EECS 559 Optimization Methods for SIPML

Lecture 5 – Stochastic Gradient Descent

Instructor: Prof. Qing Qu (qingqu@umich.edu)

Lecture Agenda

- **Method Introduction**
- Convergence Analysis
- Stochastic Variance Reduced Gradient

Stochastic Optimization

$$\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) \;=\; \underbrace{\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}}\left[f(\boldsymbol{x};\boldsymbol{\xi})\right]}_{\text{expected/population risk}}$$

- $\boldsymbol{\xi}$: randomness in the problem, $\mathcal{D}$: the distribution of $\boldsymbol{\xi}$.
- suppose $f(\cdot,\boldsymbol{\xi})$ is *convex* for every $\boldsymbol{\xi}$, hence $F(\boldsymbol{x})$ is *convex*
- evaluating high-dimensional expectation is expensive.

A Natural Solution

$$\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) \;=\; \underbrace{\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D}}\left[f(\boldsymbol{x};\boldsymbol{\xi})\right]}_{\text{expected/population risk}}$$

Under "mild" technical conditions:

$$\begin{aligned}\boldsymbol{x}_{k+1} \;&=\; \boldsymbol{x}_k - \tau_k \nabla F(\boldsymbol{x}_k) \;=\; \boldsymbol{x}_k - \tau_k \nabla \mathbb{E}\left[f(\boldsymbol{x}_k;\boldsymbol{\xi})\right] \\ &=\; \boldsymbol{x}_k - \tau_k \mathbb{E}\left[\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k;\boldsymbol{\xi})\right]\end{aligned}$$

Problems:

- The distribution of $\boldsymbol{\xi}$ might be *unknown*
- even if it is known, evaluating high-dimensional expectation is often difficult

Solution: Empirical Risk Minimization (ERM)

Let $\{\boldsymbol{a}_i, y_i\}_{i=1}^m$ be m random *i.i.d.* samples, and consider

$$\begin{aligned}\min_{\boldsymbol{x}\in\mathbb{R}^n} \; F_m(\boldsymbol{x}) \;&=\; \frac{1}{m}\sum_{i=1}^m f_i(\boldsymbol{x}), \quad f_i(\boldsymbol{x}) \;=\; h_i(\boldsymbol{x}) + \psi(\boldsymbol{x}) \\ &:=\; \frac{1}{m}\sum_{i=1}^m \underbrace{h(\boldsymbol{x};\{\boldsymbol{a}_i, y_i\})}_{h_i(\boldsymbol{x})\text{: loss for } i\text{th sample}} + \underbrace{\psi(\boldsymbol{x})}_{\text{regularizer}}.\end{aligned}$$

Empirical Risk Minimization (ERM)

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \; F_m(\boldsymbol{x}) \;=\; \frac{1}{m}\sum_{i=1}^m f_i(\boldsymbol{x}), \quad f_i(\boldsymbol{x}) \;=\; h_i(\boldsymbol{x}) + \psi(\boldsymbol{x}).$$

- **Linear regression:** $h_i(\boldsymbol{x}) = \frac{1}{2}\left(\boldsymbol{a}_i^\top \boldsymbol{x} - y_i\right)^2$, $\psi(\boldsymbol{x}) = 0$;
- **Logistic regression:** $h_i(\boldsymbol{x}) = \log\left(1 + e^{-y_i \boldsymbol{a}_i^\top \boldsymbol{x}}\right)$, $\psi(\boldsymbol{x}) = 0$;
- **Lasso:** $h_i(\boldsymbol{x}) = \frac{1}{2}\left(\boldsymbol{a}_i^\top \boldsymbol{x} - y_i\right)^2$, $\psi(\boldsymbol{x}) = \lambda \|\boldsymbol{x}\|_1$;
- **SVM:** $h_i(\boldsymbol{x}) = \max\left\{0, 1 - y_i \boldsymbol{a}_i^\top \boldsymbol{x}\right\}$, $\psi(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|_2$;
- **Training deep neural network...**

Gradient Descent (GD)

- **(Full batch) gradient descent:**

$$\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k - \tau_k \cdot \nabla F_m(\boldsymbol{x}_k), \quad \nabla F_m(\boldsymbol{x}) \;=\; \frac{1}{m}\sum_{i=1}^m \nabla f_i(\boldsymbol{x})$$

when m is large, computing the gradient could be expensive: *linear* in the number of samples $O(m)$.

Overall, complexity of evaluating the gradient is $O(mn)$.

Stochastic Gradient Descent (SGD)

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \; F(\boldsymbol{x}) \;:=\; \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}}\left[f(\boldsymbol{x};\boldsymbol{\xi})\right]$$

- The SGD in general can be written as

$$\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k \;-\; \tau_k \cdot \boldsymbol{g}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)$$

with $\boldsymbol{g}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)$ being an *unbiased* estimation of $\nabla F(\boldsymbol{x}_k)$

$$\mathbb{E}\left[\boldsymbol{g}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)\right] = \nabla F(\boldsymbol{x}_k)$$

Stochastic Gradient Descent (SGD)

- **Example of SGD:**

$$\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k - \tau_k \cdot \boldsymbol{g}(\boldsymbol{x}_k), \quad \boldsymbol{g}(\boldsymbol{x}_k) \;=\; \frac{1}{b}\sum_{i\in B_k} \nabla f_i(\boldsymbol{x})$$

approximate $\nabla F(\boldsymbol{x})$ by a *random batch* of samples $B_k \subset [m]$ of a fixed size $|B_k| \equiv b \ll m$.

- It reduces the complexity to $O(n)$;
- $\boldsymbol{g}(\boldsymbol{x})$ gives an *unbiased* estimator of $\nabla F(\boldsymbol{x})$:

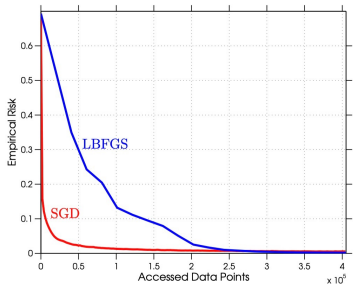
$$\mathbb{E}\left[\boldsymbol{g}(\boldsymbol{x})\right] = \nabla F(\boldsymbol{x})$$

Comparing GD vs. SGD

SGD exploits low-dim data structures more efficiently than GD:

- Data in practice usually involves lots of redundancy. Using all data simultaneously in each iteration might be inefficient;
- SGD is particularly efficient at the very beginning, as it achieves fast initial improvement with very low per-iteration cost.

SGD for Empirical Risk Minimization



Binary Classification with logistic loss and RCV1 data set

Lecture Agenda

- Method Introduction
- **Convergence Analysis**
- Stochastic Variance Reduced Gradient

Strongly Convex and Smooth Problems

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \; F \;:=\; \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}}\left[f(\boldsymbol{x};\boldsymbol{\xi})\right]$$

- f: μ-strongly convex, L-smooth;
- Given $\{\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_k\}$,

$$\boldsymbol{x}_{k+1} \;=\; \boldsymbol{x}_k \;-\; \tau_k \cdot \boldsymbol{g}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)$$

with $\boldsymbol{g}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)$ being an *unbiased* estimate of $\nabla F(\boldsymbol{x}_k)$.

- For all $\boldsymbol{x}$,

$$\mathbb{E}\left[\left\|\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\xi})\right\|_2^2\right] \;\le\; \sigma_g^2 + c_g\left\|\nabla F(\boldsymbol{x})\right\|_2^2$$

Convergence of SGD with Fixed Stepsizes

Theorem. Assume that f is μ-strongly convex and L-smooth, and

$$\mathbb{E}\left[\left\|\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\xi})\right\|_2^2\right] \;\le\; \sigma_g^2 + c_g\left\|\nabla F(\boldsymbol{x})\right\|_2^2.$$

If the stepsize $\tau_k \equiv \tau \le 1/(Lc_g)$, then SGD achieves

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}_\star) \;\le\; \underbrace{(1-\tau\mu)^k}_{\text{linear convergence}} \left(F(\boldsymbol{x}_0) - F(\boldsymbol{x}_\star)\right) + \frac{\tau L \sigma_g^2}{2\mu}$$

Check Theorem 4.6 of Bottou, Curtis, Nocedal'18 for the proof.

Implications: SGD with Fixed Stepsizes

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}_\star) \;\le\; \underbrace{(1-\tau\mu)^k}_{\text{linear convergence}} \left(F(\boldsymbol{x}_0) - F(\boldsymbol{x}_\star)\right) + \frac{\tau L \sigma_g^2}{2\mu}$$

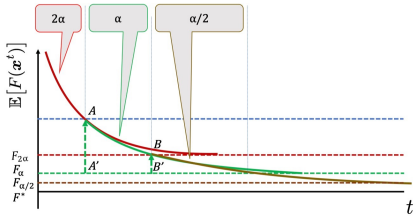
- Fast (linear!) convergence at the very beginning
- Converges to some neighborhood of $\boldsymbol{x}_\star$ due to

$$\mathbb{E}\left[\left\|\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\xi})\right\|_2^2\right] \;\le\; \sigma_g^2 + c_g\left\|\nabla F(\boldsymbol{x})\right\|_2^2$$

- when $\sigma_g = 0$, it converges linearly to optimal points
- smaller stepsizes τ yield better converging points

One Practical Strategy

Run SGD with fixed stepsizes; whenever progress stalls, reduce stepsizes and continue SGD



whenever progress stalls, we half the stepsizes and repeat

Convergence with Diminishing Stepsizes

Theorem. (Convergence of SGD with Diminishing Stepsizes) Suppose F is μ-strongly convex, and the following

$$\mathbb{E}\left[\|\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\xi})\|_2^2\right] \leq \sigma_g^2 + c_g\|\nabla F(\boldsymbol{x})\|_2^2$$

holds with $c_g = 0$. If $\tau_k = \frac{\theta}{k+1}$ for some $\theta > \frac{1}{2\mu}$, then

$$\boxed{\mathbb{E}\left[\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2^2\right] \leq \frac{c_\theta}{k+1},}$$

where $c_\theta = \max\left\{\frac{2\theta^2\sigma_g^2}{2\mu\theta - 1}, \|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2\right\}$.

Optimality of Convergence Rate

Theorem. (Informal, Nemirovski & Yudin'83) For strongly convex functions, no algorithm performing k queries to noisy first-order oracles can achieve an accuracy better than $O(1/k)$.

- SGD with stepsize τ_k can achieve optimal order of convergence rate $O(1/k)$.

Comparing SGD with GD

	Iter. Complexity	Per-Iter Cost	Total Comput. Cost
GD	$O(\log\frac{1}{\varepsilon})$	$O(mn)$	$O(mn\log\frac{1}{\varepsilon})$
SGD	$O(\frac{1}{\varepsilon})$	$O(n)$	$O(\frac{n}{\varepsilon})$

SGD is more appealing than GD for *large* m and *moderate accuracy* ε (whenever $\frac{1}{\varepsilon} < m\log\frac{1}{\varepsilon}$).

➢ it often arises in the *big data* regime in machine learning !

General Convex Problems

What if we lose *strong* convexity?

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} F := \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}}\left[f(\boldsymbol{x};\boldsymbol{\xi})\right]$$

- f : convex, L-smooth;
- Given $\{\boldsymbol{\xi}_0, \cdots, \boldsymbol{\xi}_k\}$, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \tau_k \cdot \boldsymbol{g}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)$

with $\boldsymbol{g}(\boldsymbol{x}_k;\boldsymbol{\xi}_k)$ being an *unbiased* estimate of $\nabla F(\boldsymbol{x}_k)$.

- For all $\boldsymbol{x}$, $\mathbb{E}\left[\|\boldsymbol{g}(\boldsymbol{x};\boldsymbol{\xi})\|_2^2\right] \leq \sigma_g^2 + c_g\|\nabla F(\boldsymbol{x})\|_2^2$

General Convex Problems

Suppose we return a weighted average

$$\widetilde{\boldsymbol{x}}_k := \sum_{i=0}^{k}\frac{\tau_i}{\sum_{j=0}^{k}\tau_j}\boldsymbol{x}_i, \quad \boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \tau_i\cdot\boldsymbol{g}(\boldsymbol{x}_i;\boldsymbol{\xi}_i)$$

Theorem.(Convergence for Convex Problems) Under the assumptions on the previous page, one has

$$\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}_k) - F(\boldsymbol{x}_\star)\right] \leq \frac{1}{2\sum_{i=0}^{k}\tau_i}\left(\mathbb{E}\left[\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2\right] + \sigma_g^2\sum_{i=0}^{k}\tau_i^2\right).$$

General Convex Problems

Theorem. (Convergence for Convex Problems) Under the assumptions on the previous page, one has

$$\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}_k) - F(\boldsymbol{x}_\star)\right] \leq \frac{1}{2\sum_{i=0}^{k}\tau_i}\left(\mathbb{E}\left[\|\boldsymbol{x}_0 - \boldsymbol{x}_\star\|_2^2\right] + \sigma_g^2\sum_{i=0}^{k}\tau_i^2\right).$$

If $\tau_k \asymp 1/\sqrt{k}$, then

$$\mathbb{E}\left[F(\widetilde{\boldsymbol{x}}_k) - F(\boldsymbol{x}_\star)\right] \leq O\left(\frac{\log k}{\sqrt{k}}\right)$$

Lecture Agenda

- Method Introduction
- Convergence Analysis
- **Stochastic Variance Reduced Gradient**

Vanilla SGD

For μ-strongly convex problem, what we have learned so far:

- SGD with (large) constant stepsize $\tau_k \in O(1)$ tends to oscillate around global minimum:

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}_\star) \leq (1-\tau\mu)^k\left(F(\boldsymbol{x}_0) - F(\boldsymbol{x}_\star)\right) + \frac{\tau L\sigma_g^2}{2\mu}$$

- choosing conservative $\tau_k \in O(1/k)$ mitigates oscillation, but results in slow convergence

$$\mathbb{E}\left[\|\boldsymbol{x}_k - \boldsymbol{x}_\star\|_2^2\right] \leq \frac{c_\theta}{k+1}.$$

Design $\boldsymbol{g}_k$ with Reduced Variability σ_g^2 ?

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}_\star) \;\le\; (1-\tau\mu)^k \left(F(\boldsymbol{x}_0) - F(\boldsymbol{x}_\star)\right) + \frac{\tau L \sigma_g^2}{2\mu}$$

- vanilla SGD: $\boldsymbol{g}_k = \nabla f_{i_k}(\boldsymbol{x}_k)$, $i_k \sim \mathcal{U}([m])$
  σ_g^2 is non-negligible even when $\boldsymbol{x}_k = \boldsymbol{x}_\star$;
- **question:** design $\boldsymbol{g}_k$ with reduced variability σ_g^2 ?

Design $\boldsymbol{g}_k$ with Reduced Variability σ_g^2 ?

$$F(\boldsymbol{x}_k) - F(\boldsymbol{x}_\star) \;\le\; (1-\tau\mu)^k \left(F(\boldsymbol{x}_0) - F(\boldsymbol{x}_\star)\right) + \frac{\tau L \sigma_g^2}{2\mu}$$

- **question:** design $\boldsymbol{g}_k$ with reduced variability σ_g^2 ?
- **idea:** take some $\boldsymbol{v}_k$ that (i) $\langle \boldsymbol{v}_k, \nabla f_{i_k}(\boldsymbol{x}_k)\rangle > 0$, (ii) $\mathbb{E}[\boldsymbol{v}_k] = \boldsymbol{0}$

$$\boldsymbol{g}_k \;=\; \nabla f_{i_k}(\boldsymbol{x}_k) - \boldsymbol{v}_k$$

use historical gradient information to produce such a $\boldsymbol{v}_k$

➢ so $\boldsymbol{g}_k$ is still an unbiased estimate of $\nabla F(\boldsymbol{x}_k)$ with reduced σ_g^2

Stochastic Variance Reduced Gradient (SVRG)

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \; F_m(\boldsymbol{x}) \;=\; \frac{1}{m}\sum_{i=1}^m f_i(\boldsymbol{x})$$

- f_i: convex and L-smooth, no regularizations
- F_m: μ-strongly convex
- $\kappa := L/\mu$: condition number

Stochastic Variance Reduced Gradient (SVRG)

Key idea: if we have access to a historical *anchor* point $\boldsymbol{x}^a$ and $\nabla F_m(\boldsymbol{x}^a)$, then

$$\underbrace{\nabla f_{i_k}(\boldsymbol{x}_k) - \nabla f_{i_k}(\boldsymbol{x}^a)}_{\to \boldsymbol{0} \text{ if } \boldsymbol{x}_k \approx \boldsymbol{x}^a} + \underbrace{\nabla F_m(\boldsymbol{x}^a)}_{\to \boldsymbol{0} \text{ if } \boldsymbol{x}^a \approx \boldsymbol{x}_\star}, \quad \text{with} \quad i_k \sim \mathcal{U}([m]).$$

- is an *unbiased* estimator of $\nabla F(\boldsymbol{x}_k)$
- converges to $\boldsymbol{0}$ when $\boldsymbol{x}_k \;\approx\; \boldsymbol{x}_a \;\approx\; \boldsymbol{x}_\star$

Stochastic Variance Reduced Gradient (SVRG)

- Operate in *epochs*
- In the k-th epoch:
  1. **very beginning**: record the anchor point $\boldsymbol{x}_k^{\boldsymbol{a}}$, and compute the *full batch* gradient $\nabla F_m(\boldsymbol{x}_k^a)$
  2. **inner loop**: use the anchor point $\boldsymbol{x}_k^{\boldsymbol{a}}$ to reduce the variance

$$\boldsymbol{x}_{k,t+1} \;=\; \boldsymbol{x}_{k,t} \;-\; \tau\left[\nabla f_{i_{k,t}}(\boldsymbol{x}_{k,t}) - \nabla f_{i_{k,t}}(\boldsymbol{x}_k^a) + \nabla F_m(\boldsymbol{x}_k^a)\right].$$

A hybrid approach: the *full batch* gradient is computed *only once* per epoch

Algorithm Pipeline for SVRG

Algorithm 1 SVRG for finite-sum optimization

for $k = 1, 2, \cdots$ **do**
  $\boldsymbol{x}_k^a \leftarrow \boldsymbol{x}_{k-1,N}$ and compute $\underbrace{\nabla F_m(\boldsymbol{x}_k^a)}_{\text{full batch gradient}}$
  initialize $\boldsymbol{x}_{k,0} \leftarrow \boldsymbol{x}_k^a$
  for $\underbrace{t = 0, \cdots, N-1}_{N \text{ inner loops for each epoch}}$ **do**
    draw $i_{k,t}$ uniformly from $[m]$ and compute
    $\boldsymbol{x}_{k,t+1} \;=\; \boldsymbol{x}_{k,t} \;-\; \tau\underbrace{[\nabla f_{i_{k,t}}(\boldsymbol{x}_{k,t}) - \nabla f_{i_{k,t}}(\boldsymbol{x}_k^a) + \nabla F_m(\boldsymbol{x}_k^a)]}_{\text{stochastic gradient}}$
  end for
end for

Remarks for SVRG

- We use constant stepsize τ
- Each epoch contains $2N+m$ gradient computations of f_i
  - ➢ the full batch gradient is computed only once every N iterations;
  - ➢ the average per-iteration cost of SVRG is comparable to that of SGD if $N \asymp m$.

Convergence Analysis of SVRG

Theorem (Johnson & Zhang'13)
Assume each f_i is convex and L-smooth and F is μ–strongly convex. Choose the inner loop N large enough such that

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

then we have

$$\mathbb{E}\left[F_m(\boldsymbol{x}_k^a) - F_m(\boldsymbol{x}_\star)\right] \;\le\; \rho^k \cdot \left(F_m(\boldsymbol{x}_0^a) - F(\boldsymbol{x}_\star)\right).$$

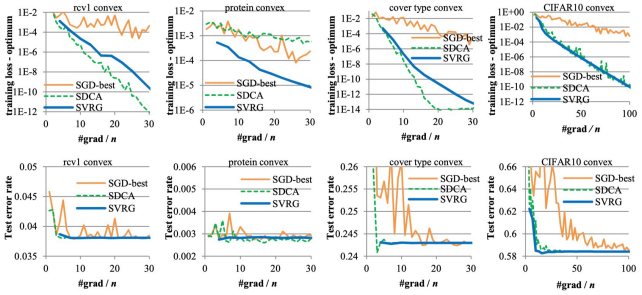
Linear convergence: choose $N \ge L/\mu = \kappa$ and stepsize $\tau \in O(1/L)$ so that $0 < \rho < 1/2$ and $O(\log\frac{1}{\varepsilon})$ epochs to attain ε-accuracy.

See also **Theorem 5.1** in Léon Bottou, Frank E. Curtis, Jorge Nocedal Optimization Methods for Large-Scale Machine Learning, 2018

Comparisons with GD and SGD

	SVRG	GD	SGD
Comp. Cost	$(n+\kappa)\log\frac{1}{\varepsilon}$	$n\kappa\log\frac{1}{\varepsilon}$	$\frac{\kappa^2}{\varepsilon}$

Numerical Example: Logistic Regression



ℓ_2-regularized logistic regression on CIFAR-10

Further Readings

- *Optimization Methods for Large-scale Machine Learning.* Léon Bottou, Frank E. Curtis, and Jorge Nocedal. SIAM Review, 2018.
- *Convex Optimization: Algorithms & Complexity.* Sebastien Bubeck, Foundation & Trends in Machine Learning (Chapter 6).
- *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications.* John Wright, Yi Ma. (**Chapter 8.6**)

Further Readings

- *Problem Complexity and Method Efficiency in Optimization.* Arkadi Nemirovski and David Yudin, Wiley, 1983.
- *Accelerating Stochastic Gradient Descent using Predictive Variance Reduction.* Rie Johnson and Tong Zhang, NeurIPS'13, 2013.
- *A Proximal Stochastic Gradient Method with Progressive Variance Reduction.* Lin Xiao and Tong Zhang, SIAM Journal on Optimization, 2014.
- *Recent Advances in Stochastic Convex & Nonconvex Optimization.* Zeyuan Allen-Zhu, ICML Tutorial, 2017. http://people.csail.mit.edu/zeyuan/topics/icml-2017