Gradient Descent

$$\min_x f(x)$$

Gradient descent

Differentiable f :

$$x_t = x_{t-1} - \eta_t \nabla f(x_{t-1})$$

Subgradient method

Subdifferentiable f : $g_t \in \partial f(x_{t-1})$

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

Proximal Point Algorithm

Subgradient Method : Equivalent formulation

Subdifferentiable f : $g_t \in \partial f(x_{t-1})$

$$x_t = \arg\min_x \left\{ \langle x, g_t \rangle + \frac{1}{2\eta_t} \|x - x_{t-1}\|^2 \right\}$$

Proximal Point Algorithm

Subdifferentiable f

$$x_t = \arg\min_x \left\{ \boldsymbol{f}(x) + \frac{1}{2\eta_t} \|x - x_{t-1}\|^2 \right\}$$

Proximal Point Algorithm

$$\min_x f(x) + \psi(x)$$

Proximal Gradient Descent

Subdifferentiable f : $g_t \in \partial f(x_{t-1})$

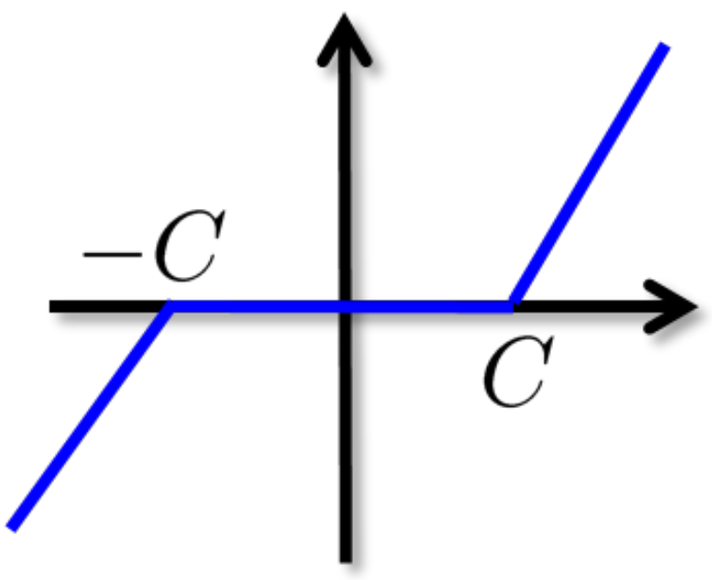
$$\begin{aligned} x_t &= \arg\min_x \left\{ \langle x, g_t \rangle + \psi(x) + \frac{1}{2\eta_t} \|x - x_{t-1}\|^2 \right\} \\ &= \arg\min_x \left\{ \eta_t \psi(x) + \frac{1}{2} \|x - (x_{t-1} - \eta_t g_t)\|^2 \right\} \\ &= \text{prox}(x_{t-1} - \eta_t g_t | \eta_t \psi) \end{aligned}$$

$$\text{prox}(u|h) := \underset{x}{\text{argmin}} \{ h(x) + \tfrac{1}{2} \|x - u\|^2 \}$$

Proximal Operator: L1

$$\min_x f(x) + \boldsymbol{\psi}(x)$$

$$L_1 \text{ Regularization : } \psi(x) = C\|x\|_1 = C\sum_i |x_i|$$



$$\text{prox}(u \,|\, C\|\cdot\|_1) = \begin{bmatrix} \vdots \\ \mathbf{ST}_C(u_i) \\ \vdots \end{bmatrix}$$

$$\mathbf{ST}_C(u) = \begin{cases} 0, & |u| \le C \\ \text{sign}(u)\max(|u| - C, 0), & |u| > C \end{cases}$$

Proximal Operator: trace norm

$$\min_X f(X) + \boldsymbol{\psi}(X)$$

Trace Norm Regularization : $\psi(X) = C\|X\|_{\mathrm{tr}} = C\sum_j \sigma_j(X)$

$$\mathrm{prox}(Y|\alpha\|\cdot\|_{\mathrm{tr}}) = U \begin{bmatrix} \mathbf{ST}_C(\sigma_1(Y)) & & 0 \\ & \ddots & \\ 0 & & \mathbf{ST}_C(\sigma_d(Y)) \end{bmatrix} V$$

$$\text{where } Y = U\Sigma V$$

Convergence of Proximal GD

$$\min_X f(X) + \boldsymbol{\psi}(X)$$

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

property of f	μ-Strongly convex	non-strongly conv
γ-Smooth	$\exp\left(-t\frac{\mu}{\gamma}\right)$	$\frac{\gamma}{t}$
Non-smooth	$\frac{1}{\mu t}$	$\frac{1}{\sqrt{t}}$

Nesterov's Acceleration

$$\min_X f(X) + \boldsymbol{\psi}(X)$$

Nesterov Accelerated Gradient

f γ-smooth. Set $s_1 = 1$ and $\boldsymbol{\eta} = \frac{\mathbf{1}}{\boldsymbol{\gamma}}$. Set y_0. Iterate:

- $g_t \in \partial f(y_t)$
- $x_t = \text{prox}(y_t - \boldsymbol{\eta} g_t | \boldsymbol{\eta}\psi)$
- $s_{t+1} = \frac{1+\sqrt{1+4s_t^2}}{2}$
- $y_t = x_t + \frac{s_t - 1}{s_{t+1}}(x_t - x_{t-1})$

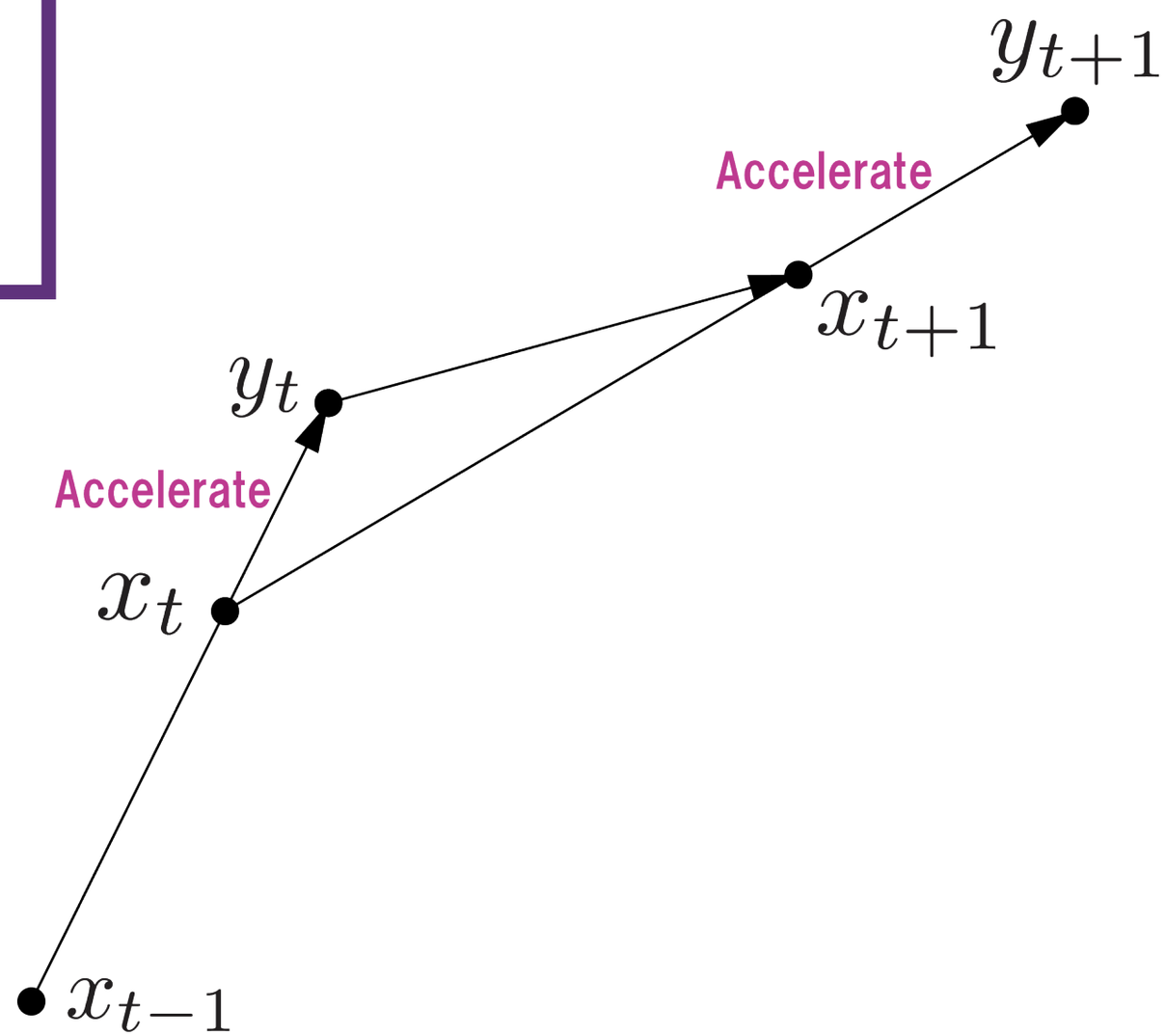
a.k.a FISTA

Nesterov's Acceleration

f γ-smooth. Set $s_1 = 1$ and $\boldsymbol{\eta = \frac{1}{\gamma}}$. Set y_0. Iterate:

- $g_t \in \partial f(y_t)$
- $x_t = \mathrm{prox}(y_t - \boldsymbol{\eta} g_t | \boldsymbol{\eta}\psi)$
- $s_{t+1} = \frac{1+\sqrt{1+4s_t^2}}{2}$
- $y_t = x_t + \frac{s_t - 1}{s_{t+1}}(x_t - x_{t-1})$

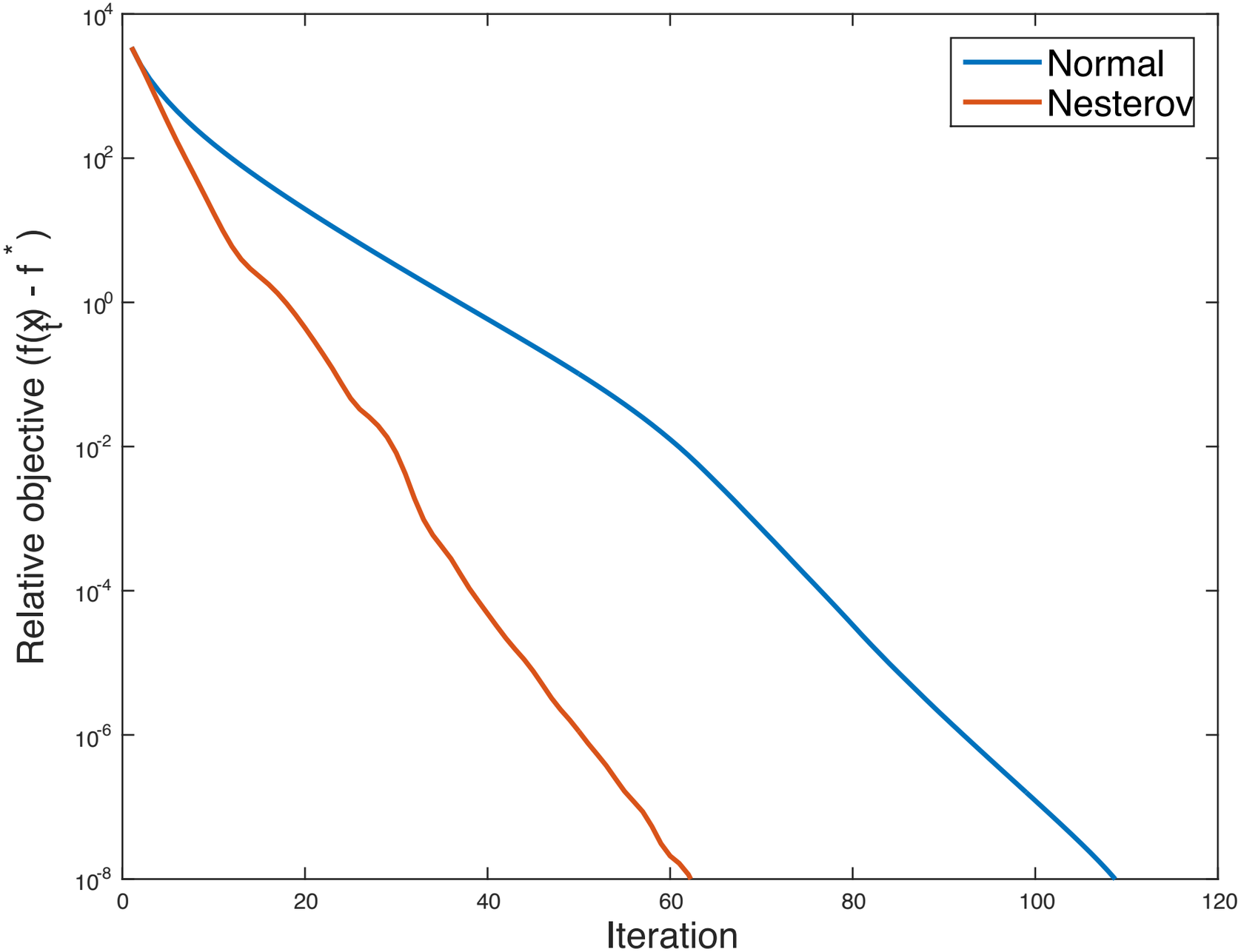
$$f(x_t) - f(x^*) \leq \frac{2\gamma \|x_t - x^*\|}{t^2}$$



source: T. Suzuki

Nesterov's Acceleration

$$\min_\theta \frac{1}{n}\sum_i (\theta^T x_i - y_i)^2 + \lambda\|\theta\|_1$$



source: T. Suzuki

Stochastic Gradient

We want to minimize

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{\boldsymbol{p}}} L(\boldsymbol{\theta}) := \mathbb{E}[\boldsymbol{l}_{\boldsymbol{\theta}}(Z)]$$

Due to practical constraints, samples only come **one by one**, each at a time t, and **cannot be stored**. Only previous parameter is stored. We use a double approximation

$$\begin{aligned}\mathbb{E}[\boldsymbol{l}(\boldsymbol{\theta}, Z)] &\approx \boldsymbol{l}(\boldsymbol{\theta}, z_t) \\ &\approx \boldsymbol{l}(\theta_{t-1}, z_t) + \langle \nabla \boldsymbol{l}(\theta_{t-1}, z_t), \boldsymbol{\theta} \rangle\end{aligned}$$

Stochastic Gradient

To approximate the minimization of

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{\boldsymbol{p}}} \mathbb{E}[\boldsymbol{l}_{\boldsymbol{\theta}}(Z)]$$

we use the approximated problem, only valid around the previous iterate

$$\theta_t := \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^{\boldsymbol{p}}} \langle \nabla \boldsymbol{l}(\theta_{t-1}, z_t), \boldsymbol{\theta} \rangle + \frac{1}{2\eta_t} \|\theta_{t-1} - \boldsymbol{\theta}\|^2$$

SG (**no** regularization)

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{\boldsymbol{p}}} L(\boldsymbol{\theta}) := \mathbb{E}[\boldsymbol{l}_{\boldsymbol{\theta}}(Z)]$$

Stochastic Gradient Method (regularization)

Set θ_0 and sequence η_t. Repeat:

Sample $z_t \sim P(Z)$.

Compute subgradient $g_t \in \partial_\theta \boldsymbol{l}(\theta, z_t)$

Update $\theta_t = \theta_{t-1} - \eta_t g_t$

Output : $\bar{\theta}_T = \frac{1}{T+1} \sum_{t=0}^{T} \theta_t$

SG (regularization)

We want to minimize now:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{\boldsymbol{p}}} L_\psi(\boldsymbol{\theta}) := \mathbb{E}[\boldsymbol{l}_{\boldsymbol{\theta}}(Z)] + \psi(\boldsymbol{\theta})$$

Stochastic Gradient Method (regularization)

Set θ_0 and sequence η_t. Repeat:

Sample $z_t \sim P(Z)$.

Compute subgradient $g_t \in \partial_\theta \boldsymbol{l}(\theta, z_t)$

Update $\theta_t = \text{prox}(\theta_{t-1} - \eta_t g_t \,|\, \boldsymbol{\eta_t}\psi)$

Output : $\bar{\theta}_T = \frac{1}{T+1} \sum_{t=0}^{T} \theta_t$

Polynomial Averaging

Stochastic Gradient Method (regularization)

Set θ_0 and sequence η_t. Repeat:

Sample $z_t \sim P(Z)$.

Compute subgradient $g_t \in \partial_\theta \boldsymbol{l}(\theta, z_t)$

Update $\theta_t = \text{prox}(\theta_{t-1} - \eta_t g_t \,|\, \boldsymbol{\eta_t} \psi)$

Output : $\bar{\theta}_T = \frac{2}{(T+1)(T+2)} \sum_{t=0}^{T} (t+1)\theta_t$

Batch Methods

- SGMethods have several drawbacks, chief among them is the choice of a stepsize.
- Is there a setting where this can be mitigated? Yes, when the expectation is in fact a large sum:

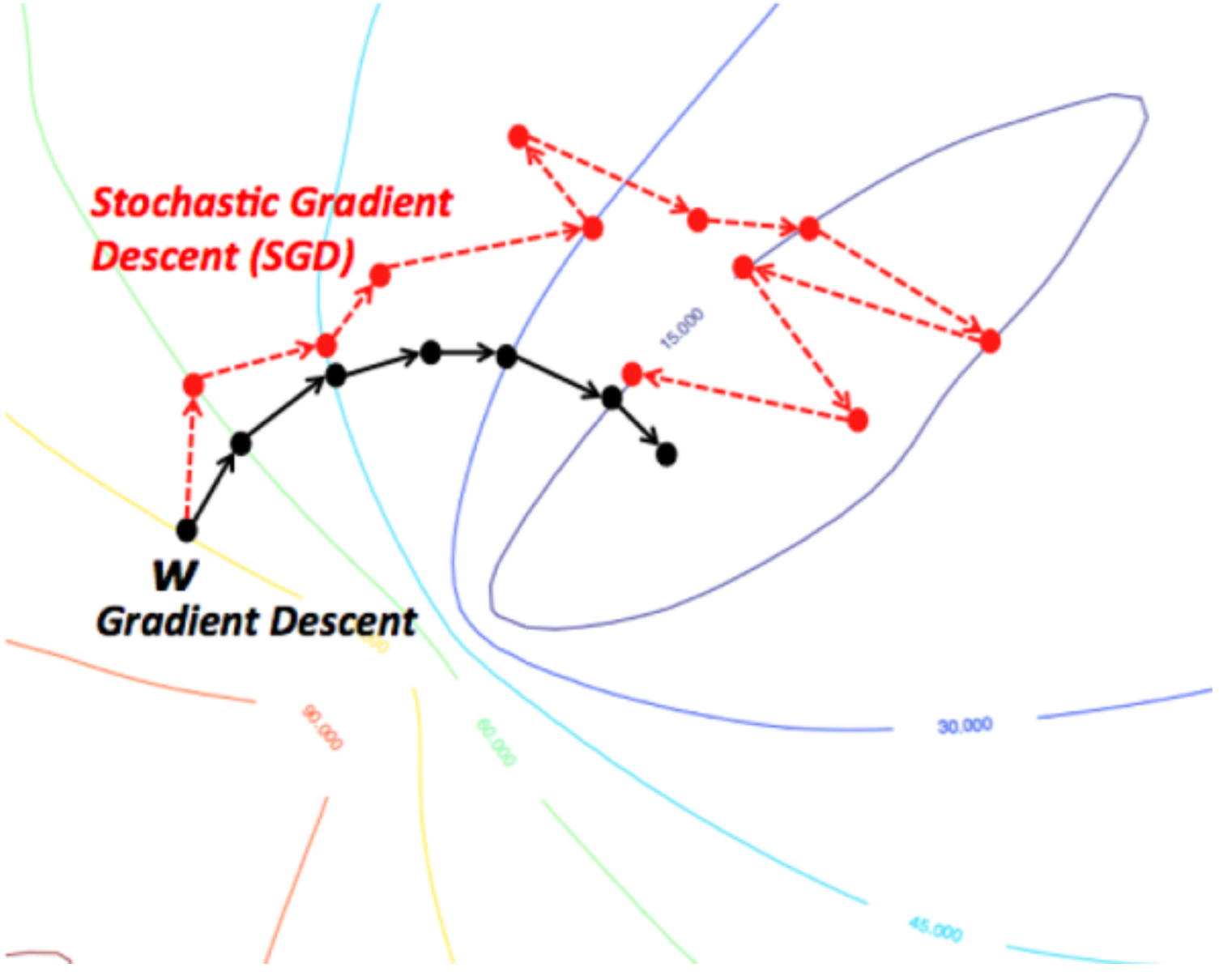
$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{\boldsymbol{p}}} L_\psi(\boldsymbol{\theta}) := \mathbb{E}[\boldsymbol{l}_{\boldsymbol{\theta}}(Z)] + \psi(\boldsymbol{\theta})$$

$$\Downarrow$$

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{\boldsymbol{p}}} \frac{1}{n}\sum_{i=1}^{n} l(\boldsymbol{\theta}, z_i) + \psi(\boldsymbol{\theta})$$

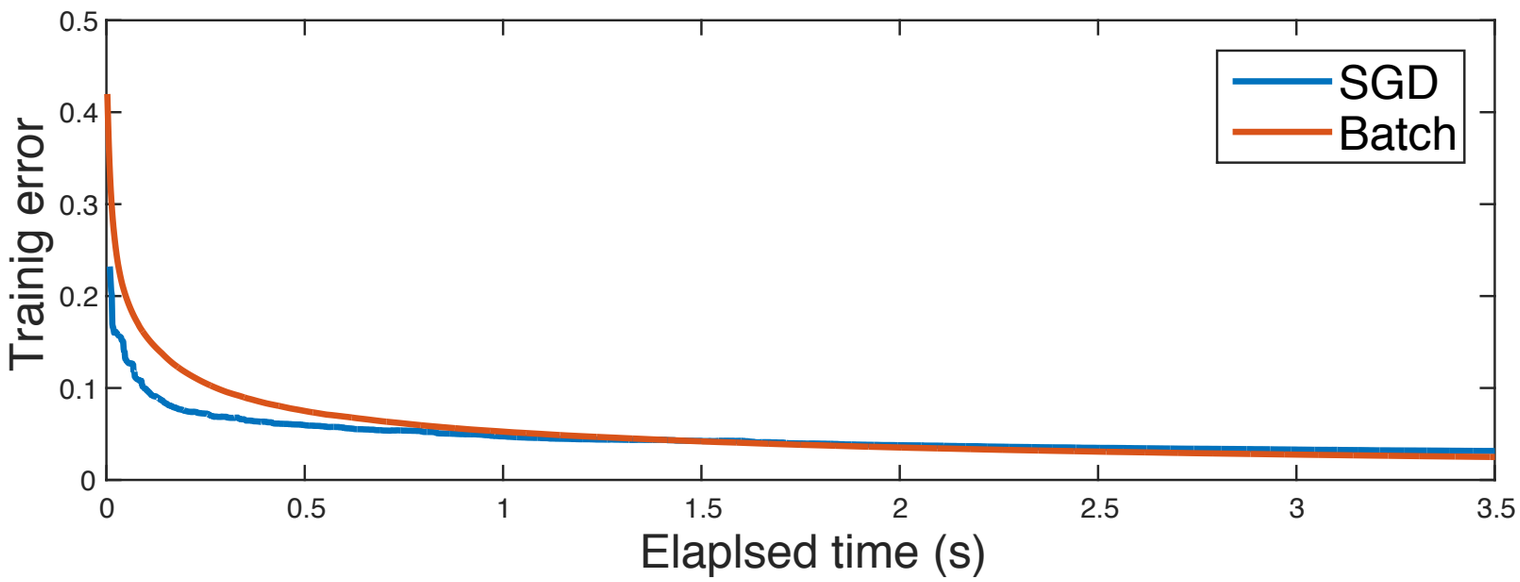
Batch Methods

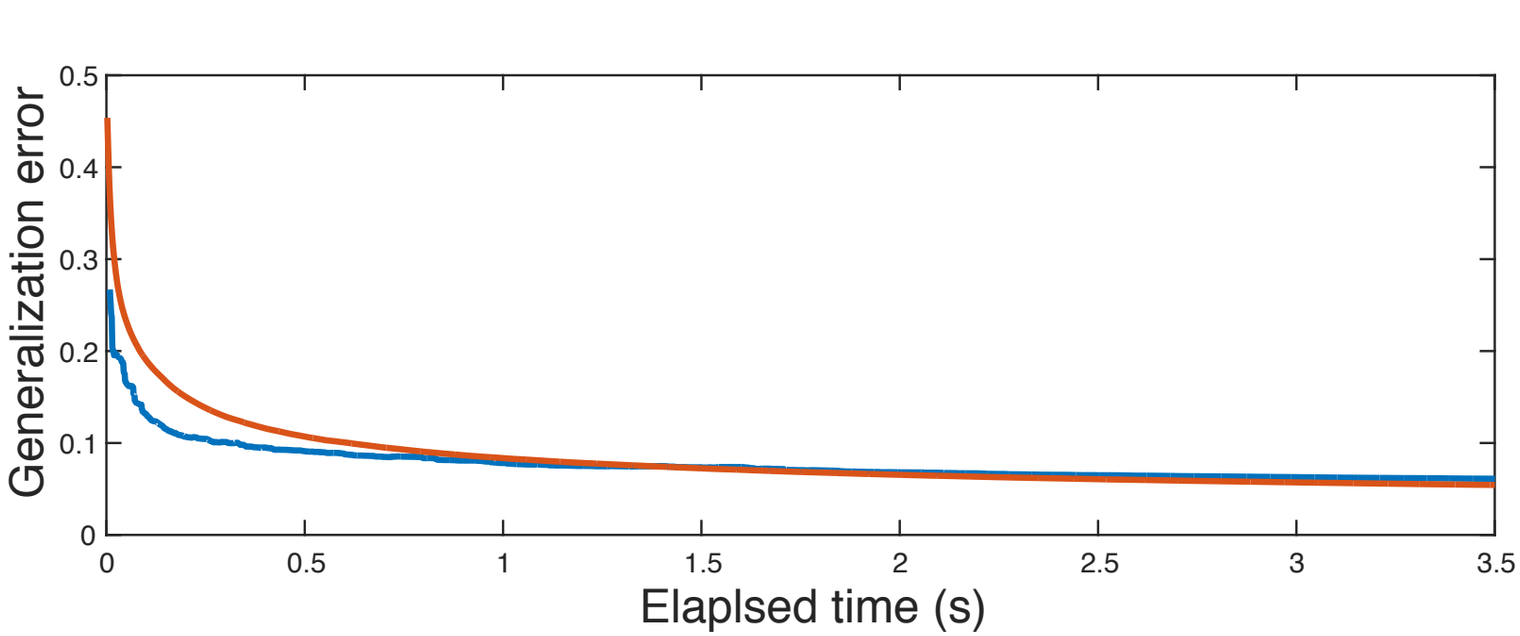
- We would like to have the benefits of SGM (low cost per iteration) without the disadvantages (slow convergence near optimum, step size selection)



source: https://wikidocs.net/3413

Batch Methods





Logistic Regression L1 regularization

source: T. Suzuki

Three Methods

- Primal methods
  - **Stochastic Average Gradient (A) descent, SAG(A)** *(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)*
- Dual methods (see Fenchel duality)
  - **Stochastic Dual Coordinate ascent**, SDCA *(Shalev-Shwartz and Zhang, 2013a)*