Acceleration of Stochastic Variance-Reduced Gradient Methods

Junqi Tang

University of Cambridge

Nov 2021

Introduction

This talk is mainly based on these papers:

- [Nitanda, NeurIPS'14] Stochastic Proximal Gradient Descent with Acceleration Techniques.
- [Allen-Zhu, JMLR'17] Katyusha: the first direct acceleration of stochastic gradient methods.
- [Tang et al, NeurIPS'18] Rest-Katyusha: Exploiting the Solution's Structure via Scheduled Restart Schemes.
- [Scieur et al, NeurIPS'17] Nonlinear Acceleration of Stochastic Algorithms.

Introduction

Imaging inverse problems and large-scale optimization

Many inverse problems involve solving convex composite optimization tasks:

$$x^\star \in \arg\min_{x \in} \left\{ F(x) := \frac{1}{n} \sum_{i=1}^{n} \bar{f}(a_i, b_i, x) + \lambda g(x) \right\}, \qquad (1)$$

Data fidelity term $f(x) := \frac{1}{n} \sum_{i=1}^{n} \bar{f}(a_i, b_i, x)$, regularization $g(x)$.

In imaging inverse problems:

- $x \in \mathbb{R}^d \rightarrow$ **vectorized image**,
  $A = [a_1; a_2; ...; a_n] \in \mathbb{R}^{n \times d}$
  $\rightarrow$ **the forward model/measurements** ,
  $b = [b_1; b_2; ...; b_n] \in \mathbb{R}^n \rightarrow$ **the observations**.

$$b = Ax^\dagger + w, \quad A \in R^{n \times d} \tag{2}$$

- Example: Total-Variation regularized least-squares

$$F(x) := \frac{1}{2n}\|Ax - b\|_2^2 + \lambda\|Dx\|_1. \tag{3}$$

($D \rightarrow$ discrete gradient operator.)

Introduction

- $$x^\star \in \arg\min_{x\in\mathbb{R}^d} \left\{F(x) := f(x) + g(x)\right\}, f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{4}$$
- The number of data-sample n and dimension d can be large.
- Randomized optimization algorithms to rescue!!

Stochastic optimization

- Stochastic gradient algorithms typically pick one (or a few) functions f_i at random to calculate an unbiased estimate of the true gradient at each iteration.
- Stochastic Gradient Descent (SGD):

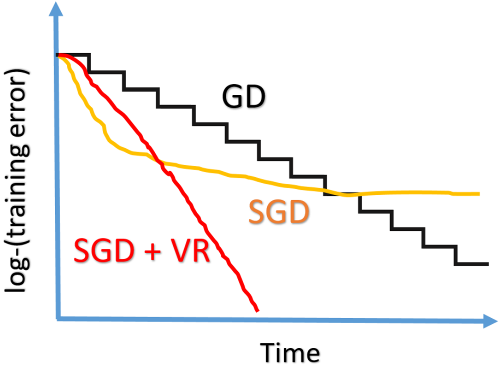
$$x^{t+1} = \mathrm{prox}_g^{\eta_t}\left(x^t - \eta_t \nabla f_j(x^t)\right), \tag{5}$$

- where the proximal operator is defined as:

$$\mathrm{prox}_g^{\eta}(\cdot) = \arg\min_{x \in \mathbb{R}^d} \frac{1}{2\eta}\|x - \cdot\|_2^2 + g(x). \tag{6}$$

Stochastic gradient methods with variance-reduction

- Recent advance: by **reducing the variance** of $\nabla f_j(x^t)$ one can achieve even faster convergence:



- Representative examples : SVRG [Johnson & Zhang, 2013], SAGA [Dafazio et al, 2014], SPDC [Zhang & Xiao, 2015], etc.

Optimal algorithms for regularized ERM

Gradient descent	$d\times \quad n\frac{L}{\mu} \quad \times \log\frac{1}{\varepsilon}$
Accelerated gradient descent	$d\times \quad n\sqrt{\frac{L}{\mu}} \quad \times \log\frac{1}{\varepsilon}$
SAG(A), SVRG, SDCA, MISO	$d\times \quad (n+\frac{L}{\mu}) \quad \times \log\frac{1}{\varepsilon}$
Accelerated versions	$d\times \, (n+\sqrt{n\frac{L}{\mu}}) \quad \times \log\frac{1}{\varepsilon}$

For example, the Katyusha (accelerated SVRG) algorithm [Allen-Zhu JMLR'17]
– Variance-reduced SGD with Nesterov-type acceleration achieves worse-case optimal convergence.

AccProxSVRG algorithm of Nitanda [NeurIPS'14]

The inner loop of AccProxSVRG consists of 3 steps :

$$
\begin{aligned}
&\text{For} \quad k = 0, 1, 2, ..., m \\
&\quad \nabla_{k+1} = \nabla f(\hat{x}^s) + \nabla f_i(x_k) - \nabla f_i(\hat{x}^s); \\
&\quad \rightarrow \text{variance reduced stochastic gradient} \\
&\quad y_{k+1} = \mathrm{prox}_g^{1/2L}(x_k - \tfrac{1}{2L}\nabla_{k+1}); \\
&\quad \qquad \rightarrow \text{proximal gradient descent} \\
&\quad x_{k+1} = y_{k+1} + +\theta_k(y_{k+1} - y_k); \\
&\quad \qquad \rightarrow \text{Nesterov momentum step}
\end{aligned}
$$

(we denote the inner-loop as $\mathcal{A}$)

Katyusha algorithm of Allen-Zhu [JMLR'17]

The inner loop of Katyusha consists of 4 steps :

For $k = 0, 1, 2, ..., m$

$$x_{k+1} = \theta z_k + \frac{1}{2}\hat{x}^s + (\frac{1}{2} - \theta) y_k;$$

$\rightarrow$ linear coupling momentum step

$$\nabla_{k+1} = \nabla f(\hat{x}^s) + \nabla f_i(x_{k+1}) - \nabla f_i(\hat{x}^s);$$

$\rightarrow$ variance reduced stochastic gradient

$$z_{k+1} = \text{prox}_g^{\frac{1}{3\theta L}}(z_k - \frac{1}{3\theta L}\nabla_{k+1});$$

$$y_{k+1} = \text{prox}_g^{\frac{1}{3L}}(x_k - \frac{1}{3L}\nabla_{k+1});$$

$\rightarrow$ proximal gradient descent

(we denote the inner-loop as $\mathcal{A}$)

Katyusha algorithm of Allen-Zhu [JMLR'17]

Algorithm Katyusha (x^0, m, S, L)

Initialize: $y^0 = z^0 = \hat{x}^0$;

for $s = 0, \ldots, S-1$ **do**

Set momentum parameter as $\theta \leftarrow \frac{2}{s+4}$,

Calculate a full gradient $\nabla f(\hat{x}^s)$,

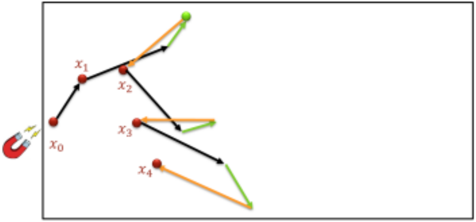
Inner-loop:

$$(\hat{x}^{s+1}, y^{s+1}, z^{s+1}) = \mathcal{A}(\hat{x}^s, y^s, z^s, \theta, \nabla f(\hat{x}^s), m)$$

end for

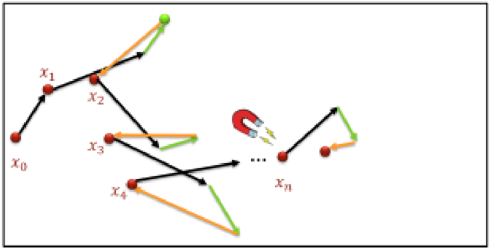
Output: $\hat{x}^S$

Katyusha acceleration of SVRG



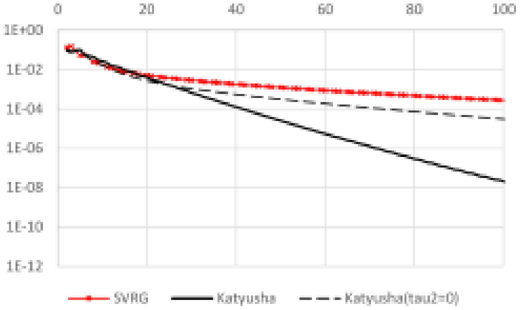
- Use negative momentum tracing towards an anchoring point to safe-guard Nesterov's momentum

Katyusha acceleration of SVRG



- Use negative momentum tracing towards an anchoring point to safe-guard Nesterov's momentum
- Updating occasionally (every O(1) epochs) the anchoring point

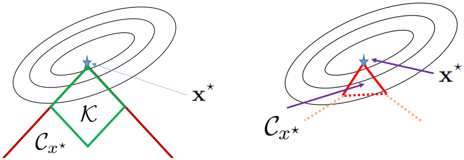
Katyusha acceleration of SVRG



- Achieving acceleration :)

Exploiting the solution's structure for faster algorithms

- Non-smooth $g(\cdot)$ injects prior information to ERM and often enforce the solution to be structured, e.g. sparse, piece-wise smooth, or low rank, etc.



(a) ERM with constraint $g(\cdot) := \iota_{\mathcal{K}}(.)$

(b) Regularized ERM

- **Can we exploit the solution's structure to design even faster optimization algorithms?**

Restricted strong-convexity due to the structure

- With the structure inducing regularization such as ℓ_1, $\ell_{2,1}$, and TV semi-norm, $f(\cdot)$ often satisfies restricted strong-convexity (RSC) w.r.t $g(\cdot)$:

$$f(x) - f(x^\star) - \langle \nabla f(x^\star), x - x^\star \rangle \geq \frac{\gamma}{2}\|x - x^\star\|_2^2 - \tau g^2(x - x^\star). \tag{7}$$

  $\forall x \in \mathbb{R}^d$.

- Let $x^\star \in \mathcal{T}$, and the complexity of subspace $\mathcal{T}$ denoted by $\Phi(\mathcal{T})$.
- We denote the effective RSC parameter as $\mu_c = \frac{\gamma}{2} - 32\tau\Phi^2(\mathcal{T})$,

$$F(x) - F(x^\star) \geq \mu_c\|x - x^\star\|_2^2 - \text{residuals}, \tag{8}$$

Exploit the structure via restart

An illustrative example for restarting the momentum-based algorithms

- FISTA algorithm solve ERM at a rate of $F(x^k) - F(x^\star) \leq \frac{4L\|x^0 - x^\star\|_2^2}{k^2}$
- If $F(\cdot)$ is μ-strongly convex, then:

$$F(x) - F(x^\star) \geq \mu\|x - x^\star\|_2^2 \tag{9}$$

Exploit the structure with restart

- Then if we run $k = \lceil 4\sqrt{L/\mu} \rceil$, we have:

$$F(x^k) - F(x^\star) \leq \frac{4L[F(x^0) - F(x^\star)]}{\mu k^2} \leq \frac{1}{4}[F(x^0) - F(x^\star)]. \tag{10}$$

- Hence if we restart FISTA every $\lceil 4\sqrt{L/\mu} \rceil$ iteration – only $k \geq \lceil 4\sqrt{\frac{L}{\mu}} \rceil \log_4 \frac{1}{\delta}$ iterations are needed to make $F(x^k) - F(x^\star) \leq \delta$.
- Without restart, FISTA needs $\frac{1}{\sqrt{\delta}}$ iterations.

Exploit the structure via restart

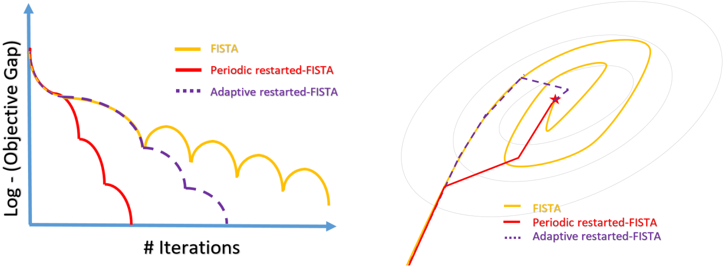


Figure: Empirical performance illustration of FISTA, periodic restarted FISTA (with exact knowledge of the strong-convexity parameter μ) and adaptive restarted-FISTA (based on enforcing monotonicity) for minimizing strongly-convex functions.

- We then leverage the restart scheme to accelerate Katyusha algorithm under RSC framework

Structure-Adaptive Accelerated Variance-Reduced SGD

Algorithm Rest-Katyusha $(x^0, \mu_c, S_0, \beta, T, L)$

Initialize: $m = 2n,\ S = \left\lceil \beta \sqrt{32 + \frac{24L}{m\mu_c}} \right\rceil$;

First stage — warm start:

$x^1 =$ Katyusha (x^0, m, S_0, L)

Second stage — exploit the restricted strong-convexity via periodic restart:

for $t = 1, ..., T$ **do**

$x^{t+1} =$ Katyusha (x^t, m, S, L)

end for

Output: x^{T+1}

- Convergence analysis: $\mathcal{O}\left(n + \sqrt{\frac{nL}{\mu_c}}\right) \log \frac{1}{\epsilon}$ gradient complexity – accelerated linear convergence

Structure-Adaptive Accelerated Variance-Reduced SGD

Algorithm Adaptive Rest-Katyusha $(x^0, \mu_0, S_0, \beta, T, L)$

Initialize: $m = 2n$; Initial restart period $S = \left\lceil \beta\sqrt{32 + \frac{12L}{n\mu_0}} \right\rceil$;

$x^1 = \text{Katyusha}\ (x^0, m, S_0, L)$

Calculate the composite gradient map:

$\mathcal{Q}(x^1) = \arg\min_x \frac{L}{2}\|x - x^1\|_2^2 + \langle \nabla f(x^1), x - x^1 \rangle + g(x).$

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

$x^{t+1} = \text{Katyusha}\ (x^t, m, S, L)$

Track the convergence speed via the composite gradient maps:

$\mathcal{Q}(x^{t+1}) = \arg\min_x \frac{L}{2}\|x - x^{t+1}\|_2^2 + \langle \nabla f(x^{t+1}), x - x^{t+1} \rangle + g(x).$

Update the estimate of RSC and tune the restart period:

if $\|\mathcal{Q}(x^{t+1}) - x^{t+1}\|_2^2 \leq \frac{1}{\beta^2}\|\mathcal{Q}(x^t) - x^t\|_2^2$

then $\mu_0 \leftarrow 2\mu_0$, **else** $\mu_0 \leftarrow \mu_0/2$. $S = \left\lceil \beta\sqrt{32 + \frac{12L}{n\mu_0}} \right\rceil$

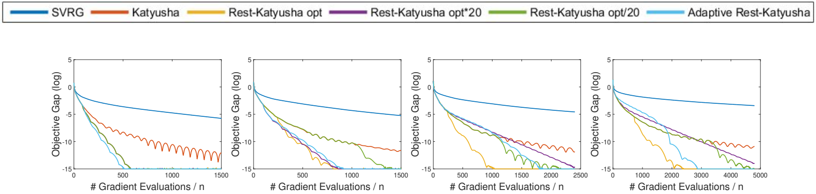
end for

Numerical experiments

- We test our algorithms' performance on LASSO problem:

$$x^\star \in \arg\min_{x \in \mathbb{R}^d} \left\{ F(x) := \frac{1}{2n} \|Ax - b\|_2^2 + \lambda \|x\|_1 \right\}. \qquad (11)$$

Figure: Lasso experiments on Reged dataset $(n, d) = [500, 999]$



$\lambda = 2 \times 10^{-5}$	$\lambda = 1 \times 10^{-5}$	$\lambda = 5 \times 10^{-6}$	$\lambda = 2 \times 10^{-6}$
$\|x^\star\|_0 = 80$	$\|x^\star\|_0 = 127$	$\|x^\star\|_0 = 209$	$\|x^\star\|_0 = 343$

Numerical experiments

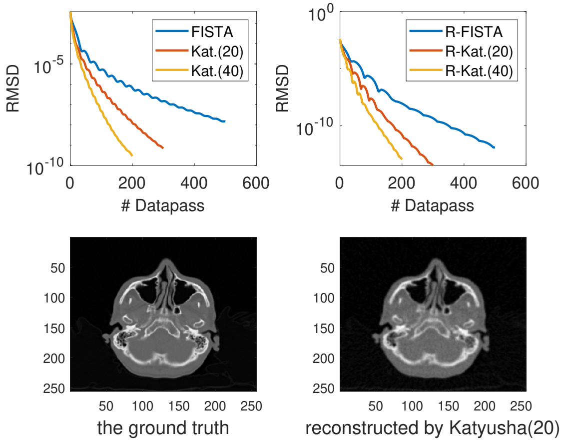
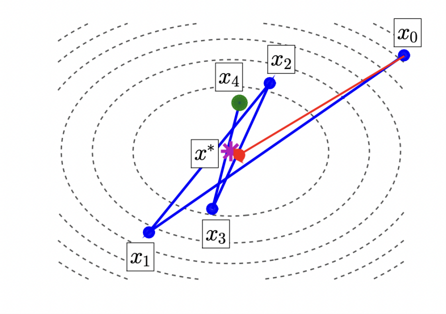


Figure: *X-ray CT image reconstruction experiment with a smooth edge-preserving regularization.* $\log_{10} \frac{\|Ax^\dagger\|_2^2}{\|w\|_2^2} \approx 3.16$.

Nonlinear Acceleration of Gradient-based Methods

1. Run a simple algorithm, e.g. gradient descent
2. "Guess" the solution using an extrapolation algorithm
3. Enjoy! ☺



Nonlinear Acceleration of Gradient-based Methods

Reegularized Nonlinear Acceleration (RNA)

Input: Sequence $\{x_0, \ldots, x_{k+1}\}$, parameter $\lambda > 0$

1: Form $R = [r_0, \ldots, r_k]$, where $r_i = x_{i+1} - x_i$ $\quad O(dk)$

2: Compute $R^T R$ $\quad O(dk^2)$

3: Compute $c^* = \frac{(R^T R + \lambda \mathbf{I})^{-1} \mathbf{1}}{\mathbf{1}^T (R^T R + \lambda \mathbf{I})^{-1} \mathbf{1}}$ $\quad O(k^3)$

Output: Return $x_{extr} = \sum_{i=0}^{k} c_i^* x_i \approx x^*$

[Scieur et al, NeurIPS'16]

Nonlinear Acceleration of Gradient-based Methods

Algorithmic complexity. In practice, $k \ll d$. Complexity is $O(d)$!

Sparse input. Complexity $O(k^2 s)$. Sparse output: $\|x_{extr}\|_0 \leq ks$.

Matlab/Python complexity. Only 5 lines of code!

Theorem (Scieur, d'Aspremont and Bach, 2016)

Asymptotic Acceleration Let $\|x_0 - x^*\| \to 0$ and λ well chosen,

$$\|x_{extr} - x^*\| \leq O\left((1-\sqrt{\kappa})^k \|x_0 - x^*\|\right) \qquad \text{(Optimal)}$$

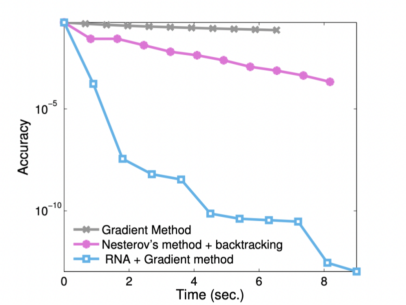
(Non-asymptotic bounds hold as well)

The gradient method on smooth and strongly convex functions meets the assumptions

Nonlinear Acceleration of Gradient-based Methods

Dataset: Madelon (2000 data points, 500 features, $\kappa = 10^{-6}$),

$$f(w) = \tau \|w\|_2^2 + \sum_{i=1}^{N} \log(1 + \exp(y_i X_i^T w)).$$



Nonlinear Acceleration of SVRG/SAGA. [Scieur et al, NeurIPS'17]

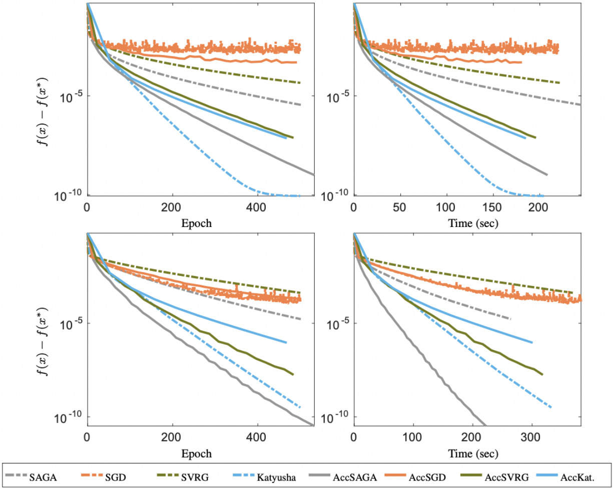


FIGURE 4. Optimization of quadratic loss (**Top**) and logistic loss (**Bottom**) with several algorithms, using the `Sid` dataset with bad conditioning. The experiments are done in Matlab. **Left:** Error vs epoch number. **Right:** Error vs time.