Generic acceleration schemes for gradient-based optimization in machine learning

Hongzhou Lin

PhD defense

Supervised by Zaid Harchaoui and Julien Mairal

Grenoble, France

Motivation: Large scale machine learning problems

Is there a Dog in the image

$\longrightarrow$

Yes

No

Input/Example

$a_i \in \mathbb{R}^d$

Output/Label.

$b_i \in \{-1, 1\}$ or $\mathbb{R}$

Motivation: Large scale machine learning problems

Is there a Dog in the image

$\longrightarrow$ { Yes / No

Supervised learning:
Model the dependence between examples a_i and labels b_i from a large labeled dataset $(a_i, b_i)_{i \in [1,n]}$.

Machine learning as optimization problems

Empirical Risk Minimization

Train a parametrized model h_x with parameters x.

$$\min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} L\left(h_x(a_i), b_i\right) + \psi(x),$$

Machine learning as optimization problems

Empirical Risk Minimization

Train a parametrized model h_x with parameters x.

$$\min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} L\left(h_x(a_i), b_i\right) + \psi(x),$$

where

- L is the loss function measuring the difference between the predicted label $h_x(a_i)$ and the true label b_i.

Machine learning as optimization problems

Empirical Risk Minimization

Train a parametrized model h_x with parameters x.

$$\min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} L\left(h_x(a_i), b_i\right) + \psi(x),$$

where

- L is the loss function measuring the difference between the predicted label $h_x(a_i)$ and the true label b_i.
- ψ is a regularization penalty.

Machine learning as optimization problems

Empirical Risk Minimization

Train a parametrized model h_x with parameters x.

$$\min_{x \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} L\left(h_x(a_i), b_i\right) + \psi(x),$$

where

- L is the loss function measuring the difference between the predicted label $h_x(a_i)$ and the true label b_i.
- ψ is a regularization penalty.
- Logistic regression, Support Vector Machine, artificial neural networks, etc.

Problem setting: convex composite finite sum problems

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\psi(x)\right\},$$

Problem setting: convex composite finite sum problems

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\psi(x)\right\},$$

where we assume

- f_i are convex, L-smooth, i.e.

$$\|\nabla f_i(y)-\nabla f_i(x)\|\leq L\|y-x\|,\quad \forall x,y.$$

Problem setting: convex composite finite sum problems

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\psi(x)\right\},$$

where we assume

- f_i are convex, L-smooth, i.e.

$$\|\nabla f_i(y)-\nabla f_i(x)\|\leq L\|y-x\|,\quad \forall x,y.$$

- ψ is convex, not necessarily differentiable.

Problem setting: convex composite finite sum problems

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\psi(x)\right\},$$

where we assume

- f_i are convex, L-smooth, i.e.

$$\|\nabla f_i(y)-\nabla f_i(x)\|\leq L\|y-x\|,\quad \forall x,y.$$

- ψ is convex, not necessarily differentiable.

- (optional) f_i are μ-strongly convex, i.e. $f_i(x)-\frac{\mu}{2}\|x\|^2$ is convex.

Problem setting: convex composite finite sum problems

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\psi(x)\right\},$$

where we assume

- f_i are convex, L-smooth, i.e.

$$\|\nabla f_i(y)-\nabla f_i(x)\|\leq L\|y-x\|,\quad \forall x,y.$$

Two major challenges:

- Large finite sum.
- Non smoothness of ψ.

Minimizing a finite sum problem

$$\min_{x\in\mathbb{R}^d} \left\{ f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right\}.$$

Stochastic Gradient methods

$$x_{k+1} = x_k - \eta_k \nabla f_{i_k}(x_k),$$

where i_k is randomly sampled from the set $\{1, \cdots, n\}$.

[Robbins and Monro, 1951, Nedić and Bertsekas, 2001,
LeCun and Bottou, 2004, Nemirovski et al., 2009, Agarwal et al., 2009]

Minimizing a finite sum problem

$$\min_{x\in\mathbb{R}^d} \left\{ f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right\}.$$

Stochastic Gradient methods

$$x_{k+1} = x_k - \eta_k \nabla f_{i_k}(x_k),$$

where i_k is randomly sampled from the set $\{1, \cdots, n\}$.

Convergence

- Constant stepsize $\Rightarrow$ Fast but does not converge in general.
- Diminishing stepsize $\Rightarrow$ Converges but slow.

[Robbins and Monro, 1951, Nedić and Bertsekas, 2001,
LeCun and Bottou, 2004, Nemirovski et al., 2009, Agarwal et al., 2009]

Incremental methods

Key idea: Variance reduction

$$x_{k+1} = x_k - \eta_k g_k,$$

with $\mathbb{E}[g_k] = \nabla f(x_k)$ and $\mathrm{Var}(g_k) \to 0$ when k grows.

Incremental methods

Key idea: Variance reduction

$$x_{k+1} = x_k - \eta_k g_k,$$

with $\mathbb{E}[g_k] = \nabla f(x_k)$ and $\mathrm{Var}(g_k) \to 0$ when k grows.

- SAG/SAGA, SVRG, SDCA, MISO/Finito.

[Shalev-Shwartz and Zhang, 2012, Schmidt et al., 2013,
Johnson and Zhang, 2013, Defazio et al., 2014a,b, Mairal, 2015]

Incremental methods

Key idea: Variance reduction

$$x_{k+1} = x_k - \eta_k g_k,$$

$$\text{with } \mathbb{E}[g_k] = \nabla f(x_k) \text{ and } \mathrm{Var}(g_k) \to 0 \text{ when } k \text{ grows.}$$

- SAG/SAGA, SVRG, SDCA, MISO/Finito.
- For strongly convex problems, linear convergence with constant stepsize vs sublinear convergence of SGD.

[Shalev-Shwartz and Zhang, 2012, Schmidt et al., 2013,
Johnson and Zhang, 2013, Defazio et al., 2014a,b, Mairal, 2015]

MISO: Minimization by Incremental Surrogate Optimization

objective :	$f = \frac{1}{n}\sum_{i=1}^{n} f_i$	f_1	f_2	f_3	f_4	$\cdots$	f_n
surrogate :	$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$	d_1	d_2	d_3	d_4	$\cdots$	d_n

Assumption

- Each f_i is L-smooth and μ-strongly convex.

Surrogates: quadratic lower bounds

- The surrogates d_i are quadratic lower bounds of f_i.

[Mairal, 2015]

MISO: Minimization by Incremental Surrogate Optimization

objective :	$f = \frac{1}{n}\sum_{i=1}^n f_i$	f_1	f_2	f_3	f_4	$\cdots$	f_n
surrogate :	$\bar{d}_{k-1} = \frac{1}{n}\sum_{i=1}^n d_i$	d_1	d_2	d_3	d_4	$\cdots$	d_n

Incremental update: iteration $k \geq 1$

- Randomly draw $i_k \in [1, n]$, say $i_k = 4$, then update

$$d_4(x) = f_4(x_{k-1}) + \nabla f_4(x_{k-1})^T(x - x_{k-1}) + \frac{\mu}{2}\|x - x_{k-1}\|^2.$$

MISO: Minimization by Incremental Surrogate Optimization

objective :	$f = \frac{1}{n}\sum_{i=1}^n f_i$	f_1	f_2	f_3	f_4	$\cdots$	f_n
surrogate :	$\bar{d}_k = \frac{1}{n}\sum_{i=1}^n d_i$	d_1	d_2	d_3	d_4	$\cdots$	d_n

Incremental update: iteration $k \geq 1$

- Randomly draw $i_k \in [1, n]$, say $i_k = 4$, then update

$$d_4(x) = f_4(x_{k-1}) + \nabla f_4(x_{k-1})^T(x - x_{k-1}) + \frac{\mu}{2}\|x - x_{k-1}\|^2.$$

- Aggregation: Update the surrogate $\bar{d}_k(x)$ and compute

$$x_k = \arg\min_{x \in \mathbb{R}^d} \{\bar{d}_k(x)\}$$

MISO: Minimization by Incremental Surrogate Optimization

objective :	$f = \frac{1}{n}\sum_{i=1}^{n} f_i$	f_1	f_2	f_3	f_4	$\cdots$	f_n
surrogate :	$\bar{d}_k = \frac{1}{n}\sum_{i=1}^{n} d_i$	d_1	d_2	d_3	d_4	$\cdots$	d_n

Convergence

When $n \geq 2\frac{L}{\mu}$, the algorithm converges linearly in expectation,

$$\mathbb{E}[f(x_k) - \bar{d}_k(x_k)] \leq C\left(1 - \frac{1}{3n}\right)^k (f(x_0) - \bar{d}_0(x_0)).$$

The quantity $f(x_k) - \bar{d}_k(x_k)$ is an optimality certificate since

$$f(x_k) - \bar{d}_k(x_k) \geq f(x_k) - f^*.$$

[Mairal, 2015]

Comparisons of complexity: when $\mu > 0$

Number of incremental gradient ∇f_i evaluated to obtain an ε-solution:

SGD	$O\left(\frac{1}{\varepsilon}\right)$
(Full) GD	$O\left(n\frac{L_f}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG, SAG, SAGA, SDCA, MISO, Finito	$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$

[Bottou et al., 2016, De Klerk et al., 2017, Taylor et al., 2017]

with diminishing stepsize

L_f is global Lipschitz constant, L is incremental Lipschitz constant.

Comparisons of complexity: when $\mu > 0$

Number of incremental gradient ∇f_i evaluated to obtain an ε-solution:

SGD	$O\left(\frac{1}{\varepsilon}\right)$
(Full) GD	$O\left(n\frac{L_f}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG, SAG, SAGA, SDCA, MISO, Finito	$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$

When $n = 10^4 = \frac{L}{\mu} = \frac{L_f}{\mu}$, we have

$$n\frac{L_f}{\mu} = 10^8 \quad >> \quad \max\left(n, \frac{L}{\mu}\right) = 10^4.$$

with diminishing stepsize

L_f is global Lipschitz constant, L is incremental Lipschitz constant.

Composite finite sum problems

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\psi(x)\right\},$$

Proximal operator

Given a convex function ψ, the proximal operator is defined by

$$\mathrm{prox}_{\psi}(x)=\arg\min_{z\in\mathbb{R}^d}\left\{\psi(z)+\frac{1}{2}\|z-x\|^2\right\}.$$

Composite finite sum problems

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\psi(x)\right\},$$

Proximal operator

Given a convex function ψ, the proximal operator is defined by

$$\mathrm{prox}_{\psi}(x)=\arg\min_{z\in\mathbb{R}^d}\left\{\psi(z)+\frac{1}{2}\|z-x\|^2\right\}.$$

- Closed form solution when ψ is simple.

[Beck and Teboulle, 2009, Wright et al., 2009, Raguet et al., 2013],...

Composite finite sum problems

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{n}\sum_{i=1}^{n}f_i(x)+\psi(x)\right\},$$

Proximal operator

Given a convex function ψ, the proximal operator is defined by

$$\mathrm{prox}_{\psi}(x)=\arg\min_{z\in\mathbb{R}^d}\left\{\psi(z)+\frac{1}{2}\|z-x\|^2\right\}.$$

- Closed form solution when ψ is simple.
- Gradient methods $\rightarrow$ proximal gradient methods.

[Beck and Teboulle, 2009, Wright et al., 2009, Raguet et al., 2013],...

Composite finite sum problems

$$\min_{x\in\mathbb{R}^d} \left\{ f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) + \psi(x) \right\},$$

Proximal operator

Given a convex function ψ, the proximal operator is defined by

$$\mathrm{prox}_{\psi}(x) = \arg\min_{z\in\mathbb{R}^d} \left\{ \psi(z) + \frac{1}{2}\|z-x\|^2 \right\}.$$

- Closed form solution when ψ is simple.

- Not all incremental methods enjoys proximal variant.
- **Contribution 1: Develop a proximal variant of MISO with theoretical guarantee.**

Acceleration: $\mu > 0$

GD	$O\left(n\frac{L_f}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$
Acc-GD	$O\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG, SAG, SAGA, SDCA, MISO, Finito	$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$

[Nesterov, 1983, 2004, 2007, Beck and Teboulle, 2009]

Acceleration: $\mu > 0$

GD	$O\left(n\frac{L_f}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$
Acc-GD	$O\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG, SAG, SAGA, SDCA, MISO, Finito	$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$

When $n = 10^4 = \frac{L}{\mu} = \frac{L_f}{\mu}$, we have

$$n\frac{L_f}{\mu} = 10^8 \quad > \quad n\sqrt{\frac{L_f}{\mu}} = 10^6.$$

[Nesterov, 1983, 2004, 2007, Beck and Teboulle, 2009]

Acceleration: $\mu > 0$

GD	$O\left(n\frac{L_f}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$
Acc-GD	$O\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$
SVRG, SAG, SAGA, SDCA, MISO, Finito	$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$

Can we do better for incremental methods?

Contribution: Two generic acceleration schemes in both convex and strongly convex settings

- **Catalyst (Nesterov's acceleration);**
- **QuickeNing (Quasi Newton).**

Main Idea:

Acceleration by Smoothing

Main Idea:

Acceleration by Smoothing

1. **Construct a smooth problem F equivalent to f.**

Main Idea:

Acceleration by Smoothing

1. Construct a smooth problem F equivalent to f.
2. Apply smooth optimization methods on F.

Smoothing via Moreau-Yosida Regularization

The Moreau-Yosida Regularization (Moreau envelope)

Given $f : \mathbb{R}^d \to \mathbb{R}$ a convex function, the Moreau-Yosida regularization of f is the function $F : \mathbb{R}^d \to \mathbb{R}$ defined by

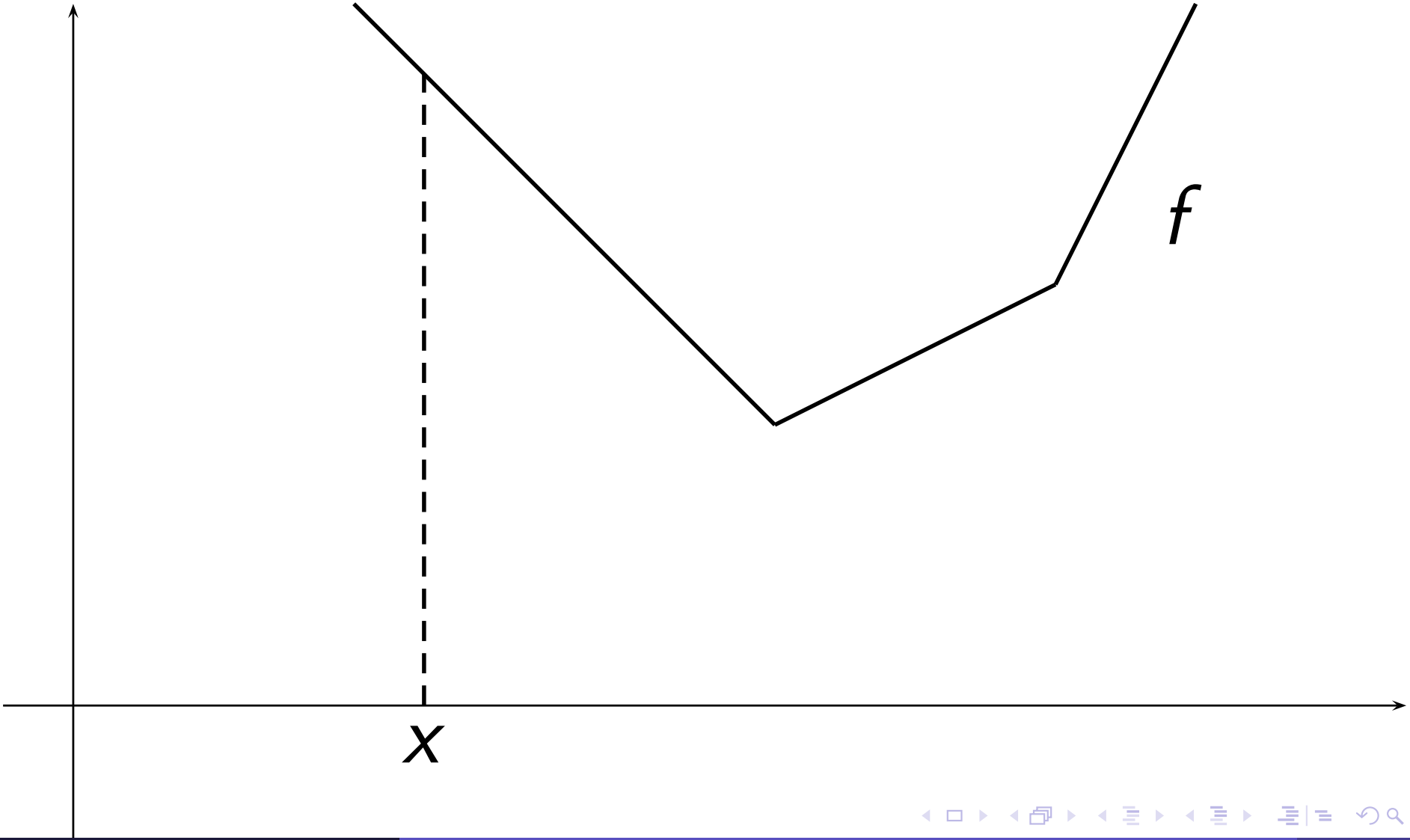
$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\}, \tag{1}$$

and the proximal operator $p(x)$ is the unique minimizer of (1).

[Moreau, 1965, Yosida, 1980]

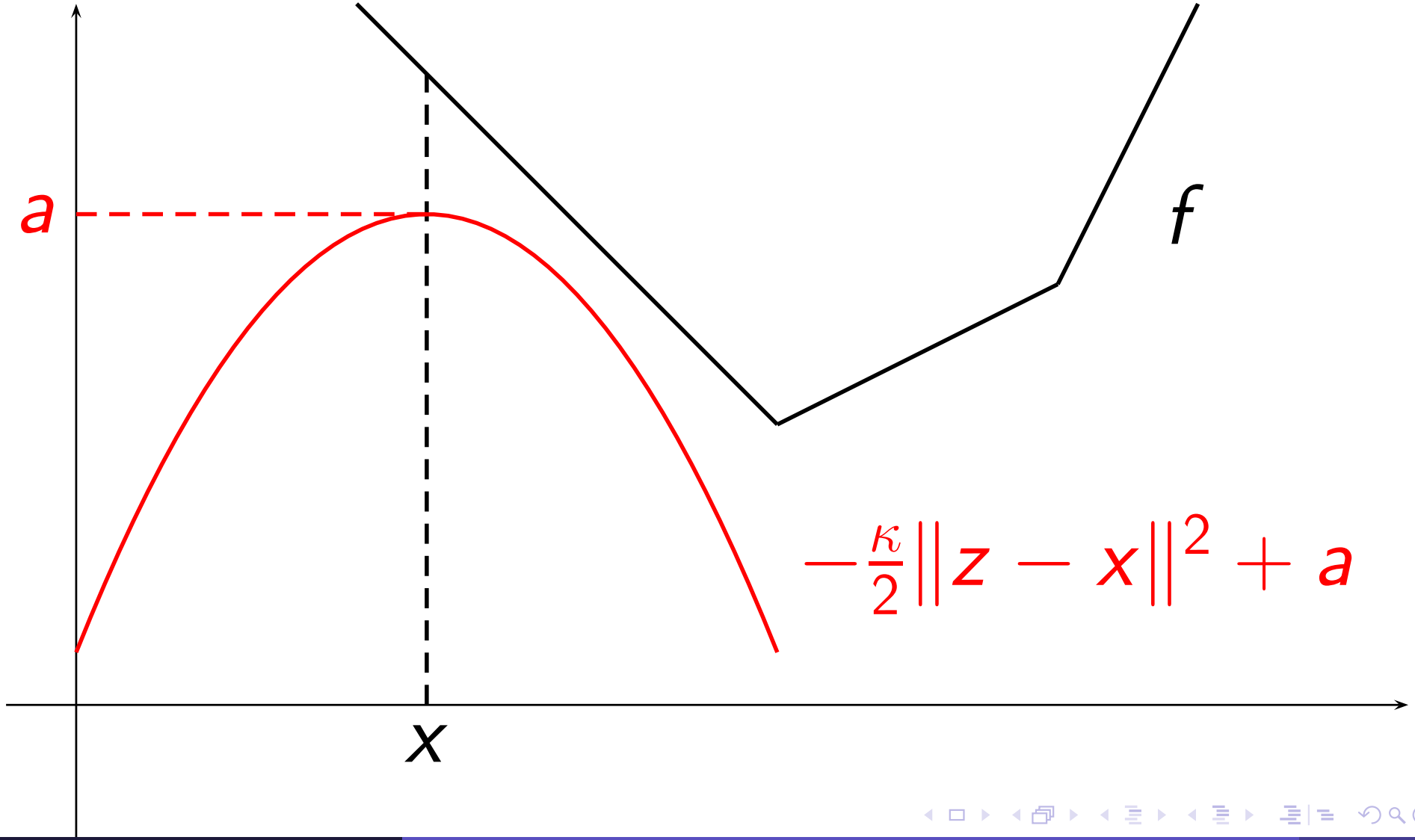
Geometric interpretation

Dual formulation: $F(x) = \max\left\{a \in \mathbb{R} \mid \forall z,\ -\frac{\kappa}{2}\|z-x\|^2 + a \le f(z)\right\}$



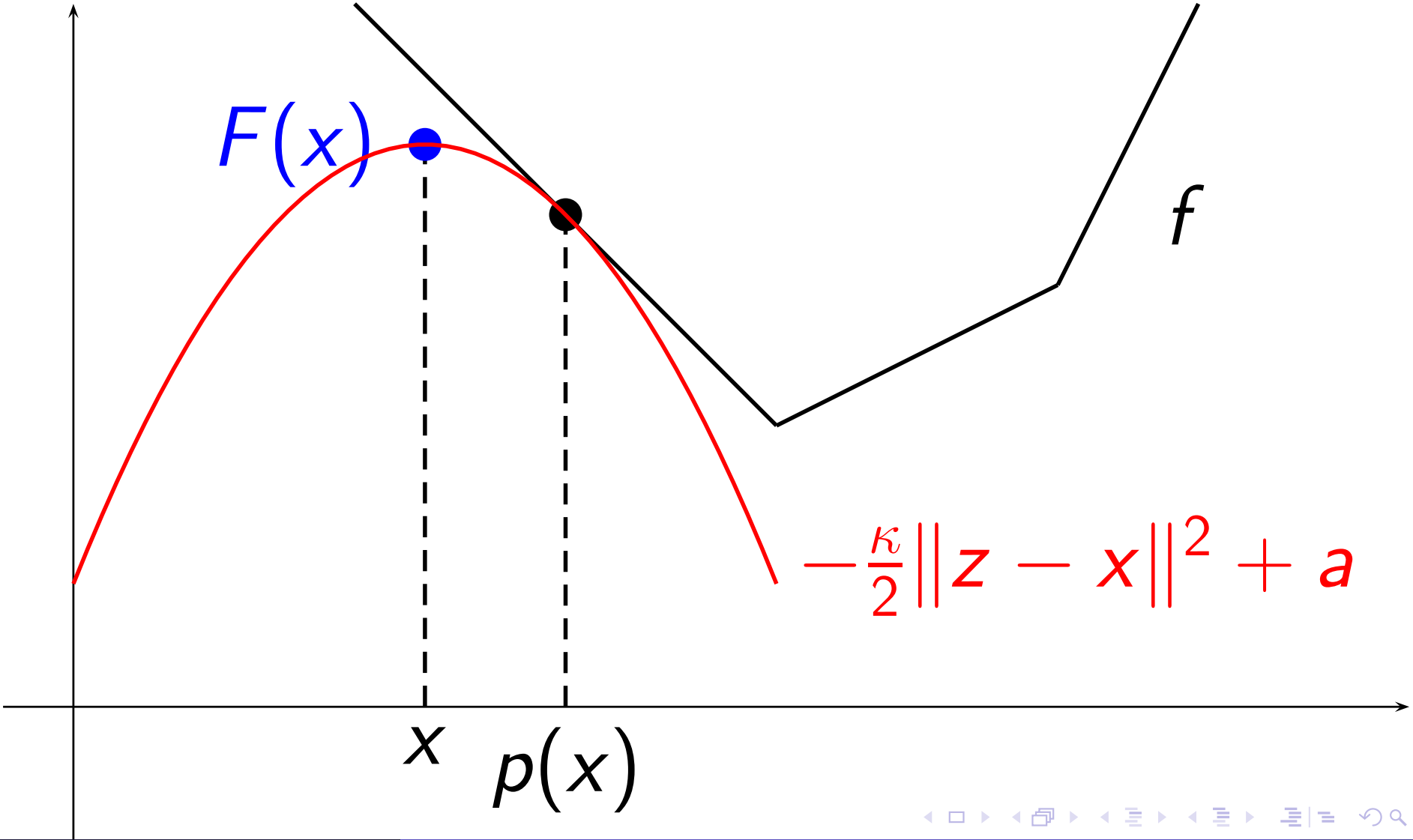
Geometric interpretation

Dual formulation: $F(x) = \max\left\{a \in \mathbb{R} \mid \forall z,\ -\frac{\kappa}{2}\|z-x\|^2 + a \leq f(z)\right\}$

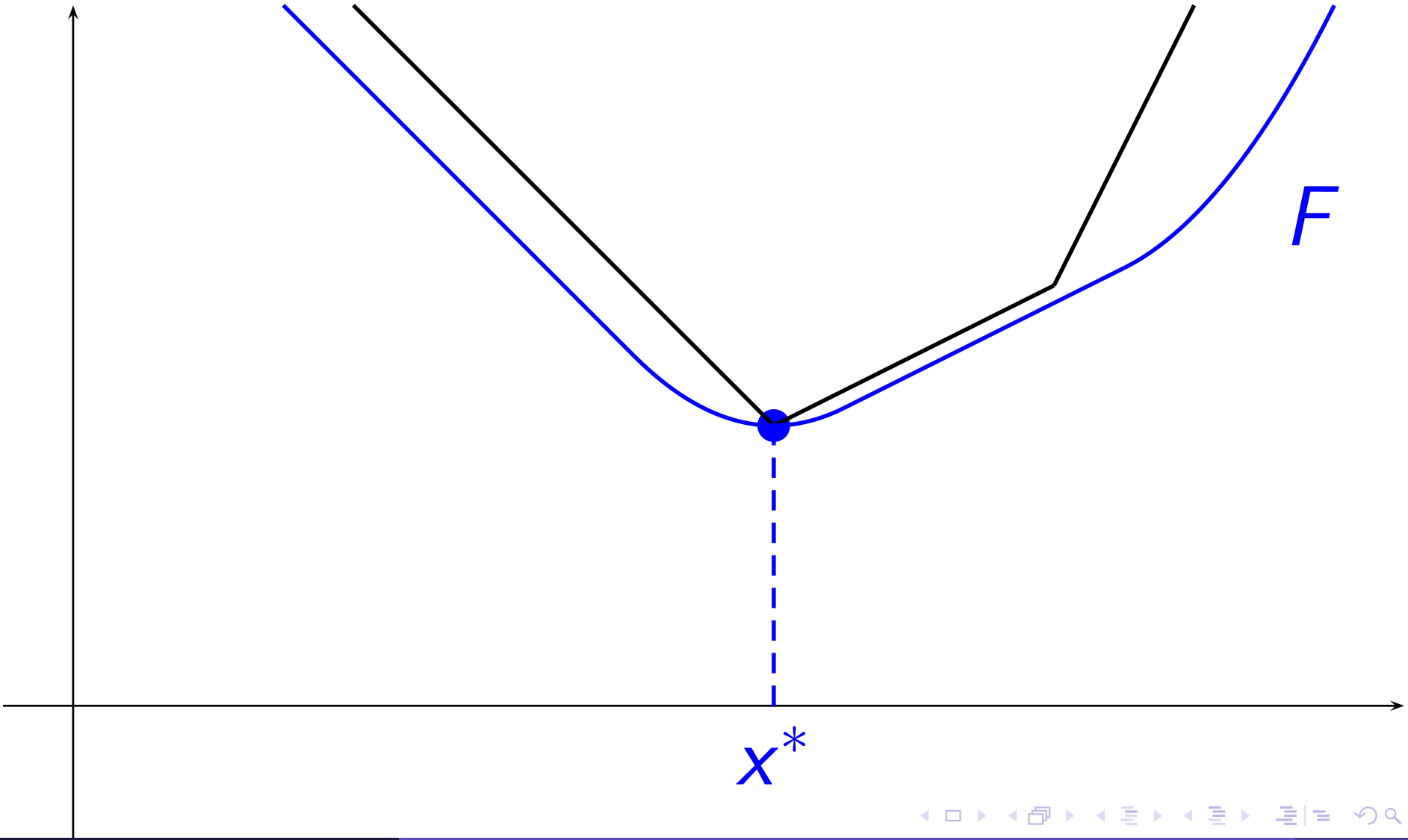


Geometric interpretation

Dual formulation: $F(x) = \max\left\{a \in \mathbb{R} \mid \forall z,\ -\frac{\kappa}{2}\|z - x\|^2 + a \leq f(z)\right\}$



Geometric interpretation



The Moreau-Yosida Regularization

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

Basic properties

The Moreau-Yosida Regularization

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

Basic properties

F is convex, differentiable,

The Moreau-Yosida Regularization

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

Basic properties

F is convex, differentiable,

$$\min_{x \in \mathbb{R}^d} F(x) = \min_{x \in \mathbb{R}^d} f(x),$$

and the solution set of the two problems coincide with each other.

The Moreau-Yosida Regularization

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

Basic properties

F is convex, differentiable,

$$\min_{x \in \mathbb{R}^d} F(x) = \min_{x \in \mathbb{R}^d} f(x),$$

and the solution set of the two problems coincide with each other.

$$\nabla F(x) = \kappa(x - p(x)). \tag{2}$$

∇F is Lipschitz continuous with $L_F = \kappa$.

The Moreau-Yosida Regularization

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

Basic properties

F is convex, differentiable,

$$\min_{x \in \mathbb{R}^d} F(x) = \min_{x \in \mathbb{R}^d} f(x),$$

and the solution set of the two problems coincide with each other.

$$\nabla F(x) = \kappa(x - p(x)). \tag{2}$$

∇F is Lipschitz continuous with $L_F = \kappa$.

f μ-strongly convex $\Rightarrow$ F μ_F-strongly convex with $\mu_F = \dfrac{\mu\kappa}{\mu + \kappa}$.

The Moreau-Yosida Regularization

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2}\|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

Basic properties

F is convex, differentiable,

$$\min_{x \in \mathbb{R}^d} F(x) = \min_{x \in \mathbb{R}^d} f(x),$$

and the solution set of the two problems coincide with each other.

$$\nabla F(x) = \kappa(x - p(x)). \tag{2}$$

F enjoys nice properties: smoothness and strong convexity.

Perform fast smooth optimization methods to minimize F.

Accelerated proximal point algorithm

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2}\|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

$$\nabla F(x) = \kappa(x - p(x)).$$

Apply accelerated gradient descent on Moreau-Yosida Regularization

$$x_{k+1} = y_k - \frac{1}{\kappa}\nabla F(y_k) = p(y_k), \quad y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k).$$

[Güler, 1992, Salzo and Villa, 2012, He and Yuan, 2012, Devolder et al., 2014]

Accelerated proximal point algorithm

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2}\|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

$$\nabla F(x) = \kappa(x - p(x)).$$

Apply accelerated gradient descent on Moreau-Yosida Regularization

$$x_{k+1} = y_k - \frac{1}{\kappa}\nabla F(y_k) = p(y_k), \quad y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k).$$

However, $p(y_k)$ does not have closed form solution!

[Güler, 1992, Salzo and Villa, 2012, He and Yuan, 2012, Devolder et al., 2014]

Accelerated proximal point algorithm

$$F(x) = \min_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{\kappa}{2}\|z - x\|^2 \right\} \quad \text{with minimizer} \quad p(x).$$

$$\nabla F(x) = \kappa(x - p(x)).$$

Apply accelerated gradient descent on Moreau-Yosida Regularization

$$x_{k+1} = y_k - \frac{1}{\kappa}\nabla F(y_k) = p(y_k), \quad y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k).$$

Main recipe

- Apply a first-order method $\mathcal{M}$ to approximately solve $p(y_k)$.
- Carefully control the magnitude of inexactness.

Catalyst is coming

Main algorithm

Catalyst, a meta-algorithm

Input: an "appropriate" optimization method $\mathcal{M}$.

- At iteration k, apply $\mathcal{M}$ to find

$$x_k \approx \arg\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$$

  such that $h_k(x_k) - h_k^* \leq \varepsilon_k$.

- Then compute the next prox-center y_k using an extrapolation step

$$y_k = x_k + \beta_k(x_k - x_{k-1}).$$

Main algorithm

Catalyst, a meta-algorithm

Input: an "appropriate" optimization method $\mathcal{M}$.

- At iteration k, apply $\mathcal{M}$ to find

$$x_k \approx \arg\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$$

  such that $h_k(x_k) - h_k^* \leq \varepsilon_k$.

- Then compute the next prox-center y_k using an extrapolation step

Catalyst is an instance of inexact accelerated proximal point algorithm [Güler, 1992].

Contribution:

- Provide specific choices of all parameters $\kappa, \epsilon_k, \beta_k$.
- Provide global complexity analysis showing the **acceleration**.

Appropriate $\mathcal{M}$ = Linear convergence rate

Requirement on $\mathcal{M}$

- When h is strongly convex, $\mathcal{M}$ produces a sequence of iterates $(z_t)_{t\geq 0}$ such that

$$h(z_t) - h^\star \leq C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^t(h(z_0) - h^\star).$$

or

$$\mathbb{E}[h(z_t)] - h^\star \leq C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^t(h(z_0) - h^\star).$$

Appropriate $\mathcal{M}$ = Linear convergence rate

Requirement on $\mathcal{M}$

- When h is strongly convex, $\mathcal{M}$ produces a sequence of iterates $(z_t)_{t\geq 0}$ such that

$$h(z_t) - h^\star \leq C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^t(h(z_0) - h^\star).$$

or

$$\mathbb{E}[h(z_t)] - h^\star \leq C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^t(h(z_0) - h^\star).$$

$\tau_{\mathcal{M}}$ usually depends on the **condition number** L/μ, e.g.,

- $\tau_{\mathcal{M}} = \frac{\mu}{L_f}$ for GD,
- $\tau_{\mathcal{M}} = \min\left\{\frac{1}{2n}, \frac{\mu}{4L}\right\}$ for MISO.

Appropriate $\mathcal{M}$ = Linear convergence rate

Requirement on $\mathcal{M}$

- When h is strongly convex, $\mathcal{M}$ produces a sequence of iterates $(z_t)_{t \geq 0}$ such that

$$h(z_t) - h^\star \leq C_{\mathcal{M}}(1 - \tau_{\mathcal{M}})^t (h(z_0) - h^\star).$$

or

$$\mathbb{E}[h(z_t)] - h^\star \leq C_{\mathcal{M}}(1 - \tau_{\mathcal{M}})^t (h(z_0) - h^\star).$$

Important remarks:

- We do not make any assumption on the convergence rate for non strongly convex objectives.
- Catalyst provides support to non strongly convex functions even $\mathcal{M}$ is not defined for $\mu = 0$.

Applications

			With Catalyst	
	$\mu > 0$	$\mu = 0$	$\mu > 0$	$\mu = 0$
(Full) GD	$O\left(n\frac{L}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)^a$	$\tilde{O}\left(n\sqrt{\frac{L_f}{\varepsilon}}\right)$
SAG/SAGA	$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(\max\left(n, \sqrt{\frac{nL}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$\tilde{O}\left(\sqrt{\frac{nL}{\varepsilon}}\right)$
MISO		not avail.		
SDCA				
SVRG				
Acc-FG	$O\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L_f}{\sqrt{\varepsilon}}\right)$	no acceleration	
Acc-SDCA	$\tilde{O}\left(\max\left(n, \sqrt{\frac{nL}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	not avail.		

[a] where $\tilde{O}$ hides logarithmic factors.

Applications

			With Catalyst	
	$\mu > 0$	$\mu = 0$	$\mu > 0$	$\mu = 0$
(Full) GD	$O\left(n\frac{L}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)^a$	$\tilde{O}\left(n\sqrt{\frac{L_f}{\varepsilon}}\right)$
SAG/SAGA	$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(\max\left(n, \sqrt{\frac{nL}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$\tilde{O}\left(\sqrt{\frac{nL}{\varepsilon}}\right)$
MISO		not avail.		
SDCA				
SVRG				
Acc-FG	$O\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L_f}{\sqrt{\varepsilon}}\right)$	no acceleration	
Acc-SDCA	$\tilde{O}\left(\max\left(n, \sqrt{\frac{nL}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	not avail.		

- Acceleration occurs when $n < \dfrac{L}{\mu}$. If $n = 10^4$ and $\dfrac{L}{\mu} = 10^6$,

$$\max\left(n, \frac{L}{\mu}\right) = 10^6 \quad > \quad \max\left(n, \sqrt{n\frac{L}{\mu}}\right) = 10^5.$$

Applications

			With Catalyst	
	$\mu > 0$	$\mu = 0$	$\mu > 0$	$\mu = 0$
(Full) GD	$O\left(n\frac{L}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)^a$	$\tilde{O}\left(n\sqrt{\frac{L_f}{\varepsilon}}\right)$
SAG/SAGA	$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(\max\left(n, \sqrt{\frac{nL}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$\tilde{O}\left(\sqrt{\frac{nL}{\varepsilon}}\right)$
MISO		not avail.		
SDCA				
SVRG				
Acc-FG	$O\left(n\sqrt{\frac{L_f}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L_f}{\sqrt{\varepsilon}}\right)$	no acceleration	
Acc-SDCA	$\tilde{O}\left(\max\left(n, \sqrt{\frac{nL}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	not avail.		

- Catalyst is optimal up to logarithmic factors.
  [Woodworth and Srebro, 2016, Arjevani and Shamir, 2016]

[a] where $\tilde{O}$ hides logarithmic factors.

Convergence analysis when $\mu > 0$

A two-stage analysis

- **How many subproblems do we need to solve?**

  Roughly the same as Nesterov's method but with errors.

  Key: control the error accumulation.

Convergence analysis when $\mu > 0$

A two-stage analysis

- **How many subproblems do we need to solve?**

  Roughly the same as Nesterov's method but with errors.

  Key: control the error accumulation.

- **How many iterations of $\mathcal{M}$ do we need for each subproblem?**

  The required accuracy ε_k is decreasing.

  Key: warm start the subproblems.

How many subproblems do we need to solve?

If no error, Nesterov's method converges linearly with rate depending on the square root of the condition number of F:

$$\sqrt{q} = \sqrt{\frac{\mu_F}{L_F}} = \sqrt{\frac{\mu}{\mu + \kappa}}.$$

Accumulation of errors

Set $A_k = (1 - \sqrt{q})^k$, then the sequence $(x_k)_{k \in \mathbb{N}}$ satisfies

$$f(x_k) - f^* \leq 2(1 - \sqrt{q})^k (f(x_0) - f^*) \left(1 + 3 \sum_{j=1}^{k} \sqrt{\frac{\varepsilon_j}{2A_j(f(x_0) - f^*)}} \right)^2.$$

How many subproblems do we need to solve?

If no error, Nesterov's method converges linearly with rate depending on the square root of the condition number of F:

$$\sqrt{q} = \sqrt{\frac{\mu_F}{L_F}} = \sqrt{\frac{\mu}{\mu + \kappa}}.$$

Accumulation of errors

Set $A_k = (1 - \sqrt{q})^k$, then the sequence $(x_k)_{k \in \mathbb{N}}$ satisfies

$$f(x_k) - f^* < 2(1 - \sqrt{q})^k (f(x_0) - f^*) \left(1 + 3 \sum^{k} \sqrt{\varepsilon_j} \right)^2.$$

Choice of ε_k: Choose ε_k such that the series of errors converge

$$\varepsilon_k = \frac{2}{9}(f(x_0) - f^*)(1 - \rho)^k \quad \text{with} \quad \rho < \sqrt{q}.$$

(Remark: ε_k is in the same order as $f(x_k) - f^*$.)

How many iterations of $\mathcal{M}$ for each subproblem?

Main recipe

- Warm start the subproblem

$$\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\}$$

using the latest iterates $\approx \varepsilon_{k-1}$ solution.

How many iterations of $\mathcal{M}$ for each subproblem?

Main recipe

- Warm start the subproblem

$$\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\}$$

  using the latest iterates $\approx \varepsilon_{k-1}$ solution.
- Only need to decrease the fraction $\varepsilon_k/\varepsilon_{k-1}$.

How many iterations of $\mathcal{M}$ for each subproblem?

Main recipe

- Warm start the subproblem

$$\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\}$$

  using the latest iterates $\approx \varepsilon_{k-1}$ solution.

- Only need to decrease the fraction $\varepsilon_k/\varepsilon_{k-1}$.

A constant number of iterations

$$T_{\mathcal{M}} = \tilde{O}(\frac{1}{\tau_{\mathcal{M}}})$$

is enough to achieve an ε_k solution.

Towards the global complexity

Complexity to achieve an ε-solution of f

- The number of subproblems to be solved

$$O\left(\sqrt{\frac{\mu+\kappa}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right).$$

Towards the global complexity

Complexity to achieve an ε-solution of f

- The number of subproblems to be solved

$$O\left(\sqrt{\frac{\mu+\kappa}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right).$$

- The number of iterations of $\mathcal{M}$ to solve each subproblem

$$T_{\mathcal{M}} = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\right).$$

Towards the global complexity

Complexity to achieve an ε-solution of f

- The number of subproblems to be solved

$$O\left(\sqrt{\frac{\mu+\kappa}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right).$$

- The number of iterations of $\mathcal{M}$ to solve each subproblem

$$T_{\mathcal{M}} = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\right).$$

Global Complexity: the total number of iterations of $\mathcal{M}$ to guarantee $f(x_k) - f^\star \leq \varepsilon$ is at most

$$\tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\sqrt{\frac{\mu+\kappa}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$$

Example: apply Catalyst to MISO

Example: MISO

$$\tau_{\mathcal{M}} = \min\left\{\frac{1}{2n}, \frac{\mu+\kappa}{4(L+\kappa)}\right\}$$

Apply Catalyst yields

$$\tilde{O}\left(\max\left(n\sqrt{\frac{\mu+\kappa}{\mu}}, \frac{L+\kappa}{\sqrt{(\mu+\kappa)\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right).$$

Minimize the complexity respect to κ.

- When $n \geq \frac{L}{\mu}$, no acceleration.
- When $n < \frac{L}{\mu}$, minimum at $\kappa = \frac{(L-\mu)}{n-1} - \mu$, yielding

$$\tilde{O}\left(\sqrt{n\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right).$$

Example: apply Catalyst to MISO

Example: MISO

$$\tau_{\mathcal{M}} = \min\left\{\frac{1}{2n}, \frac{\mu+\kappa}{4(L+\kappa)}\right\}$$

Apply Catalyst yields

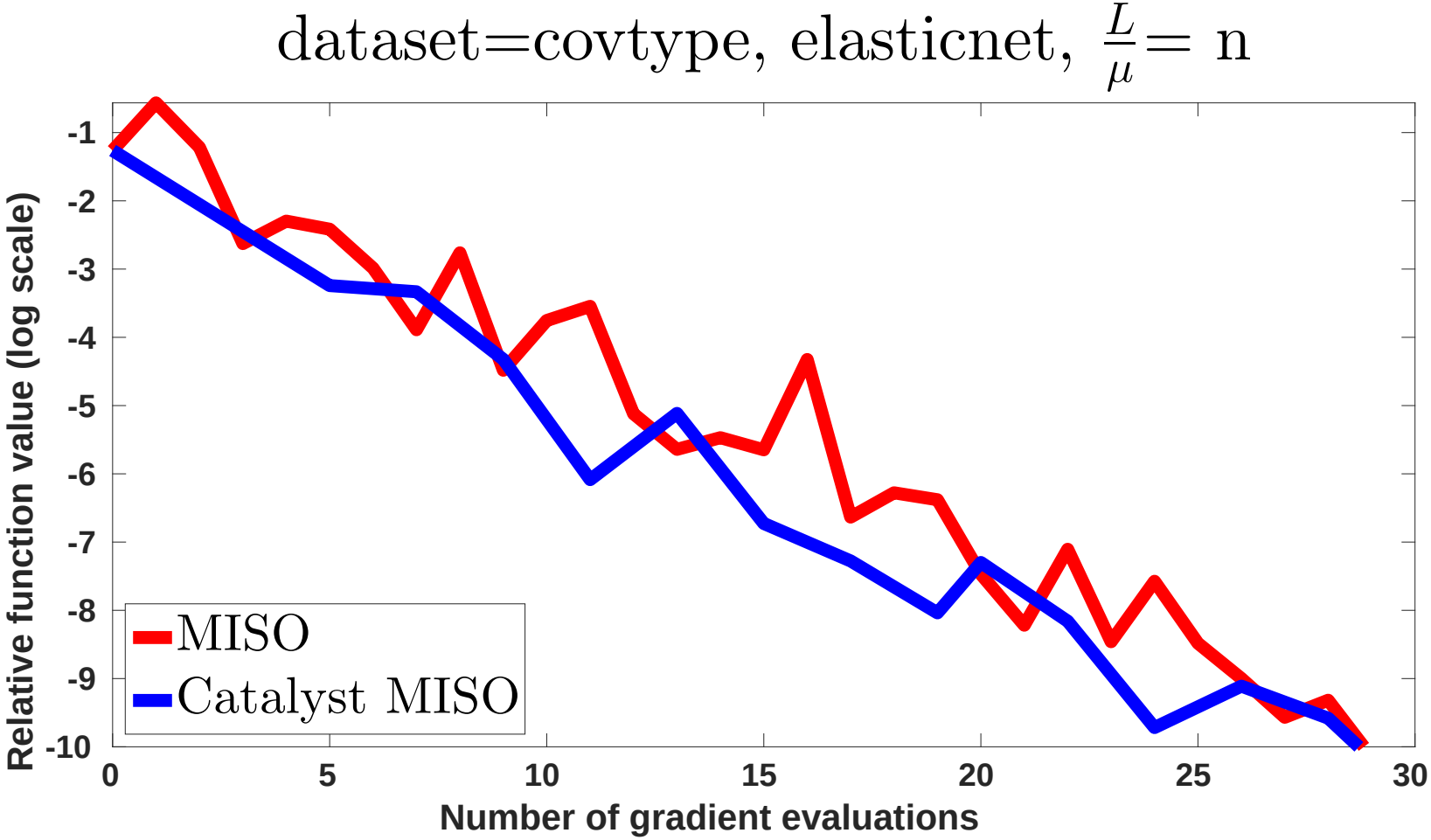
$$\tilde{O}\left(\max\left(n\sqrt{\frac{\mu+\kappa}{\mu}}, \frac{L+\kappa}{\sqrt{(\mu+\kappa)\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right).$$

Minimize the complexity respect to κ.

How to choose κ?

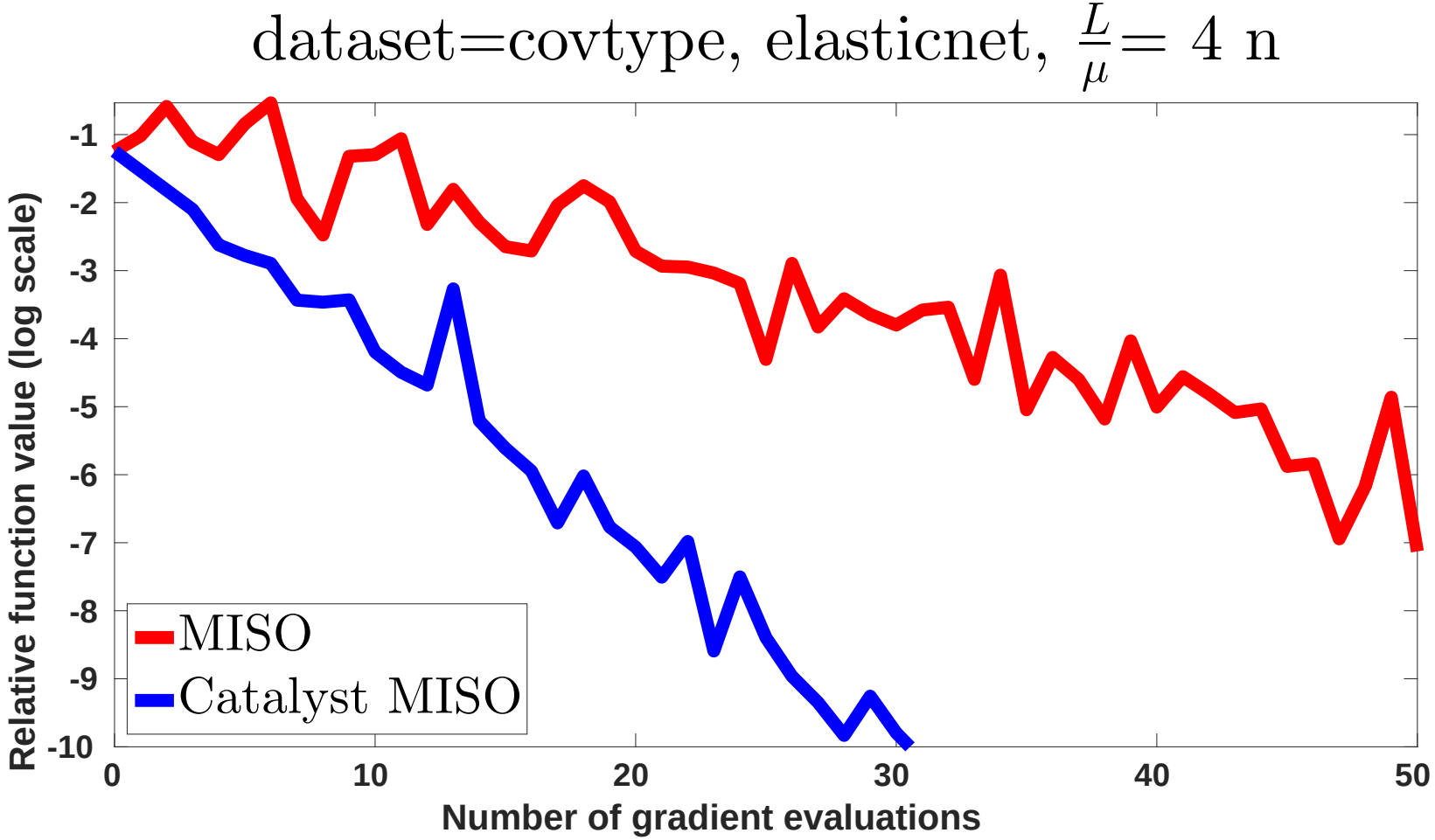
The global complexity of Catalyst is a function of κ, we choose it to minimize the global complexity.

Catalyst in practice: Elastic-net

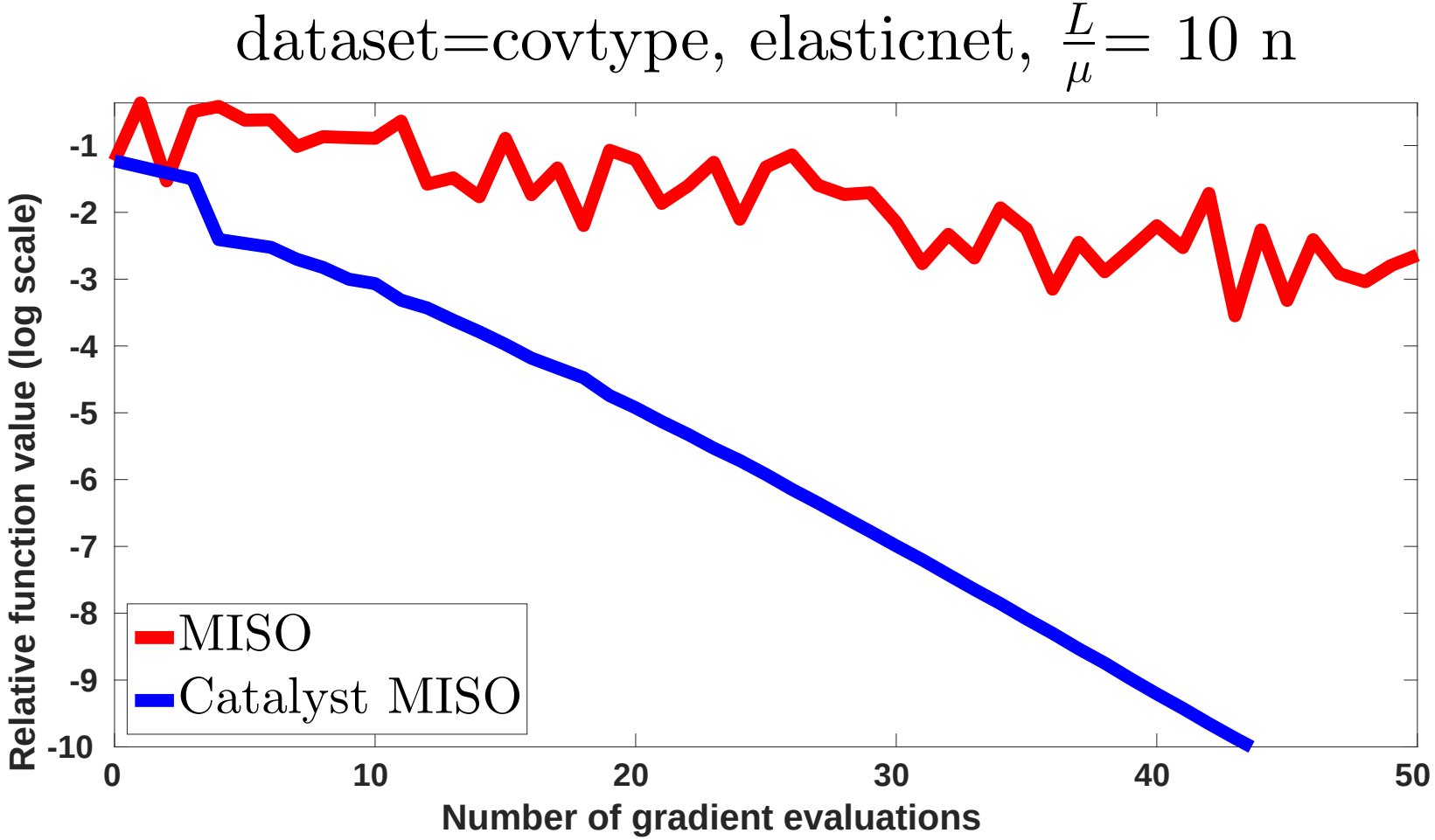


No acceleration when $n = \frac{L}{\mu}$!

Catalyst in practice: Elastic-net



Catalyst in practice: Elastic-net



Conclusions

- Significant improvement for ill-conditioned problems.
- Improve the numerical stability.

Summary: Catalyst

- Theoretical acceleration for both strongly convex and non strongly convex problems.
- Significant improvement in practice for ill-conditioned problems.

Summary: Catalyst

- Theoretical acceleration for both strongly convex and non strongly convex problems.
- Significant improvement in practice for ill-conditioned problems.

Can we do better?

- For worst case complexity, Catalyst is near optimal.
- Exploit curvature information may lead to better practical performance. $\rightarrow$ Quasi-Newton methods.

Variable metric proximal point algorithm

Apply Quasi Newton methods on Moreau-Yosida Regularization

$$x_{k+1} = x_k - \eta_k H_k \nabla F(x_k)$$

where H_k is an approximate inverse Hessian.

Variable metric proximal point algorithm

Apply Quasi Newton methods on Moreau-Yosida Regularization

$$x_{k+1} = x_k - \eta_k H_k \nabla F(x_k)$$

where H_k is an approximate inverse Hessian.

- $\nabla F(x_k)$ requires to solve $p(x_k)$.

  $\Rightarrow$ Use first order methods $\mathcal{M}$ to approximate.

[Fukushima and Qi, 1996, Chen and Fukushima, 1999, Burke and Qian, 2000, Fuentes et al., 2012]

Variable metric proximal point algorithm

Apply Quasi Newton methods on Moreau-Yosida Regularization

$$x_{k+1} = x_k - \eta_k H_k \nabla F(x_k)$$

where H_k is an approximate inverse Hessian.

- $\nabla F(x_k)$ requires to solve $p(x_k)$.

  $\Rightarrow$ Use first order methods $\mathcal{M}$ to approximate.

- Storing H_k is memory consuming.

  $\Rightarrow$ Use the limited memory variant L-BFGS.

[Broyden, 1970, Fletcher, 1970, Goldfarb, 1970, Shanno, 1970, Nocedal, 1980, Friedlander and Schmidt, 2012]

Variable metric proximal point algorithm

Apply Quasi Newton methods on Moreau-Yosida Regularization

$$x_{k+1} = x_k - \eta_k H_k \nabla F(x_k)$$

where H_k is an approximate inverse Hessian.

- $\nabla F(x_k)$ requires to solve $p(x_k)$.

  $\Rightarrow$ Use first order methods $\mathcal{M}$ to approximate.

- Saving H_k is ...

QuickeNing: Apply L-BFGS on F with inexact gradients.

QuickeNing: A Generic Quasi-Newton framework

- Perform a Quasi-Newton step

$$x_{k+1} = x_k - H_k g_k.$$

- Use $\mathcal{M}$ to approximate gradient and function value at x_{k+1},

$$g_{k+1} \approx \nabla F(x_{k+1}) \quad \text{and} \quad F_{k+1} \approx F(x_{k+1}).$$

- If $F_{k+1} > F_k - \frac{1}{2\kappa}\|g_k\|^2$,

$$\text{Reset} \quad x_{k+1} = x_k - \frac{1}{\kappa} g_k;$$

$$\text{Re-evaluate} \quad g_{k+1} \approx \nabla F(x_{k+1}) \text{ and } F_{k+1} \approx F(x_{k+1}).$$

- Construct H_{k+1} with L-BFGS update.

QuickeNing: A Generic Quasi-Newton framework

- Perform a Quasi-Newton step

$$x_{k+1} = x_k - H_k g_k.$$

- Use $\mathcal{M}$ to approximate gradient and function value at x_{k+1},

$$g_{k+1} \approx \nabla F(x_{k+1}) \quad \text{and} \quad F_{k+1} \approx F(x_{k+1}).$$

- If $F_{k+1} > F_k - \frac{1}{2\kappa}\|g_k\|^2$,

$$\text{Reset} \quad x_{k+1} = x_k - \frac{1}{\kappa} g_k;$$

$$\text{Re-evaluate} \quad g_{k+1} \approx \nabla F(x_{k+1}) \text{ and } F_{k+1} \approx F(x_{k+1}).$$

- Construct H_{k+1} with L-BFGS update.

QuickeNing: A Generic Quasi-Newton framework

- Perform a Quasi-Newton step

$$x_{k+1} = x_k - H_k g_k.$$

- Use $\mathcal{M}$ to approximate gradient and function value at x_{k+1},

$$g_{k+1} \approx \nabla F(x_{k+1}) \quad \text{and} \quad F_{k+1} \approx F(x_{k+1}).$$

- If $F_{k+1} > F_k - \frac{1}{2\kappa}\|g_k\|^2$,

$$\text{Reset} \quad x_{k+1} = x_k - \frac{1}{\kappa} g_k;$$

$$\text{Re-evaluate} \quad g_{k+1} \approx \nabla F(x_{k+1}) \text{ and } F_{k+1} \approx F(x_{k+1}).$$

- Construct H_{k+1} with L-BFGS update.

Towards the global complexity: $\mu > 0$

Complexity analysis

- Outer-loop: the number of subproblems to be solved

$$O\left(\frac{\mu+\kappa}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right). \quad \text{(not better than GD)}$$

Towards the global complexity: $\mu > 0$

Complexity analysis

- Outer-loop: the number of subproblems to be solved

$$O\left(\frac{\mu+\kappa}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right). \quad \text{(not better than GD)}$$

- Inner-loop: each subproblem requires at most $T_{\mathcal{M}}$ iterations of $\mathcal{M}$

$$T_{\mathcal{M}} = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\right).$$

Towards the global complexity: $\mu > 0$

Complexity analysis

- Outer-loop: the number of subproblems to be solved

$$O\left(\frac{\mu+\kappa}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right). \quad \text{(not better than GD)}$$

- Inner-loop: each subproblem requires at most $T_{\mathcal{M}}$ iterations of $\mathcal{M}$

$$T_{\mathcal{M}} = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\right).$$

Global Complexity: the total iterations of $\mathcal{M}$ to obtain an ε-solution is at most

$$\tilde{O}\left(\frac{\mu+\kappa}{\tau_{\mathcal{M}}\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$$

Example: apply QuickeNing to MISO

Example: MISO, The inner convergence rate is given

$$\tau_{\mathcal{M}} = \min\left\{\frac{1}{2n}, \frac{\mu+\kappa}{4(L+\kappa)}\right\}$$

Example: apply QuickeNing to MISO

Example: MISO, The inner convergence rate is given

$$\tau_{\mathcal{M}} = \min\left\{\frac{1}{2n}, \frac{\mu+\kappa}{4(L+\kappa)}\right\}$$

which yields the global complexity

$$\tilde{O}\left(\max\left(\frac{\mu+\kappa}{\mu}n, \frac{L+\kappa}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right).$$

Example: apply QuickeNing to MISO

Example: MISO, The inner convergence rate is given

$$\tau_{\mathcal{M}} = \min\left\{\frac{1}{2n}, \frac{\mu+\kappa}{4(L+\kappa)}\right\}$$

which yields the global complexity

$$\tilde{O}\left(\max\left(\frac{\mu+\kappa}{\mu}n, \frac{L+\kappa}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right).$$

vs

$$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right).$$

Example: apply QuickeNing to MISO

Example: MISO, The inner convergence rate is given

$$\tau_{\mathcal{M}} = \min\left\{\frac{1}{2n}, \frac{\mu+\kappa}{4(L+\kappa)}\right\}$$

which yields the global complexity

$$\tilde{O}\left(\max\left(\frac{\mu+\kappa}{\mu}n, \frac{L+\kappa}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right).$$

vs

$$O\left(\max\left(n, \frac{L}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right).$$

QuickeNing does not provide any theoretical acceleration, but it does not degrade significantly the worst-case performance of $\mathcal{M}$.

Example: apply QuickeNing to MISO

Example: MISO, The inner convergence rate is given

$$\tau_{\mathcal{M}} = \min\left\{\frac{1}{2n}, \frac{\mu+\kappa}{4(L+\kappa)}\right\}$$

which yields the global complexity

$$\tilde{O}\left(\max\left(\frac{\mu+\kappa}{\mu}n, \frac{L+\kappa}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right).$$

vs

Then, how to choose κ?

- Assume that L-BFGS performs as well as Nesterov.
- **Use Catalyst's κ.**

Practical variant

- Each subproblem requires at most $T_{\mathcal{M}}$ iterations of $\mathcal{M}$.

Practical variant

- Each subproblem requires at most $T_{\mathcal{M}}$ iterations of $\mathcal{M}$.

  $\Rightarrow$ Compute $T_{\mathcal{M}}$ in advance, blindly run $T_{\mathcal{M}}$ iterations of $\mathcal{M}$ in each subproblem.

  Benefit: no need to check the stopping condition.

Practical variant

- Each subproblem requires at most $T_{\mathcal{M}}$ iterations of $\mathcal{M}$.

  $\Rightarrow$ Compute $T_{\mathcal{M}}$ in advance, blindly run $T_{\mathcal{M}}$ iterations of $\mathcal{M}$ in each subproblem.
  Benefit: no need to check the stopping condition.

- More aggressive heuristics: $T_{\mathcal{M}} =$ one pass over the data.

Practical variant

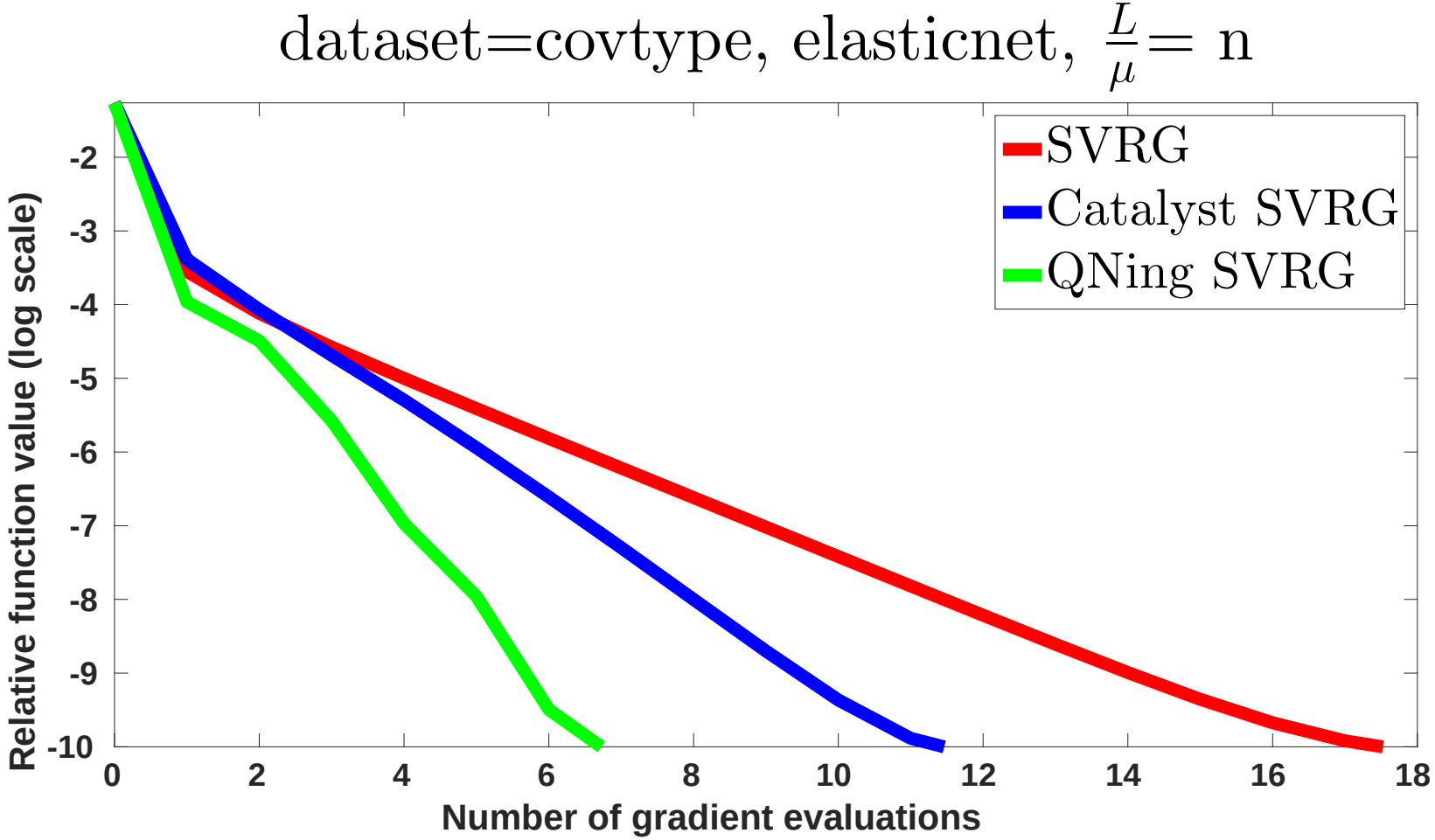
- Each subproblem requires at most $T_{\mathcal{M}}$ iterations of $\mathcal{M}$.

  $\Rightarrow$ Compute $T_{\mathcal{M}}$ in advance, blindly run $T_{\mathcal{M}}$ iterations of $\mathcal{M}$ in each subproblem.
  Benefit: no need to check the stopping condition.

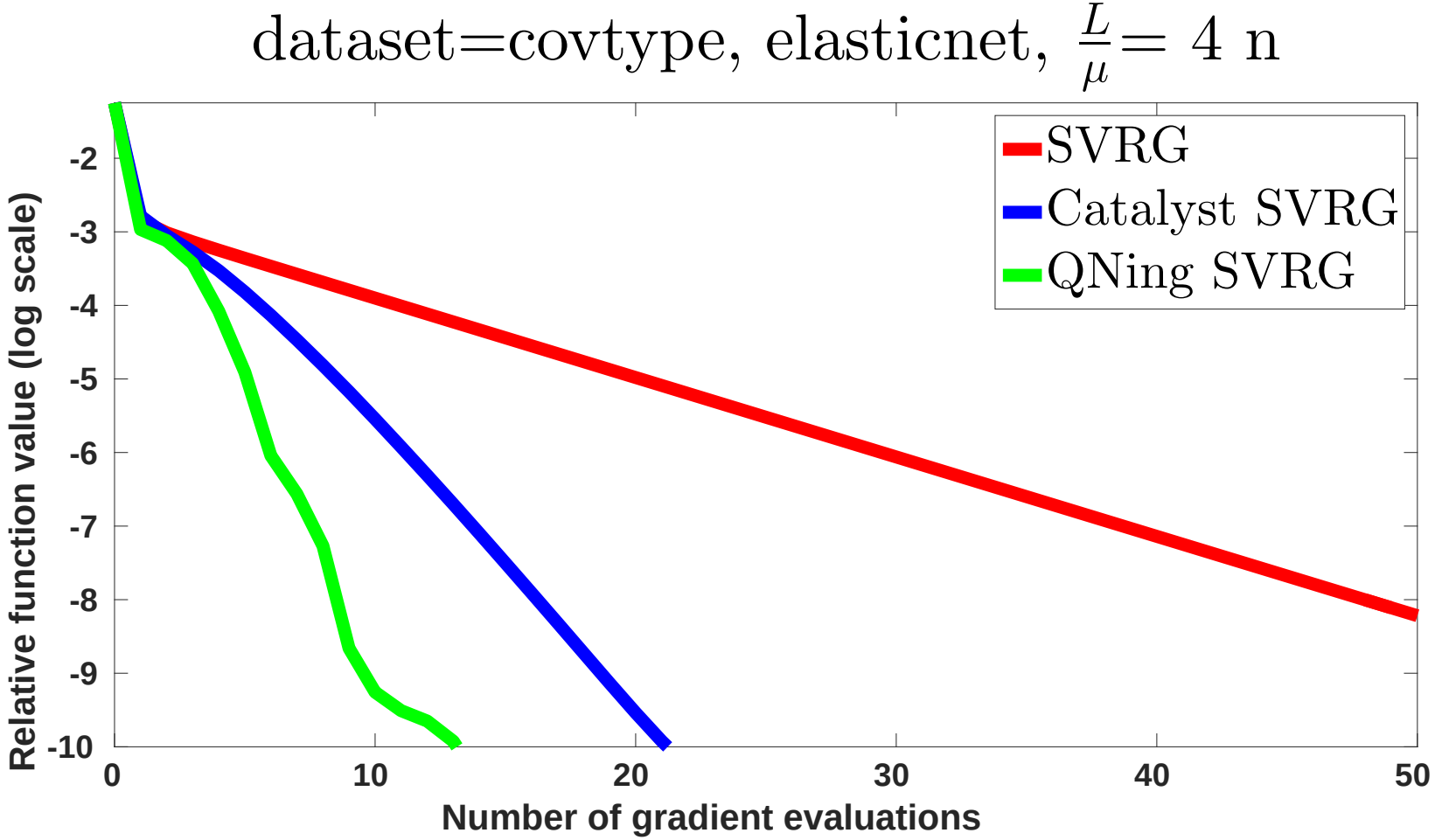
- More aggressive heuristics: $T_{\mathcal{M}} =$ one pass over the data.

In the following experiments, we perform one-pass heuristic for both Catalyst and QuickeNing.

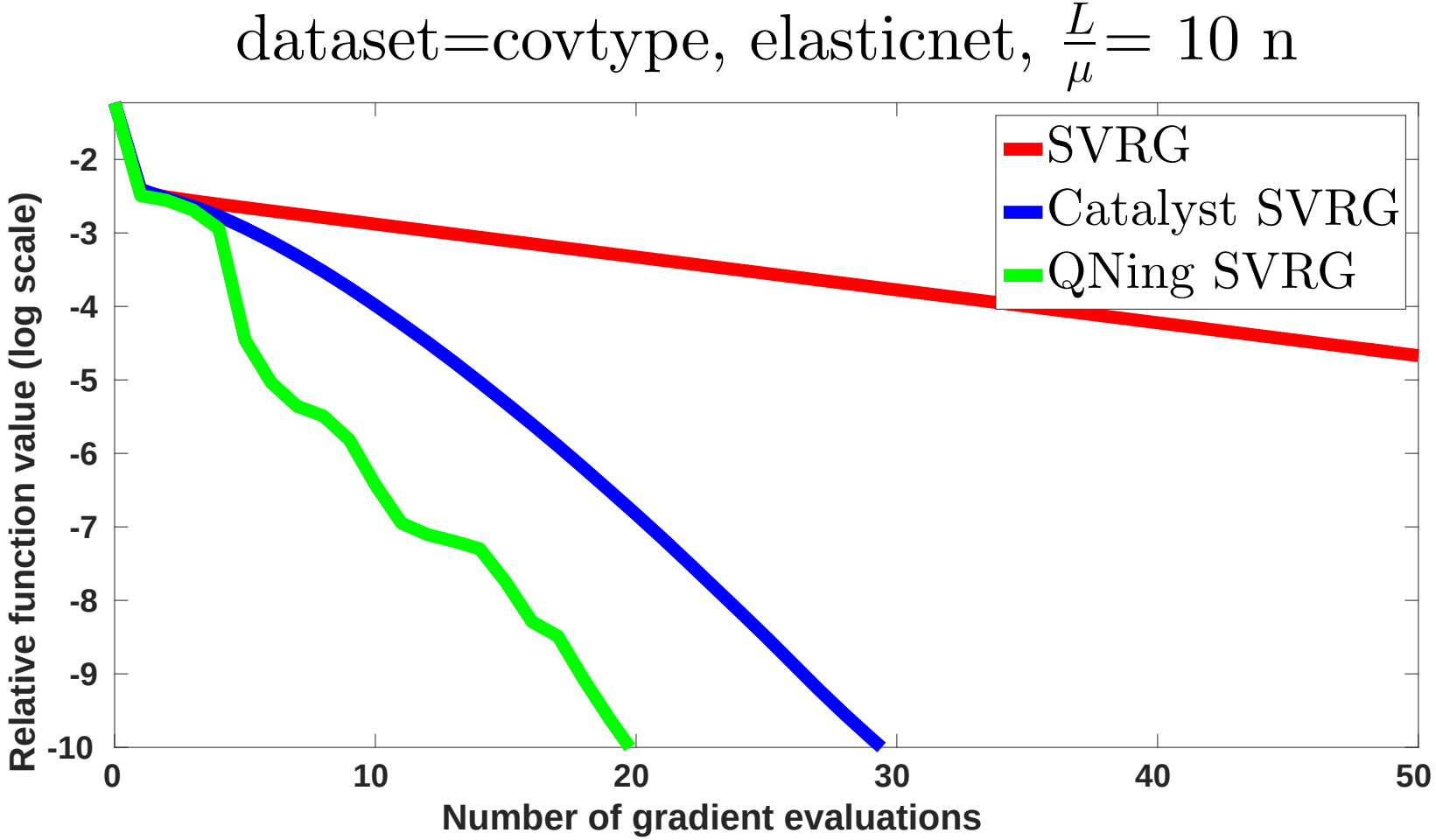
QuickeNing-SVRG: Elasticnet



QuickeNing-SVRG: Elasticnet



QuickeNing-SVRG: Elasticnet



Conclusions

- In 10/12 cases, QuickeNing outperforms Catalyst.
- Big gap between theory and practice.

Extension to non convex problems: 4WD-Catalyst

(Joint work with Courtney Paquette and Dmitriy Drusvyatskiy from UW.)

Extension to non convex problems: 4WD-Catalyst

(Joint work with Courtney Paquette and Dmitriy Drusvyatskiy from UW.)

- Assumption: Weakly convex function, i.e. $f_i(x) + \frac{\rho}{2}\|x\|^2$ is convex.

Extension to non convex problems: 4WD-Catalyst

(Joint work with Courtney Paquette and Dmitriy Drusvyatskiy from UW.)

- Assumption: Weakly convex function, i.e. $f_i(x) + \frac{\rho}{2}\|x\|^2$ is convex.
- Goal: design an algorithm which does not need to know in advance whether the objective function is convex.

Extension to non convex problems: 4WD-Catalyst

(Joint work with Courtney Paquette and Dmitriy Drusvyatskiy from UW.)

- Assumption: Weakly convex function, i.e. $f_i(x) + \frac{\rho}{2}\|x\|^2$ is convex.
- Goal: design an algorithm which does not need to know in advance whether the objective function is convex.

Extension to non convex problems: 4WD-Catalyst

(Joint work with Courtney Paquette and Dmitriy Drusvyatskiy from UW.)

- Assumption: Weakly convex function, i.e. $f_i(x) + \frac{\rho}{2}\|x\|^2$ is convex.
- Goal: design an algorithm which does not need to know in advance whether the objective function is convex.

Main idea

- If κ is large enough, the subproblems are strongly convex.

Extension to non convex problems: 4WD-Catalyst

(Joint work with Courtney Paquette and Dmitriy Drusvyatskiy from UW.)

- Assumption: Weakly convex function, i.e. $f_i(x) + \frac{\rho}{2}\|x\|^2$ is convex.
- Goal: design an algorithm which does not need to know in advance whether the objective function is convex.

Main idea

- If κ is large enough, the subproblems are strongly convex.
- If the subproblems are strongly convex, constant iterations of $\mathcal{M}$ is enough to achieve the desired accuracy.

Extension to non convex problems: 4WD-Catalyst

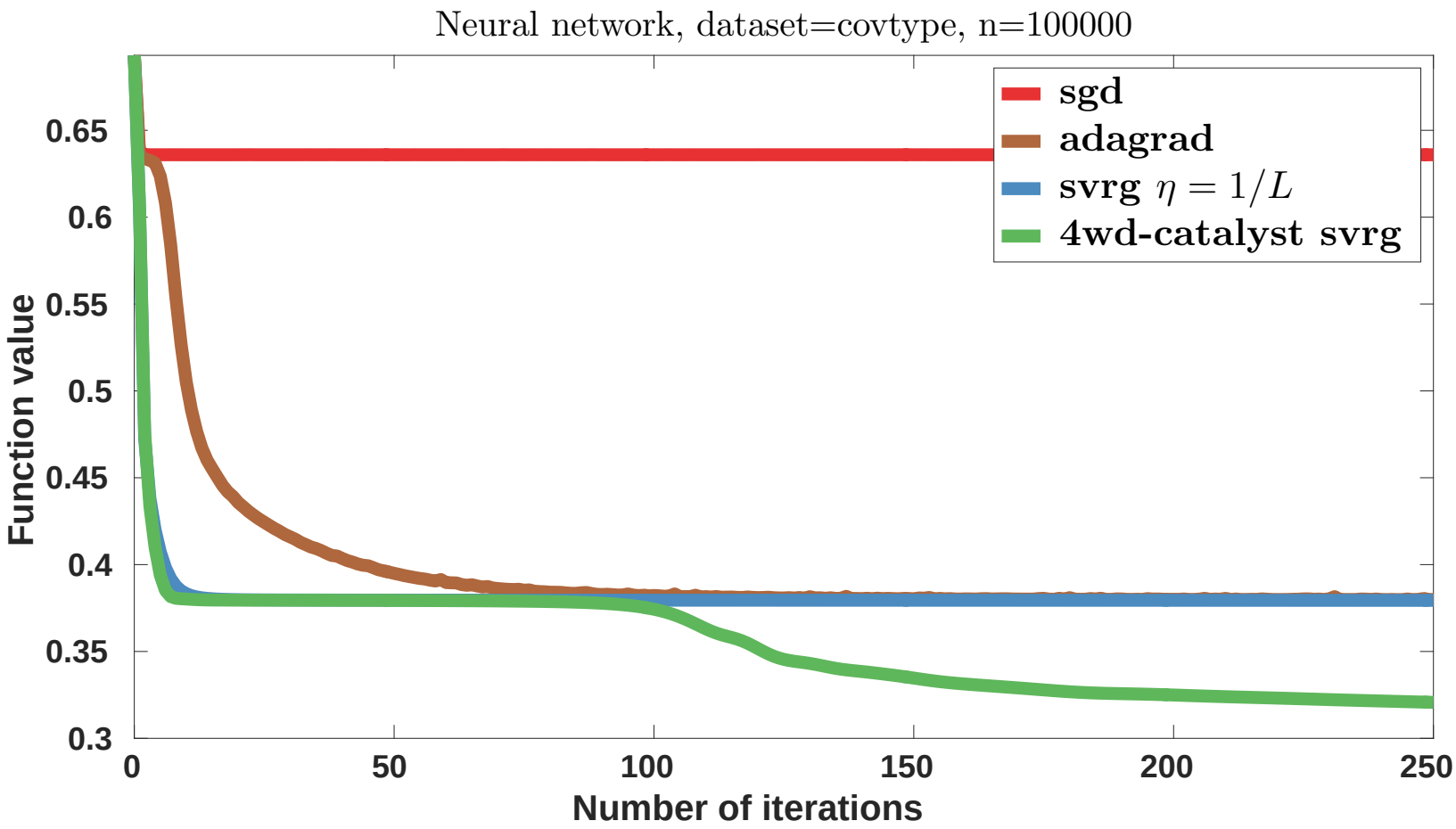
(Joint work with Courtney Paquette and Dmitriy Drusvyatskiy from UW.)

- Assumption: Weakly convex function, i.e. $f_i(x) + \frac{\rho}{2}\|x\|^2$ is convex.
- Goal: design an algorithm which does not need to know in advance whether the objective function is convex.

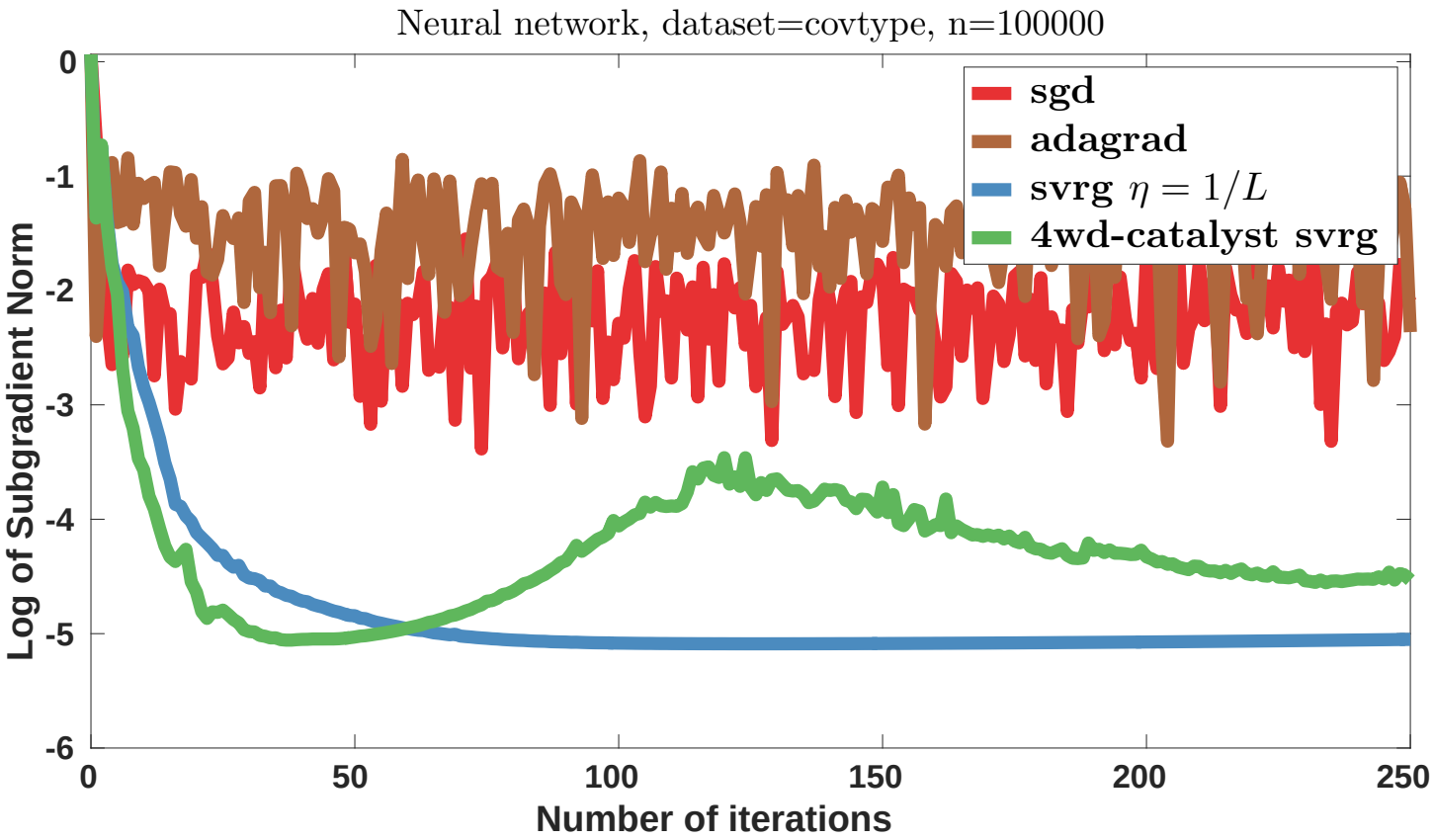
Main idea

- If κ is large enough, the subproblems are strongly convex.
- If the subproblems are strongly convex, constant iterations of $\mathcal{M}$ is enough to achieve the desired accuracy.
- Line search on κ until the subproblems are solved correctly.

4WD-Catalyst: Two-layer neural network



4WD-Catalyst: Two-layer neural network



Conclusions

- Acceleration in terms of function values.
- It seems like 4WD-Catalyst is helpful to escape bad stationary points.

Conclusions and perspectives

Conclusions

- Develop theoretical grounded generic acceleration schemes.
- Significant acceleration in practice for ill-conditioned problems.
- Extension to non convex problems.

Conclusions

- Develop theoretical grounded generic acceleration schemes.
- Significant acceleration in practice for ill-conditioned problems.
- Extension to non convex problems.

Publications and preprint

- H. Lin, J. Mairal and Z. Harchaoui.
  "A Universal Catalyst for First-Order Optimization", *NIPS 2015*.
- H. Lin, J. Mairal and Z. Harchaoui.
  "A Generic Quasi-Newton Algorithm for Faster Gradient-Based Optimization", *submitted to SIAM Optimization in 2017.*
- C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, Z. Harchaoui
  "Catalyst Acceleration for Gradient-Based Non-Convex Optimization," *submitted to AISTATS 2018.*
- H. Lin, J. Mairal and Z. Harchaoui.
  "Catalyst Acceleration for Gradient-Based Optimization: from Theory to Practice", *in preparation*.

Future work and perspective

Inexact proximal point

At iteration k, apply $\mathcal{M}$ to find

$$x_k \approx \arg\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$$

such that $h_k(x_k) - h_k^* \leq \varepsilon_k$.

- Extension to non-Euclidean metrics?

Future work and perspective

Inexact proximal point

At iteration k, apply $\mathcal{M}$ to find

$$x_k \approx \arg\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$$

such that $h_k(x_k) - h_k^* \leq \varepsilon_k$.

- Extension to non-Euclidean metrics?
- Develop parameter free acceleration schemes.

Future work and perspective

Inexact proximal point

At iteration k, apply $\mathcal{M}$ to find

$$x_k \approx \arg\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$$

such that $h_k(x_k) - h_k^* \leq \varepsilon_k$.

- Extension to non-Euclidean metrics?
- Develop parameter free acceleration schemes.
- Is the smoothing helpful to escape saddle points?

Future work and perspective

Inexact proximal point

At iteration k, apply $\mathcal{M}$ to find

$$x_k \approx \arg\min_x \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$$

such that $h_k(x_k) - h_k^* \leq \varepsilon_k$.

- Extension to non-Euclidean metrics?
- Develop parameter free acceleration schemes.
- Is the smoothing helpful to escape saddle points?
- Quasi-Newton methods for non convex problems? Gap between theory and practice.

Thank you for your attention!

Adaptive stopping criterion

Algorithm 1 Procedure ApproxGradient(x, κ)

1: Run $\mathcal{M}$ to find:

$$z_{\mathcal{M}} \approx \arg\min_{z \in \mathbb{R}^d} \left\{ h(z) \triangleq f(z) + \frac{\kappa}{2}\|z - x\|^2 \right\},$$

until

$$\boldsymbol{h(z_{\mathcal{M}}) - h^* \leq \frac{\kappa}{36}\|z_{\mathcal{M}} - x\|^2.}$$

2: Evaluate

$$g \triangleq \kappa(x - z_{\mathcal{M}}), \qquad \text{approximate gradient;}$$
$$F_a \triangleq h(z_{\mathcal{M}}), \qquad \text{approximate function value.}$$

output $(z_{\mathcal{M}}, g, F_a)$

Adaptive stopping criterion

Algorithm 2 Procedure ApproxGradient(x, κ)

1: Run $\mathcal{M}$ to find:

$$z_{\mathcal{M}} \approx \arg\min_{z \in \mathbb{R}^d} \left\{ h(z) \triangleq f(z) + \frac{\kappa}{2}\|z - x\|^2 \right\},$$

until

$$\boldsymbol{h(z_{\mathcal{M}}) - h^* \leq \frac{\kappa}{36}\|z_{\mathcal{M}} - x\|^2.}$$

- No need to predefine the accuracy.
- The criterion is adaptive, both sides depend on $z_{\mathcal{M}}$.
- It is feasible since

$$h(z_{\mathcal{M}}) - h^* \to 0 \quad \text{while} \quad \kappa\|z_{\mathcal{M}} - x\| \to \|\nabla F(x)\|.$$

Warm start of subproblems

Inexact proximal point

At iteration k, warm start the subproblem

$$\min_z \left\{ h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$$

When $\psi = 0$.

- When using predefined sequence, warm start at

$$z_0 = x_{k-1} + \frac{\kappa}{\kappa + \mu}(y_{k-1} - y_{k-2}).$$

- When using adaptive stopping, warm start at

$$z_0 = y_{k-1}.$$

When ψ is non smooth, an additional proximal gradient step is required.

4WD-Catalyst

Let us set

$$f_\kappa(z, x) = f(z) + \frac{\kappa}{2}\|z - x\|^2.$$

Algorithm 3 Auto-adapt (x, κ, T)

---

input $x \in \mathbb{R}^p$, method $\mathcal{M}$, $\kappa > 0$, number of iterations T.

Repeat Run T iterations of $\mathcal{M}$ to obtain

$$z_T \approx \arg\min_{z \in \mathbb{R}^p} f_\kappa(z; x).$$

If $f_\kappa(z_T; x) \leq f_\kappa(x; x)$ and $\mathrm{dist}(0, \partial f_\kappa(z_T; x)) \leq \kappa\|z_T - x\|$,

then go to output.

else repeat with $\kappa \to 2\kappa$.

output (z_T, κ).

Algorithm 4 4WD-Catalyst

input $x_0 \in \operatorname{dom} f$, $\kappa_0, \kappa_{\text{cvx}} > 0$ and $T, S > 0$, and $\mathcal{M}$.

initialization: $\alpha_1 = 1$, $v_0 = x_0$.

repeat for $k = 1, 2, \ldots$

- Compute $(\bar{x}_k, \kappa_k) = \text{Auto-adapt } (x_{k-1}, \kappa_{k-1}, T)$.
- Compute $y_k = \alpha_k v_{k-1} + (1 - \alpha_k) x_{k-1}$ and apply $S \log(k+1)$ iterations of $\mathcal{M}$ to find

$$\tilde{x}_k \approx \arg\min_{x \in \mathbb{R}^p} f_{\kappa_{\text{cvx}}}(x, y_k).$$

- Update v_k and α_{k+1} by

$$v_k = x_{k-1} + \frac{1}{\alpha_k}(\tilde{x}_k - x_{k-1}) \quad \text{and} \quad \alpha_{k+1} = \frac{\sqrt{\alpha_k^4 + 4\alpha_k^2} - \alpha_k^2}{2}.$$

- Choose x_k to be any point satisfying $f(x_k) = \min\{f(\bar{x}_k), f(\tilde{x}_k)\}$.

until the stopping criterion $\operatorname{dist}\big(0, \partial f(\bar{x}_k)\big) < \varepsilon$

First order oracle and lower bound

Definition

We call an algorithm $\mathcal{M}$ an iterative first-order method if it generates a sequence of iterates $(x_k)_{k\geq 0}$ such that

$$x_k \in x_0 + \text{Span}\left\{\nabla f(x_0), \cdots, \nabla f(x_{k-1})\right\}, \quad \text{for } k \geq 1.$$

Theorem (Lower bounds for convex functions)

Given the dimension d, for any k with $1 \leq k \leq \frac{1}{2}(d-1)$, and any x_0 in $\mathbb{R}^d$, there exists a convex L-smooth function f such that for any first-order method $\mathcal{M}$,

$$f(x_k) - f^* \geq \frac{3L\|x_0 - x^*\|^2}{32(k+1)^2},$$

$$\|x_k - x^*\|^2 \geq \frac{1}{8}\|x_0 - x^*\|^2.$$

[Nemirovskii et al., 1983, Nesterov, 2004]

Proximal Newton methods

$$\min_{x \in \mathbb{R}^d} \{f(x) = f_0(x) + \psi(x)\},$$

Proximal Newton methods

- Approximate the Hessian B_k of the smooth part

$$x_{k+1} = \arg\min \left\{ f_0(x_k) + \langle \nabla f_0(x_k), x - x_k \rangle + \frac{1}{2}\|x - x_k\|_{B_k}^2 + \psi(x) \right\}.$$

- **NO** closed form solution of such subproblem.
- In practice: one pass of coordinate descent method.

[Yu et al., 2008, Lee et al., 2012, Byrd et al., 2015, Ghadimi et al., 2015, Scheinberg and Tang, 2016]

Hessian free: Quasi-Newton methods

- Intuition: a quadratic function $f(x) = x^T Bx$ satisfies the **secant equation**

$$\nabla f(z) - \nabla f(x) = B(z - x).$$

- The objective is locally quadratic. Construct B_k such that

$$\nabla f(x_k) - \nabla f(x_{k-1}) = B_k(x_k - x_{k-1})$$

$$\text{set} \quad y_{k-1} \triangleq \nabla f(x_k) - \nabla f(x_{k-1}), \quad s_{k-1} \triangleq x_k - x_{k-1}.$$

- Update $x_{k+1} = x_k - \eta_k B_k^{-1} \nabla f(x_k)$.
- Such B_k is not unique, the Broyden-Fletcher-Goldfarb-Shanno **(BFGS)** method chooses $B_{k+1} = \arg\min_B \|B - B_k\|$, which is given by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k} + \frac{y_k y_k^\top}{y_k^\top s_k}.$$

Hessian free: Quasi-Newton methods

- Intuition: a quadratic function $f(x) = x^T Bx$ satisfies the **secant equation**

$$\nabla f(z) - \nabla f(x) = B(z - x).$$

- The objective is locally quadratic. Construct B_k such that

$$\nabla f(x_k) - \nabla f(x_{k-1}) = B_k(x_k - x_{k-1})$$

$$\text{set} \quad y_{k-1} \triangleq \nabla f(x_k) - \nabla f(x_{k-1}), \quad s_{k-1} \triangleq x_k - x_{k-1}.$$

Quasi-Newton method

- Enjoys **superlinear convergence** rate if the stepsize η_k is chosen by a line search with Wolfe conditions.
- **No need of matrix inversion**, B_{k+1}^{-1} can be obtained in closed form using B_k^{-1}.
- Disadvantage: still need to store a dense $d \times d$ matrix B_k in memory.

Limited memory methods: L-BFGS

$$B_{k+1} = B_k - \frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k} + \frac{y_k y_k^\top}{y_k^\top s_k}.$$

- Observation: B_k can be uniquely determined by B_0 and **all the past** (s_i, y_i), for $i = 1, .., k$.
- Keep the most recent l vectors (s_i, y_i), for $i = k - l, .., k$.
- Compute $B_k^{-1} \nabla f(x_k)$ by sequentially compute vector products with vectors in memory, named as two-loop recursion step [Nocedal, 1980].
- The step size η_k is determined by a line search.

Limited memory methods: L-BFGS

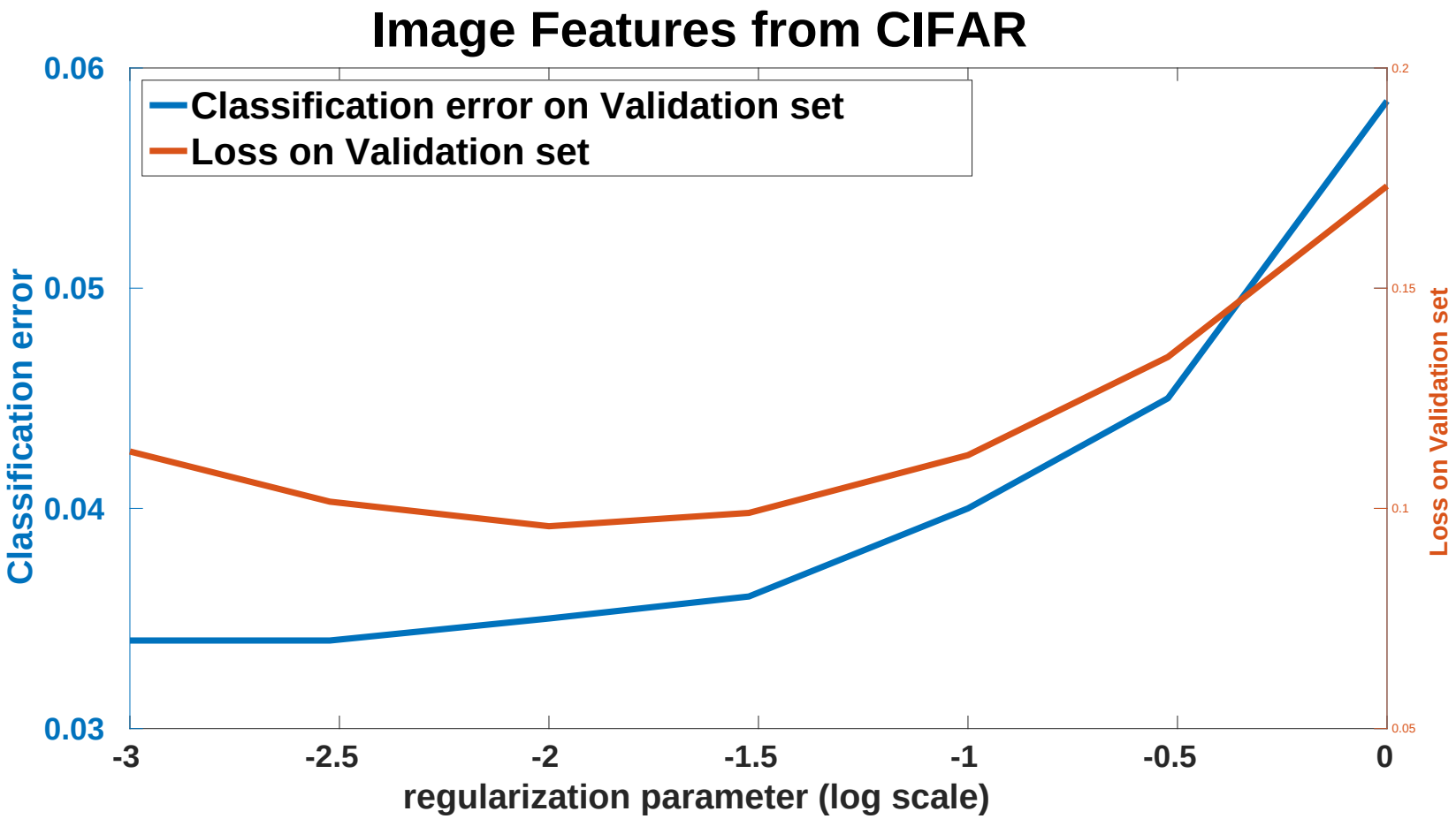
$$B_{k+1} = B_k - \frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k} + \frac{y_k y_k^\top}{y_k^\top s_k}.$$

- Observation: B_k can be uniquely determined by B_0 and **all the past** (s_i, y_i), for $i = 1, .., k$.
- Keep the most recent l vectors (s_i, y_i), for $i = k - l, .., k$.
- Compute $B_k^{-1}\nabla f(x_k)$ by sequentially compute vector products with vectors in memory, named as two-loop recursion step [Nocedal, 1980].

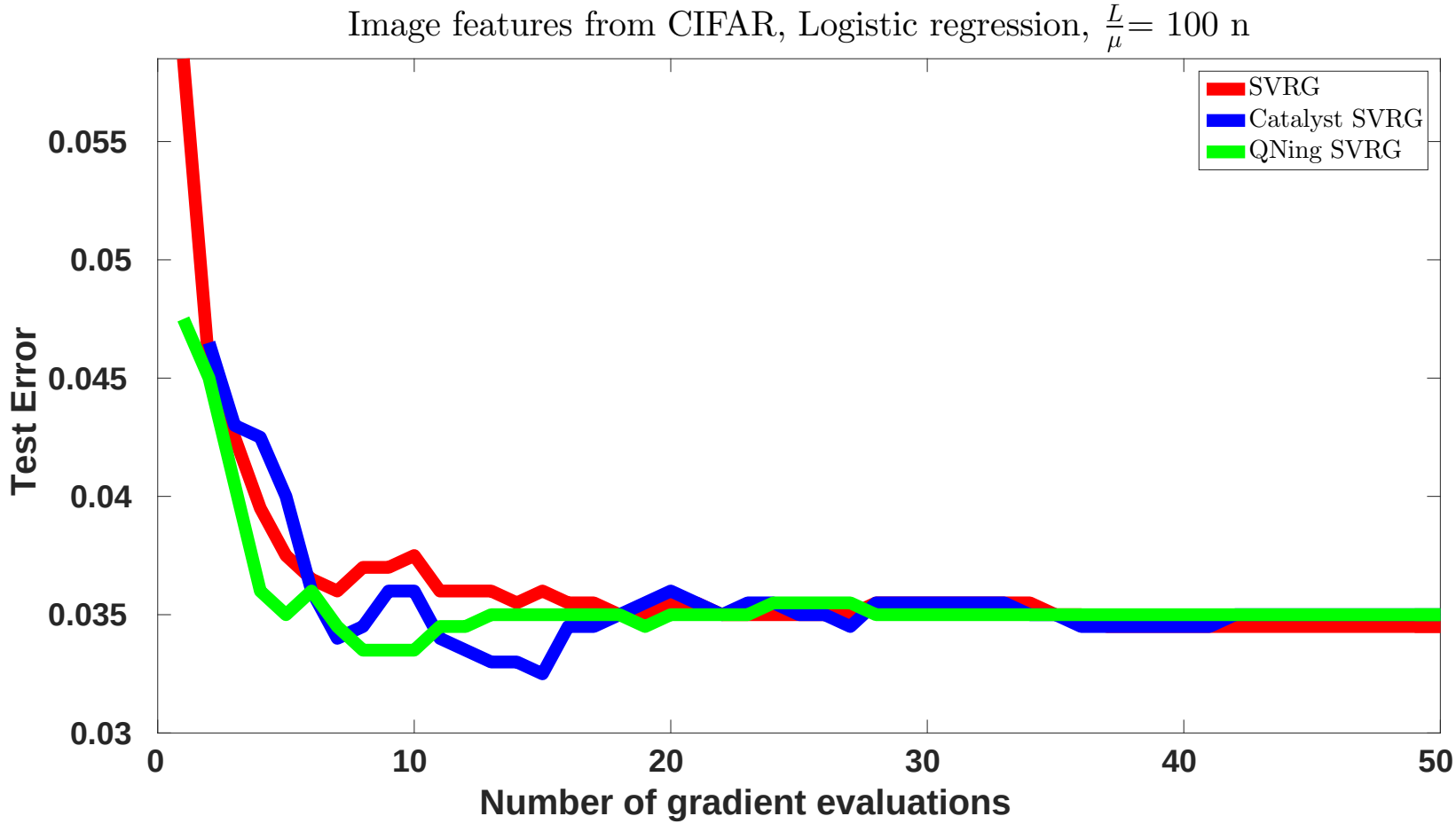
It is shown that L-BFGS enjoys

- **a big practical success of smooth optimization**.
- linear convergence in worst case scenario, **no better than the gradient descent method**.

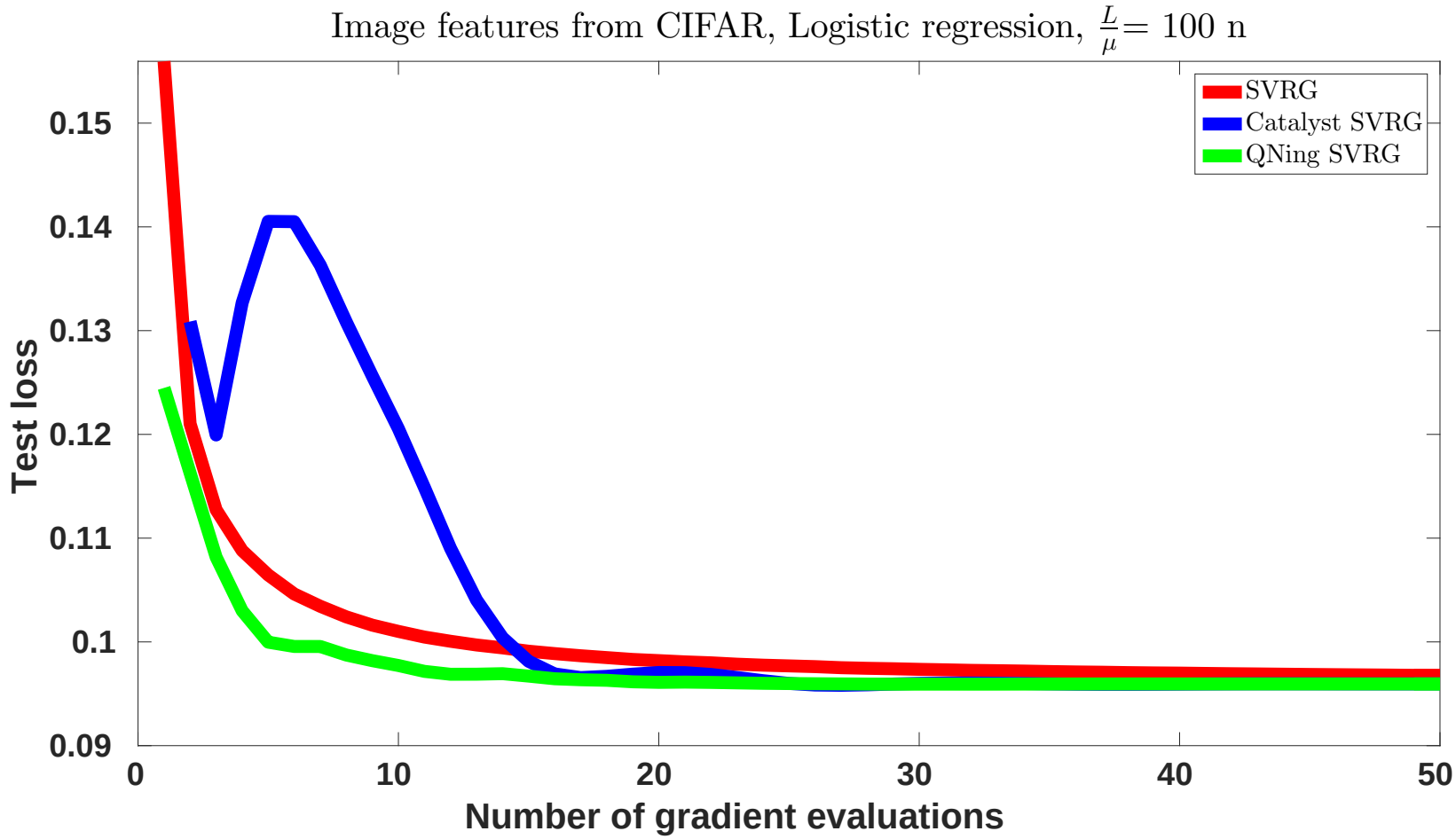
Cross Validation and Testing



Cross Validation and Testing



Cross Validation and Testing



References I

Alekh Agarwal, Martin J Wainwright, Peter L Bartlett, and Pradeep K Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.

Y. Arjevani and O. Shamir. Dimension-free iteration complexity of finite sum optimization problems. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6 (1):76–90, 1970.

References II

James V Burke and Maijian Qian. On the superlinear convergence of the variable metric proximal point algorithm using broyden and bfgs matrix secant updating. *Mathematical Programming*, 88(1):157–181, 2000.

R. H. Byrd, J. Nocedal, and F. Oztoprak. An inexact successive quadratic approximation method for L-1 regularized optimization. *Mathematical Programming*, 157(2):375–396, 2015.

Xiaojun Chen and Masao Fukushima. Proximal quasi-newton methods for nondifferentiable convex optimization. *Mathematical Programming*, 85(2): 313–334, 1999.

E. De Klerk, F. Glineur, and A. B. Taylor. On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions. *Optimization Letters*, 11(7):1185–1199, 2017.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014a.

References III

A. J. Defazio, T. S. Caetano, and J. Domke. Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014b.

O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146 (1-2):37–75, 2014.

R. Fletcher. A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322, 1970.

Michael P Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3): A1380–A1405, 2012.

Marc Fuentes, Jérôme Malick, and Claude Lemaréchal. Descentwise inexact proximal algorithms for smooth optimization. *Computational Optimization and Applications*, 53(3):755–769, 2012.

Masao Fukushima and Liqun Qi. A globally and superlinearly convergent algorithm for nonsmooth convex minimization. *SIAM Journal on Optimization*, 6(4):1106–1120, 1996.

References IV

S. Ghadimi, G. Lan, and H. Zhang. Generalized Uniformly Optimal Methods for Nonlinear Programming. *arxiv:1508.07384*, 2015.

D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.

O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

B. He and X. Yuan. An accelerated inexact proximal point algorithm for convex minimization. *Journal of Optimization Theory and Applications*, 154(2): 536–548, 2012.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Yann LeCun and Leon Bottou. Large scale online learning. *Advances in neural information processing systems*, 16:217, 2004.

Jason Lee, Yuekai Sun, and Michael Saunders. Proximal Newton-type methods for convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

References V

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2): 829–855, 2015.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

A. Nedić and D. Bertsekas. Convergence rate of incremental subgradient algorithms. In *Stochastic optimization: algorithms and applications*, pages 223–264. Springer, 2001.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

A. Nemirovskii, D. B. Yudin, and E. R. Dawson. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Publishers, 2004.

Y. Nesterov. Gradient methods for minimizing composite objective function. *Mathematical Programming*, 140(1):125–161, 2007.

References VI

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/k2). In *Doklady an SSSR*, volume 269, pages 543–547, 1983.

Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

Hugo Raguet, Jalal Fadili, and Gabriel Peyré. A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 6(3):1199–1226, 2013.

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

Katya Scheinberg and Xiaocheng Tang. Practical inexact proximal quasi-Newton method with global complexity analysis. *Mathematical Programming*, 160(1):495–529, 2016.

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2013.

References VII

S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *arXiv:1211.2717*, 2012.

D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

A. B. Taylor, J. M. Hendrickx, and F. Glineur. Exact worst-case convergence rates of the proximal gradient method for composite convex minimization. *arXiv:1705.04398*, 2017.

B. E. Woodworth and N. Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

S.J. Wright, R.D. Nowak, and M.A.T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7): 2479–2493, 2009.

K. Yosida. *Functional Analysis*. Classics in mathematics. World Publishing Company, 1980. ISBN 9780387102108. URL `https://books.google.com/books?id=1zewQgAACAAJ`.

References VIII

Jin Yu, SVN Vishwanathan, Simon Günter, and Nicol N Schraudolph. A quasi-Newton approach to non-smooth convex optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.