# Optimization for Machine Learning

**(Lecture 3-A - Convex)**

**Suvrit Sra**

**Massachusetts Institute of Technology**

**Special thanks: Francis Bach (INRIA, ENS)**

(for sharing this material, and permitting its use)

**MPI-IS Tübingen**

**Machine Learning Summer School, June 2017**

# Course materials

- http://suvrit.de/teaching.html
- Some references:
  - *Introductory lectures on convex optimization* – Nesterov
  - *Convex optimization* – Boyd & Vandenberghe
  - *Nonlinear programming* – Bertsekas
  - *Convex Analysis* – Rockafellar
  - *Fundamentals of convex analysis* – Urruty, Lemaréchal
  - *Lectures on modern convex optimization* – Nemirovski
  - *Optimization for Machine Learning* – Sra, Nowozin, Wright
  - *NIPS 2016 Optimization Tutorial* – Bach, Sra
- Some related courses:
  - EE227A, Spring 2013, (Sra, UC Berkeley)
  - 10-801, Spring 2014 (Sra, CMU)
  - EE364a,b (Boyd, Stanford)
  - EE236b,c (Vandenberghe, UCLA)
- Venues: NIPS, ICML, UAI, AISTATS, SIOPT, Math. Prog.

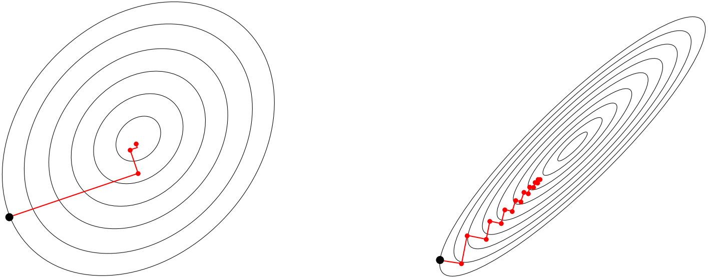
# Lecture Plan

– Introduction
– Recap of convexity, sets, functions
– Recap of duality, optimality, problems
– First-order optimization algorithms and techniques
– Large-scale optimization (SGD and friends)
– Directions in non-convex optimization

# Iteration complexity: smooth problems

- **Assumption**: $f$ convex and $L$-smooth on $\mathbb{R}^d$
- **Gradient descent:** $\theta_t = \theta_{t-1} - \gamma_k g'(\theta_{t-1})$

  $g(\theta_t) - g(x^*) \leqslant O(1/t)$
  $g(\theta_t) - g(x^*) \leqslant O(e^{-t(\mu/L)}) = O(e^{-t/\kappa})$ if $\mu$-strongly convex



(small $\kappa = L/\mu$) (large $\kappa = L/\mu$)

# Iteration complexity: smooth problems

- **Assumption**: $f$ convex and $L$-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_k g'(\theta_{t-1})$
  $O(1/t)$ convergence rate for convex functions
  $O(e^{-\frac{t}{\kappa}})$ if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$

# Iteration complexity: smooth problems

- **Assumption**: $f$ convex and $L$-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_k g'(\theta_{t-1})$
  $O(1/t)$ convergence rate for convex functions
  $O(e^{-\frac{t}{\kappa}})$ if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$

▶ **Key insights for ML (Bottou and Bousquet, 2008)**

1. No need to optimize below statistical error

# Iteration complexity: smooth problems

– **Assumption**: $f$ convex and $L$-smooth on $\mathbb{R}^d$

– **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_k g'(\theta_{t-1})$

$O(1/t)$ convergence rate for convex functions

$O(e^{-\frac{t}{\kappa}})$ if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$

▶ **Key insights for ML (Bottou and Bousquet, 2008)**

1. No need to optimize below statistical error
2. Cost functions are averages

# Iteration complexity: smooth problems

– **Assumption**: $f$ convex and $L$-smooth on $\mathbb{R}^d$

– **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_k g'(\theta_{t-1})$

$O(1/t)$ convergence rate for convex functions

$O(e^{-\frac{t}{\kappa}})$ if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$

▶ **Key insights for ML (Bottou and Bousquet, 2008)**

1. No need to optimize below statistical error
2. Cost functions are averages
3. Testing error is more important than training error

# Stochastic gradient descent for finite sums

$$\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

- **Iteration**: $\theta_t = \theta_{t-1} - \gamma_k f'_{i(t)}(\theta_{t-1})$
  - Sampling with replacement: $i(t) \sim \text{Unif}(\{1, \ldots, n\})$
  - Polyak-Ruppert averaging: $\bar{\theta}_t = \frac{1}{t+1} \sum_{u=0}^{t} \theta_u$
- **Convergence rate** if each $f_i$ is convex $L$-smooth and $f$ $\mu$-strongly-convex:

$$\mathbb{E}[g(\bar{\theta}_t) - g(\theta^*)] \leqslant \begin{cases} O(1/\sqrt{k}) & \text{if } \gamma_k = 1/(L\sqrt{k}) \\ O(L/(\mu k)) = O(\kappa/k) & \text{if } \gamma_k = 1/(\mu k) \end{cases}$$

# Stochastic vs. deterministic – strongly cvx

- Min $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$
- Batch gradient descent:

$$\theta_t = \theta_{t-1} - \gamma_k g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_k}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$$

– Linear (e.g., exponential) convergence rate in $O(e^{-t/\kappa})$

– Iteration complexity is linear in $n$

# Stochastic vs. deterministic – strongly cvx

- Min $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$
- Batch gradient descent:

$$\theta_t = \theta_{t-1} - \gamma_k g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_k}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$$

  – Linear (e.g., exponential) convergence rate in $O(e^{-t/\kappa})$

  – Iteration complexity is linear in $n$
- Stochastic gradient descent: $\theta_t = \theta_{t-1} - \gamma_k f'_{i(t)}(\theta_{t-1})$

  – Sampling with replacement: $i(t)$ random element of $\{1, \ldots, n\}$

  – Convergence rate in $O(\kappa/t)$

  – Iteration complexity is independent of $n$

# Stochastic vs. deterministic – strongly cvx

- Min $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i,\theta)\big) + \lambda\Omega(\theta)$
- Batch gradient descent:

$$\theta_t = \theta_{t-1} - \gamma_k g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_k}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$$

  – Linear (e.g., exponential) convergence rate in $O(e^{-t/\kappa})$

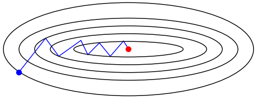
  – Iteration complexity is linear in $n$

- Stochastic gradient descent: $\theta_t = \theta_{t-1} - \gamma_k f'_{i(t)}(\theta_{t-1})$

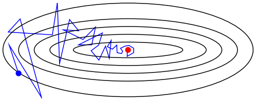
  – Sampling with replacement: $i(t)$ random element of $\{1,\ldots,n\}$

  – Convergence rate in $O(\kappa/t)$

  – Iteration complexity is independent of $n$



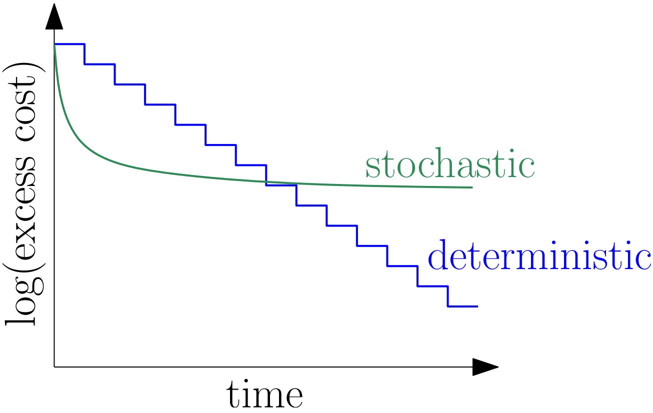
GD



SGD

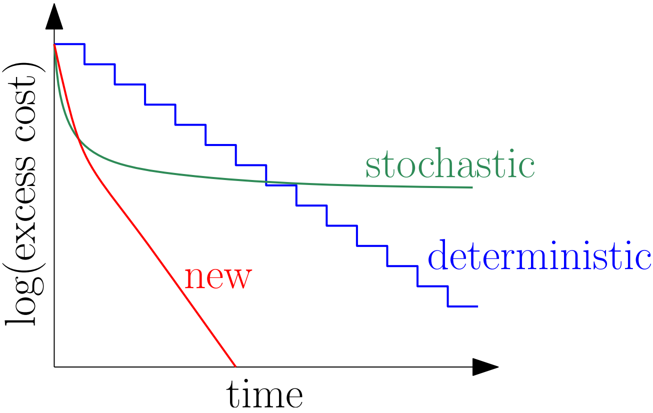
# Stochastic vs. deterministic methods

**Goal** = best of both worlds: Linear rate with $O(d)$ iteration cost
Simple choice of step size



# Stochastic vs. deterministic methods

**Goal** = best of both worlds: Linear rate with $O(d)$ iteration cost
Simple choice of step size



# Linearly convergent stochastic gradient algorithms

- **Many related algorithms**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SDCA (Shalev-Shwartz and Zhang, 2013)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - MISO (Mairal, 2015)
  - Finito (Defazio et al., 2014b)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014a)
  - ...
- **Similar rates of convergence and iterations**

# Linearly convergent stochastic gradient algorithms

- **Many related algorithms**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SDCA (Shalev-Shwartz and Zhang, 2013)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - MISO (Mairal, 2015)
  - Finito (Defazio et al., 2014b)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014a)
  - ...
- **Similar rates of convergence and iterations**
- **Different interpretations and proofs / proof lengths**
  - Lazy gradient evaluations
  - Variance reduction

# Running-time comparisons (strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each $f_i$ convex $L$-smooth and $f$ is $\mu$-strongly convex

# Running-time comparisons (strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each $f_i$ convex $L$-smooth and $f$ is $\mu$-strongly convex

| | | | |
|---|---|---|---|
| Stochastic gradient descent | $d\times$ | $\frac{L}{\mu}$ | $\times \quad \frac{1}{\varepsilon}$ |
| Gradient descent | $d\times$ | $n\frac{L}{\mu}$ | $\times \log \frac{1}{\varepsilon}$ |
| Accelerated gradient descent | $d\times$ | $n\sqrt{\frac{L}{\mu}}$ | $\times \log \frac{1}{\varepsilon}$ |
| SAG/SVRG | $d\times$ | $(n + \frac{L}{\mu})$ | $\times \log \frac{1}{\varepsilon}$ |

# Running-time comparisons (strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each $f_i$ convex $L$-smooth and $f$ is $\mu$-strongly convex

| | | | |
|---|---|---|---|
| Stochastic gradient descent | $d\times$ | $\frac{L}{\mu}$ | $\times \quad \frac{1}{\varepsilon}$ |
| Gradient descent | $d\times$ | $n\frac{L}{\mu}$ | $\times \log \frac{1}{\varepsilon}$ |
| Accelerated gradient descent | $d\times$ | $n\sqrt{\frac{L}{\mu}}$ | $\times \log \frac{1}{\varepsilon}$ |
| SAG/SVRG | $d\times$ | $(n + \frac{L}{\mu})$ | $\times \log \frac{1}{\varepsilon}$ |

- **Beating two lower bounds** (Nemirovski and Yudin, 1983; Nesterov, 2004): with additional assumptions
  (1) stochastic gradient: exponential rate for finite sums
  (2) full gradient: better exponential rate using the sum structure

# Running-time comparisons (non-strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each $f_i$ convex $L$-smooth
  - Ill conditioned problems: $f$ may not be strongly-convex

| | | |
|---|---|---|
| Stochastic gradient descent | $d\times$ | $1/\varepsilon^2$ |
| Gradient descent | $d\times$ | $n/\varepsilon$ |
| Accelerated gradient descent | $d\times$ | $n/\sqrt{\varepsilon}$ |
| SAG/SVRG | $d\times$ | $\sqrt{n}/\varepsilon$ |

# Running-time comparisons (non-strongly-convex)

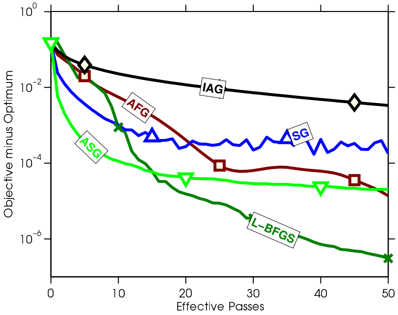
- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each $f_i$ convex $L$-smooth
  - Ill conditioned problems: $f$ may not be strongly-convex

| | | |
|---|---|---|
| Stochastic gradient descent | $d\times$ | $1/\varepsilon^2$ |
| Gradient descent | $d\times$ | $n/\varepsilon$ |
| Accelerated gradient descent | $d\times$ | $n/\sqrt{\varepsilon}$ |
| SAG/SVRG | $d\times$ | $\sqrt{n}/\varepsilon$ |

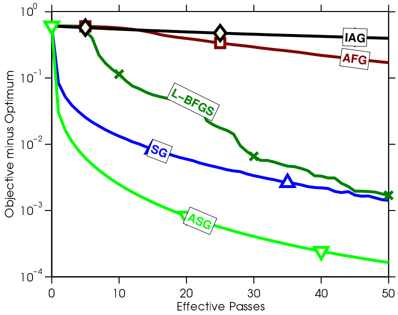
- Adaptivity to potentially hidden strong convexity
- No need to know the local/global strong-convexity constant

# Experimental results (logistic regression)

`quantum` dataset
$(n = 50\,000, d = 78)$

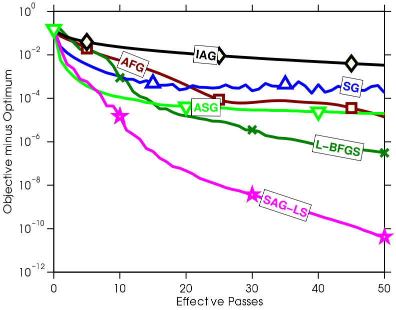



`rcv1` dataset
$(n = 697\,641, d = 47\,236)$

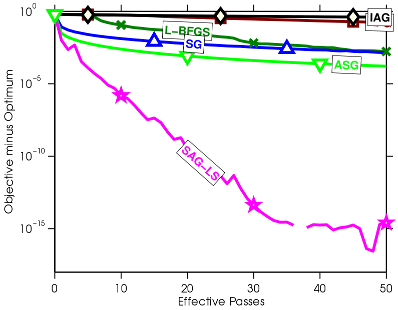


# Experimental results (logistic regression)

`quantum` dataset
($n = 50\,000, d = 78$)




`rcv1` dataset
($n = 697\,641, d = 47\,236$)



# Key Idea: Variance reduction

**Principle**: reducing variance of sample of $X$ by using a sample from another random variable $Y$ with known expectation

$$Z_\alpha = \alpha(X - Y) + \mathbb{E}Y$$

- $\mathbb{E}Z_\alpha = \alpha\mathbb{E}X + (1-\alpha)\mathbb{E}Y$
- $\mathrm{var}(Z_\alpha) = \alpha^2\big[\mathrm{var}(X) + \mathrm{var}(Y) - 2\mathrm{cov}(X, Y)\big]$
- $\alpha = 1$: no bias, $\alpha < 1$: potential bias (but reduced variance)
- Useful if $Y$ positively correlated with $X$

# Key Idea: Variance reduction

**Principle**: reducing variance of sample of $X$ by using a sample from another random variable $Y$ with known expectation

$$Z_\alpha = \alpha(X - Y) + \mathbb{E}Y$$

- $\mathbb{E}Z_\alpha = \alpha\mathbb{E}X + (1-\alpha)\mathbb{E}Y$
- $\text{var}(Z_\alpha) = \alpha^2\big[\text{var}(X) + \text{var}(Y) - 2\text{cov}(X, Y)\big]$
- $\alpha = 1$: no bias, $\alpha < 1$: potential bias (but reduced variance)
- Useful if $Y$ positively correlated with $X$

**Application to gradient estimation** (Johnson and Zhang, 2013; Zhang, Mahdavi, and Jin, 2013)

- SVRG: $X = f'_{i(t)}(\theta_{t-1})$, $Y = f'_{i(t)}(\tilde{\theta})$, $\alpha = 1$, with $\tilde{\theta}$ stored
- $\mathbb{E}Y = \frac{1}{n}\sum_{i=1}^n f'_i(\tilde{\theta})$ full gradient at $\tilde{\theta}$;
  $X - Y = f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\tilde{\theta})$

# Stochastic variance reduced gradient (SVRG)

- Initialize $\tilde{\theta} \in \mathbb{R}^d$
- For $i_{\text{epoch}} = 1$ to # of epochs
  - Compute all gradients $f_i'(\tilde{\theta})$ ; store $g'(\tilde{\theta}) = \frac{1}{n}\sum_{i=1}^n f_i'(\tilde{\theta})$
  - Initialize $x_0 = \tilde{\theta}$
  - For $t = 1$ to length of epochs

$$\theta_t = \theta_{t-1} - \gamma\Big[g'(\tilde{\theta}) + \big(f_{i(t)}'(\theta_{t-1}) - f_{i(t)}'(\tilde{\theta})\big)\Big]$$

  - Update $\tilde{\theta} = \theta_t$
- Output: $\tilde{\theta}$

– two gradient evaluations per inner step; no need to store gradients (SAG needs storage)

– Two parameters: length of epochs + step-size $\gamma$

– Same linear convergence rate as SAG, simpler proof

# SVRG vs. SAGA

- **SAGA update**:

$$\theta_t = \theta_{t-1} - \gamma \Big[ \frac{1}{n} \sum_{i=1}^n y_i^{t-1} + \big( f'_{i(t)}(\theta_{t-1}) - y_{i(t)}^{t-1} \big) \Big]$$

- **SVRG update**:

$$\theta_t = \theta_{t-1} - \gamma \Big[ \frac{1}{n} \sum_{i=1}^n f'_i(\tilde{\theta}) + \big( f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\tilde{\theta}) \big) \Big]$$

| | SAGA | SVRG |
|---|---|---|
| **Storage of gradients** | **yes** | **no** |
| Epoch-based | no | yes |
| Parameters | step-size | step-size & epoch lengths |
| Gradient evaluations per step | 1 | at least 2 |
| Adaptivity to strong-convexity | yes | no |
| Robustness to ill-conditioning | yes | no |

# Proximal extensions

– **Composite optimization problems**: $\min_{x \in \mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) + h(\theta)$

- $f_i$ smooth and convex
- $h$ convex, potentially non-smooth
- Constrained optimization: $h$ an indicator function
- Sparsity-inducing norms, e.g., $h(\theta) = \|\theta\|_1$

– **Proximal methods (a.k.a. splitting methods)**

- Projection / soft-thresholding step after gradient update
- See, e.g., Combettes and Pesquet (2011); Bach, Jenatton, Mairal, and Obozinski (2012); Parikh and Boyd (2014)

– **Directly extends to variance-reduced gradient techniques**
Same rates of convergence

# SGD minimizes the testing cost!

- **Goal**: minimize $g(\theta) = \mathbb{E}_{p(x,y)} \ell(y, \theta^\top \Phi(x))$
  - Given $n$ independent samples $(x_i, y_i)_{i=1}^n$, from $p(x,y)$
  - Given a single pass of stochastic gradient descent
  - Bounds on the excess testing cost $\mathbb{E} g(\bar{\theta}_n) - \inf_{x \in \mathbb{R}^d} g(\theta)$
- **Optimal convergence rates**: $O(1/\sqrt{n})$ and $O(1/(n\mu))$
  - Optimal for non-smooth (Nemirovski and Yudin, 1983)
  - Attained by averaged SGD with decaying step-sizes

# References

R. Babanezhad, M. O. Ahmed, A. Virani, M. W. Schmidt, J. Konecný, and S. Sallinen. Stopwasting my gradients: Practical SVRG. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4 (1):1–106, 2012.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Adv. NIPS*, 2008.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014a.

A. Defazio, J. Domke, and T. S. Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *Proc. ICML*, 2014b.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

A. S. Nemirovski and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley & Sons, 1983.

Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Kluwer, 2004.

N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14 (Feb):567–599, 2013.

L. Zhang, M. Mahdavi, and R. Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, 2013.