Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 7: Stochastic gradient methods

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

EE-556 (Fall 2018)

License Information for Mathematics of Data Slides

Outline

- This class
  1. Stochastic programming
  2. Stochastic gradient descent
  3. Variance reduction technique
     - SVRG
     - SAGA
- Next class
  1. Composite convex minimization

Recommended reading materials

1. V. Cevher; S. Becker, and M. Schmidt. Convex optimization for big data. *IEEE Signal Process. Mag.*, vol. 31, pp. 32–43, 2014.
2. A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming.
3. L. Bottou., F. E. Curtis and J. Nocedal. Optimization methods for large-scale machine learning. *arXiv:1606.04838,* 2016 Jun 15.

Recall: Gradient descent

Problem (Unconstrained convex problem)

Consider the following convex minimization problem:

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

- $f(\mathbf{x})$ *is proper, closed, and convex (perhaps strongly-convex and/or smooth).*

Gradient descent

Choose a starting point $\mathbf{x}^0$ and iterate

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

where γ_k is a step-size to be chosen so that $\mathbf{x}^k$ converges to $\mathbf{x}^\star$.

GD (accelerated GD) has fast (optimal) convergence rate when $f \in \mathcal{F}_L$.
Why should we study anything else?

Statistical learning

A basic statistical learning model [1]

A statistical learning model consists of the following three elements.

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

Statistical learning

A basic statistical learning model [1]

A statistical learning model consists of the following three elements.

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

Definition (Risk)

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

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

Statistical learning seeks to find a $f^\star \in \mathcal{F}$ that minimizes the risk, i.e., it solves

$$f^\star \in \arg\min_{f \in \mathcal{F}} R(f).$$

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 $f \in \mathcal{F}$,

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

when n is large enough, with high probability.

Statistical learning with Empirical risk minimization (ERM) [1]

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

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

Example: Least squares

Recall that the LS estimator is given by

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

where we define $\mathbf{b} := (b_1, \ldots, b_n)^T$ and $\mathbf{a}_j^T$ to be the j-th row of $\mathbf{A}$.

Empirical risk minimization (ERM) II

Example: Logistic regression

Recall the logistic regression formulation

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

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

Gradient descent for ERM

$$f^{k+1} = f^k - \gamma_k\nabla R_n(f) = f^k - \gamma_k\frac{1}{n}\sum_{j=1}^{n}\nabla L(f(\mathbf{a}_j),b_j).$$

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 $f^\star \in \mathcal{F}$ that minimizes the *expected* risk,

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

In practice, data can arrive in a *streaming* way.

Example: Markowitz portfolio optimization

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

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

Gradient method

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

This can not be implemented in practice as the distribution of $(\mathbf{a}, b)$ is unknown.

Stochastic programming

Problem (**Mathematical formulation**)

Consider the following convex minimization problem:

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

- θ *is a random vector whose probability distribution is supported on set* Θ.
- $f(\mathbf{x}) := \mathbb{E}[h(\mathbf{x},\theta)]$ *is proper, closed, and convex.*
- *The solution set* $\mathcal{S}^\star := \{\mathbf{x}^\star \in \mathrm{dom}(f) : f(\mathbf{x}^\star) = f^\star\}$ *is nonempty.*

Stochastic gradient descent (SGD)

Stochastic gradient descent (SGD)

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

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

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

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

Stochastic gradient descent (SGD)

Stochastic gradient descent (SGD)

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

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

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

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

Remark

- The cost of computing $G(\mathbf{x}^k, \theta_k)$ is n times cheaper than that of $\nabla f(\mathbf{x}^k)$.
- As $G(\mathbf{x}^k, \theta_k)$ 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

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

Stochastic gradient descent (SGD)

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k \nabla f_i(\mathbf{x}^k) \qquad i \text{ is uniformly distributed over} \{1, ..., n\}$$

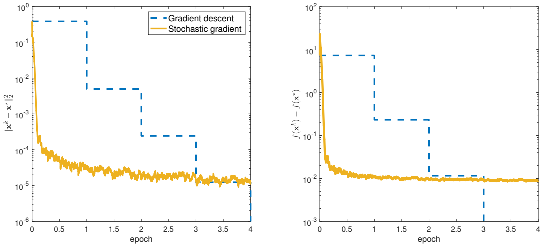
- Note: $\mathbb{E}_i[\nabla f_i(\mathbf{x}^k)] = \sum_{j=1}^n \nabla f_j(\mathbf{x}^k)/n = \nabla f(\mathbf{x}^k)$.
- The computational cost of SGD per iteration is p.

Synthetic least-squares problem

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

Setup

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



- 1 epoch $= 1$ pass over the full gradient

Convergence of SGD for strongly convex problems I

Theorem (strongly convex objective, fixed step-size [11])

Assume

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

Then

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

- Converge fast (linearly) to a neighborhood around $\mathbf{x}^\star$
- Zero variance ($\sigma = 0$) $\Longrightarrow$ linear convergence
- Smaller step-sizes $\gamma \Longrightarrow$ converge to a better point, but with a slower rate

Randomized Kaczmarz algorithm

Problem

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

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

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

Randomized Kaczmarz algorithm (RKA)

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

Linear convergence [15]

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

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

- RKA can be seen as a particular case of SGD [16].

Convergence of SGD for strongly convex problems II

Theorem (strongly convex objective, decaying step-size [11])

Assume

- *f is μ-strongly convex and L-smooth,*
- *$\mathbb{E}[\|G(\mathbf{x}^k, \theta_k)\|^2]_2 \leq \sigma^2 + M\|\nabla f(\mathbf{x}^k)\|_2^2$ (bounded variance),*
- *$\gamma_k = \frac{c}{k_0 + k}$ with some appropriate constants c and k_0.*

Then

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

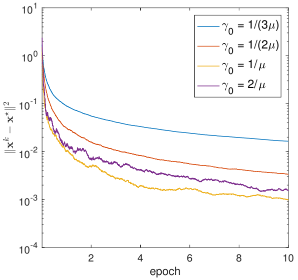
where C is a constant independent of k.

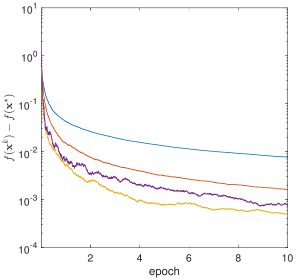
- Using the smooth property,

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

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

Example: SGD with different step sizes

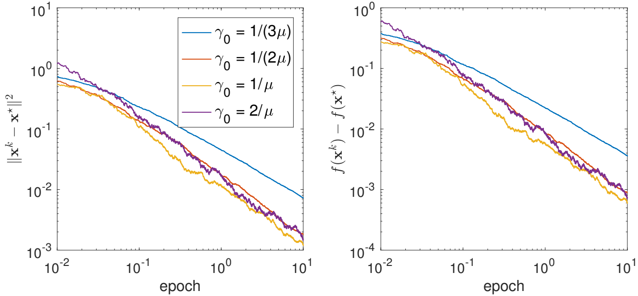




Setup

- 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

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p}\left\{f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x})\right\}.}$$

- f: μ-strongly convex with L-Lipschitz smooth.

	rate	iteration complexity	cost per iteration	total cost
GD	ρ^k	$\log(1/\epsilon)$	n	$n\log(1/\epsilon)$
SGD	$1/k$	$1/\epsilon$	1	$1/\epsilon$

- SGD is more favorable when n is large — large-scale optimization problems

Convergence of SGD without strong convexity

Theorem (decaying step-size [7])

Assume

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

Then

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

- $\mathcal{O}(1/\sqrt{k})$ rate is optimal for SG if we do not consider the strong convexity.

Motivation for SGD with Averaging

- SGD iterates tend to oscillate around global minimizers
- Averaging iterates can reduce the oscillation effect
- Two types of averaging:

$$\bar{\mathbf{x}}^k = \frac{1}{k}\sum_{j=1}^{k}\gamma_j \mathbf{x}^j \quad \text{(vanilla averaging)}$$

$$\bar{\mathbf{x}}^k = \frac{\sum_{j=1}^{k}\gamma_j \mathbf{x}^j}{\sum_{j=1}^{k}\gamma_j} \quad \text{(weighted averaing)}$$

Convergence for SG-A I: strongly convex case

Stochastic gradient method with averaging (SG-A)

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

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

2b. $\bar{\mathbf{x}}^k = \frac{1}{k}\sum_{j=1}^k \mathbf{x}^j$.

Theorem (Convergence of SG-A [8])

Assume

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

Then

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

- Same convergence rate with vanilla SGD.

Convergence for SG-A II: non-strongly convex case

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

Theorem (Convergence of SG-A [2])

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

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

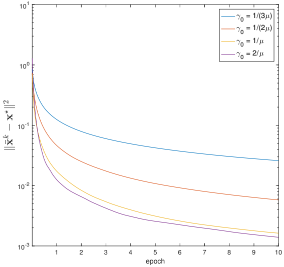
In addition, choosing $\gamma_k = D/(M\sqrt{k+1})$*, we get,*

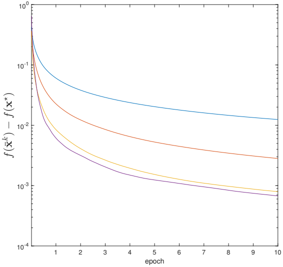
$$\mathbb{E}[f(\bar{\mathbf{x}}^k) - f(\mathbf{x}^\star)] \leq \frac{MD(2 + \log k)}{\sqrt{k}}.$$

- Same convergence rate with vanilla SGD.

Example: SG-A method with different step sizes

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



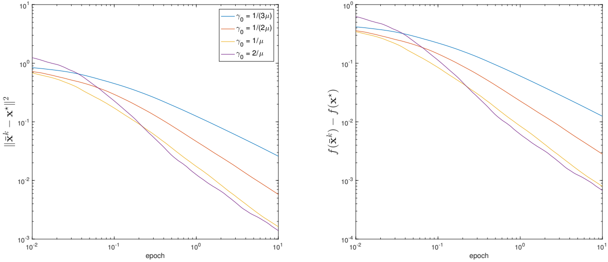


Setup

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

Example: SG-A method with different step sizes

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



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

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

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

Stochastic gradient method with averaging

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

$$\mathbf{x}^k = \mathbf{x}^{k-1} - \gamma\left(\langle \mathbf{a}_k, \mathbf{x}^{k-1}\rangle - b_k\right)\mathbf{a}_k.$$

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

$O(1/n)$ convergence rate, without strongly convexity [17]

Let $\|\mathbf{a}_j\|_2 \le R$ and $|\langle \mathbf{a}_j, \mathbf{x}^\star\rangle - b_j| \le \sigma$ a.s.. Pick $\gamma = 1/(4R^2)$. Then

$$\mathbb{E}f(\bar{\mathbf{x}}^{n-1}) - f^* \le \frac{2}{n}\left(\sigma\sqrt{p} + R\|\mathbf{x}^0 - \mathbf{x}^\star\|_2\right)^2.$$

Popular SGD Variants

- Mini-batch SGD: For each iteration,

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

  - γ_k: step-size
  - b : mini-batch size
  - Γ : a set of random variables θ of size b

- Accelerated SGD (Nesterov accelerated technique)

- SGD with Momentum

- AdaGrad, AdaDelta, AdaM ...

Adaptive stochastic gradient methods (Adagrad)

AdaGrad (diagonal form) [10]

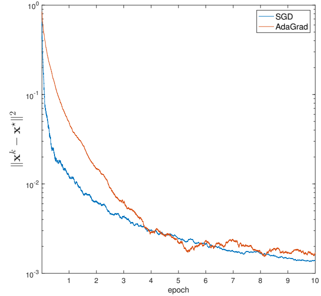
1. Choose $\mathbf{x}^0 \in \mathbb{R}^p$ and δ.
2. For $k = 0, 1, \ldots$ perform:

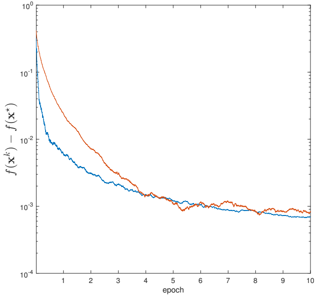
$$\begin{cases} H_k = \delta I + \text{diag}\left(\sum_{i=1}^{k} G(\mathbf{x}^i, \theta_i) G(\mathbf{x}^i, \theta_i)^T\right) \\ \mathbf{x}^{k+1} = \mathbf{x}^k - \gamma H_k^{-1/2} G(\mathbf{x}^k, \theta_k). \end{cases}$$

- The step-size for each coordinate is different.
- The algorithm is a stochastic version of the adaptive GD from Lecture 4.

Example: AdaGrad vs SG

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





Setup

- Synthesis leas-squares problem as before
- $\gamma_k = 1/(\mu(k + k_0))$ for SG.
- $\delta = 10^{-2}$ for AdaGrad.

Important remark!

All the results we have shown so far can be generalized for the non-smooth objectives, simply by replacing the gradient with a subgradient.

We will talk about the subgradient methods in the next lecture.

Convex optimization with finite sums

Problem (Convex optimization with finite sums)

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

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

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

- One prevalent choice is given by

$$G(\mathbf{x}^k, i_k) = \nabla f_{i_k}(\mathbf{x}^k), \qquad i_k \text{ is uniformly distributed over } \{1, 2, \cdots, n\}$$

An observation of SGD step

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

Lemma

Assume f is Lipschitz smooth with constant L. Then,

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k) \leq (\gamma_k^2 L - \gamma_k)\|\nabla f(\mathbf{x}^k)\|^2.$$

An observation of SGD step

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} - \gamma_k G(\mathbf{x}^{k}, i_k) \quad \text{(SGD)}$$

Lemma

Assume f is Lipschitz smooth with constant L. Then,

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

An observation of SGD step

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma_k G(\mathbf{x}^k, i_k) \quad \text{(SGD)}$$

Lemma

Assume f is Lipschitz smooth with constant L. Then,

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

- The first term dominates at the beginning and the variance in gradient will dominate later (as if $\nabla f(\mathbf{x}^k) \to 0$).
- 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. $\mathbb{E}\left[\|G(\mathbf{x}^k; i_k)\|^2\right] \to 0$.

Variance reduction techniques: SVRG

- Select the stochastic gradient ∇f_{i_k}, and compute a gradient estimate

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

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

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

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

- Therefore,

$$\mathbb{E}\left[\|\nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}})\|^2\right] \to 0.$$

Stochastic gradient algorithm with variance reduction

Stochastic gradient with variance reduction (SVRG) [9, 5]

1. Choose $\widetilde{\mathbf{x}}^0 \in \mathbb{R}^p$ as a starting point and $\gamma > 0$ and $q \in \mathbb{N}_+$.
2. For $s = 0, 1, 2 \cdots$, perform:
   - **2a**. $\widetilde{\mathbf{x}} = \widetilde{\mathbf{x}}^s, \quad \widetilde{\mathbf{v}} = \nabla f(\widetilde{\mathbf{x}}), \quad \mathbf{x}^0 = \widetilde{\mathbf{x}}$.
   - **2b**. For $k = 0, 1, \cdots q-1$, perform:

$$\begin{cases} \text{Pick } i_k \in \{1, \ldots, n\} \text{ uniformly at random} \\ \mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\widetilde{\mathbf{x}}) + \widetilde{\mathbf{v}} \\ \mathbf{x}^{k+1} := \mathbf{x}^k - \gamma \mathbf{r}_k, \end{cases} \tag{1}$$

   - **2c**. Update $\widetilde{\mathbf{x}}^{s+1} = \frac{1}{m} \sum_{j=0}^{q-1} \mathbf{x}^j$.

Common features

- The SVRG method uses a multistage scheme to reduce the variance of the stochastic gradient $\mathbf{r}_k$ where $\mathbf{x}^k$ and $\widetilde{\mathbf{x}}^s$ tend to $\mathbf{x}_\star$.
- Learning rate γ 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 μ-strongly convex

(ii) The learning rate $0 < \gamma < 1/(4L_{\max})$, where $L_{\max} = \max_{1 \le j \le n} L_j$.

(iii) q is large enough such that

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

Theorem

Assumptions:

- *The sequence $\{\widetilde{\mathbf{x}^s}\}_{k \ge 0}$ is generated by SVRG.*
- *Assumption A5 is satisfied.*

Conclusion: *Linear convergence is obtained:*

$$\mathbb{E} f(\widetilde{\mathbf{x}^s}) - f(\mathbf{x}^\star) \le \kappa^s (f(\widetilde{\mathbf{x}^0}) - f(\mathbf{x}^\star)).$$

Choice of γ and q, and complexity

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

For example

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

Complexity

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

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

$$\mathcal{O}\Big((n + L_{\max}/\mu)\log(1/\epsilon)\Big).$$

Variance reduction techniques: SAGA

Stochastic Average Gradient (SAGA) [6]
1a. Choose $\tilde{\mathbf{x}}_i^0 = \mathbf{x}^0 \in \mathbb{R}^p, \forall i,\ q \in \mathbb{N}_+$ and stepsize $\gamma > 0$.
1b. Store $\nabla f_i(\tilde{\mathbf{x}}_i^0)$ in a table data-structure with length n.
2. For $k = 0, 1 \ldots$ perform:
2a. pick $i_k \in \{1, \ldots, n\}$ uniformly at random
2b. Take $\tilde{\mathbf{x}}_{i_k}^{k+1} = \mathbf{x}^k$, store $\nabla f_{i_k}(\tilde{\mathbf{x}}_{i_k}^{k+1})$ in the table and leave other entries the same.
2c. $\mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}_{i_k}^k) + \frac{1}{n}\sum_{j=1}^n \nabla f_j(\tilde{\mathbf{x}}_j^k)$
3. $\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma \mathbf{r}_k$

Recipe:

In each iteration:

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

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

Variance reduction techniques: SAGA

- Select the stochastic gradient $\mathbf{r}_k$ as

$$\mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}^k_{i_k}) + \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\tilde{\mathbf{x}}^k_j),$$

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

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

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

- Therefore,

$$\mathbb{E}\Big[\|\nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}^k_{i_k}) + \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(\tilde{\mathbf{x}}^k_j)\|^2\Big] \to 0.$$

Convergence of SAGA

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x}) \right\}.}$$

Theorem (Convergence of SAGA [6])

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

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

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

Then

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

- Allows the constant step-size.
- Obtains linear rate convergence.

SVRG vs SAGA

- SVRG update:

$$\begin{cases} \mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\widetilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}) \\ \mathbf{x}^{k+1} := \mathbf{x}^k - \gamma \mathbf{r}_k, \end{cases}$$

- SAGA update:

$$\begin{cases} \mathbf{r}_k = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}_{i_k}^k) + \frac{1}{n}\sum_{j=1}^n \nabla f_j(\tilde{\mathbf{x}}_j^k) \\ \mathbf{x}^{k+1} := \mathbf{x}^k - \gamma \mathbf{r}_k, \end{cases}$$

	SVRG	SAGA
Storage of gradients	no	yes
Epoch-base	yes	no
Parameters	stepsize & epoch lengths	stepsize
Gradient evaluations per step	at least 2	1

Table: Comparisons of SVRG and SAGA [6]

Taxonomy of algorithms

$$\boxed{f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\sum_{j=1}^{n} f_j(\mathbf{x}) \right\}.}$$

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

Gradient descent	SVRG/SAGA	SGM
Linear	Linear	Sublinear

Table: Rate of convergence.

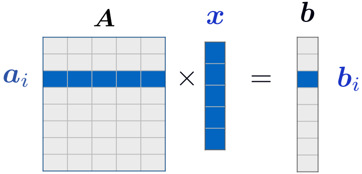
- $\kappa = L/\mu$ and $s_0 = 8\sqrt{\kappa}n(\sqrt{2}\alpha(n-1) + 8\sqrt{\kappa})^{-1}$ for $0 < \alpha \leq 1/8$.

SVRG/SAGA	AccGrad	SGM
$\mathcal{O}((n+\kappa)\log(1/\varepsilon))$	$\mathcal{O}((n\kappa)\log(1/\varepsilon))$	$1/\epsilon$

Table: Complexity to obtain ε-solution.

*Another way of parsing data

Example (Least squares): $\min_{\mathbf{x}} \left\{ f(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 : \mathbf{x} \in \mathbb{R}^p \right\}$



Using a subset of rows

We have mainly focused on using a subset of rows instead of the full data at each iteration.
This way, we compute an unbiased estimate $G(\mathbf{x}^k, i_k)$ of the gradient using

- a subset of data points: $(\mathbf{a}_{i_k}, b_{i_k})$,
- and the whole decision variable $\mathbf{x}^k$:

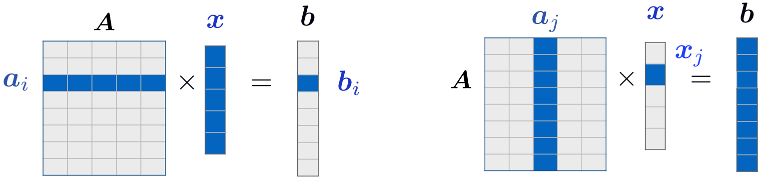
$$G(\mathbf{x}^k, i_k) = \mathbf{a}_{i_k}^T(\langle \mathbf{a}_{i_k}^T, \mathbf{x}^k \rangle - \mathbf{b}_{i_k}).$$

Estimate $G(\mathbf{x}^k, i_k)$ is dense, so we update the whole decision variable.

Next: Using a subset of columns.

*Another way of parsing data

Example (Least squares): $\min_{\mathbf{x}} \left\{ f(\mathbf{x}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 : \mathbf{x} \in \mathbb{R}^p \right\}$



Using a subset of columns

Denote the standard basis vectors by $\mathbf{e}_i$, and the corresponding directional derivatives by ∇_i. Let $\mathbf{a}_i$ represent the ith column of matrix $\mathbf{A}$. Consider the following unbiased estimate:

$$G(\mathbf{x}^k, i_k) = p\nabla_{i_k} f(\mathbf{x}^k)\mathbf{e}_{i_k} = p\langle \mathbf{a}_{i_k}, \mathbf{a}_{i_k}\mathbf{x}^k_{i_k} - \mathbf{b}\rangle \mathbf{e}_{i_k}.$$

This way, we compute an unbiased estimate $G(\mathbf{x}^k, i_k)$ of the gradient using

- a subset of columns $(\mathbf{a}_{i_k})$ and the whole measurement vector $\mathbf{b}$,
- and only the chosen coordinates of decision variable: $\mathbf{x}^k_{i_k}$.

Estimate $G(\mathbf{x}^k, i_k)$ is sparse, only coordinates chosen by i_k are nonzero. Hence, we update these coordinates only.

References I

[1] V. N. Vapnik.
An overview of statistical learning theory.
IEEE Trans. Inf. Theory, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[2] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro.
Robust stochastic approximation approach to stochastic programming.
SIAM J. Optim., vol. 19, pp. 1574–1609, 2008.

[3] J. T. Kwok, C. Hu and W. Pan.
Accelerated gradient methods for stochastic optimization and online learning.
Advances in Neural Information Processing Systems, vol. 22, pp. 781–789, 2009.

[4] A. Nitanda.
Stochastic proximal gradient descent with acceleration techniques.
Advances in Neural Information Processing Systems, pp. 1574–1582, 2014.

[5] L. Xiao, and T. Zhang.
A proximal stochastic gradient method with progressive variance reduction.
SIAM Journal on Optimization 2057-2075, 2014.

[6] A. Defazio, F. Bach, and S. Lacoste-Julien.
SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives.
Advances in Neural Information Processing Systems, pp. 1646–1654, 2014.

[7] S., Ohad, and T. Zhang.
Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes.
International Conference on Machine Learning, 2013.

[8] S.-S., Shai, et al.
Pegasos: Primal estimated sub-gradient solver for svm.
Mathematical Programming 127.1 (2011): 3-30.

References II

[9] R. Johnson, and T. Zhang.
Accelerating stochastic gradient descent using predictive variance reduction.
Advances in neural information processing systems 315–323, 2013

[10] J. Duchi, E. Hazan, and Y. Singer.
Adaptive subgradient methods for online learning and stochastic optimization.
Journal of Machine Learning Research 12, 2121-2159.

[11] L. Bottou., F. E. Curtis and J. Nocedal.
Optimization methods for large-scale machine learning.
arXiv:1606.04838, 2016 Jun 15.

[12] H. Robbins and S. Monro.
A stochastic approximation method.
Annals of Mathematical Statistics, 22:400–407, 1951.

[13] B. Polyak and A. Juditsky.
Acceleration of stochastic approximation by averaging. SIAM Journal on Control and Optimization, 1992.

[14] A. Nemirovski and D. Yudin.
Problem complexity and method efficiency in optimization. Wiley, 1983.

[15] T. Strohmer and R. Vershynin..
A randomized Kaczmarz algorithm with exponential convergence.
Journal of Fourier Analysis and Applications, 15(2), 262.

[16] D. Needell, R. Ward, and N. Srebro, N.
Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm.
Advances in Neural Information Processing Systems 2014.

[17] F. Bach and E. Moulines.
Non-strongly-convex smooth stochastic approximation with convergence rate O (1/n).
In Advances in neural information processing systems. 2013