Stochastic Optimization
Online and batch stochastic optimization methods

† ‡ Taiji Suzuki

†Tokyo Institute of Technology
Graduate School of Information Science and Engineering
Department of Mathematical and Computing Sciences
‡JST, PRESTO

Intensive course @ Nagoya University

Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

Outline

Two types of stochastic optimization

- Online type stochastic optimization:
  - We observe data sequentially.
  - We don't need to wait until the whole sample is obtained.
  - Each observation is obtained just once (basically).

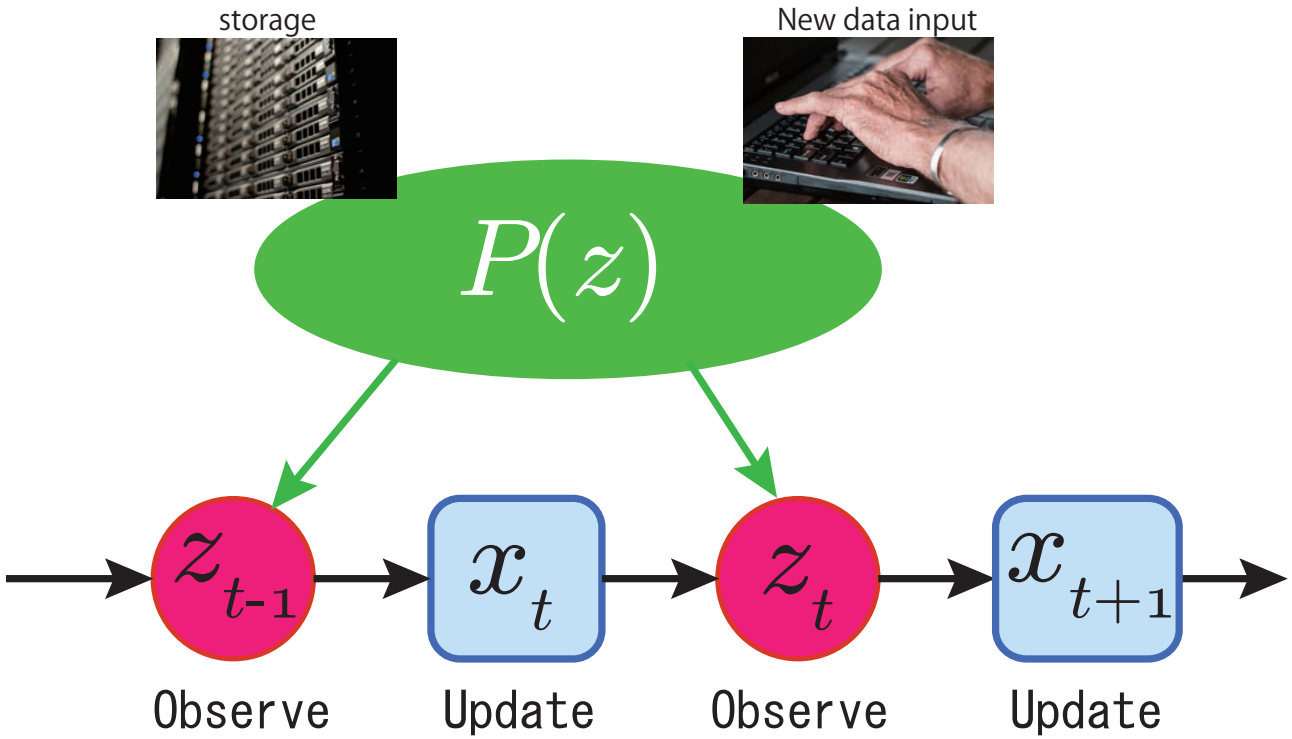
$$\min_x \ \mathrm{E}_Z[\ell(Z, x)]$$

- Batch type stochastic optimization
  - The whole sample has been already observed.
  - We can make use of the (finite and fixed) sample size.
  - We may use sample multiple times.

$$\min_x \ \frac{1}{n} \sum_{i=1}^{n} \ell(z_i, x)$$

Online method

You don't need to wait until the whole sample arrives.
Update the parameter at each data observation.



$$\min_x \quad \underline{\mathrm{E}[\ell(Z,x)]} + \psi(x) \simeq \min_x \quad \underline{\frac{1}{T}\sum_{t=1}^{T} \ell(z_t,x)} + \psi(x)$$

The objective of online stochastic optimization

Let $\ell(z, x)$ be a loss of x for an observation z.

(Expected loss) $$L(x) = \mathrm{E}_Z[\ell(Z, x)]$$

or

(EL with regularization) $$L_\psi(x) = \mathrm{E}_Z[\ell(Z, x)] + \psi(x)$$

The distribution of Z could be

- the true population
  $\rightarrow$ $L(x)$ is the **generalization error**.
- an empirical distribution of stored data in a storage
  $\rightarrow$ L (or L_ψ) is the (regularized) empirical risk.

Online stochastic optimization itself is learning!

Outline

Three steps to stochastic gradient descent

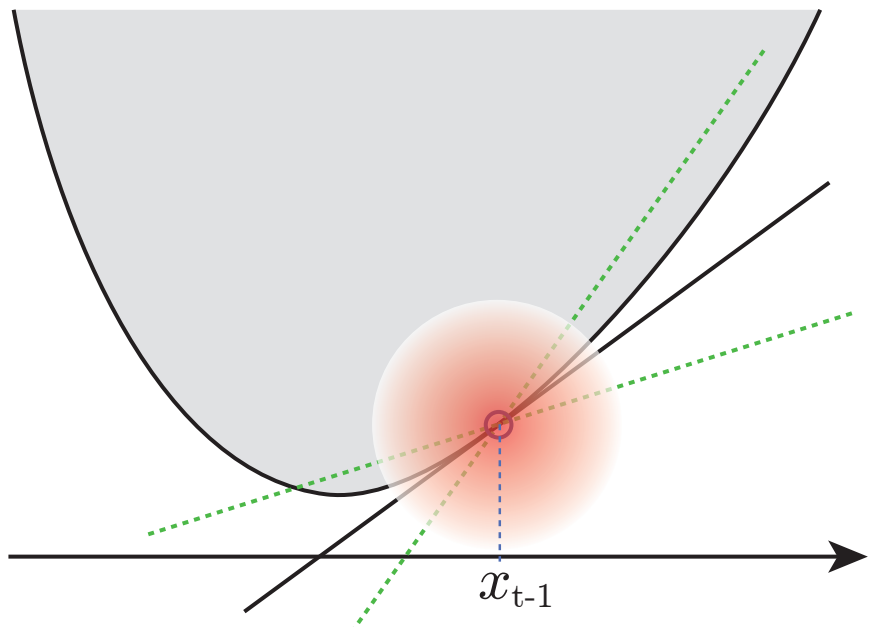
$$\mathrm{E}[\ell(Z,x)] = \int \ell(Z,x)\mathrm{d}P(Z)$$

$$\underset{(1)}{\simeq} \ell(z_t, x_{t-1}) \qquad \text{(sampling)}$$

$$\underset{(2)}{\simeq} \langle \nabla_x \ell(z_t, x_{t-1}), x \rangle \qquad \text{(linearlization)}$$

The approximation is correct just around x_{t-1}.

$$\min_x \mathrm{E}[\ell(Z,x)] \underset{(3)}{\simeq} \min_x \left\{ \langle \nabla_x \ell(z_t, x_{t-1}), x \rangle + \frac{1}{2\eta_t} \|x - x_{t-1}\|^2 \right\} \qquad \text{(proximation)}$$



Stochastic gradient descent (SGD)

SGD (without regularization)

- Observe $z_t \sim P(Z)$, and let $\ell_t(x) := \ell(z_t, x)$.
- Calculate subgradient:

$$g_t \in \partial_x \ell_t(x_{t-1}).$$

- Update x as

$$x_t = x_{t-1} - \eta_t g_t.$$

- We just need to observe one training data z_t at each iteration.
  $\rightarrow$ $O(1)$ computation per iteration ($O(n)$ for batch gradient descent).
- We do not need to go through the whole sample $\{z_i\}_{i=1}^n$

Reminder: $\mathrm{prox}(q|\psi) := \mathrm{argmin}_x \left\{\psi(x) + \frac{1}{2}\|x - q\|^2\right\}$.

Stochastic gradient descent (SGD)

SGD (with regularization)

- Observe $z_t \sim P(Z)$, and let $\ell_t(x) := \ell(z_t, x)$.
- Calculate subgradient:

$$g_t \in \partial_x \ell_t(x_{t-1}).$$

- Update x as

$$x_t = \mathrm{prox}(x_{t-1} - \eta_t g_t | \eta_t \psi).$$

- We just need to observe one training data z_t at each iteration.
  $\rightarrow$ $O(1)$ computation per iteration ($O(n)$ for batch gradient descent).
- We do not need to go through the whole sample $\{z_i\}_{i=1}^n$

Reminder: $\mathrm{prox}(q|\psi) := \mathrm{argmin}_x \left\{ \psi(x) + \frac{1}{2}\|x - q\|^2 \right\}$.

Convergence analysis of SGD

Assumption

(A1) $\mathrm{E}[\|g_t\|^2] \leq G^2$.
(A2) $\mathrm{E}[\|x_t - x^*\|^2] \leq D^2$.

Theorem

Let $\bar{x}_T = \frac{1}{T+1}\sum_{t=0}^{T} x_t$ *(Polyak-Ruppert averaging). For* $\eta_t = \frac{\eta_0}{\sqrt{t}}$, *it holds*

$$\mathrm{E}_{z_{1:T}}[L_\psi(\bar{x}_T) - L_\psi(x^*)] \leq \frac{\eta_0 G^2 + D^2/\eta_0}{\sqrt{T}}.$$

- For $\eta_0 = \frac{D}{G}$, we have
$$\frac{2GD}{\sqrt{T}}.$$
- This is minimax optimal (up to constant).
- G is independent of ψ thanks to the proximal mapping. Note that $\|\partial\psi(x)\| \leq C\sqrt{p}$ for L_1-reg.

Convergence analysis of SGD (strongly convex)

Assumption

(A1) $\mathrm{E}[\|g_t\|^2] \leq G^2$.

(A3) L_ψ is μ-strongly convex.

Theorem

Let $\bar{x}_T = \frac{1}{T+1} \sum_{t=0}^{T} x_t$. *For* $\eta_t = \frac{1}{\mu t}$, *it holds*

$$\mathrm{E}_{z_{1:T}}[L_\psi(\bar{x}_T) - L_\psi(x^*)] \leq \frac{G^2 \log(T)}{T\mu}.$$

Better than non-strongly convex situation.
But, this is not minimax optimal.
The bound is tight (Rakhlin et al., 2012).

Polynomial averaging for strongly convex risk

Assumption

(A1) $\mathrm{E}[\|g_t\|^2] \leq G^2$.

(A3) L_ψ is μ-strongly convex.

Modify the update rule as

$$x_t = \mathrm{prox}\left(x_{t-1} - \eta_t \frac{t}{t+1} g_t | \eta_t \psi \right),$$

and **take the weighted average** $\bar{x}_T = \dfrac{2}{(T+1)(T+2)} \displaystyle\sum_{t=0}^{T} (t+1) x_t$.

Theorem

For $\eta_t = \frac{2}{\mu t}$, *it holds* $\quad \mathrm{E}_{z_{1:T}}[L_\psi(\bar{x}_T) - L_\psi(x^*)] \leq \dfrac{2G^2}{T\mu}$.

$\log(T)$ is removed.

This is minimax optimal (explained later).

Remark on polynomial averaging

$$\bar{x}_T = \frac{2}{(T+1)(T+2)} \sum_{t=0}^{T} (t+1) x_t$$

$O(T)$ computation? **No**.
$\bar{x}_T$ can be efficiently updated:

$$\bar{x}_t = \frac{t}{t+2} \bar{x}_{t-1} + \frac{2}{t+2} x_t.$$

General step size and weighting policy

Let s_t $(t = 1, 2, \ldots, T+1)$ be a positive sequence such that $\sum_{t=1}^{T+1} s_t = 1$.

$$x_t = \mathrm{prox}\left(x_{t-1} - \eta_t \frac{s_t}{s_{t+1}} g_t | \eta_t \psi\right) \quad (t = 1, \ldots, T)$$

$$\bar{x}_T = \sum_{t=0}^{T} s_{t+1} x_t.$$

Assumption: (A1) $\mathrm{E}[\|g_t\|^2] \leq G^2$, (A2) $\mathrm{E}[\|x_t - x^*\|^2] \leq D^2$, (A3) L_ψ is μ-strongly convex.

Theorem

$$\mathrm{E}_{z_{1:T}}[L_\psi(\bar{x}_T) - L_\psi(x^*)]$$
$$\leq \sum_{t=1}^{T} \frac{s_{t+1}\eta_{t+1}}{2} G^2 + \sum_{t=0}^{T-1} \frac{\max\{\frac{s_{t+2}}{\eta_{t+1}} - s_{t+1}(\frac{1}{\eta_t} + \mu), 0\} D^2}{2}$$

As for $t = 0$, we set $1/\eta_0 = 0$.

Special case

Let the weight proportion to the step size (step size could be seen as **importance**):

$$s_t = \frac{\eta_t}{\sum_{\tau=1}^{T+1} \eta_\tau}.$$

In this setting, the previous theorem gives

$$\mathrm{E}_{z_{1:T}}[L_\psi(\bar{x}_T) - L_\psi(x^*)] \leq \frac{\sum_{t=1}^{T} \eta_t^2 G^2 + D^2}{2\sum_{t=1}^{T} \eta_t}$$

$$\sum_{t=1}^{\infty} \eta_t = \infty$$

$$\sum_{t=1}^{\infty} \eta_t^2 < \infty$$

ensures the convergence.

Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

Stochastic regularized dual averaging (SRDA)

The second assumption $\mathrm{E}[\|x_t - x^*\|^2] \leq D^2$ can be removed by using **dual averaging** (Nesterov, 2009, Xiao, 2009).

SRDA

- Observe $z_t \sim P(Z)$, and let $\ell_t(x) := \ell(z_t, x)$.
- Calculate gradient: $g_t \in \partial_x \ell_t(x_{t-1})$.
- Take the average of the gradients:

$$\bar{g}_t = \frac{1}{t} \sum_{\tau=1}^{t} g_\tau.$$

- Update as

$$\begin{aligned} x_t &= \operatorname*{argmin}_{x \in \mathbb{R}^p} \left\{ \langle \bar{g}_t, x \rangle + \psi(x) + \frac{1}{2\eta_t} \|x\|^2 \right\} \\ &= \operatorname{prox}(-\eta_t \bar{g}_t | \eta_t \psi). \end{aligned}$$

The information of old observations is maintained by taking average of gradients.

Convergence analysis of SRDA

Assumption

(A1) $\mathrm{E}[\|g_t\|^2] \leq G^2$.

(A2) ~~$\mathrm{E}[\|x_t - x^*\|^2] \leq D^2$.~~

Theorem

Let $\bar{x}_T = \frac{1}{T+1}\sum_{t=0}^{T} x_t$. *For* $\eta_t = \eta_0\sqrt{t}$, *it holds*

$$\mathrm{E}_{z_{1:T}}[L_\psi(\bar{x}_T) - L_\psi(x^*)] \leq \frac{\eta_0 G^2 + \|x^* - x_0\|^2/\eta_0}{\sqrt{T}}.$$

- If $\|x^* - x_0\| \leq R$, then for $\eta_0 = \frac{R}{G}$, we have
$$\frac{2RG}{\sqrt{T}}.$$
This is minimax optimal (up to constant).
- The norm of intermediate solution x_t is well controlled. **Thus (A2) is not required**.

Convergence analysis of SRDA (strongly convex)

Assumption

(A1) $\mathrm{E}[\|g_t\|^2] \leq G^2$.

(A2) ~~$\mathrm{E}[\|x_t - x^*\|^2] \leq D^2$~~.

(A3) ψ is μ-strongly convex.

Modify the update rule as

$$\bar{g}_t = \frac{2}{(t+1)(t+2)} \sum_{\tau=1}^{t} \tau g_\tau, \quad x_t = \mathrm{prox}\big(- \eta_t \bar{g}_t | \eta_t \psi \big),$$

and take the weighted average $\bar{x}_T = \frac{2}{(T+1)(T+2)} \sum_{t=0}^{T} (t+1) x_t$.

Theorem

For $\eta_t = (t+1)(t+2)/\xi$, it holds

$$\mathrm{E}_{z_{1:T}}[L_\psi(\bar{x}_T) - L_\psi(x^*)] \leq \frac{\xi \|x^* - x_0\|^2}{T^2} + \frac{2G^2}{T\mu}.$$

$\xi \to 0$ yields $\frac{2G^2}{T\mu}$: minimax optimal.

General convergence analysis

Let the weight $s_t > 0$ $(t = 1, \dots)$ is any positive sequence. We generalize the update rule as

$$\bar{g}_t = \frac{\sum_{\tau=1}^{t} s_\tau g_\tau}{\sum_{\tau=1}^{t+1} s_\tau},$$
$$x_t = \mathrm{prox}(-\eta_t \bar{g}_t | \eta_t \psi) \quad (t = 1, \dots, T).$$

Let the weighted average of $(x_t)_t$ be $\bar{x}_T = \frac{\sum_{\tau=0}^{T} s_{\tau+1} x_\tau}{\sum_{\tau=0}^{T} s_{\tau+1}}$.

Assumption: (A1) $\mathrm{E}[\|g_t\|^2] \leq G^2$, (A3) L_ψ is μ-strongly convex (μ can be 0).

Theorem

Suppse that $\eta_t / (\sum_{\tau=1}^{t+1} s_\tau)$ is non-decreasing, then

$$\mathrm{E}_{z_{1:T}}[L_\psi(\bar{x}_T) - L_\psi(x^*)]$$
$$\leq \frac{1}{\sum_{t=1}^{T+1} s_t} \left(\sum_{t=1}^{T+1} \frac{s_t^2}{2[(\sum_{\tau=1}^{t} s_\tau)(\mu + 1/\eta_{t-1})]} G^2 + \frac{\sum_{t=1}^{T+2} s_t}{2\eta_{T+1}} \|x^* - x_0\|^2 \right).$$

Computational cost and generalization error

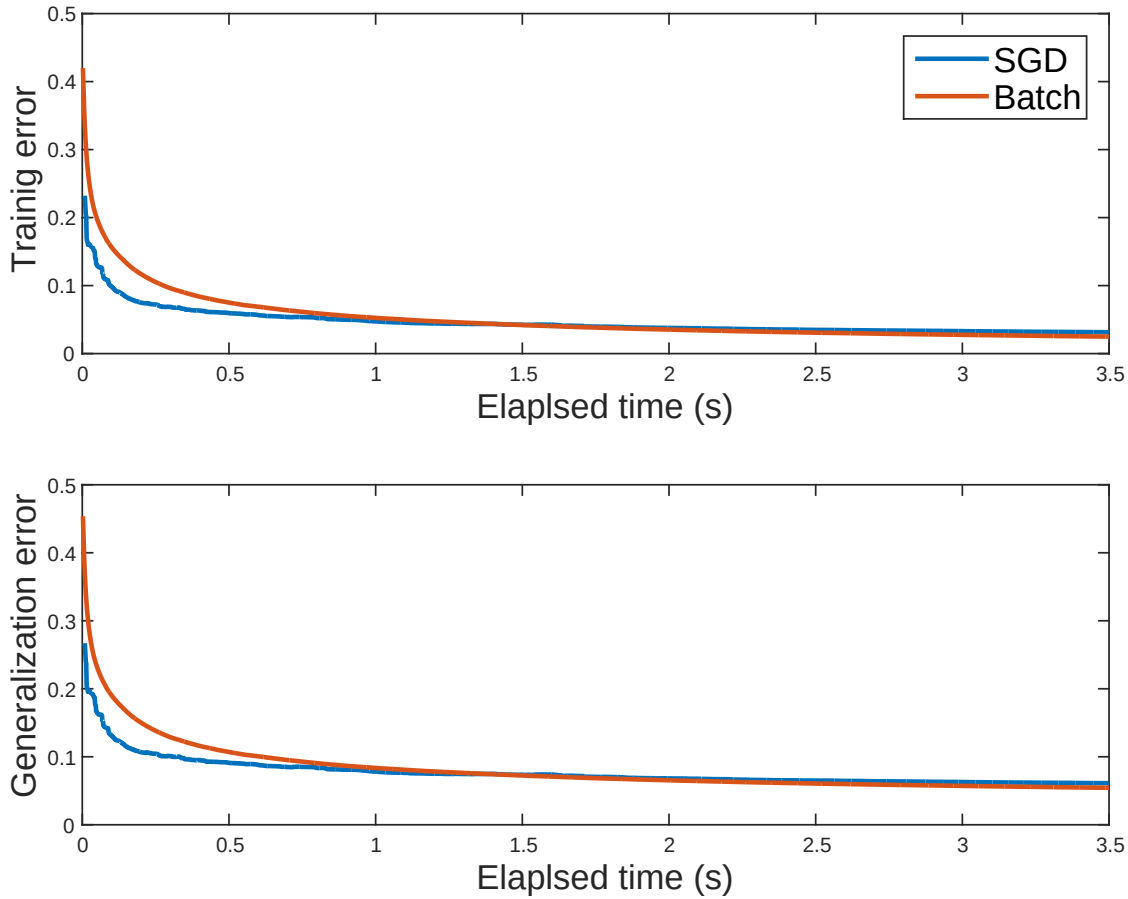
- The optimal learning rate for a strongly convex expected risk (generalization error) is $O(1/n)$ (n is the sample size).
- To achieve $O(1/n)$ generalization error, we need to decrease the training error to $O(1/n)$.

	Normal gradient des.	SGD
Time per iteration	n	1
Number of iterations until ϵ error	$\log(1/\epsilon)$	$1/\epsilon$
Time until ϵ error	$n \log(1/\epsilon)$	$1/\epsilon$
Time until $1/n$ error	$n \log(n)$	n

(Bottou, 2010)

SGD is $O(\log(n))$ faster with respect to the generalization error.

Typical behavior



Normal gradient descent v.s. SGD

Logistic regression with L_1-regularization: $n = 10,000$, $p = 2$.

SGD decreases the objective rapidly, and after a while, the batch gradient method catches up and slightly surpasses.

Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

Changing the metric (divergence)

$$\min_x \quad L(x) + \psi(x)$$

$$x^{(t)} = \operatorname*{argmin}_{x \in \mathbb{R}^p} \left\{ \langle g_t, x \rangle + \psi(x) + \frac{1}{2\eta} \|x - x^{(t-1)}\|^2 \right\}$$

Changing the metric (divergence)

$$\min_x \quad L(x) + \psi(x)$$

$$x^{(t)} = \operatorname*{argmin}_{x \in \mathbb{R}^p} \left\{ \langle g_t, x \rangle + \psi(x) + \frac{1}{2\eta} \|x - x^{(t-1)}\|^2_{H_t} \right\}$$

$\|x\|^2_H := x^\top H x.$

Changing the metric (divergence)

$$\min_x \quad L(x) + \psi(x)$$

$$x^{(t)} = \operatorname*{argmin}_{x \in \mathbb{R}^p} \left\{ \langle g_t, x \rangle + \psi(x) + \frac{1}{2\eta} \|x - x^{(t-1)}\|_{H_t}^2 \right\}$$

$\|x\|_H^2 := x^\top H x$.

Choice of H_t

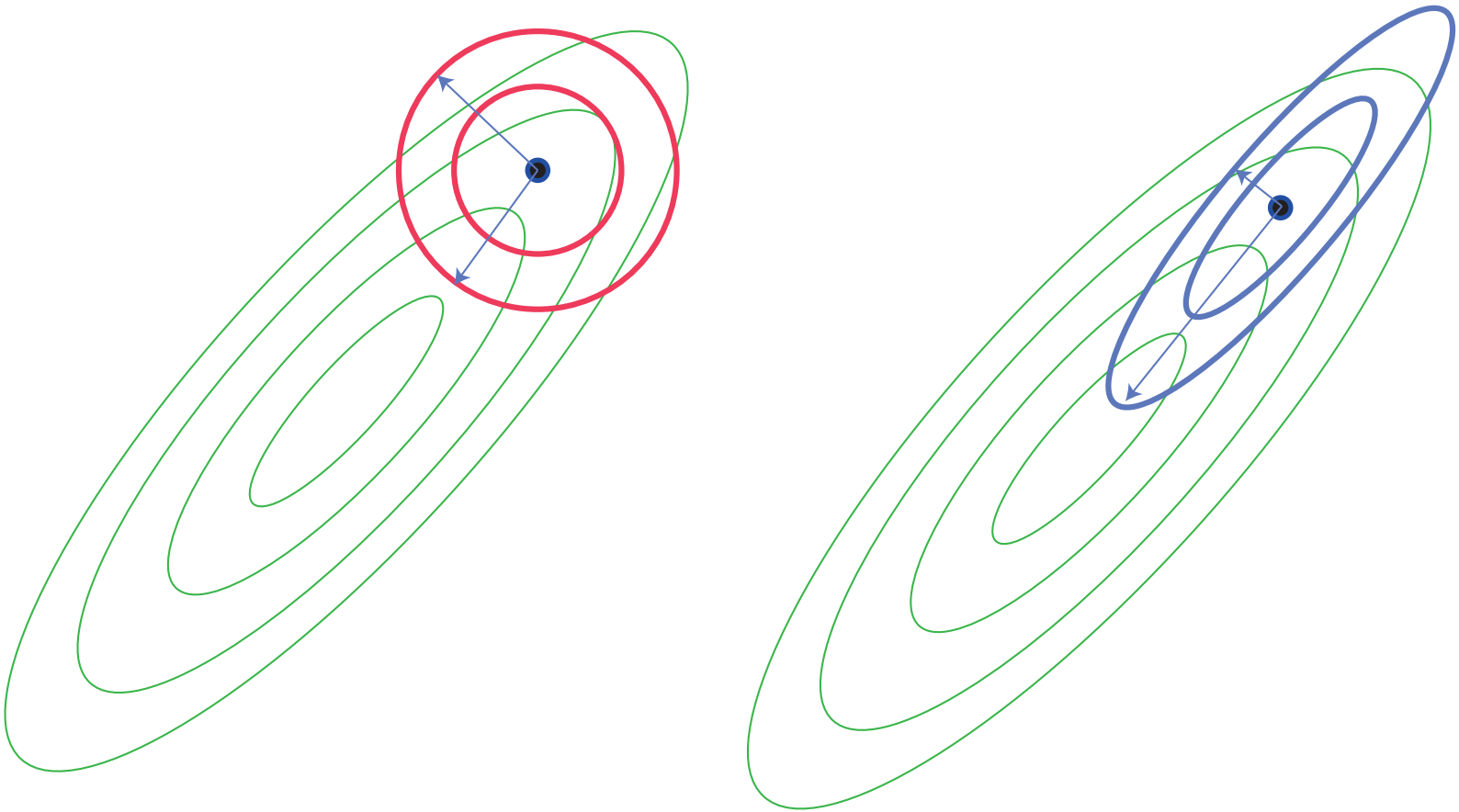
- Hessian $H_t = \nabla \nabla^\top L(x^{(t-1)})$: **Newton method**
- Fisher information matrix $H_t = \mathrm{E}_{Z|x^{(t-1)}}[-\nabla_x \nabla_x^\top p_x(Z)|_{x=x^{(t-1)}}]$: **Natural gradient**
  (x is a parameter of a parametric model $\{p_x\}_x$)

---

c.f. Bregman divergence.

$$B_\phi(x||x') := \phi(x) - \phi(x') - \langle \nabla \phi(x'), x - x' \rangle.$$

$\rightarrow$ Mirror descent



AdaGrad (Duchi et al. (2011))

Let

$$H_t = G_t^{\frac{1}{2}} + \delta I$$

for some $\delta \geq 0$, where G_t is either of the followings:

$$\text{(Full)} \qquad G_t = \sum_{\tau=1}^{t} g_\tau g_\tau^\top,$$

$$\text{(Diag)} \qquad G_t = \operatorname{diag}\left(\sum_{\tau=1}^{t} g_\tau g_\tau^\top\right).$$

AdaGrad stretches flat directions and shrinks steep directions.

- Ada-SGD:

$$x^{(t)} = \operatorname*{argmin}_{x \in \mathbb{R}^p} \left\{ \langle g_t, x \rangle + \psi(x) + \frac{1}{2\eta} \|x - x^{(t-1)}\|_{H_t}^2 \right\}.$$

- Ada-SRDA: for $\bar{g}_t = \frac{1}{t} \sum_{\tau=1}^{t} g_\tau$,

$$x^{(t)} = \operatorname*{argmin}_{x} \left\{ \langle \bar{g}_t, x \rangle + \psi(x) + \frac{1}{2t\eta} \|x\|_{H_t}^2 \right\}.$$

Analysis of AdaGrad

Theorem

Let $q = 2$ for FULL, and $q = \infty$ for Diag. Define the regret as

$$Q(T) := \frac{1}{T}\sum_{t=1}^{T}\left(\ell_{t+1}(x^{(t)}) + \psi(x^{(t)}) - \ell_t(\beta^*) - \psi(\beta^*)\right).$$

- *Ada-SGD:* $\forall \delta \geq 0$,

$$Q(T) \leq \frac{\delta}{T\eta}\|x^*\|_2^2 + \frac{\max_{t\leq T}\{\|x^* - x^{(t)}\|_q^2\}/\eta + 2\eta}{2T}\mathrm{tr}\left[G_T^{1/2}\right].$$

- *Ada-SRDA: for* $\delta \geq \max_t \|g_t\|_2$,

$$Q(T) \leq \frac{\delta}{T\eta}\|\beta^*\|_2^2 + \frac{\|x^*\|_q^2/\eta + 2\eta}{2T}\mathrm{tr}\left[G_T^{1/2}\right].$$

Analysis of AdaGrad

Suppose

- The gradient is **unbalanced**:

$$|g_{t,j}|^2 \leq Gj^{-2} \quad (j = 1, \ldots, p, \ \forall t).$$

(Ada-SGD) $$\mathrm{E}[L(x^{(T)}) - L(x^*)] \leq C\frac{\log(p)}{\sqrt{T}}$$

(ordinary SGD) $$\mathrm{E}[L(x^{(T)}) - L(x^*)] \leq C\frac{G\mathrm{E}[\max_t \|x^{(t)}\|]}{\sqrt{T}} \leq C\frac{\sqrt{p}}{\sqrt{T}}$$

$$\sqrt{p} \quad \to \quad \log(p)$$

Much improvement.

AdaGrad is used in various applications including sparse learning and **deep learning**.

In deep learning, we often encounter a phenomenon called plateau, that is, we are stuck in a flat region.

It is hard to get out from plateau by standard SGD. AdaGrad adaptively adjust the search space to get out of plateau.

AdaGrad is one of the standard optimization methods for deep learning. Related methods: AdaDelta (Zeiler, 2012), RMSProp (Tieleman and Hinton, 2012), Adam (Kingma and Ba, 2014).

Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

Nesterov's acceleration of SGD

Assumption:

- the expected loss $L(x)$ is γ-smooth.
- the variance of gradient is bounded by σ^2:

$$\mathrm{E}_Z[\|\nabla_\beta \ell(Z,\beta) - \nabla L(\beta)\|^2] \leq \sigma^2.$$

$\rightarrow$ combining with Nesterov's acceleration, the convergence can be got faster.

- Acceleration for SGD: Hu et al. (2009)
- Accleration for SRDA: Xiao (2010), Chen et al. (2012)
- General method and analysis (including non-convex): Lan (2012), Ghadimi and Lan (2012, 2013)

$$\mathrm{E}_{z_{1:T}}[L_\psi(x^{(T)})] - L_\psi(x^*) \leq C\left(\frac{\sigma D}{\sqrt{T}} + \frac{D^2\gamma}{T^2}\right)$$

(D is the diameter: $\mathrm{E}[\|x^{(t)} - x^*\|^2] \leq D^2$ $(\forall t)$)

Speed up of accelerated SGD

$$\mathrm{E}_{z_{1:T}}[L_\psi(x^{(T)})] - L_\psi(x^*) \leq C\left(\frac{\sigma D}{\sqrt{T}} + \frac{D^2\gamma}{T^2}\right)$$

σ^2 is the variance of the gradient estimate:

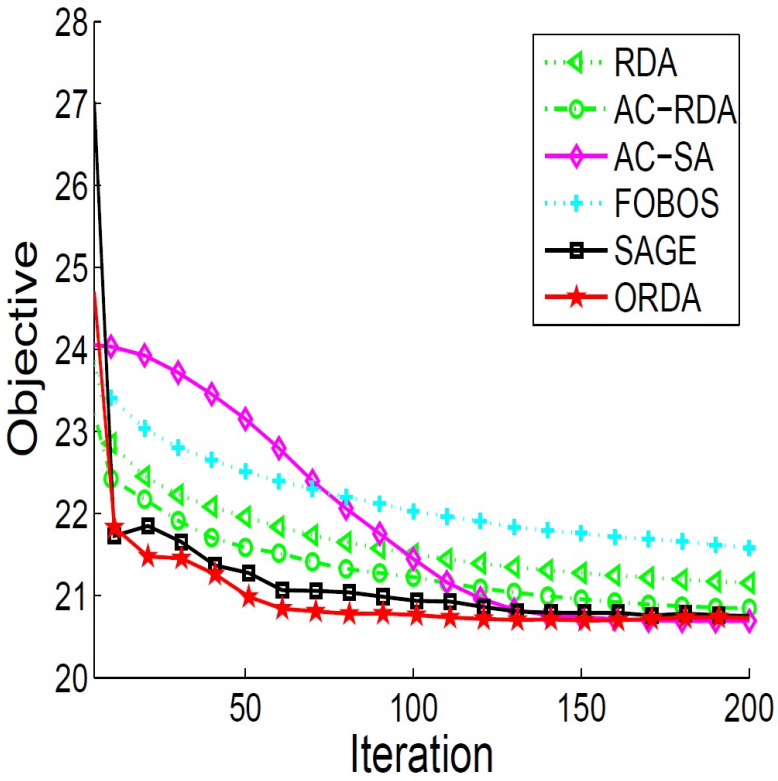
$$\mathrm{E}_Z[\|\nabla_\beta \ell(Z,\beta) - \nabla L(\beta)\|^2] \leq \sigma^2.$$

The variance can be reduced by simply taking average:

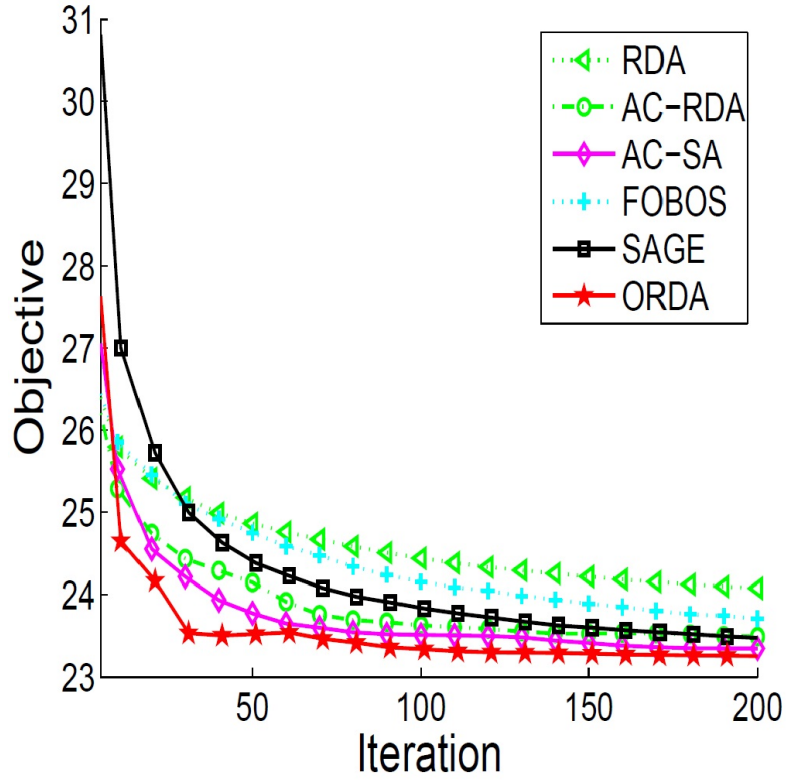
$$g = \nabla\ell(z, x^{(t-1)}) \quad \Rightarrow \quad g = \frac{1}{K}\sum_{k=1}^{K} \nabla\ell(z_k, x^{(t-1)})$$

(Variance) σ^2 σ^2/K

- Computing independent gradients can be parallelized.
- As $\sigma \to 0$, the bound goes to $O(1/T^2)$: non-stochastic Nesterov's acceleration.



(a) Objective for L_1



(b) Objective for Elastic-net

Numerical comparison on synthetic data with (a) L_1 regularization (Lasso) and (b) Elastic-net regularization (figure is from Chen et al. (2012)).

SAGE: Accelerated SGD (Hu et al., 2009), AC-RDA: Accelerated stochastic RDA (Xiao, 2010), AC-SA: Accelerated stochastic approximation Ghadimi and Lan (2012), ORDA: Optimal stochastic RDA (Chen et al., 2012)

Accelerated SA for strongly convex objective

Assumption: Objective is μ-strongly convex and γ-smooth.

Accelerated stochastic approximation: Hu et al. (2009), Ghadimi and Lan (2012)

$$\mathrm{E}_{z_{1:T}}[L_\psi(x^{(T)})] - L_\psi(x^*) \leq C\left(\frac{\sigma^2}{\mu T} + \frac{\gamma R^2}{T^2}\right).$$

Multi-stage accelerated stochastic approximation: Chen et al. (2012), Ghadimi and Lan (2013)

$$\mathrm{E}_{z_{1:T}}[L_\psi(x^{(T)})] - L_\psi(x^*) \leq C\left(\frac{\sigma^2}{\mu T} + \exp\left(-C\sqrt{\frac{\mu}{\gamma}}T\right)\right)$$

$\sigma = 0$ gives the batch optimal rate.

Summary of convergence rates

- Online methods (expected risk minimization):
  - $\frac{GR}{\sqrt{T}}$ (non-smooth, non-strongly convex) Polyak-Ruppert averaging
  - $\frac{G^2}{\mu T}$ (non-smooth, strongly convex) Polynomial averaging
  - $\frac{\sigma R}{\sqrt{T}} + \frac{R^2 L}{T^2}$ (smooth, non-strongly convex) Acceleration
  - $\frac{\sigma^2}{\mu T} + \exp\left(-\sqrt{\frac{\mu}{L}} T\right)$ (smooth, strongly convex) Acceleration

---

G: upper bound of norm of gradient, R: diameter of the domain,
L: smoothness, μ: strong convexity, σ: variance of the gradient

Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

Minimax optimal rate of stochastic first order methods

$$\min_{x \in \mathcal{B}} L(x) = \min_{x \in \mathcal{B}} \mathrm{E}_Z[\ell(Z, x)]$$

Condition

- $\hat{g}_x \in \partial_x \ell(Z, x)$ is bounded as $\|\mathrm{E}[\hat{g}_x]\| \leq G$ $(\forall x \in \mathcal{B})$.
- The domain $\mathcal{B}$ contains a ball with radius R.
- $L(x)$ is μ-strongly convex ($\mu = 0$ is allowed).

Theorem (Minimax optimality (Agarwal et al., 2012, Nemirovsky and Yudin, 1983))

For any first order algorithm, there exist loss function ℓ and distribution $P(Z)$ satisfying the assumption on which the algorithm must suffer

$$\mathrm{E}[L(x^{(T)}) - L(x^*)] \geq c \min\left\{\frac{GR}{\sqrt{T}}, \frac{G^2}{\mu T}, \frac{GR}{\sqrt{p}}\right\}.$$

SGD and SRDA achieve this optimal rate.

First order algorithm: an algorithm that depends on only the loss and its gradient $(\ell(Z, x), \hat{g}_x)$ for a query point x. (SGD, SRDA are included.)

Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

From expectation to finite sum

Online:

$$P(x) = \mathrm{E}[\ell(Z, x)] = \int \ell(Z, x) \mathrm{d}P(Z)$$

$$\downarrow$$

Batch:

$$P(x) = \frac{1}{n} \sum_{i=1}^{n} \ell(z_i, x)$$

From online to batch

In the batch setting, the data are fixed. We just minimize the objective function defined by

$$P(x) = \frac{1}{n} \sum_{i=1}^{n} \ell_i(x) + \psi(x).$$

We construct a method that

- uses few observations per iteration (like online method),
- converges linearly (unlike online method):

$$T > (n + \gamma/\lambda) \log(1/\epsilon)$$

to achieve ϵ accuracy for γ-smooth loss and λ-strongly convex regularization.

Three methods that must be remembered

- **Stochastic Average Gradient descent, SAG** (Le Roux et al., 2012, Schmidt et al., 2013, Defazio et al., 2014)
- **Stochastic Variance Reduced Gradient descent, SVRG** (Johnson and Zhang, 2013, Xiao and Zhang, 2014)
- **Stochastic Dual Coordinate Ascent, SDCA** (Shalev-Shwartz and Zhang, 2013a)

- SAG and SVRG are methods performed on the primal.
- SDCA is on the dual.

Assumptions

$$P(x) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell_i(x)}_{\text{smooth}} + \underbrace{\psi(x)}_{\text{strongly convex}}$$

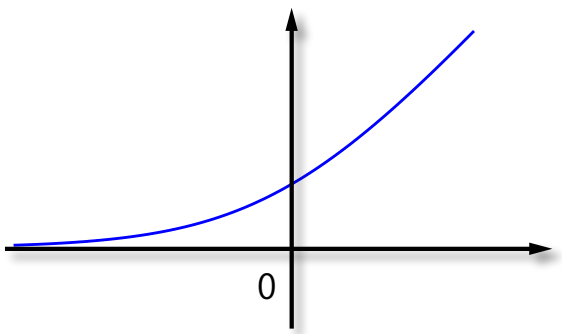
Assumption:

- ℓ_i: Loss is γ-smooth.
- ψ: reg func is λ-strongly convex. Typically $\lambda = O(1/n)$ or $O(1/\sqrt{n})$.

Example:

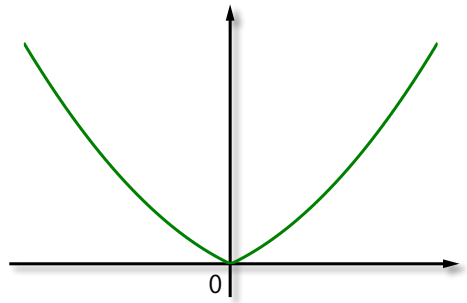
Loss function

- smoothed hinge loss
- logistic loss



Regularization function

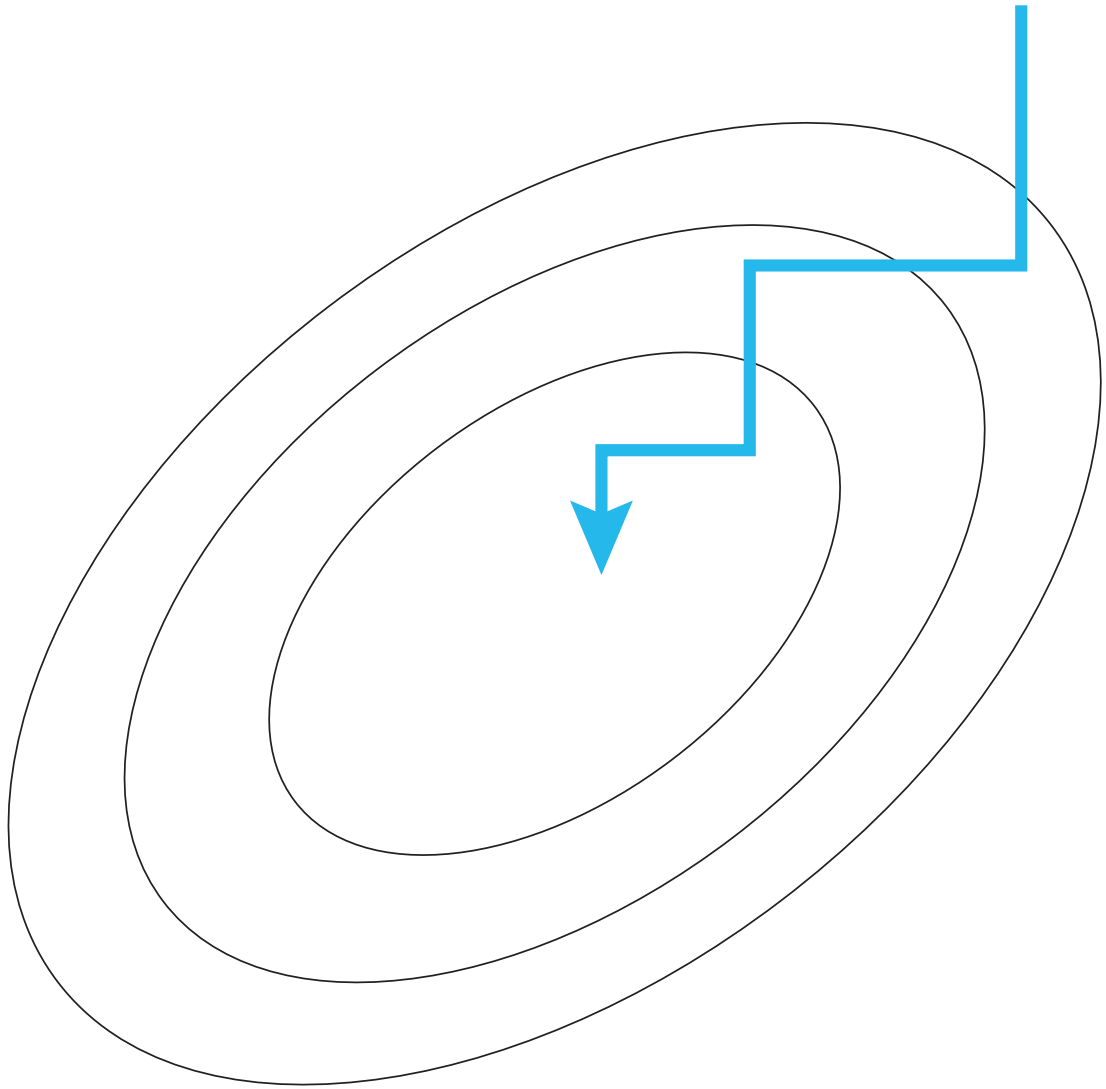
- L_2 regularization
- Elastic net regularization
- $\tilde{\psi}(x) + \lambda\|x\|^2$ (with small h)



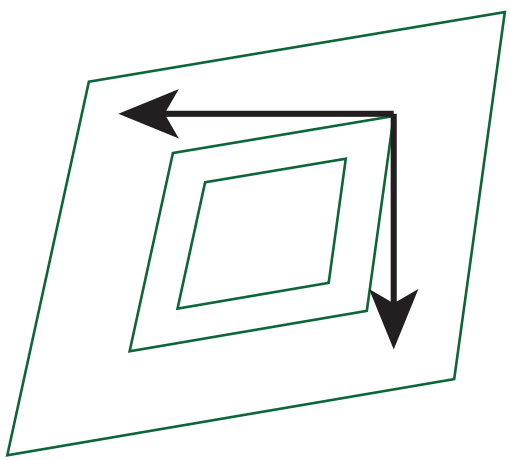
Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

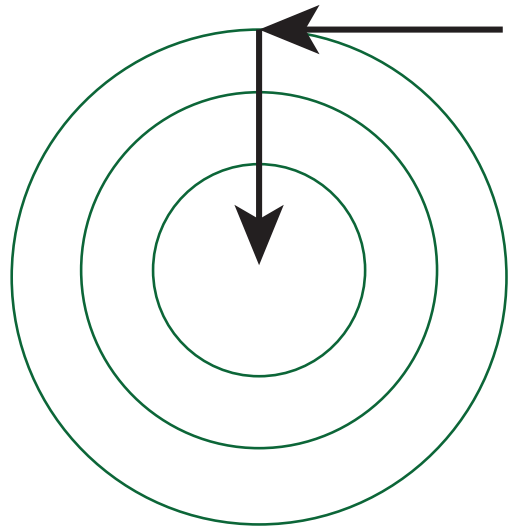
Coordinate Descent



Note on CD



failure



success

- Left hand side: CD fails. No descent direction.
- To make CD success, the objective should have descent direction. Ideally, separable $f(x) = \sum_{j=1}^{p} f_j(x_j)$.

Coordinate descent in primal

$$\min_x\{P(x)\} = \min_x\{f(x) + \psi(x)\} = \min_x\{f(x) + \sum_{j=1}^{p} \psi_j(x_j)\}$$

Coordinate descent (sketch)

1. Choose $j \in \{1, \ldots, p\}$ in some way. (typically, random choice)
2. j-th coordinate x_j is updated so that the objective is decreased,

- Usually a block of coordinates are updated instead of one coordinate (block coordinate descent).

Coordinate descent in primal

$$\min_x\{P(x)\} = \min_x\{f(x) + \psi(x)\} = \min_x\{f(x) + \sum_{j=1}^{p} \psi_j(x_j)\}$$

Coordinate descent (sketch)

1. Choose $j \in \{1, \ldots, p\}$ in some way. (typically, random choice)
2. j-th coordinate x_j is updated so that the objective is decreased, e.g.,
   - $x_j^{(t)} \leftarrow \mathrm{argmin}_{x_j} P(x_1^{(t-1)}, \ldots, x_j, \ldots, x_p^{(t-1)})$,
     or
   - for $g_j = \frac{\partial f(x^{(t)})}{\partial x_j}$
     $x_j^{(t+1)} \leftarrow \mathrm{argmin}_{x_j} \quad \langle g_j, x_j \rangle + \psi_j(x_j) + \frac{1}{2\eta_t}\|x_j - x_j^{(t-1)}\|^2$.

- Usually a block of coordinates are updated instead of one coordinate (block coordinate descent).

Convergence of primal CD method

We consider a separable regularization:

$$\min_x\{P(x)\} = \min_x\{f(x) + \psi(x)\} = \min_x\{f(x) + \sum_{j=1}^{p} \psi_j(x_j)\}.$$

Assumption: f is γ-smooth ($\|\nabla f(x) - \nabla f(x')\| \leq \gamma\|x - x'\|$)

- Cyclic (Saha and Tewari, 2013, Beck and Tetruashvili, 2013)

$$P(x^{(t)}) - R(x^*) \leq \frac{\gamma p\|x^{(0)} - x^*\|^2}{2t} = O(1/t) \text{ (with isotonicity).}$$

- Random choice (Nesterov, 2012, Richtárik and Takáč, 2014)
  - No acceleration: $O(1/t)$.
  - Nesterov's acceleration: $O(1/t^2)$ (Fercoq and Richtárik, 2013).
  - f is α-strongly convex: $O(\exp(-C(\alpha/\gamma)t))$.
  - f is α-strongly conv + acceleration: $O(\exp(-C\sqrt{\alpha/\gamma}t))$ (Lin et al., 2014).

Nice review is given by Wright (2015).

Stochastic Dual Coordinate Ascent, SDCA

Suppose that $\exists f_i : \mathbb{R} \to \mathbb{R}$ such that $\ell(z_i, x) = f_i(a_i^\top x)$.
Let $A = [a_1, \ldots, a_n]$.

(Primal)
$$\inf_{x \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i(a_i^\top x) \; + \; \psi(x) \right\}$$

Stochastic Dual Coordinate Ascent, SDCA

Suppose that $\exists f_i : \mathbb{R} \to \mathbb{R}$ such that $\ell(z_i, x) = f_i(a_i^\top x)$.
Let $A = [a_1, \ldots, a_n]$.

(Primal)
$$\inf_{x \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^n f_i(a_i^\top x) \; + \; \psi(x) \right\}$$

[Fenchel's duality theorem]

$$\inf_{x \in \mathbb{R}^p} \{ f(A^\top x) + n\psi(x) \} = - \inf_{y \in \mathbb{R}^n} \{ f^*(y) + n\psi^*(-Ay/n) \}$$

(Dual)
$$\inf_{y \in \mathbb{R}^n} \left\{ \frac{1}{n} \sum_{i=1}^n f_i^*(y_i) \; + \; \psi^* \left(-\frac{1}{n} Ay \right) \right\}$$

We used the following facts:

- For $f(\alpha) = \sum_{i=1}^n f_i(\alpha_i)$, we have $f^*(\beta) = \sum_{i=1}^n f_i^*(\beta_i)$.
- For $\tilde{\psi}(x) = n\psi(x)$, we have $\tilde{\psi}^*(y) = n\psi^*(y/n)$.

Remarks

$$\sup_{y \in \mathbb{R}^n} \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i^*(y_i) \; + \; \psi^* \left(-\frac{1}{n} A y \right) \right\}$$

- The dual loss term $\sum_{i=1}^n f_i^*(y_i)$ is separable.
- Each coordinate y_i affects the objective through only the i-th data:

$$f_i^*(y_i),$$
$$\psi^* \left(-\frac{1}{n}(a_1 y_1 + \cdots + a_i y_i + \cdots + a_n y_n) \right).$$

$\rightarrow$ **Coordinate descent behaves like online methods!**

- The loss f_i is smooth $\Leftrightarrow$ f_i^* is strongly convex.
- The reg func ψ is strongly convex $\Leftrightarrow$ ψ^* is smooth.

Algorithm of SDCA

SDCA (Shalev-Shwartz and Zhang, 2013a)

Iterate the following for $t = 1, 2, \ldots$

1. Pick up an index $i \in \{1, \ldots, n\}$ uniformly at random.
2. Update the i-th coordinate y_i so that the objective function is decreased.

Algorithm of SDCA

SDCA (Shalev-Shwartz and Zhang, 2013a)

Iterate the following for $t = 1, 2, \ldots$

1. Pick up an index $i \in \{1, \ldots, n\}$ uniformly at random.
2. Update the i-th coordinate y_i:
   (let $A_{\setminus i} = [a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n]$, and $y_{\setminus i} = (y_j)_{j \neq i}$)

   - $$y_i^{(t)} \in \operatorname*{argmin}_{y_i \in \mathbb{R}} \left\{ f_i^*(y_i) + n\psi^*\left(-\frac{1}{n}(a_i y_i + A_{\setminus i} y_{\setminus i}^{(t-1)}) \right) + \frac{1}{2\eta}\|y_i - y_i^{(t-1)}\|^2 \right\},$$
   - $y_j^{(t)} = y_j^{(t-1)}$ (for $j \neq i$).

Algorithm of SDCA

SDCA (linearized version) (Shalev-Shwartz and Zhang, 2013a)

Iterate the following for $t = 1, 2, \ldots$

1. Pick up an index $i \in \{1, \ldots, n\}$ uniformly at random.
2. Calculate $x^{(t-1)} = \nabla \psi^*(-Ay^{(t-1)}/n)$.
3. Update the i-th coordinate y_i:

   - $y_i^{(t)} \in \underset{y_i \in \mathbb{R}}{\operatorname{argmin}} \left\{ f_i^*(y_i) - \langle x^{(t-1)}, a_i y_i \rangle + \frac{1}{2\eta} \|y_i - y_i^{(t-1)}\|^2 \right\}$

   - $y_j^{(t)} = y_j^{(t-1)} \quad (\text{for } j \neq i)$.

- If the reg func ψ is λ-strongly covnex, ψ^* is $1/\lambda$-smooth and thus differentiable: $x^{(t)} = \nabla \psi^*(-Ay^{(t)}/n)$.
- $x^{(t)}$ is actually the primal variable.
- Computational complexity per iteration is same as online methods!
- Important relation: $\operatorname{prox}(q|g^*) = q - \operatorname{prox}(q|g)$. primal!

Algorithm of SDCA

SDCA (linearized version) (Shalev-Shwartz and Zhang, 2013a)

Iterate the following for $t = 1, 2, \ldots$

1. Pick up an index $i \in \{1, \ldots, n\}$ uniformly at random.
2. Calculate $x^{(t-1)} = \nabla\psi^*(-Ay^{(t-1)}/n)$.
3. Update the i-th coordinate y_i:

   - $$y_i^{(t)} \in \operatorname*{argmin}_{y_i \in \mathbb{R}} \left\{ f_i^*(y_i) - \langle x^{(t-1)}, a_i y_i \rangle + \frac{1}{2\eta} \|y_i - y_i^{(t-1)}\|^2 \right\}$$
     $$= \mathrm{prox}(y_i^{(t-1)} + \eta a_i^\top x^{(t-1)} | \eta f_i^*),$$
   - $y_j^{(t)} = y_j^{(t-1)}$ (for $j \neq i$).

- If the reg func ψ is λ-strongly covnex, ψ^* is $1/\lambda$-smooth and thus differentiable: $x^{(t)} = \nabla\psi^*(-Ay^{(t)}/n)$.
- $x^{(t)}$ is actually the primal variable.
- Computational complexity per iteration is same as online methods!
- Important relation: $\mathrm{prox}(q|g^*) = q - \mathrm{prox}(q|g)$. primal!

Convergence analysis of SDCA

Assuption:

- f_i **is** γ**-smooth.**
- ψ **is** λ**-strongly convex.**

Theorem

Suppose there exists R *such that* $\|a_i\| \leq R$*. Then, for* $\eta = \lambda n / R^2$*, we have*

$$\mathrm{E}[P(x^{(T)}) + D(y^{(T)})] \leq \left(n + \frac{R^2\gamma}{\lambda}\right) \exp\left(-\frac{T}{n + \frac{R^2\gamma}{\lambda}}\right) (D(y^{(0)}) - D(y^*)).$$

$\mathrm{E}[\cdot]$ *is taken w.r.t. the choice of coordinates.*

- **Linear convergence!**
- Required number of iterations to achieve ϵ:

$$T \geq C\left(n + \frac{R^2\gamma}{\lambda}\right) \log\left((n + \gamma/\lambda)/\epsilon\right).$$

Comparison with the non-stochastic method

How much computation is required to achieve $\mathrm{E}[P(x^{(T)}) - P(x^*)] \leq \epsilon$?
Let $\kappa = \gamma/\lambda$ (condition number).

- SDCA:

$$(n + \kappa) \log\left((n + \kappa)/\epsilon\right)$$

$$\Omega((n + \kappa) \log(1/\epsilon)) \text{ iterations} \quad \times \quad \Omega(1) \text{ per iteration}$$

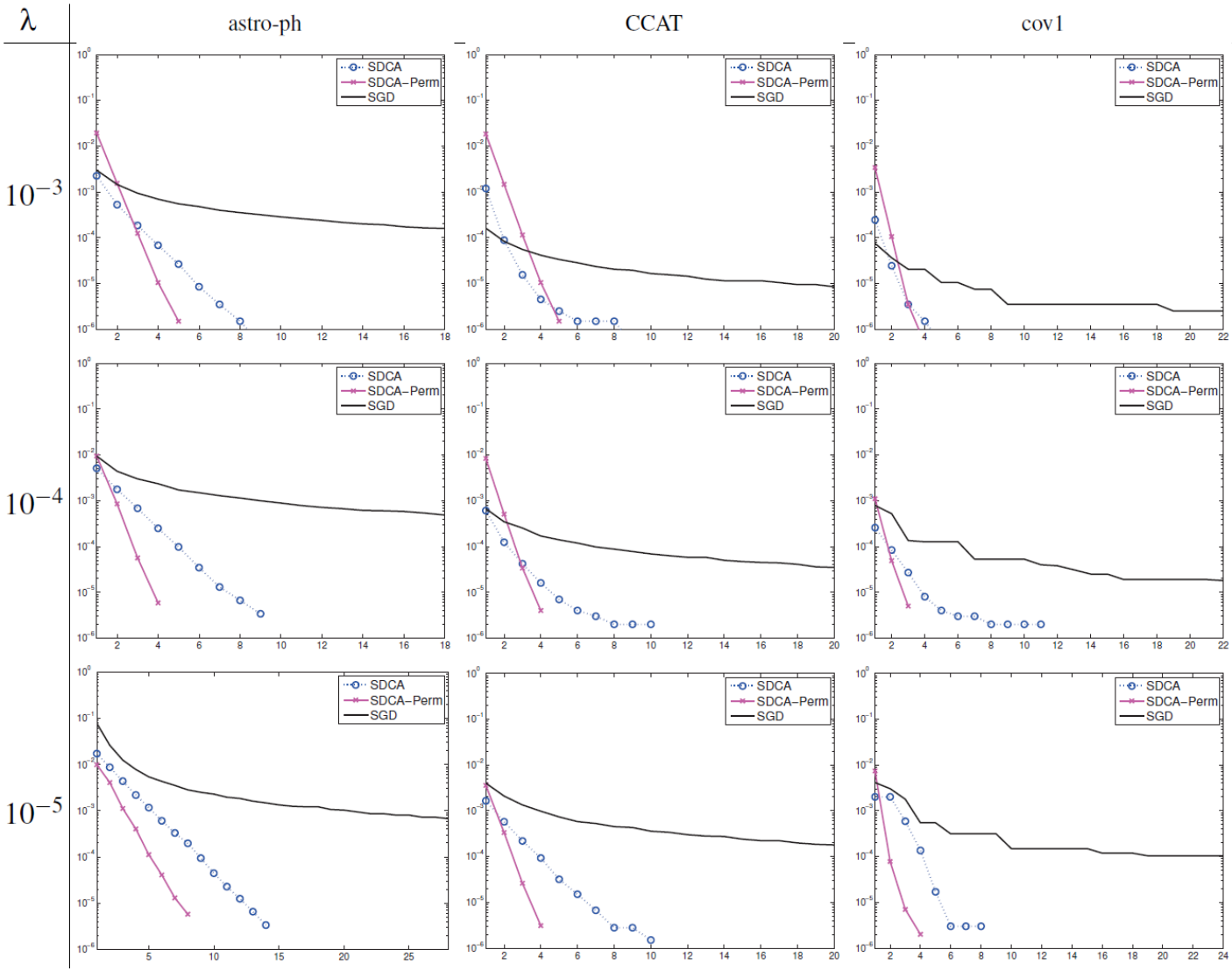
- Non-stochastic first order method:

$$n\kappa \log(1/\epsilon)$$

$$\Omega(\kappa \log(1/\epsilon)) \text{ iterations} \quad \times \quad \Omega(n) \text{ per iteration}$$

Sample size $n = 100,000$, reg param $\lambda = 1/1000$, smoothness $\gamma = 1$:

$$n \times \kappa = 10^8, \quad n + \kappa = 10^5.$$



Numerical comparison between SDCA, SDCA-perm (randomly shuffled cyclic), SGD (figure is from Shalev-Shwartz and Zhang (2013a)).

Nesterov's acceleration of SDCA

Accelerated SDCA (Lin et al., 2014)

Set $\alpha = \frac{1}{n}\sqrt{\frac{\lambda}{\gamma}}$.

1. $y^{(t)} = \frac{\bar{y}^{(t-1)} + \alpha w^{(t-1)}}{1+\alpha}$
2. Pick up an index $i \in \{1, \ldots, n\}$ uniformly at random.
3. Calculate $x^{(t-1)} = \nabla\psi^*(-Ay^{(t-1)}/n)$.
4. Update the i-th coordinate:
   - $w_i^{(t)} \in \operatorname{argmin}_{w_i \in \mathbb{R}} \left\{ f_i^*(w_i) - \langle x^{(t-1)}, a_i w_i \rangle + \frac{\alpha n}{2\gamma} \| w_i - y_i^{(t)} - (1-\alpha) w_i^{(t-1)} \|^2 \right\}$
   - $w_j^{(t)} = (1-\alpha) w_j^{(t-1)} + y_i^{(t)} \quad (\text{for } j \neq i)$.
5. $\bar{y}_i^{(t)} = y_i^{(t)} + n\alpha(w_i^{(t)} - (1-\alpha) w_i^{(t-1)} - y_i^{(t)})$,
   $\bar{y}_j^{(t)} = y_j^{(t)} \quad (\text{for } j \neq i)$.

Shalev-Shwartz and Zhang (2014) also proposed a double-loop acceleration

Convergence of accelerated SDCA

$\|a_i\| \leq R$ $(\forall i)$ $\|A\|$: spectral norm of A

Theorem

Convergence of acc. SDCA If

$$T \geq \left(n + \sqrt{\frac{\gamma n R^2}{\lambda}} \right) \log \left(\frac{C\gamma \|A\|_2^2}{\lambda n \epsilon} \right),$$

then

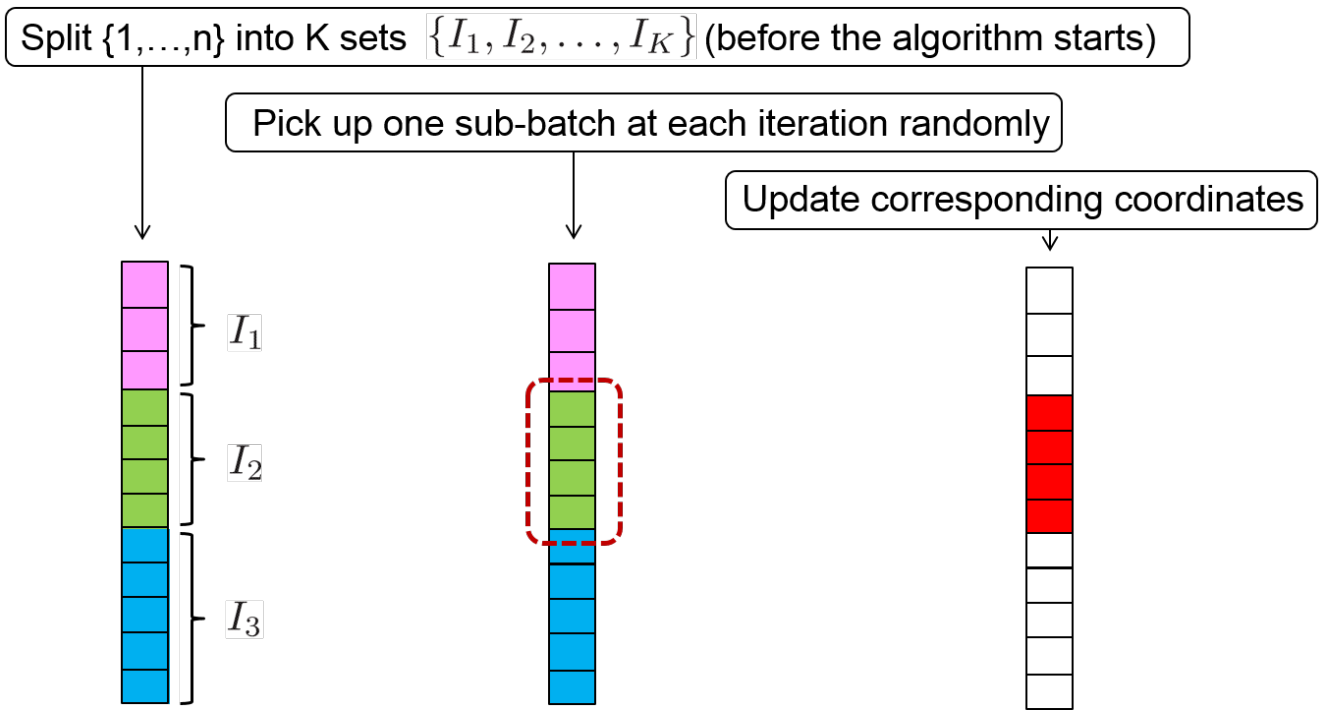
$$\text{(Duality gap)} \qquad \mathrm{E}[P(x^{(T)}) - D(y^{(T)})] \leq \epsilon.$$

$$(\textit{normal}) \qquad \left(n + \frac{\gamma}{\lambda} \right) \log((n+\kappa)/\epsilon)$$

$$(\textit{accelerated}) \qquad \left(n + \sqrt{\frac{\gamma n}{\lambda}} \right) \log((n+\kappa)/\epsilon)$$

Mini-batch SDCA

Instead of choosing one coordinate y_i, we may choose a block of coordinates y_I where $I \subseteq \{1, \ldots, n\}$.
Typically, $\{1, \ldots, n\}$ is divided into K equally sized groups: $I_1, \ldots, I_K$ s.t. $|I_k| = n/K$, $\bigcup_k I_k = \{1, \ldots, n\}$, $I_k \cap I_{k'} = \emptyset$.



Mini-batch technique (Takáč et al., 2013, Shalev-Shwartz and Zhang, 2013b).
If $K = n$, we observe only one data at each iteration.

Mini-batch SDCA

Mini-batch SDCA (stochastic block coordinate descent)

For $t = 1, 2, \ldots,$, iterate the following:

1. Randomly pick up a mini-batch $I \subseteq \{1, \ldots, n\}$ so that $P(i \in I) = 1/K$ $(\forall i)$.
2. $x^{(t-1)} = \nabla\psi^*(-Ay^{(t-1)}/n)$.
3. Update $y^{(t)}$ as
   - $y_I^{(t)} \in \underset{y_i \ (i \in I)}{\operatorname{argmin}} \left\{ \sum_{i=1}^{|I|} f_i^*(y_i) - \langle x^{(t-1)}, A_I y_I \rangle + \frac{1}{2\eta} \|y_I - y_I^{(t-1)}\|^2 \right\}$,
   - $y_i^{(t)} = y_i^{(t-1)} \quad (i \notin I)$.

The update of y_i can be parallelized:

$$y_i = \operatorname{prox}(y_i^{(t-1)} + \eta a_i^\top x^{(t-1)} | \eta f_i^*) \quad (i \in I).$$

Mini-batch SDCA

Mini-batch SDCA (stochastic block coordinate descent)

For $t = 1, 2, \ldots,$, iterate the following:

1. Randomly pick up a mini-batch $I \subseteq \{1, \ldots, n\}$ so that $P(i \in I) = 1/K$ $(\forall i)$.
2. $x^{(t-1)} = \nabla\psi^*(-Ay^{(t-1)}/n)$.
3. Update $y^{(t)}$ as
   - $y_I^{(t)} \in \underset{y_i \ (i \in I)}{\operatorname{argmin}} \left\{ \sum_{i=1}^{|I|} \left[f_i^*(y_i) - \langle x^{(t-1)}, A_i y_i \rangle + \frac{1}{2\eta} \|y_i - y_i^{(t-1)}\|^2 \right] \right\}$,
   - $y_i^{(t)} = y_i^{(t-1)} \quad (i \notin I)$.

The update of y_i can be parallelized:

$$y_i = \operatorname{prox}(y_i^{(t-1)} + \eta a_i^\top x^{(t-1)} | \eta f_i^*) \quad (i \in I).$$

Convergence of mini-batch SDCA

Assuption:

- f_i **is** γ**-smooth.**
- ψ **is** λ**-strongly convex.**

Theorem

Suppose there exists R *such that* $\|A_I^\top A_I\| \leq R^2$ $(\forall I)$. *Then, for* $\eta = \lambda n / R^2$, *we have*

$$\mathrm{E}[P(\bar{x}^{(T)}) - D(\bar{y}^{(T)})] \leq \left(K + \frac{R^2\gamma}{\lambda}\right) \exp\left(-\frac{T}{K + \frac{R^2\gamma}{\lambda}}\right) (D(y^{(0)}) - D(y^*)).$$

$\mathrm{E}[\cdot]$ *is taken w.r.t. the choice of coordinates.*

$$T \geq C\left(K + \frac{R^2\gamma}{\lambda}\right) \log((n+\kappa)/\epsilon)$$

achieves ϵ accuracy. $\rightarrow$ iteration complexity is improved (if R^2 is not large and parallelization is used).

Outline

Primal methods

The key idea: reduce the variance of gradient estimate.

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell_i(x)}_{\text{How to approximate this?}} \qquad +\psi(x)$$

Primal methods

The key idea: reduce the variance of gradient estimate.

$$\underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell_i(x)}_{\text{How to approximate this?}} \qquad +\psi(x)$$

$\langle g, x \rangle$

Online method: pick up $\hat{i} \in \{1, \ldots, n\}$ randomly, and use linear approximation.

$$g = \nabla \ell_{\hat{i}}(x) \quad \Rightarrow \quad \mathrm{E}[g] = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \nabla \ell_i(x)$$

This is an unbiased estimator of the full gradient.

How about variance?

$\rightarrow$ **Variance is the problem!**

$\rightarrow$ In the batch setting, it is easy to reduce the variance.

Stochastic Variance Reduced Gradient descent, SVRG (Johnson and Zhang, 2013, Xiao and Zhang, 2014)

$$\min_x\{L(x)+\psi(x)\} = \min_x\{\tfrac{1}{n}\sum_{i=1}^n \ell_i(x)+\psi(x)\}$$

With fixed reference point $\hat{x}$ which is close to x, a reduced variance gradient estimator is given as

$$g = \nabla\ell_i(x) - \nabla\ell_i(\hat{x}) + \underbrace{\tfrac{1}{n}\sum_{j=1}^n \nabla\ell_j(\hat{x})}_{\nabla L(\hat{x})}.$$

Bias: unbiased,

$$\mathrm{E}[g] = \frac{1}{n}\sum_{i=1}^n [\nabla\ell_i(x) - \nabla\ell_i(\hat{x}) + \nabla L(\hat{x})] = \frac{1}{n}\sum_{i=1}^n \nabla\ell_i(x) = \nabla L(x).$$

Variance ?

A key observation

$$g = \nabla \ell_i(x) - \nabla \ell_i(\hat{x}) + \nabla L(\hat{x}).$$

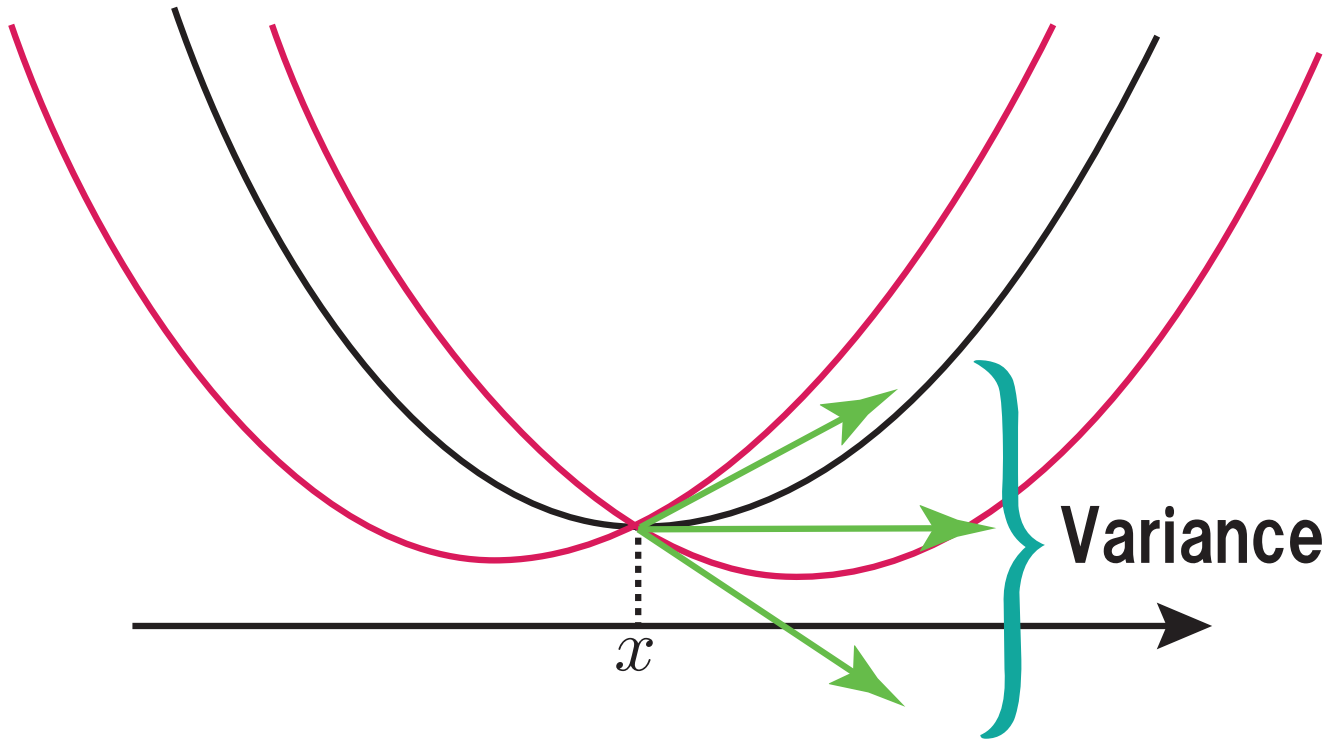
Variance:

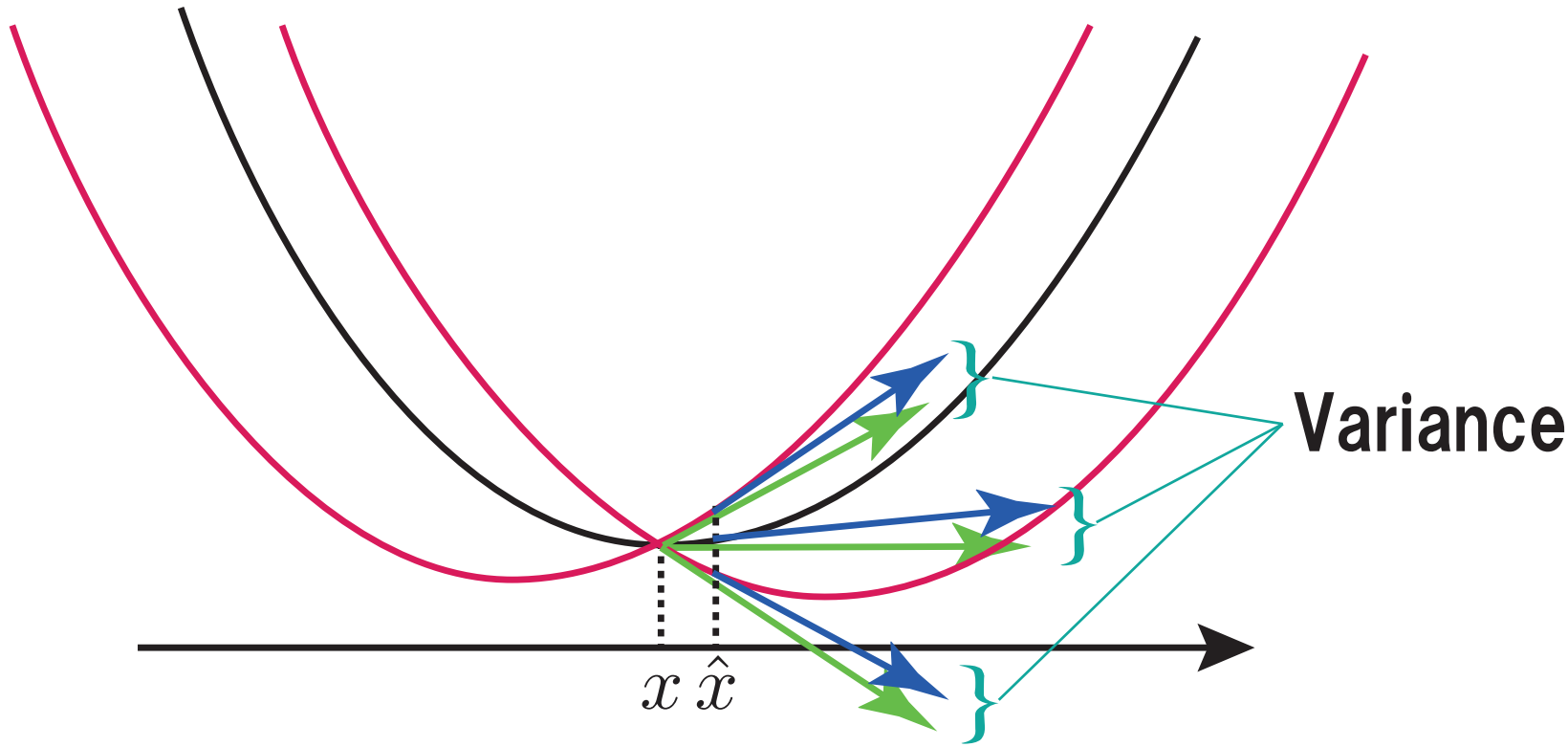
$$\begin{aligned}
\mathrm{Var}[g] &= \tfrac{1}{n} \textstyle\sum_{i=1}^n \|\nabla \ell_i(x) - \nabla \ell_i(\hat{x}) + \nabla L(\hat{x}) - \nabla L(x)\|^2 \\
&= \tfrac{1}{n} \textstyle\sum_{i=1}^n \left\|\nabla \ell_i(x) - \nabla \ell_i(\hat{x})\|^2 - \|\nabla L(\hat{x}) - \nabla L(x)\right\|^2 \\
&\qquad (\because \mathrm{Var}[X] = \mathrm{E}[\|X\|^2] - \|\mathrm{E}[X]\|^2) \\
&\leq \tfrac{1}{n} \textstyle\sum_{i=1}^n \|\nabla \ell_i(x) - \nabla \ell_i(\hat{x})\|^2 \\
&\leq \gamma \|x - \hat{x}\|^2.
\end{aligned}$$

The variance could be small if x and $\hat{x}$ are close and ℓ_i is smooth.

Main strategy:

- Calculate the full gradient at $\hat{x}$.
- Update x_t several times, say, $O(n)$ times.
- Set $\hat{x} = x_t$.


Variance
x


Variance
x
$\hat{x}$

Algorithm of SVRG

The algorithm consists of inner loop and outer loop.

SVRG

For $t = 1, 2, \ldots,$ iterate the following:

1. Set $\hat{x} = \hat{x}^{(t-1)}$, $x_{[0]} = \hat{x}$,

$$\hat{g} = \nabla L(\hat{x}) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell_i(\hat{x}). \quad \text{(full gradient)}$$

2. For $k = 1, \ldots, m$, execute the following:
   1. Uniformly sample $i \in \{1, \ldots, n\}$.
   2. Set
   $$g = \nabla \ell_i(x_{[k-1]}) - \nabla \ell_i(\hat{x}) + \hat{g}. \quad \text{(variance reduction)}$$
   3. Update $x_{[k]}$ as
   $$x_{[k]} = \text{prox}(x_{[k-1]} - \eta g | \eta \psi).$$
3. Set $\hat{x}^{(t)} = \frac{1}{m} \sum_{k=1}^{m} x_{[k]}$.

Computational complexity until t iteration: $O(t \times (n + m))$.

Convergence analysis

Assuption: ℓ_i is γ-smooth, and ψ is λ-strongly convex.

Theorem

If η and m satisfy $\eta > 4\gamma$ and

$$\rho := \frac{\eta}{\lambda(1-4\gamma/\eta)m} + \frac{4\gamma(m+1)}{\eta(1-4\gamma/\eta)m} < 1,$$

then, after T iteration the objective is bounded by

$$\mathrm{E}[P(\hat{x}^{(T)}) - P(x^*)] \leq \rho^T (P(\hat{x}^{(0)}) - P(x^*))$$

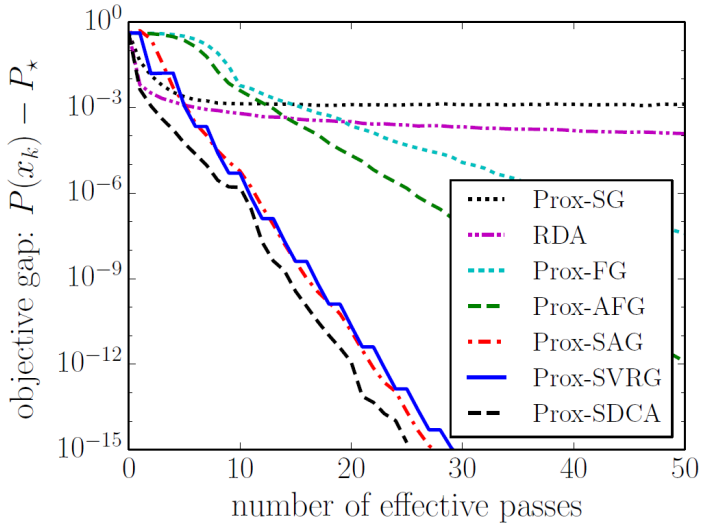
The assumption of the theorem is satisfied by

$$m \geq \Omega\left(\frac{\gamma}{\lambda}\right).$$

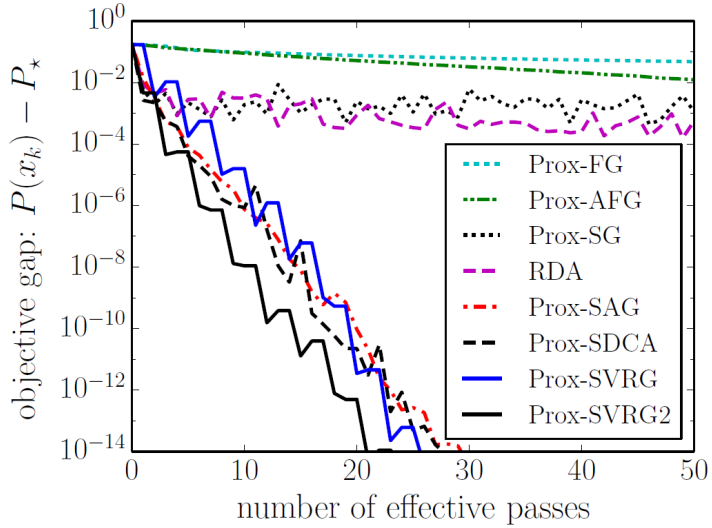
- Inner loop computation $O(n+m)$ for each t.
- Outer loop iteration $T = O(\log(1/\epsilon))$ until ϵ accuracy.

$\Rightarrow$ The whole computation :

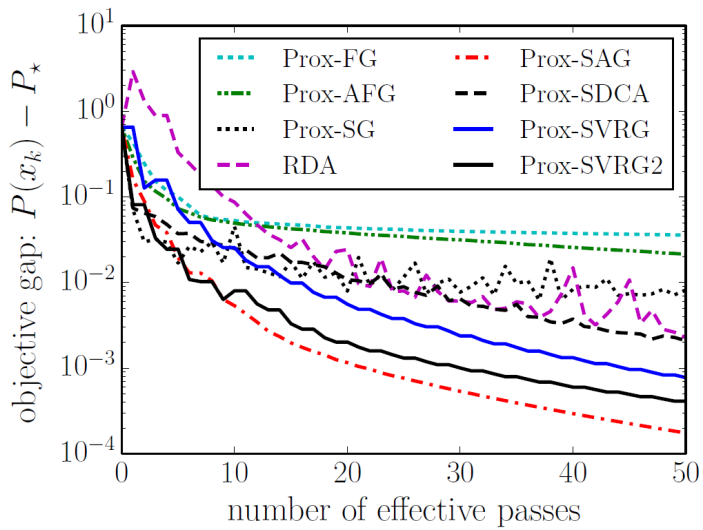
$$O\left((n+m)\log(1/\epsilon)\right) = O\left((n+\frac{\gamma}{\lambda})\log(1/\epsilon)\right)$$



(c) rcv1



(d) covertype



(e) sido0

Numerical comparison between several stochastic methods on a batch setting (figure is from Xiao and Zhang (2014)).

Related method: SAGA

SAGA (Defazio et al., 2014) does not require the double-loop, but requires more memory.
Difference: the gradient estimate g

$$\text{(SAGA)} \qquad g = \nabla \ell_i(x^{(t-1)}) - \nabla \ell_i(\hat{x}_i) + \frac{1}{n}\sum_{j=1}^{n} \nabla \ell_j(\hat{x}_j)$$

$$\text{(SVRG)} \qquad g = \nabla \ell_i(x^{(t-1)}) - \nabla \ell_i(\hat{x}) + \frac{1}{n}\sum_{j=1}^{n} \nabla \ell_i(\hat{x})$$

- $\hat{x}$ depends on the data index $i \in \{1, \ldots, n\}$.
- $\hat{x}_i$ is updated at every iteration:

$$\begin{cases} \hat{x}_i = x^{(t-1)} & (\text{if } i \text{ is chosen at the } t\text{-th round}), \\ \hat{x}_j \text{ is not changed} & (\forall j \neq i). \end{cases}$$

- Update rule of $x^{(t)}$ is same: $x^{(t)} = \mathrm{prox}(x^{(t-1)} - \eta g | \eta\psi)$.
- We need to store all gradients $\nabla \ell_i(\hat{x}_i)$ $(i = 1, \ldots, n)$.

Algorithm of SAGA

SAGA (Defazio et al., 2014)

1. Pick up $i \in \{1, \ldots, n\}$ uniformly at random.
2. Update $g_j^{(t)}$ $(j = 1, \ldots, n)$ as

$$g_j^{(t)} = \begin{cases} \nabla \ell_i(x^{(t-1)}) & (i = j), \\ g_j^{(t-1)} & (\text{otherwise}). \end{cases}$$

3. Update $x^{(t)}$ as

$$v_t = g_i^{(t)} - g_i^{(t-1)} + \frac{1}{n} \sum_{j=1}^{n} g_j^{(t-1)},$$

$$x^{(t)} = \mathrm{prox}(x^{(t-1)} - \eta v_t | \eta \psi).$$

Convergence of SAGA

Assumption: ℓ_i is γ-smooth and λ-strongly convex ($\lambda = 0$ is allowed) ($\forall i = 1, \ldots, n$).

Theorem

Set $\eta = 1/3\gamma$. Then

- $\lambda = 0$: *for $\bar{x}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} x^{(t)}$, it holds that*

$$\mathrm{E}[P(\bar{x}^{(T)}) - P(x^*)] \leq \frac{4n}{T} C_0.$$

- $\lambda > 0$:

$$\mathrm{E}[\|x^{(T)} - x^*\|^2] \leq \left(1 - \min\left\{\frac{1}{4n}, \frac{\lambda}{3\gamma}\right\}\right)^T C_1.$$

SAGA is adaptive to the strong convexity λ.

Stochastic Average Gradient descent, SAG

- Like SAGA, SAG is also a **single-loop** method (Le Roux et al., 2012, Schmidt et al., 2013).
- Historically SAG was proposed earlier than SVRG and SAGA.
- Proximal technique can not be involved in SAG
  $\rightarrow$ SAGA was proposed to overcome this drawback.

$$P(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x) \simeq \langle g, x \rangle.$$

(SAG)
$$g = \frac{\nabla \ell_i(x^{(t-1)}) - \nabla \ell_i(\hat{x}_i)}{n} + \frac{1}{n}\sum_{j=1}^{n} \nabla \ell_j(\hat{x}_j)$$

g is biased.

(SAGA) $g = \nabla \ell_i(x^{(t-1)}) - \nabla \ell_i(\hat{x}_i) + \frac{1}{n}\sum_{j=1}^{n} \nabla \ell_j(\hat{x}_j)$

(SVRG) $g = \nabla \ell_i(x^{(t-1)}) - \nabla \ell_i(\hat{x}) + \frac{1}{n}\sum_{j=1}^{n} \nabla \ell_i(\hat{x})$

Algorithm of SAG

SAG

Initialize $g_i^{(0)} = \mathbf{0}$ $(i = 1, \ldots, n)$.
For $t = 1, 2, \ldots,$ iterate the following:

1. Pick up $i \in \{1, \ldots, n\}$ uniformly at random.
2. Update $g_{i'}^{(t)}$ $(i' = 1, \ldots, n)$ as

$$g_{i'}^{(t)} = \begin{cases} \nabla \ell_i(x^{(t-1)}) & (i = i'), \\ g_{i'}^{(t-1)} & (\text{otherwise}). \end{cases}$$

3. Update $x^{(t)}$ as

$$x^{(t)} = x^{(t-1)} - \frac{\eta}{n} \sum_{j=1}^{n} g_j^{(t)}.$$

Convergence analysis of SAG

Assumption: ℓ_i is γ-smooth and $P(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x)$ is λ-strongly convex ($\lambda = 0$ is allowed).
Milder condition than SAGA because the strong convexity is about $P(x)$ rather than the loss function ℓ_i.

Theorem (Convergence rate of SAG)

Set $\eta = \frac{1}{16\gamma}$. Then SAG converges as

- $\lambda = 0$: $\bar{x}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} x^{(t)}$ に対し,

$$\mathrm{E}[P(\bar{x}^{(T)}) - P(x^*)] \leq \frac{32n}{T} C_0$$

- $\lambda > 0$:

$$\mathrm{E}[\|x^{(T)} - x^*\|^2] \leq \left(1 - \min\left\{\frac{1}{8n}, \frac{\lambda}{16\gamma}\right\}\right)^T C_0.$$

SAG also has adaptivity.

Catalyst: Acceleration of SVRG, SAG, SAGA

Catalyst (Lin et al., 2015)

Iterate the following for $t = 1, 2, \ldots$:

1. Find an approximated solution of a modified problem which has higher strong convexity:

$$x^{(t)} \simeq \operatorname{argmin}_x \left\{ P(x) + \tfrac{\alpha}{2}\|x - y^{(t-1)}\|^2 \right\} \quad (\text{up to } \epsilon_t \text{ precision}).$$

2. Accelerate the solution: $y^{(t)} = x^{(t)} + \beta_t (x^{(t)} - x^{(t-1)})$.

– Catalyst is an acceleration method of an inexact proximal point alg.

– For $\epsilon_t = C(1 - \sqrt{\lambda/2(\lambda+\alpha)})^t$,

$$P(x^{(t)}) - P(x^*) \leq C' \left(1 - \sqrt{\tfrac{\lambda}{2(\lambda+\alpha)}}\right)^t.$$

– Using SVRG, SAG, SAGA with $\alpha = \max\{c\frac{\gamma}{n}, \lambda\}$ in the inner loop achieves $(n + \sqrt{\frac{\gamma n}{\lambda}}) \log(1/\epsilon)$ overall computation.

– This is a universal method but is sensitive to the choice of the inner loop iteration number and α.

Summary and comparison of batch methods

Properties of the batch methods

Method	SDCA	SVRG	SAG
P/D	Dual	Primal	Primal
Memory efficiency	✓	✓	△
Acceleration ($\mu > 0$)	✓	Catalyst	Catalyst
Other remark	$\ell_i(\beta) = f_i(x_i^\top \beta)$	double loop	smooth reg.

Summary and comparison of batch methods

Properties of the batch methods

Method	SDCA	SVRG	SAG
P/D	Dual	Primal	Primal
Memory efficiency	✓	✓	△
Acceleration ($\mu > 0$)	✓	Catalyst	Catalyst
Other remark	$\ell_i(\beta) = f_i(x_i^\top \beta)$	double loop	smooth reg.

Convergence rate (up to log term of $\gamma,\ \mu$)

Method	$\lambda > 0$	$\lambda = 0$	Acceleration ($\mu > 0$)
SDCA	$(n + \frac{\gamma}{\lambda}) \log(1/\epsilon)$	-	$(n + \sqrt{\frac{n\gamma}{\lambda}}) \log(1/\epsilon)$
SVRG	$(n + \frac{\gamma}{\lambda}) \log(1/\epsilon)$	-	$(n + \sqrt{\frac{n\gamma}{\lambda}}) \log(1/\epsilon)$
SAG	$(n + \frac{\gamma}{\lambda}) \log(1/\epsilon)$	$\gamma n/\epsilon$	$(n + \sqrt{\frac{n\gamma}{\lambda}}) \log(1/\epsilon)$

: Catalyst.

As for $\mu = 0$, Catalyst gives an acceleration with convergence rate $O(n\sqrt{\frac{\gamma}{\epsilon}})$.

Outline

1. Online stochastic optimization
   - Stochastic gradient descent
   - Stochastic regularized dual averaging
2. Getting stochastic gradient methods faster
   - Bregman divergence and AdaGrad
   - Acceleration of stochastic gradient methods
   - Minimax optimality of first order online stochastic methods
3. Batch stochastic methods
   - Dual method: stochastic dual coordinate ascent
   - Primal method: SVRG, SAG and SAGA
   - Minimax optimality of first order batch stochastic methods

Minimax optimal convergence rate

Let $\kappa = \frac{\gamma}{\lambda}$ be the condition number.

The iteration number

$$T \geq (n + \kappa) \log(1/\epsilon)$$

is almost minimax, but not minimax.

The accelerated version

$$T \geq \left(n + \sqrt{n\kappa}\right) \log(1/\epsilon)$$

is minimax up to $\log(1/\epsilon)$ (Agarwal and Bottou, 2015).

Minimax optimality in the batch setting

$$P(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x) + \frac{\lambda}{2}\|x\|^2$$

Assumption: ℓ_i is $(\gamma - \lambda)$-smooth ($\gamma > \lambda$).

First order oracle: for an input (x, i), it returns the pair $(\ell_i(x), \nabla \ell_i(x))$.
First order algorithm: an algorithm that depends on only the return of the first order oracle for a query point x.

(SAG, SAGA, SVRG are included. SDCA is not included.)

Theorem (Minimax optimal rate for FOA (Agarwal and Bottou, 2015))

For any first order algorithm, there exist functions ℓ_i $(i = 1, \ldots, n)$ satisfying the assumption on which the algorithm must perform at least

$$T \geq \Omega\left(n + \sqrt{n(\kappa - 1)} \log(1/\epsilon)\right)$$

calls for the first order oracle to get $\|x^{(T)} - x^*\| \leq \epsilon \|x^*\|$.

A. Agarwal and L. Bottou. A lower bound for the optimization of finite sums. In the 32nd International Conference on Machine Learning, pages 78–86, 2015.

A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. IEEE Transcations on Information Theory, 58(5):3235–3249, 2012.

A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. SIAM Journal on Optimization, 23(4): 2037–2060, 2013.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010, pages 177–186. Springer, 2010.

X. Chen, Q. Lin, and J. Pena. Optimal regularized dual averaging methods for stochastic optimization. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 395–403. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4543-optimal-regularized-dual-averaging-methods-for-stochas` `pdf`.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, Advances in Neural Information Processing Systems 27, pages 1646–1654. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5258-saga-a-fast-incremental-gradient-method-with-support-f pdf.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12:2121–2159, 2011.

O. Fercoq and P. Richtárik. Accelerated, parallel and proximal coordinate descent. Technical report, 2013. arXiv:1312.5799.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. SIAM Journal on Optimization, 22(4): 1469–1492, 2012.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking

procedures and optimal algorithms. SIAM Journal on Optimization, 23 (4):2061–2089, 2013.

C. Hu, W. Pan, and J. T. Kwok. Accelerated gradient methods for stochastic optimization and online learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, Advances in Neural Information Processing Systems 22, pages 781–789. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3817-accelerated-gradient-methods-for-stochastic-optimizati pdf.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 315–323. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/4937-accelerating-stochastic-gradient-descent-using-predict pdf.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

G. Lan. An optimal method for stochastic composite optimization. Mathematical Programming, 133(1-2):365–397, 2012.

N. Le Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 2663–2671. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4633-a-stochastic-gradient-method-with-an-exponential-conve rate-for-finite-training-sets.pdf.

H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. Technical report, 2015. arXiv:1506.02186.

Q. Lin, Z. Lu, and L. Xiao. An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization. Technical report, 2014. arXiv:1407.1296.

A. Nemirovsky and D. Yudin. Problem complexity and method efficiency in optimization. John Wiley, New York, 1983.

Y. Nesterov. Primal-dual subgradient methods for convex problems. Mathematical Programming, 120(1):221–259, 2009.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization, 22(2):341–362, 2012.

A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In J. Langford and J. Pineau, editors, Proceedings of the 29th International Conference on Machine Learning, pages 449–456. Omnipress, 2012. ISBN 978-1-4503-1285-1.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Mathematical Programming, 144:1–38, 2014.

A. Saha and A. Tewari. On the non-asymptotic convergence of cyclic coordinate descent methods. SIAM Journal on Optimization, 23(1): 576–601, 2013.

M. Schmidt, N. Le Roux, and F. R. Bach. Minimizing finite sums with the stochastic average gradient, 2013. hal-00860051.

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

S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In Advances in Neural Information Processing Systems 26, 2013b.

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In Proceedings of The 31st International Conference on Machine Learning, pages 64–72, 2014.

M. Takáč, A. Bijral, P. Richtárik, and N. Srebro. Mini-batch primal and dual methods for SVMs. In Proceedings of the 30th International Conference on Machine Learning, 2013.

T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

S. J. Wright. Coordinate descent algorithms. Mathematical Programming, 151(1):3–34, 2015.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. In Advances in Neural Information Processing Systems 23. 2009.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. Journal of Machine Learning Research, 11: 2543–2596, 2010.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. SIAM Journal on Optimization, 24: 2057–2075, 2014.

M. D. Zeiler. ADADELTA: an adaptive learning rate method. CoRR, abs/1212.5701, 2012. URL http://arxiv.org/abs/1212.5701.