Stochastic gradient methods

Pierre Ablin

CNRS, Université Paris-Dauphine

Let's build an algorithm that tells whether there is a cat in a picture:

Yes

No

Let's build an algorithm that tells whether there is a cat in a picture:

Yes

Let's build an algorithm that tells whether there is a cat in a picture:

$\longrightarrow$ **Yes**

Let's build an algorithm that tells whether there is a cat in a picture:

→ **No**

Let's build an algorithm that tells whether there is a cat in a picture:

Yes

Let's build an algorithm that tells whether there is a cat in a picture:

$\xrightarrow{h}$ $\left\{ \begin{array}{l} \textbf{Yes} \\ \textbf{No} \end{array} \right.$

x: Input/Feature

y: Output/Target

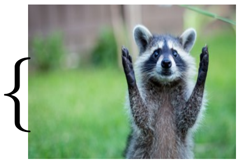
Find mapping h that assigns the "correct" target to each input

$$h : x \in \mathbb{R}^d \longrightarrow \mathrm{y} = \pm 1$$

Labelled Data: The training set

$x^1 \{$  $y^1 = 1$

$x^2 \{$  $y^2 = 1$

$x^3 \{$  $y^3 = -1$

$\cdots\ x^n \{$  $y^n = 1$

y= -1 means no/false

Learning Algorithm

$h : x \in \mathbb{R}^d \rightarrow y \in \pm 1$

$h\Big($  $\Big) \longrightarrow$ **-1**

A parametrized decision function

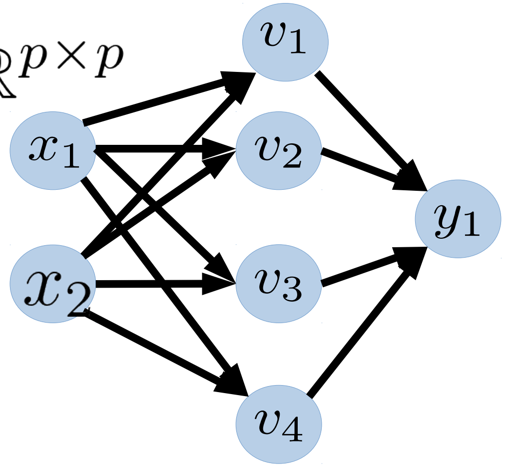
$$h : x \in \mathbb{R}^d \to y$$

h is a function parametrized by parameters $\mathbf{w}$

Examples

Linear: $h_{\mathbf{w}}(x) = w_1 x_1 + \cdots + w_p x_p, \quad \mathbf{w} \in \mathbb{R}^p$

Polynomial: $h_{\mathbf{w}}(x) = \sum_{ij} x_i x_j w_{ij}, \quad \mathbf{w} \in \mathbb{R}^{p \times p}$



Neural network: $h_{\mathbf{w}}(x) = \mathbf{w}_2 \sigma(\mathbf{w}_1 x)$

$\mathbf{w}_2 \in \mathbb{R}^q, \quad \mathbf{w}_1 \in \mathbb{R}^{q \times p}$

Learning parameters

Goal :

Find $\mathbf{w}$ such that for $(x,\ y)$ in our dataset :

$$h_{\mathbf{w}}(x) \simeq y$$

Mathematical reformulation

Find $\mathbf{w}$ that minimizes a discrepancy:

$$\min F(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(h_{\mathbf{w}}(x_i), y_i)$$

Learning parameters

Goal :

Find $\mathbf{w}$ such that for $(x,\ y)$ in our dataset :

$$h_{\mathbf{w}}(x) \simeq y$$

Mathematical reformulation

Find $\mathbf{w}$ that minimizes a discrepancy:

As many terms as images !

$$\min F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(h_{\mathbf{w}}(x_i), y_i)$$

Solving the Finite Sum Training Problem

Recap

Training Problem

$$\min_{w\in\mathbf{R}^d} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right)}_{L(w)} + \lambda R(w) =: f(w)$$

General methods

$$\min f(w)$$

- Gradient Descent

Optimization Sum of Terms

A Datum Function

$$f_i(w) := \ell\left(h_w(x^i), y^i\right) + \lambda R(w)$$

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right) + \lambda R(w) &= \frac{1}{n}\sum_{i=1}^{n} \left(\ell\left(h_w(x^i), y^i\right) + \lambda R(w)\right) \\ &= \frac{1}{n}\sum_{i=1}^{n} f_i(w)\end{aligned}$$

Finite Sum Training Problem

$$\min_{w \in \mathbf{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(w) =: f(w)$$

Can we use this sum structure?

The Training Problem

Solving the *training problem*: $$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

Reference method: Gradient descent

$$\nabla \left(\frac{1}{n} \sum_{i=1}^{n} f_i(w) \right) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w)$$

Gradient Descent Algorithm

Set $w^0 = 0$, choose $\alpha > 0$.

for $t = 0, 1, 2, \ldots, T-1$

$$w^{t+1} = w^t - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla f_i(w^t)$$

Output w^T

The Training Problem

Solving the *training problem*: $$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

Problem with Gradient Descent:
Each iteration requires computing a gradient $\nabla f_i(w)$ for each data point. One gradient for each cat on the internet!

Gradient Descent Algorithm

Set $w^0 = 0$, choose $\alpha > 0$.
for $t = 0, 1, 2, \ldots, T$
$$w^{t+1} = w^t - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla f_i(w^t)$$
Output w^T

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \ldots, n\}$ selected uniformly at random. Then

$$\mathbb{E}_j[\nabla f_j(w)] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = \nabla f(w)$$

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \ldots, n\}$ selected uniformly at random. Then

$$\mathbb{E}_j[\nabla f_j(w)] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = \nabla f(w)$$

Use $\nabla f_j(w) \approx \nabla f(w)$

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \ldots, n\}$ selected uniformly at random. Then

$$\mathbb{E}_j[\nabla f_j(w)] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = \nabla f(w)$$

Use $\nabla f_j(w) \approx \nabla f(w)$

EXE: Let $\sum_{i=1}^{n} p_i = 1$ and $j \sim p_j$. Show $\mathbb{E}[\nabla f_j(w)/(np_j)] = \nabla f(w)$

Stochastic Gradient Descent

SGD 0.0 Constant stepsize

Set $w^0 = 0$, choose $\alpha > 0$

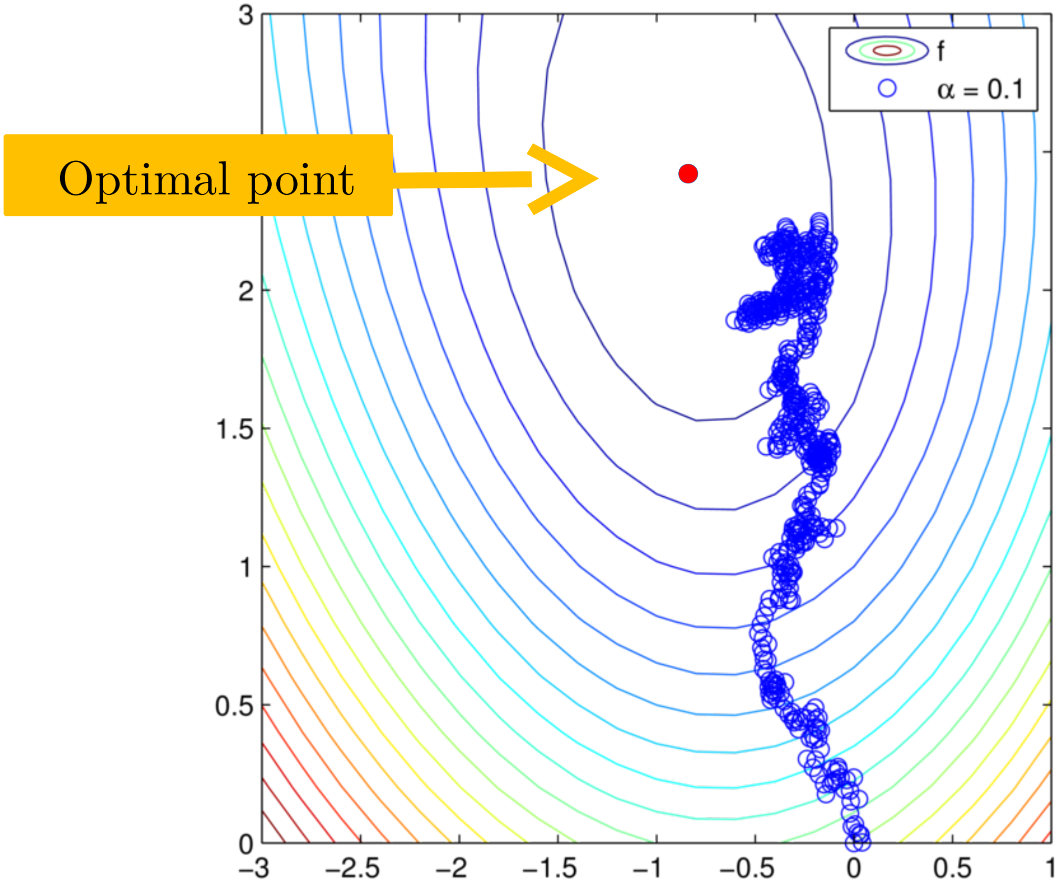
for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha \nabla f_j(w^t)$

Output w^T

Stochastic Gradient Descent



Convergence Strongly Convex and Bounded Gradient

Theorem If f is $\mu-$ strongly convex and $\mathbb{E}[\|\nabla f_i(w)\|^2] \leq B^2$

If $0 < \alpha \leq \frac{1}{\mu}$ then the iterates of the SGD method satisfy

$$\mathbb{E}\left[||w^t - w^*||_2^2\right] \leq (1-\alpha\mu)^t ||w^0 - w^*||_2^2 + \frac{\alpha}{\mu} B^2$$

Shows that $\alpha \approx \frac{1}{\mu}$

Shows that $\alpha \approx 0$

Proof: $$w^{t+1} = w^t - \alpha \nabla f_j(w^t), \quad j \sim [1, \ldots, n]$$

1) Show that

$$||w^{t+1} - w^*||_2^2 \quad = \quad ||w^t - w^*||_2^2 - 2\alpha \langle \nabla f_j(w^t), w^t - w^* \rangle + \alpha^2 ||\nabla f_j(w^t)||_2^2.$$

2) Show that

$$\mathbb{E}_j \left[||w^{t+1} - w^*||_2^2 \right] \leq ||w^t - w^*||_2^2 - 2\alpha \langle \nabla f(w^t), w^t - w^* \rangle + \alpha^2 B^2$$

3) Using strong convexity, demonstrate that

$$\mathbb{E}_j \left[||w^{t+1} - w^*||_2^2 \right] \leq (1 - \alpha\mu) ||w^t - w^*||_2^2 + \alpha^2 B^2$$

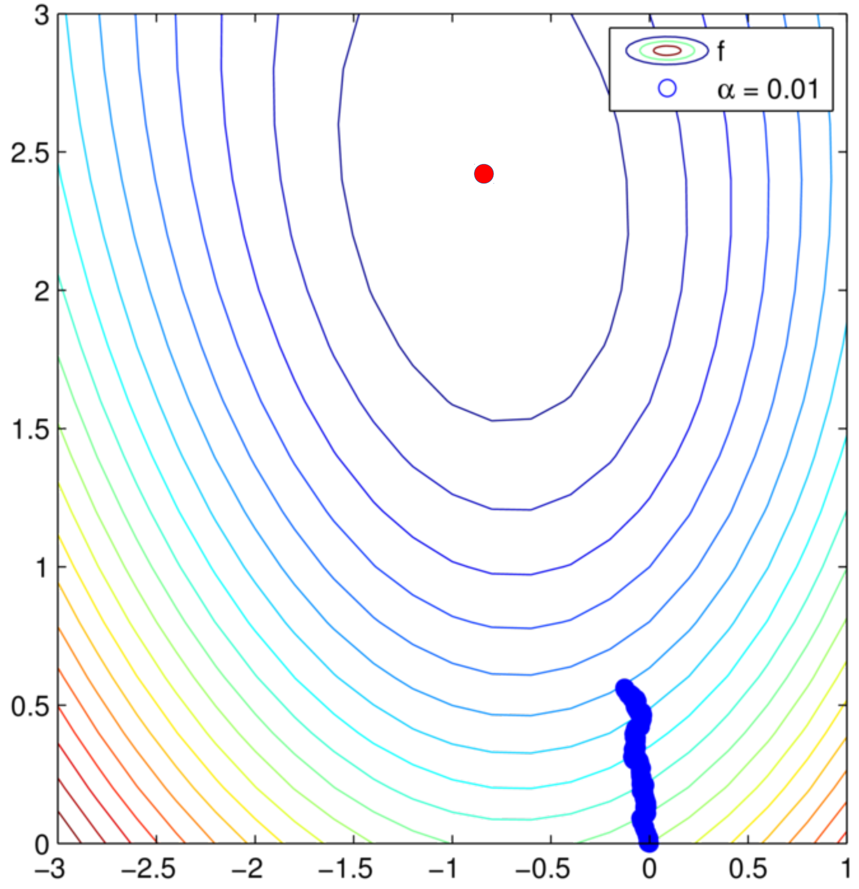
4) Show that

$$\mathbb{E} \left[||w^{t+1} - w^*||_2^2 \right] \leq (1 - \alpha\mu) \mathbb{E} \left[||w^t - w^*||_2^2 \right] + \alpha^2 B^2$$

Where the expectation is taken w.r.t. the whole past. Conclude.

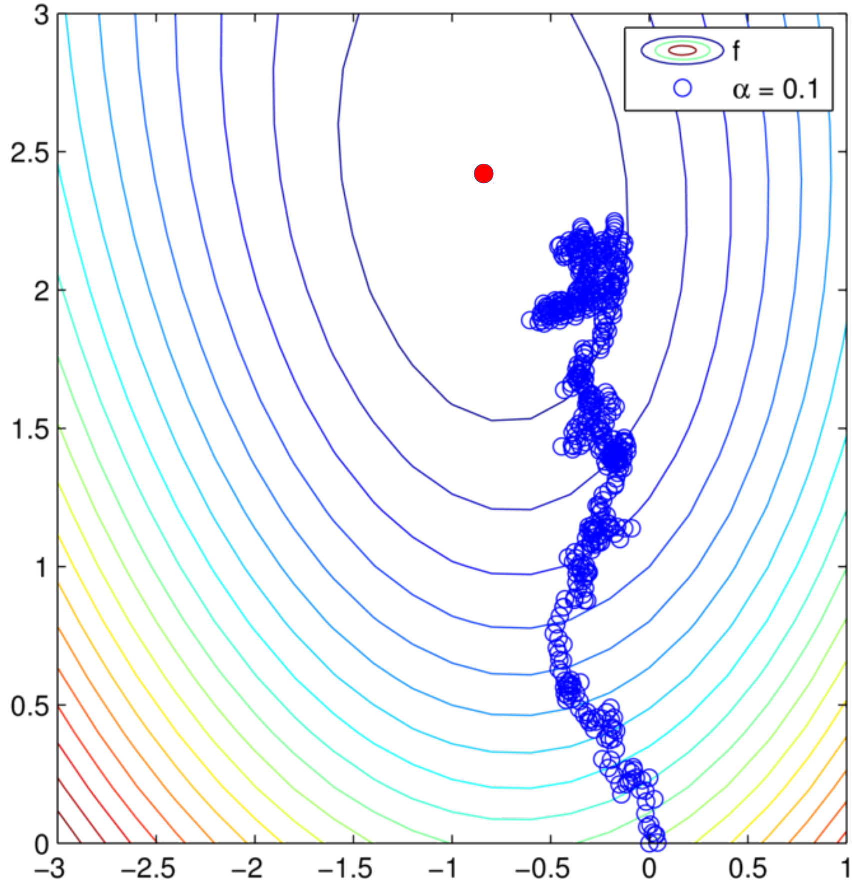
Stochastic Gradient Descent

α =0.01



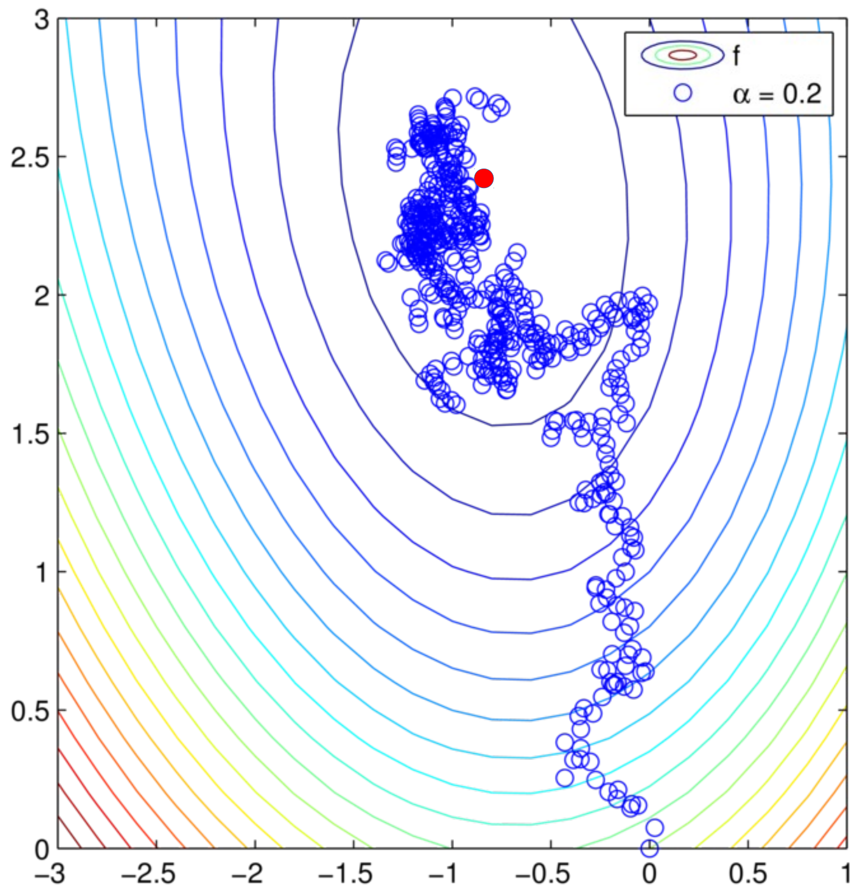
Stochastic Gradient Descent

α =0.1



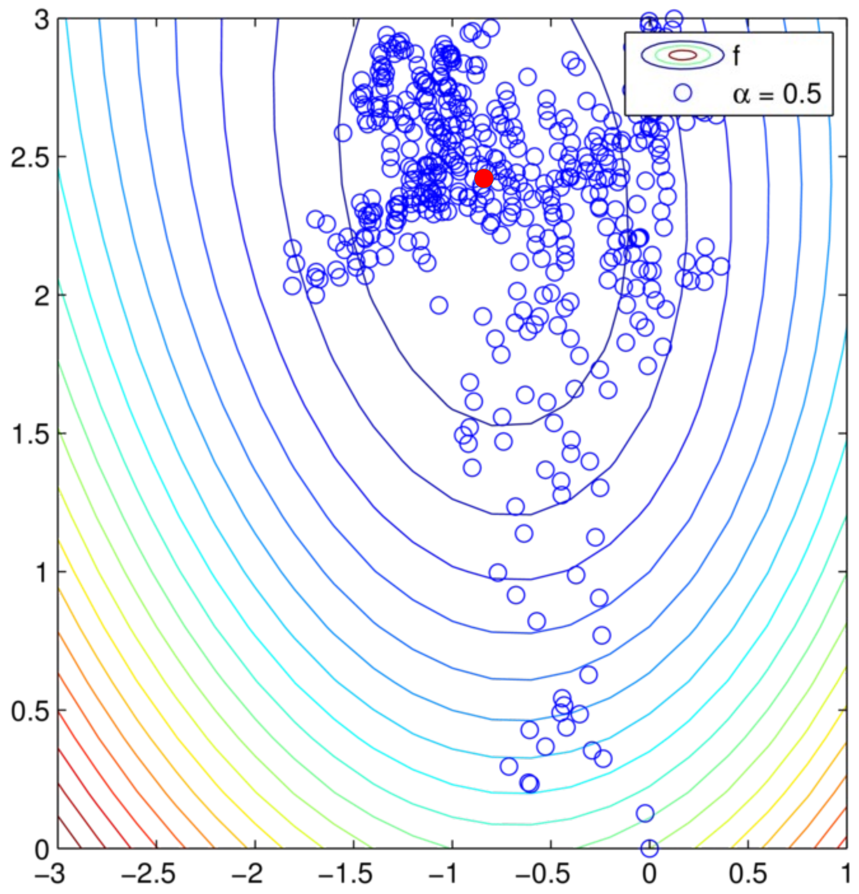
Stochastic Gradient Descent

α =0.2



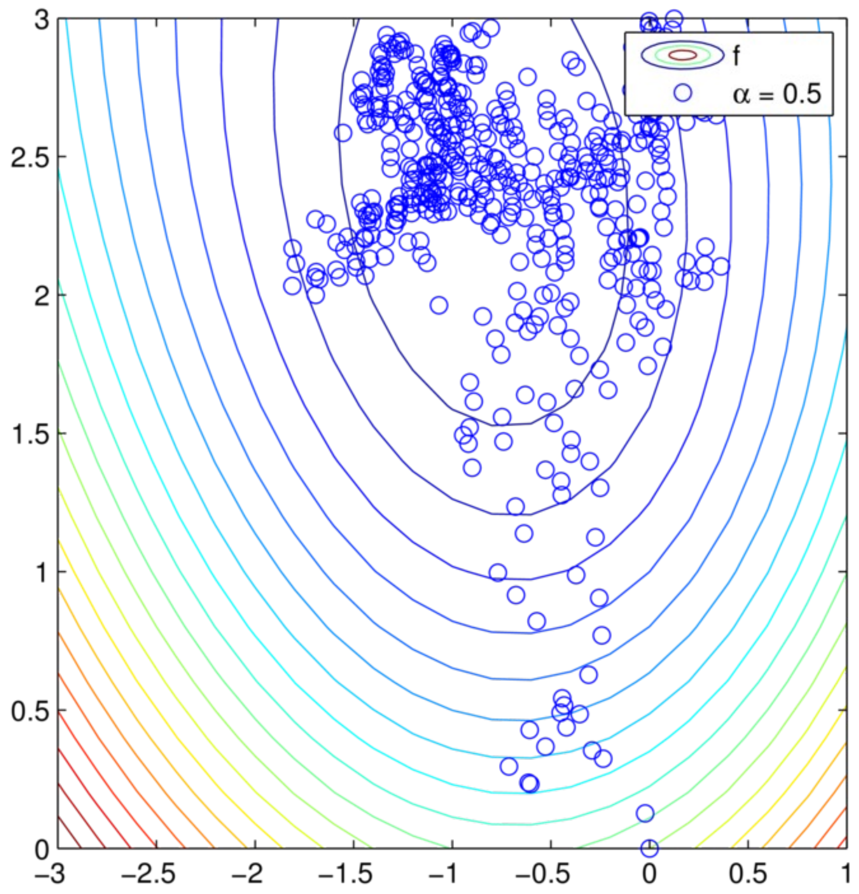
Stochastic Gradient Descent

α =0.5



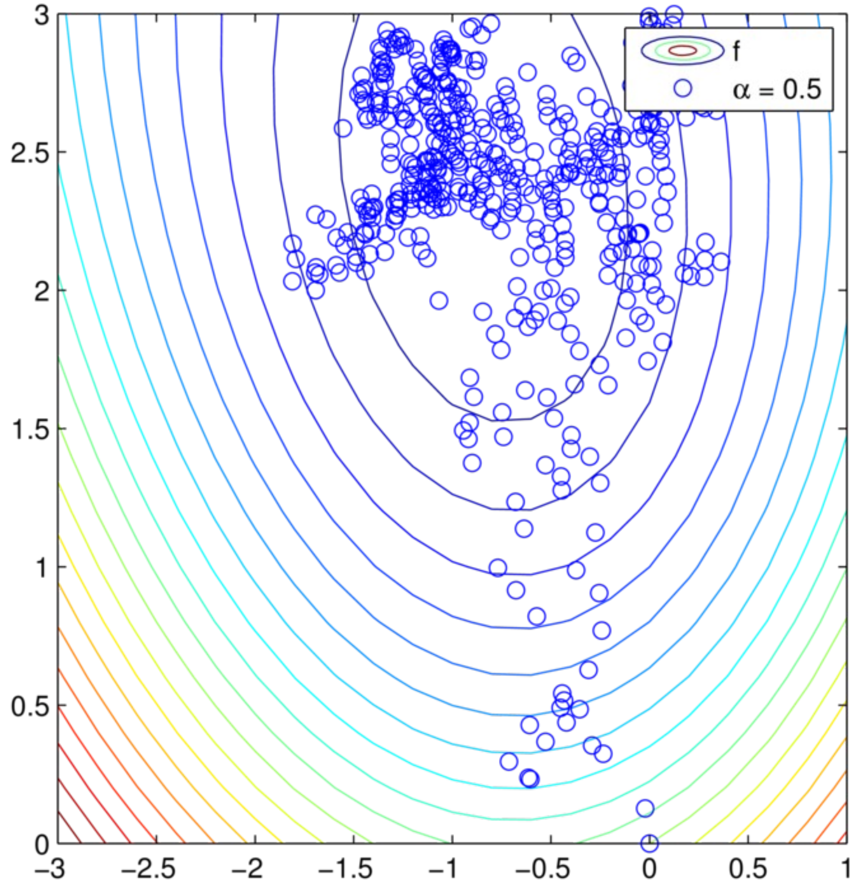
Stochastic Gradient Descent

α =0.5



Stochastic Gradient Descent

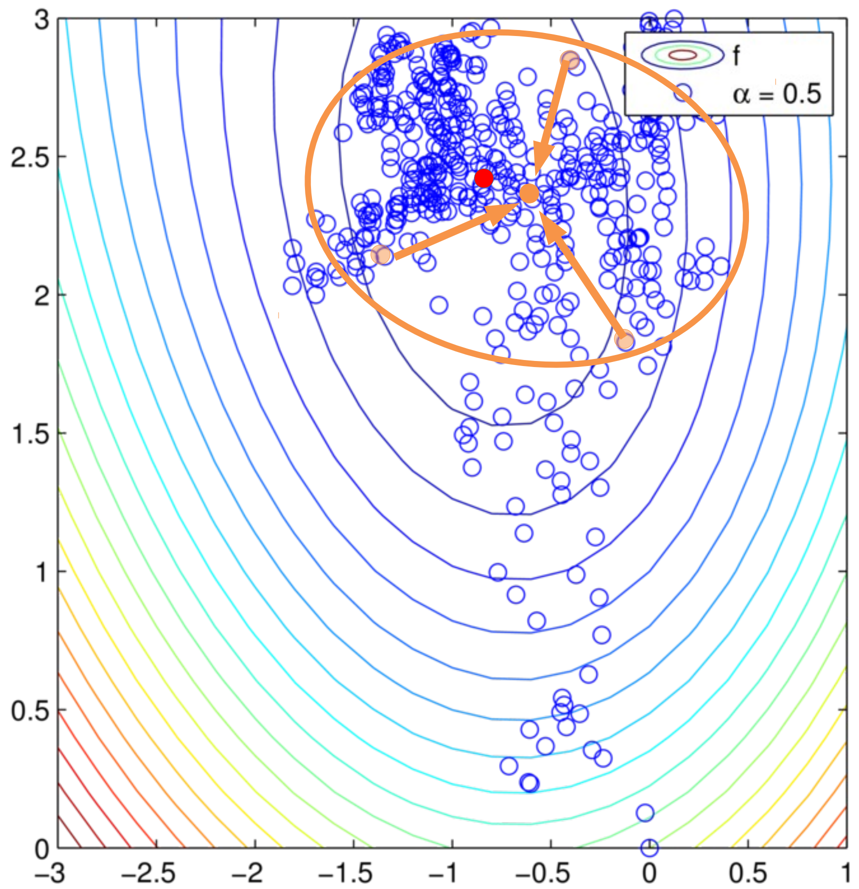
α =0.5



1) Start with big steps and end with smaller steps

Stochastic Gradient Descent

α =0.5



1) Start with big steps and end with smaller steps

2) Try averaging the points

SGD shrinking stepsize

SGD 1.0: Descreasing stepsize

Set $w^0 = 0$

Choose $\alpha_t > 0$, $\alpha_t \to 0$, $\sum_{t=0}^{\infty} \alpha_t = \infty$

for $t = 0, 1, 2, \ldots, T-1$

sample $j \in \{1, \ldots, n\}$

$w^{t+1} = w^t - \alpha_t \nabla f_j(w^t)$

Output w^T

Shrinking Stepsize

SGD shrinking stepsize

SGD 1.0: Descreasing stepsize

Set $w^0 = 0$

Choose $\alpha_t > 0$, $\alpha_t \rightarrow 0$, $\sum_{t=0}^{\infty} \alpha_t = \infty$

for $t = 0, 1, 2, \ldots, T-1$

sample $j \in \{1, \ldots, n\}$

$w^{t+1} = w^t - \alpha_t \nabla f_j(w^t)$

Output w^T

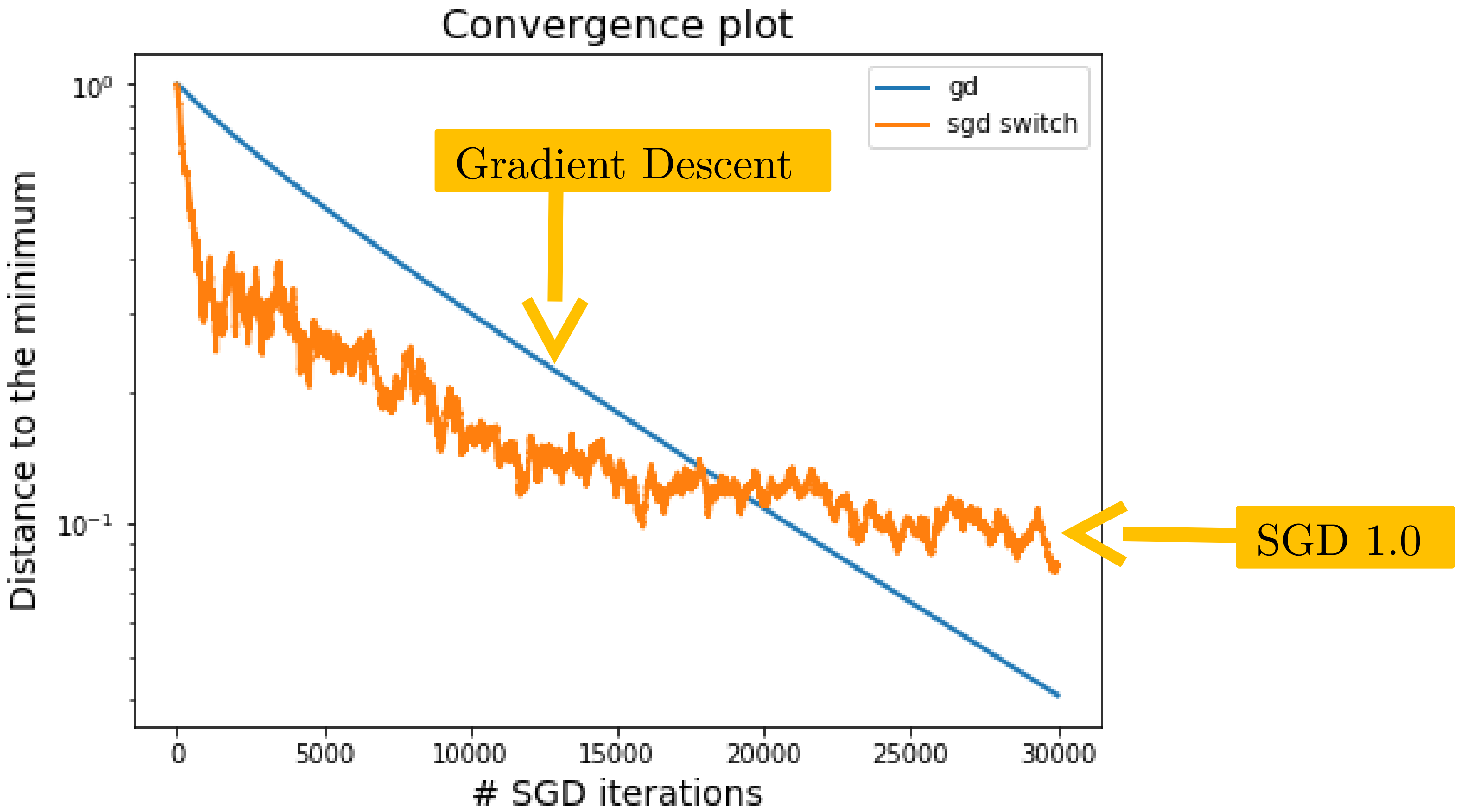
Shrinking Stepsize

How should we sample j ?

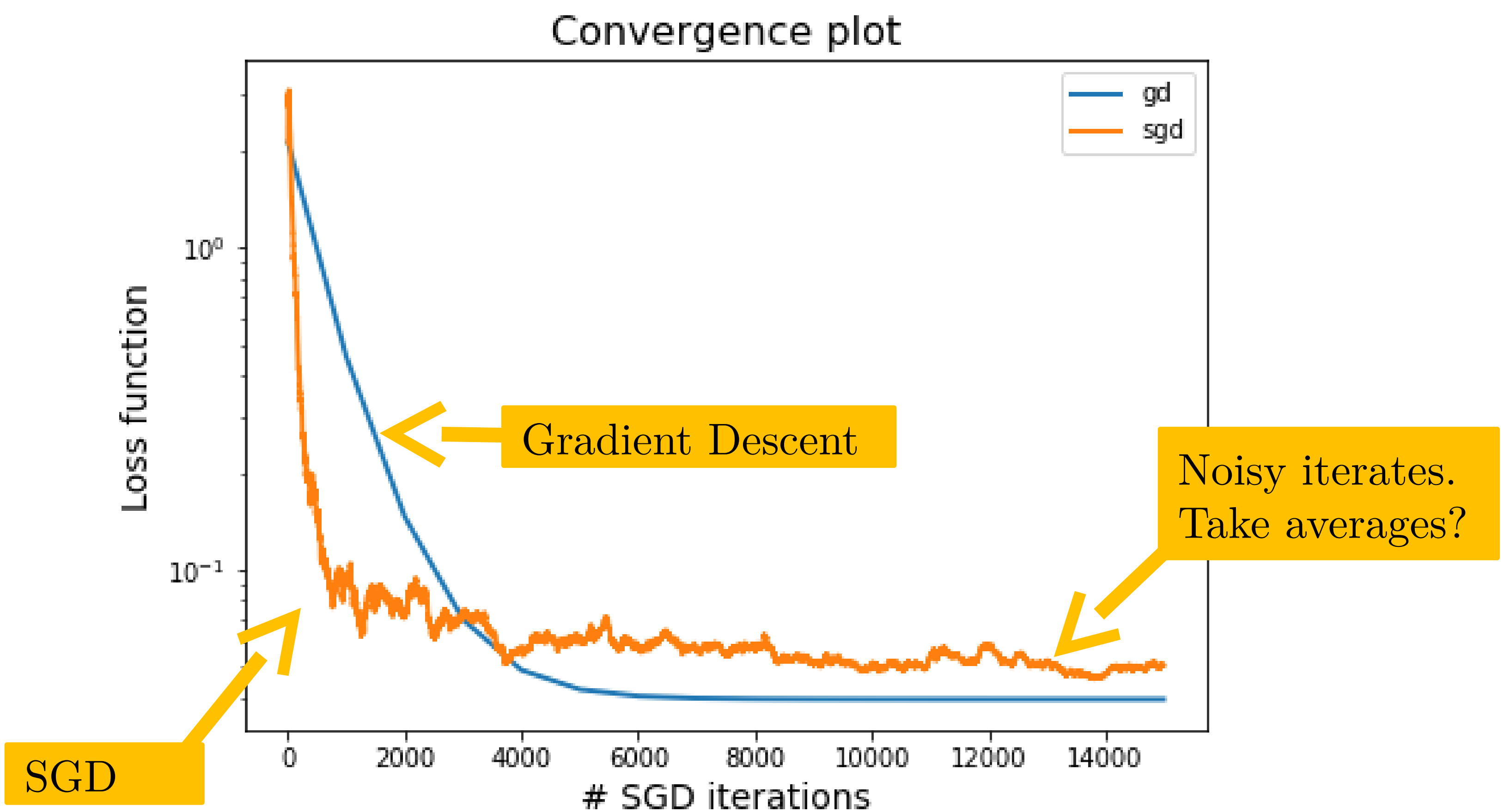
How fast $\alpha_t \rightarrow 0$?

Does this converge?

SGD with shrinking stepsize
Compared with Gradient Descent



SGD with shrinking stepsize
Compared with Gradient Descent



Proof: $$w^{t+1} = w^t - \alpha_t \nabla f_j(w^t), \quad j \sim [1, \ldots, n]$$

1) Recall that $\mathbb{E}\left[||w^{t+1} - w^*||_2^2\right] \leq (1 - \alpha_t \mu)\mathbb{E}\left[||w^t - w^*||_2^2\right] + \alpha_t^2 B^2$

Let $\delta_t = \mathbb{E}[\|w_t - w^*\|^2]$ and $\pi_t^i = (1 - \alpha_{t-1}\mu) \times \cdots \times (1 - \alpha_i \mu)$

$$\delta_t \leq \pi_t^0 + \sum_{t=0}^{t-1} \pi_t^i \alpha_i^2 B^2$$

2) Show that if $\sum_{t=0}^{+\infty} \alpha_t = +\infty$ then $\lim_{t \to +\infty} \pi_t^0 = 0$

3) Using $\pi_t^i \leq \pi_t^0$, show that if $\sum_{t=0}^{+\infty} \alpha_i^2 < +\infty$, then $\lim_{t \to +\infty} \sum_{t=0}^{t-1} \pi_t^i \alpha_i^2 = 0$

Convergence when $\sum_{t=0}^{+\infty} \alpha_i = +\infty$ and $\sum_{t=0}^{+\infty} \alpha_i^2 < +\infty$

SGD with (late start) averaging

SGDA 1.1

Set $w^0 = 0$

Choose $\alpha_t > 0,\ \alpha_t \to 0,\ \sum_{t=0}^{\infty} \alpha_t = \infty$

Choose averaging start $s_0 \in \mathbb{N}$

for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha_t \nabla f_j(w^t)$

  if $t > s_0$

    $\overline{w} = \frac{1}{t-s_0} \sum_{i=s_0}^{t} w^t$

  else: $\overline{w} = w$

Output $\overline{w}$

B. T. Polyak and A. B. Juditsky, SIAM Journal on Control and Optimization (1992)

Acceleration of stochastic approximation by averaging

SGD with (late start) averaging

SGDA 1.1

Set $w^0 = 0$

Choose $\alpha_t > 0,\ \alpha_t \rightarrow 0,\ \sum_{t=0}^{\infty} \alpha_t = \infty$

Choose averaging start $s_0 \in \mathbb{N}$

for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha_t \nabla f_j(w^t)$

  if $t > s_0$

    $\overline{w} = \frac{1}{t - s_0} \sum_{i=s_0}^{t} w^t$

  else: $\overline{w} = w$

Output $\overline{w}$

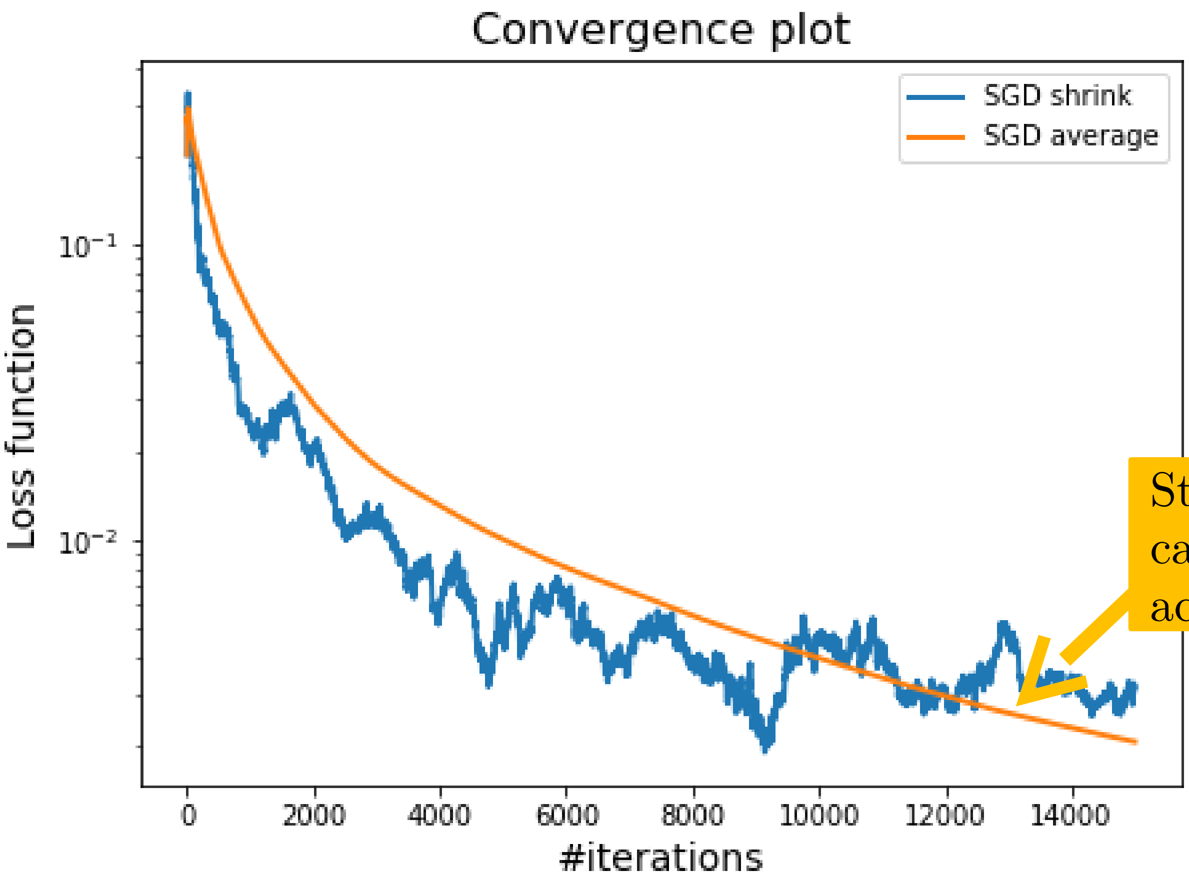
This is not efficient. How to make this efficient?

B. T. Polyak and A. B. Juditsky, SIAM Journal on Control and Optimization (1992)

Acceleration of stochastic approximation by averaging

Stochastic Gradient Descent

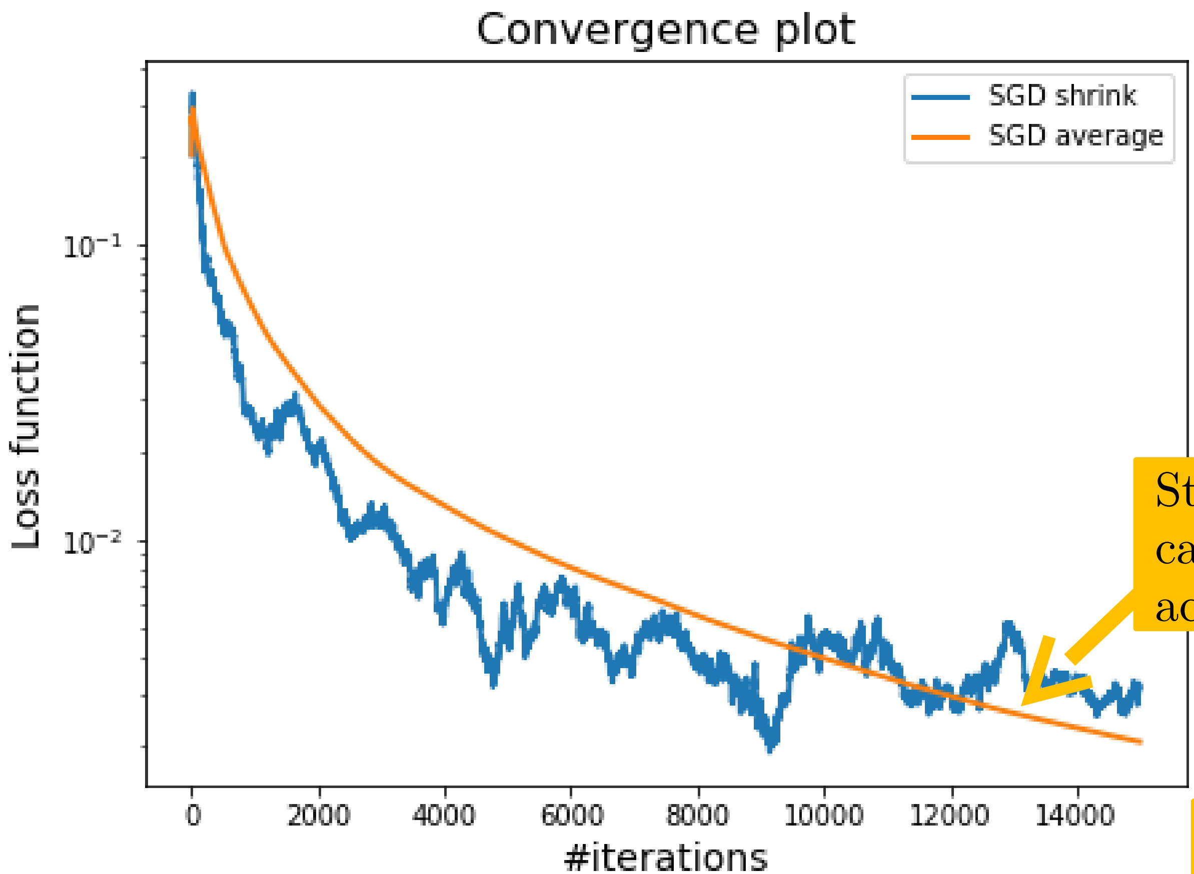
With and without averaging



Starts slow, but can reach higher accuracy

Stochastic Gradient Descent

With and without averaging

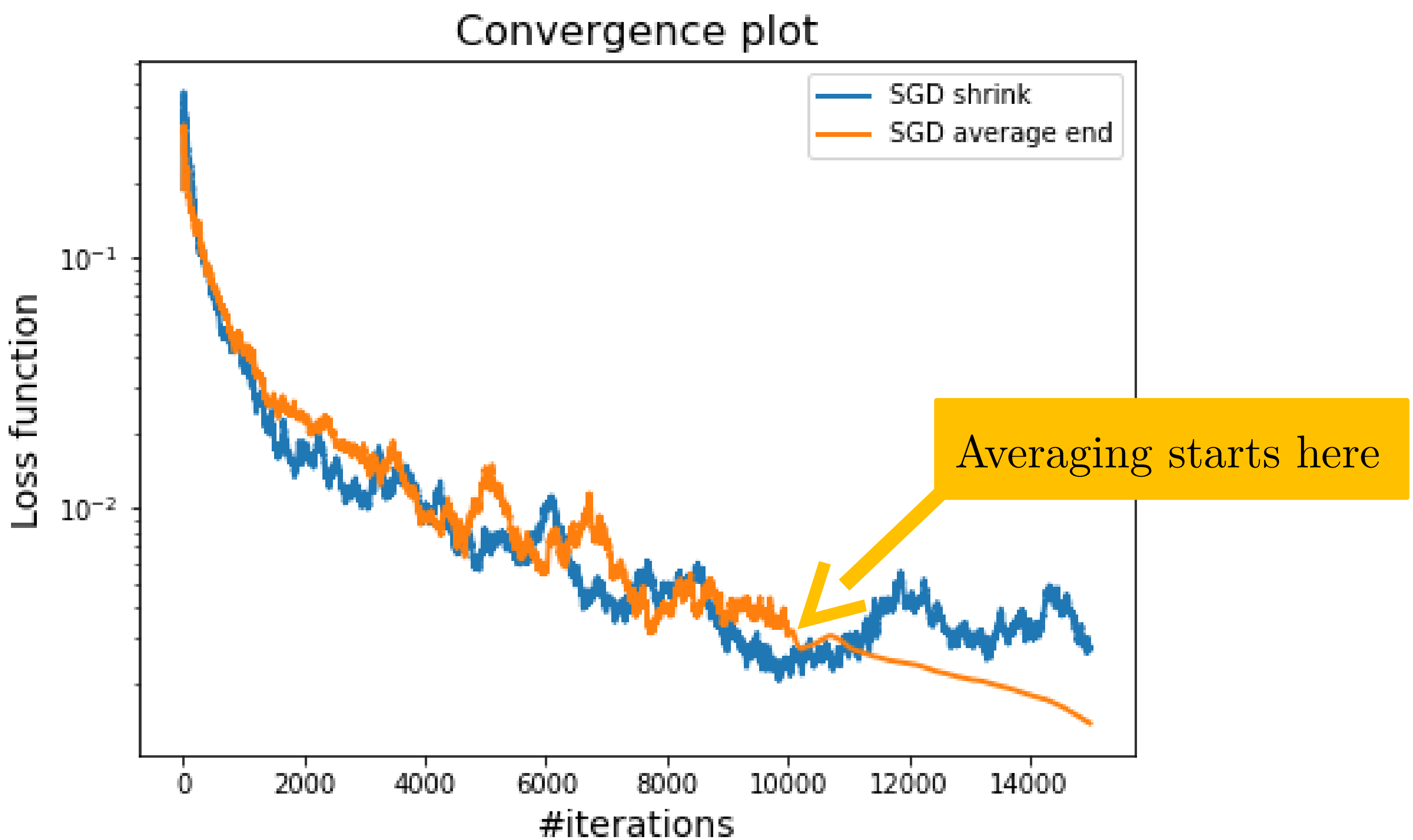


Starts slow, but can reach higher accuracy

Only use averaging towards the end?

Stochastic Gradient Descent

Averaging the last few iterates



Comparison GD and SGD for strongly convex

	SGD	**GD**
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$

Comparison GD and SGD for strongly convex

	SGD	**GD**
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$
Cost of an iteration	$O\left(1\right)$	$O\left(n\right)$

Comparison GD and SGD for strongly convex

	SGD	**GD**
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$
Cost of an iteration	$O(1)$	$O(n)$
Total complexity*	$O\left(\frac{1}{\epsilon}\right)$	$O\left(n\log\left(\frac{1}{\epsilon}\right)\right)$

Comparison GD and SGD for strongly convex

	SGD	**GD**
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$
Cost of an iteration	$O\left(1\right)$	$O\left(n\right)$
Total complexity*	$O\left(\frac{1}{\epsilon}\right)$	$O\left(n\log\left(\frac{1}{\epsilon}\right)\right)$

*Total complexity $=$ (Iteration complexity) $\times$ (Cost of an iteration)

Comparison GD and SGD for strongly convex

	SGD	**GD**
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$
Cost of an iteration	$O(1)$	$O(n)$
Total complexity*	$O\left(\frac{1}{\epsilon}\right)$	$O\left(n\log\left(\frac{1}{\epsilon}\right)\right)$

What happens if ϵ is small?

What happens if n is big?

*Total complexity $=$ (Iteration complexity) $\times$ (Cost of an iteration)

Why Machine Learners Like SGD

Why Machine Learners like SGD

Though we solve:

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right) + \lambda R(w)$$

We want to solve:

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

$$\min_{w \in \mathbf{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(h_w(x), y\right)\right]$$

SGD can solve the statistical learning problem!

Why Machine Learners like SGD

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

$$\min_{w \in \mathbf{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\ell\left(h_w(x), y\right)\right]$$

SGD ∞.0 for learning

Set $w^0 = 0$, $\alpha > 0$

for $t = 0, 1, 2, \ldots, T-1$

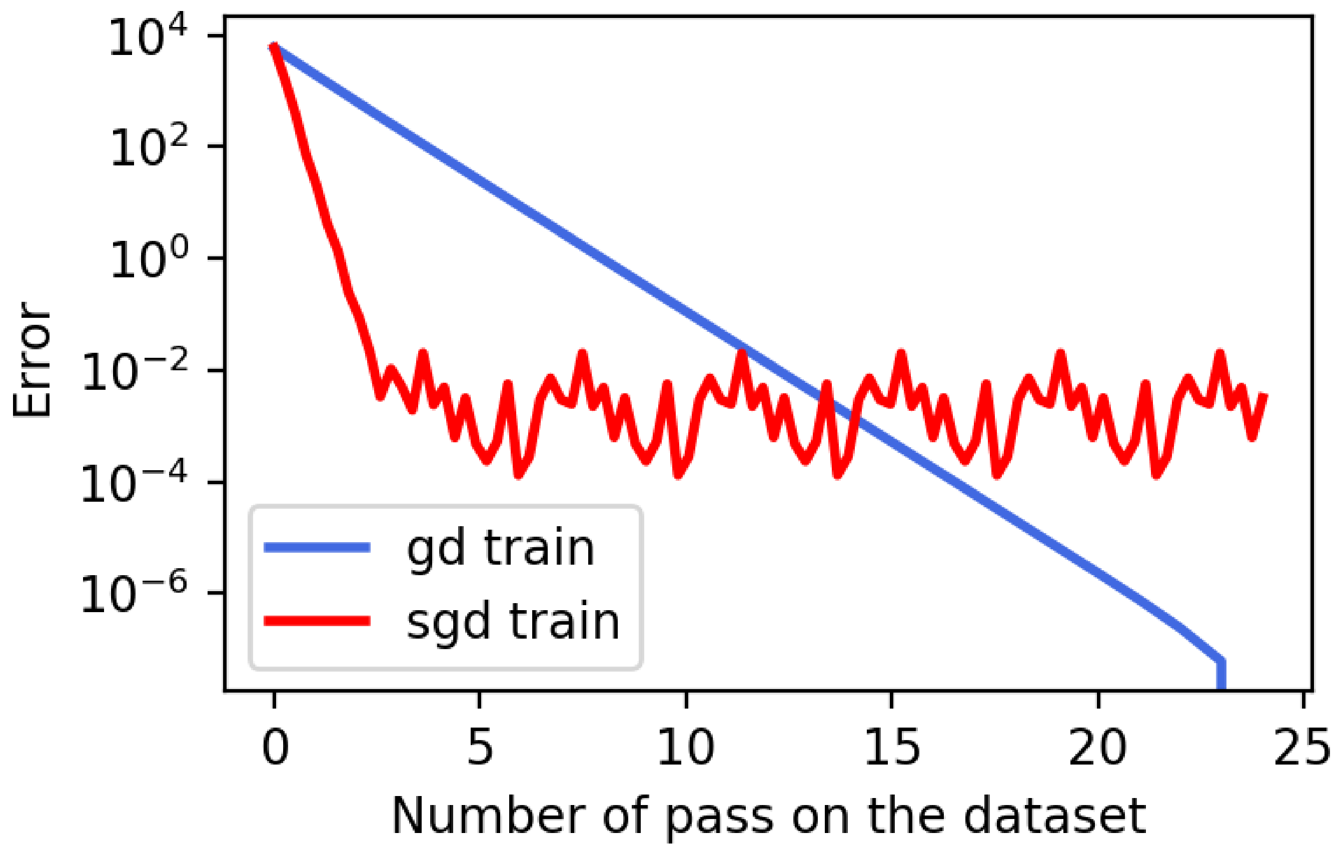
sample $(x, y) \sim \mathcal{D}$

calculate $v_t \in \partial\ell(h_{w^t}(x), y)$

$w^{t+1} = w^t - \alpha v_t$

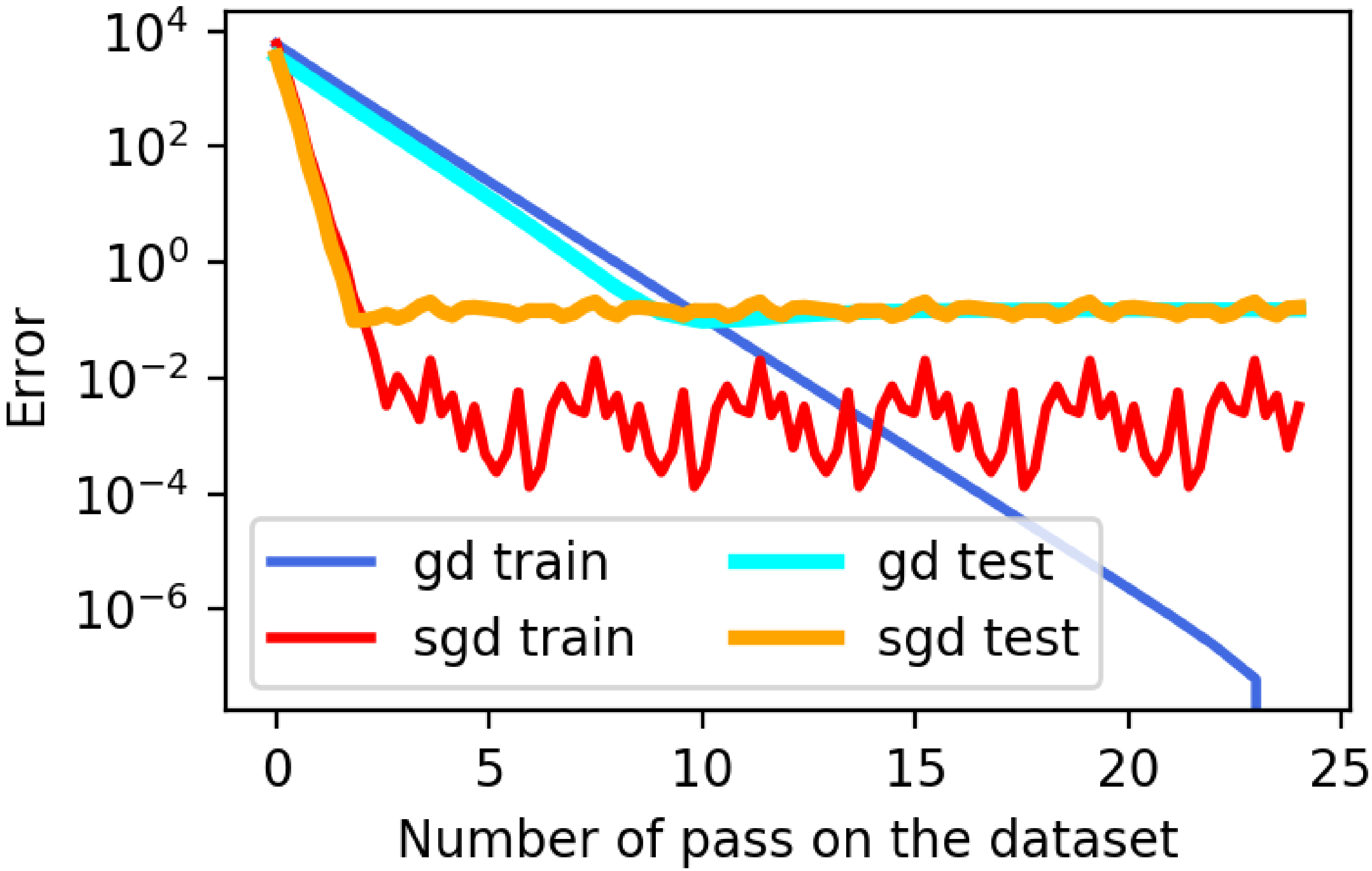
Output $\overline{w}^T = \frac{1}{T}\sum_{t=1}^{T} w^t$

Train error



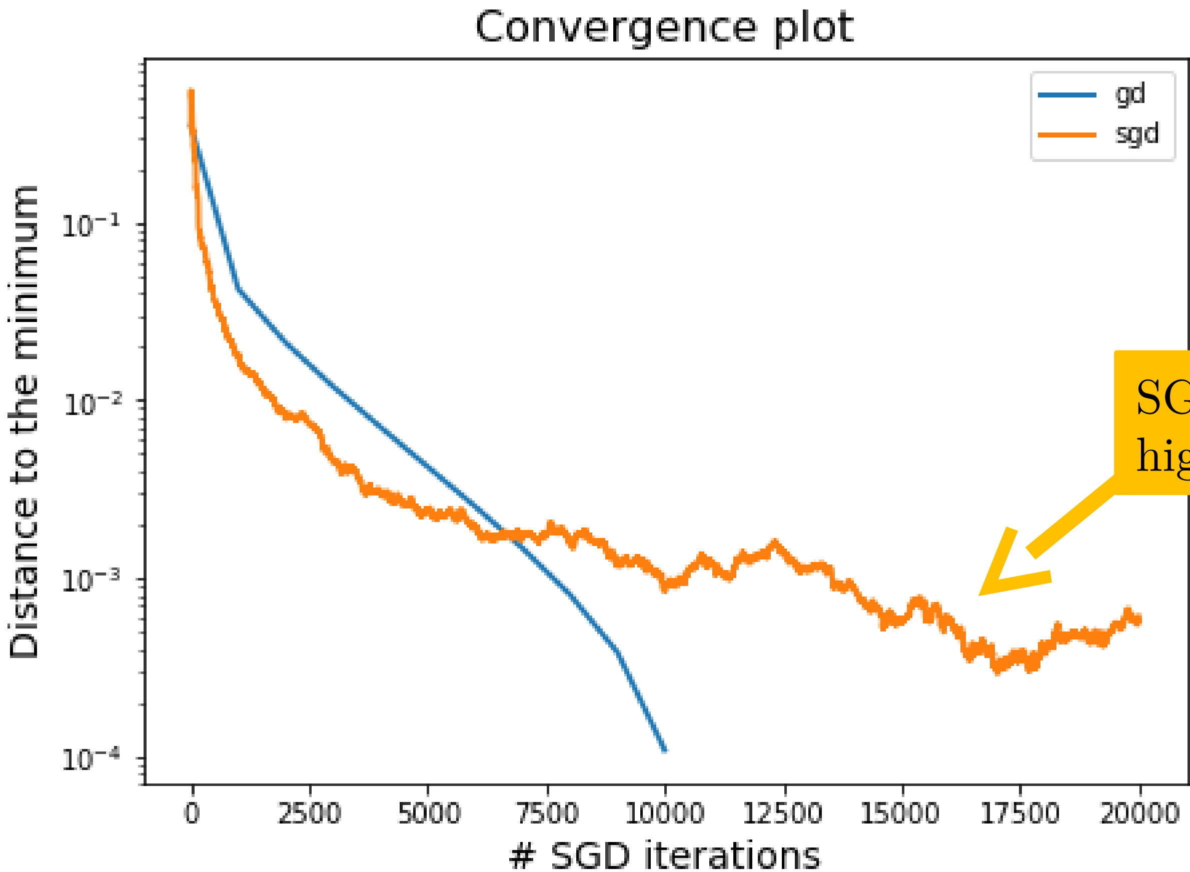
10^4
10^2
10^0
10^{-2}
10^{-4}
10^{-6}
Error
gd train
sgd train
0
5
10
15
20
25
Number of pass on the dataset

Train error and test error



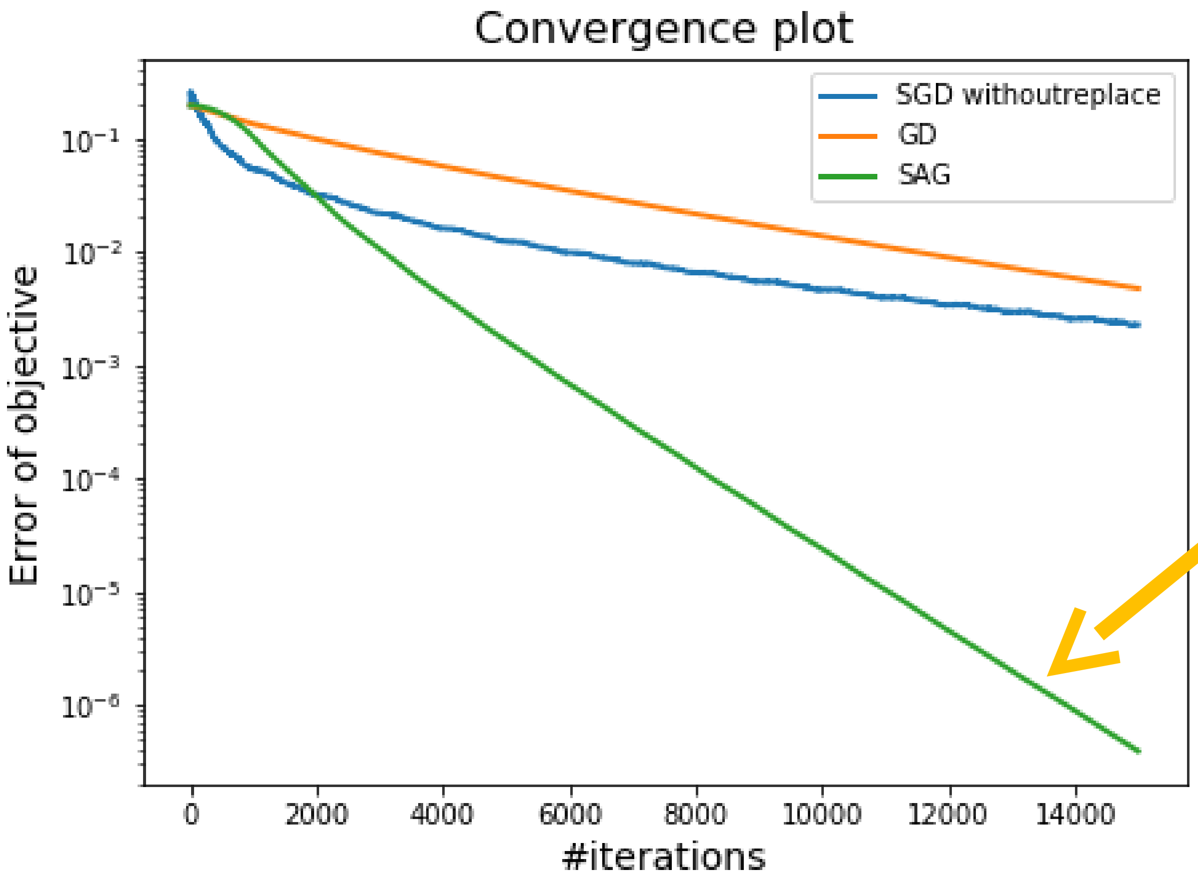
Variance reduction methods

SGD initially fast, slow later



SGD suffers from high variance

Can we get best of both?



Let's learn about methods like this one

Build an Estimate of the Gradient

Instead of using directly $\nabla f_j(w^t) \approx \nabla f(w^t)$
Use $\nabla f_j(w^t)$ to update estimate $g_t \approx \nabla f(w^t)$

$$w^{t+1} = w^t - \gamma g^t$$

We would like gradient estimate such that:

Typically unbiased
$\mathbf{E}[g^t] = \nabla f(w^t)$

Similar

$$g^t \approx \nabla f(w^t)$$

Solves problem of
$\alpha_t \underset{t\to\infty}{\longrightarrow} 0$

Converges in *L2*

$$\mathbb{E}||g^t||_2^2 \underset{w^t \to w^*}{\longrightarrow} 0$$

Controlled Stochastic Reformulation

Covariate functions:

$$z_i : w \mapsto z_i(w) \in \mathbb{R}, \quad \text{for } i = 1, \ldots, n$$

Cancel out

$$\frac{1}{n}\sum_{i=1}^{n} f_i(w) = \mathbb{E}[f_i(w)] = \mathbb{E}[f_i(w)] - \mathbb{E}[z_i(w)] + \mathbb{E}[z_i(w)]$$

$$i \sim \frac{1}{n} \qquad = \mathbb{E}\left[f_i(w) - z_i(w) + \mathbb{E}[z_i(w)]\right]$$

Original finite sum problem

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

$\Longleftrightarrow$

Controlled Stochastic Reformulation

$$\min_{w\in\mathbb{R}^d} \mathbb{E}\left[f_i(w) - z_i(w) + \mathbb{E}[z_i(w)]\right]$$

Use covariates to **control the variance**

Variance reduction as SGD on another function

$$\min_{w \in \mathbb{R}^d} \mathbb{E}\left[f_i(w) - z_i(w) + \mathbb{E}[z_i(w)]\right]$$

By design we have that $\mathbb{E}[g_i(w^t)] = \nabla f(w^t)$

Sample $i \sim \frac{1}{n}$

$$w^{t+1} = w^t - \gamma g_i(w^t)$$

$$g_i(w) := \nabla f_i(w) - \nabla z_i(w) + \mathbb{E}[\nabla z_i(w)]$$

How to choose $z_i(w)$?

Covariates

$$\mathrm{cov}(x,z) := \mathbb{E}[(x-\mathbb{E}[x])(z-\mathbb{E}[z])]$$

Let x and z be random variables. We say that x and z are covariates if:

$$\mathrm{cov}(x,z) \geq 0$$

Variance Reduced Estimate:

$$x_z = x - z + \mathbb{E}[z]$$

EXE:
1. Show that $\mathbb{E}[x_z] = \mathbb{E}[x]$
2. $\mathbb{VAR}[x_z] = \mathbb{E}[(x_z - \mathbb{E}[x_z])^2] = ?$
3. When is $\mathbb{VAR}[x_z] \leq \mathbb{VAR}[x]$

$$\begin{aligned}
\mathbb{E}[(x_z - \mathbb{E}[x_z])^2] &= \mathbb{E}[(x-\mathbb{E}[x] - (z-\mathbb{E}[z]))^2] \\
&= \mathbb{E}[(x-\mathbb{E}[x])^2] - 2\mathbb{E}[(x-\mathbb{E}[x])(z-\mathbb{E}[z])] \\
&\quad +\mathbb{E}[(z-\mathbb{E}[z])^2] \\
&= \mathbb{VAR}[x] - 2\mathrm{cov}(x,z) + \mathbb{VAR}[z]
\end{aligned}$$

Larger covariance between x and z is good

Covariates

$$\operatorname{cov}(x,z) := \mathbb{E}[(x-\mathbb{E}[x])(z-\mathbb{E}[z])]$$

Let x and z be random variables. We say that x and z are covariates if:

$$\operatorname{cov}(x,z) \geq 0$$

Variance Reduced Estimate:

$$x_z = x - z + \mathbb{E}[z]$$

$$g_i(w) := \underbrace{\nabla f_i(w)}_{x} - \underbrace{\nabla z_i(w)}_{z} + \underbrace{\mathbb{E}[\nabla z_i(w)]}_{\mathbb{E}[z]}$$

$$\nabla z_i(w) \approx \nabla f_i(w) \quad \Rightarrow \quad \operatorname{cov}(\nabla z_i(w), \nabla f_i(w))$$

Choosing the covariate as a linear approximation

$$\text{Sample } i \sim \frac{1}{n}$$

$$w^{t+1} = w^t - \gamma g_i(w^t) := \nabla f_i(w) - \nabla z_i(w) + \mathbb{E}[\nabla z_i(w)]$$

We would like: $\nabla z_i(w) \approx \nabla f_i(w)$

Linear approximation around w

$$z_i(w) = f_i(\tilde{w}) + \langle \nabla f_i(\tilde{w}), w - \tilde{w} \rangle$$

A reference point/ snap shot

SVRG: Stochastic Variance reduced method gradient

Johnson & Zhang, 2013 NIPS

$$w^{t+1} = w^t - \gamma g_i(w^t)$$

Reference point	$\tilde{w} \in \mathbb{R}^d$
Sample	$\nabla f_i(w^t), \qquad$ i.i.d sample with prob $\frac{1}{n}$
Grad. estimate	$g_i(w^t) = \nabla f_i(w^t) - \nabla f_i(\tilde{w}) + \nabla f(\tilde{w})$

It's unbiased because:

$$\begin{aligned}\mathbb{E}[g_i(w)] &= \mathbb{E}[\nabla f_i(w)] - \mathbb{E}[\nabla f_i(\tilde{w})] + \nabla f(\tilde{w}) \\ &= \nabla f(w) - \cancel{\nabla f(\tilde{w})} + \cancel{\nabla f(\tilde{w})}\end{aligned}$$

free-SVRG: Stochastic Variance Reduced Gradients

PDF Adobe Jonhson & Zhang NIPS 2013

PDF Adobe Sebbouh, et. al 2019 Neurips 2019

Set $\tilde{w}^0 = 0 = x_0^m$, choose $\gamma > 0, m \in \mathbb{N}$,

$\alpha_t > 0$ with $\sum_{t=0}^{m-1} \alpha_t = 1$

for $s = 1, 2, \ldots, T$

$x_s^0 = x_{s-1}^m$

for $t = 0, 1, 2, \ldots, m-1$

i.i.d sample $i \sim \frac{1}{n}$

$g^t = \nabla f_i(x_s^t) - \nabla f_i(\tilde{w}^{s-1}) + \nabla f(\tilde{w}^{s-1})$

$x_s^{t+1} = x_s^t - \gamma g^t$

$\tilde{w}^{s+1} = \sum_{t=0}^{m-1} \alpha_t x_s^t$

Output $\tilde{w}^{T+1}$

Adding indices in t and s

Reference point is an average of inner iterates

Most iterates cost $O(1)$

Tune inner loop size m

SAGA: Stochastic Average Gradient

Defazio, Bach, & Lacoste-Julien, 2014 NIPs

$$w^{t+1} = w^t - \gamma g_i(w^t)$$

Sample

$\nabla f_i(w^t)$, i.i.d sample with prob $\frac{1}{n}$

Grad. estimate

$$g_i(w^t) = \nabla f_i(w^t) - \nabla f_i(w^{t_i}) + \frac{1}{n}\sum_{j=1}^{n} \nabla f_j(w^{t_j})$$

$z_i(w) = f_i(w^{t_i}) + \langle \nabla f_i(w^{t_i}), w - w^{t_i} \rangle$

$\nabla z_i(w^t) = \nabla f_i(w^{t_i})$

$\mathbb{E}[\nabla z_i(w^t)]$

Store grad.

$$\nabla f_i(w^{t_i}) = \nabla f_i(w^t)$$

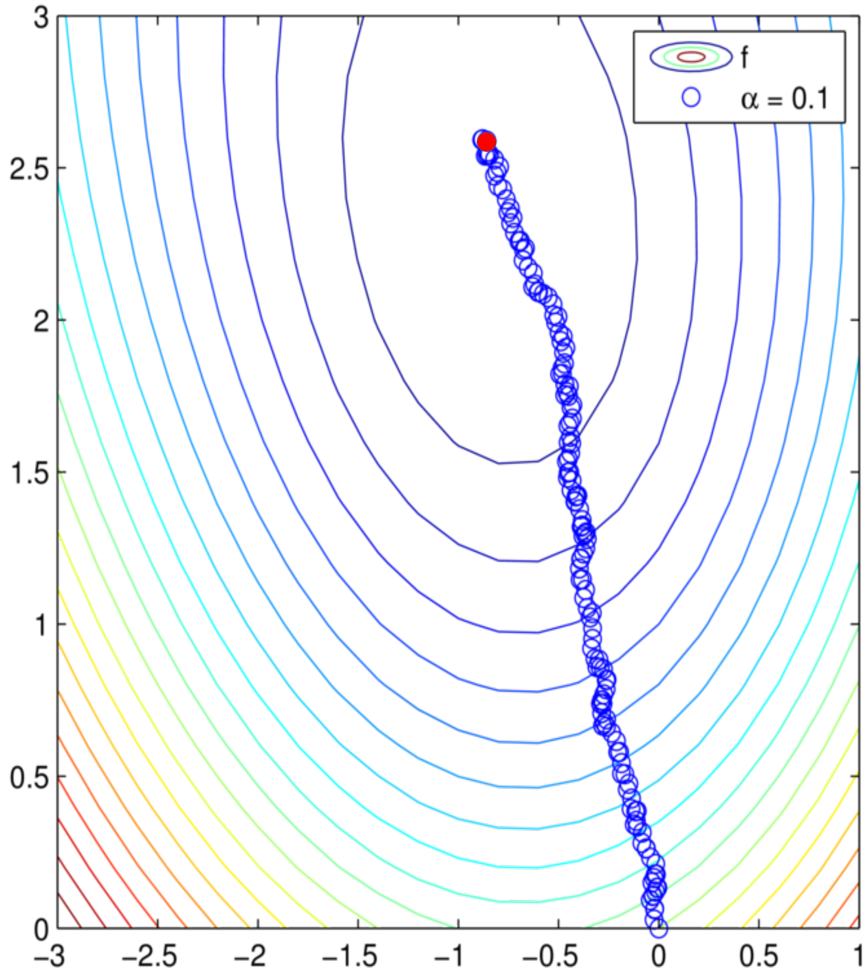
SAGA: Stochastic Average Gradient

$$
\begin{array}{l}
\text{Set } w^0 = 0, g_i = \nabla f_i(w^0), \ \text{ for } i = 1\ldots, n \\
\text{Choose } \gamma > 0 \\
\text{for } t = 0, 1, 2, \ldots, T-1 \\
\qquad \text{sample } i \in \{1, \ldots, n\} \\
\qquad g^t = \nabla f_i(w^t) - g_i + \frac{1}{n}\sum_{j=1}^{n} g_j \\
\qquad w^{t+1} = w^t - \gamma g^t \\
\qquad g_i = \nabla f_i(w^t) \\
\text{Output } w^T
\end{array}
$$

No inner loop, rolling update

Stores a $d \times n$ matrix

The Stochastic Average Gradient



How to prove this converges? Is this the only option?

Convergence Theorems

Assumptions for Convergence

Strong Convexity

$$f(w) \geq f(y) + \langle \nabla f(y), w - y \rangle + \frac{\mu}{2}||w - y||_2^2$$

Smoothness + convexity

$$f_i(w) \leq f_i(y) + \langle \nabla f_i(y), w - y \rangle + \frac{L_i}{2}||w - y||_2^2$$

$$f_i(w) \geq f_i(y) + \langle \nabla f_i(y), w - y \rangle \qquad \text{for } i = 1, \ldots, n$$

$$L_{\max} := \max_{i=1,\ldots,n} L_i$$

Convergence SAGA

Theorem SAGA

If $f(w)$ is μ–strongly convex, $f_i(w)$ is $L_{\max}$–smooth and $\alpha = 1/(3L_{\max})$ then

$$\mathbb{E}\left[||w^t - w^*||_2^2\right] \leq \left(1 - \min\left\{\frac{1}{4n}, \frac{\mu}{3L_{\max}}\right\}\right)^t C_0$$

where $C_0 = \dfrac{2n}{3L_{\max}}(f(w^0) - f(w^*)) + ||w^0 - w^*||_2^2 \geq 0$

An even more practical convergence result!

Difficult proof technique

A. Defazio, F. Bach and J. Lacoste-Julien (2014) NIPS, **SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives.**

Comparisons in total complexity for strongly convex

Approximate solution

$$\mathbb{E}[f(w^T)] - f(w^*) \leq \epsilon \quad \text{or} \quad \mathbb{E}\|w^t - w^*\|^2 \leq \epsilon$$

SGD

$$O\left(\frac{1}{\epsilon}\right)$$

Gradient descent

$$O\left(\frac{nL}{\mu}\log\left(\frac{1}{\epsilon}\right)\right)$$

SVRG/SAGA/SAG

$$O\left(\left(n + \frac{L_{\max}}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)\right)$$

Variance reduction faster than GD when

$$L \geq \mu + L_{\max}/n$$

Practicals implementation of SAG for Linear Classifiers

L2 regularizer + linear hypothesis

Finite Sum Training Problem

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(\langle w, x^i \rangle, y^i\right) + \frac{\lambda}{2}||w||_2^2$$

$$\nabla f_i(w) = \underbrace{\ell'(\langle w, x^i \rangle, y^i)}_{\text{Nonlinear in } w} x^i + \underbrace{\lambda w}_{\text{Linear in } w}$$

Reduce Storage to $O(n)$

Only store real number

$$\beta_i = \ell'(\langle w^{t_i}, x^i \rangle, y^i)$$

Stoch. gradient estimate

$$\nabla f_i(w^{t_i}) = \beta_i x^i + \lambda w^t$$

Full gradient estimate

$$g^t = \frac{1}{n} \sum_{j=1}^{n} \beta_j x_j + \lambda w^t$$

Take for home Variance Reduction

- Variance reduced methods use only **one stochastic gradient per iteration** and converge linearly on strongly convex functions

- Choice of **fixed stepsize** possible

- **SAGA** only needs to know the smoothness parameter to work, but requires storing n past stochastic gradients

- **SVRG** only has $O(d)$ storage, but requires full gradient computations every so often. Has an extra “number of inner iterations” parameter to tune