A Superlinearly-Convergent Proximal Newton-Type Method for the Optimization of Finite Sums

Anton Rodomanov[1,2] Dmitry Kropotov[2,3]

[1]Higher School of Economics

p(B|A)yesgroup.ru

[2]Bayesian Methods Research Group

[3]Lomonosov Moscow State University

Moscow, Russia

22 June 2016

International Conference on Machine Learning (ICML-2016), New York, USA

Introduction

Consider the minimization of the composite finite average:

$$\min_{x \in \mathbb{R}^d} \left[\phi(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x) \right]$$

Introduction

Consider the minimization of the composite finite average:

$$\min_{x \in \mathbb{R}^d} \left[\phi(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x) \right]$$

Assumptions:

- each f_i is twice-continuously differentiable and convex
- h is a general convex function (but simple)
- ϕ is strongly convex

Introduction

Consider the minimization of the composite finite average:

$$\min_{x \in \mathbb{R}^d} \left[\phi(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x) \right]$$

Assumptions:

- each f_i is twice-continuously differentiable and convex
- h is a general convex function (but simple)
- ϕ is strongly convex

Examples: linear regression, logistic regression, CRF etc.

Introduction

Consider the minimization of the composite finite average:

$$\min_{x \in \mathbb{R}^d} \left[\phi(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x) \right]$$

Assumptions:

- each f_i is twice-continuously differentiable and convex
- h is a general convex function (but simple)
- ϕ is strongly convex

Examples: linear regression, logistic regression, CRF etc.

- n is very large

Motivation

We are interested in **incremental methods** [Bertsekas, 2011] whose iteration cost is independent of n:

Motivation

We are interested in **incremental methods** [Bertsekas, 2011] whose iteration cost is independent of n:

- Stochastic methods for $\min_x \{\mathbb{E}_z[f(x;z)]\} = \min_x \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$:
  - **Examples:** SGD [Robbins-Monro, 1951], oLBFGS [Schraudolph et al., 2007], AdaGrad [Duchi et al., 2011], SQN [Byrd et al., 2014], Adam [Kingma, 2014] etc.
  - **Iteration:** $x_{k+1} = x_k - \alpha_k B_k \nabla f_{i_k}(x_k)$.
  - **Convergence rate:** sublinear, usually $\mathcal{O}(1/k)$.

Motivation

We are interested in **incremental methods** [Bertsekas, 2011] whose iteration cost is independent of n:

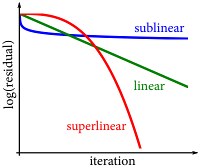
- Stochastic methods for $\min_x \{\mathbb{E}_z[f(x;z)]\} = \min_x \left\{\frac{1}{n}\sum_{i=1}^{n} f_i(x)\right\}$:
  - **Examples:** SGD [Robbins-Monro, 1951], oLBFGS [Schraudolph et al., 2007], AdaGrad [Duchi et al., 2011], SQN [Byrd et al., 2014], Adam [Kingma, 2014] etc.
  - **Iteration:** $x_{k+1} = x_k - \alpha_k B_k \nabla f_{i_k}(x_k)$.
  - **Convergence rate:** sublinear, usually $\mathcal{O}(1/k)$.
- Methods for $\min_x \left\{\frac{1}{n}\sum_{i=1}^{n} f_i(x)\right\}$:
  - **Examples:** SAG [Le Roux et al., 2012], SVRG [Johnson & Zhang, 2013], FINITO [Defazio et al., 2014b], SAGA [Defazio et al., 2014a], MISO [Mairal, 2015] etc.
  - **Main idea:** variance reduction.
  - **Convergence rate:** linear, $\mathcal{O}(c^k)$.

Motivation

We are interested in **incremental methods** [Bertsekas, 2011] whose iteration cost is independent of n:

- Stochastic methods for $\min_x \{\mathbb{E}_z[f(x;z)]\} = \min_x \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$:
  - **Examples:** SGD [Robbins-Monro, 1951], oLBFGS [Schraudolph et al., 2007], AdaGrad [Duchi et al., 2011], SQN [Byrd et al., 2014], Adam [Kingma, 2014] etc.
  - **Iteration:** $x_{k+1} = x_k - \alpha_k B_k \nabla f_{i_k}(x_k)$.
  - **Convergence rate:** sublinear, usually $\mathcal{O}(1/k)$.
- Methods for $\min_x \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$:
  - **Examples:** SAG [Le Roux et al., 2012], SVRG [Johnson & Zhang, 2013], FINITO [Defazio et al., 2014b], SAGA [Defazio et al., 2014a], MISO [Mairal, 2015] etc.
  - **Main idea:** variance reduction.
  - **Convergence rate:** linear, $\mathcal{O}(c^k)$.

Goal: an incremental method with a superlinear convergence rate.



Main contributions

Our main contributions:

- New method: Newton-type Incremental Method (NIM)
- Theorem establishing superlinear convergence of NIM

NIM: Idea

Problem: $\min_x \left[\phi(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x) \right].$

NIM: Idea

Problem: $\min_x \left[\phi(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) + h(x)\right].$

- Build the second-order Taylor approximation of each f_i:
  $$f_i(x) \approx m_k^i(x) := f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i).$$

NIM: Idea

Problem: $\min_x \left[\phi(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x) \right]$.

- Build the second-order Taylor approximation of each f_i:
  $$f_i(x) \approx m_k^i(x) := f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i).$$
- Then $\phi(x) \approx m_k(x) := \frac{1}{n} \sum_{i=1}^{n} m_k^i(x) + h(x)$.

NIM: Idea

Problem: $\min_x \left[\phi(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) + h(x)\right].$

- Build the second-order Taylor approximation of each f_i:
$$f_i(x) \approx m_k^i(x) := f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i).$$
- Then $\phi(x) \approx m_k(x) := \frac{1}{n}\sum_{i=1}^{n} m_k^i(x) + h(x)$.
- Find the minimizer of the model $\bar{x}_k := \operatorname{argmin}_x m_k(x)$.
- Choose next iterate: $x_{k+1} = x_k + \alpha(\bar{x}_k - x_k)$.

NIM: Idea

Problem: $\min_x \left[\phi(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x) \right].$

- Build the second-order Taylor approximation of each f_i:
$$f_i(x) \approx m_k^i(x) := f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i).$$
- Then $\phi(x) \approx m_k(x) := \frac{1}{n} \sum_{i=1}^{n} m_k^i(x) + h(x)$.
- Find the minimizer of the model $\bar{x}_k := \operatorname{argmin}_x m_k(x)$.
- Choose next iterate: $x_{k+1} = x_k + \alpha(\bar{x}_k - x_k)$.
- **(Standard Newton method)** $v_k^i = x_k$ for all $i = 1, \ldots, n$.

NIM: Idea

Problem: $\min_x \left[\phi(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) + h(x)\right].$

- Build the second-order Taylor approximation of each f_i:
  $$f_i(x) \approx m_k^i(x) := f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i).$$
- Then $\phi(x) \approx m_k(x) := \frac{1}{n}\sum_{i=1}^{n} m_k^i(x) + h(x)$.
- Find the minimizer of the model $\bar{x}_k := \operatorname{argmin}_x m_k(x)$.
- Choose next iterate: $x_{k+1} = x_k + \alpha(\bar{x}_k - x_k)$.
- **(Standard Newton method)** $v_k^i = x_k$ for all $i = 1, \dots, n$.
- **(NIM)** Update only one v_k^i: choose $i_k \in \{1, \dots, n\}$ and set
  $$v_{k+1}^i := \begin{cases} x_{k+1} & \text{if } i = i_k, \\ v_k^i & \text{otherwise.} \end{cases}$$

  Iteration cost is independent of n.

NIM: Model update

Recall:

$$m_k^i(x) = f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i)$$

$$m_k(x) = \frac{1}{n}\sum_{i=1}^{n} m_k^i(x) + h(x)$$

NIM: Model update

$$m_k(x) = \frac{1}{n}\sum_{i=1}^{n}\left[f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i)\right] + h(x).$$

NIM: Model update

$$m_k(x) = \frac{1}{n}\sum_{i=1}^{n}\left[f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i)\right] + h(x).$$

Note: m_k is a (composite) quadratic,

$$m_k(x) = (g_k - u_k)^\top x + \frac{1}{2}x^\top H_k x + h(x) + \mathrm{const},$$

and determined only by the following three quantities:

$$H_k := \frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(v_k^i), \quad u_k := \frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(v_k^i)v_k^i, \quad g_k := \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(v_k^i).$$

NIM: Model update

$$m_k(x) = \frac{1}{n}\sum_{i=1}^{n}\left[f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i)\right] + h(x).$$

Note: m_k is a (composite) quadratic,

$$m_k(x) = (g_k - u_k)^\top x + \frac{1}{2}x^\top H_k x + h(x) + \text{const},$$

and determined only by the following three quantities:

$$H_k := \frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(v_k^i), \quad u_k := \frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(v_k^i)v_k^i, \quad g_k := \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(v_k^i).$$

Since only one v_k^i is updated at every iteration, we have for $i = i_k$

$$H_{k+1} = H_k + \frac{1}{n}\left[\nabla^2 f_i(v_{k+1}^i) - \nabla^2 f_i(v_k^i)\right]$$

$$u_{k+1} = u_k + \frac{1}{n}\left[\nabla^2 f_i(v_{k+1}^i)v_{k+1}^i - \nabla^2 f_i(v_k^i)v_k^i\right]$$

$$g_{k+1} = g_k + \frac{1}{n}\left[\nabla f_i(v_{k+1}^i) - \nabla f_i(v_k^i)\right].$$

NIM: Algorithm

Input: $x_0, \dots, x_{n-1} \in \mathbb{R}^d$: initial points; $\alpha > 0$: step length.

Initialize model: $v^i := x_{i-1}$ for $i = 1, \dots, n$ and

$$H := \frac{1}{n} \sum_{i=1}^{n} \nabla^2 f_i(v^i), \quad u := \frac{1}{n} \sum_{i=1}^{n} \nabla^2 f_i(v^i) v^i, \quad g := \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(v^i)$$

for $k \geq n - 1$ **do**

NIM: Algorithm

Input: $x_0, \ldots, x_{n-1} \in \mathbb{R}^d$: initial points; $\alpha > 0$: step length.

Initialize model: $v^i := x_{i-1}$ for $i = 1, \ldots, n$ and

$$H := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i), \quad u := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i) v^i, \quad g := \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(v^i)$$

for $k \geq n-1$ **do**

Compute minimizer: $\bar{x}_k := \operatorname{argmin}_x \left[(g-u)^\top x + \frac{1}{2} x^\top H x + h(x)\right]$

Make a step: $x_{k+1} := x_k + \alpha(\bar{x}_k - x_k)$

NIM: Algorithm

Input: $x_0, \ldots, x_{n-1} \in \mathbb{R}^d$: initial points; $\alpha > 0$: step length.

Initialize model: $v^i := x_{i-1}$ for $i = 1, \ldots, n$ and

$$H := \frac{1}{n} \sum_{i=1}^{n} \nabla^2 f_i(v^i), \quad u := \frac{1}{n} \sum_{i=1}^{n} \nabla^2 f_i(v^i) v^i, \quad g := \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(v^i)$$

for $k \geq n - 1$ **do**

Compute minimizer: $\bar{x}_k := \operatorname{argmin}_x \left[(g - u)^\top x + \frac{1}{2} x^\top H x + h(x) \right]$

Make a step: $x_{k+1} := x_k + \alpha(\bar{x}_k - x_k)$

Update model for $i := (k+1) \bmod n + 1$ (cyclic order):

$$H := H + \frac{1}{n} \left[\nabla^2 f_i(x_{k+1}) - \nabla^2 f_i(v^i) \right]$$
$$u := u + \frac{1}{n} \left[\nabla^2 f_i(x_{k+1}) x_{k+1} - \nabla^2 f_i(v^i) v^i \right]$$
$$g := g + \frac{1}{n} \left[\nabla f_i(x_{k+1}) - \nabla f_i(v^i) \right]$$
$$v^i := x_{k+1}$$

end for

NIM: Algorithm

Input: $x_0, \dots, x_{n-1} \in \mathbb{R}^d$: initial points; $\alpha > 0$: step length.

Initialize model: $v^i := x_{i-1}$ for $i = 1, \dots, n$ and

$$H := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i), \quad u := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i) v^i, \quad g := \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(v^i)$$

for $k \geq n-1$ **do**

Compute minimizer: $\bar{x}_k := \operatorname{argmin}_x \left[(g-u)^\top x + \frac{1}{2} x^\top H x + h(x) \right]$

Make a step: $x_{k+1} := x_k + \alpha(\bar{x}_k - x_k)$

Update model for $i := (k+1) \bmod n + 1$ (cyclic order):

$$H := H + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) - \nabla^2 f_i(v^i)\right]$$
$$u := u + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) x_{k+1} - \nabla^2 f_i(v^i) v^i\right]$$
$$g := g + \frac{1}{n}\left[\nabla f_i(x_{k+1}) - \nabla f_i(v^i)\right]$$
$$v^i := x_{k+1}$$

end for

Note: H, u, g and v^i are kept in memory.
Required memory: $\mathcal{O}(d^2 + nd)$.

Convergence rate (local)

Theorem

Suppose $\nabla^2 f_i$ are Lipschitz-continuous with constant M_f. Assume x^* is a minimizer of ϕ with $\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*) \succeq \mu_f I \succ 0$, and all the initial points are close enough to x^*: $\|x_i - x^*\| \leq R$ for $0 \leq i \leq n-1$.

Convergence rate (local)

Theorem

Suppose $\nabla^2 f_i$ *are Lipschitz-continuous with constant* M_f*. Assume* x^* *is a minimizer of* ϕ *with* $\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*) \succeq \mu_f I \succ 0$*, and all the initial points are close enough to* x^**:* $\|x_i - x^*\| \leq R$ *for* $0 \leq i \leq n-1$*.*
Then the sequence of iterates $\{x_k\}$ *of NIM with* $\alpha \equiv 1$ *converges to* x^* *at an R-superlinear rate, i.e. there exist* $\{z_k\}$ *and* $\{q_k\}$ *such that for* $k \geq n$

$$\|x_k - x^*\| \leq z_k, \qquad z_{k+1} \leq q_k z_k, \qquad q_k \to 0,$$

where

$$R := \frac{\mu_f}{2M_f}, \qquad q_k := \left(1 - \frac{3}{4n}\right)^{2^{\lceil k/n \rceil - 1}}.$$

More precisely, the rate of convergence is n-step quadratic:

$$z_{k+n} \leq \frac{M_f}{\mu_f} z_k^2.$$

Convergence rate (global)

Problem: $\min_x \left[\phi(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) + h(x)\right].$

Assume $h(x) := \frac{\mu}{2}\|x\|^2$.

Theorem

Denote the condition number of ϕ as $\kappa_\phi := (L_f + \mu)/\mu$ and the minimizer of ϕ as x^*. Then, for any initial points $x_0, \ldots, x_{n-1}$, NIM with a constant step length $\alpha \equiv \kappa_\phi^{-3}(1 + 19\kappa_\phi(n-1))^{-1}$ converges to x^* at a linear rate:

$$\phi(x_k) - \phi(x^*) \le c^k[\phi(x_0) - \phi(x^*)],$$

where

$$c := (1 - \kappa_\phi^{-4}(1 + 19\kappa_\phi(n-1))^{-1})^{\frac{1}{1+2(n-1)}}.$$

N.B.: This result is qualitative.

NIM: Model minimization?

Input: $x_0, \dots, x_{n-1} \in \mathbb{R}^d$: initial points; $\alpha > 0$: step length.

Initialize model: $v^i := x_{i-1}$ for $i = 1, \dots, n$ and

$$H := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i), \quad u := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i) v^i, \quad g := \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(v^i)$$

for $k \geq n-1$ **do**

Compute minimizer: $\bar{x}_k := \mathrm{argmin}_x \left[(g-u)^\top x + \frac{1}{2} x^\top H x + h(x)\right]$

Make a step: $x_{k+1} := x_k + \alpha(\bar{x}_k - x_k)$

Update model for $i := (k+1) \bmod n + 1$ (cyclic order):

$$H := H + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) - \nabla^2 f_i(v^i)\right]$$
$$u := u + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) x_{k+1} - \nabla^2 f_i(v^i) v^i\right]$$
$$g := g + \frac{1}{n}\left[\nabla f_i(x_{k+1}) - \nabla f_i(v^i)\right]$$
$$v^i := x_{k+1}$$

end for

NIM: Model minimization?

Input: $x_0, \dots, x_{n-1} \in \mathbb{R}^d$: initial points; $\alpha > 0$: step length.

Initialize model: $v^i := x_{i-1}$ for $i = 1, \dots, n$ and

$$H := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i), \quad u := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i) v^i, \quad g := \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(v^i)$$

for $k \geq n - 1$ **do**

Compute minimizer: $\bar{x}_k := \operatorname{argmin}_x \left[(g - u)^\top x + \frac{1}{2} x^\top H x + h(x)\right]$

Make a step: $x_{k+1} := x_k + \alpha(\bar{x}_k - x_k)$

Update model for $i := (k+1) \bmod n + 1$ (cyclic order):

$H := H + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) - \nabla^2 f_i(v^i)\right]$

$u := u + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) x_{k+1} - \nabla^2 f_i(v^i) v^i\right]$

$g := g + \frac{1}{n}\left[\nabla f_i(x_{k+1}) - \nabla f_i(v^i)\right]$

$v^i := x_{k+1}$

end for

- If $h \equiv 0$, then $\bar{x}_k = H^{-1}(u - g)$.

NIM: Model minimization?

Input: $x_0, \dots, x_{n-1} \in \mathbb{R}^d$: initial points; $\alpha > 0$: step length.

Initialize model: $v^i := x_{i-1}$ for $i = 1, \dots, n$ and

$$H := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i), \quad u := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i) v^i, \quad g := \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(v^i)$$

for $k \geq n-1$ **do**

Compute minimizer: $\bar{x}_k := \mathrm{argmin}_x \left[(g-u)^\top x + \frac{1}{2}x^\top H x + h(x)\right]$

Make a step: $x_{k+1} := x_k + \alpha(\bar{x}_k - x_k)$

Update model for $i := (k+1) \bmod n + 1$ (cyclic order):

$$H := H + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) - \nabla^2 f_i(v^i)\right]$$

$$u := u + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) x_{k+1} - \nabla^2 f_i(v^i) v^i\right]$$

$$g := g + \frac{1}{n}\left[\nabla f_i(x_{k+1}) - \nabla f_i(v^i)\right]$$

$$v^i := x_{k+1}$$

end for

- If $h \equiv 0$, then $\bar{x}_k = H^{-1}(u - g)$.
- Otherwise, use an iterative method for finding $\bar{x}_k$.

NIM: Model minimization?

Input: $x_0, \dots, x_{n-1} \in \mathbb{R}^d$: initial points; $\alpha > 0$: step length.

Initialize model: $v^i := x_{i-1}$ for $i = 1, \dots, n$ and

$$H := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i), \quad u := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v^i) v^i, \quad g := \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(v^i)$$

for $k \geq n-1$ **do**

Compute minimizer: $\bar{x}_k := \mathrm{argmin}_x \left[(g-u)^\top x + \frac{1}{2}x^\top H x + h(x)\right]$

Make a step: $x_{k+1} := x_k + \alpha(\bar{x}_k - x_k)$

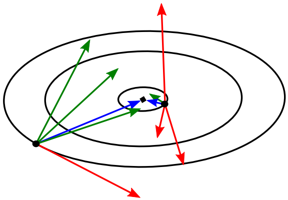
Update model for $i := (k+1) \bmod n + 1$ (cyclic order):

$$H := H + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) - \nabla^2 f_i(v^i)\right]$$
$$u := u + \frac{1}{n}\left[\nabla^2 f_i(x_{k+1}) x_{k+1} - \nabla^2 f_i(v^i) v^i\right]$$
$$g := g + \frac{1}{n}\left[\nabla f_i(x_{k+1}) - \nabla f_i(v^i)\right]$$
$$v^i := x_{k+1}$$

end for

- If $h \equiv 0$, then $\bar{x}_k = H^{-1}(u - g)$.
- Otherwise, use an iterative method for finding $\bar{x}_k$.
- **Idea:** $\bar{x}_k$ may be computed inexactly (as in inexact Newton methods).

NIM: Inexact model minimization



Problem: $\min_x \left[\phi(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right]$.

(Assume $h \equiv 0$ for simplicity.)

Model: $m_k(x) = (g_k - u_k)^\top x + \frac{1}{2}x^\top H_k x + \text{const.}$

NIM iteration: $x_{k+1} = x_k + \alpha(\bar{x}_k - x_k)$, where $\bar{x}_k := \operatorname{argmin} m_k(x)$.

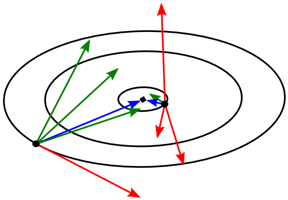
Inexact minimization: instead of $\bar{x}_k$, use $\hat{x}_k$ such that

$$\|\nabla m_k(\hat{x}_k)\| \le \eta_k \|\nabla \phi(x_k)\|, \qquad \eta_k := \left\{0.5, \sqrt{\|\nabla \phi(x_k)\|}\right\}.$$

NIM: Inexact model minimization

Problem: $\min_x \left[\phi(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right]$.

(Assume $h \equiv 0$ for simplicity.)



Model: $m_k(x) = (g_k - u_k)^\top x + \frac{1}{2}x^\top H_k x + \mathrm{const.}$

NIM iteration: $x_{k+1} = x_k + \alpha(\bar{x}_k - x_k)$, where $\bar{x}_k := \operatorname{argmin} m_k(x)$.

Inexact minimization: instead of $\bar{x}_k$, use $\hat{x}_k$ such that

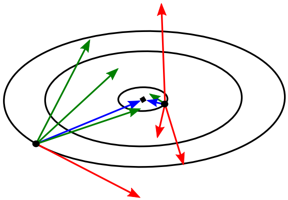
$$\|\nabla m_k(\hat{x}_k)\| \leq \eta_k \|\nabla\phi(x_k)\|, \qquad \eta_k := \left\{0.5, \sqrt{\|\nabla\phi(x_k)\|}\right\}.$$

Problem: cannot compute $\|\nabla\phi(x_k)\|$ (this in incremental optimization!).

NIM: Inexact model minimization

Problem: $\min_x \left[\phi(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right]$.

(Assume $h \equiv 0$ for simplicity.)



Model: $m_k(x) = (g_k - u_k)^\top x + \frac{1}{2}x^\top H_k x + \mathrm{const}.$

NIM iteration: $x_{k+1} = x_k + \alpha(\bar{x}_k - x_k)$, where $\bar{x}_k := \operatorname{argmin} m_k(x)$.

Inexact minimization: instead of $\bar{x}_k$, use $\hat{x}_k$ such that

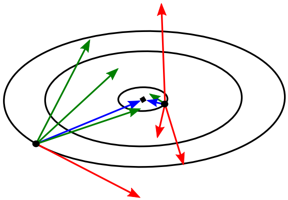
$$\|\nabla m_k(\hat{x}_k)\| \le \eta_k \|g_k\|, \qquad \eta_k := \left\{0.5, \sqrt{\|g_k\|}\right\}.$$

Recall: $g_k := \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(v_k^i) \approx \nabla\phi(x_k)$.

Convergence rate remains superlinear!

NIM: Inexact model minimization

Problem: $\min_x \left[\phi(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right]$.



(Assume $h \equiv 0$ for simplicity.)

Model: $m_k(x) = (g_k - u_k)^\top x + \frac{1}{2} x^\top H_k x + \mathrm{const.}$

NIM iteration: $x_{k+1} = x_k + \alpha(\bar{x}_k - x_k)$, where $\bar{x}_k := \operatorname{argmin} m_k(x)$.

Inexact minimization: instead of $\bar{x}_k$, use $\hat{x}_k$ such that

$$\|\nabla m_k(\hat{x}_k)\| \le \eta_k \|g_k\|, \qquad \eta_k := \left\{0.5, \sqrt{\|g_k\|}\right\}.$$

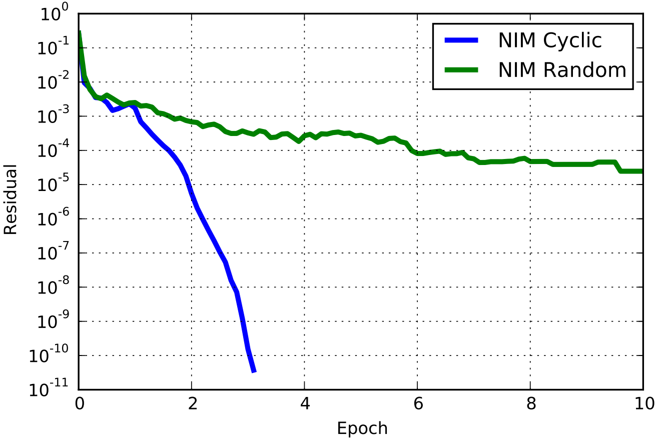
Recall: $g_k := \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(v_k^i) \approx \nabla \phi(x_k)$.

Convergence rate remains superlinear!

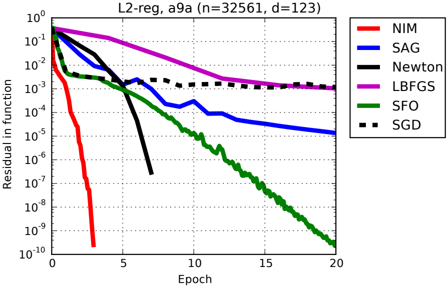
For $h \not\equiv 0$, all of this can be generalized using the **composite gradient mapping** (see paper for details).

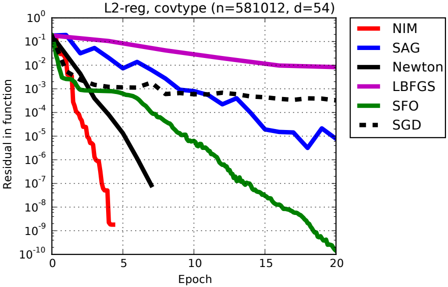
Order of component selection (cyclic vs randomized)

- What if randomized order is used in NIM instead of cyclic?



Experiments (ℓ_2-regularized logistic regression): Epochs





Experiments (ℓ_2-regularized logistic regression): Real time

L2-reg	*alpha (n=500 000, d=500)*				*mnist8m (n=8 100 000, d=784)*			
Res	NIM	SAG	Newton	LBFGS	NIM	SAG	Newton	LBFGS
10^{-1}	1.91s	**1.36s**	1.6m	4.01s	57.68s	**34.91s**	47.8m	1.1m
10^{-2}	13.37s	**6.72s**	2.6m	17.68s	**1.6m**	2.1m	1.4h	5.2m
10^{-3}	28.56s	**17.73s**	3.0m	37.70s	**3.2m**	3.9m	-	22.9m
10^{-4}	36.65s	**36.04s**	3.4m	58.35s	16.7m	**7.1m**	-	1.6h
10^{-5}	**46.66s**	1.0m	3.6m	1.4m	**26.7m**	1.0h	-	-
10^{-6}	**53.92s**	1.5m	4.0m	1.9m	**33.5m**	-	-	-
10^{-7}	**57.63s**	2.0m	4.0m	2.4m	**40.1m**	-	-	-
10^{-8}	**1.0m**	2.7m	4.1m	2.8m	**46.0m**	-	-	-
10^{-9}	**1.1m**	3.5m	4.3m	3.2m	**49.6m**	-	-	-
10^{-10}	**1.2m**	4.3m	4.7m	3.4m	**53.3m**	-	-	-

Inner solver: Conjugate Gradient Method.

Experiments (ℓ_1-regularized logistic regression): Real time

L1-reg	*alpha (n=500 000, d=500)*			*mnist8m (n=8 100 000, d=784)*		
Res	NIM	SAG	Newton	NIM	SAG	Newton
10^{-1}	26.76s	**1.31s**	1.1m	15.7m	**33.62s**	53.6m
10^{-2}	44.94s	**6.52s**	1.8m	37.0m	**2.1m**	1.8h
10^{-3}	55.56s	**17.26s**	2.3m	46.9m	**4.0m**	2.5h
10^{-4}	1.1m	**35.51s**	2.5m	1.0h	**7.3m**	3.1h
10^{-5}	1.3m	**1.0m**	2.9m	**1.2h**	1.4h	-
10^{-6}	**1.3m**	1.5m	3.1m	**1.5h**	-	-
10^{-7}	**1.4m**	2.1m	3.1m	**1.8h**	-	-
10^{-8}	**1.5m**	2.9m	3.5m	**2.3h**	-	-
10^{-9}	**1.6m**	3.8m	4.4m	**2.9h**	-	-
10^{-10}	**1.6m**	4.8m	4.5m	**3.4h**	-	-

Inner solver: Fast Gradient Method [Nesterov, 2013].

Conclusion

- The presented Newton-type Incremental Method (NIM) is the first incremental method with a superlinear rate of convergence.
- Method NIM can be seen as an incremental variant of the standard Newton method.
- NIM has the same advantages and disadvantages as the classic Newton method:
  - \+ Fast superlinear rate of convergence with the unit step length.
  - − Superlinear convergence is guaranteed only locally.
  - − Not applicable to high-dimensional problems.

Thank you!