Stochastic Proximal Quasi-Newton methods for Nonconvex Composite Optimization

Ya-xiang Yuan

State Key Laboratory of Scientific and Engineering Computing
Inst of Comput. Math. and Scientific/Engineering Computing
AMSS, CAS, Beijing 100190, China
Email: yyx@lsec.cc.ac.cn
http://lsec.cc.ac.cn/∼yyx

joint work with Xiaoyu Wang and Xiao Wang

MOA 2018, Beijing, June 16-18, 2018

Outline

Outline

Outline

Outline

Outline

Outline

Introduction

Problem

$$\min_{x \in \mathcal{X}} \quad P(x) = F(x) + h(x), \tag{1}$$

where $\mathcal{X}$ is a closed and convex set in $\Re^d$,

$$F(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x). \tag{2}$$

- $f_i (i = 1, \cdots, n)$: smooth (possibly non-convex)
- h: convex, nonsmooth (often simple)

Applications in

- machine learning
- statistics
- ... , ...

Gradient-type Methods

Gradient Method (h=0)

$$x^{k+1} = x^k - \eta \nabla F(x^k) = x^k - \frac{\eta}{n} \sum_{i=1}^{n} \nabla f_i(x^k)$$

η: stepsize

Stochastic Gradient Method

$$x^{k+1} = x^k - \eta g_k$$

$$g_k = \nabla f_{i_k}(x^k) \quad \text{or} \quad g_k = \frac{1}{|S_k|} \sum_{j \in S_k} \nabla f_j(x^k)$$

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400-407, 1951.

Stochastic Quasi-Newton Method

$$x^{x+1} = x^k - \eta B_k^{-1} g_k$$

B_k: **quasi-Newton matrix**

For example:

- oLBFGS (Schraudolph, Yu and Gunte, 2007)
- SGD-QN (Bordes, Bottou and Gallinari, 2009)
- RES (Mokhtari and Ribeiro, 2014)
- SQN (Byrd, Hansen, Nocedal and Singer, 2016)
- SdLBFGS (Wang, Ma, Goldfarb and Liu, 2017)
- SC-L-BFGS (Curtis, 2016)
- ... ...

SQN based on Variation Reduction

Stochastic Graident: variance reduction techniques:

- SAG (Schmidt, Roux and Bach, 2017)
- SVRG (Johnson and Zhang, 2013)
- SDCA (Shalev-Shwartz and Zhang, 2017)
- SAGA (Defazio, Bach and Julien, 2014)
- MISO (Mairal, 2015)

SQN methods based on variance reduction

- SLBFGS (Moritz, Nishihara and Jordan, 2016)
- LiSSA (Agarwal, Bullins and Hazan, 2017)
- SdLBFGS-VR (Wang, Ma, Goldfarb and Liu, 2017)
- IQN (Mokhtari, Eisen and Ribeiro, 2017)

Proximal methods for min $F + h$

Proximal Gradient method:

$$x^{k+1} = \text{prox}_h(x^k - \eta \nabla F(x^k)), \tag{3}$$

where $\text{prox}_h(y) = \arg\min_y \left\{ h(y) + \frac{1}{2} \|x - y\|^2 \right\}$.

Proximal Second Order Method:

$$Q_k(y) = F(x^k) + \left\langle \nabla F(x^k), y - x^k \right\rangle + \frac{1}{2}(y - x^k)^T B_k (y - x^k) + h(y) \tag{4}$$

Becker and Fadili (2012)
Ghanbari and Scheinberg (2016)
Lee, Sun and Saunders (2014)
Lin, Mairal and Harchaoui(2016)

Our approach: proximal + second order + stochastic setting.

Outline

Preliminaries

Assumptions

AS1 (a) each f_i is twice continuously differentiable, bounded below and L-smooth, i.e. there is a constant $L > 0$ such that
$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L \|x - y\|, \quad \forall\, x, y \in \mathcal{X}$.

(b) $h(x)$ is lower semi-continuous

(c) $\mathcal{X}$ is a closed convex subset of $\Re^d$.

Generalized projected gradient (Ghadimi, Lan, Zhang, 2016)

$$P_{\mathcal{X}}(x, g, \alpha) = \alpha(x - x^+) \tag{5}$$

is used for convergence analysis, where

$$x^+ = \arg\min_{y \in \mathcal{X}} \left\{ \langle g, y - x \rangle + \frac{\alpha}{2} \|y - x\|^2 + h(y) - h(x) \right\} \tag{6}$$

with $\alpha > 0$.

Proximal Polyak-Lojasiew Inequality

$$\frac{1}{2}\mathcal{D}_h(x, L) \geq \mu(P(x) - P^*), \tag{7}$$

where ($\forall\, \alpha > 0$)

$$\mathcal{D}_h(x, \alpha) = -2\alpha \min_y \left\{ \langle \nabla F(x), y - x \rangle + \frac{\alpha}{2} \|y - x\|^2 + h(y) - h(x) \right\}. \tag{8}$$

Here P^* is the optimal value of objective function.

For constrained case, we define ($\forall\, \alpha > 0, x \in \mathcal{X}$)

$$\mathcal{D}_h^{\mathcal{X}}(x, g, B, \alpha) = -2\alpha \min_{y \in \mathcal{X}} \left\{ \langle g, y - x \rangle + \frac{\alpha}{2} \|y - x\|_B^2 + h(y) - h(x) \right\}, \tag{9}$$

where $B \in \mathbb{R}^{d \times d}$ is symmetric positive definite.

Some Lemmas

Lemma (1)

If x^+ is given by (6), then for any $x \in \mathcal{X}$, we have

$$\langle g, P_{\mathcal{X}}(x, g, \alpha)\rangle \geq \|P_{\mathcal{X}}(x, g, \alpha)\|_B^2 + \alpha(h(x^+) - h(x)).$$

Lemma (2)

For any fixed $B \succ 0$, we have

$$\mathcal{D}_h^{\mathcal{X}}(x, g, B, \alpha) \geq \|P_{\mathcal{X}}(x, g, \alpha)\|_B^2, \quad \forall\, x \in \mathcal{X}, \alpha > 0.$$

Lemma (3)

For differentiable f and convex h, for fixed x, g, B, we have

$$\mathcal{D}_h^{\mathcal{X}}(x, g, B, \alpha_2) \geq \mathcal{D}_h^{\mathcal{X}}(x, g, B, \alpha_1), \quad \forall\, \alpha_2 \geq \alpha_1 > 0.$$

PL Inequality

Definition

A Constrained Proximal Polyak-Lojasiewicz (CP-PL) inequality holds if there exists a constant $\mu > 0$ such that

$$\frac{1}{2}\mathcal{D}_h^{\mathcal{X}}(x, \nabla F(x), \mathrm{I}_d, L) \geq \mu(P(x) - P^*), \quad \forall\, x \in \mathcal{X}, \tag{10}$$

where P^* is the optimal value of problem (1)-(2).

Definition

A point $\bar{x} \in \mathcal{X}$ is an ϵ-approximate solution of (1)-(2), if

$$\mathbf{E}[\mathcal{D}_h^{\mathcal{X}}(\bar{x}, \nabla F(\bar{x}), \mathrm{I}_d, \alpha)] \leq \epsilon.$$

Outline

Algorithm (SPQN(x^0, B_0, T, m, b, η))

Input: $\hat{x}^0 = x_m^0 = x^0 \in \mathcal{X}$, $B_0^1 = B_0$, m, b and η

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

  Set $x_0^{t+1} = x_m^t$. Calculate $\hat{g} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\hat{x}^t)$

  for $j = 0$ *to* $m-1$ **do**

    Pick a minibatch set M_j^t such that $|M_j^t| = b$

    Calculate $g_j^{t+1} = \frac{1}{b}\sum_{i\in M_j}(\nabla f_i(x_j^{t+1}) - \nabla f_i(\hat{x}^t)) + \hat{g}$

    Obtain the exact solution x_{j+1}^{t+1} of the following subproblem

$$\min_{y\in\mathcal{X}} \quad \left\langle g_j^{t+1}, y - x_j^{t+1} \right\rangle + \frac{1}{2\eta}\left\| y - x_j^{t+1} \right\|_{B_j^{t+1}}^2 + h(y) - h(x_j^{t+1})$$

    Generate a symmetric positive definite matrix B_{j+1}^{t+1}

  end for

  Set $\hat{x}^{t+1} = x_m^{t+1}$

end for

Output: x_a, uniformly chosen from $\left\{ \left\{ x_j^{t+1} \right\}_{j=0}^{m-1} \right\}_{t=0}^{S-1}$

Theoretical Properties of SPQN

Assumption (AS2)

For any $t = 0, \ldots, S-1$, $j = 0, \ldots, m-1$, B_j^t is independent of M_j^t and there exist two positive constants $\underline{\lambda}, \overline{\lambda}$ such that

$$\underline{\lambda} \mathrm{I}_d \preccurlyeq B_j^t \preccurlyeq \overline{\lambda} \mathrm{I}_d.$$

Theorem (1)

Under AS1-AS2, $c_m = 0$, $c_j = c_{j+1}(1 + \frac{1}{\beta}) + \frac{2L^2}{b\theta}$ $(\beta, \theta > 0)$, $\eta \leq \frac{\underline{\lambda}}{(\theta + L + 2c_0(1+\beta))}$, and $T = Sm$. For output x_a of SPQN,

$$\mathbf{E}[\mathcal{D}_g^{\mathcal{X}}(x_a, \nabla F(x_a), \mathrm{I}_d, \frac{\overline{\lambda}}{\eta})] \leq \frac{2\overline{\lambda}(P(x^0) - P^*)}{\eta T} \tag{11}$$

where P^* is the optimal value of problem (1)-(2).

Theorem (2)

Under the same conditions as Theorem (1), further assume that $m = \lfloor n^r \rfloor (r > 0)$, $\beta = n^r$, $\theta = \frac{Ln^r}{\sqrt{b}}$, *then there exists a constant* $\nu > 0$ *such that* $\eta = \frac{\nu \underline{\lambda} \sqrt{b}}{Ln^r}$, *and*

$$\mathbf{E}[\mathcal{D}_h^{\mathcal{X}}(x_a, \nabla F(x_a), \mathrm{I}_d, \frac{\overline{\lambda}}{\eta})] \leq \frac{2L\overline{\lambda}n^r}{\nu \underline{\lambda} \sqrt{b} T}(P(x^0) - P^*). \tag{12}$$

Corollary (Complexity)

Under the same conditions as Theorem (2), assume that $b = n^{2/3}$, $m = n^{1/3}$, *then the step size* $\eta \leq \frac{\nu \underline{\lambda}}{L}$. *Therefore, the SFO and CPO complexity of algorithm 3.1 to achieve an* ϵ*-approximate solution of* (1)-(2) *are* $O(n + \frac{\kappa_1 n^{2/3}}{\epsilon})$ *and* $O(\frac{\kappa_1}{\epsilon})$, *respectively, where* $\kappa_1 = \frac{\overline{\lambda}}{\underline{\lambda}}$.

SFO: stochastic first order oracle, cost of computing $\nabla f_i(x)$
CPO: constrained proximal oracle, cost of obtaining $D_h^{\mathcal{X}}$

Algorithm (GD-SPQN(x^0, B_0, T, m, b, η))

1: ***Input***: *starting vector* $\hat{x}^0 = x_m^0 = x^0 \in \mathcal{X}$, *initial matrix* B_0, *innerloop update frequencey* m *and learning rate* η
2: ***for*** $k = 0, 1, \ldots, K-1$ ***do***
3: $\quad x^{k+1} = SPQN(x^k, B_k, T, m, b, \eta)$
4: ***end for***
5: ***Output:*** x^K

Theorem (3)

Under the same conditions as Theorem (2), assume the CP-PL inequality (10) *holds with the parameter* $\mu > 0$ *and set the parameter* T *as* $T = \lceil \frac{L\bar{\lambda}}{2\mu\nu\underline{\lambda}}(\frac{n^r}{\sqrt{b}}) \rceil$, *then we have*

$$\mathbf{E}[P(x^k) - P^*] \leq (2^{-k})[P(x^0) - P^*)]. \tag{13}$$

Corollary (Complexity)

Under the same conditions as Theorem (3), the SFO and CPO complexity of GD-SPQN to achieve ϵ*-approximate solution are* $O((n + \kappa_1\kappa_2(\frac{n}{\sqrt{b}} + n^r\sqrt{b}))\log(\frac{1}{\epsilon}))$ *and* $O(\frac{\kappa_1\kappa_2 n^r}{\sqrt{b}}\log(\frac{1}{\epsilon}))$, *respectively, where* $\kappa_1 = \frac{\bar{\lambda}}{\underline{\lambda}}$, $\kappa_2 = \frac{L}{\mu}$.

Outline

A Modified Self-Scaling SR1 method

Symmetric Rank-1

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$$

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}.$$

self-scaling SR1 method, OCSSR1 (Osborne and Sun, 1999)

$$H_{k+1} = \tau H_k + \frac{(s_k - \tau H_k y_k)(s_k - \tau H_k y_k)^T}{(s_k - \tau H_k y_k)^T y_k}, \tag{14}$$

$\tau = \frac{a}{b} - \sqrt{(\frac{a}{b})^2 - \frac{a}{c}}$, with $a = s_k^T B_k s_k$, $b = y_k^T s_k$ and $c = y_k^T H_k y_k$.

Proximal Operator Calculation

Theorem (4)

Let h be a proper and lower semi-continuous convex function, and $H = D + \sigma uu^T$ (σ is +1 or -1), where D is a diagonal matrix with positive diagonal elements and $u \in \mathbb{R}^d$. Then we have

$$prox_h^H(x) = D^{-1/2} \circ prox_{h \circ D^{-1/2}}(D^{1/2}x - \sigma v),$$

where $v = \alpha D^{-1/2}u$ and α is the unique root of the function

$$p(\alpha) = \left\langle u, x - D^{-1/2} \circ prox_{h \circ D^{-1/2}} \circ D^{1/2}(x - \sigma\alpha D^{-1}u) \right\rangle + \alpha$$

which is a Lipschitz continuous and strictly increasing function on $\mathbb{R}$.

$\sigma = 1$ (Thorem 7, Becker and Fadili, 2014).
$\sigma = -1$ is **new**!

Modified QN update

Replacing y_k by v_k: (idea originally from Powell, 1978)

$$v_k = \beta s_k + (1-\beta)\eta y_k$$

for some $\beta \in [0,1)$ such that

$$\theta_1 \leq \frac{v_j^T s_j}{s_j^T s_j}, \frac{v_j^T v_j}{v_j^T s_j} \leq \theta_2. \tag{15}$$

Algorithm (Modified Self-scaling Symmetric Rank one (MSSR1))

1: ***Input***: *Given* $\epsilon > 0$, $\theta_1 \in (0,1)$ *and* $\theta_2 \in (1,\infty)$
2: *Set* $s_j = x_{j+1}^{t+1} - x_j^{t+1}$, $y_j = \bar{g}_{j+1}^{t+1} - g_j^{t+1}$
3: *Compute*
$\beta_j = \arg\min \{\beta \in [0,1] \,|\, v(\beta) = \beta s_j + (1-\beta)\eta y_j \text{ satisfies (15)}\}$
4: *Set* $v_j = v(\beta_j)$
5: *Compute* $\tau = \frac{s_j^T s_j}{v_j^T s_j} - (\frac{(s_j^T s_j)^2}{(v_j^T s_j)^2} - \frac{s_j^T s_j}{v_j^T v_j})^{\frac{1}{2}}$ *and* $\rho = v_j^T s_j - \tau v_j^T v_j$
6: ***if*** $\rho \le \epsilon \|s_j - \tau v_j\|_2 \|v_j\|_2$
7: *set* $u_j = 0$ ***else*** *set* $u_j = \frac{s_j - \tau v_j}{\sqrt{\rho}}$
8: ***end if***
9: *Set* $H_{j+1}^{t+1} = \tau \mathrm{I}_d + u_j u_j^T$

$$\bar{g}_{j+1}^{t+1} := \frac{1}{b} \sum_{i \in M_j} \nabla f_i(x_{j+1}^{t+1}).$$

Positive Definite Property of MSSR1

Theorem

Let H_{j+1}^{t+1} *be updated by MSSR1. Then we have*

$$\underline{\lambda} \mathrm{I}_d \preccurlyeq H_{j+1}^{t+1} \preccurlyeq \overline{\lambda} \mathrm{I}_d$$

where $\underline{\lambda} = \frac{1}{2d\theta_2}$, $\overline{\lambda} = \tau d + \frac{1}{\epsilon\theta_1}$.

Our Algorithm **StSR1**: applying **MSSR1** in the framework of **SPQN**.

Outline

Numerical Results

Test Problems

$$P(x) = \frac{1}{n}\sum_{i=1}^{n}(1 - \tanh(b_i \langle a_i, x\rangle)) + \lambda \|x\|_1 \tag{16}$$

- $a_i \in \mathbb{R}^d$: feature vector
- $b_i \in \{-1, +1\}$ corresponding label
- $\lambda \geq 0$: regularization parameter

we compare our algorithm with

- Prox-SVRG (Reddi, Sra, Pczos and Smola, 2016)
- Prox-GD (Mine and Fukushima, 1981)

All the methods were implemented in Matlab 2014b on Dell desktop with Intel(R) Core(TM) i7-4790U CPU 3.6GHz, 8GB Memory.

Datasets

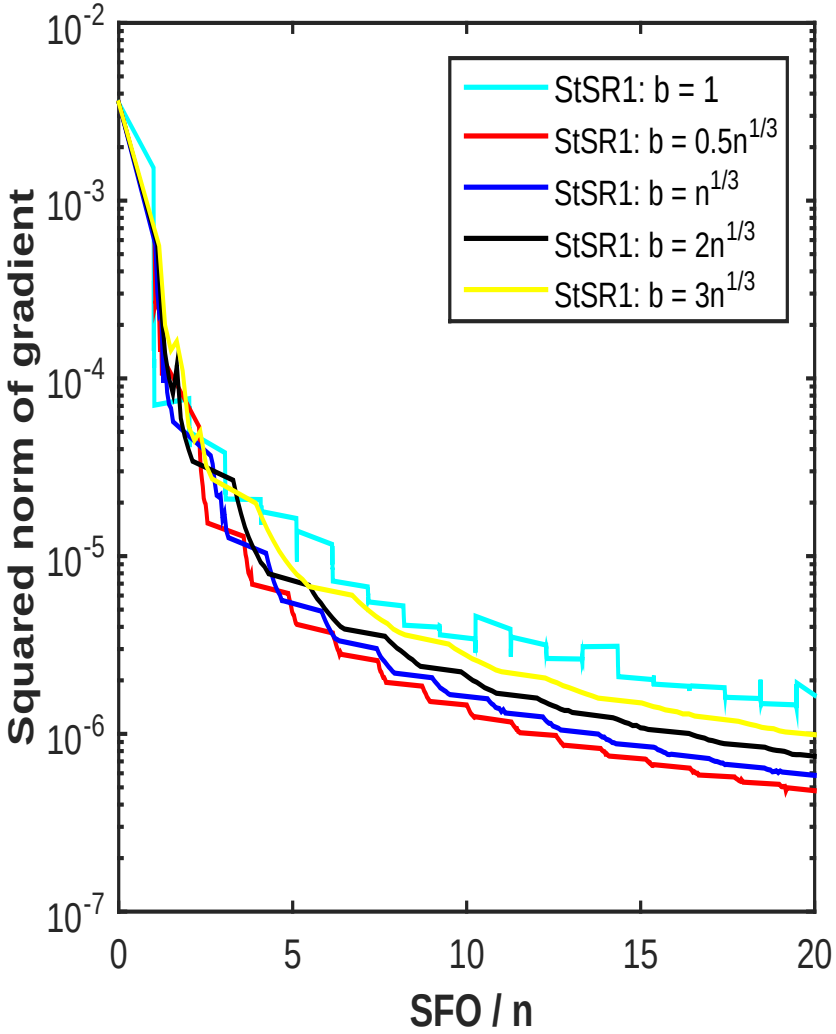
We tested three datasets from LIBSVM website[1]

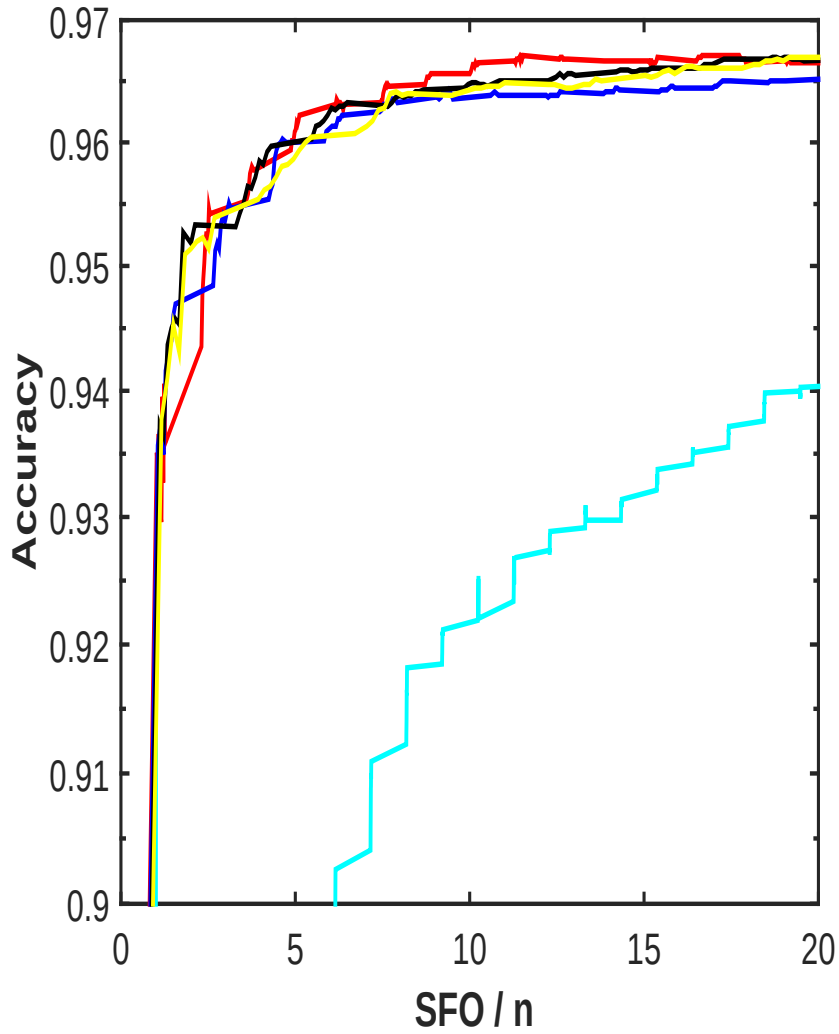
- n: number of the training data;
- N: number of the whole data;
- d: the dimension of the dataset;
- λ: regularization parameter.

Dataset	n/ N	d	λ
rcv1.binary	13495/20242	47236	10^{-5}
w6a	17188/49749	300	10^{-5}
real-sim	48206/72309	20958	10^{-5}

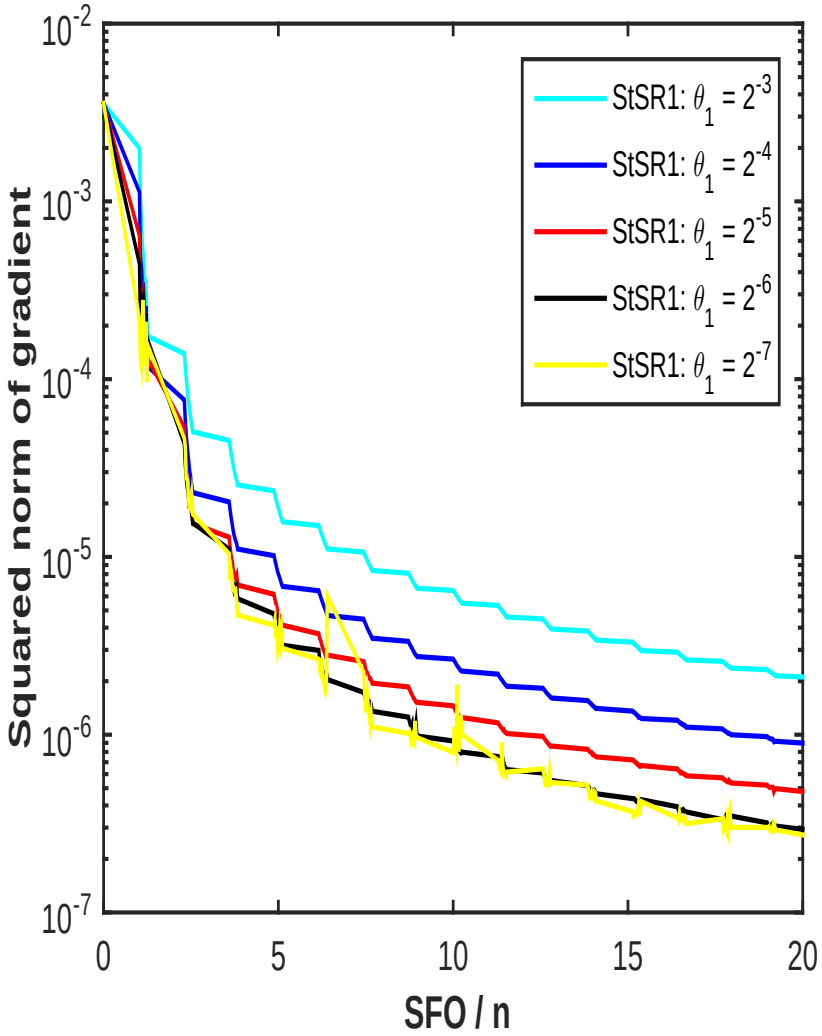
[1] https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/

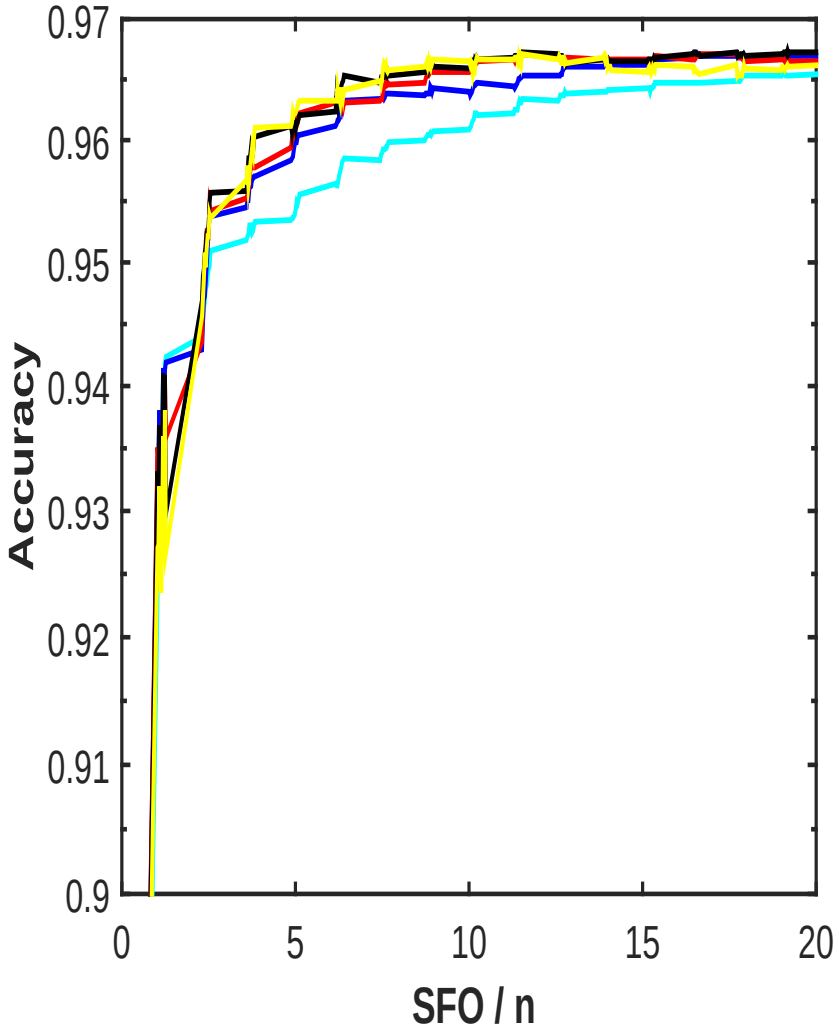
rcv1.binary with different minibatch size



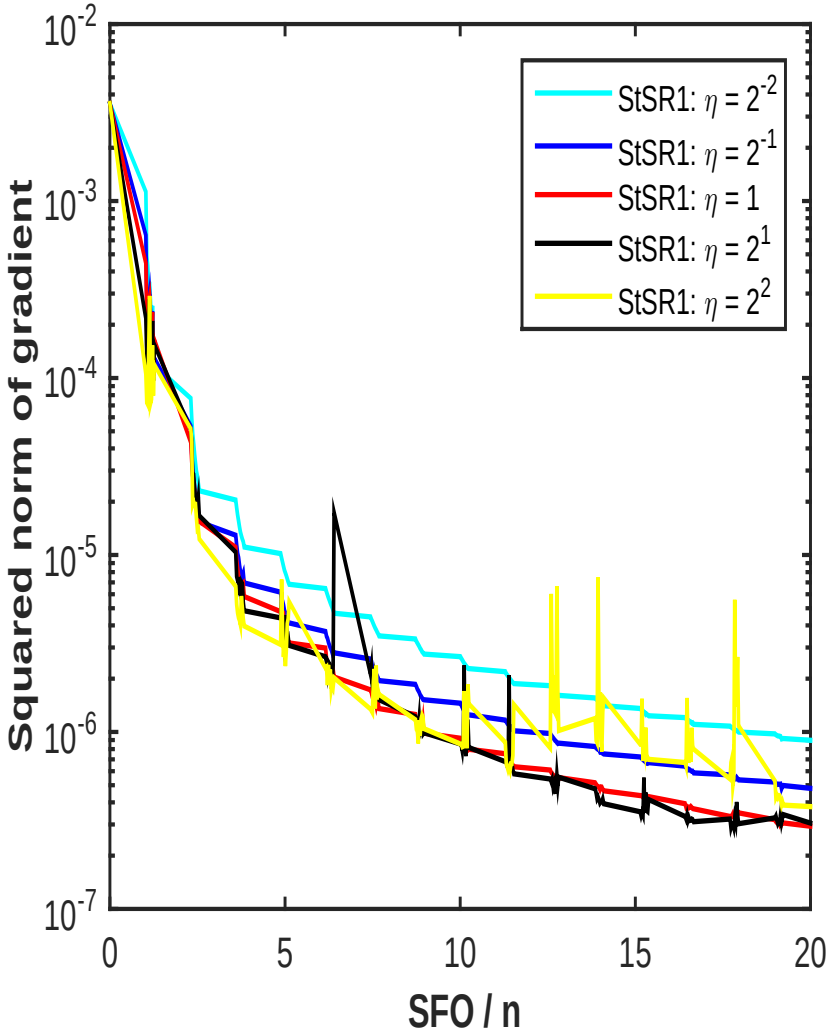


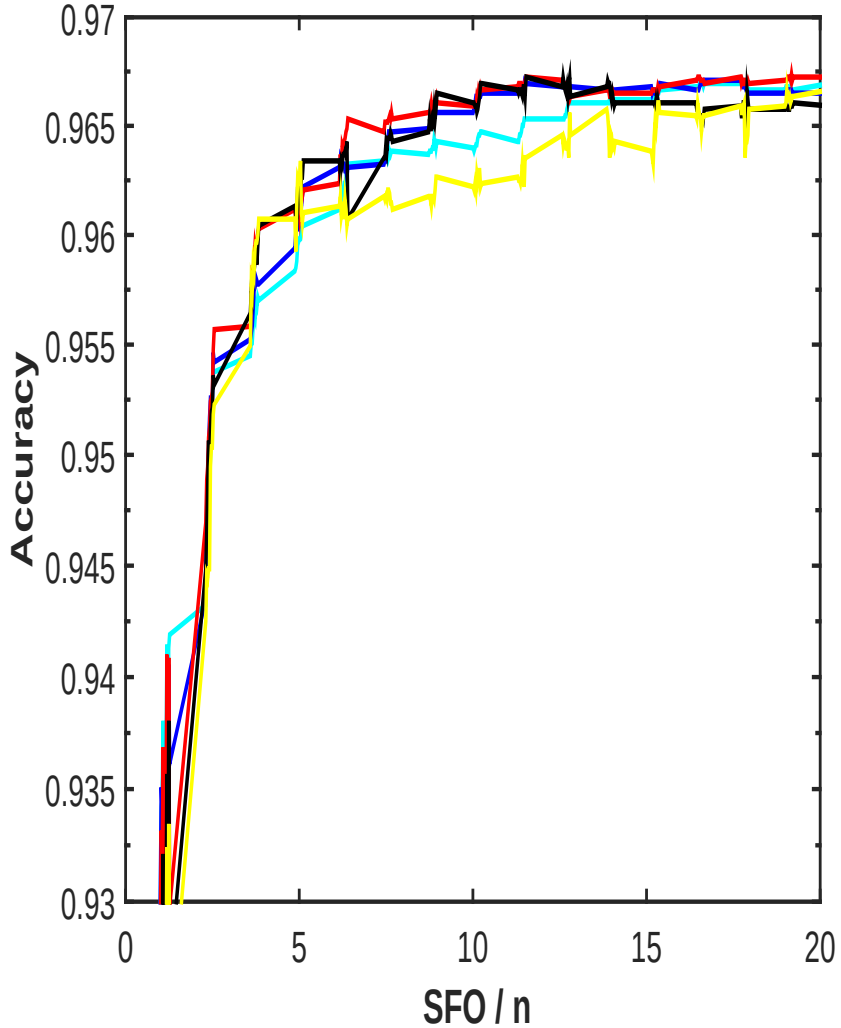
rcv1.binary dataset with different θ_1



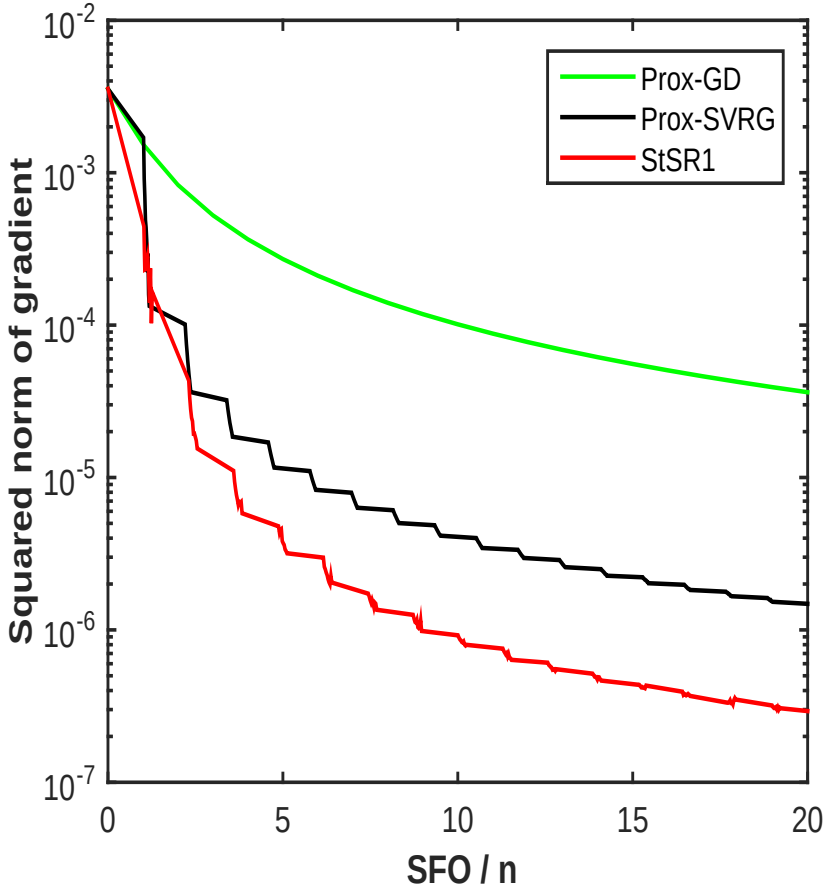


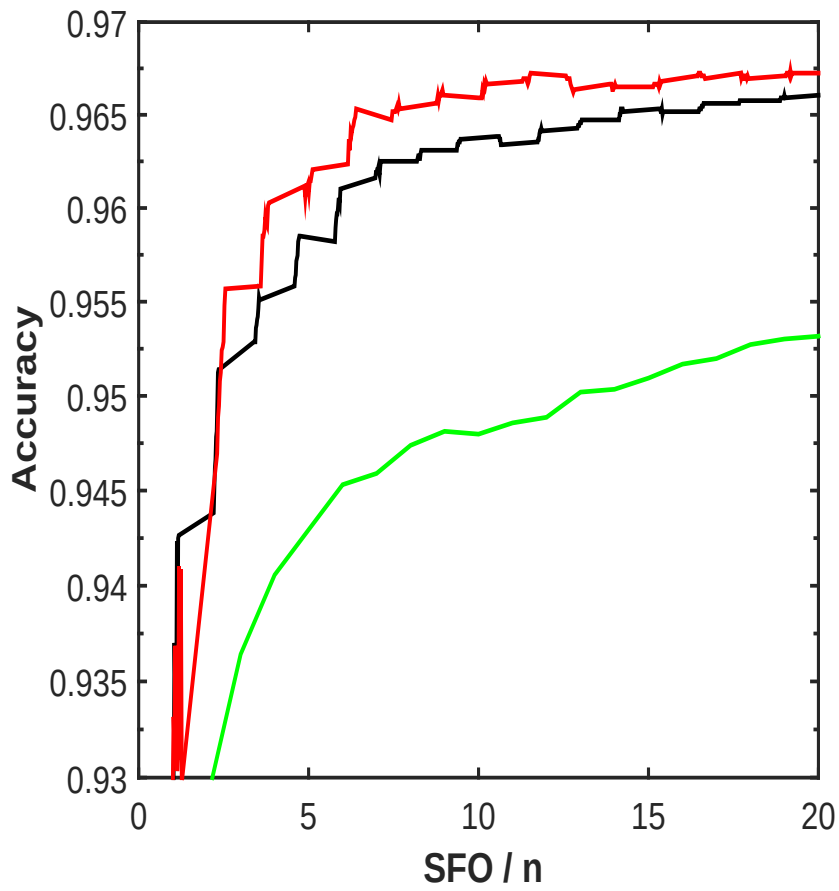
rcv1.binary dataset with different stepsize η





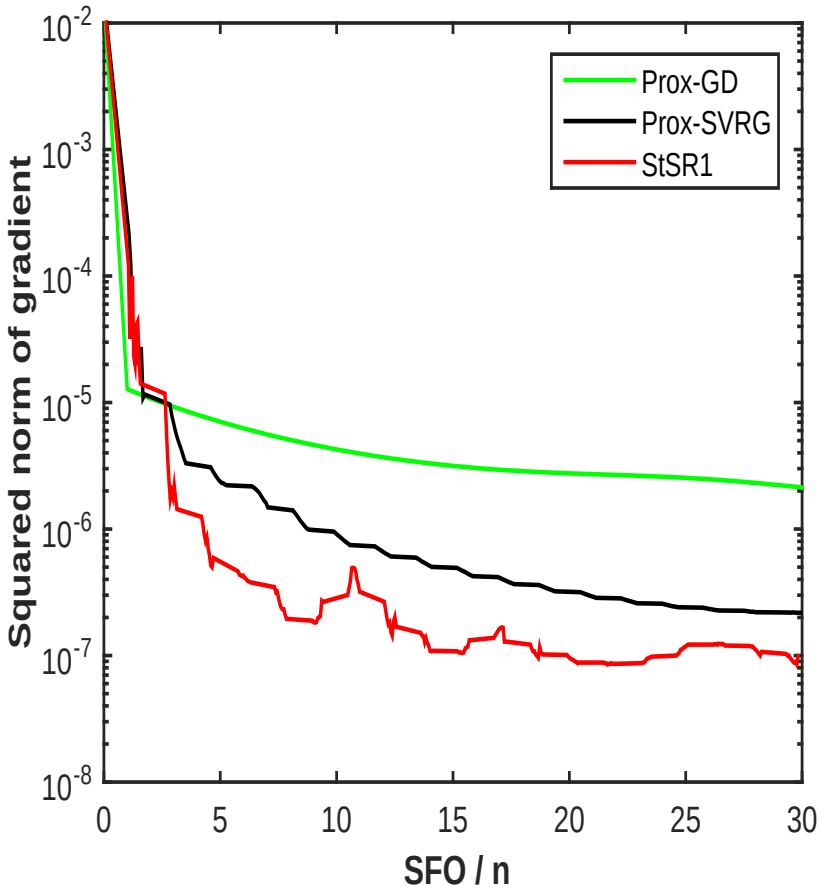
Comparison of different algorithms on rcv1.binary

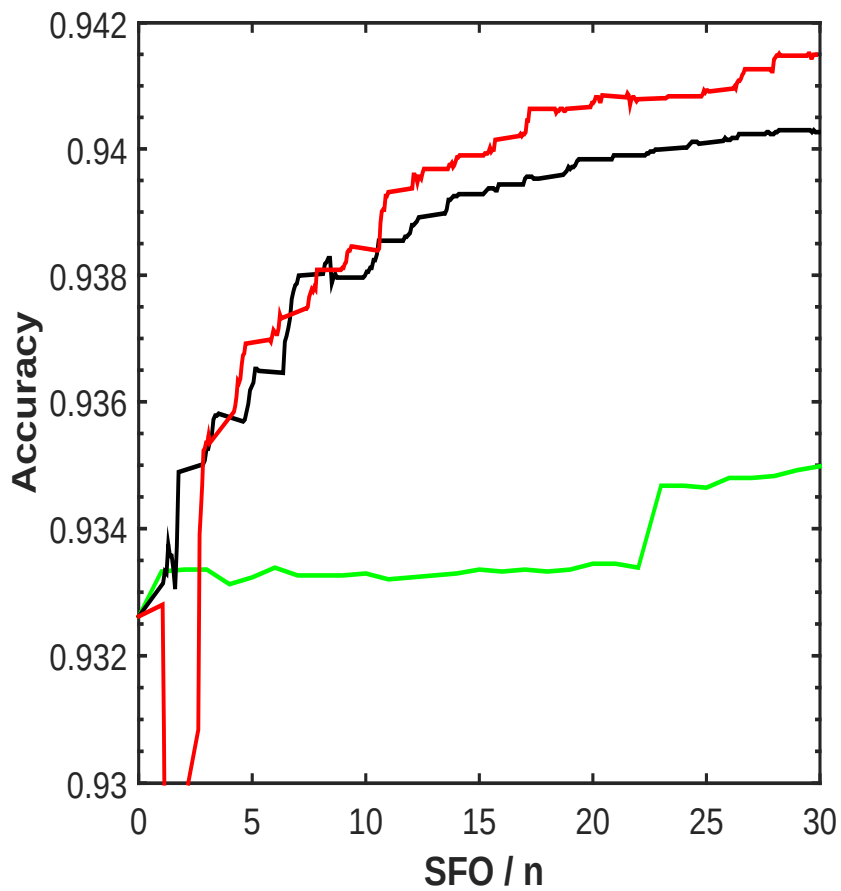




For StSR1 $(b, \theta_1, \theta_2, \eta) = (0.5n^{1/3}, 2^{-6}, 2^2, 1)$,
while $\eta = 10^2$ for Prox-GD and $\eta = 10$ for Prox-SVRG.

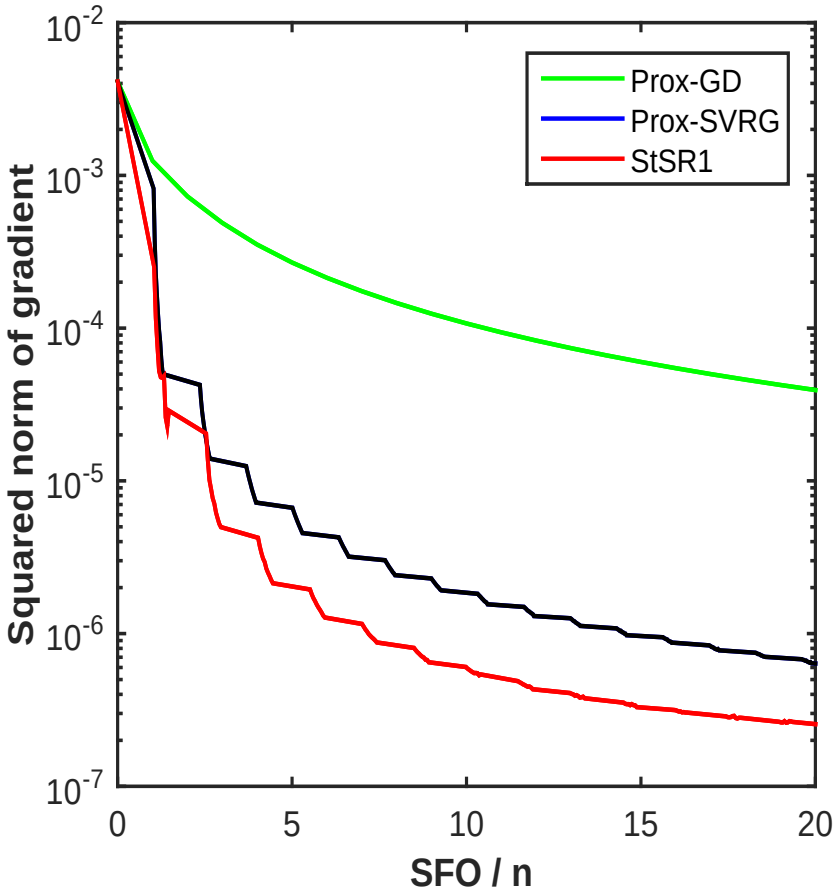
Comparison of three algorithms on **w6a**

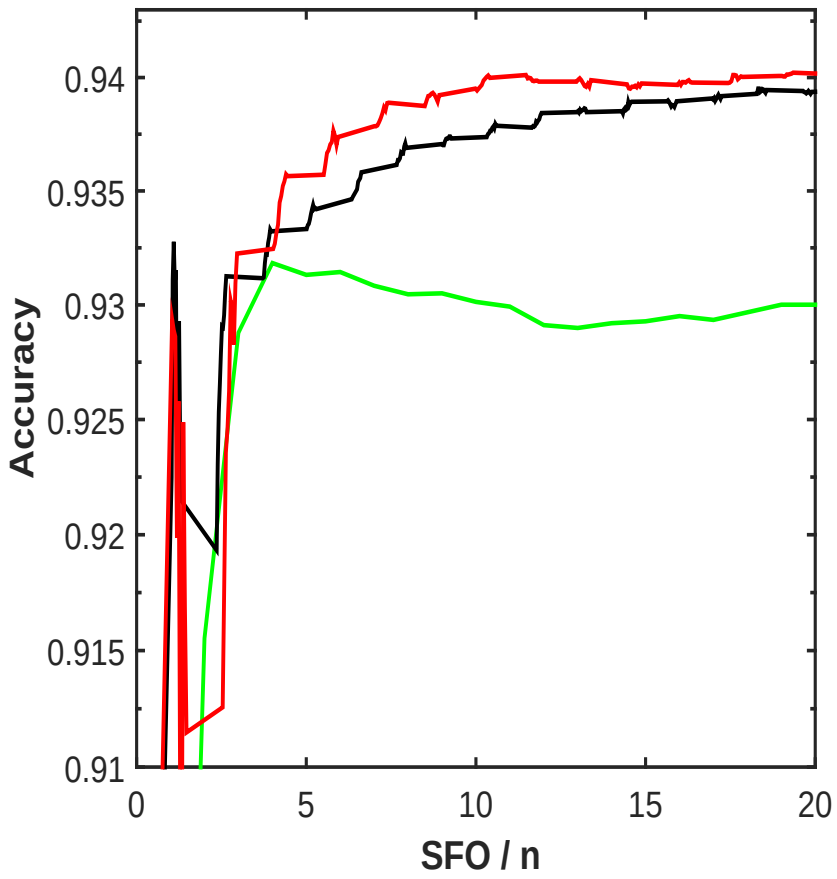




For StSR1 $(b, \theta_1, \theta_2, \eta) = (n^{1/3}, 2^{-5}, 2^2, 1)$,
while $\eta = 10^2$ for Prox-GD and $\eta = 10$ for Prox-SVRG.

Comparison of three algorithms on **real-sim**





For StSR1 $(b, \theta_1, \theta_2, \eta) = (n^{1/3}, 2^{-5}, 2^2, 1)$,
while $\eta = 10^2$ for Prox-GD and $\eta = 10$ for Prox-SVRG.

Conclusion

What we have done

- proposed a general framework, **SPQN**, stochastic proximal quasi-Newton methods
- proved the **global linear convergence rate** under CP-PL inequality.
- proposed a modified self-scaling symmetric rank one (MSSR1) method, leading to our **StSR1 method**.
- reported **numerical results** which show the comparable performance of StSR1 method to proximal SVRG and proximal GD methods.

to appear in *Optimization Methods and Software* (2018)

Thanks very much!