Nesterov's Acceleration

$$\min_X f(X) + \psi(X)$$

Nesterov Accelerated Gradient

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

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

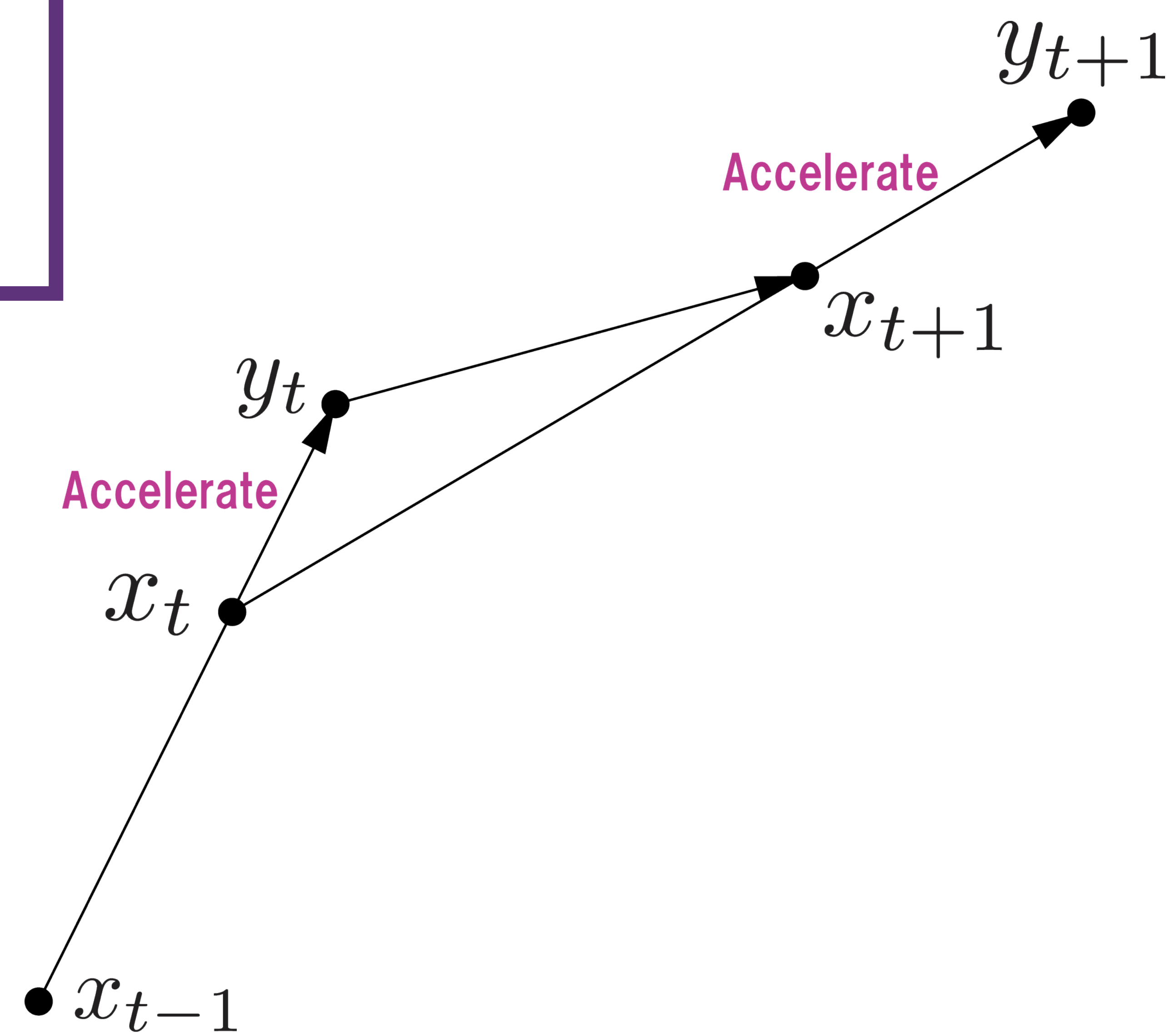
a.k.a
FISTA

Nesterov's Acceleration

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

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

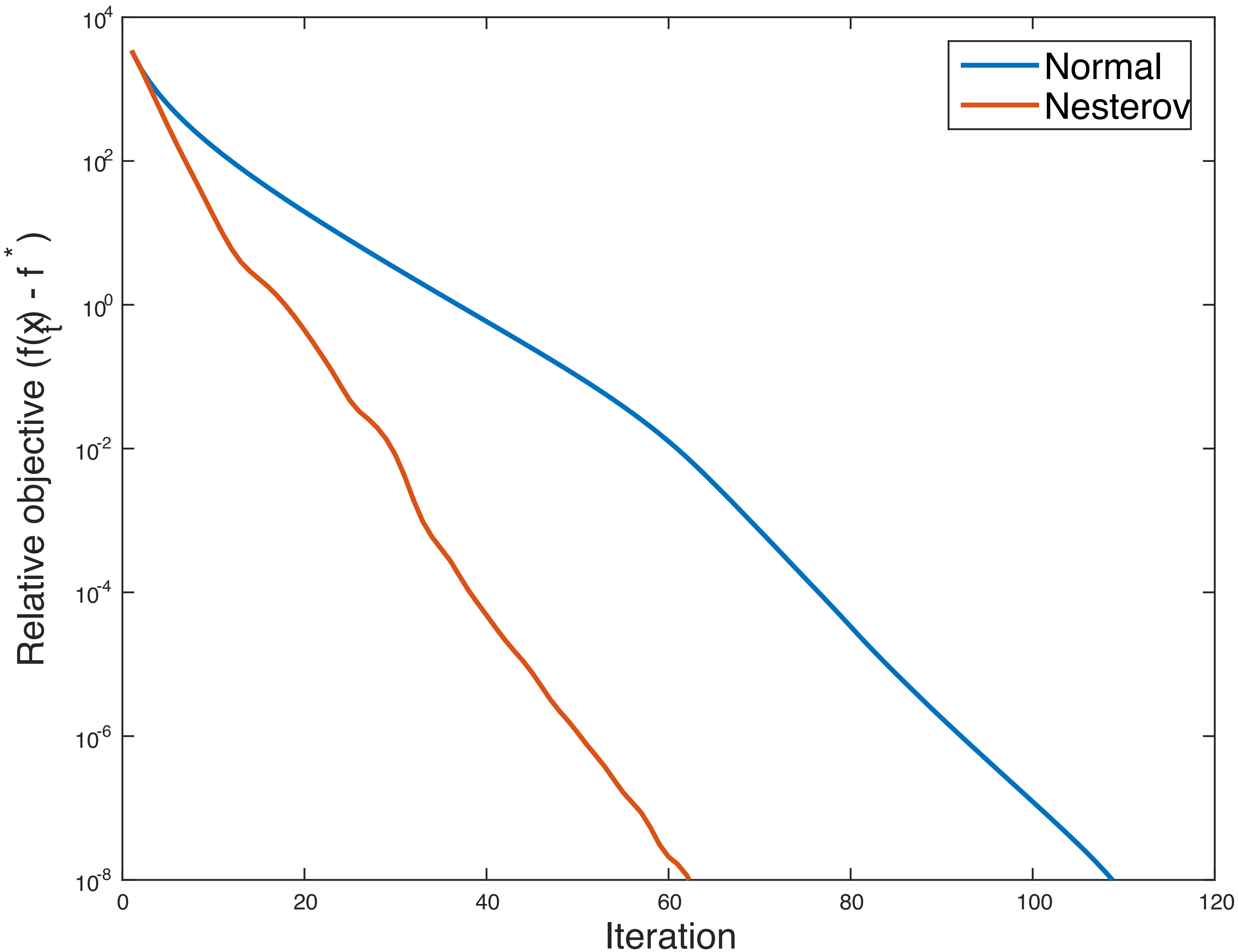
$$f(x_t) - f(x^*) \leq \frac{2\gamma \|x_t - x^*\|}{t^2}$$



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:

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

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

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

Batch Problems

- 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:

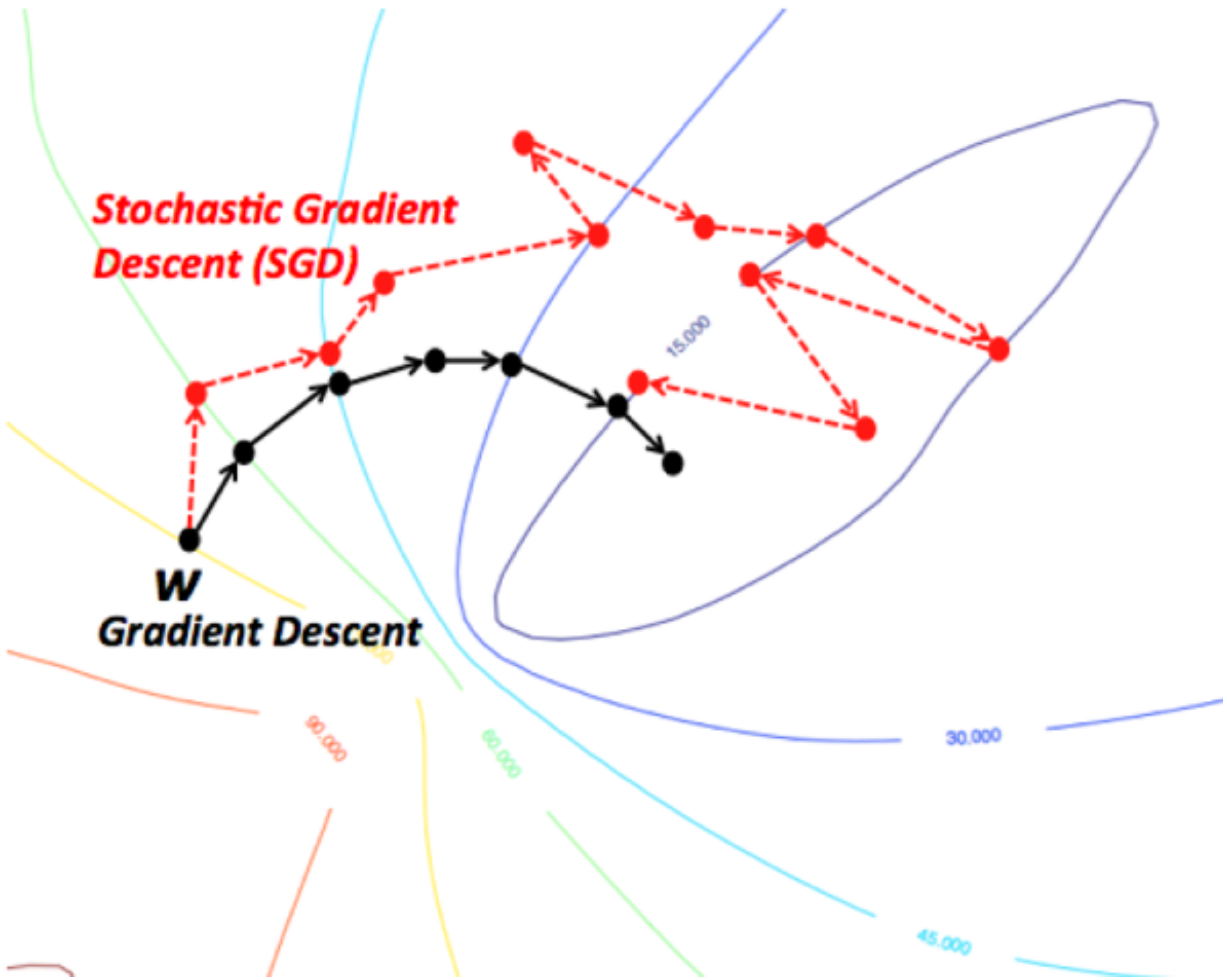
$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{\boldsymbol{p}}} L_{\psi}(\boldsymbol{\theta}) := \mathbb{E}[\boldsymbol{l_{\theta}}(Z)] + \psi(\boldsymbol{\theta})$$

$$\Downarrow$$

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

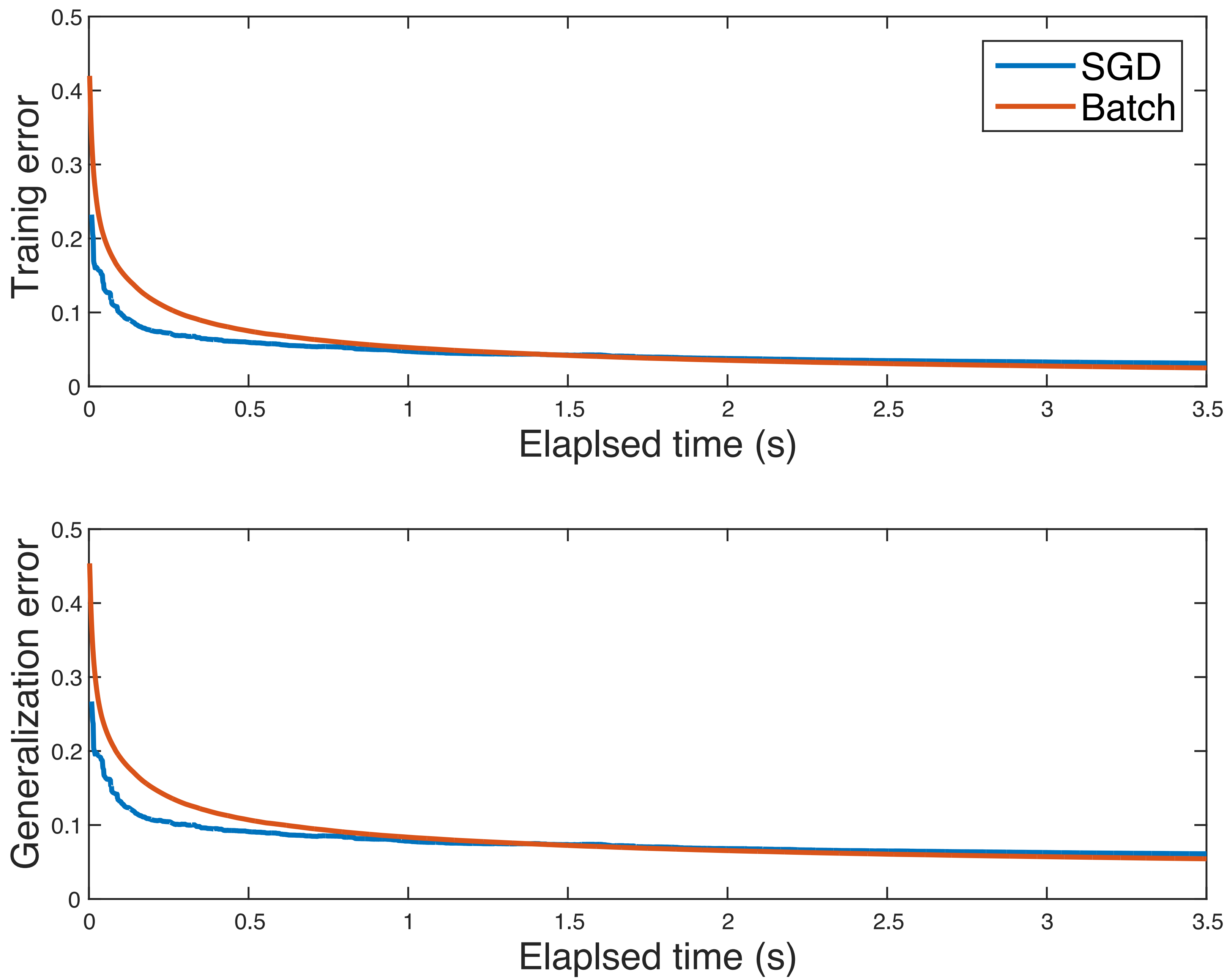
Batch Methods

- 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)*

Primal Methods

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

smooth **strongly convex**

$$\boxed{\text{We want to approximate } \nabla \frac{1}{n} \sum_{i=1}^{n} l_i(\boldsymbol{\theta})}$$

Primal Methods

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

smooth **strongly convex**

$$\boxed{\text{We want to approximate } \nabla \tfrac{1}{n} \textstyle\sum_{i=1}^{n} l_i(\boldsymbol{\theta})}$$

$$\mathbb{E}_{i \sim \mathrm{unif}\{1,\ldots,\mathrm{n}\}}[\nabla l_i(\boldsymbol{\theta})] = \frac{1}{n} \sum_{i} \nabla l_i(\boldsymbol{\theta}) = \nabla \mathcal{L}(\boldsymbol{\theta})$$

Randomizing points in the dataset gives a way to get an unbiased estimator of the gradient.

Primal Methods

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

smooth **strongly convex**

$$\boxed{\text{We want to approximate } \nabla \tfrac{1}{n} \textstyle\sum_{i=1}^{n} l_i(\boldsymbol{\theta})}$$

$$\mathbb{E}_{i \sim \mathrm{unif}\{1,\ldots,\mathrm{n}\}}[\nabla l_i(\boldsymbol{\theta})] = \frac{1}{n} \sum_{i} \nabla l_i(\boldsymbol{\theta}) = \nabla \mathcal{L}(\boldsymbol{\theta})$$

Problem: Variance !

SVRG

$$g = \nabla l_i(\boldsymbol{\theta}) - \nabla l_i(\hat{\theta}) + \frac{1}{n}\sum_{j=1}^{n} \nabla l_j(\hat{\theta})$$

- easy to show that this gradient estimate is unbiased
- Variance is controlled by how far $\boldsymbol{\theta}$, $\hat{\theta}$ are.

SVRG

$$
\begin{aligned}
\mathrm{var}[g] &= \frac{1}{n}\sum_{i=1}^{n} \|\nabla l_i(\theta) - \nabla l_i(\hat{\theta}) + \nabla \mathcal{L}(\hat{\theta}) - \nabla \mathcal{L}(\theta)\|^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} \|\nabla l_i(\theta) - \nabla l_i(\hat{\theta})\|^2 - \|\nabla \mathcal{L}(\hat{\theta}) - \nabla \mathcal{L}(\theta)\|^2 \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \|\nabla l_i(\theta) - \nabla l_i(\hat{\theta})\|^2 \\
&\leq \gamma^2 \|\theta - \hat{\theta}\|^2
\end{aligned}
$$

SVRG

SVRG

Set $\hat{\theta}_0$. For $t = 1, \ldots, T$,

- Set $\hat{\theta} = \hat{\theta}^{t-1}$. $\theta_0 = \hat{\theta}$.
- $\hat{g} = \frac{1}{n}\sum_{i=1}^{n} \nabla l_i(\hat{\theta})$: full gradient, at $\hat{\theta}$.
- For $k = 1, \ldots, m$
  - Sample $i \sim \{1, \ldots, n\}$
  - $g = \nabla l_i(\theta_{k-1}) - \nabla l_i(\hat{\theta}) + \hat{g}$: variance reduction
  - $\theta_k = \text{prox}(\theta_{k-1} - \eta g \,|\, \eta\psi)$
- $\hat{\theta}^t = \frac{1}{m}\sum_{k=1}^{m} \theta_k$

SAGA

$$\boxed{\hat{\theta} \text{ depends on the data index.}}$$

$$\text{(SVRG)} \qquad g = \nabla l_i(\boldsymbol{\theta^{t-1}}) - \nabla l_i(\hat{\theta}) + \tfrac{1}{n}\textstyle\sum_{j=1}^{n} \nabla l_j(\hat{\theta})$$

$$\text{(SAGA)} \qquad g = \nabla l_i(\boldsymbol{\theta^{t-1}}) - \nabla l_i(\hat{\theta}_{\boldsymbol{i}}) + \tfrac{1}{n}\textstyle\sum_{j=1}^{n} \nabla l_j(\hat{\theta}_{\boldsymbol{j}})$$

$\hat{\theta}_i$ is updated at every iteration.

$$\begin{cases} \hat{\theta}_i = \theta^{t-1} & i \text{ chosen} \\ \hat{\theta}_i \text{ unchanged} & \text{otherwise.} \end{cases}$$

Consequence: larger storage is necessary, but no double loop

SAGA

SAGA

- Set $\hat{g}_i = \bar{g} = 0, i \in \{1, \ldots, n\}$, Set θ..
  - Pick $i \in \{1, \ldots, n\}$ randomly.
  - Update $g_i = \nabla l_i(\theta)$
  - Estimate gradient $\hat{g} = g_i - \hat{g}_i + \bar{g}$
  - Update average gradient $\bar{g} = \bar{g} + \frac{1}{n}(g_i - \hat{g}_i)$.
  - Update stored gradients $\hat{g}_i = g_i$.
  - Update $\theta \leftarrow \mathrm{prox}(\theta - \eta\hat{g}|\eta\psi)$

Step size: ~1/γ , convergence guaranteed.

In practice: important to use mini-batches.

Recent Extensions: Point SAGA

Algorithm 1

Pick some starting point x^0 and step size γ. Initialize each $g_i^0 = f_i'(x^0)$, where $f_i'(x^0)$ is any gradient/subgradient at x^0.
Then at step $k+1$:

1. Pick index j from 1 to n uniformly at random.
2. Update x:

$$z_j^k = x^k + \gamma \left[g_j^k - \frac{1}{n} \sum_{i=1}^{n} g_i^k \right],$$

$$x^{k+1} = \text{prox}_j^{\gamma} \left(z_j^k \right).$$

3. Update the gradient table: Set $g_j^{k+1} = \frac{1}{\gamma} \left(z_j^k - x^{k+1} \right)$, and leave the rest of the entries unchanged ($g_i^{k+1} = g_i^k$ for $i \neq j$).

No regularizer required, proximal operator of each function.

Dual Methods

- Set $\mathbf{x}^0 = (x_1^0, \ldots, x_n^0)$,
- For $k = 1, \ldots, K$
  - $x_i^{k+1} = \arg\min_{y \in \mathbb{R}} f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, y, x_{i+1}^k, \ldots, x_n^k)$

source: wikipedia

Dual Methods

Reminders on Coordinate Descent

- Set $\mathbf{x}^0 = (x_1^0, \ldots, x_n^0)$,
- For $k = 1, \ldots, K$
  - $x_i^{k+1} = \arg\min_{y \in \mathbb{R}} f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, y, x_{i+1}^k, \ldots, x_n^k)$

source: wikipedia

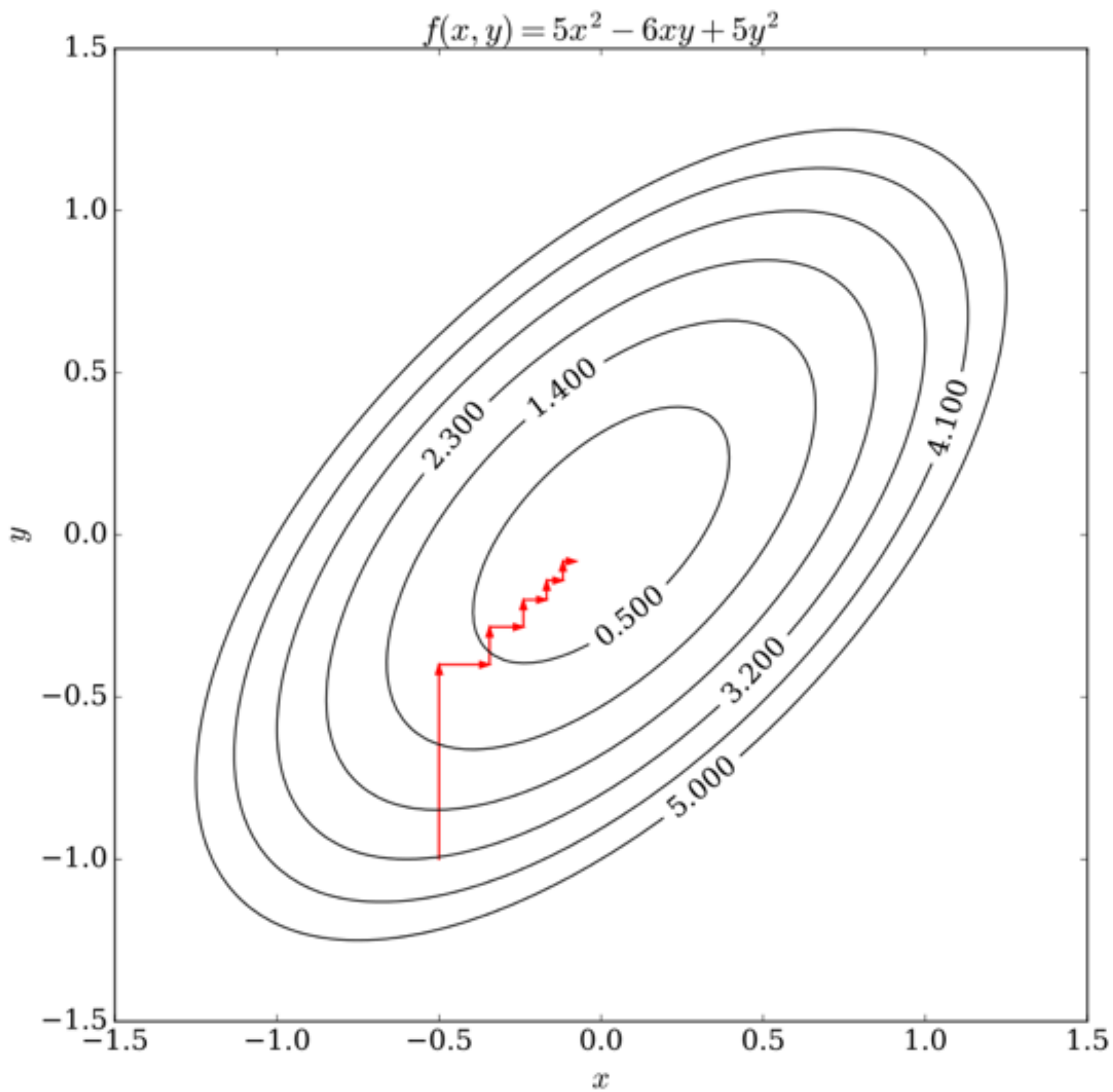
Dual Methods

Reminders on Coordinate Descent

source: wikipedia

Dual Methods

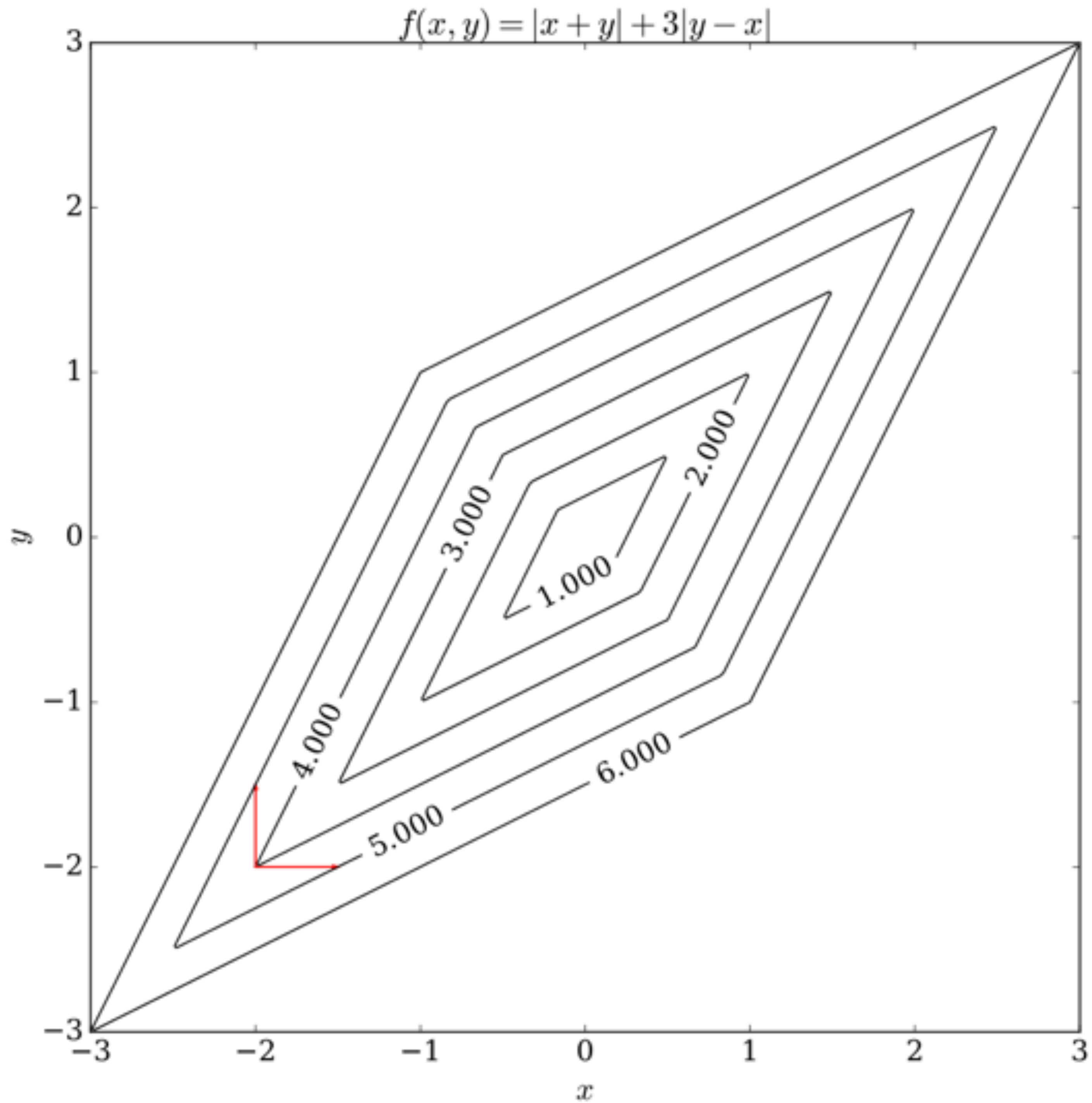
Reminders on Coordinate Descent



source: wikipedia

Dual Methods

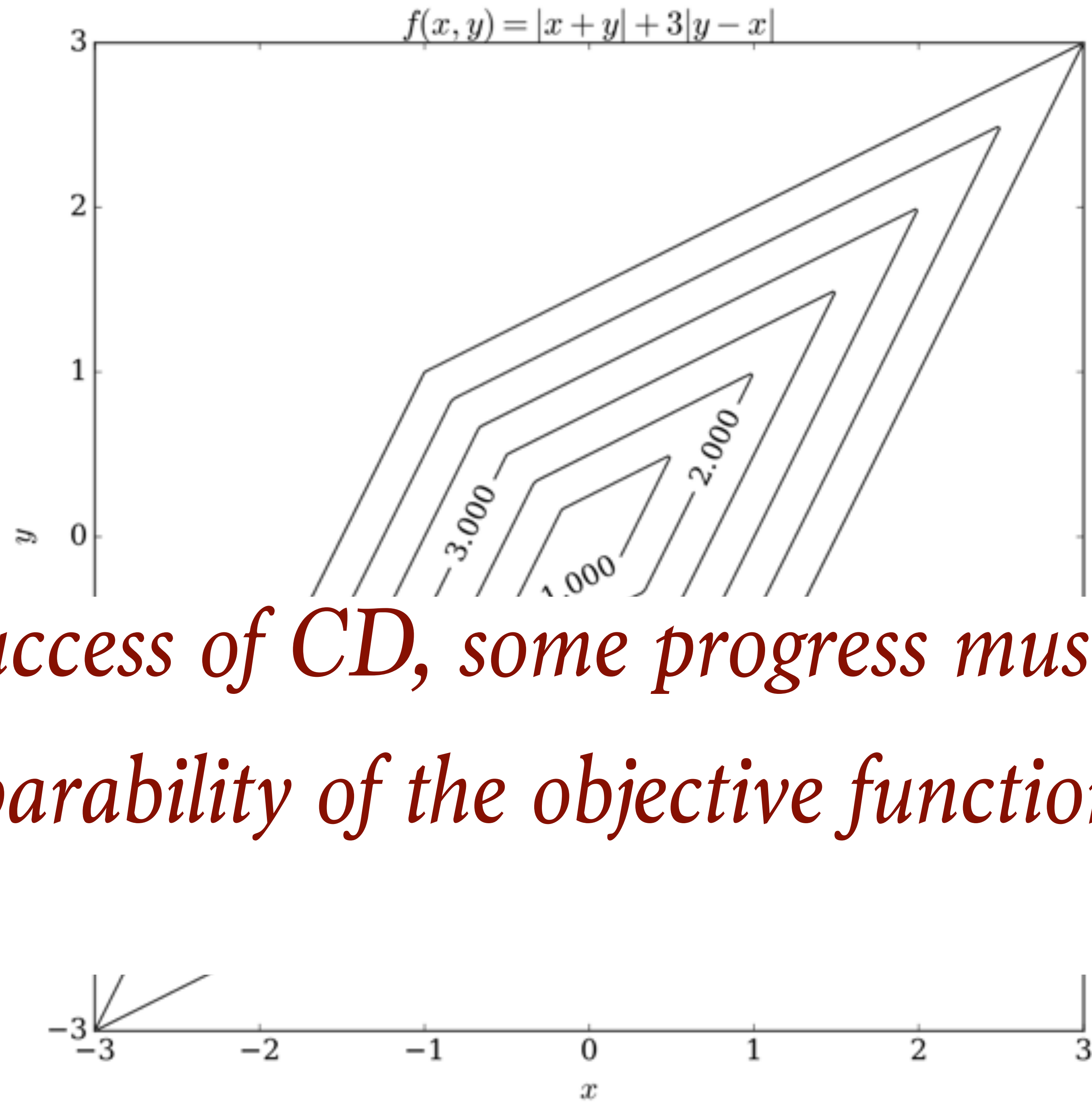
Reminders on Coordinate Descent



source: wikipedia

Dual Methods

Reminders on Coordinate Descent



To ensure success of CD, some progress must be guaranteed.

Separability of the objective function helps.

source: wikipedia

Dual Methods

- Set $\theta^0 = (\theta_1^0, \ldots, \theta_p^0)$,
- For $k = 1, \ldots, K$
  - Sample j.
  - Compute $g_j = \partial f(\theta) / \partial \theta_j$
  - $\theta_j \leftarrow \underset{y \in \mathbb{R}}{\arg\min}\; g_j y + \psi_j(y) + \frac{1}{2\eta_t} \|y - \theta_j\|^2$

source: wikipedia

Dual Methods

Coordinate Descent on Primal Problem

- Set $\theta^0 = (\theta_1^0, \ldots, \theta_p^0)$,
- For $k = 1, \ldots, K$
  - Sample j.
  - Compute $g_j = \partial f(\theta) / \partial \theta_j$
  - $\theta_j \leftarrow \arg\min_{y \in \mathbb{R}} g_j y + \psi_j(y) + \frac{1}{2\eta_t} \|y - \theta_j\|^2$

Regularizer must be separable.

source: wikipedia

Fenchel Duality Theorem

Theorem

Let $f : \mathbb{R}^p \to \bar{R}$ and $g : \mathbb{R}^q \to \bar{R}$ be closed convex, and $A \in \mathbb{R}^{q \times p}$ a linear map. Suppose that either condition (a) or (b) is satisfied. Then

$$\inf_{x \in \mathbb{R}^p} f(x) + g(Ax) = \sup_{y \in \mathbb{R}^q} -f^*(A^T y) - g^*(-y)$$

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

$$l_{\boldsymbol{\theta}}(z_i) = l(y_i, x_i^T \boldsymbol{\theta})$$

$$\boxed{\sup_{\boldsymbol{y} \in \mathbb{R}^{\boldsymbol{n}}} \frac{1}{n} \sum_i l_i^*(y_i) + \psi^*(-X^T y/n)}$$

$$\boldsymbol{\theta}^* = \nabla \psi^*(-X^T \boldsymbol{y}^*/n)$$