{N}} \in ]0, +\infty[\mathbb{N}$.
2. For $k = 0, 1, \ldots$ perform:

$$
x^{k+1} = x^k - \gamma_k G(x^k, \theta_k).
$$

- $G(x^k, \theta_k)$ is an unbiased estimate of the full gradient:

$$\mathbb{E}[G(x^k, \theta_k)] = \nabla f(x^k).$$

Remark

- The cost of computing $G(x^k, \theta_k)$ is $n$ times cheaper than that of $\nabla f(x^k)$.
- As $G(x^k, \theta_k)$ is an unbiased estimate of the full gradient, SG would perform well.
- We assume $\{\theta_k\}$ are jointly independent.
- SG is not a monotonic descent method.
Example: Convex optimization with finite sums

Convex optimization with finite sums

The problem

\[
\arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n} \sum_{j=1}^{n} f_j(\mathbf{x}) \right\},
\]

can be rewritten as

\[
\arg\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ f(\mathbf{x}) := \mathbb{E}_i[f_i(\mathbf{x})] \right\}, \quad i \text{ is uniformly distributed over } \{1, 2, \cdots, n\}.
\]

Stochastic gradient descent (SGD)

\[
\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k \nabla f_i(\mathbf{x}^k) \quad i \text{ is uniformly distributed over } \{1, \ldots, n\}
\]

- Note: \( \mathbb{E}_i[\nabla f_i(\mathbf{x}^k)] = \sum_{j=1}^{n} \nabla f_j(\mathbf{x}^k)/n = \nabla f(\mathbf{x}^k). \)

- The computational cost of SGD per iteration is \( p \).
Synthetic least-squares problem

$$\min_x \left\{ f(x) := \frac{1}{2n} \|Ax - b\|^2_2 : x \in \mathbb{R}^p \right\}$$

Setup

- $A := \text{randn}(n, p)$ - standard Gaussian $\mathcal{N}(0, I)$, with $n = 10^4$, $p = 10^2$.
- $x^\sharp$ is 50 sparse with zero mean Gaussian i.i.d. entries, normalized to $\|x^\sharp\|_2 = 1$.
- $b := Ax^\sharp + w$, where $w$ is Gaussian white noise with variance 1.

- 1 epoch = 1 pass over the full gradient
Convergence of SGD without strong convexity

Theorem (decaying step-size [14])

Assume
- \( \mathbb{E}[\|x^k - x^*\|^2] \leq D^2 \) for all \( k \),
- \( \mathbb{E}[\|G(x^k, \theta_k)\|^2] \leq M^2 \), (bounded gradient)
- \( \gamma_k = \frac{\gamma_0}{\sqrt{k}} \)

Then
\[
\mathbb{E}[f(x^k) - f(x^*)] \leq \left( \frac{D^2}{\gamma_0} + \gamma_0 M^2 \right) \frac{2 + \log k}{\sqrt{k}}.
\]

- \( O(1/\sqrt{k}) \) rate is optimal for SG if we do not consider the strong convexity.
Theorem (strongly convex objective, fixed step-size [2])

Assume

- $f$ is $\mu$-strongly convex and $L$-smooth,
- $\mathbb{E}[\|G(x^k, \theta_k)\|^2_2] \leq \sigma^2 + M\|\nabla f(x^k)\|^2_2$ (Bounded variance),
- $\gamma_k = \gamma \leq \frac{1}{LM}$.

Then

$$\mathbb{E}[f(x^k) - f(x^*)] \leq \frac{\gamma L\sigma^2}{2\mu} + (1 - \mu\gamma)^{k-1} \left(f(x^1) - f^*\right).$$

- Converge fast (linearly) to a neighborhood around $x^*$
- Zero variance ($\sigma = 0$) $\implies$ linear convergence
- Smaller step-sizes $\gamma \implies$ converge to a better point, but with a slower rate
Theorem (strongly convex objective, decaying step-size [2])

Assume

- $f$ is $\mu$-strongly convex and $L$-smooth,
- $\mathbb{E}[\|G(x^k, \theta_k)\|^2] \leq \sigma^2 + M\|\nabla f(x^k)\|^2$ (bounded variance),
- $\gamma_k = \frac{c}{k_0+k}$ with some appropriate constants $c$ and $k_0$.

Then

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \frac{C}{k+1},$$

where $C$ is a constant independent of $k$.

- Using the smooth property,

$$\mathbb{E}[f(x^k) - f(x^*)] \leq L\mathbb{E}[\|x^k - x^*\|^2] \leq \frac{C}{k+1}.$$

- The rate is optimal if $\sigma^2 > 0$ with the assumption of strongly-convexity.
Randomized Kaczmarz algorithm

Problem

Given a full-column-rank matrix $A \in \mathbb{R}^{n \times p}$ and $b \in \mathbb{R}^n$, solve the linear system

$$Ax = b.$$ 

Notations: $b := (b_1, \ldots, b_n)^T$ and $a_j^T$ is the $j$-th row of $A$.

Randomized Kaczmarz algorithm (RKA)

1. Choose $x^0 \in \mathbb{R}^p$.
2. For $k = 0, 1, \ldots$ perform:
   2a. Pick $j_k \in \{1, \ldots, n\}$ randomly with $\Pr(j_k = i) = \|a_i\|_2^2/\|A\|_F^2$.
   2b. $x^{k+1} = x^k - \left(\langle a_{j_k}, x^k \rangle - b_{j_k}\right) a_{j_k}/\|a_{j_k}\|_2^2$.

Linear convergence [15]

Let $x^*$ be the solution of $Ax = b$ and $\kappa = \|A\|_F\|A^{-1}\|$. Then

$$\mathbb{E}\|x^k - x^*\|_2^2 \leq \left(1 - \kappa^{-2}\right)^k \|x^0 - x^*\|_2^2$$

- RKA can be seen as a particular case of SGD [10].
Example: SGD with different step sizes

\[ \gamma_k = \frac{\gamma_0}{k + k_0}. \]

Setup

- Synthetic least-squares problem as before
- \[ \gamma_k = \gamma_0/(k + k_0). \]
Example: SGD with different step sizes

Setup

- Synthetic least-squares problem as before
- $\gamma_k = \gamma_0 / (k + k_0)$.

$\gamma_0 = 1/\mu$ is the best choice.
Comparison with GD

\[
f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) := \frac{1}{n} \sum_{j=1}^{n} f_j(x) \right\}.
\]

• \( f \): \( \mu \)-strongly convex with \( L \)-Lipschitz smooth.

<table>
<thead>
<tr>
<th></th>
<th>rate</th>
<th>iteration complexity</th>
<th>cost per iteration</th>
<th>total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>GD</td>
<td>( \rho^k )</td>
<td>( \log(1/\epsilon) )</td>
<td>( n )</td>
<td>( n \log(1/\epsilon) )</td>
</tr>
<tr>
<td>SGD</td>
<td>( 1/k )</td>
<td>( 1/\epsilon )</td>
<td>( 1 )</td>
<td>( 1/\epsilon )</td>
</tr>
</tbody>
</table>

• SGD is more favorable when \( n \) is large — large-scale optimization problems
Motivation for SGD with Averaging

- SGD iterates tend to oscillate around global minimizers
- Averaging iterates can reduce the oscillation effect
- Two types of averaging:
  \[
  \bar{x}^k = \frac{1}{k} \sum_{j=1}^{k} \gamma_j x^j \quad \text{(vanilla averaging)}
  \]
  \[
  \bar{x}^k = \frac{\sum_{j=1}^{k} \gamma_j x^j}{\sum_{j=1}^{k} \gamma_j} \quad \text{(weighted averaging)}
  \]
Convergence for SG-A I: strongly convex case

<table>
<thead>
<tr>
<th>Stochastic gradient method with averaging (SG-A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $x^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k \in \mathbb{N}} \in ]0, +\infty[^{\mathbb{N}}$.</td>
</tr>
<tr>
<td>2a. For $k = 0, 1, \ldots$ perform: $x^{k+1} = x^k - \gamma_k G(x^k, \theta_k)$.</td>
</tr>
<tr>
<td>2b. $\bar{x}^k = \frac{1}{k} \sum_{j=1}^{k} x^j$.</td>
</tr>
</tbody>
</table>

Theorem (Convergence of SG-A [13])

Assume

- $f$ is $\mu$-strongly convex,
- $\mathbb{E}[\|G(x^k, \theta_k)\|^2] \leq M^2$,
- $\gamma_k = \gamma_0 / k$ for some $\gamma_0 \geq 1 / \mu$.

Then

$$\mathbb{E}[f(\bar{x}^k) - f(x^*)] \leq \frac{\gamma_0 M^2 (1 + \log k)}{2k}.$$ 

- Same convergence rate with vanilla SGD.
**Convergence for SG-A II: non-strongly convex case**

<table>
<thead>
<tr>
<th>Stochastic gradient method with averaging (SG-A)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Choose $x^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k \in \mathbb{N}} \in ]0, +\infty[\mathbb{N}$.</td>
</tr>
<tr>
<td><strong>2a.</strong> For $k = 0, 1, \ldots$ perform:</td>
</tr>
<tr>
<td>$x^{k+1} = x^k - \gamma_k G(x^k, \theta_k)$.</td>
</tr>
<tr>
<td><strong>2b.</strong> $\bar{x}^k = (\sum_{j=0}^{k} \gamma_j)^{-1} \sum_{j=0}^{k} \gamma_j x^j$.</td>
</tr>
</tbody>
</table>

**Theorem (Convergence of SG-A [11])**

Let $D = \|x^0 - x^*\|$ and $\mathbb{E}[\|G(x^k, \theta_k)\|^2] \leq M^2$.

Then,

$$\mathbb{E}[f(\bar{x}^{k+1}) - f(x^*)] \leq \frac{D^2 + M^2 \sum_{j=0}^{k} \gamma_j^2}{2 \sum_{j=0}^{k} \gamma_j}.$$  

In addition, choosing $\gamma_k = \frac{D}{(M \sqrt{k} + 1)}$, we get,

$$\mathbb{E}[f(\bar{x}^k) - f(x^*)] \leq \frac{MD(2 + \log k)}{\sqrt{k}}.$$  

- Same convergence rate with vanilla SGD.
Example: SG-A method with different step sizes

\[
\min_x \left\{ f(x) := \frac{1}{2n} \|Ax - b\|_2^2 : x \in \mathbb{R}^p \right\}
\]

Setup

- Synthetic least-squares problem as before
- \( \gamma_k = \gamma_0 / (k + k_0) \).
Example: SG-A method with different step sizes

$$\min_x \left\{ f(x) := \frac{1}{2n} \|Ax - b\|_2^2 : x \in \mathbb{R}^p \right\}$$

**Setup**

- Synthetic least-squares problem as before
- $$\gamma_k = \gamma_0/(k + k_0)$$.

*SG-A is more stable than SG.*

$$\gamma_0 = 2/\mu$$ is the best choice.
Least mean squares algorithm

Least-square regression problem

Solve

\[ x^* \in \arg \min_{x \in \mathbb{R}^p} \left\{ f(x) := \frac{1}{2} \mathbb{E}(a, b) \left( \langle a, x \rangle - b \right)^2 \right\}, \]

given i.i.d. samples \( \{(a_j, b_j)\}_{j=1}^n \) (particularly in a streaming way).

Stochastic gradient method with averaging

1. Choose \( x^0 \in \mathbb{R}^p \) and \( \gamma > 0 \).
2a. For \( k = 1, \ldots, n \) perform:

\[ x^k = x^{k-1} - \gamma \left( \langle a_k, x^{k-1} \rangle - b_k \right) a_k. \]

2b. \( \bar{x}^k = \frac{1}{k+1} \sum_{j=0}^{k} x^j. \)

\( O(1/n) \) convergence rate, without strongly convexity \([1]\)

Let \( \|a_j\|_2 \leq R \) and \( |\langle a_j, x^* \rangle - b_j| \leq \sigma \) a.s.. Pick \( \gamma = 1/(4R^2) \). Then

\[ \mathbb{E} f(\bar{x}^{n-1}) - f^* \leq \frac{2}{n} \left( \sigma \sqrt{p} + R \|x^0 - x^*\|_2 \right)^2. \]
Popular SGD Variants

- Mini-batch SGD: For each iteration,

\[ x^{k+1} = x^k - \gamma_k \frac{1}{b} \sum_{\theta \in \Gamma} G(x^k, \theta). \]

- \( \gamma_k \): step-size
- \( b \): mini-batch size
- \( \Gamma \): a set of random variables \( \theta \) of size \( b \)

- Accelerated SGD (Nesterov accelerated technique)

- SGD with Momentum

- Adaptive stochastic methods: AdaGrad...
Adaptive methods for stochastic optimization

Remark

- Adaptive methods have extensive applications in stochastic optimization.
- We will see another nature of adaptive methods in this lecture.
- Mild additional assumption: bounded variance of gradient estimates.
AdaGrad for stochastic optimization

- Only modification: \( \nabla f(x) \Rightarrow G(x, \theta) \)

<table>
<thead>
<tr>
<th>AdaGrad with ( H_k = \lambda_k I ) [8]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set ( Q_0 = 0 ).</td>
</tr>
<tr>
<td>2. For ( k = 0, 1, \ldots, T ), iterate</td>
</tr>
</tbody>
</table>
| \[
\begin{align*}
    Q^k & = Q^{k-1} + \|G(x^k, \theta)\|^2 \\
    H_k & = \sqrt{Q_t I} \\
    x_{k+1} & = x_t - \alpha_k H_k^{-1} G(x^k, \theta)
\end{align*}
|  

Theorem (Convergence rate: stochastic, convex optimization [8])

Assume \( f \) is convex and \( L \)-smooth, such that minimizer of \( f \) lies in a convex, compact set \( \mathcal{K} \) with diameter \( D \). Also consider bounded variance for unbiased gradient estimates, i.e., \( \mathbb{E} \left[ \|G(x, \theta) - \nabla f(x)\|^2 \right] \leq \sigma^2 \). Then,

\[
\mathbb{E}[f(x)] - \min_{x \in \mathbb{R}^d} f(x) = O \left( \frac{\sigma D}{\sqrt{T}} \right)
\]

- AdaGrad is *adaptive* also in the sense that it adapt to nature of the oracle.
AcceleGrad for stochastic optimization

- Similar to AdaGrad, replace $\nabla f(x) \Rightarrow G(x, \theta)$

<table>
<thead>
<tr>
<th>AcceleGrad (Accelerated Adaptive Gradient Method)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong>: Number of iterations $T$, $x_0 \in \mathcal{K}$, diameter $D$, weights ${\alpha_t}<em>{t \in [T]}$, learning rate ${\eta_t}</em>{t \in [T]}$</td>
</tr>
<tr>
<td><strong>1.</strong> Set $y_0 = z_0 = x_0$</td>
</tr>
<tr>
<td><strong>2.</strong> For $k = 0, 1, \ldots, T$, iterate</td>
</tr>
<tr>
<td>$\tau_t := 1/\alpha_t$</td>
</tr>
<tr>
<td>$x_{t+1} = \tau_t z_t + (1 - \tau_t)y_t$, define $g_t := \nabla f(x_{t+1})$</td>
</tr>
<tr>
<td>$z_{t+1} = \Pi_{\mathcal{K}}(z_t - \alpha_t \eta_t g_t)$</td>
</tr>
<tr>
<td>$y_{t+1} = x_{t+1} - \eta_t g_t$</td>
</tr>
<tr>
<td><strong>Output</strong>: $\overline{y}<em>T \propto \sum</em>{t=0}^{T-1} \alpha_t y_{t+1}$</td>
</tr>
</tbody>
</table>

**Theorem (Convergence rate [9])**

Assume $f$ is convex and $G$-Lipschitz and that minimizer of $f$ lies in a convex, compact set $\mathcal{K}$ with diameter $D$. Also consider bounded variance for unbiased gradient estimates, i.e., $\mathbb{E} \left[ ||G(x, \theta) - \nabla f(x)||^2 | x \right] \leq \sigma^2$. Then,

$$\mathbb{E}[f(\overline{y}_T)] - \min_x f(x) = O \left( \frac{GD \sqrt{\log T}}{\sqrt{T}} \right).$$
Example: Synthetic least squares

• $A \in \mathbb{R}^{n \times d}$, where $n = 200$ and $d = 50$.
• Number of epochs: 20.
• Algorithms: SGD, AdaGrad & AcceleGrad.
Convex optimization with finite sums

Problem (Convex optimization with finite sums)

We consider the following simple example in the next few slides:

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) := \frac{1}{n} \sum_{j=1}^{n} f_j(x) \right\} \]

- \( f_j \) is proper, closed, and convex.
- \( \nabla f_j \) is \( L_j \)-Lipschitz continuous for \( j = 1, \ldots, n \).
- The solution set \( S^* := \{ x^* \in \text{dom}(f) : f(x^*) = f^* \} \) is nonempty.

- One prevalent choice is given by

\[ G(x^k, i_k) = \nabla f_{i_k}(x^k), \quad i_k \text{ is uniformly distributed over } \{1, 2, \cdots, n\} \]
An observation of SGD step

\[ x^{k+1} = x^k - \gamma_k \nabla f(x^k) \] (GD)

Lemma

Assume \( f \) is Lipschitz smooth with constant \( L \). Then,

\[ f(x^{k+1}) - f(x^k) \leq (\gamma_k^2 L - \gamma_k) \| \nabla f(x^k) \|^2. \]
An observation of SGD step

\[ x^{k+1} = x^k - \gamma_k G(x^k, i_k) \]  (SGD)

**Lemma**

Assume \( f \) is Lipschitz smooth with constant \( L \). Then,

\[
\mathbb{E}[f(x^{k+1}) - f(x^k)] \leq (\gamma_k^2 L - \gamma_k) \mathbb{E}[\|\nabla f(x^k)\|^2] + L \gamma_k^2 \mathbb{E}[\|G(x^k, i_k) - \nabla f(x^k)\|^2]
\]
An observation of SGD step

\[ x^{k+1} = x^k - \gamma_k G(x^k,i_k) \] (SGD)

**Lemma**

Assume \( f \) is Lipschitz smooth with constant \( L \). Then,

\[
\mathbb{E}[f(x^{k+1}) - f(x^k)] \leq (\gamma_k^2 L - \gamma_k) \mathbb{E}[\|\nabla f(x^k)\|^2] + L\gamma_k^2 \mathbb{E}[\|G(x^k,i_k) - \nabla f(x^k)\|^2]
\]

- The variance in gradient dominates later (as if \( \nabla f(x^k) \to 0 \)).
- To ensure convergence, \( \gamma_k \to 0 \). \( \implies \) Slow convergence!

*Can we decrease the variance while using a constant step-size?*

- Choose a stochastic gradient, s.t. \( \mathbb{E}[\|G(x^k,i_k)\|^2] \to 0 \).
Variance reduction techniques: SVRG

- Select the stochastic gradient $\nabla f_{i_k}$, and compute a gradient estimate

$$r_k = \nabla f_{i_k}(x^k) - \nabla f_{i_k} (\tilde{x}) + \nabla f(\tilde{x}),$$

where $\tilde{x}$ is a good approximation of $x^\star$.

- As $\tilde{x} \to x^\star$ and $x^k \to x^\star$,

$$\nabla f_{i_k}(x^k) - \nabla f_{i_k} (\tilde{x}) + \nabla f(\tilde{x}) \to 0.$$

- Therefore,

$$\mathbb{E}\left[\|\nabla f_{i_k}(x^k) - \nabla f_{i_k} (\tilde{x}) + \nabla f(\tilde{x})\|^2\right] \to 0.$$
## Stochastic gradient algorithm with variance reduction (SVRG) [7, 18]

1. Choose $\tilde{x}^0 \in \mathbb{R}^p$ as a starting point and $\gamma > 0$ and $q \in \mathbb{N}_+$.  
2. For $s = 0, 1, 2 \cdots$, perform:
   
   2a. $\tilde{x} = \tilde{x}^s$, $\tilde{v} = \nabla f(\tilde{x})$, $x^0 = \tilde{x}$.
   
   2b. For $k = 0, 1, \cdots q - 1$, perform:
   
   \[
   \begin{align}
   \text{Pick } i_k \in \{1, \ldots, n\} \text{ uniformly at random} \\
   r_k &= \nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}) + \tilde{v} \\
   x^{k+1} &= x^k - \gamma r_k,
   \end{align}
   \]
   
   2c. Update $\tilde{x}^{s+1} = \frac{1}{m} \sum_{j=0}^{q-1} x^j$.

### Common features

- The SVRG method uses a multistage scheme to reduce the variance of the stochastic gradient $r_k$ where $x^k$ and $\tilde{x}^s$ tend to $x_*$.
- Learning rate $\gamma$ does not necessarily tend to 0.
- Each stage, SVRG uses $n + 2q$ component gradient evaluations: $n$ for the full gradient at the beginning of each stage, and $2q$ for each of the $q$ stochastic gradient steps.
Convergence analysis

Assumption A5.

(i) \( f \) is \( \mu \)-strongly convex
(ii) The learning rate \( 0 < \gamma < 1/(4L_{\text{max}}) \), where \( L_{\text{max}} = \max_{1 \leq j \leq n} L_j \).
(iii) \( q \) is large enough such that

\[
\kappa = \frac{1}{\mu \gamma (1 - 4\gamma L_{\text{max}})q} + \frac{4\gamma L_{\text{max}}(q + 1)}{(1 - 4\gamma L_{\text{max}})q} < 1.
\]

Theorem

Assumptions:
- The sequence \( \{\tilde{x}^s\}_{k \geq 0} \) is generated by SVRG.
- Assumption A5 is satisfied.

Conclusion: Linear convergence is obtained:

\[
\mathbb{E} f(\tilde{x}^s) - f(x^*) \leq \kappa^s (f(\tilde{x}^0) - f(x^*)�).
\]
Choice of $\gamma$ and $q$, and complexity

Chose $\gamma$ and $q$ such that $\kappa \in (0, 1)$:

For example

$$\gamma = 0.1/L_{\text{max}}, \quad q = 100(L_{\text{max}}/\mu) \implies \kappa \approx 5/6.$$  

Complexity

$$\mathbb{E} f(\tilde{x}^s) - f(x^*) \leq \varepsilon, \quad \text{when } s \geq \log((f(\tilde{x}^0) - f(x^*)) / \varepsilon) / \log(\kappa^{-1})$$  

Since at each stage needs $n + 2q$ component gradient evaluations, with $q = \mathcal{O}(L_{\text{max}}/\mu)$, we get the overall complexity is

$$\mathcal{O}\left((n + L_{\text{max}}/\mu) \log(1/\varepsilon)\right).$$
Variance reduction techniques: SAGA

<table>
<thead>
<tr>
<th>Stochastic Average Gradient (SAGA) [4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a. Choose ( \tilde{x}<em>i^0 = x^0 \in \mathbb{R}^p ), ( \forall i, q \in \mathbb{N}</em>+ ) and stepsize ( \gamma &gt; 0 ).</td>
</tr>
<tr>
<td>1b. Store ( \nabla f_i(\tilde{x}_i^0) ) in a table data-structure with length ( n ).</td>
</tr>
<tr>
<td>2. For ( k = 0, 1 \ldots ) perform:</td>
</tr>
<tr>
<td>2a. pick ( i_k \in {1, \ldots, n} ) uniformly at random</td>
</tr>
<tr>
<td>2b. Take ( \tilde{x}<em>{i_k}^{k+1} = x^k ), store ( \nabla f</em>{i_k}(\tilde{x}_{i_k}^{k+1}) ) in the table and leave other entries the same.</td>
</tr>
<tr>
<td>2c. ( r_k = \nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}<em>{i_k}^k) + \frac{1}{n} \sum</em>{j=1}^{n} \nabla f_j(\tilde{x}_j^k) )</td>
</tr>
<tr>
<td>3. ( x^{k+1} = x^k - \gamma r_k )</td>
</tr>
</tbody>
</table>

Recipe:

In each iteration:
- Store last gradient evaluated at each datapoint.
- Previous gradient for datapoint \( j \) is \( \nabla f_j(\tilde{x}_j^k) \).
- Perform SG-iterations with the following stochastic gradient

\[ r_k = \nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}_{i_k}^k) + \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(\tilde{x}_j^k). \]
Variance reduction techniques: SAGA

- Select the stochastic gradient $r_k$ as

$$r_k = \nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}_{i_k}^k) + \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(\tilde{x}_j^k),$$

where, at each iteration, $\tilde{x}$ is updated as $\tilde{x}_{i_k}^k = x^k$ and $\tilde{x}_j^k$ stays the same for $j \neq i_k$.

- As $\tilde{x}_j^k \to x^*$ and $x^k \to x^*$,

$$\nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}_{i_k}^k) + \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(\tilde{x}_j^k) \to 0.$$  

- Therefore,

$$\mathbb{E}\left[\|\nabla f_{i_k}(x^k) - \nabla f_{i_k}(\tilde{x}_{i_k}^k) + \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(\tilde{x}_j^k)\|^2\right] \to 0.$$
Convergence of SAGA

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) := \frac{1}{n} \sum_{j=1}^{n} f_j(x) \right\}. \]

Theorem (Convergence of SAGA [4])

Suppose that \( f \) is \( \mu \)-strongly convex and that the stepsize is \( \gamma = \frac{1}{2(\mu n + L)} \) with

\[ \rho = 1 - \frac{\mu}{2(\mu n + L)} < 1, \]

\[ C = \|x^0 - x^*\|^2 + \frac{n}{\mu n + L} \left[ f(x^0) - \langle \nabla f(x^*), x^0 - x^* \rangle - f(x^*) \right] \]

Then

\[ \mathbb{E}[\|x^k - x^*\|^2] \leq \rho^k C. \]

- Allows the constant step-size.
- Obtains linear rate convergence.
SVRG vs SAGA

- SVRG update:

\[
\begin{align*}
    \mathbf{r}_k &= \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}) \\
    \mathbf{x}^{k+1} &= \mathbf{x}^k - \gamma \mathbf{r}_k,
\end{align*}
\]

- SAGA update:

\[
\begin{align*}
    \mathbf{r}_k &= \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}_{i_k}^k) + \frac{1}{n} \sum_{j=1}^{n} \nabla f_j(\tilde{\mathbf{x}}_j^k) \\
    \mathbf{x}^{k+1} &= \mathbf{x}^k - \gamma \mathbf{r}_k,
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>SVRG</th>
<th>SAGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage of gradients</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Epoch-base</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Parameters</td>
<td>stepsize &amp; epoch lengths</td>
<td>stepsize</td>
</tr>
<tr>
<td>Gradient evaluations per step</td>
<td>at least 2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table: Comparisons of SVRG and SAGA [4]
Taxonomy of algorithms

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) := \frac{1}{n} \sum_{j=1}^{n} f_j(x) \right\}. \]

- \( f(x) = \frac{1}{n} \sum_{j=1}^{n} f_j(x) \): \( \mu \)-strongly convex with \( L \)-Lipschitz continuous gradient.

<table>
<thead>
<tr>
<th>Gradient descent</th>
<th>SVRG/SAGA</th>
<th>SGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>Linear</td>
<td>Sublinear</td>
</tr>
</tbody>
</table>

**Table:** Rate of convergence.

- \( \kappa = L/\mu \) and \( s_0 = 8 \sqrt{\kappa} n (\sqrt{2} \alpha (n - 1) + 8 \sqrt{\kappa})^{-1} \) for \( 0 < \alpha \leq 1/8 \).

<table>
<thead>
<tr>
<th>SVRG/SAGA</th>
<th>AccGrad</th>
<th>SGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O((n + \kappa) \log(1/\varepsilon)) )</td>
<td>( O((n \kappa) \log(1/\varepsilon)) )</td>
<td>( 1/\varepsilon )</td>
</tr>
</tbody>
</table>

**Table:** Complexity to obtain \( \varepsilon \)-solution.
## Stochastic methods for non-convex problems

### Remark (Convex optimization)

- Large scale convex optimization $\Rightarrow$ demands stochastic methods.
- SGD, AdaGrad & AcceleGrad are optimal for general convex functions.
- Adaptive methods can also adapt to **the stochasticity of the gradient oracle**.

### Remark (Non-convex optimization)

- Large scale non-convex optimization $\Rightarrow$ demands stochastic methods.
- AdaGrad, ADAM, RMSProp are frequently used in neural network optimization (more on next lecture!)
SGD - Non-convex stochastic optimization

- SGD is not as well-studied for non-convex problems as for convex problems.
- There is a gap between SGD’s practical performance and theoretical understanding.
- Recall SGD update rule:

\[ x^{k+1} = x^k - \alpha_k G(x^k, \theta) \]

**Theorem (A well-known result for SGD & Non-convex problems [5])**

Let \( f \) be a non-convex and \( L \)-smooth function. Set \( \alpha_k = \min \left\{ \frac{1}{L}, \frac{C}{\sigma \sqrt{T}} \right\} \), \( \forall k = 1, \ldots, T \), where \( \sigma^2 \) is the variance of the gradients and \( C > 0 \) is constant. Then,

\[ \mathbb{E}[\|\nabla f(x^R)\|^2] = O \left( \frac{\sigma}{\sqrt{T}} \right), \]

where \( \mathbb{P}(R = k) = \frac{2\alpha_k - L\alpha_k^2}{\sum_{k=1}^T (2\alpha_k - L\alpha_k^2)} \).
Non-convergence of ADAM and a new method: AmsGrad

- It has been shown that ADAM may not converge for some objective functions [12].
- An ADAM alternative is proposed that is proved to be convergent [12].

### AmsGrad

**Input.** Step size \( \{\alpha_k\}_{k=1}^T \), exponential decay rates \( \{\beta_{1k}\}_{k=1}^T \),

\[\beta_2\]

1. Set \( m_0 = 0, v_0 = 0 \) and \( \hat{v}_0 = 0 \)
2. For \( k = 1, 2, \ldots, T \), iterate

\[
\begin{align*}
&g_k = G(x^k, \theta) \\
&m_k = \beta_{1k} m_{k-1} + (1 - \beta_{1k}) g_k \quad \leftarrow \text{1st order estimate} \\
&v_k = \beta_2 v_{k-1} + (1 - \beta_2) g_k^2 \quad \leftarrow \text{2nd order estimate} \\
&\hat{v}_k = \max\{\hat{v}_{k-1}, v_k\} \quad \text{and} \quad \hat{V}_k = \text{diag}(\hat{v}_k) \\
&H_k = \sqrt{\hat{v}_k} \\
&x^{k+1} = \Pi_{\mathcal{X}} \left( x^k - \alpha_k m_k / H_k \right)
\end{align*}
\]

where \( \Pi^A_\mathcal{X}(y) = \arg\min_{x \in \mathcal{X}} \langle (x - y), A(x - y) \rangle \) (weighted projection onto \( \mathcal{X} \)).
AdaGrad & AmsGrad for non-convex optimization

**Theorem (AdaGrad convergence rate: stochastic, non-convex [17])**

Assume $f$ is non-convex and $L$-smooth, such that $\|\nabla f(x)\|^2 \leq G^2$ and $f^* = \inf_x f(x) > \infty$. Also consider bounded variance for unbiased gradient estimates, i.e., $\mathbb{E} \left[ \|G(x, \theta) - \nabla f(x)\|^2 | x \right] \leq \sigma^2$. Then with probability $1 - \delta$,

$$\min_{k \in \{1, \ldots, T-1\}} \|\nabla f(x^k)\|^2 = \tilde{O} \left( \frac{\sigma}{\delta^{3/2} \sqrt{T}} \right)$$

**Note:** As $1 - \delta \to 1$, the rate deteriorates by a factor of $\delta^{-3/2}$.

**Theorem (AmsGrad convergence rate 1: stochastic, non-convex [3])**

Let $g_k = G(x^k, \theta)$. Assume $|g_{1,i}| > c > 0$, $\forall i \in [d]$ and $\|g_k\| \leq G$. Consider a non-increasing sequence $\beta_{1k}$ and $\beta_{1k} \leq \beta_1 \in [0, 1)$. Set $\alpha_k = 1/\sqrt{t}$. Then,

$$\min_{t \in [T]} \mathbb{E} \left[ \|\nabla f(x^k)\|^2 \right] = O \left( \frac{\log T}{\sqrt{T}} \right).$$
AdaGrad & AmsGrad for non-convex optimization

Theorem (AdaGrad convergence rate: stochastic, non-convex [17])

Assume $f$ is non-convex and $L$-smooth, such that $\|\nabla f(x)\|_2^2 \leq G^2$ and $f^* = \inf_x f(x) > \infty$. Also consider bounded variance for unbiased gradient estimates, i.e., $\mathbb{E} \left[ \|G(x, \theta) - \nabla f(x)\|_2^2 | x \right] \leq \sigma^2$. Then with probability $1 - \delta$,

$$\min_{k \in \{1, \ldots, T-1\}} \|\nabla f(x^k)\|_2^2 = \tilde{O} \left( \frac{\sigma}{\delta^{3/2} \sqrt{T}} \right)$$

- **Note:** As $1 - \delta \to 1$, the rate deteriorates by a factor of $\delta^{-3/2}$.

Theorem (AmsGrad convergence rate 2: stochastic, non-convex [19])

Consider $f : \mathbb{R}^d \to \mathbb{R}$ to be non-convex ans $L$-smooth. Assume $\|G(x, \theta)\|_\infty \leq G_\infty$ and set $\alpha_k = 1 / \sqrt{dT}$. Also define $x_{out} = x^k$, for $k = 1, \ldots, T$ with probability $\alpha_k / \sum_{i=1}^T \alpha_i$. Then,

$$\mathbb{E} \left[ \|\nabla f(x_{out})\|^2 \right] = O \left( \sqrt{\frac{d}{T}} \right).$$
Example: Logistic regression with non-convex regularizer

- Synthetic data: $A \in \mathbb{R}^{n \times d}$, $n = 2000$, $d = 200$.
- Batch size: 20 samples.
- Algorithms: SGD, AdaGrad, AmsGrad.
References


References II

Stochastic first-and zeroth-order methods for nonconvex stochastic programming.

Accelerated gradient methods for nonconvex nonlinear and stochastic
programming.

Accelerating stochastic gradient descent using predictive variance reduction.

Online to offline conversions, universality and adaptive minibatch sizes.

Online adaptive methods, universality and acceleration.
References III

    Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm.
    In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors,

    Robust stochastic approximation approach to stochastic programming.

    On the convergence of adam and beyond.

[13] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter.
    Pegasos: primal estimated sub-gradient solver for svm.

    Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes.
A randomized kaczmarz algorithm with exponential convergence.

An overview of statistical learning theory.

AdaGrad stepsizes: Sharp convergence over nonconvex landscapes.

[18] Lin Xiao and Tong Zhang.
A proximal stochastic gradient method with progressive variance reduction.
*SIAM Journal on Optimization*, 24, 03 2014.

On the convergence of adaptive gradient methods for nonconvex optimization.