Stochastic Block BFGS: Squeezing More Curvature out of Data

Robert Mansel Gower
Joint work with Donald Goldfarb and Peter Richtárik

International Conference on Machine Learning, New York, June 2016

The Problem

$$\min_{w \in \mathbf{R}^d} f(w) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(w).$$

- Each f_i is strongly convex and twice continuously differentiable.
- Far more data samples than features $n \gg d$, access through subsampling

The Problem

Motivation is from stochastic optimization, such as empirical risk minimization

$$\min_{w \in \mathbf{R}^d} f(w) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(w).$$

- Each f_i is strongly convex and twice continuously differentiable.
- Far more data samples than features $n \gg d$, access through subsampling

The Problem

Motivation is from stochastic optimization, such as empirical risk minimization

$$\min_{w \in \mathbf{R}^d} f(w) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(w).$$

- Each f_i is strongly convex and twice continuously differentiable.
- Far more data samples than features $n \gg d$, access through subsampling

$$\nabla f_S(w) \overset{\text{def}}{=} \frac{1}{|S|} \sum_{i \in S} \nabla f_i(w) \qquad \nabla^2 f_T(x) \overset{\text{def}}{=} \frac{1}{|T|} \sum_{i \in T} \nabla^2 f_i(x)$$

$S \subset \{1, \ldots, n\}$ $\qquad$ $T \subset \{1, \ldots, n\}$

Variable Metric Method

$$w_{t+1} = w_t - \eta H_t g_t$$

Variable Metric Method

$$w_{t+1} = w_t - \eta H_t g_t$$

stepsize

Variable Metric Method

$$w_{t+1} = w_t - \eta H_t g_t$$

stepsize

$\mathbf{E}[g_t] = \nabla f(w_t)$

Variable Metric Method

$H_t \approx \nabla^2 f(w_t)^{-1}$

$$w_{t+1} = w_t - \eta H_t g_t$$

stepsize

$\mathbf{E}[g_t] = \nabla f(w_t)$

Variable Metric Method

$$H_t \approx \nabla^2 f(w_t)^{-1}$$

$$w_{t+1} = w_t - \eta H_t g_t$$

stepsize

$$\mathbf{E}[g_t] = \nabla f(w_t)$$

Exe:

- Newton's Method

$$w_{t+1} = w_t - \eta \nabla^2 f(w_t)^{-1} \nabla f(w_t)$$

- Steepest descent

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

- Stochastic gradient descent (SGD)

$$w_{t+1} = w_t - \eta \nabla f_S(w_t)$$

- SAG, SVRG, S2GD, ...etc

Variable Metric Method

$H_t \approx \nabla^2 f(w_t)^{-1}$

$$w_{t+1} = w_t - \eta H_t g_t$$

stepsize

$\mathbf{E}[g_t] = \nabla f(w_t)$

Exe:

- Newton's Method

$$w_{t+1} = w_t - \eta \nabla^2 f(w_t)^{-1} \nabla f(w_t)$$

- Steepest descent

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

- Stochastic gradient descent (SGD)

$$w_{t+1} = w_t - \eta \nabla f_S(w_t)$$

- SAG, SVRG, S2GD, ...etc

Challenge how to construct an effective H_t that is cheap to calculate?

Stochastic Second order Methods

H_t is directly estimated from $\nabla^2 f_T(x_t)$

- Low rank decomposition (Agarwal, Bullins and Hazan 2016)
- SVD decomposition (Erdogdu and Montanari 2015)
- Sketching full Hessian (Pilanci and Wainwright 2015)

H_t is updated using the (L)BFGS update

- (Schraudolph, Yu and Gunter 2007)
- (Mokhtari and Ribeiro 2014, 2015)
- (Byrd, Hansen, Nocedal and Singer 2015)
- (**MNJ** Moritz, Nishihara, Jordan 2016)

Hessian Sketching

Fact: Calculating a directional derivative of the gradient is cheap

$$\nabla^2 f_T(x_t)v = \frac{d}{d\alpha}\nabla f_T(x_t + \alpha v)\Big|_{\alpha=0}$$

Ideally H_t should satisfy the *inverse equation*

$$H_t \nabla^2 f_T(x_t) = I$$

Solving the **sketched inverse equation** is easier

$$H_t \nabla^2 f_T(x_t) D_t = D_t$$

Hessian Sketching

Fact: Calculating a directional derivative of the

This can be computed at a cost of $O(\mathrm{eval}(f_T(x))$ with Automatic Differentiation.

$$\nabla^2 f_T(x_t)v = \frac{d}{d\alpha}\nabla f_T(x_t+\alpha v)\Big|_{\alpha=0}$$

Ideally H_t should satisfy the *inverse equation*

$$H_t\nabla^2 f_T(x_t) = I$$

Solving the **sketched inverse equation** is easier

$$H_t\nabla^2 f_T(x_t)D_t = D_t$$

Hessian Sketching

This can be computed at a cost of $O(\text{eval}(f_T(x))$ with Automatic Differentiation.

Fact: Calculating a directional derivative of the

$$\nabla^2 f_T(x_t)v = \frac{d}{d\alpha}\nabla f_T(x_t + \alpha v)\Big|_{\alpha=0}$$

Ideally H_t should satisfy the *inverse equation*

$$H_t \nabla^2 f_T(x_t) = I$$

Solving the **sketched inverse equation** is easier

$$H_t \nabla^2 f_T(x_t) D_t = D_t$$

$D_t \in \mathbf{R}^{d\times q}, q \ll d$

Hessian Sketching

Fact: Calculating a directional derivative of the

This can be computed at a cost of $O(\mathrm{eval}(f_T(x))$ with Automatic Differentiation.

$$\nabla^2 f_T(x_t)v = \frac{d}{d\alpha}\nabla f_T(x_t + \alpha v)\Big|_{\alpha=0}$$

Ideally H_t should satisfy the *inverse equation*

$$H_t \nabla^2 f_T(x_t) = I$$

Solving the **sketched inverse equation** is easier

$$H_t \nabla^2 f_T(x_t) D_t = D_t$$

Cost of evaluating $\nabla^2 f_T(x_T) D_t$ is $\mathrm{q} \times O(\mathrm{eval}(f_T(x))$

$D_t \in \mathbf{R}^{d \times q}, q \ll d$

Block BFGS: Least change formulation

$$H_t = \arg\min_{H \in \mathbb{R}^{d\times d}} ||H - H_{t-1}||_t^2$$

$$\text{subject to} \quad H\nabla^2 f_T(x_t) D_t = D_t, \quad H = H^T$$

$$\text{where } ||H||_t^2 \stackrel{\text{def}}{=} \mathbf{Tr}\left(H\nabla^2 f_T(x_t) H^T \nabla^2 f_T(x_t)\right).$$

The constraint serves as a fidelity term, enforcing that a sketch of the inverse equation be satisfied

The objectives serves as a regularizor, enforcing a low rank update

Goldfarb, D. (1970). **A Family of Variable-Metric Methods Derived by Variational Means**. Mathematics of Computation, 24(109), 23.

Block BFGS: Random update formulation

Cost of update:

$$O(d^2 \times q)$$

$$H_t = D_t \Delta_t D_t^T + \left(I - D_t \Delta_t Y_t^T\right) H_{t-1} \left(I - Y_t \Delta_t D_t^T\right),$$

$$\text{where } Y_t = \nabla^2 f_T(x_t) D_t \text{ and } \Delta_t = (D_t^T Y_t)^{-1}$$

RMG and Peter Richtárik (2016). **Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms**. arXiv:1602.01768

Stochastic Block BFGS Method

Initialize $H_{-1} = I, w_0 \in \mathbf{R}^d,$ stepsize $\eta > 0$

For $t = 0, 1, \ldots,$

1 Calculate g_t

2. Sample $T_t \subseteq [n],$ independently

3. Form $D_t \in \mathbf{R}^{d \times q}$

4. Compute sketch $Y_t = \nabla^2 f_{T_t}(w_t) D_t$

5. $$H_t = D_t \Delta_t D_t^T + \left(I - D_t \Delta_t Y_t^T\right) H_{t-1} \left(I - Y_t \Delta_t D_t\right)$$

6. $d_t = H_t g_t$

7. $w_{t+1} = w_t - \eta d_t$

Output w_{t+1}

Stochastic Block BFGS Method

Initialize $H_{-1} = I, w_0 \in \mathbf{R}^d, \text{ stepsize } \eta > 0$

For $t = 0, 1, \ldots,$

1 Calculate g_t

2. Sample $T_t \subseteq [n], \text{ independently}$

3. Form $D_t \in \mathbf{R}^{d \times q}$

4. Compute sketch $Y_t = \nabla^2 f_{T_t}(w_t) D_t$

5. $H_t = D_t \Delta_t D_t^T + \left(I - D_t \Delta_t Y_t^T\right) H_{t-1} \left(I - Y_t \Delta_t D_t\right)$

6. $d_t = H_t g_t$

7. $w_{t+1} = w_t - \eta d_t$

Output w_{t+1}

How to choose D_t?

Stochastic Block BFGS Method

Initialize $H_{-1} = I, w_0 \in \mathbf{R}^d, \text{ stepsize } \eta > 0$

For $t = 0, 1, \ldots,$

1 Calculate g_t

2. Sample $T_t \subseteq [n], \text{ independently}$

3. Form $D_t \in \mathbf{R}^{d \times q}$

4. Compute sketch $Y_t = \nabla^2 f_{T_t}(w_t) D_t$

5. $H_t = D_t \Delta_t D_t^T + \left(I - D_t \Delta_t Y_t^T\right) H_{t-1} \left(I - Y_t \Delta_t D_t\right)$

6. $d_t = H_t g_t$

7. $w_{t+1} = w_t - \eta d_t$

Output w_{t+1}

How to choose D_t?

Do we need to store H_t?

Choosing the sketch matrix

$$H_t \nabla^2 f_T(x_t) D_t = D_t$$

We employ one of three strategies

- **gauss:** $D_t \sim \mathcal{N}(0, I)$ has Gaussian entries sampled i.i.d at each iteration
- **prev** (**prev**ious search directions delayed) : Let $d_t = H_t g_t$. Store q previous search directions $D_t = [d_{t-q}, \ldots, d_{t-1}]$, update H_t once every q iterations
- **fact** (**fact**orized self-conditioning) : Sample the columns of a factored form L_t of H_t ($i.e.\, L_t L_t^T = H_t$) uniformly at random. Fortunately we can maintain and update L_t efficiently.

Limited Memory Block BFGS

Expanding $M \in \mathbb{N}$ block BFGS updates gives

$$
\begin{aligned}
H_t &= \left(I - D_t \Delta_t Y_t^T\right) H_{t-1} \left(I - Y_t \Delta_t D_t^T\right) + D_t \Delta_t D_t^T \\
&\vdots \\
&= \mathrm{FUNCTION}\left(H_{t-M}, D_t, Y_t, \Delta_t, \ldots, D_{t+1-M}, Y_{t+1-M}, \Delta_{t+1-M}\right)
\end{aligned}
$$

Limited Memory Block BFGS

Expanding $M \in \mathbb{N}$ block BFGS updates gives

$$
\begin{aligned}
H_t &= \left(I - D_t \Delta_t Y_t^T\right) H_{t-1} \left(I - Y_t \Delta_t D_t^T\right) + D_t \Delta_t D_t^T \\
&\vdots \\
&= \mathrm{FUNCTION}\left(H_{t-M}, D_t, Y_t, \Delta_t, \ldots, D_{t+1-M}, Y_{t+1-M}, \Delta_{t+1-M}\right)
\end{aligned}
$$

H_t is a function of H_{t+1-M} and $(D_{t+1-i}, Y_{t+1-i}, \Delta_{t+1-i})$ for $i = 1, \ldots M$

Limited Memory Block BFGS

Expanding $M \in \mathbb{N}$ block BFGS updates gives

$$
\begin{aligned}
H_t &= \left(I - D_t \Delta_t Y_t^T\right) H_{t-1} \left(I - Y_t \Delta_t D_t^T\right) + D_t \Delta_t D_t^T \\
&\;\vdots \\
&= \text{FUNCTION}\left(\cancel{H_{t-M}}, D_t, Y_t, \Delta_t, \ldots, D_{t+1-M}, Y_{t+1-M}, \Delta_{t+1-M}\right) \\
&= \text{FUNCTION}\left(D_t, Y_t, \Delta_t, \ldots, D_{t+1-M}, Y_{t+1-M}, \Delta_{t+1-M}\right)
\end{aligned}
$$

H_t is a function of H_{t+1-M} and $(D_{t+1-i}, Y_{t+1-i}, \Delta_{t+1-i})$ for $i = 1, \ldots M$

To simplify $H_{t-M} = I$

Limited Memory Block BFGS

Store the M block triples

$$\left(D_t, Y_t, \Delta_t\right), \ldots, \left(D_{t+1-M}, Y_{t+1-M}, \Delta_{t+1-M}\right)$$

Limited Memory Block BFGS

Store the M block triples

$$\text{Store } M(2qd + q^2) \text{ doubles}$$

$$(D_t, Y_t, \Delta_t), \ldots, (D_{t+1-M}, Y_{t+1-M}, \Delta_{t+1-M})$$

Limited Memory Block BFGS

Store the M block triples

Store $M(2qd + q^2)$ doubles

$$(D_t, Y_t, \Delta_t), \ldots, (D_{t+1-M}, Y_{t+1-M}, \Delta_{t+1-M})$$

Calculate $H_t g_t$ using the following algorithm

Two-loop recursion

inputs $g_t \in \mathbf{R}^d, D_i, Y_i \in \mathbf{R}^{d\times q}$ and $\Delta_i \in \mathbf{R}^{q\times q}$

For $i = t, \ldots, t - M + 1$

$\quad \alpha_i \leftarrow \Delta_i D_i^T v$

$\quad v \leftarrow v - Y_i \alpha_i$

For $i = t - M + 1, \ldots, t$

$\quad \beta_i \leftarrow \Delta_i Y_i^T v$

$\quad v \leftarrow v + D_i(\alpha_i - \beta_i)$

output $H_t g_t \leftarrow v$

Costs $Mq(4d + 2q)$ to apply

Stochastic Block BFGS Method

Initialize $H_{-1} = I, w_0 \in \mathbf{R}^d$, stepsize $\eta > 0$

For $t = 0, 1, \ldots,$

1. Calculate g_t ← Using SVRG
2. Sample $T_t \subseteq [n]$, independently
3. Form $D_t \in \mathbf{R}^{d \times q}$
4. Compute sketch $Y_t = \nabla^2 f_T(w_t) D_t$
5. $d_t = H_t g_t$ ← Two-loop recursion
6. $w_{t+1} = w_t - \eta d_t$

Output w_{t+1}

Full Algorithm

Stochastic Block BFGS Method

Initialize $H_{-1} = I, w_0 \in \mathbf{R}^d,$ stepsize $\eta > 0$

For $t = 0, 1, \ldots,$

1. Compute $\mu = \nabla f(w_t)$
2. Set $x_0 = w_t$

   For $k = 0, 1, \ldots, m-1$

3. Sample $S_k, T_k \subseteq [n],$ independently
4. $g_k = \nabla f_{S_k}(x_k) - \nabla f_{S_k}(w_t) + \mu$
5. Form $D_k \in \mathbf{R}^{d \times q}$
6. $x_{k+1} = x_k - \eta H_k g_k$
7. **Option I:** Set $w_{t+1} = x_m$
8. **Option I:** Set $w_{t+1} = x_i,$ where i is selected uniformly at random from $[m] = \{1, 2, \ldots, m\}$

Output w_{t+1}

Back

Experiments

Logistic regression with L2 regularizer

Test problem

$$\min_w \sum_{i=1}^{n} \ln\left(1 + \exp(-y_i a^i, w)\right) + \frac{1}{n}||w||_2^2,$$

where $[a^1, \ldots, a^n] \in \mathbf{R}^{d\times n}$ and $y \in \{0,1\}^n$ are the given data.

Data taken from LIBSVM

Key to methods

SVRG	Johnson Zhang (2013)
MNJ	Moritz, Nishihara Jordan (2013)
gauss_q_M	Gaussian elements
fact_q_M	Self-conditioning factorized sampling
prev_q_M	Previous search directions delayed

Key to methods

SVRG	Johnson Zhang (2013)
MNJ	Moritz, Nishihara Jordan (2013)
gauss_q_M	Gaussian elements
fact_q_M	Self-conditioning factorized sampling
prev_q_M	Previous search directions delayed

Stochastic Block BFGS Methods

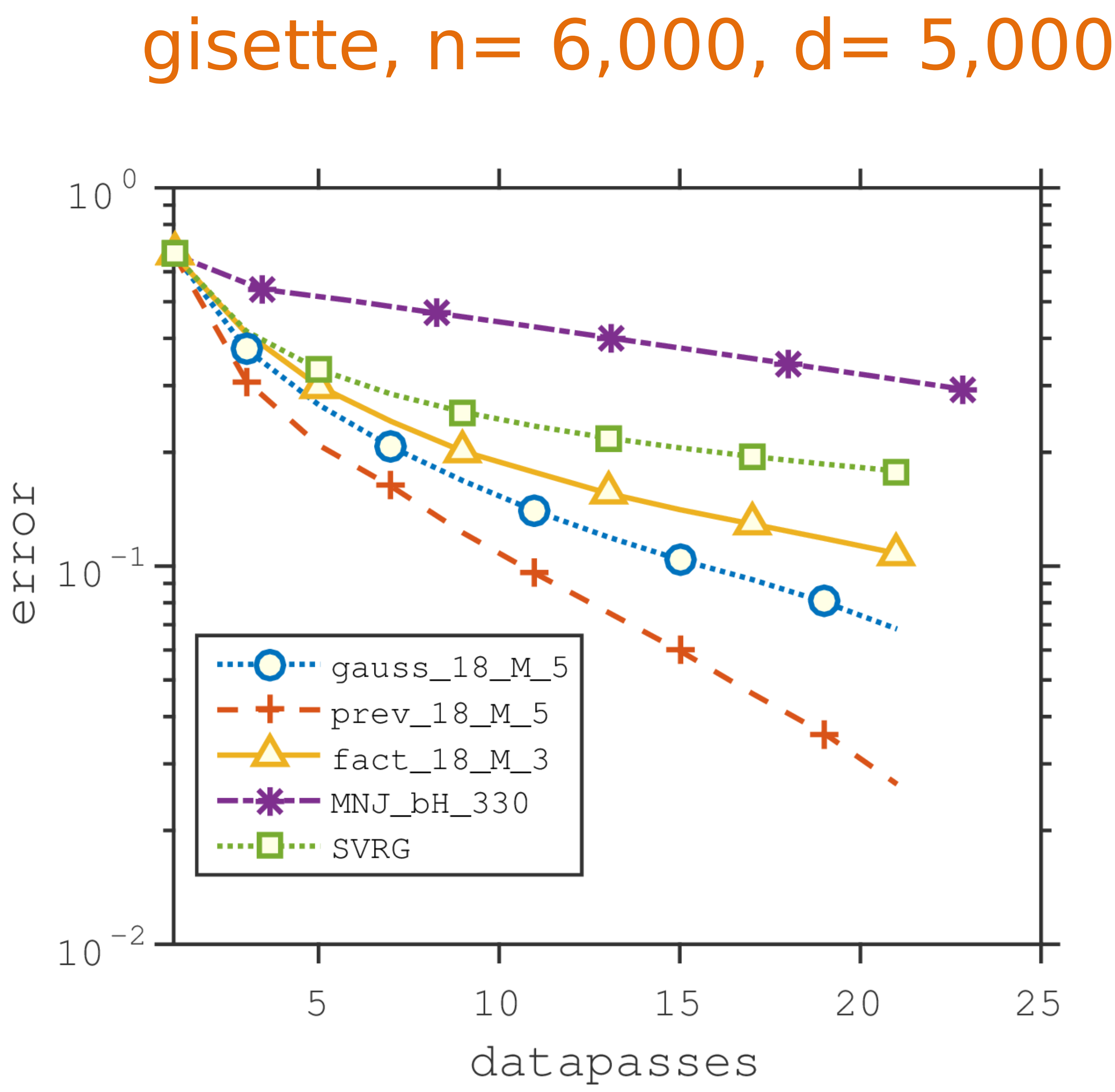
Key to methods

SVRG	Johnson Zhang (2013)
MNJ	Moritz, Nishihara Jordan (2013)
gauss_q_M	Gaussian elements
fact_q_M	Self-conditioning factorized sampling
prev_q_M	Previous search directions delayed

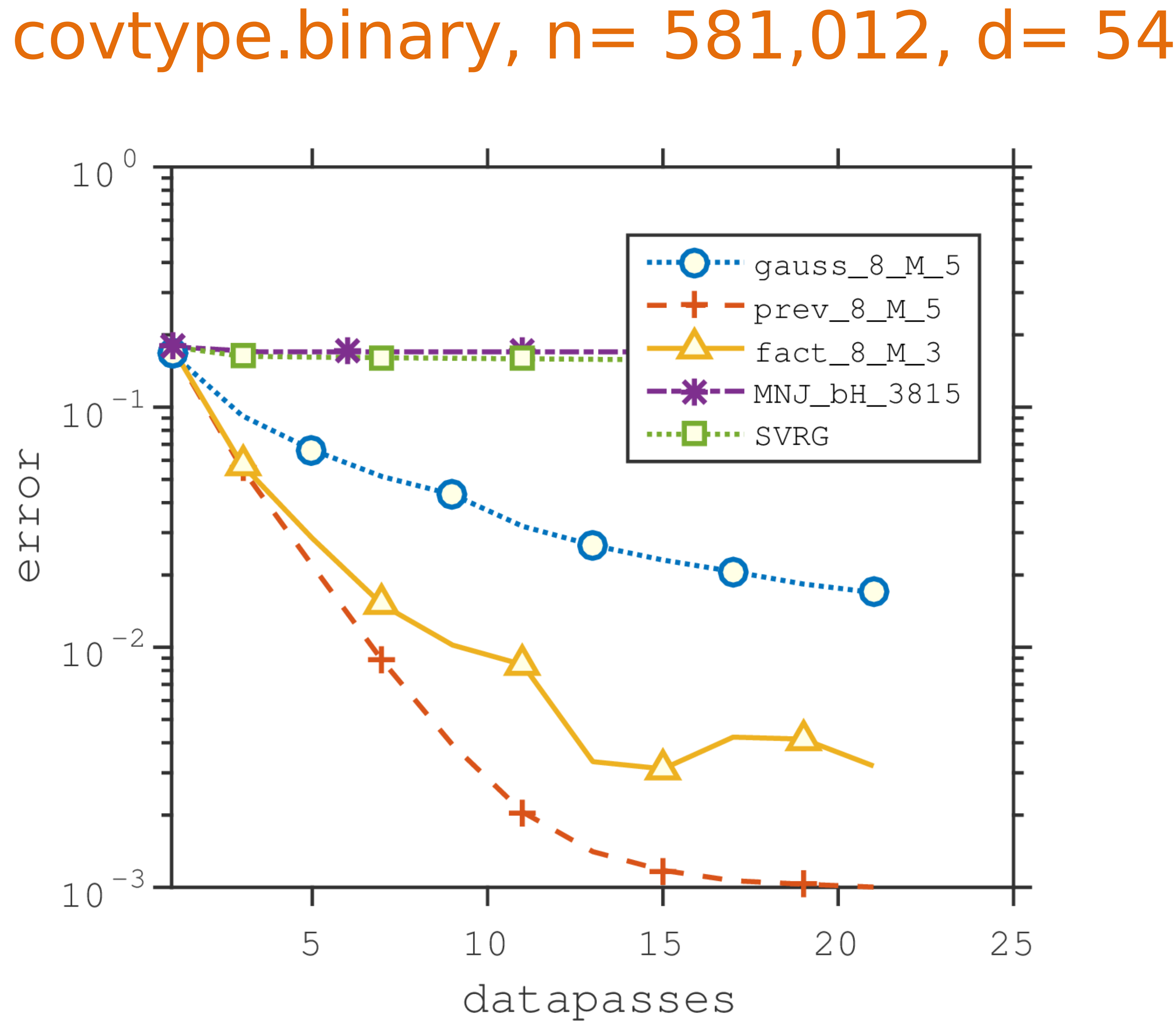
Stochastic Block BFGS Methods

$D_t \in \mathbf{R}^{n\times q}$

$M \in \mathbf{N}$ number of block triples stored

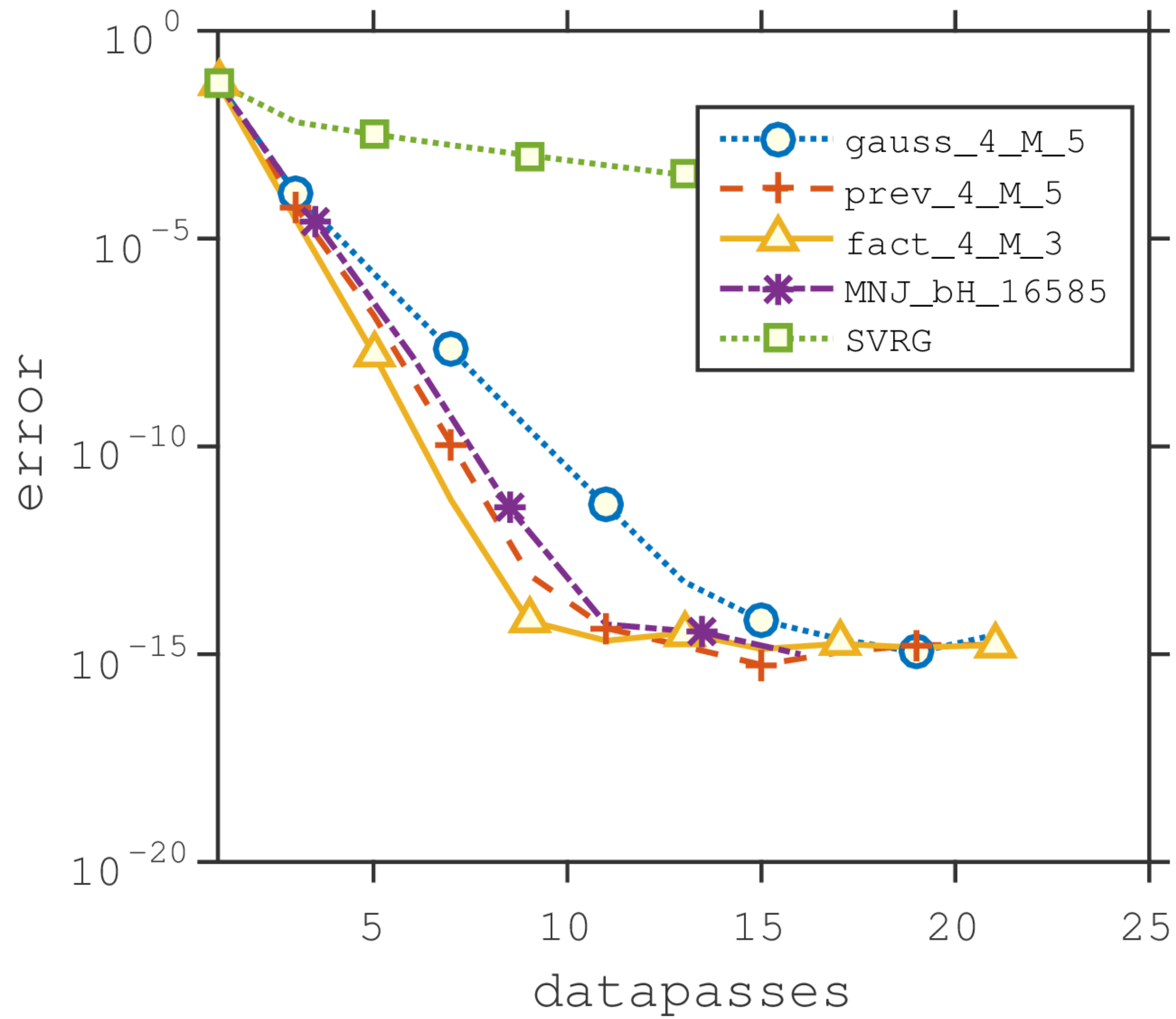


gisette, n= 6,000, d= 5,000
10^0
10^{-1}
10^{-2}
error
5
10
15
20
25
datapasses
gauss_18_M_5
prev_18_M_5
fact_18_M_3
MNJ_bH_330
SVRG

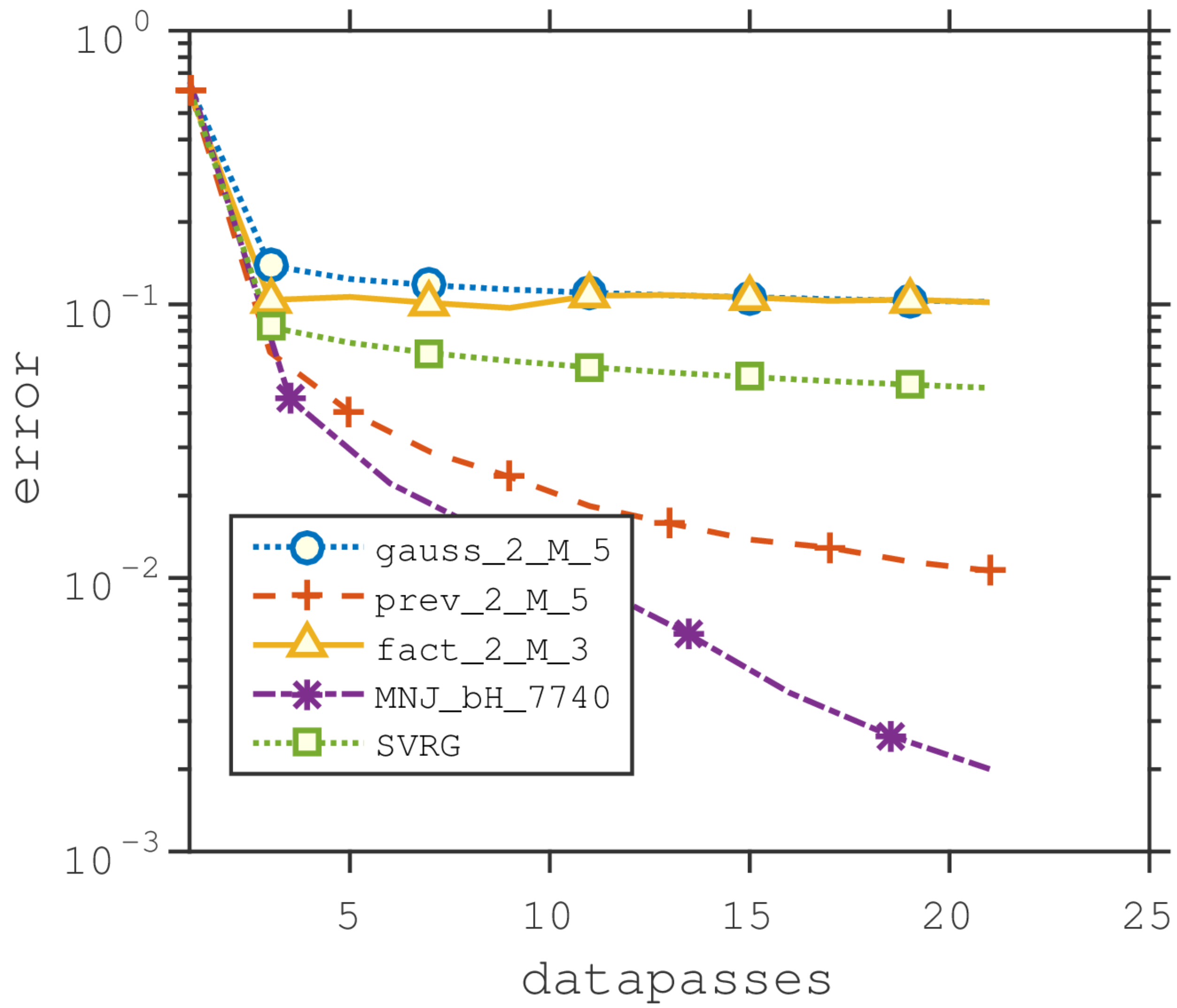


covtype.binary, n= 581,012, d= 54
error
datapasses
10^0
10^-1
10^-2
10^-3
5
10
15
20
25
gauss_8_M_5
prev_8_M_5
fact_8_M_3
MNJ_bH_3815
SVRG

Higgs, n=11,000,000 , d= 28



url-combined n = 2,396,130, d = 3,231,961



Conclusions

- New metric learning framework. A block BFGS framework for gradually learning the metric of the underlying function using sketches of subsampled Hessian matrices
- New limited memory block BFGS method. May also be of interest for non-stochastic optimization
- Several matrix sketching possibilities.
- More reasonable bounds on eigenvalues of H_k which lead to more reasonable conditions for step size, as compared to MNJ

More Numerics

Convergence results

R. Johnson and T. Zhang (2013). **Accelerating stochastic gradient descent using predictive variance reduction**. NIPS.

P. Moritz, R. Nishihara and M. I. Jordan (2016). **A Linearly-Convergent Stochastic L-BFGS Algorithm**, AISTATS.

RMG and Peter Richtárik (2016) **Randomized Quasi-Newton Updates are Linearly Convergent Matrix Inversion Algorithms** arXiv:1602.01768

Convergence

Convergence

Assumption

There exists $0 < \lambda \leq \Lambda$ such that

$$\lambda I \preceq \nabla^2 f_T(x) \preceq \Lambda I$$

For all $x \in \mathbf{R}^d$ and all $T \subseteq [n]$.

Lemma [GGR'16]

There exists $0 < \gamma \leq \Gamma$ such that

$$\gamma I \preceq H_t \preceq \Gamma I, \qquad \forall t$$

Furthermore

$$\frac{1}{1 + M\Lambda} \leq \gamma \leq \Gamma \leq (1 + \sqrt{\kappa})^{2M}(1 + \frac{1}{\lambda(2\sqrt{\kappa} + \kappa)})$$

where $\kappa = \Lambda/\lambda$

Complexity / Convergence

Theorem [GGR‘16]

If

$$m \geq \frac{1}{2\eta\left(\gamma\lambda - \eta\Gamma^2\Lambda(2\Lambda - \lambda)\right)} \qquad \eta < \gamma\lambda/(2\Gamma^2\Lambda^2)$$

Complexity / Convergence

Theorem [GGR'16]

If

$$m \geq \frac{1}{2\eta\left(\gamma\lambda - \eta\Gamma^2\Lambda(2\Lambda - \lambda)\right)} \qquad \eta < \gamma\lambda/(2\Gamma^2\Lambda^2)$$

Complexity / Convergence

Theorem [GGR'16]

If

$$m \geq \frac{1}{2\eta\left(\gamma\lambda - \eta\Gamma^2\Lambda(2\Lambda - \lambda)\right)} \qquad \eta < \gamma\lambda/(2\Gamma^2\Lambda^2)$$

Inner iterations of SVRG

Complexity / Convergence

Theorem [GGR'16]

Step size

If

$$m \geq \frac{1}{2\eta\left(\gamma\lambda - \eta\Gamma^2\Lambda(2\Lambda - \lambda)\right)} \qquad \eta < \gamma\lambda/(2\Gamma^2\Lambda^2)$$

Inner iterations of SVRG

Complexity / Convergence

Theorem [GGR'16]

If

$$m \geq \frac{1}{2\eta\left(\gamma\lambda - \eta\Gamma^2\Lambda(2\Lambda - \lambda)\right)} \qquad \eta < \gamma\lambda/(2\Gamma^2\Lambda^2)$$

Inner iterations of SVRG

Step size

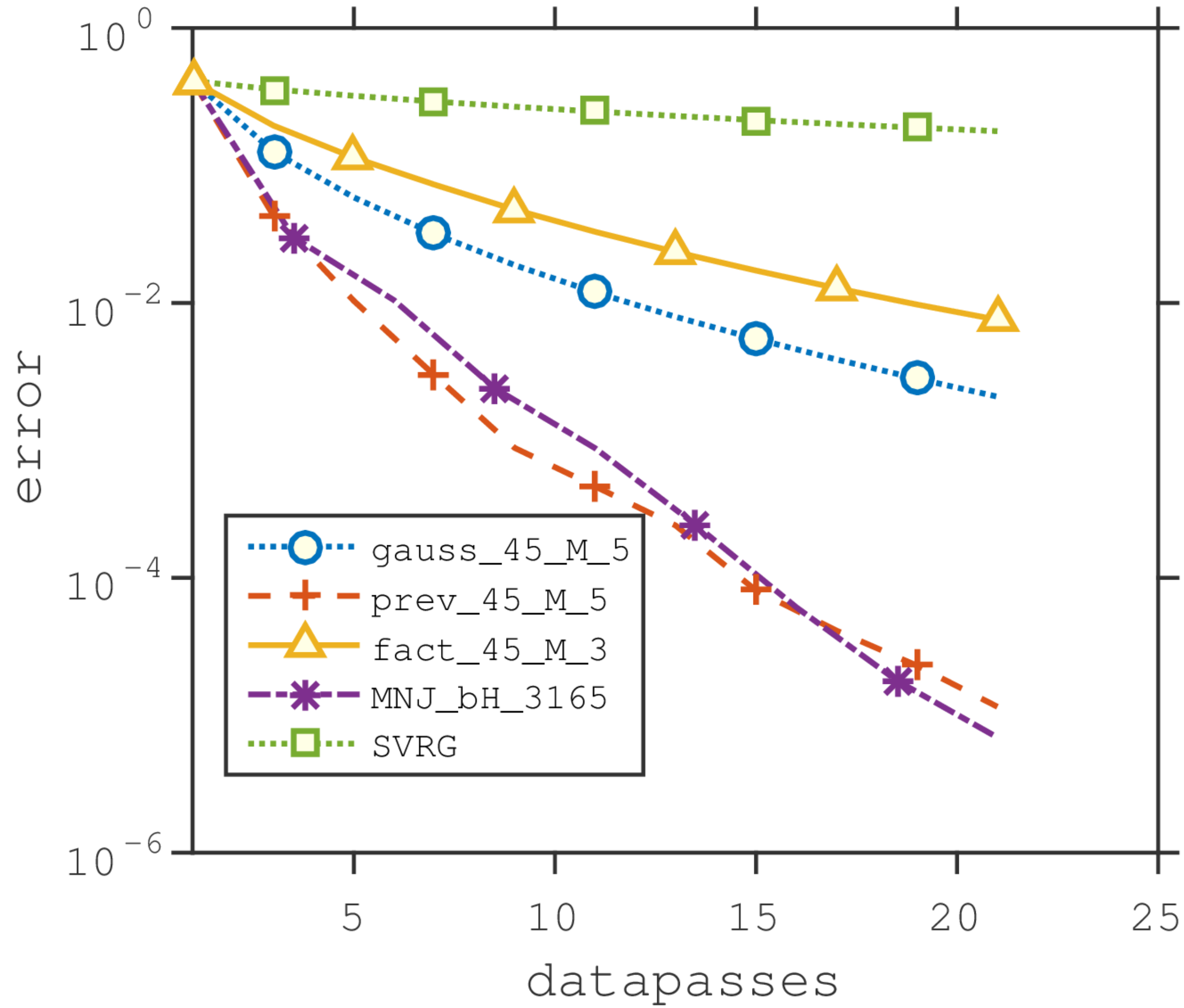
$$\mathbf{E}[f(w_t) - f(w_*)] \leq \rho^t(f(w_0) - f(w_*))$$

Where,

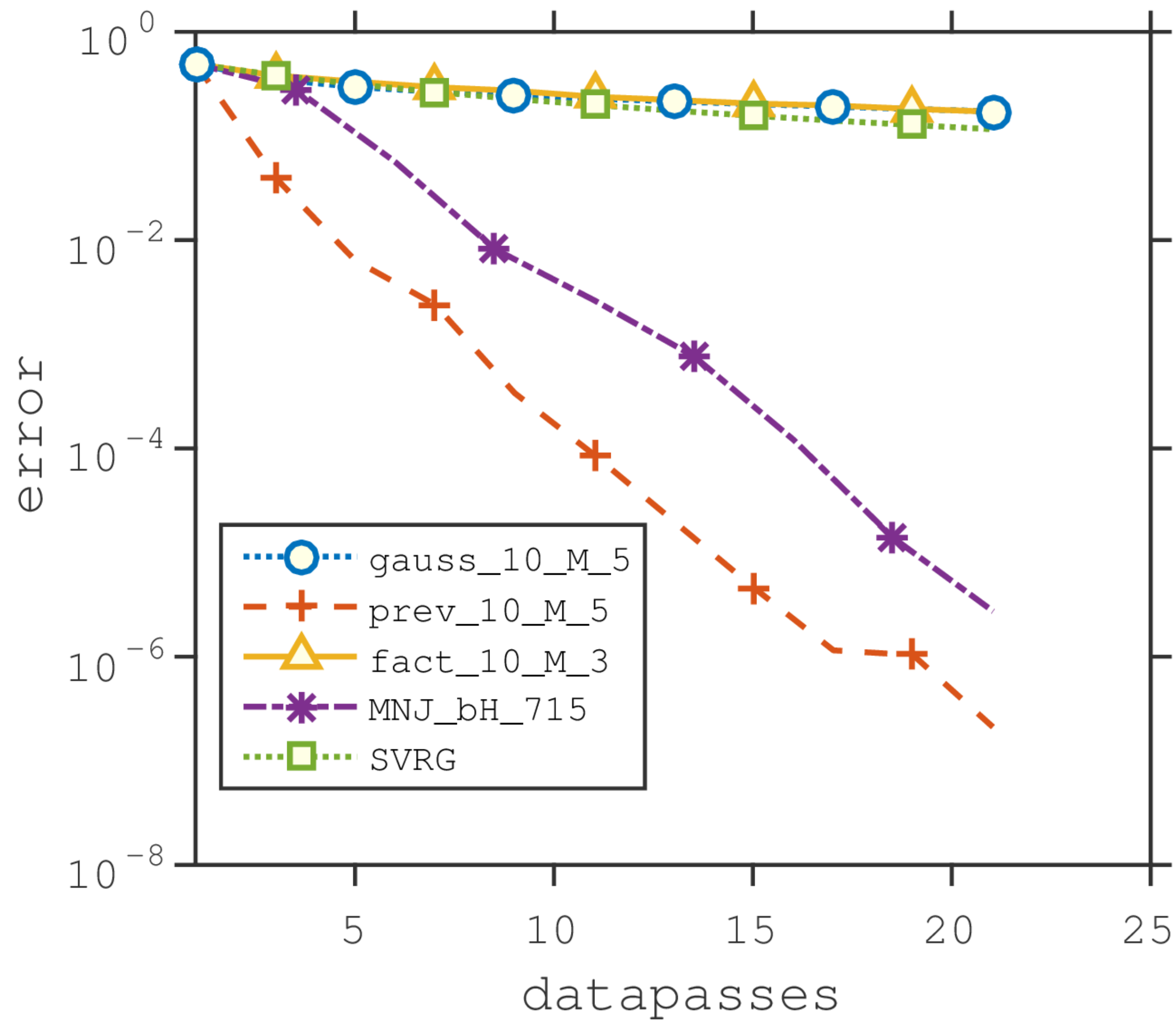
$$\rho = \frac{1/2m\eta + \eta\Gamma^2\Lambda(\Lambda - \lambda)}{\gamma\lambda - \eta\Gamma^2\Lambda^2} < 1$$

Experimental results
error X datapasses

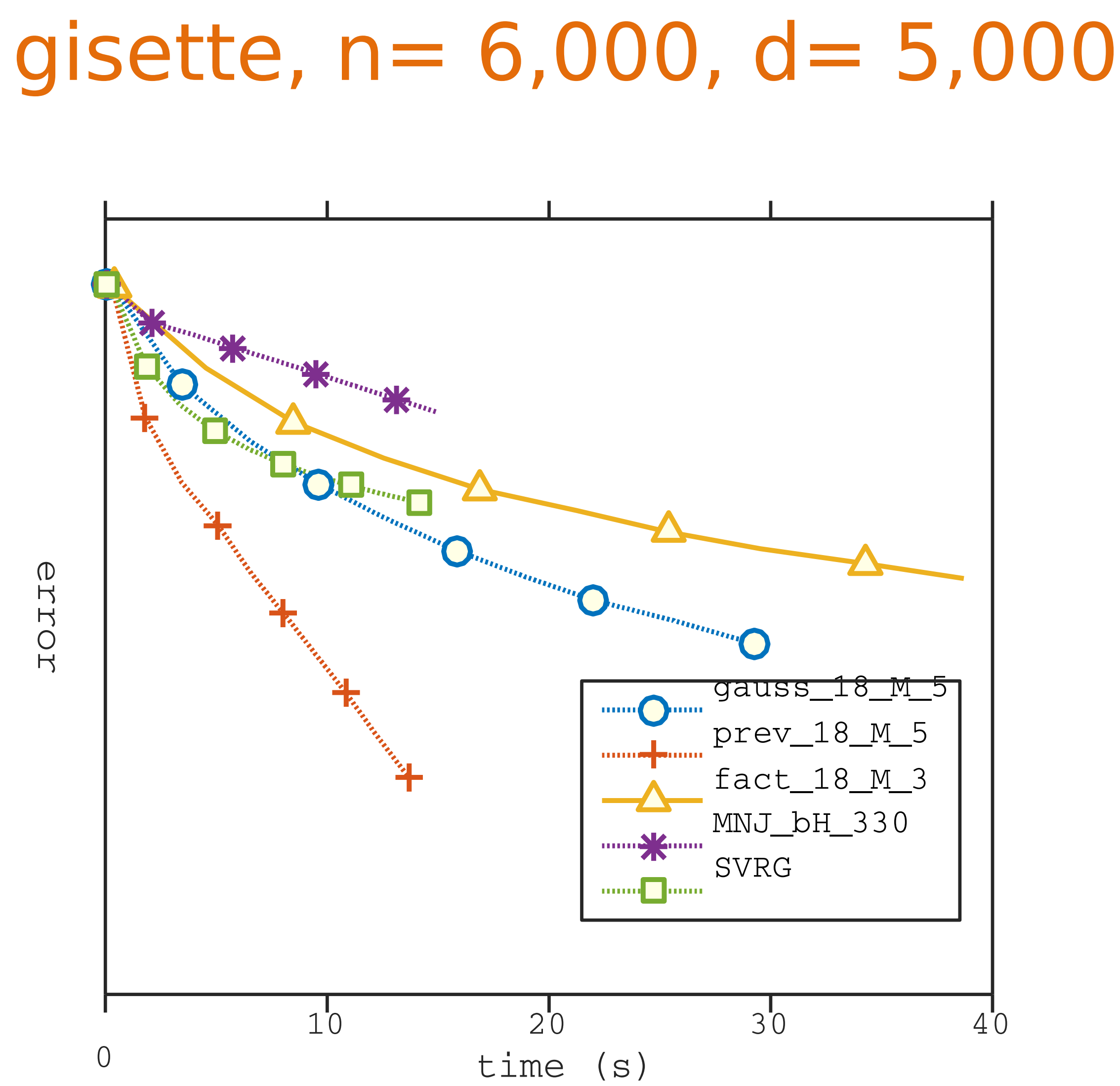
epsilon_normalized n= 400,000 , d=2,000

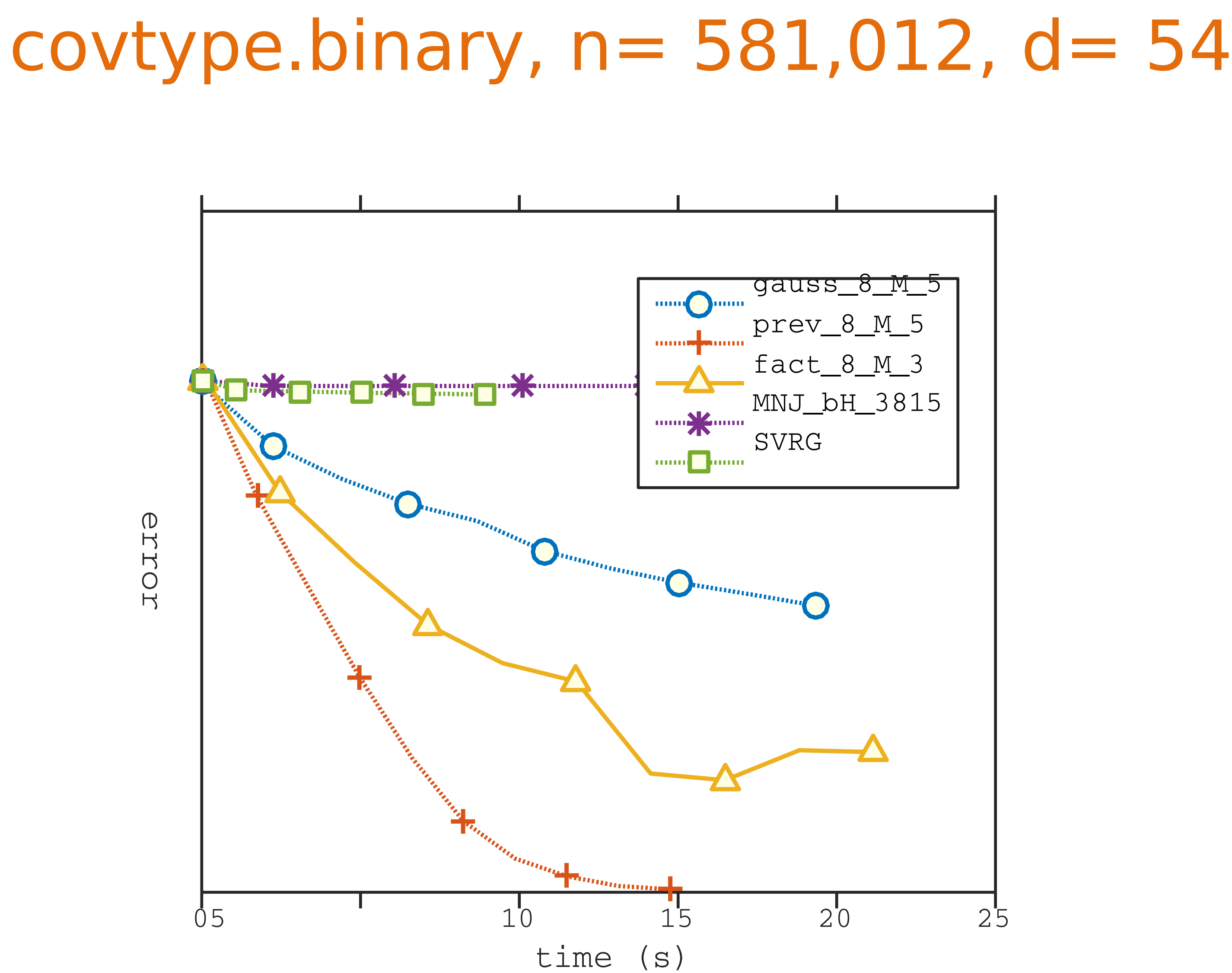


rcv1-training n = 20,242, d = 47,236



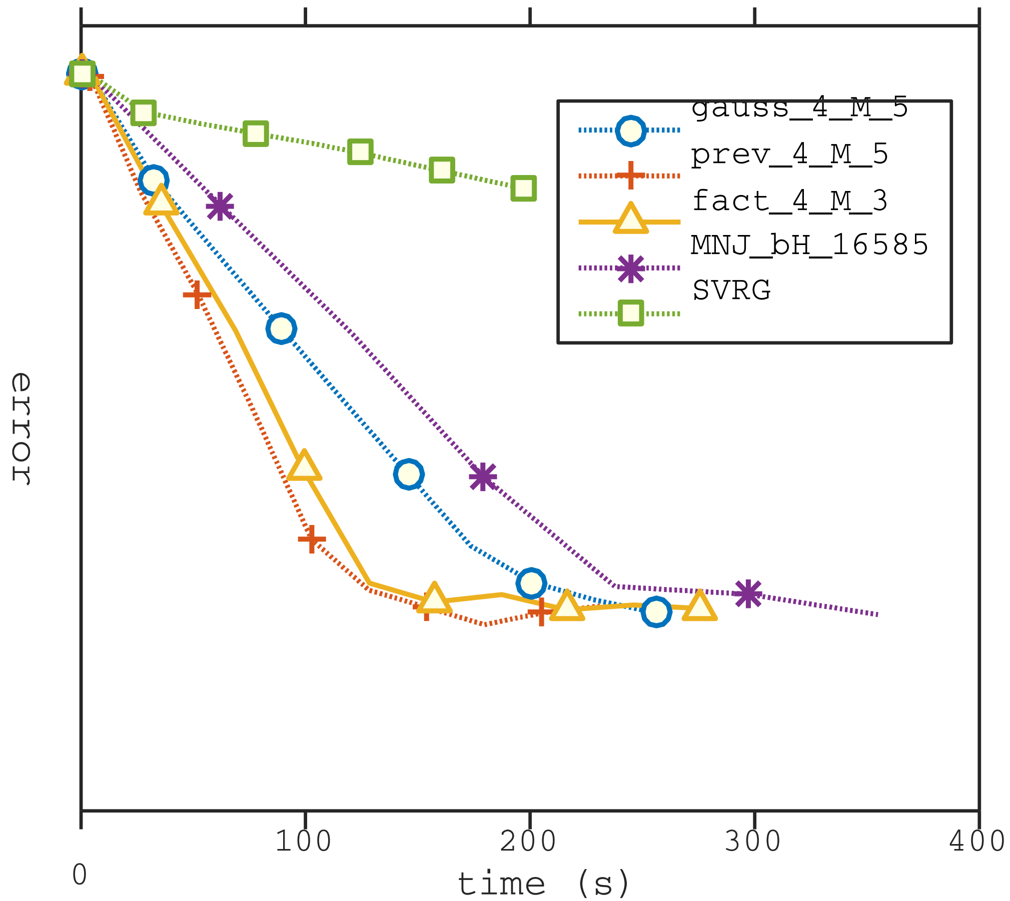
Experimental results
error X time


gisette, n= 6,000, d= 5,000
error
time (s)
0
10
20
30
40
gauss_18_M_5
prev_18_M_5
fact_18_M_3
MNJ_bH_330
SVRG

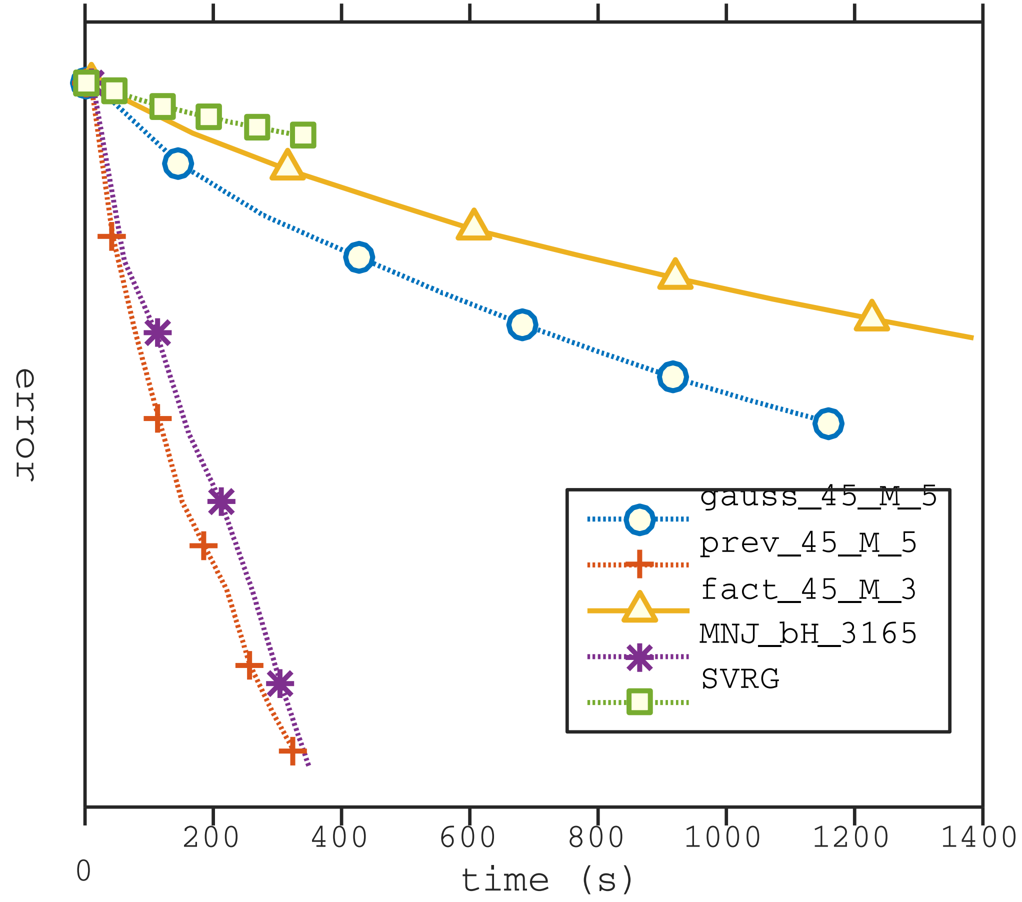


covtype.binary, n= 581,012, d= 54
gauss_8_M_5
prev_8_M_5
fact_8_M_3
MNJ_bH_3815
SVRG
error
0
5
10
15
20
25
time (s)

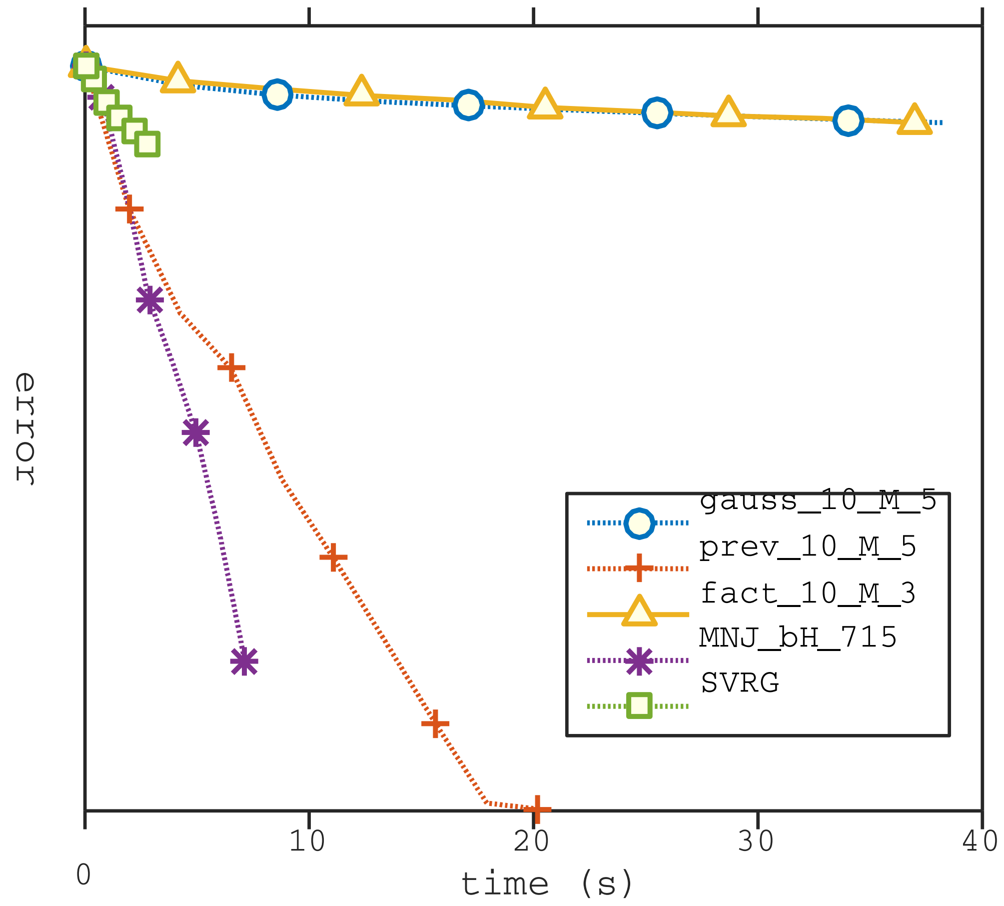
Higgs, n=11,000,000 , d= 28



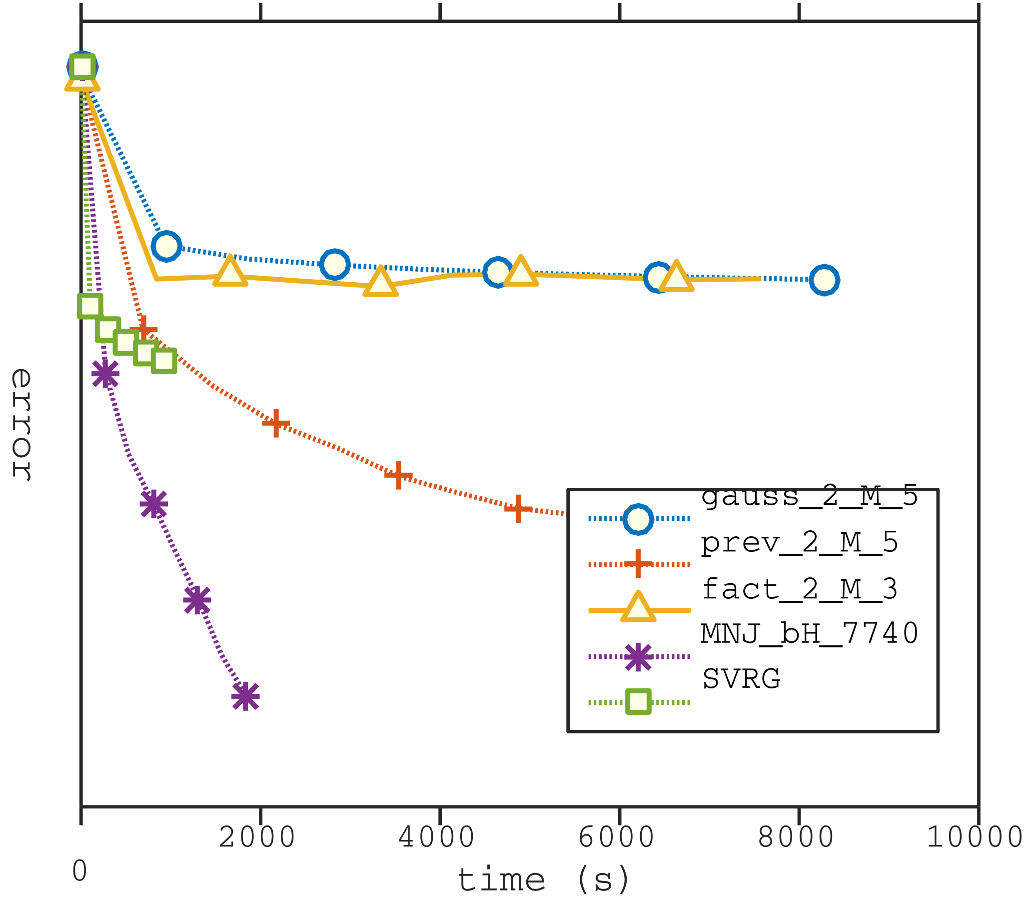
epsilon_normalized n= 400,000 , d=2,000



rcv1-training n = 20,242, d = 47,236

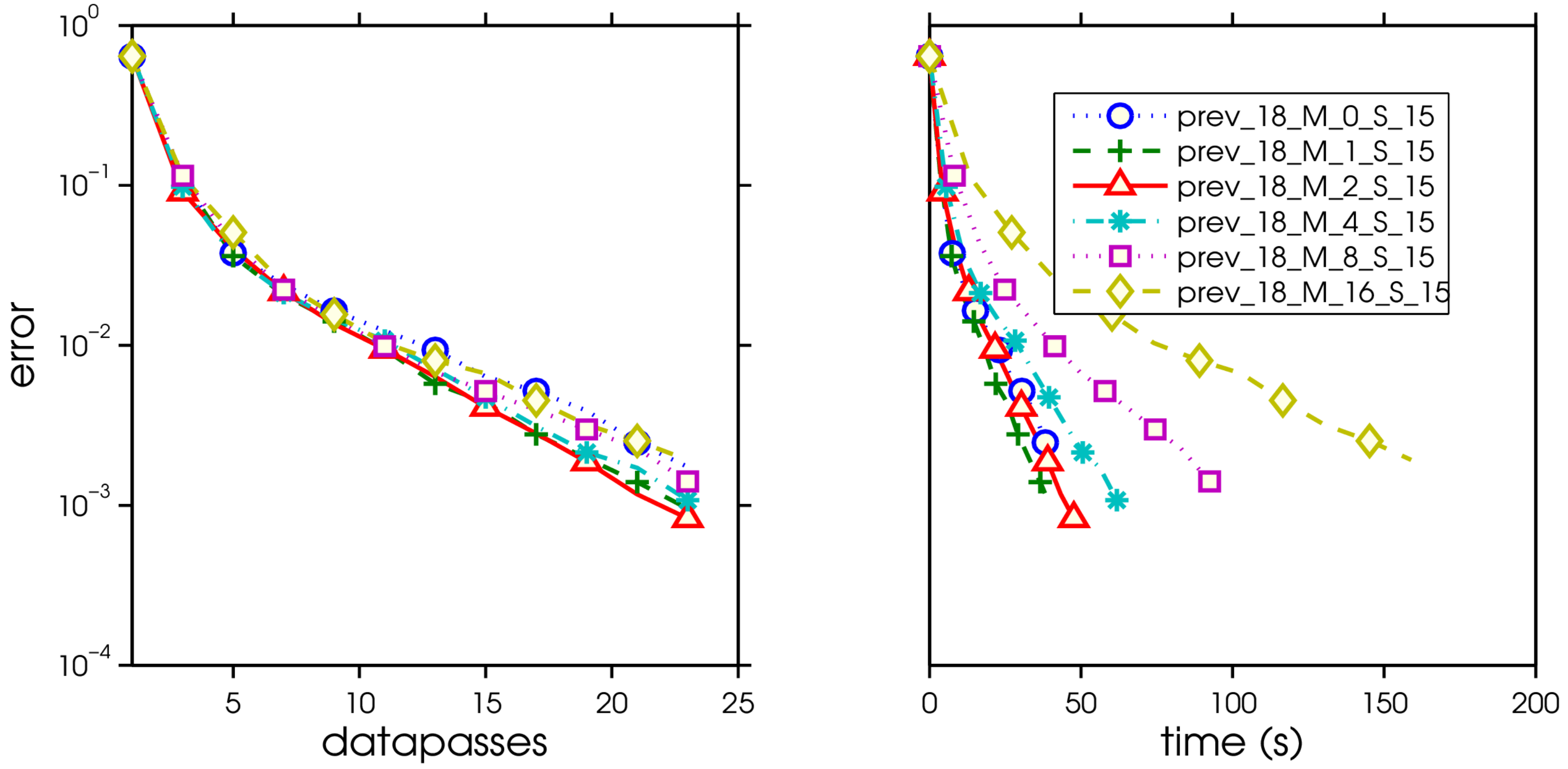


url-combined n = 2,396,130, d = 3,231,961



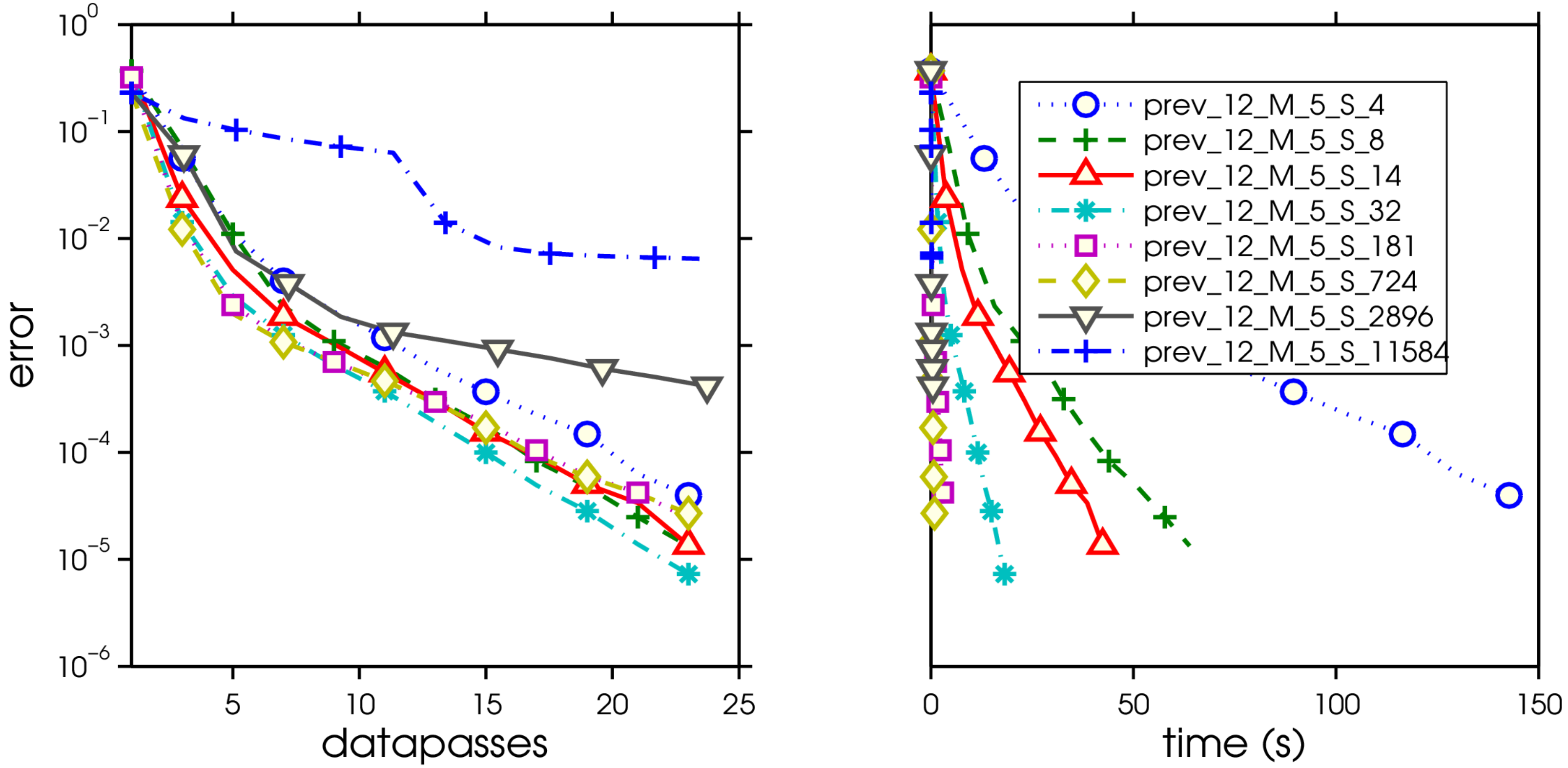
Experimental results parameter exploration

w8a n = 49,749, d = 300



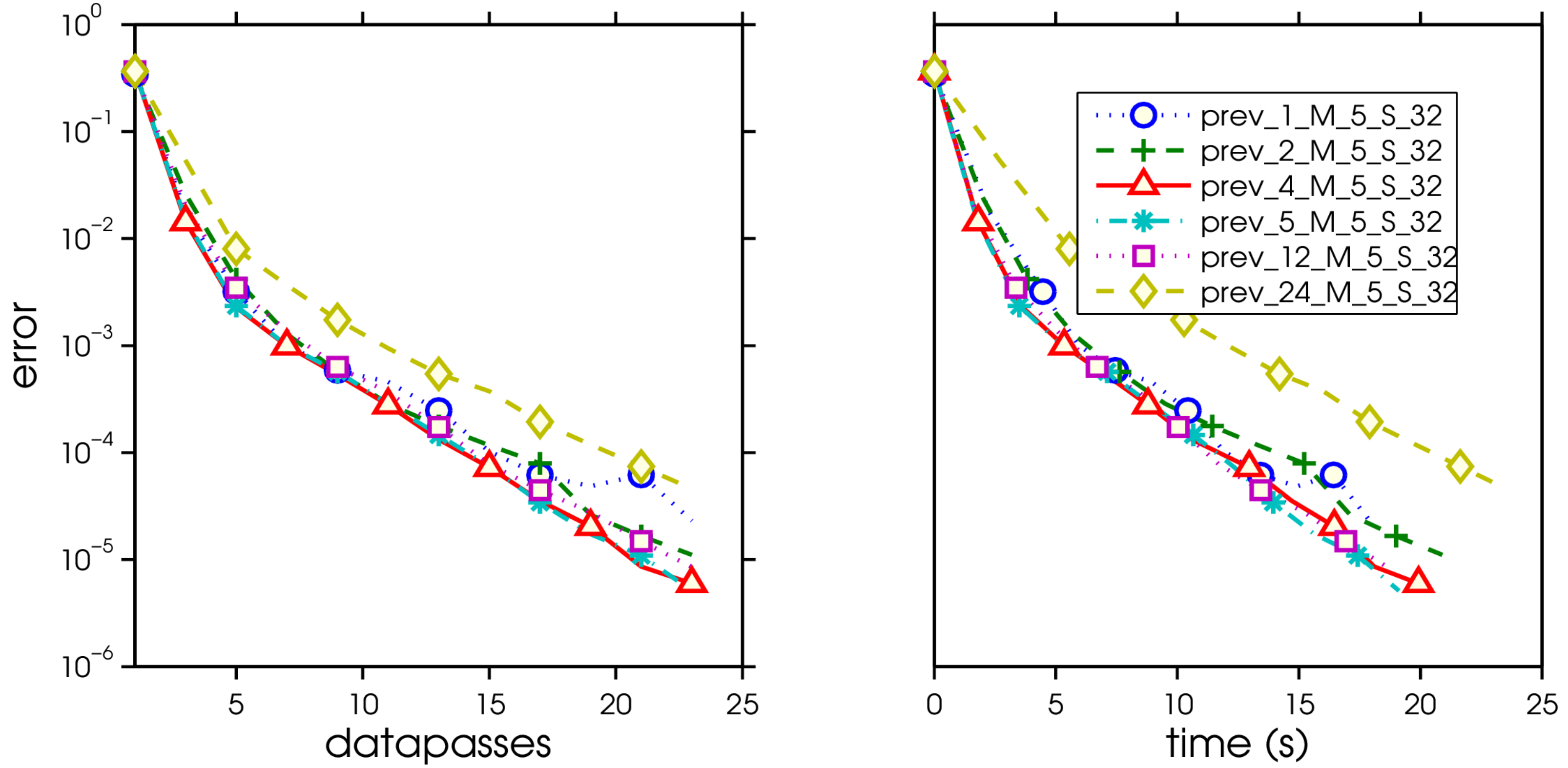
Testing the memory parameter M

a9a n = 32,561, d = 123



Testing the subsampling size with S=T

a9a n = 32,561, d = 123



Testing update size (delay size) q

SVRG

The Stochastic Variance Reduced Gradient

$$g_t = \nabla f_S(w_t) - \nabla f_S(x_k) + \nabla f(x_k)$$

Where x_k is a reference point.

$$\textbf{Unbiased : } \mathbf{E}_S[g_t] = \mathbf{E}[\nabla f_S(w_t)] - \mathbf{E}[\nabla f_S(x_k)] + \nabla f(x_k)$$
$$= \nabla f(w_t) + \nabla f(x_k) - \nabla f(x_k)$$
$$= \nabla f(w_t)$$

Maintain x_k fixed and
iterate in t for m iterations

R. Johnson and T. Zhang (2013). **Accelerating stochastic gradient descent using predictive variance reduction**. NIPS, 1(3), 1–9.