Bridging Duality, Variance Reduction, and Acceleration in Large-Scale Nonsmooth Optimization

Laurent Condat

Senior Research Scientist

King Abdullah Univ. of
Science and Technology
(KAUST)
Saudi Arabia

UESTC, June 2026

Optimization

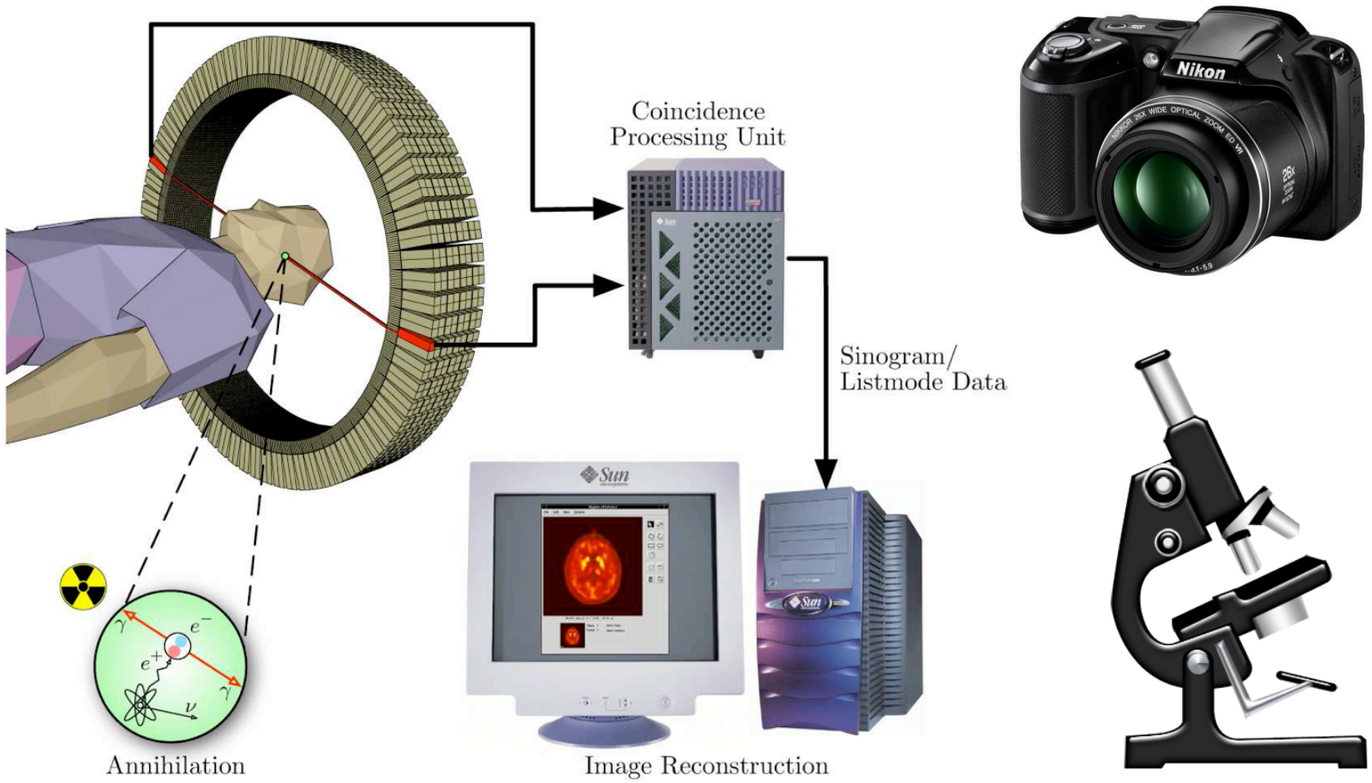
$$\text{Find } x^\star \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \ \Psi(x)$$

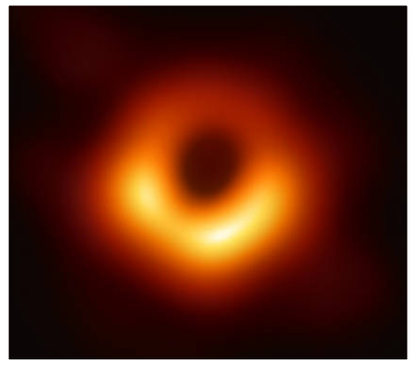
$\mathcal{X}$ is a real Hilbert space (e.g., $\mathbb{R}^d$)

Optimization

$$\text{Find } x^\star \in \underset{x \in \mathcal{X}}{\arg\min}\ \Psi(x)$$

$$\text{Example: } \Psi(x) = \|Ax - y\|^2 + R(x)$$





Optimization

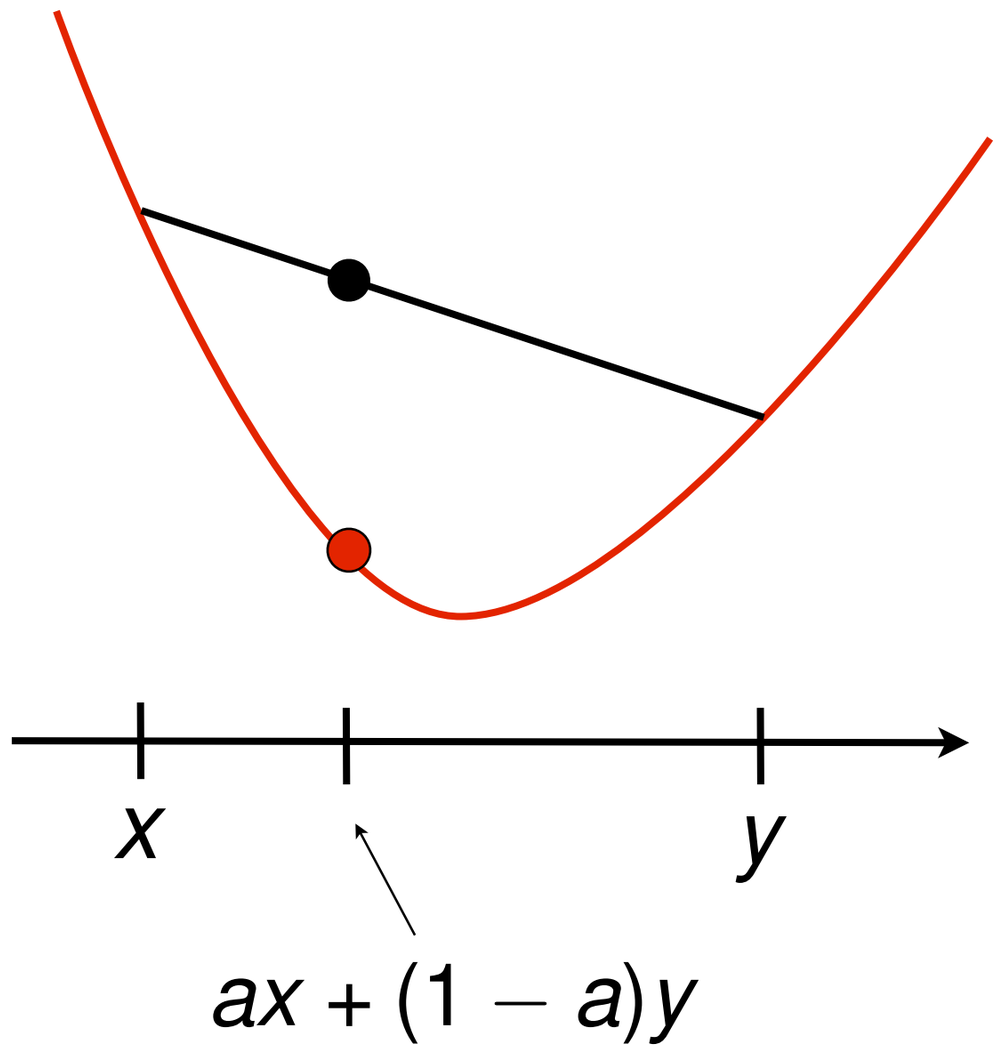
$$\text{Find } x^\star \in \arg\min_{x \in \mathcal{X}} \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

with

- linear operators $L_m : \mathcal{X} \to \mathcal{U}_m$
- real Hilbert spaces $\mathcal{X}, \mathcal{U}_m$
- **convex** functions $g_m : \mathcal{U}_m \to \mathbb{R} \cup \{+\infty\}$

Convex functions

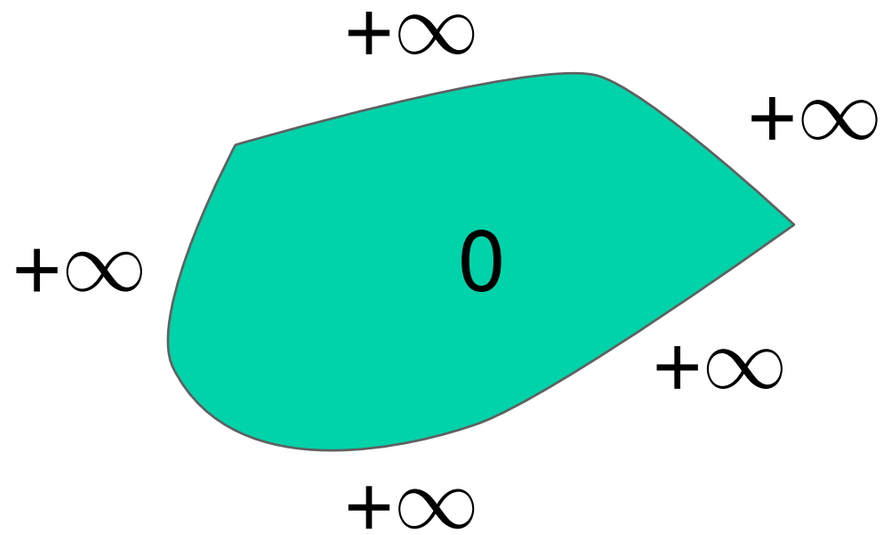
f is **convex** if $\forall x, y \in \mathcal{X}$ and $a \in [0, 1]$,
$f(ax + (1 - a)y) \leq af(x) + (1 - a)f(y)$



Convex functions

For a closed convex set $\Omega \subset \mathcal{X}$, its **indicator function** is

$$I_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega, \\ +\infty & \text{else.} \end{cases}$$

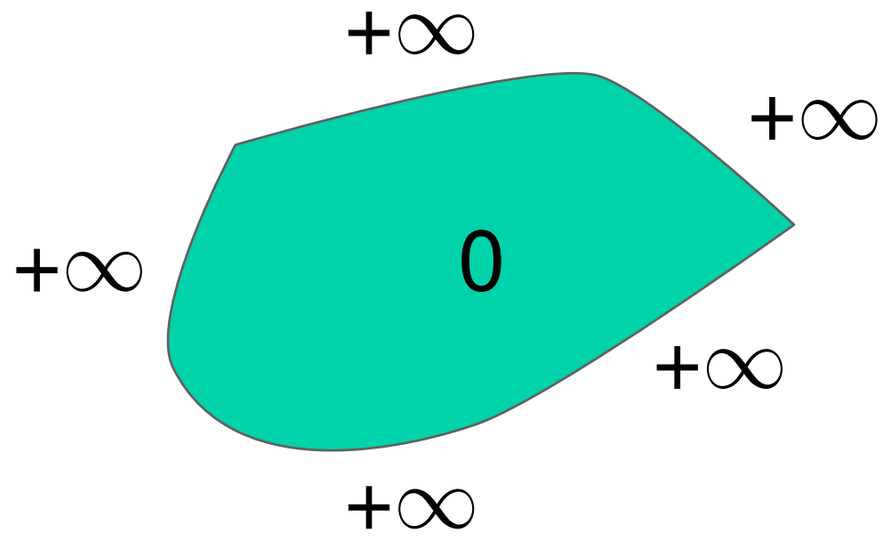


I_Ω is convex.

Convex functions

For a closed convex set $\Omega \subset \mathcal{X}$, its **indicator function** is

$$I_\Omega(x) = \begin{cases} 0 & \text{if } x \in \Omega, \\ +\infty & \text{else.} \end{cases}$$

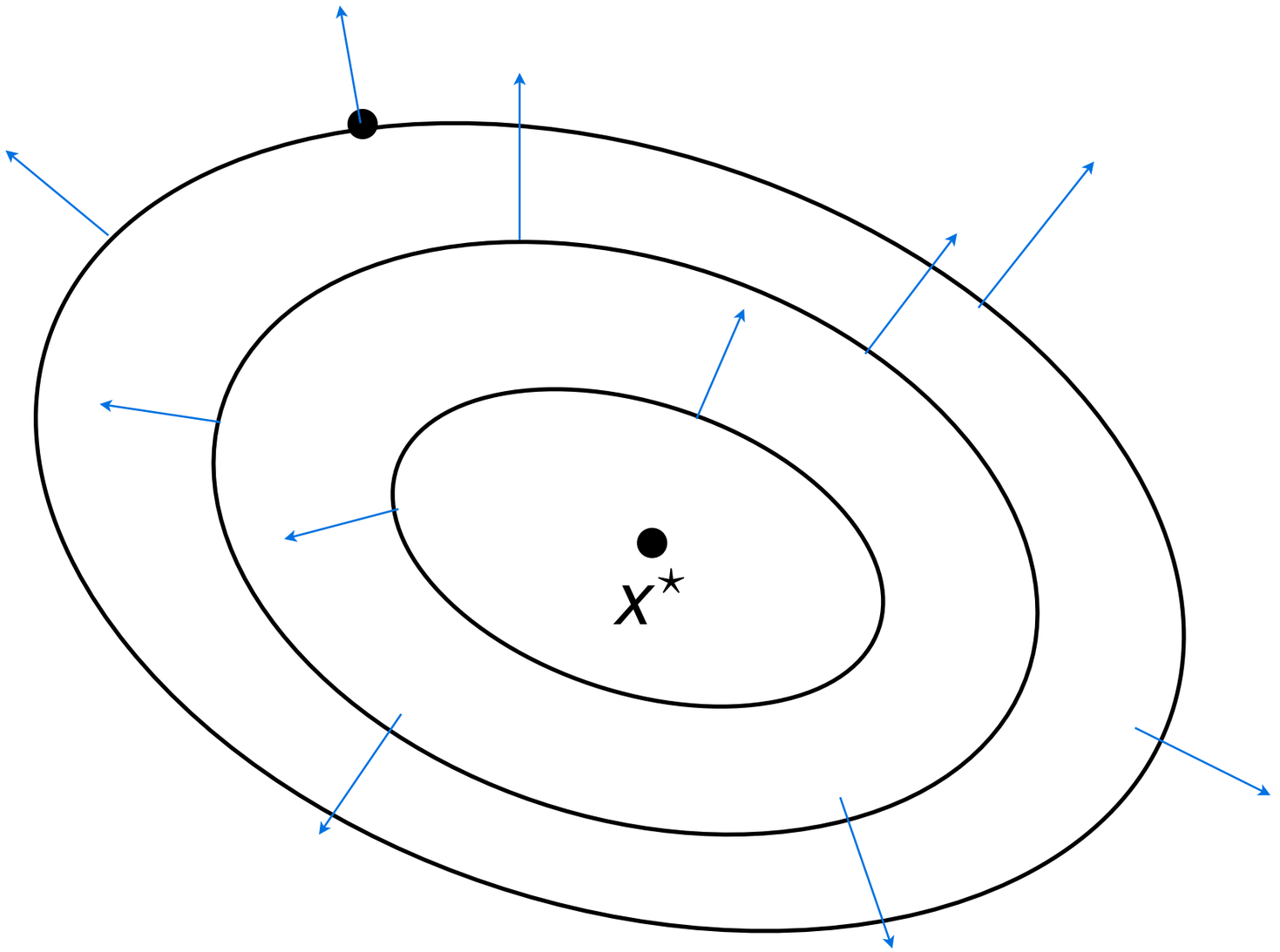


I_Ω is convex.

$$\underset{x \in \Omega}{\text{minimize}}\ f(x) \quad \equiv \quad \underset{x \in \mathcal{X}}{\text{minimize}}\ f(x) + I_\Omega(x)$$

Gradient descent

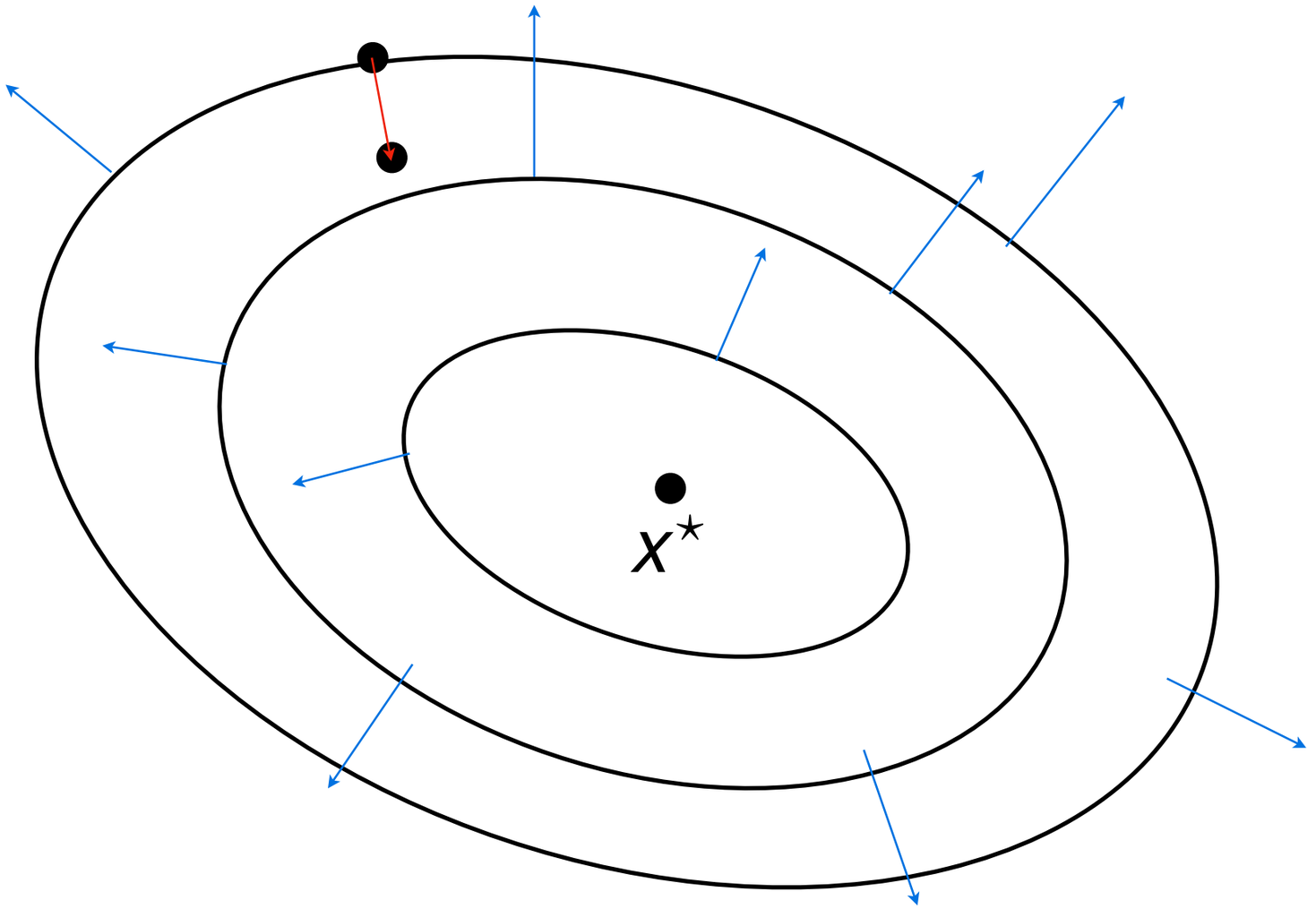
$$x^{t+1} = x^t - \gamma \nabla \Psi(x^t)$$



KAUST

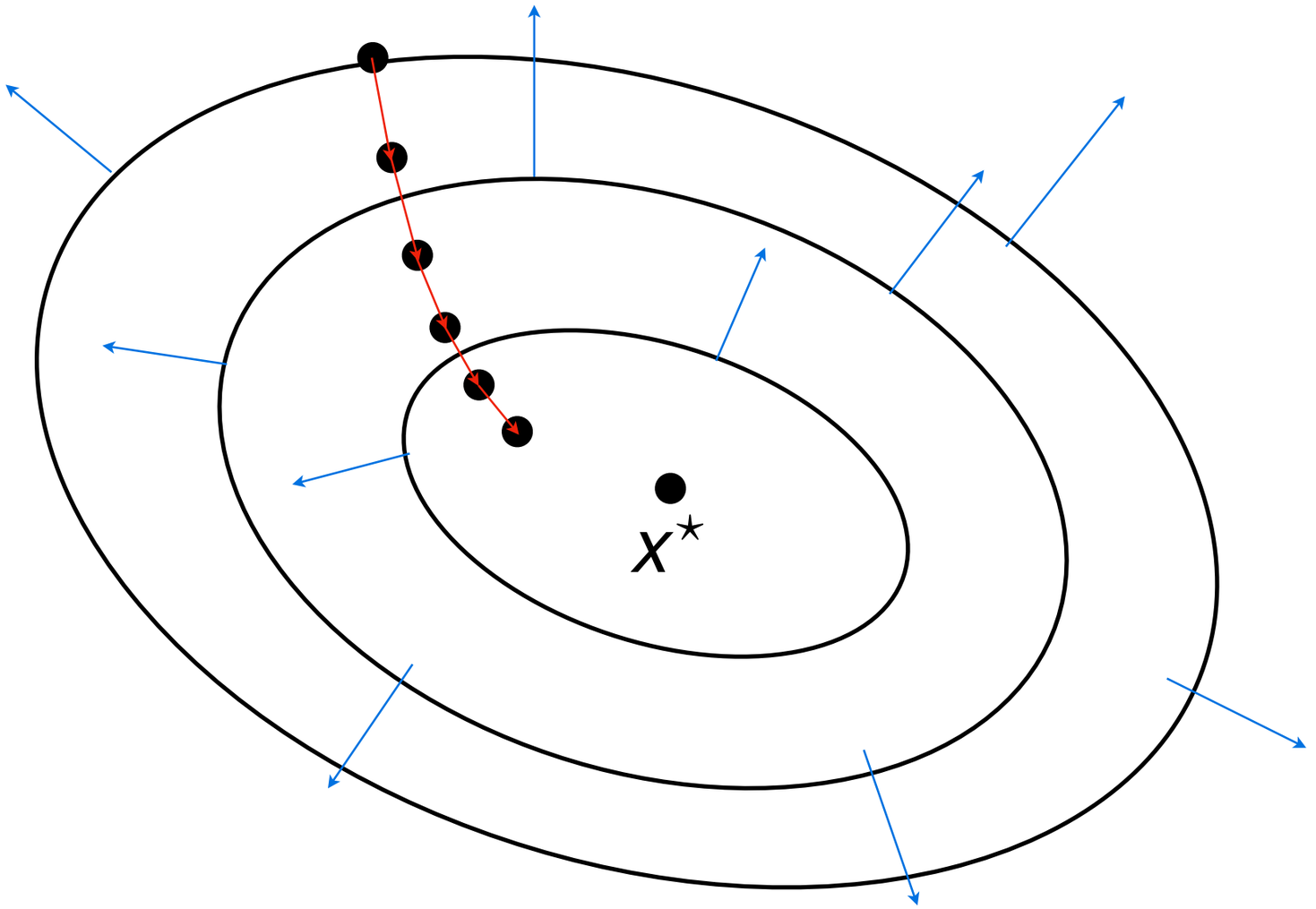
Gradient descent

$$x^{t+1} = x^t - \gamma \nabla \Psi(x^t)$$

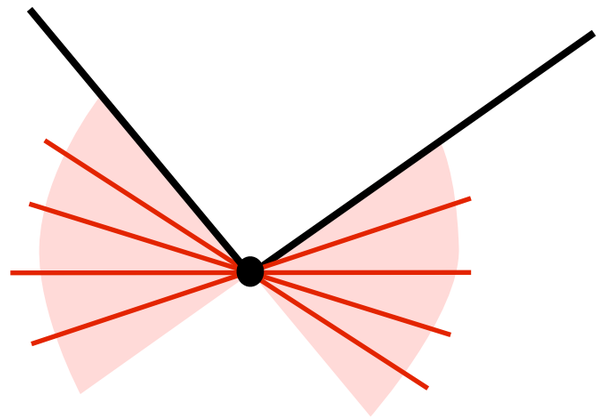


Gradient descent

$$x^{t+1} = x^t - \gamma \nabla \Psi(x^t)$$



The subdifferential



$$\partial f : \mathcal{X} \to 2^{\mathcal{X}}$$

$$x \mapsto \left\{ u \in \mathcal{X} \ : \ \forall y \in \mathcal{X},\ f(x) + \langle y - x, u \rangle \leq f(y) \right\}$$

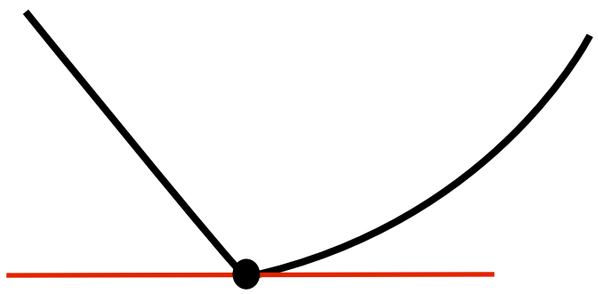
☞ $\partial f(x)$ is the set of gradients of the affine minorants of f at x

Fermat's rule

$$x^\star \in \arg\min f$$

$$\Leftrightarrow$$

$$0 \in \partial f(x^\star)$$



Pierre de Fermat, 1601-1665

First-order conditions

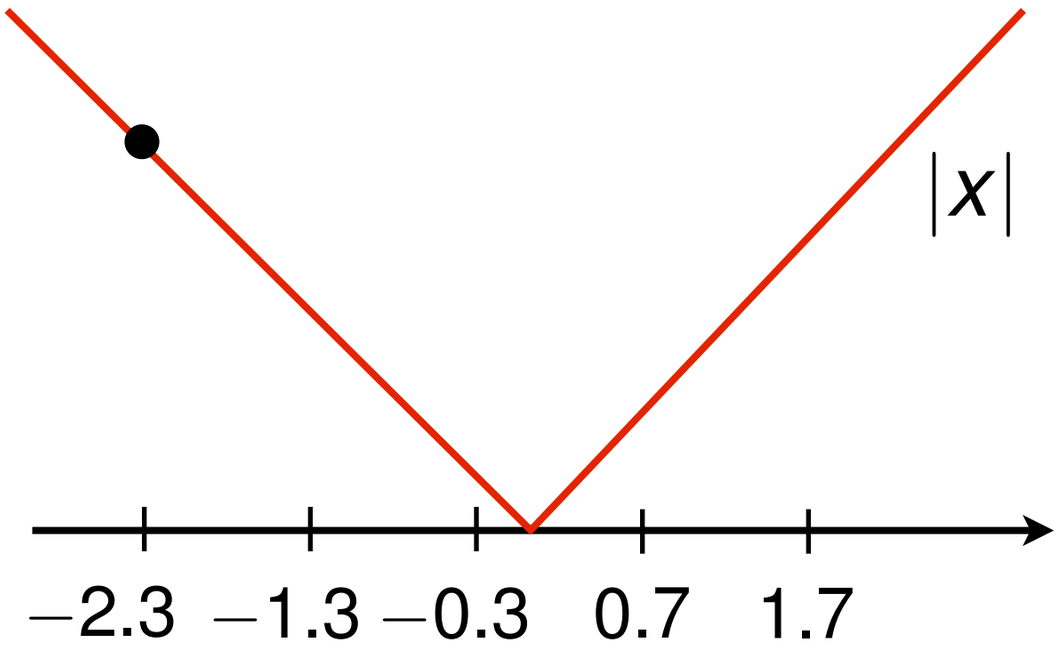
$$\text{Find } x^\star \in \arg\min_{x \in \mathcal{X}} \ \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

$$\Uparrow$$

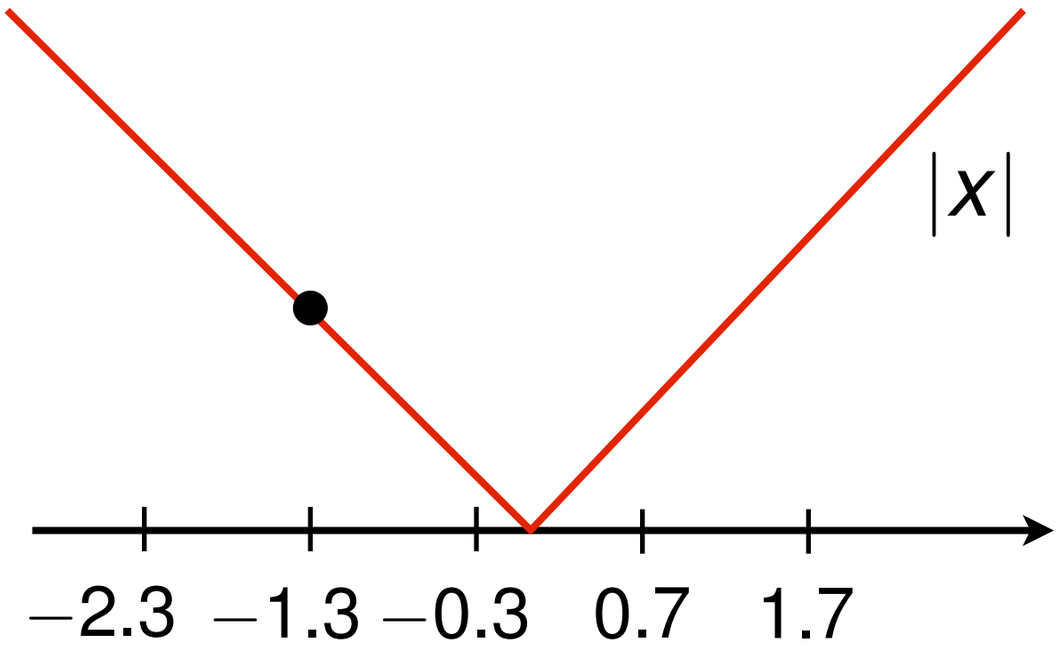
find $x^\star \in \mathcal{X}$ such that

$$0 \in \sum_{m=1}^{M} L_m^* \partial g_m(L_m x^\star)$$

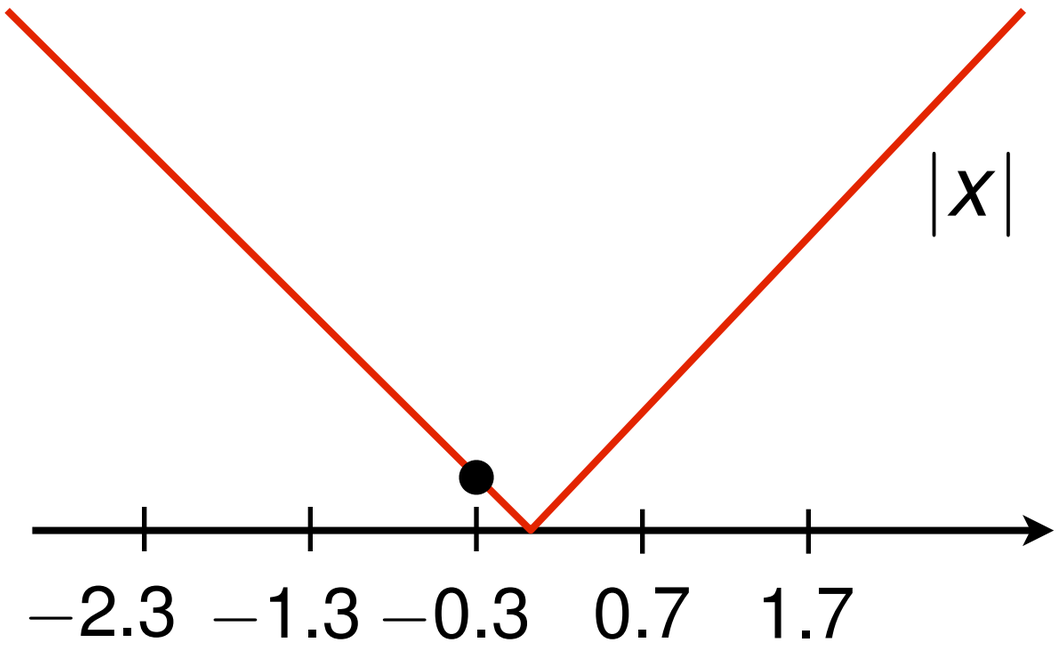
Subgradient descent



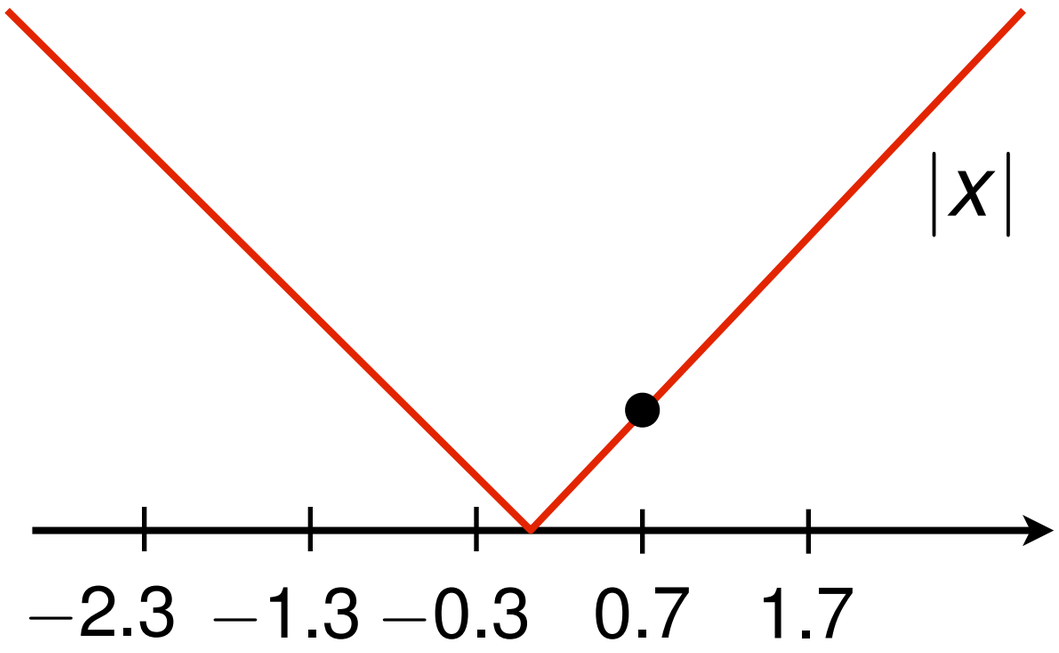
Subgradient descent



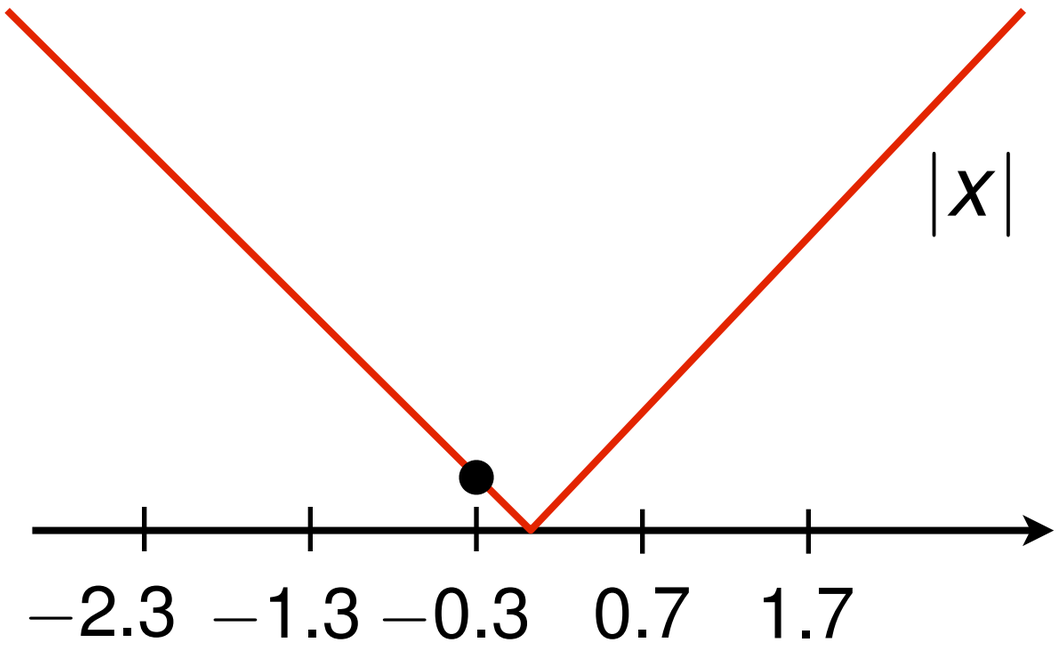
Subgradient descent



Subgradient descent

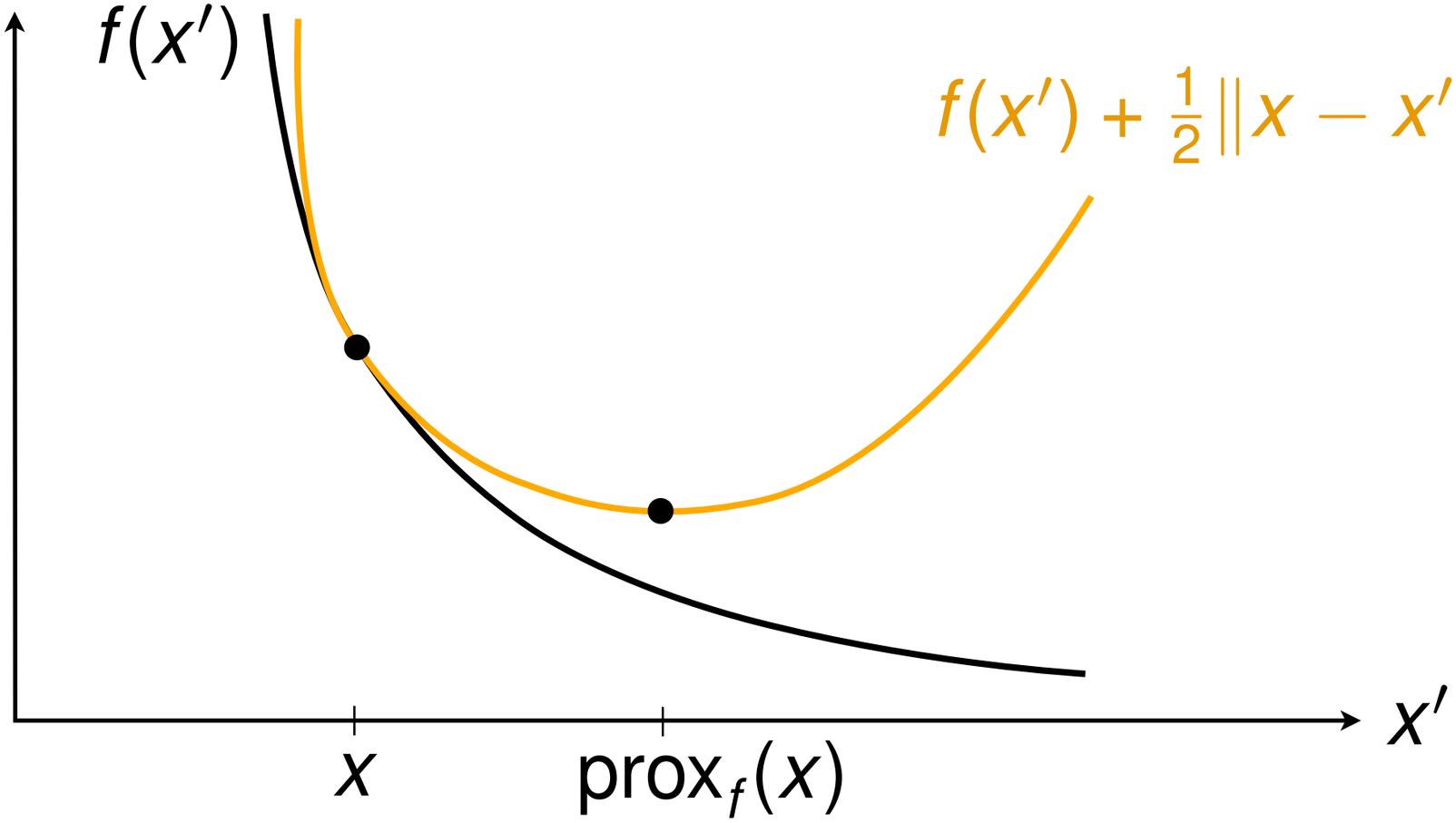


Subgradient descent



The proximity operator

$$\text{prox}_f : \mathcal{X} \to \mathcal{X} : x \mapsto \underset{x' \in \mathcal{X}}{\arg\min} \left(f(x') + \tfrac{1}{2} \|x - x'\|^2 \right)$$



The proximity operator

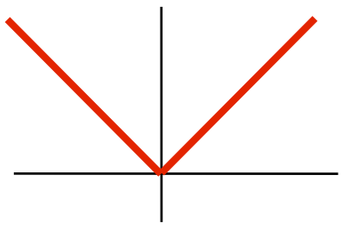
$$\text{prox}_f : \mathcal{X} \to \mathcal{X} : x \mapsto \underset{x' \in \mathcal{X}}{\arg\min} \left(f(x') + \tfrac{1}{2} \|x - x'\|^2 \right)$$

$$\text{prox}_f = (\partial f + \text{Id})^{-1}$$

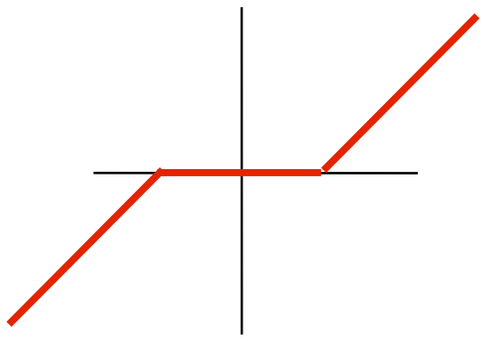
$$y = \text{prox}_{\gamma f}(x) \quad \equiv \quad y = x - \gamma \tilde{\nabla} f(y)$$

The proximity operator

$$\text{prox}_f : \mathcal{X} \to \mathcal{X} : x \mapsto \underset{x' \in \mathcal{X}}{\arg\min} \left(f(x') + \tfrac{1}{2} \|x - x'\|^2 \right)$$



$f(x) = |x|$



$$\text{prox}_f(x) = \text{sgn}(x) \max(|x| - 1, 0)$$

The proximity operator

Exact, finite time, algorithms are available to compute the proximity operator of:

- $\|X\|_*$ $\rightarrow$ SVD, $O(N^3)$
- 1-D TV $\rightarrow$ taut-string alg., $O(N)$
- 2-D anisotropic TV $\rightarrow$ graph cuts
- proj. onto the simplex $\rightarrow$ $O(N)$

...

LC, "A direct algorithm for 1-D total variation...", 2013

LC, "Fast projection onto the simplex...", 2016

Pustelnik, LC, "Proximity operator of a sum...", 2017

Proximal splitting algorithms

$$\text{Find } x^\star \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \sum_{m=1}^{M} g_m(L_m x)$$

No easy form of $\text{prox}_{g_1+g_2}$ or $\text{prox}_{g \circ L}$

Proximal splitting algorithms

$$\text{Find } x^\star \in \arg\min_{x \in \mathcal{X}} \sum_{m=1}^{M} g_m(L_m x)$$

We want full splitting, with individual activation of L_m, L_m^*, the gradient or proximity operator of g_m.

1) Splitting

Minimization of 3 functions

$$\text{Find } x^\star \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \left(f(x) + \sum_{m=1}^{M} g_m(L_m x) + h(x) \right)$$

$$\text{prox}_{\gamma f} \qquad \text{prox}_{\tau g_m} \qquad \nabla h \qquad L_m,\ L_m^*$$

Product space trick

$$\text{Find } x^\star \in \arg\min_{x \in \mathcal{X}} \Big(f(x) + g(Lx) + h(x) \Big)$$

$$\boldsymbol{g}(\boldsymbol{u}) = \sum_{m=1}^{M} g_m(u_m)$$

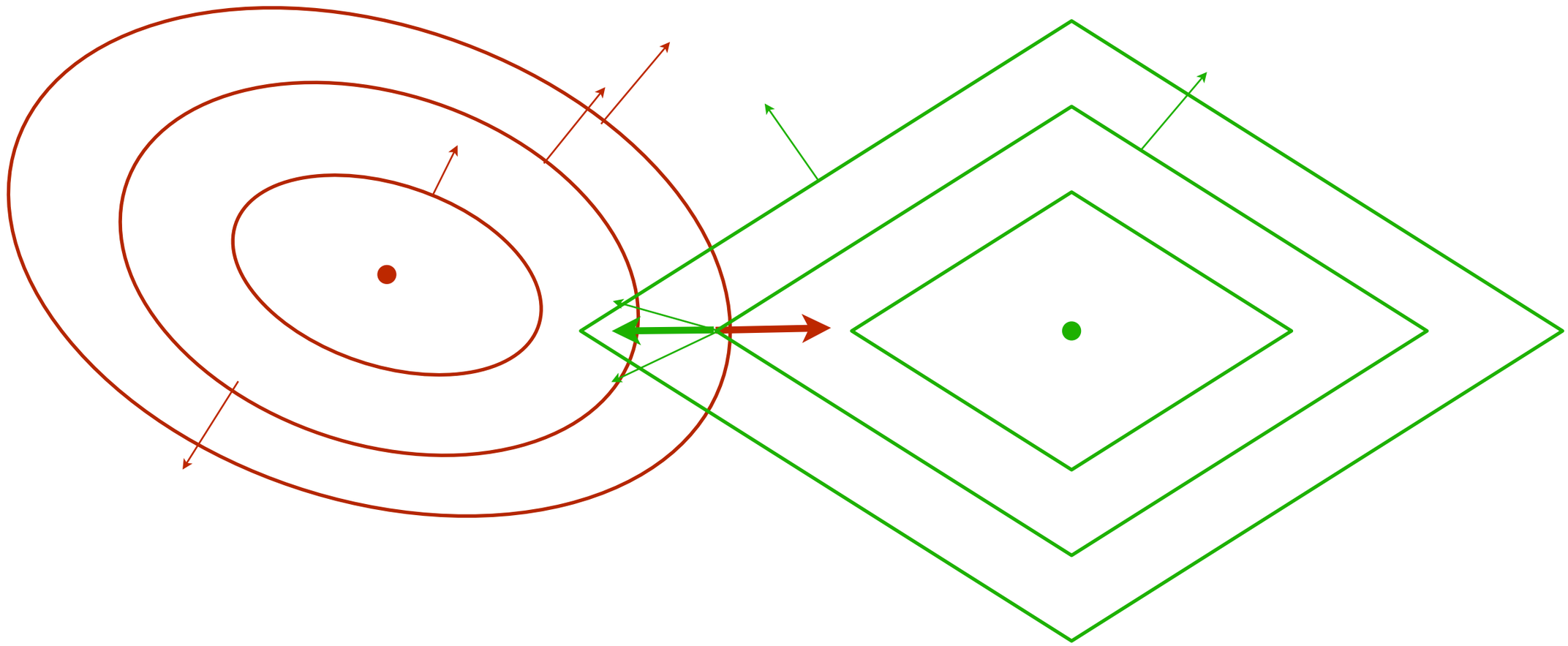
$$\boldsymbol{L}x = (L_1 x, \dots, L_m x)$$

☞

$$\boldsymbol{g}(\boldsymbol{L}x) = \sum_{m=1}^{M} g_m(L_m x)$$

Minimization of 2 functions

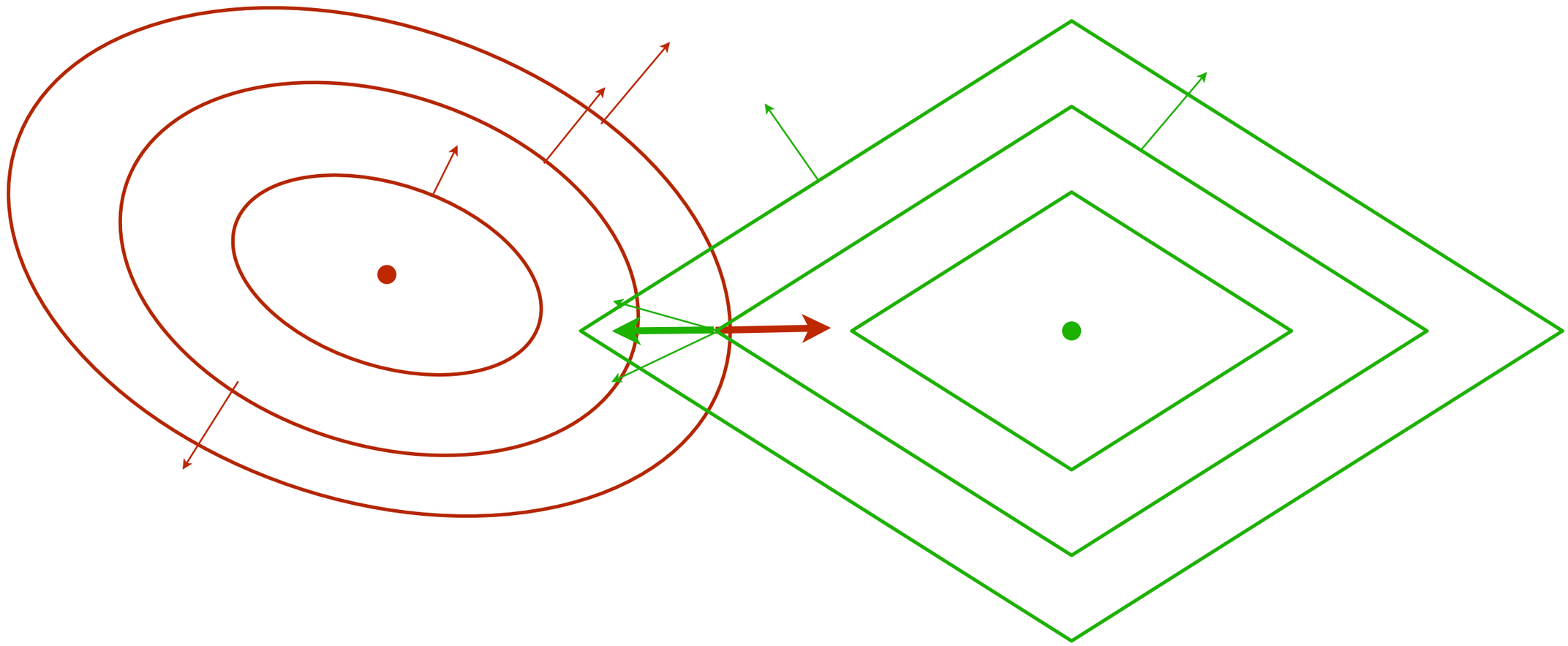
$$\text{Find } x^\star \in \arg\min_{x \in \mathcal{X}} \Big(f(x) + g(x) \Big)$$



$$\text{Find } x^\star \in \mathcal{X} \ : \ 0 \in \partial f(x^\star) + \partial g(x^\star)$$

Minimization of 2 functions

$$\text{Find } x^\star \in \underset{x \in \mathcal{X}}{\arg\min} \Big(f(x) + g(x) \Big)$$



$$\text{Find } (x^\star, u^\star) \in \mathcal{X}^2 \ : \ -u \in \partial f(x^\star), u \in \partial g(x^\star)$$

Minimization of 2 functions

$$\text{Find } x^\star \in \underset{x\in\mathcal{X}}{\arg\min} \Big(f(x) + g(x) \Big)$$

$$\sim\equiv$$

$$0 \in \partial f(x^\star) + \partial g(x^\star)$$

$$\equiv$$

$$\begin{cases} 0 \in \partial f(x^\star) + u^\star \\ u^\star \in \partial g(x^\star) \end{cases}$$

$$\equiv$$

$$\begin{cases} 0 \in \partial f(x^\star) + u^\star \\ 0 \in -x^\star + \partial g^*(u^\star) \end{cases} \qquad (\partial g^* = (\partial g)^{-1})$$

Minimization of 3 terms

$$\text{Find } x^\star \in \underset{x\in\mathcal{X}}{\arg\min} \Big(f(x) + g(Lx) + h(x) \Big)$$

$$\sim\equiv$$

$$0 \in \partial f(x^\star) + L^* \partial g(Lx^\star) + \nabla h(x^\star)$$

$$\equiv$$

$$\begin{cases} 0 \in \partial f(x^\star) + \nabla h(x^\star) + L^* u^\star \\ u^\star \in \partial g(Lx^\star) \end{cases}$$

$$\equiv$$

$$\begin{cases} 0 \in \partial f(x^\star) + \nabla h(x^\star) + L^* u^\star \\ 0 \in -Lx^\star + \partial g^*(u^\star) \end{cases}$$

Proximal splitting algorithms

minimize $f + g \circ L + h$

1979	$f + h$	☞	forward-backward
	$f + g$	☞	Douglas-Rachford / ADMM
2011	$f + g \circ L$	☞	Chambolle-Pock = PDHG
2011	$g \circ L + h$	☞	PAPC

Proximal splitting algorithms

minimize $f + g \circ L + h$

1979	$f + h$	☞	forward-backward
1979	$f + g$	☞	Douglas-Rachford / ADMM
2011	$f + g \circ L$	☞	Chambolle-Pock = PDHG
2011	$g \circ L + h$	☞	PAPC
2013	$f + g \circ L + h$	☞	Condat, Vu

LC, “A primal-dual splitting method for convex optimization…” 2013

Vu, “A splitting algorithm for dual monotone inclusions…” 2013

Proximal splitting algorithms

minimize $f + g \circ L + h$

1979	$f + h$	☞	forward-backward
	$f + g$	☞	Douglas-Rachford / ADMM
2011	$f + g \circ L$	☞	Chambolle-Pock = PDHG
2011	$g \circ L + h$	☞	PAPC
2013	$f + g \circ L + h$	☞	Condat, Vu
2017	$f + g + h$	☞	Davis-Yin
2018	$f + g \circ L + h$	☞	PD3O

Proximal splitting algorithms

minimize $f + g \circ L + h$

1979	$f + h$	☞	forward-backward
	$f + g$	☞	Douglas-Rachford / ADMM
2011	$f + g \circ L$	☞	Chambolle-Pock = PDHG
2011	$g \circ L + h$	☞	PAPC
2013	$f + g \circ L + h$	☞	Condat, Vu
2017	$f + g + h$	☞	Davis-Yin
2018	$f + g \circ L + h$	☞	PD3O
2022	$f + g \circ L + h$	☞	PDDY

Salim, LC et al., “Dualize, split, randomize: Toward fast nonsmooth optimization algorithms,” *JOTA*, 2022

4 primal-dual algorithms

minimize
$f + g \circ L + h$

Condat–Vu algorithm form I

$$\begin{cases} x^{t+1} = \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right) \\ u^{t+1} = \operatorname{prox}_{\tau g^*}\left(u^t + \tau L(2x^{t+1} - x^t)\right) \end{cases}$$

Condat–Vu algorithm form II

$$\begin{cases} u^{t+1} = \operatorname{prox}_{\tau g^*}\left(u^t + \tau L x^t\right) \\ x^{t+1} = \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^*(2u^{t+1} - u^t)\right) \end{cases}$$

PD3O algorithm

$$\begin{cases} x^{t+1} = \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right) \\ u^{t+1} = \operatorname{prox}_{\tau g^*}\left(u^t + \tau L(2x^{t+1} - x^t - \gamma \nabla h(x^{t+1}) + \gamma \nabla h(x^t))\right) \end{cases}$$

PDDY algorithm

$$\begin{cases} u^{t+1} = \operatorname{prox}_{\tau g^*}\left(u^t + \tau L x^t\right) \\ x^{t+1} = \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t - \gamma L^*(u^{t+1} - u^t)) - \gamma L^*(2u^{t+1} - u^t)\right) \end{cases}$$

Summary

Large-scale problems with nonsmooth functions and linear operators

proximal splitting algorithms designed using monotone and fixed-point operator theory

LC et al. “Proximal Splitting Algorithms for Convex Optimization: A Tour of Recent Advances, with New Twists,” *SIAM Review*, 2023

Convergence rates

Theorem – PD3O, with g continuous around $Lx^\star$:

$$\Psi(x^t) - \Psi(x^\star) = o(1/\sqrt{t})$$

Theorem – accelerated PD3O and PDDY when h or f strongly convex, with varying stepsizes:

$$\|x^t - x^\star\|^2 = O(1/t^2)$$

Theorem – linear convergence of PD3O and PDDY when h or f strongly convex and g smooth:

$$\|x^t - x^\star\|^2 \leq \rho^t \Psi^0$$

LC, Malinovsky, Richtárik, "Distributed Proximal Splitting Algorithms with Rates and Acceleration," 2022

The power of randomness

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n} f_i(x) \quad \text{using the } \nabla f_i$$

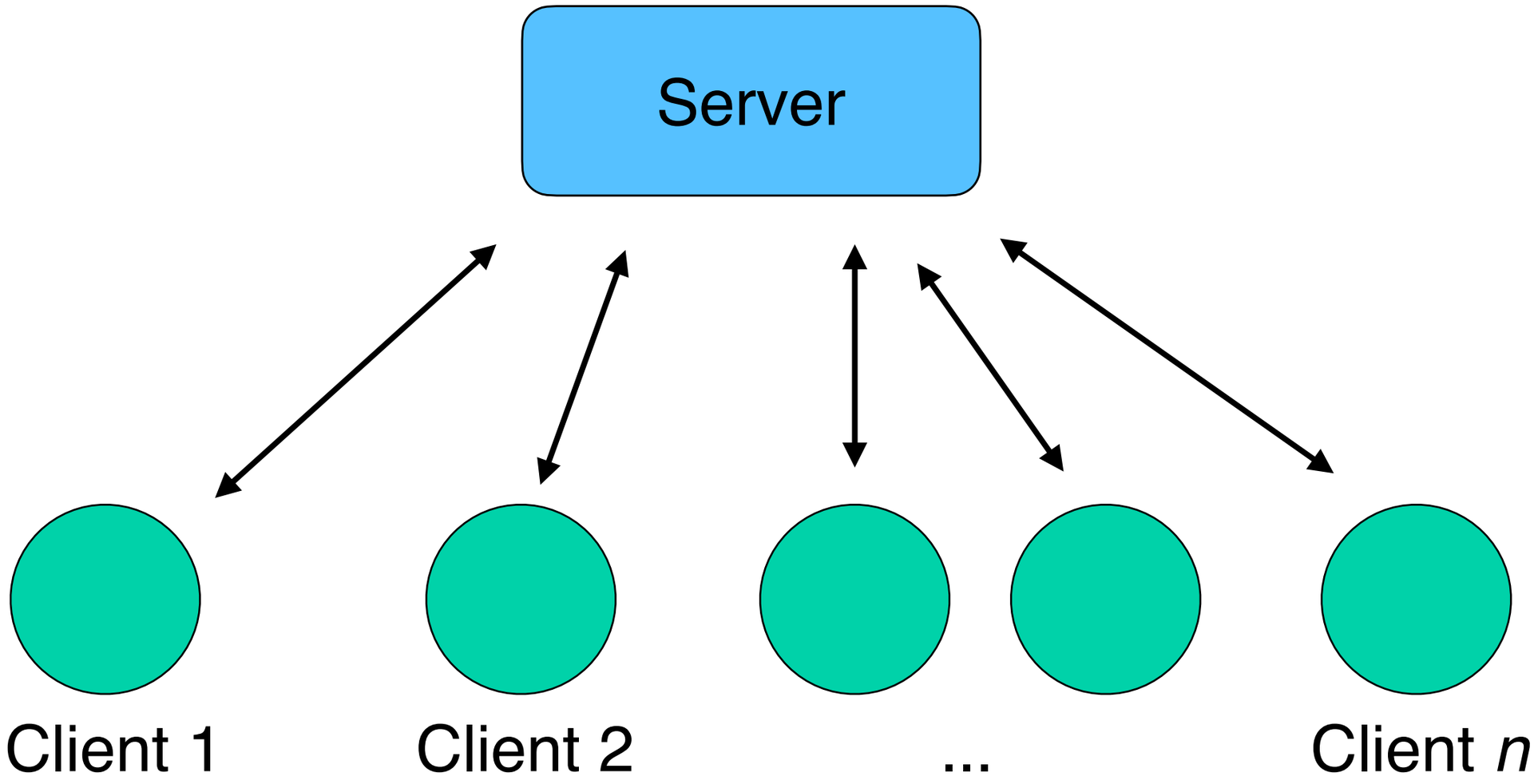
with every f_i β-smooth and μ-strongly convex

☞ lower bounds in Woodworth & Srebro [2016] on number of gradient calls:

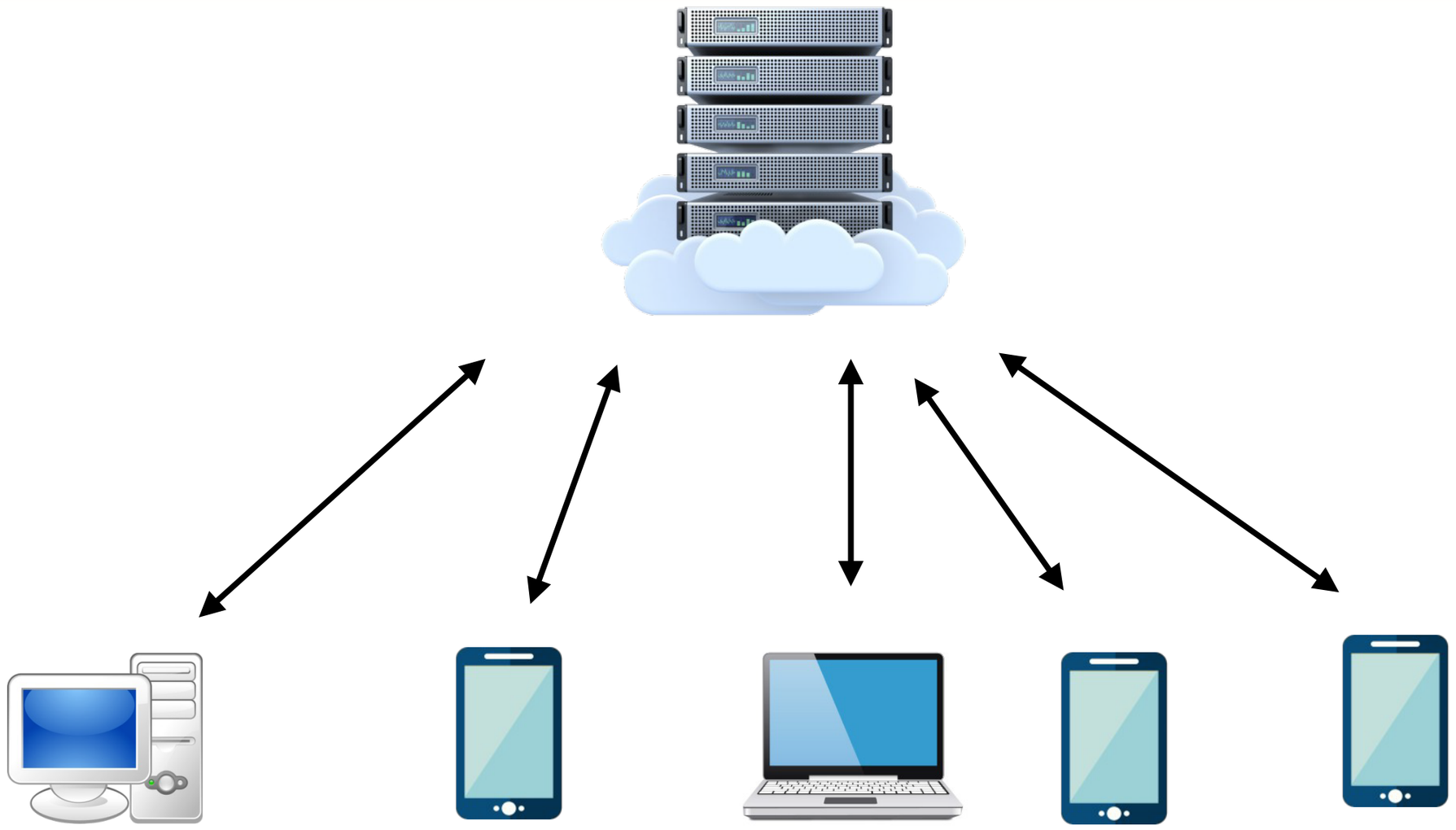
- deterministic algorithms: $\Omega\left(n\sqrt{\beta/\mu}\log\epsilon^{-1}\right)$
- randomized algorithms: $\Omega\left((n+\sqrt{n\beta/\mu})\log\epsilon^{-1}\right)$

2) Distributed optimization

Distributed optimization



Federated learning



Distributed optimization

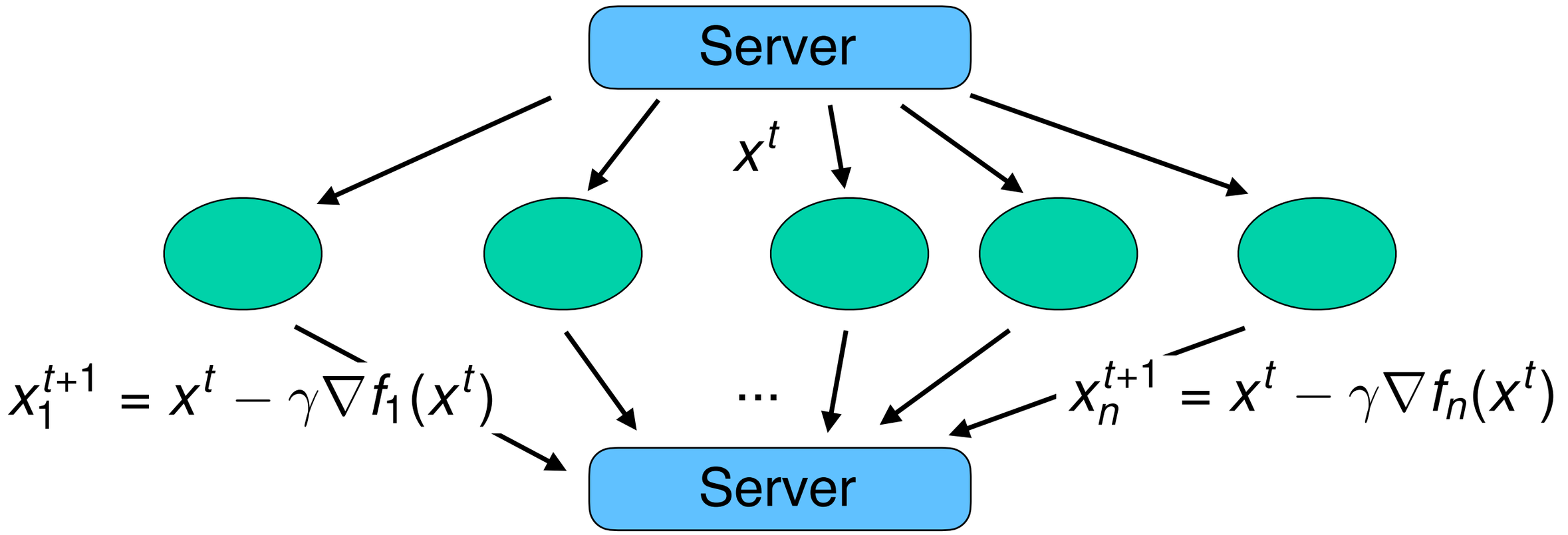
$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

Distributed optimization

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

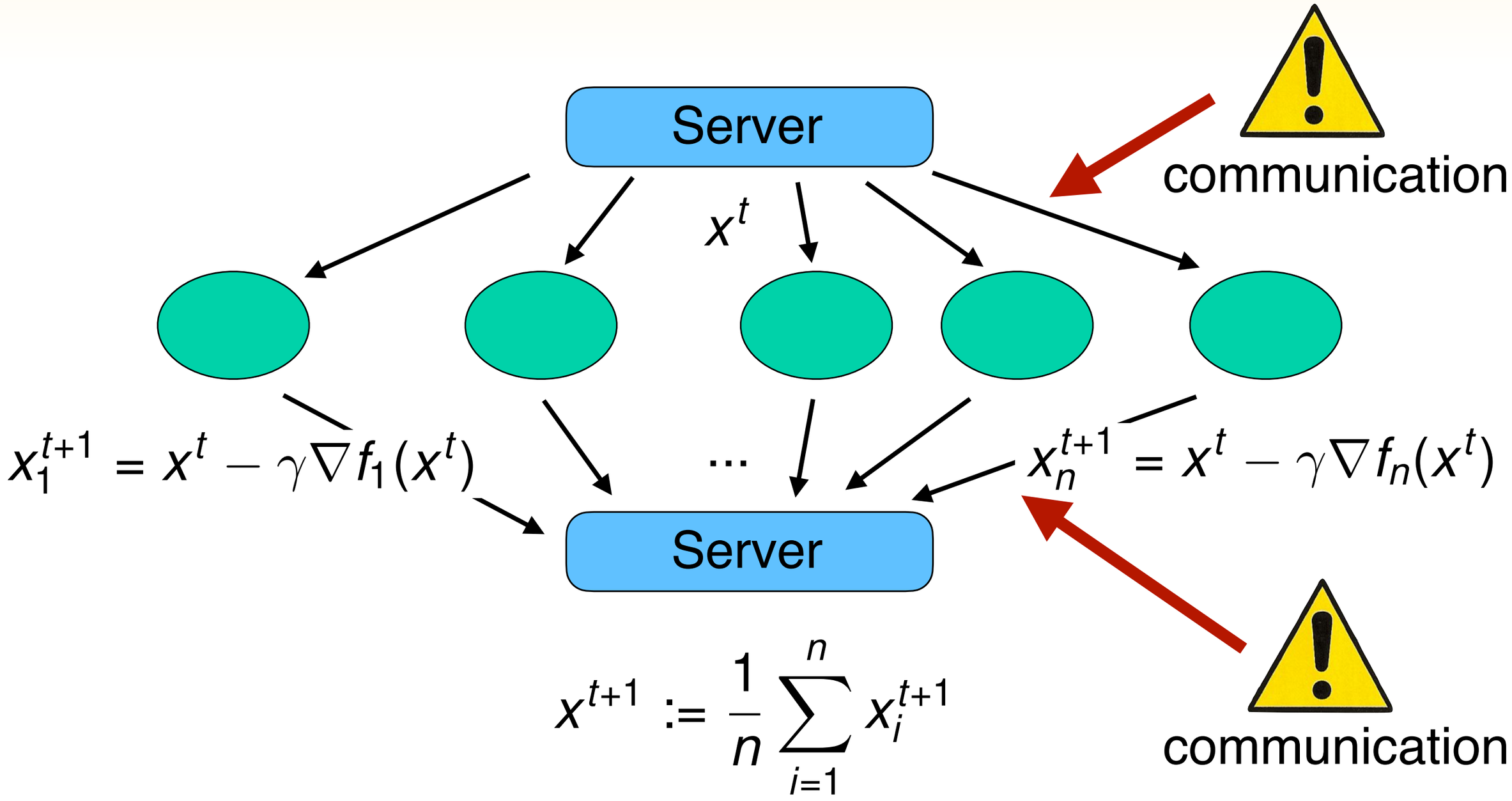
Heterogeneous setting: f_i arbitrarily different

Distributed GD



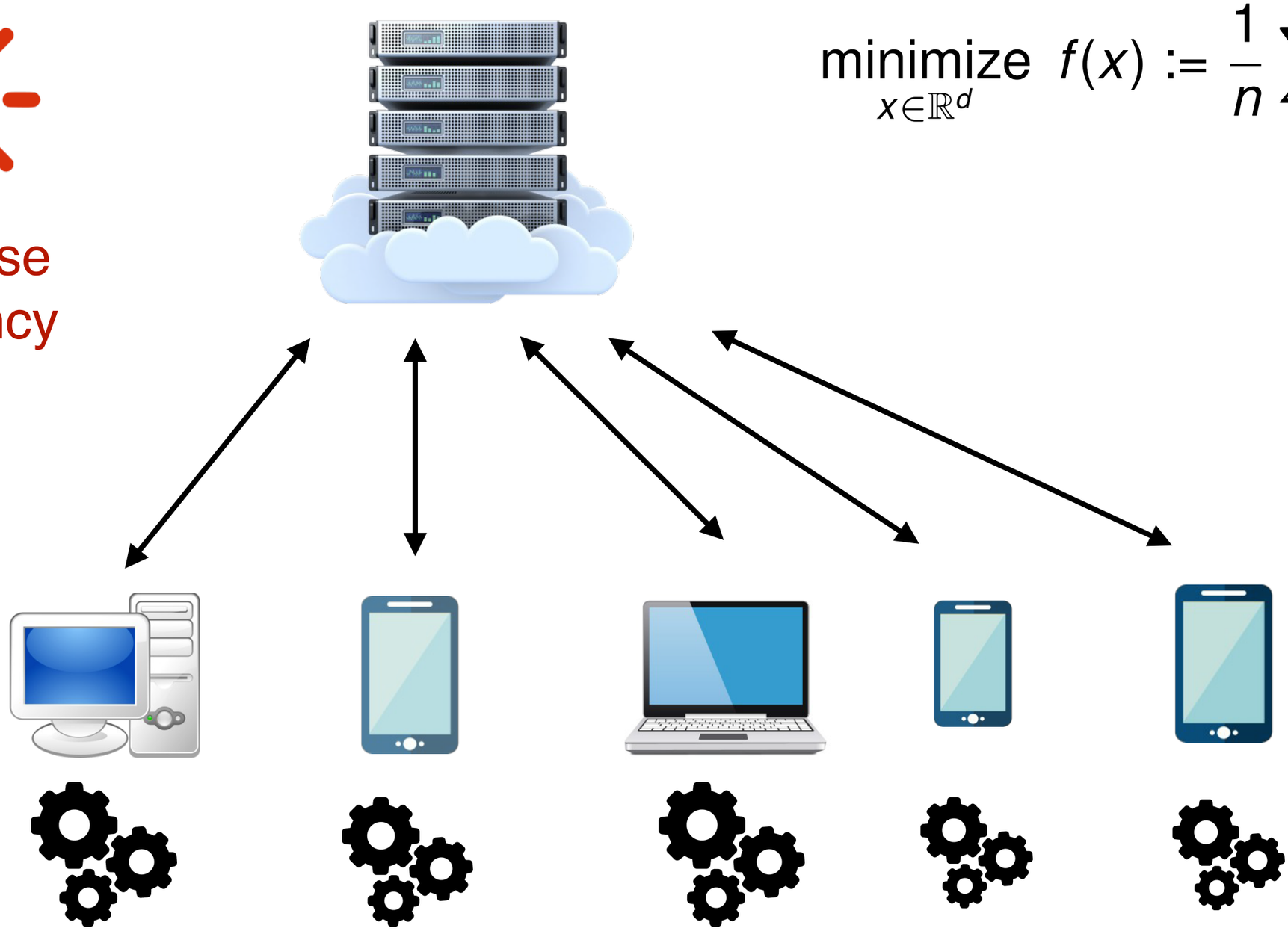
$$x^{t+1} := \frac{1}{n} \sum_{i=1}^{n} x_i^{t+1}$$

Distributed GD



Local training

decrease frequency

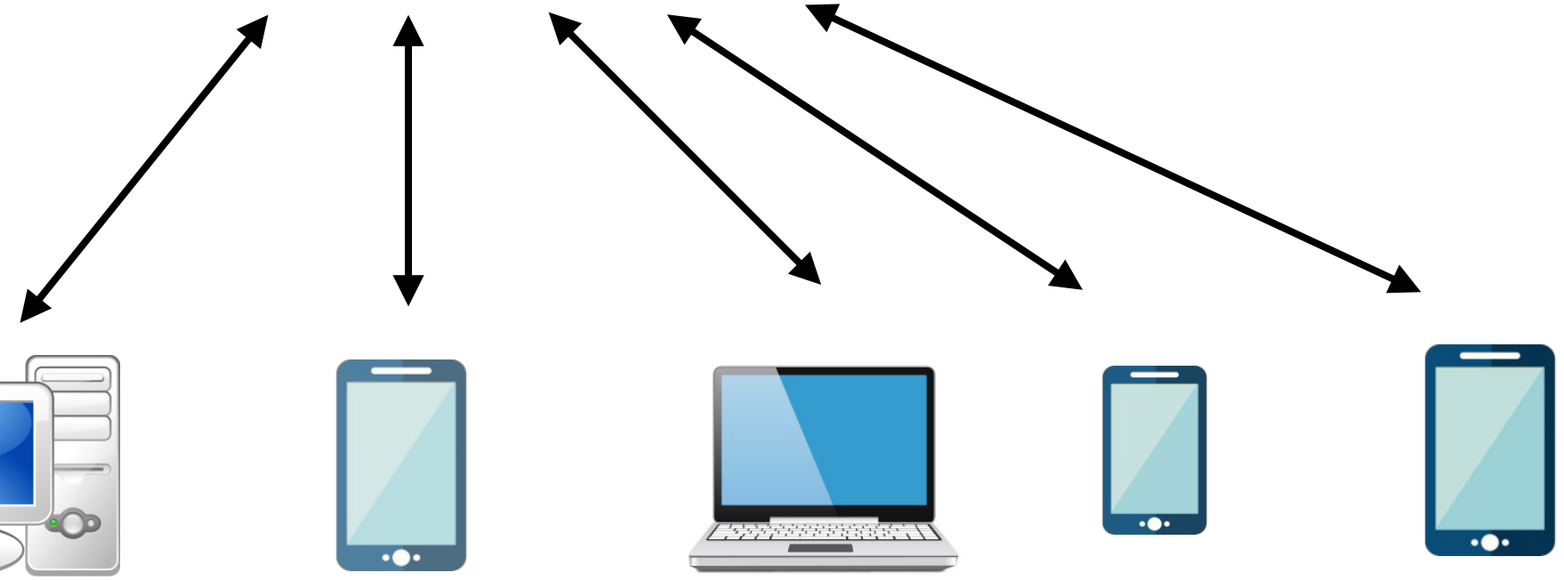


$$\underset{x\in\mathbb{R}^d}{\text{minimize}}\ f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

Compression

compress vectors

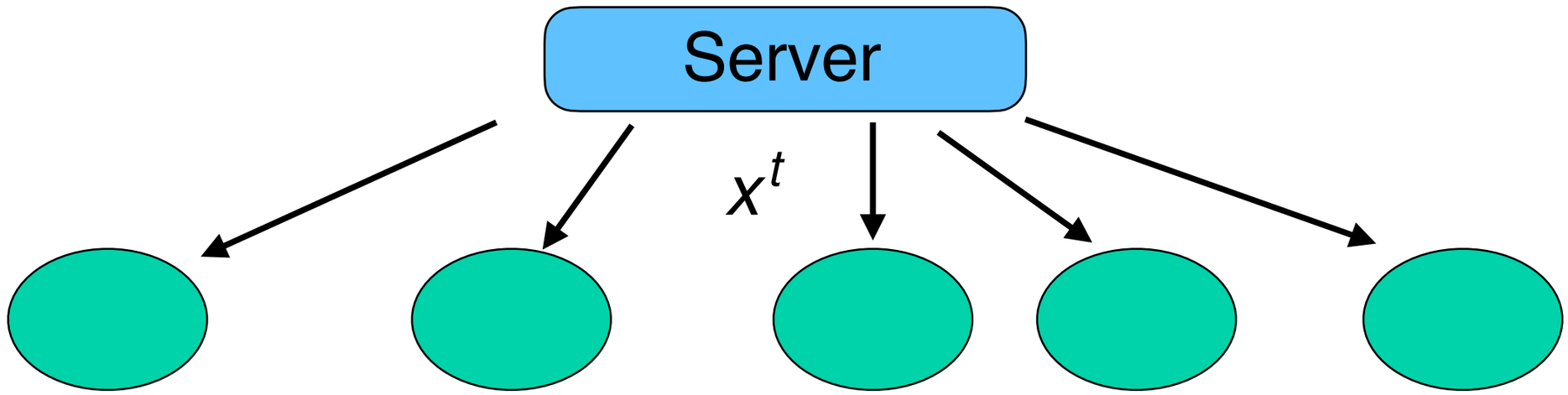
e.g.



$$\underset{x\in\mathbb{R}^d}{\text{minimize}}\ f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

2.1) Local training

Distributed **Local** GD = FedAvg

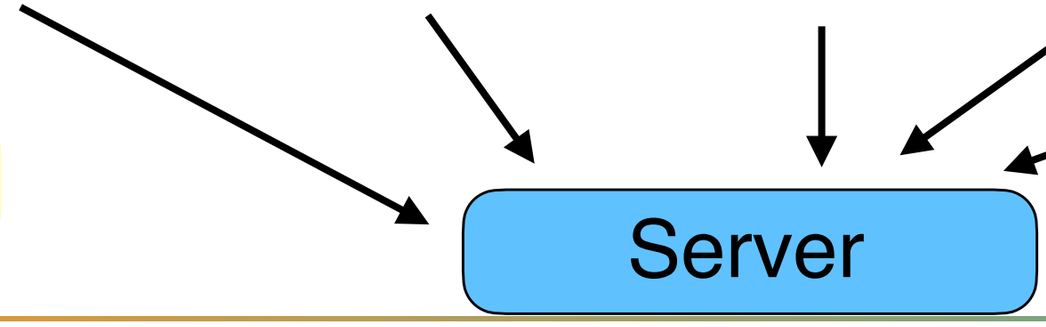


$$x_1^{t+1} = x^t - \gamma \nabla f_1(x^t) \qquad \cdots \qquad x_n^{t+1} = x^t - \gamma \nabla f_n(x^t)$$

$$x_1^{t+2} = x_1^{t+1} - \gamma \nabla f_1(x_1^{t+1}) \qquad x_n^{t+2} = x_n^{t+1} - \gamma \nabla f_n(x_n^{t+1})$$

$$\cdots \qquad\qquad\qquad\qquad \cdots$$

$$x_1^{t+H} = x_1^{t+H-1} - \gamma \nabla f_1(x_1^{t+H-1}) \qquad x_n^{t+H} = x_n^{t+H-1} - \gamma \nabla f_n(x_n^{t+H-1})$$

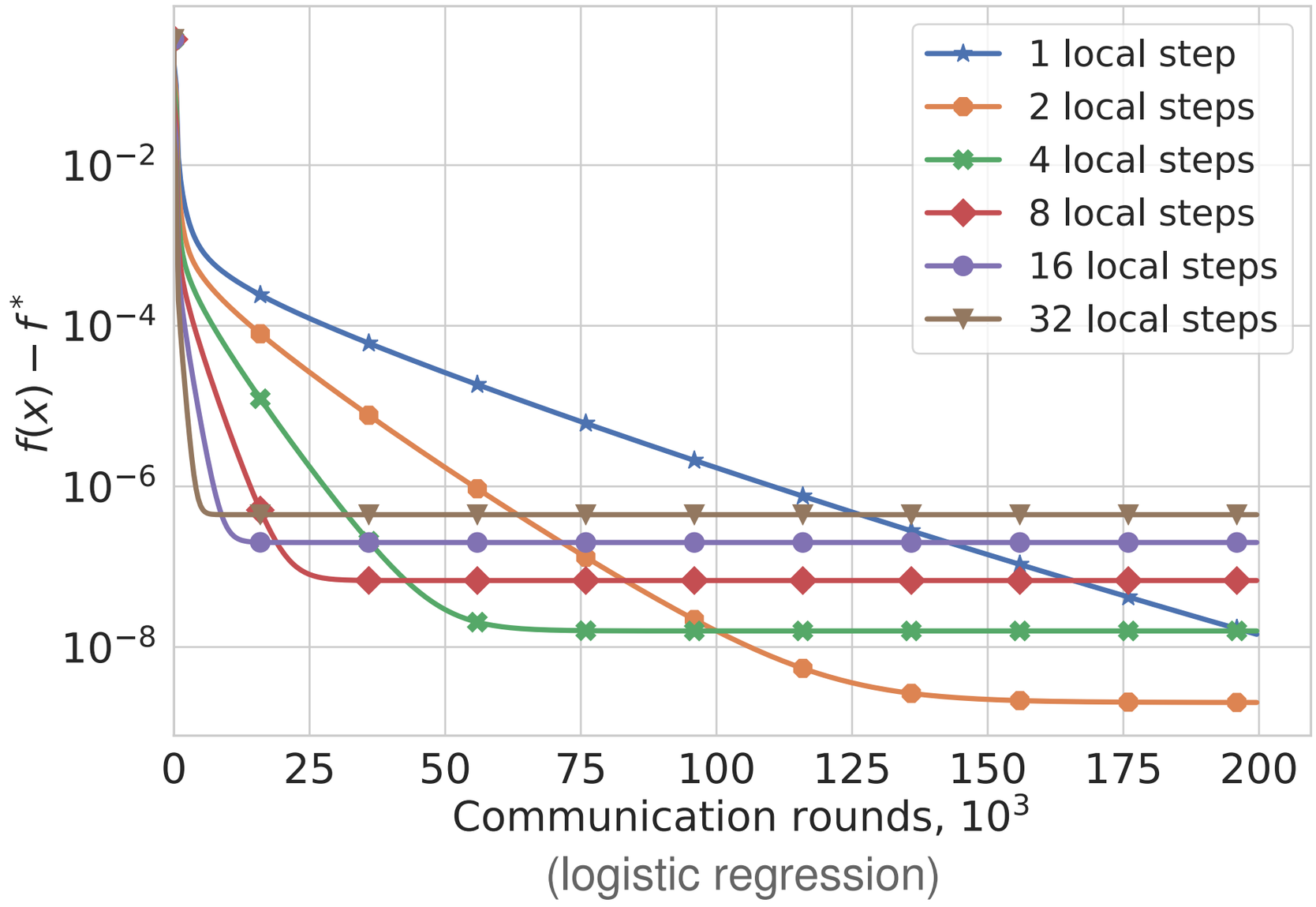


$H \geq 1$

$$x^{t+H} := \frac{1}{n} \sum_{i=1}^{n} x_i^{t+H}$$

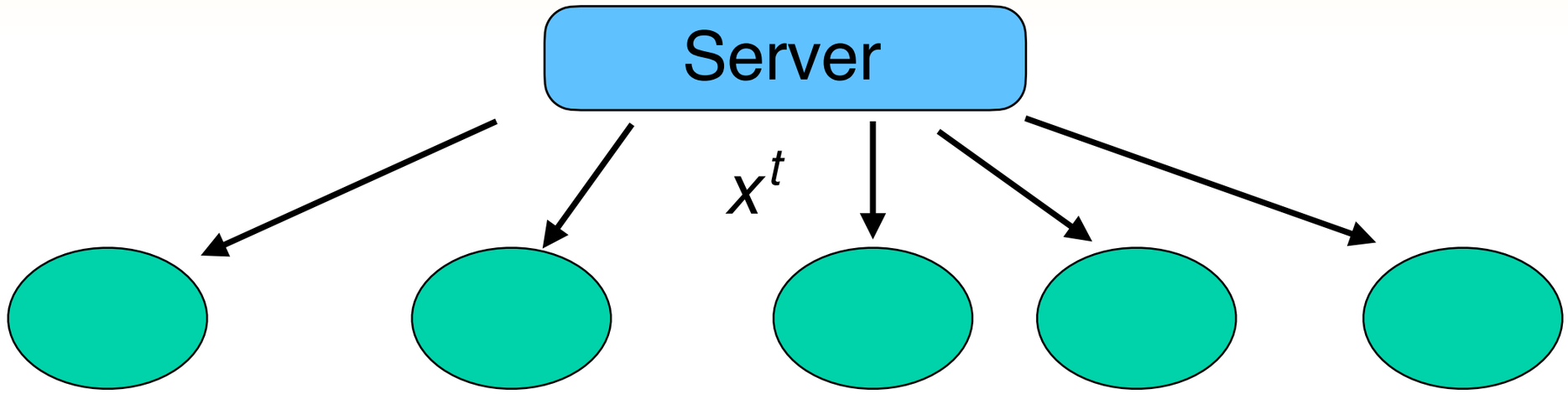
KAUST

Local GD: analysis



Malinovsky, Kovalev, Gasanov, LC, Richtárik, “From local SGD to local fixed point methods for federated learning,” ICML 2020

Scaffnew: **Variance-reduced** local GD



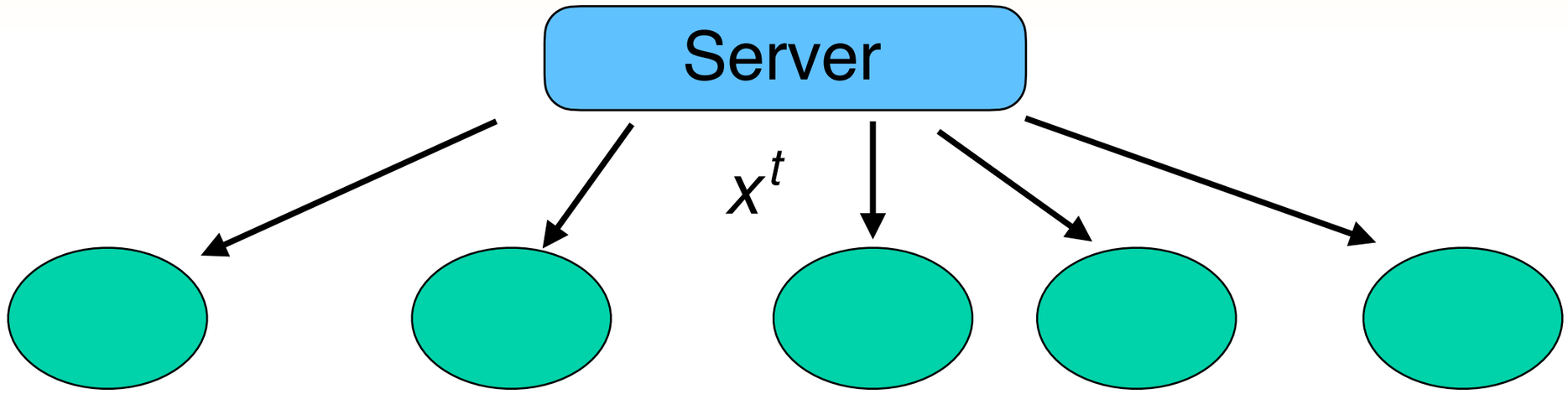
$$\hat{x}_1^t = x_1^t - \gamma \nabla f_1(x_1^t) + \gamma u_1^t \qquad \cdots \qquad \hat{x}_n^t = x_n^t - \gamma \nabla f_n(x_n^t) + \gamma u_n^t$$

Probabilistic Com: $x_i^{t+1} := \begin{cases} \frac{1}{n}\sum_{j=1}^n \hat{x}_j^t & \text{with probability } p \\ \hat{x}_i^t & \text{with probability } 1-p \end{cases}$

Mishchenko, Malinovsky, Stich, Richtárik, “ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration!,” ICML 2022

$p = \frac{1}{\sqrt{\kappa}}$ ☞ TotalCom $\mathcal{O}\left(d\sqrt{\kappa}\log\epsilon^{-1}\right)$

Scaffnew: **Variance-reduced** local GD



$$\hat{x}_1^t = x_1^t - \gamma \nabla f_1(x_1^t) + \gamma u_1^t \qquad \cdots \qquad \hat{x}_n^t = x_n^t - \gamma \nabla f_n(x_n^t) + \gamma u_n^t$$

Probabilistic Com: $x_i^{t+1} := \begin{cases} \frac{1}{n}\sum_{j=1}^n \hat{x}_j^t & \text{with probability } p \\ \hat{x}_i^t & \text{with probability } 1-p \end{cases}$

Mishchenko, Malinovsky, Stich, Richtárik, “ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration!,” ICML 2022

LC and Richtárik, “RandProx: Primal-Dual Optimization Algorithms with Randomized Proximal Updates,” ICLR 2023

PDDY

PDDY

for $t = 0, 1, \ldots$ **do**

$$\hat{x}^t := \mathrm{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$$

$$u^{t+1} := \mathrm{prox}_{\tau g^*}\left(u^t + \tau L \hat{x}^t\right)$$

$$x^{t+1} := \hat{x}^t - \gamma L^*(u^{t+1} - u^t)$$

end for

PDDY

PDDY

for $t = 0, 1, \ldots$ **do**

$$\hat{x}^t := \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$$

$$u^{t+1} := \operatorname{prox}_{\tau g^*}\left(u^t + \tau L \hat{x}^t\right)$$

$$x^{t+1} := \hat{x}^t - \gamma L^*(u^{t+1} - u^t)$$

end for

$\operatorname{prox}_{\tau g^*}$ can be costly

RandProx

RandProx

for $t = 0, 1, \ldots$ **do**

$$\hat{x}^t := \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$$

$$u^{t+1} := u^t + \frac{1}{1+\omega}\mathcal{R}^t\left(\operatorname{prox}_{\tau g^*}(u^t + \tau L \hat{x}^t) - u^t\right)$$

$$x^{t+1} := \hat{x}^t - \gamma(1+\omega)L^*(u^{t+1} - u^t)$$

end for

$$\mathbb{E}\left[\mathcal{R}(r^t)\right] = r^t \quad \text{and} \quad \mathbb{E}\left[\left\|\mathcal{R}(r^t) - r^t\right\|^2\right] \leq \omega \left\|r^t\right\|^2$$

LC, Richtárik, "RandProx: Primal-Dual Optimization Algorithms with Randomized Proximal Updates," ICLR, 2023

RandProx

$$\min \; h + f + \sum_{i=1}^{n} g_i$$

RandProx-minibatch

for $t = 0, 1, \ldots$ **do**
  $\hat{x}^t := \mathrm{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma v^t\right)$
  pick $\Omega^t \subset \{1, \ldots, n\}$ of size s unif. at random
  for $i \in \Omega^t$ **do**
    $u_i^{t+1} := \mathrm{prox}_{\frac{1}{\gamma n} g_i^*}(u_i^t + \frac{1}{\gamma n}\hat{x}^t)$
  end for
  for $i \in \{1, \ldots, n\} \backslash \Omega^t$ **do**
    $u_i^{t+1} := u_i^t$
  end for
  $v^{t+1} := \sum_{i=1}^{n} u_i^{t+1}$
  $x^{t+1} := \hat{x}^t - \frac{\gamma n}{s}(v^{t+1} - v^t)$
end for

$\mathcal{R}^t$: sampling

RandProx

$$\min \; h + f + \sum_{i=1}^{n} g_i$$

RandProx-minibatch

for $t = 0, 1, \ldots$ **do**
  $\hat{x}^t := \mathrm{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma v^t\right)$
  pick $\Omega^t \subset \{1, \ldots, n\}$ of size s unif. at random
  for $i \in \Omega^t$ **do**
    $u_i^{t+1} := \mathrm{prox}_{\frac{1}{\gamma n} g_i^*}(u_i^t + \frac{1}{\gamma n}\hat{x}^t)$
  end for
  for $i \in \{1, \ldots, n\} \backslash \Omega^t$ **do**
    $u_i^{t+1} := u_i^t$
  end for
  $v^{t+1} := \sum_{i=1}^{n} u_i^{t+1}$
  $x^{t+1} := \hat{x}^t - \frac{\gamma n}{s}(v^{t+1} - v^t)$
end for

$\mathcal{R}^t$: sampling

☞ $\sqrt{n}$ acceleration

RandProx

RandProx-skip

for $t = 0, 1, \ldots$ **do**
  Flip a coin θ^t = (1 with probability p, 0 else)
  if $\theta^t = 1$ **then**
    $\hat{x}^t := \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$
    $u^{t+1} := \operatorname{prox}_{\tau g^*}(u^t + \tau L \hat{x}^t)$
    $x^{t+1} := \hat{x}^t - \frac{\gamma}{p} L^*(u^{t+1} - u^t)$
  else
    $x^{t+1} := \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$
    $u^{t+1} := u^t$
  end if
end for

$$\mathcal{R}^t : r^t \mapsto \begin{cases} \frac{1}{p} r^t, & \text{proba. } p \\ 0, & \text{proba. } 1-p \end{cases}$$

RandProx

RandProx-skip

for $t = 0, 1, \ldots$ **do**
  Flip a coin θ^t = (1 with probability p, 0 else)
  if $\theta^t = 1$ **then**
    $\hat{x}^t := \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$
    $u^{t+1} := \operatorname{prox}_{\tau g^*}(u^t + \tau L \hat{x}^t)$
    $x^{t+1} := \hat{x}^t - \frac{\gamma}{p} L^*(u^{t+1} - u^t)$
  else
    $x^{t+1} := \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$
    $u^{t+1} := u^t$
  end if
end for

$\mathcal{R}^t : r^t \mapsto$
$\begin{cases} \frac{1}{p} r^t, & \text{proba. } p \\ 0, & \text{proba. } 1-p \end{cases}$

☞ $\sqrt{\kappa}$ acceleration

RandProx

RandProx-skip

for $t = 0, 1, \ldots$ **do**
  Flip a coin θ^t = (1 with probability p, 0 else)
  if $\theta^t = 1$ **then**
    $\hat{x}^t := \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$
    $u^{t+1} := \operatorname{prox}_{\tau g^*}(u^t + \tau L \hat{x}^t)$
    $x^{t+1} := \hat{x}^t - \frac{\gamma}{p} L^*(u^{t+1} - u^t)$
  else
    $x^{t+1} := \operatorname{prox}_{\gamma f}\left(x^t - \gamma \nabla h(x^t) - \gamma L^* u^t\right)$
    $u^{t+1} := u^t$
  end if
end for

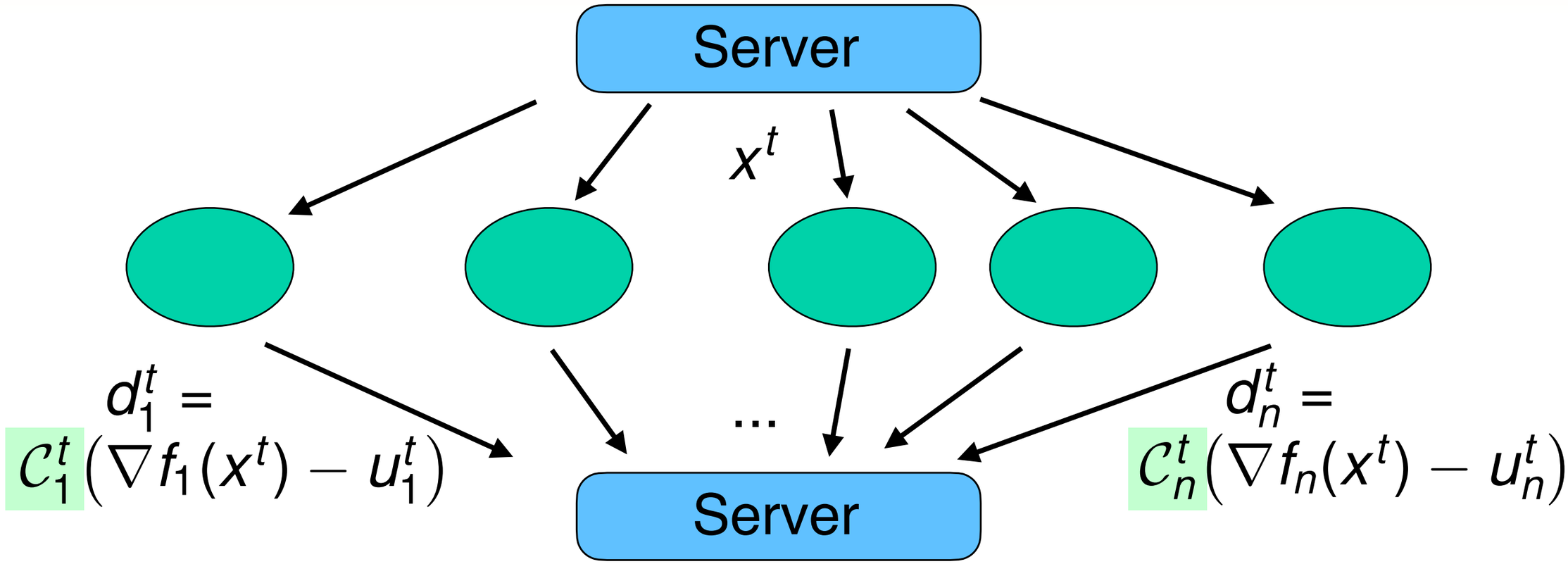
$\mathcal{R}^t : r^t \mapsto$
$\begin{cases} \frac{1}{p} r^t, & \text{proba. } p \\ 0, & \text{proba. } 1-p \end{cases}$

☞ $\sqrt{\kappa}$ acceleration

→ Scaffnew

2.2) Compression

DIANA

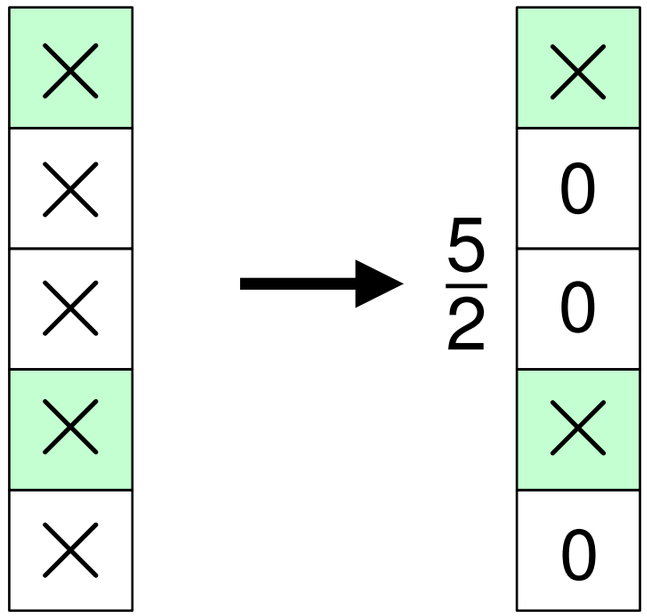


$$x^{t+1} := x^t - \gamma g^t \quad \text{with} \quad g^t = u^t + \frac{1}{n}\sum_{i=1}^{n} d_i^t$$

Mishchenko, Gorbunov, Takáč, Richtárik, “Distributed Learning with Compressed Gradient Differences,” 2019, published in 2024

Unbiased random compression

- `rand-`k: k elements out of d chosen unif. at random and scaled by $\frac{d}{k}$, other ones set to 0.



Unbiased random compression

- `rand-`k: k elements out of d chosen unif. at random and scaled by $\frac{d}{k}$, other ones set to 0.

- quantization

$$\mathcal{C}(1.2) = \begin{cases} 1 \text{ with probability } \frac{4}{5} \\ 2 \text{ with probability } \frac{1}{5} \end{cases}$$

Unbiased random compression

- `rand-`k: k elements out of d chosen unif. at random and scaled by $\frac{d}{k}$, other ones set to 0.

- quantization

$$\mathcal{C}(1.2) = \begin{cases} 1 \text{ with probability } \frac{4}{5} \\ 2 \text{ with probability } \frac{1}{5} \end{cases}$$

Albasyoni, Safaryan, LC, Richtárik, “Optimal Gradient Compression for Distributed and Federated Learning,” 2020

MURANA

MURANA

for $t = 0, 1, \ldots$ **do**
  for $i = 1, \ldots, n$ **do**
    $d_i^{t+1} := \mathcal{C}_i^t\left(\nabla f_i(x^t) - u_i^t\right)$
    $r_i^{t+1} := \mathcal{R}_i^t\left(\nabla f_i(x^t) - u_i^t\right)$
    $u_i^{t+1} := u_i^t + \lambda r_i^{t+1}$
  end for
  $d^{t+1} := \frac{1}{n}\sum_{i=1}^n d_i^{t+1}$
  $\tilde{x}^{t+1} := \mathrm{prox}_{\gamma R}\left(x^t - \gamma(u^t + d^{t+1})\right)$
  $x^{t+1} := x^t + \rho\mathcal{C}_s^t(\tilde{x}^{t+1} - x^t)$
  $u^{t+1} := u^t + \frac{\lambda}{n}\sum_{i=1}^n r_i^{t+1}$
end for

$$\min_{x\in\mathcal{X}} \frac{1}{n}\sum_{i=1}^n f_i(x) + R(x)$$

LC and Richtárik, "MURANA: A Generic Framework for Stochastic Variance-Reduced Optimization," MSML 2022

MURANA

MURANA

for $t = 0, 1, \ldots$ **do**
  for $i = 1, \ldots, n$ **do**
    $d_i^{t+1} := \mathcal{C}_i^t\left(\nabla f_i(x^t) - u_i^t\right)$
    $r_i^{t+1} := \mathcal{R}_i^t\left(\nabla f_i(x^t) - u_i^t\right)$
    $u_i^{t+1} := u_i^t + \lambda r_i^{t+1}$
  end for
  $d^{t+1} := \frac{1}{n}\sum_{i=1}^n d_i^{t+1}$
  $\tilde{x}^{t+1} := \mathrm{prox}_{\gamma R}\left(x^t - \gamma(u^t + d^{t+1})\right)$
  $x^{t+1} := x^t + \rho\mathcal{C}_s^t(\tilde{x}^{t+1} - x^t)$
  $u^{t+1} := u^t + \frac{\lambda}{n}\sum_{i=1}^n r_i^{t+1}$
end for

$$\min_{x\in\mathcal{X}} \frac{1}{n}\sum_{i=1}^n f_i(x) + R(x)$$

- DIANA
- SAGA
- L-SVRG

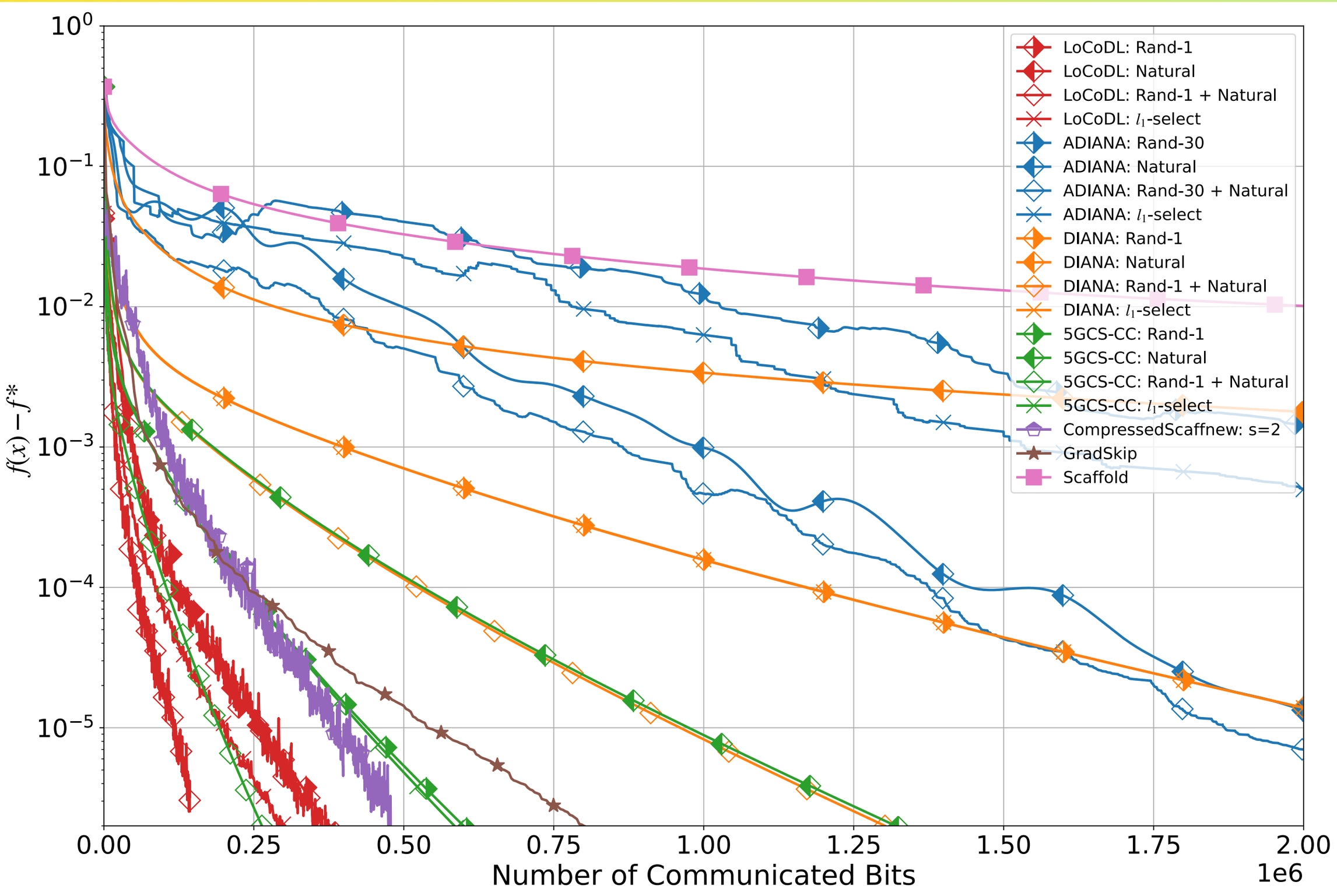
Double acceleration

Uplink communication complexity (UpCom):

- GD: $\mathcal{O}\left(d\kappa \log \epsilon^{-1}\right)$
- Scaffnew / AGD: $\mathcal{O}\left(d\sqrt{\kappa} \log \epsilon^{-1}\right)$
- DIANA: $\mathcal{O}\left(\left(\frac{d\kappa}{n} + \kappa + d\right) \log \epsilon^{-1}\right)$
- **CompressedScaffnew / TAMUNA / LoCoDL:**

$$\mathcal{O}\left(\left(\frac{d\sqrt{\kappa}}{\sqrt{n}} + \sqrt{d\kappa} + d\right) \log \epsilon^{-1}\right)$$

LC, Maranjyan, and Richtárik, “LoCoDL: Communication-Efficient Distributed Learning with Local Training and Compression,” ICLR 2025

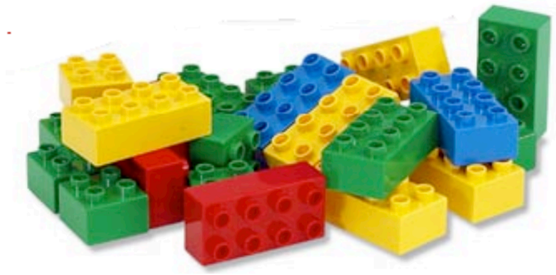


Logistic regression with the ‘a5a’ dataset of LibSVM. $d = 122$, $n = 288$

Summary

Splitting: decomposition as a sequence of simple operations.

For this, lifting using dual variables.



Summary

Splitting: decomposition as a sequence of simple operations. For this, lifting using dual variables.

Randomized operations (local training, compression, partial participation, sampling…) + **variance reduction** (via control variates = dual variables) ☞ **acceleration** 

3) Acceleration

Revisiting Nesterov acceleration

Nesterov, “A method of solving a convex programming problem with convergence rate O(1/k^2),” 1983

$$\underset{x\in\mathcal{X}}{\text{minimize}}\ \ \Psi(x) = f(x) + g(x)$$

APGD (2006)

$$\left\lfloor \begin{array}{l} y^t := \left(1 - \frac{1}{a_{t+1}}\right) x^t + \frac{1}{a_{t+1}} z^t \\ z^{t+1} := \operatorname{prox}_{a_{t+1}\gamma g}\left(z^t - a_{t+1}\gamma \nabla f(y^t)\right) \\ x^{t+1} := \left(1 - \frac{1}{a_{t+1}}\right) x^t + \frac{1}{a_{t+1}} z^{t+1} \end{array} \right.$$

With the right choice of $(a_t)_{t\geq 0}$ ☞ from $\mathcal{O}(1/t)$ to $\mathcal{O}(1/t^2)$ and from $\mathcal{O}\left(e^{-t/\kappa}\right)$ to $\mathcal{O}\left(e^{-t/\sqrt{\kappa}}\right)$ (**optimal**)

Revisiting Nesterov acceleration

Nesterov, “A method of solving a convex programming problem with convergence rate O(1/k^2),” 1983

$$\underset{x\in\mathcal{X}}{\text{minimize}}\ \ \Psi(x) = f(x) + g(x)$$

APGD (2006)

$$\left\lfloor \begin{aligned} y^t &:= \left(1 - \tfrac{1}{a_{t+1}}\right) x^t + \tfrac{1}{a_{t+1}} z^t \\ z^{t+1} &:= \operatorname{prox}_{a_{t+1}\gamma g}\left(z^t - a_{t+1}\gamma \nabla f(y^t)\right) \\ x^{t+1} &:= \left(1 - \tfrac{1}{a_{t+1}}\right) x^t + \tfrac{1}{a_{t+1}} z^{t+1} \end{aligned} \right.$$

Lemma 2.2 of LC et al., “A Nesterov-accelerated primal-dual splitting algorithm for convex nonsmooth optimization,” 2026:

$$\begin{aligned} &- a_{t+1}\left\langle \tilde{\nabla} g(z^{t+1}) + \nabla f(y^t), z^{t+1} - x^\star \right\rangle \\ &\leq -a_{t+1}^2\left(\Psi(x^{t+1}) - \Psi^\star\right) + (a_{t+1}^2 - a_{t+1})\left(\Psi(x^t) - \Psi^\star\right) + \tfrac{\beta_f}{2}\|z^{t+1} - z^t\|^2 \end{aligned}$$

APAPC

$$\underset{x\in\mathcal{X}}{\text{minimize}} \; f(x) + \frac{\mu_g}{2}\|x\|^2 + h(Lx)$$

APAPC

---

for $t = 0, 1, \ldots$ **do**

$$y^t := \left(1 - \frac{1}{a_{t+1}}\right) x^t + \frac{1}{a_{t+1}} z^t$$

$$\hat{z}^t := (1 + a_{t+1}\gamma\mu_g)^{-1} \left(z^t - a_{t+1}\gamma\nabla f(y^t) - a_{t+1}\gamma L^* v^t\right)$$

$$v^{t+1} := \operatorname{prox}_{\frac{\tau}{a_{t+1}} h^*} \left(v^t + \frac{\tau}{a_{t+1}} L\hat{z}^t\right)$$

$$z^{t+1} := (1 + a_{t+1}\gamma\mu_g)^{-1} \left(z^t - a_{t+1}\gamma\nabla f(y^t) - a_{t+1}\gamma L^* v^{t+1}\right)$$

$$x^{t+1} := \left(1 - \frac{1}{a_{t+1}}\right) x^t + \frac{1}{a_{t+1}} z^{t+1}$$

$$u^{t+1} := \left(1 - \frac{1}{a_{t+1}}\right) u^t + \frac{1}{a_{t+1}} v^{t+1}$$

end for

---

with **optimal** convergence rates in several cases

Perspectives: efficient algorithms

(compute, communication, energy, memory…)

- Main direction: design of **accelerated stochastic primal-dual** algorithms
- nonconvex / weakly convex settings
- decentralized optimization
- non-Euclidean geometry
- asynchronous algorithms
- applications to network communications, signal processing, medical imaging, operations research…