SARAH Algorithm

Lam M. Nguyen, Jie Liu, Katya Scheinberg, Martin Takáč

INFORMS Annual Meeting
October 24, 2017

Problem Description

Optimize a finite sum with large number of elements n

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

Problem Description

Optimize a finite sum with large number of elements n

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

Training set: $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

f_i- **strongly convex**: linear regression, binary classification

ℓ_2-regularized least squares regression: $f_i(w) = \left(x_i^T w - y_i\right)^2 + \frac{\lambda}{2}\|w\|^2$

ℓ_2-regularized logistic regression: $f_i(w) = \log(1 + \exp(-y_i x_i^T w)) + \frac{\lambda}{2}\|w\|^2$

f_i- **nonconvex**: neural networks

Problem Description

Optimize a finite sum with large number of elements n

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

Training set: $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

f_i- **strongly convex**: linear regression, binary classification

ℓ_2-regularized least squares regression: $f_i(w) = \left(x_i^T w - y_i\right)^2 + \frac{\lambda}{2}\|w\|^2$

ℓ_2-regularized logistic regression: $f_i(w) = \log(1 + \exp(-y_i x_i^T w)) + \frac{\lambda}{2}\|w\|^2$

f_i- **nonconvex**: neural networks

HOW WE COULD SOLVE THIS OPTIMIZATION PROBLEM?

Gradient Methods

Optimize a finite sum with large number of elements n

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

Gradient Methods

Optimize a finite sum with large number of elements n

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

P is L-smooth and μ-strongly convex

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Goal: achieve ϵ-accurate solution w_T such that

- $P(w_T) - P(w^*) \leq \epsilon$
- $\|w_T - w^*\|^2 \leq \epsilon$
- $\|\nabla P(w_T)\|^2 \leq \epsilon$

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Goal: achieve ϵ-accurate solution w_T such that

- $P(w_T) - P(w^*) \leq \epsilon$
- $\|w_T - w^*\|^2 \leq \epsilon$
- $\|\nabla P(w_T)\|^2 \leq \epsilon$

Gradient Descent: $w_{t+1} = w_t - \eta \nabla P(w_t)$

Newton Method: $w_{t+1} = w_t - [\mathbf{H}\, \mathrm{P}(\mathrm{w_t})]^{-1} \nabla P(w_t)$

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Goal: achieve ϵ-accurate solution w_T such that

- $P(w_T) - P(w^*) \leq \epsilon$
- $\|w_T - w^*\|^2 \leq \epsilon$
- $\|\nabla P(w_T)\|^2 \leq \epsilon$

Gradient Descent: $w_{t+1} = w_t - \eta \nabla P(w_t)$

Newton Method: $w_{t+1} = w_t - [\mathbf{H}\, \mathrm{P}(\mathrm{w_t})]^{-1} \nabla P(w_t)$

The total work complexity: Number of component gradient evaluations

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Goal: achieve ϵ-accurate solution w_T such that

- $P(w_T) - P(w^*) \leq \epsilon$
- $\|w_T - w^*\|^2 \leq \epsilon$
- $\|\nabla P(w_T)\|^2 \leq \epsilon$

Need "*n*" work per iteration

Gradient Descent: $w_{t+1} = w_t - \eta \nabla P(w_t)$

Newton Method: $w_{t+1} = w_t - [\mathbf{H}\, \mathrm{P}(\mathrm{w_t})]^{-1} \nabla P(w_t)$

The total work complexity: Number of component gradient evaluations

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Goal: achieve ϵ-accurate solution w_T such that

- $P(w_T) - P(w^*) \leq \epsilon$
- $\|w_T - w^*\|^2 \leq \epsilon$
- $\|\nabla P(w_T)\|^2 \leq \epsilon$

Gradient Descent: $w_{t+1} = w_t - \eta \nabla P(w_t)$

Newton Method: $w_{t+1} = w_t - [\mathbf{H}\, \mathrm{P}(\mathrm{w_t})]^{-1} \nabla P(w_t)$

Need "n" work per iteration

Need "n^2" work per iteration

The total work complexity: Number of component gradient evaluations

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Goal: achieve ϵ-accurate solution w_T such that

- $P(w_T) - P(w^*) \leq \epsilon$
- $\|w_T - w^*\|^2 \leq \epsilon$
- $\|\nabla P(w_T)\|^2 \leq \epsilon$

Gradient Descent: $w_{t+1} = w_t - \eta \nabla P(w_t)$

Newton Method: $w_{t+1} = w_t - [\mathbf{H}\, \mathrm{P}(\mathrm{w_t})]^{-1} \nabla P(w_t)$

Need "n" work per iteration

Need "n^2" work per iteration

The total work complexity: Number of component gradient evaluations

Machine learning and Big Data applications $\Longrightarrow \boldsymbol{n \gg 1}$ ("n" is very large)

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Goal: achieve ϵ-accurate solution w_T such that

- $P(w_T) - P(w^*) \leq \epsilon$
- $\|w_T - w^*\|^2 \leq \epsilon$
- $\|\nabla P(w_T)\|^2 \leq \epsilon$

Gradient Descent: $w_{t+1} = w_t - \eta \nabla P(w_t)$

Newton Method: $w_{t+1} = w_t - [\mathbf{H}\, \mathrm{P}(\mathrm{w_t})]^{-1} \nabla P(w_t)$

Need "n" work per iteration

Need "n^2" work per iteration

The total work complexity: Number of component gradient evaluations

Machine learning and Big Data applications $\Longrightarrow \boldsymbol{n \gg 1}$ ("n" is very large)

TOO MUCH WORK !!!

Gradient Methods

Optimize a finite sum with large number of elements n

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

***P* is *L*-smooth and μ-strongly convex**

Iterative methods (using gradient)

Given initial point w_0. Update: $w_{t+1} = w_t - \eta_t v_t \ , t = 0,1,2, \ldots$

Goal: achieve ϵ-accurate solution w_T such that

- $P(w_T) - P(w^*) \leq \epsilon$
- $\|w_T - w^*\|^2 \leq \epsilon$
- $\|\nabla P(w_T)\|^2 \leq \epsilon$

Gradient Descent: $w_{t+1} = w_t - \eta \nabla P(w_t)$

Newton Method: $w_{t+1} = w_t - [\mathbf{H}\, \mathrm{P}(\mathrm{w_t})]^{-1} \nabla P(w_t)$

Need "n" work per iteration

Need "n^2" work per iteration

The total work complexity: Number of component gradient evaluations

Machine learning and Big Data applications $\Longrightarrow \boldsymbol{n \gg 1}$ ("n" is very large)

TOO MUCH WORK !!!

Computing "less" work per iteration

Stochastic Gradient Descent (SGD)

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

Need only "1" work per iteration

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951

Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

Need only "1" work per iteration

To guarantee convergence:

$$\sum_{t=0}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

Need only "1" work per iteration

To guarantee convergence:

$$\sum_{t=0}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

If $\eta_t = \frac{d}{\gamma + t}$ then $\mathbb{E}[P(w_t) - P(w^*)] \leq \frac{c}{\gamma + t}$

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

Need only "1" work per iteration

To guarantee convergence:

$$\sum_{t=0}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

If $\eta_t = \frac{d}{\gamma + t}$ then $\mathbb{E}[P(w_t) - P(w^*)] \leq \frac{c}{\gamma + t}$

=> Require $O\left(\frac{1}{\epsilon}\right)$ total work to achieve ϵ-accurate solution

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

Need only "1" work per iteration

To guarantee convergence:

$$\sum_{t=0}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

If $\eta_t = \frac{d}{\gamma + t}$ then $\mathbb{E}[P(w_t) - P(w^*)] \leq \frac{c}{\gamma + t}$

=> Require $O\left(\frac{1}{\epsilon}\right)$ total work to achieve ϵ-accurate solution

Pros:

- Each iteration is independent on "n"

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

Need only "1" work per iteration

To guarantee convergence:

$$\sum_{t=0}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

If $\eta_t = \frac{d}{\gamma + t}$ then $\mathbb{E}[P(w_t) - P(w^*)] \leq \frac{c}{\gamma + t}$

=> Require $O\left(\frac{1}{\epsilon}\right)$ total work to achieve ϵ-accurate solution

Pros:

- Each iteration is independent on "n"

Cons:

- **Sublinear** convergence rate

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

Need only "1" work per iteration

To guarantee convergence:

$$\sum_{t=0}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

If $\eta_t = \frac{d}{\gamma + t}$ then $\mathbb{E}[P(w_t) - P(w^*)] \leq \frac{c}{\gamma + t}$

=> Require $O\left(\frac{1}{\epsilon}\right)$ total work to achieve ϵ-accurate solution

Pros:

- Each iteration is independent on "n"

Cons:

- **Sublinear** convergence rate

Can we get a linear convergence rate?

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (**SGD**) [H. Robbins & S. Monro, 1951]

1. Choose initial point w_0
2. For $t = 0,1,2, \ldots$
3. $w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t),\ i_t \in \{1, \ldots, n\}$

Need only "1" work per iteration

To guarantee convergence:

$$\sum_{t=0}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=0}^{\infty} \eta_t^2 < \infty$$

If $\eta_t = \frac{d}{\gamma + t}$ then $\mathbb{E}[P(w_t) - P(w^*)] \leq \frac{c}{\gamma + t}$

=> Require $O\left(\frac{1}{\epsilon}\right)$ total work to achieve ϵ-accurate solution

Pros:

- Each iteration is independent on "n"

Cons:

- **Sublinear** convergence rate

Can we get a linear convergence rate?

Modifying stochastic gradient

H. Robbins and S. Monro. **A Stochastic Approximation Method**, 1951
Léon Bottou, Frank E Curtis, Jorge Nocedal. **Optimization methods for large-scale machine learning**, 2016

SAG/SAGA

M. Schmidt, N. Le Roux, F. Bach. **Minimizing Finite Sums with the Stochastic Average Gradient**, 2013
A. Defazio, F. Bach, S. Lacoste-Julien. **SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives**, 2014

SAG/SAGA

SAG [M. Schmidt et. al., 2013] and **SAGA** [A. Defazio et. al., 2014]

- Compute the full gradient at the initial point
- Keep a table of "past" gradients
- In each iteration, update one "gradient" in the table

M. Schmidt, N. Le Roux, F. Bach. **Minimizing Finite Sums with the Stochastic Average Gradient**, 2013
A. Defazio, F. Bach, S. Lacoste-Julien. **SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives**, 2014

SAG/SAGA

SAG [M. Schmidt et. al., 2013] and **SAGA** [A. Defazio et. al., 2014]

- Compute the full gradient at the initial point
- Keep a table of "past" gradients
- In each iteration, update one "gradient" in the table

(SAG) $$w_{t+1} = w_t - \eta_t \cdot \frac{1}{n}\sum_{i=1}^{n} y_{i,t}$$ ← $$y_{i,t} = \begin{cases} \nabla f_i(w_t), & if\ i_t = i \\ y_{i,t-1}, & otherwise \end{cases}$$

(SAGA) $$w_{t+1} = w_t - \eta\left(\nabla f_{i_t}(w_t) - y_{i_t,t-1} + \frac{1}{n}\sum_{i=1}^{n} y_{i,t-1}\right)$$

M. Schmidt, N. Le Roux, F. Bach. **Minimizing Finite Sums with the Stochastic Average Gradient**, 2013
A. Defazio, F. Bach, S. Lacoste-Julien. **SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives**, 2014

SAG/SAGA

SAG [M. Schmidt et. al., 2013] and **SAGA** [A. Defazio et. al., 2014]

- Compute the full gradient at the initial point
- Keep a table of "past" gradients
- In each iteration, update one "gradient" in the table

(SAG) $$w_{t+1} = w_t - \eta_t \cdot \frac{1}{n}\sum_{i=1}^{n} y_{i,t}$$ ← $$y_{i,t} = \begin{cases} \nabla f_i(w_t), & if\ i_t = i \\ y_{i,t-1}, & otherwise \end{cases}$$

(SAGA) $$w_{t+1} = w_t - \eta\left(\nabla f_{i_t}(w_t) - y_{i_t,t-1} + \frac{1}{n}\sum_{i=1}^{n} y_{i,t-1}\right)$$

Pros:

- **Linear** convergence rate

M. Schmidt, N. Le Roux, F. Bach. **Minimizing Finite Sums with the Stochastic Average Gradient**, 2013
A. Defazio, F. Bach, S. Lacoste-Julien. **SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives**, 2014

SAG/SAGA

SAG [M. Schmidt et. al., 2013] and **SAGA** [A. Defazio et. al., 2014]

- Compute the full gradient at the initial point
- Keep a table of "past" gradients
- In each iteration, update one "gradient" in the table

(SAG) $$w_{t+1} = w_t - \eta_t \cdot \frac{1}{n}\sum_{i=1}^{n} y_{i,t}$$ ← $$y_{i,t} = \begin{cases} \nabla f_i(w_t), & if\ i_t = i \\ y_{i,t-1}, & otherwise \end{cases}$$

(SAGA) $$w_{t+1} = w_t - \eta\left(\nabla f_{i_t}(w_t) - y_{i_t,t-1} + \frac{1}{n}\sum_{i=1}^{n} y_{i,t-1}\right)$$

Pros:

- **Linear** convergence rate

Cons:

- **Extra storage!** Need to store "***n***" gradients

M. Schmidt, N. Le Roux, F. Bach. **Minimizing Finite Sums with the Stochastic Average Gradient**, 2013
A. Defazio, F. Bach, S. Lacoste-Julien. **SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives**, 2014

SAG/SAGA

SAG [M. Schmidt et. al., 2013] and **SAGA** [A. Defazio et. al., 2014]

- Compute the full gradient at the initial point
- Keep a table of "past" gradients
- In each iteration, update one "gradient" in the table

(SAG) $$w_{t+1} = w_t - \eta_t \cdot \frac{1}{n}\sum_{i=1}^{n} y_{i,t}$$ ← $$y_{i,t} = \begin{cases} \nabla f_i(w_t), & if\ i_t = i \\ y_{i,t-1}, & otherwise \end{cases}$$

(SAGA) $$w_{t+1} = w_t - \eta\left(\nabla f_{i_t}(w_t) - y_{i_t,t-1} + \frac{1}{n}\sum_{i=1}^{n} y_{i,t-1}\right)$$

Pros:

- **Linear** convergence rate

Cons:

- **Extra storage!** Need to store "***n***" gradients

Can we eliminate the extra storage and get a linear convergence rate?

M. Schmidt, N. Le Roux, F. Bach. **Minimizing Finite Sums with the Stochastic Average Gradient**, 2013
A. Defazio, F. Bach, S. Lacoste-Julien. **SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives**, 2014

Stochastic Variance Reduced Gradient (SVRG)

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

Stochastic Variance Reduced Gradient (SVRG)

SVRG [R. Johnson & T. Zhang, 2013]

- Modify stochastic gradient
  1. Choose initial point w_0
  2. Set $\widetilde{w} = w_0$
  3. For $t = 0,1,2, \dots, m$
  4. $$w_{t+1} = w_t - \eta \underbrace{\left(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}) + \nabla P(\widetilde{w}) \right)}_{v_t}, \; i_t \in \{1, \dots, n\}$$

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

Stochastic Variance Reduced Gradient (SVRG)

SVRG [R. Johnson & T. Zhang, 2013]

- Modify stochastic gradient
  1. Choose initial point w_0
  2. Set $\widetilde{w} = w_0$
  3. For $t = 0,1,2, \dots, m$
  4. $$w_{t+1} = w_t - \eta \underbrace{\left(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}) + \nabla P(\widetilde{w}) \right)}_{v_t}, \; i_t \in \{1, \dots, n\}$$

- Let $\widetilde{w}^+ \in \{w_0, w_1, \dots, w_{m-1}\}$
- Choose $\eta < \frac{1}{4L}$ and m such that

$$\alpha := \frac{1}{\mu\eta(1 - 2L\eta)m} + \frac{2L\eta}{1 - 2L\eta} < 1$$

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

Stochastic Variance Reduced Gradient (SVRG)

SVRG [R. Johnson & T. Zhang, 2013]

- Modify stochastic gradient
  1. Choose initial point w_0
  2. Set $\widetilde{w} = w_0$
  3. For $t = 0,1,2,\dots,m$
  4. $$w_{t+1} = w_t - \eta \underbrace{\left(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}) + \nabla P(\widetilde{w})\right)}_{v_t}, \; i_t \in \{1,\dots,n\}$$

- Let $\widetilde{w}^+ \in \{w_0, w_1, \dots, w_{m-1}\}$
- Choose $\eta < \frac{1}{4L}$ and m such that

$$\alpha := \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1$$

Then $\mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}^+) - \boldsymbol{P}(\boldsymbol{w}^*)] \leq \boldsymbol{\alpha} \cdot \mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}) - \boldsymbol{P}(\boldsymbol{w}^*)]$

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

Stochastic Variance Reduced Gradient (SVRG)

SVRG [R. Johnson & T. Zhang, 2013]

- Modify stochastic gradient
  1. Choose initial point w_0
  2. Set $\widetilde{w} = w_0$
  3. For $t = 0,1,2,\dots,m$
  4. $$w_{t+1} = w_t - \eta \underbrace{\left(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}) + \nabla P(\widetilde{w})\right)}_{v_t}, \; i_t \in \{1,\dots,n\}$$

- Let $\widetilde{w}^+ \in \{w_0, w_1, \dots, w_{m-1}\}$
- Choose $\eta < \frac{1}{4L}$ and m such that

$$\alpha := \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1$$

Then $\mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}^+) - \boldsymbol{P}(\boldsymbol{w}^*)] \leq \boldsymbol{\alpha} \cdot \mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}) - \boldsymbol{P}(\boldsymbol{w}^*)]$

Note: For fixed η, it would not converge to the optimal solution!

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

Stochastic Variance Reduced Gradient (SVRG)

SVRG [R. Johnson & T. Zhang, 2013]

- Modify stochastic gradient
  1. Choose initial point w_0
  2. Set $\widetilde{w} = w_0$
  3. For $t = 0,1,2,\dots,m$
  4. $$w_{t+1} = w_t - \eta \underbrace{\left(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}) + \nabla P(\widetilde{w})\right)}_{v_t}, \; i_t \in \{1,\dots,n\}$$

- Let $\widetilde{w}^+ \in \{w_0, w_1, \dots, w_{m-1}\}$
- Choose $\eta < \frac{1}{4L}$ and m such that

$$\alpha := \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1$$

Then $\mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}^+) - \boldsymbol{P}(\boldsymbol{w}^*)] \leq \boldsymbol{\alpha} \cdot \mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}) - \boldsymbol{P}(\boldsymbol{w}^*)]$

Note: For fixed η, it would not converge to the optimal solution!

Restarting

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

Stochastic Variance Reduced Gradient (SVRG)

SVRG [R. Johnson & T. Zhang, 2013]

- Modify stochastic gradient
  1. Choose initial point w_0
  2. Set $\widetilde{w} = w_0$
  3. For $t = 0,1,2,\dots,m$
  4. $$w_{t+1} = w_t - \eta \underbrace{\left(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}) + \nabla P(\widetilde{w})\right)}_{v_t}, \; i_t \in \{1,\dots,n\}$$

- Let $\widetilde{w}^+ \in \{w_0, w_1, \dots, w_{m-1}\}$
- Choose $\eta < \frac{1}{4L}$ and m such that

$$\alpha := \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1$$

Then $\mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}^+) - \boldsymbol{P}(\boldsymbol{w}^*)] \leq \boldsymbol{\alpha} \cdot \mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}) - \boldsymbol{P}(\boldsymbol{w}^*)]$

Note: For fixed η, it would not converge to the optimal solution!

Restarting

$$\widetilde{\boldsymbol{w}}^{(\mathbf{0})} \to \widetilde{\boldsymbol{w}}^{(\mathbf{1})} \to \cdots \to \widetilde{\boldsymbol{w}}^{(\boldsymbol{s})}$$

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

Stochastic Variance Reduced Gradient (SVRG)

SVRG [R. Johnson & T. Zhang, 2013]

- Modify stochastic gradient
  1. Choose initial point w_0
  2. Set $\widetilde{w} = w_0$
  3. For $t = 0,1,2,\dots,m$
  4. $$w_{t+1} = w_t - \eta \underbrace{\left(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}) + \nabla P(\widetilde{w})\right)}_{v_t}, \; i_t \in \{1,\dots,n\}$$

- Let $\widetilde{w}^+ \in \{w_0, w_1, \dots, w_{m-1}\}$
- Choose $\eta < \frac{1}{4L}$ and m such that

$$\alpha := \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1$$

Then $\mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}^+) - \boldsymbol{P}(\boldsymbol{w}^*)] \leq \boldsymbol{\alpha} \cdot \mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}) - \boldsymbol{P}(\boldsymbol{w}^*)]$

Note: For fixed η, it would not converge to the optimal solution!

Restarting

$$\widetilde{\boldsymbol{w}}^{(\boldsymbol{0})} \to \widetilde{\boldsymbol{w}}^{(\boldsymbol{1})} \to \cdots \to \widetilde{\boldsymbol{w}}^{(\boldsymbol{s})}$$

Hence, $\mathbb{E}\left[\boldsymbol{P}\left(\widetilde{\boldsymbol{w}}^{(\boldsymbol{s})}\right) - \boldsymbol{P}(\boldsymbol{w}^*)\right] \leq \boldsymbol{\alpha}^{\boldsymbol{s}} \cdot \left[\boldsymbol{P}\left(\widetilde{\boldsymbol{w}}^{(\boldsymbol{0})}\right) - \boldsymbol{P}(\boldsymbol{w}^*)\right]$

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

Stochastic Variance Reduced Gradient (SVRG)

SVRG [R. Johnson & T. Zhang, 2013]

- Modify stochastic gradient
  1. Choose initial point w_0
  2. Set $\widetilde{w} = w_0$
  3. For $t = 0,1,2,\dots,m$
  4. $$w_{t+1} = w_t - \eta \underbrace{\left(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\widetilde{w}) + \nabla P(\widetilde{w})\right)}_{v_t}, \; i_t \in \{1,\dots,n\}$$

- Let $\widetilde{w}^+ \in \{w_0, w_1, \dots, w_{m-1}\}$
- Choose $\eta < \frac{1}{4L}$ and m such that

$$\alpha := \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1$$

Then $\mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}^+) - \boldsymbol{P}(\boldsymbol{w}^*)] \leq \boldsymbol{\alpha} \cdot \mathbb{E}[\boldsymbol{P}(\widetilde{\boldsymbol{w}}) - \boldsymbol{P}(\boldsymbol{w}^*)]$

Note: For fixed η, it would not converge to the optimal solution!

Restarting

$$\widetilde{\boldsymbol{w}}^{(\boldsymbol{0})} \to \widetilde{\boldsymbol{w}}^{(\boldsymbol{1})} \to \cdots \to \widetilde{\boldsymbol{w}}^{(\boldsymbol{s})}$$

Hence, $\mathbb{E}\left[\boldsymbol{P}\left(\widetilde{\boldsymbol{w}}^{(\boldsymbol{s})}\right) - \boldsymbol{P}(\boldsymbol{w}^*)\right] \leq \boldsymbol{\alpha}^{\boldsymbol{s}} \cdot \left[\boldsymbol{P}\left(\widetilde{\boldsymbol{w}}^{(\boldsymbol{0})}\right) - \boldsymbol{P}(\boldsymbol{w}^*)\right]$ **No storage is required!**

R. Johnson and T. Zhang. **Accelerating Stochastic Gradient Descent using Predictive Variance Reduction**, 2013

SVRG One Outer Loop Behavior

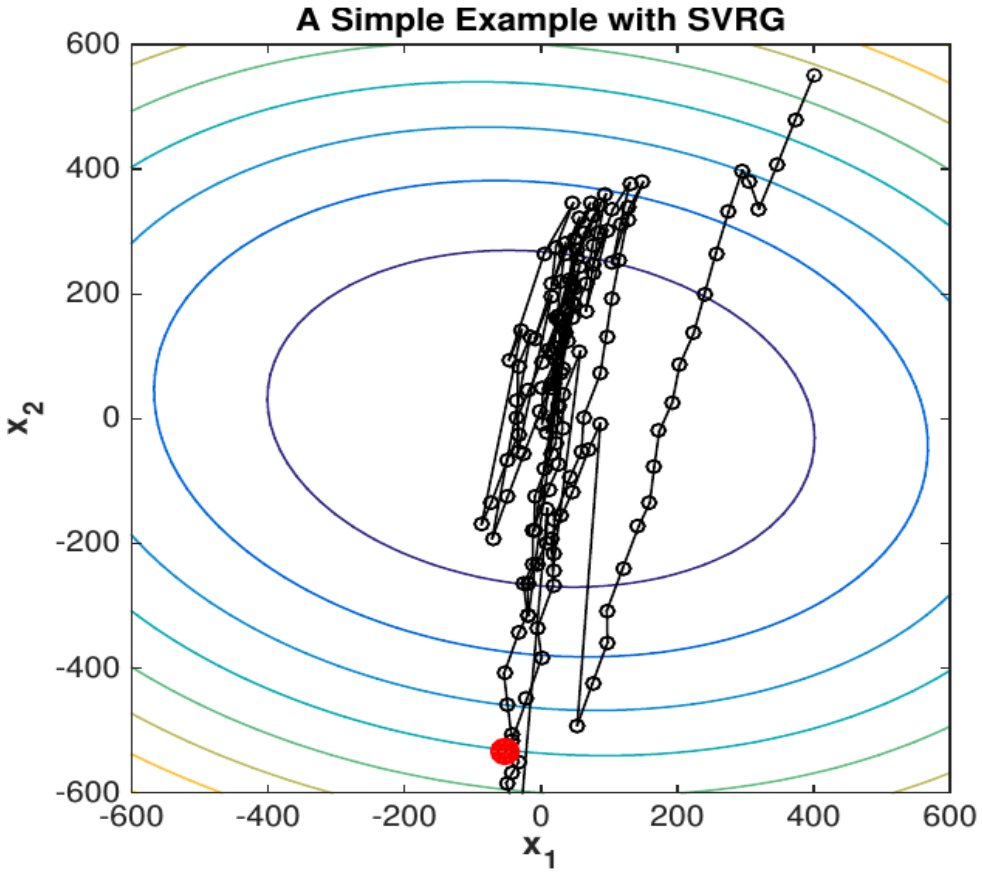
An issue:

- How to choose "*m*" in algorithm?

SVRG One Outer Loop Behavior

An issue:

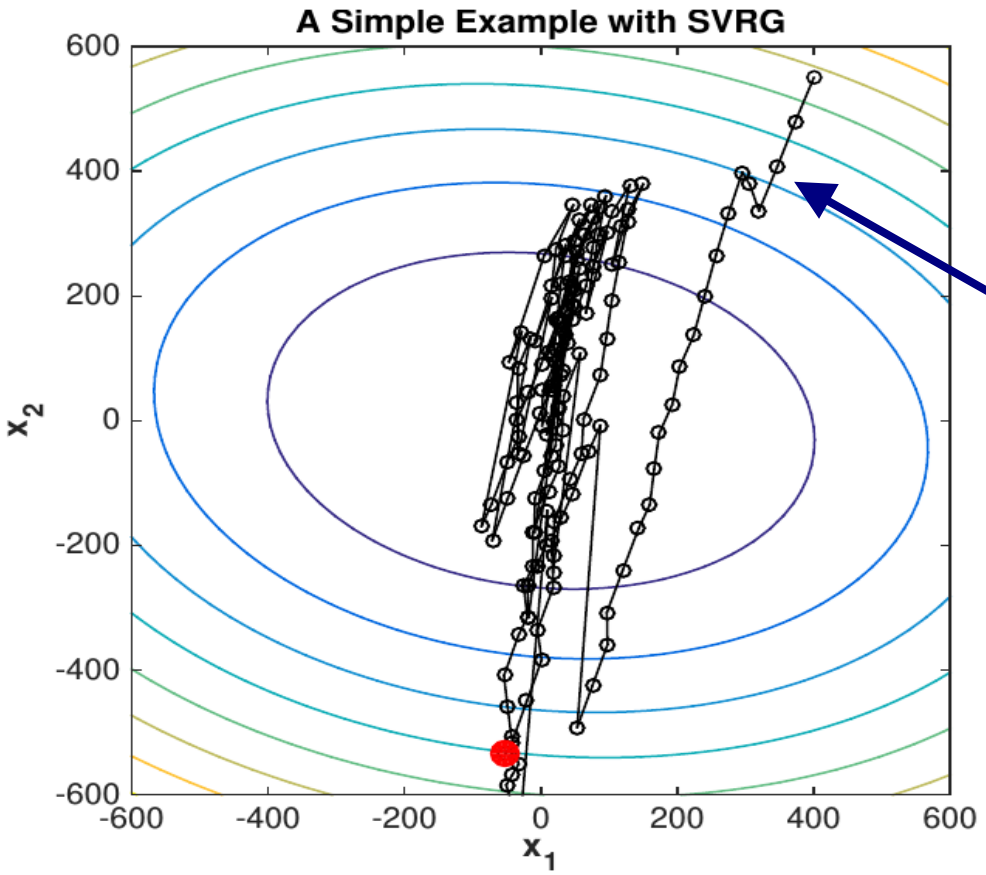
- How to choose "*m*" in algorithm?



SVRG One Outer Loop Behavior

An issue:

- How to choose "*m*" in algorithm?

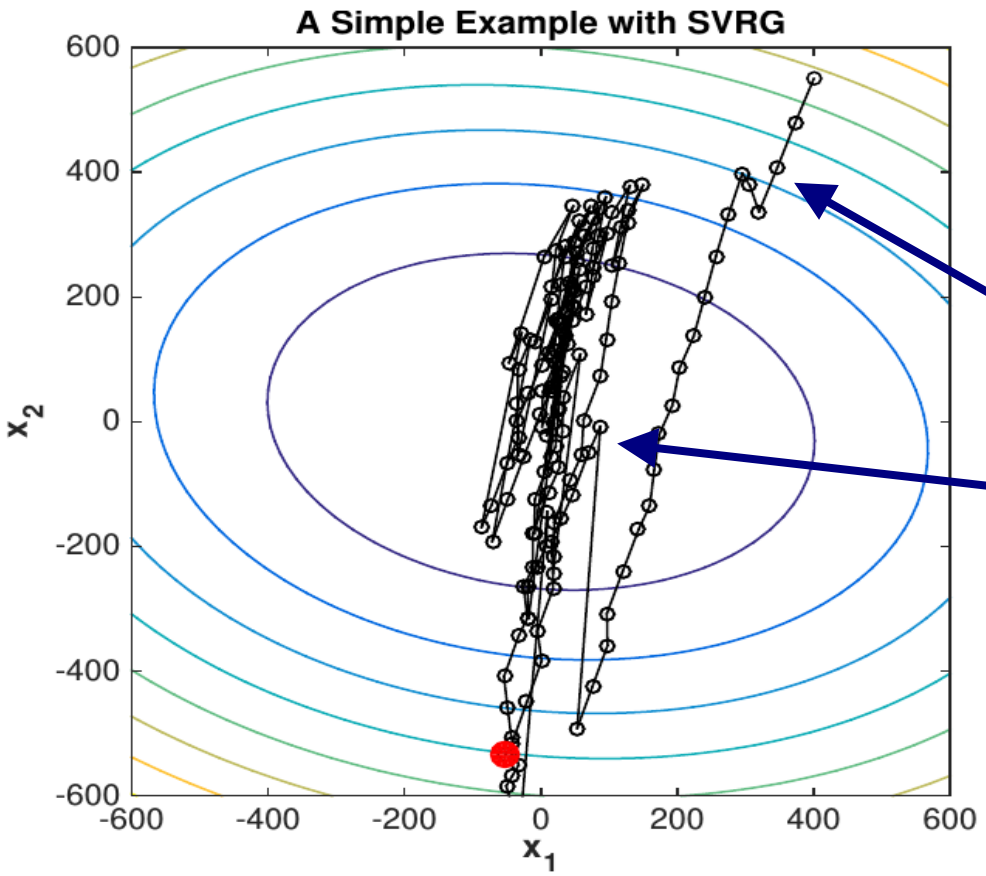


"*m*" too small!

SVRG One Outer Loop Behavior

An issue:

- How to choose "*m*" in algorithm?



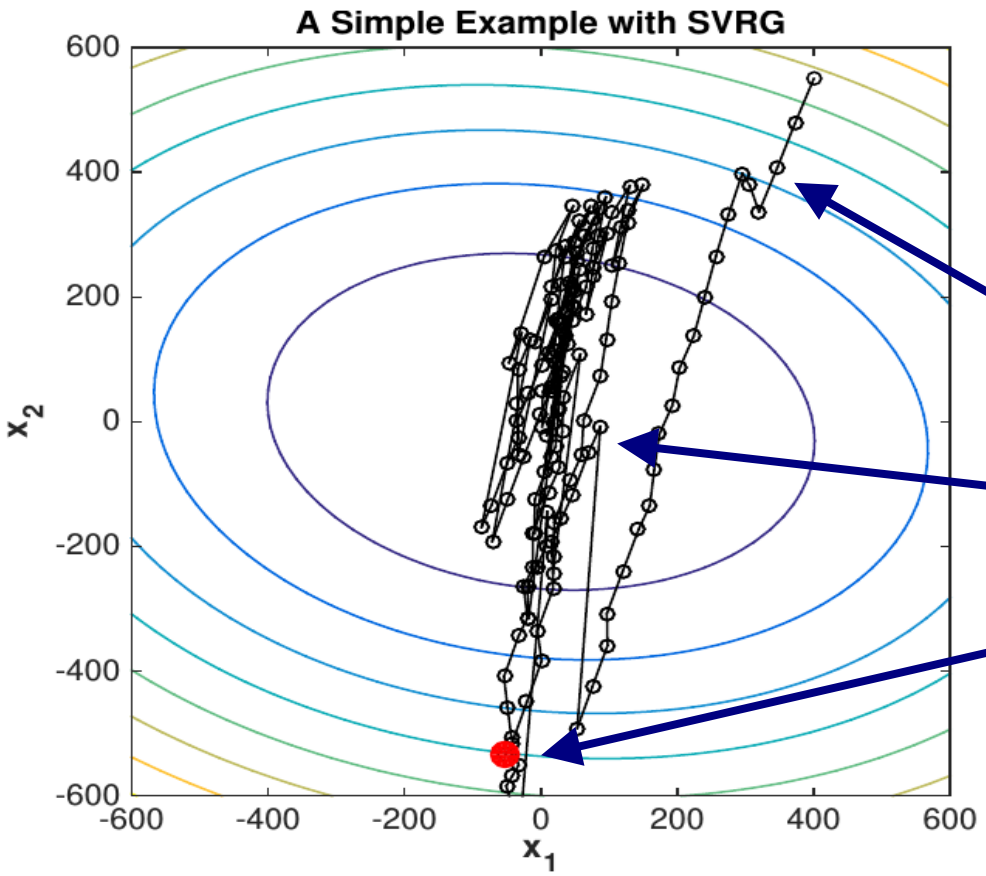
"*m*" too small!

good "*m*"

SVRG One Outer Loop Behavior

An issue:

- How to choose "*m*" in algorithm?



"*m*" too small!

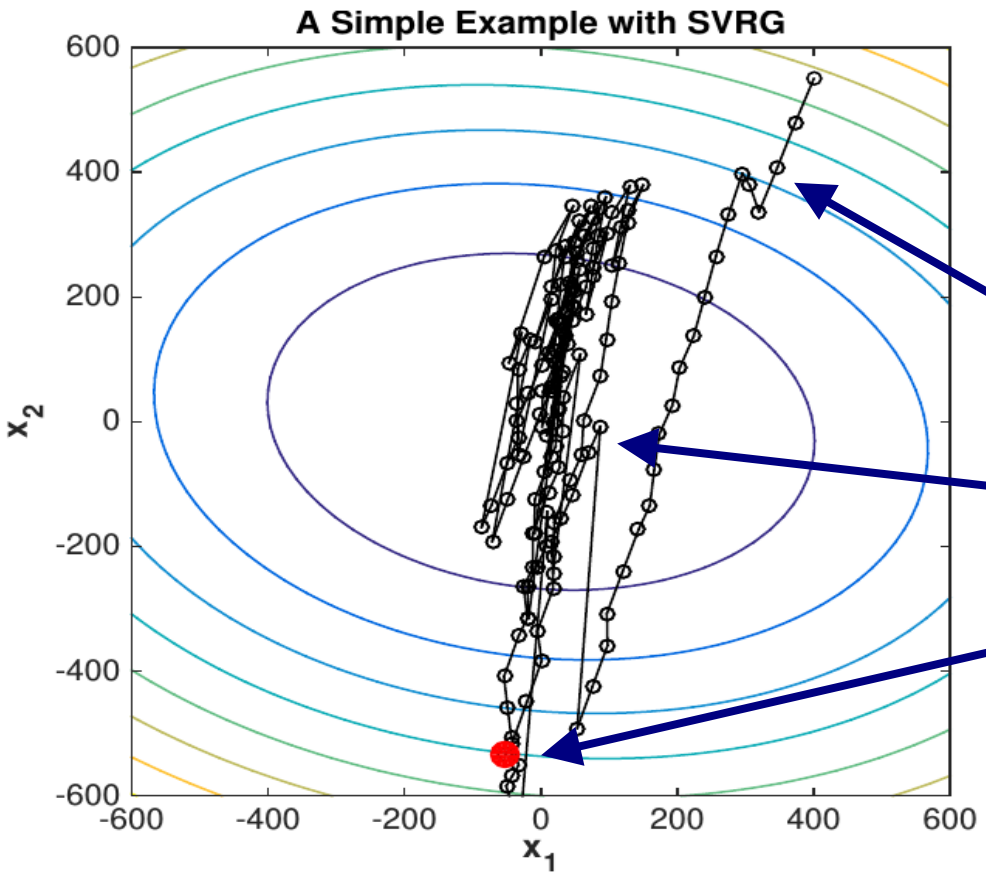
good "*m*"

"*m*" too large!

SVRG One Outer Loop Behavior

An issue:

- How to choose "*m*" in algorithm?



"*m*" too small!

good "*m*"

"*m*" too large!

The trajectory for one outer loop is very "unstable"

SARAH

- L. Nguyen, J. Liu, K. Scheinberg, and M. Takac. **SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient**, 2017
- L. Nguyen, J. Liu, K. Scheinberg, and M. Takac. **Stochastic Recursive Gradient Algorithm for Nonconvex Optimization**, 2017

SARAH Algorithm

SARAH Algorithm

- It also does **restarting** as SVRG
- It takes **recursive** gradient estimator

SARAH Algorithm

- It also does **restarting** as SVRG
- It takes **recursive** gradient estimator

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  $v_0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w}_s = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
end for

SARAH Algorithm

- It also does **restarting** as SVRG
- It takes **recursive** gradient estimator

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  $v_0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w}_s = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
end for

Outer loop

SARAH Algorithm

- It also does **restarting** as SVRG
- It takes **recursive** gradient estimator

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  $v_0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w}_s = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
end for

Inner loop

Outer loop

SARAH Algorithm

- It also does **restarting** as SVRG
- It takes **recursive** gradient estimator

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  $v_0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w}_s = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
end for

Full gradient computing

Inner loop

Outer loop

SARAH Algorithm

- It also does **restarting** as SVRG
- It takes **recursive** gradient estimator

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  $v_0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w}_s = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
end for

Full gradient computing

Inner loop

Outer loop

SARAH update (stochastic gradient computing)

SARAH Algorithm

- It also does **restarting** as SVRG
- It takes **recursive** gradient estimator

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  $v_0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w}_s = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
end for

Full gradient computing

Inner loop

Outer loop

SARAH update (stochastic gradient computing)

No storage is required!

Derivation of SARAH update

- Try to approximate **Gradient Descent**

Derivation of SARAH update

- Try to approximate **Gradient Descent**

Recall the update: $w_{t+1} = w_t - \eta v_t$

We want: $\boldsymbol{v_t \approx \nabla P(w_t)}$

Derivation of SARAH update

- Try to approximate **Gradient Descent**

Recall the update: $w_{t+1} = w_t - \eta v_t$

We want: $\boldsymbol{v_t \approx \nabla P(w_t)}$

According to L-Lipschitz smooth property, we have

$$||\nabla P(w_t) - \nabla P(w_{t-1})|| \leq L||w_t - w_{t-1}|| = L\eta||v_{t-1}||$$

$$||\nabla f_i(w_t) - \nabla f_i(w_{t-1})|| \leq L||w_t - w_{t-1}|| = L\eta||v_{t-1}||, \forall i \in \{1, \dots, n\}$$

Derivation of SARAH update

- Try to approximate **Gradient Descent**

Recall the update: $w_{t+1} = w_t - \eta v_t$

We want: $\boldsymbol{v_t \approx \nabla P(w_t)}$

According to L-Lipschitz smooth property, we have

$$||\nabla P(w_t) - \nabla P(w_{t-1})|| \leq L||w_t - w_{t-1}|| = L\eta||v_{t-1}||$$

$$||\nabla f_i(w_t) - \nabla f_i(w_{t-1})|| \leq L||w_t - w_{t-1}|| = L\eta||v_{t-1}||, \forall i \in \{1, \dots, n\}$$

When η is small enough, we have

$$\nabla P(w_t) \approx \nabla f_i(w_t) - \nabla f_i(w_{t-1}) + \nabla P(w_{t-1})$$

Derivation of SARAH update

- Try to approximate **Gradient Descent**

Recall the update: $w_{t+1} = w_t - \eta v_t$

We want: $\boldsymbol{v_t \approx \nabla P(w_t)}$

According to L-Lipschitz smooth property, we have

$$||\nabla P(w_t) - \nabla P(w_{t-1})|| \leq L||w_t - w_{t-1}|| = L\eta||v_{t-1}||$$

$$||\nabla f_i(w_t) - \nabla f_i(w_{t-1})|| \leq L||w_t - w_{t-1}|| = L\eta||v_{t-1}||, \forall i \in \{1, \dots, n\}$$

When η is small enough, we have

$$\nabla P(w_t) \approx \nabla f_i(w_t) - \nabla f_i(w_{t-1}) + \nabla P(w_{t-1})$$

$$\Rightarrow v_t \approx \nabla f_i(w_t) - \nabla f_i(w_{t-1}) + v_{t-1}$$

SARAH has a biased estimator of gradient

- **SGD, SAGA, SVRG are conditionally unbiased**

$$\mathbb{E}[v_t|\mathcal{F}_t] = \nabla P(w_t)$$

SARAH has a biased estimator of gradient

- **SGD, SAGA, SVRG are conditionally unbiased**

$$\mathbb{E}[v_t|\mathcal{F}_t] = \nabla P(w_t)$$

- **SARAH is conditionally biased**

SARAH has a biased estimator of gradient

- **SGD, SAGA, SVRG are conditionally unbiased**

$$\mathbb{E}[v_t|\mathcal{F}_t] = \nabla P(w_t)$$

- **SARAH is conditionally biased**

Recall: $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$

We have

$$\mathbb{E}[v_t|\mathcal{F}_t] = \nabla P(w_t) - \nabla P(w_{t-1}) + v_{t-1} \neq \nabla P(w_t)$$

Conditioned on $\{w_0, i_0, i_1, \dots, i_{t-1}\}$

SARAH has a biased estimator of gradient

- **SGD, SAGA, SVRG are conditionally unbiased**

$$\mathbb{E}[v_t|\mathcal{F}_t] = \nabla P(w_t)$$

- **SARAH is conditionally biased**

Recall: $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$

We have

$$\mathbb{E}[v_t|\mathcal{F}_t] = \nabla P(w_t) - \nabla P(w_{t-1}) + v_{t-1} \neq \nabla P(w_t)$$

Conditioned on $\{w_0, i_0, i_1, \dots, i_{t-1}\}$

However,

$$\mathbb{E}[v_t] = \mathbb{E}[\nabla P(w_t)]$$

SARAH Convergence

- Choose $\eta < \frac{1}{L}$ and m such that

$$\sigma := \frac{1}{\mu\eta(m+1)} + \frac{L\eta}{2 - L\eta} < 1$$

SARAH Convergence

- Choose $\eta < \frac{1}{L}$ and m such that

$$\sigma := \frac{1}{\mu\eta(m+1)} + \frac{L\eta}{2 - L\eta} < 1$$

Then, $\mathbb{E}\left[||\nabla P\left(\tilde{w}^{(s)}\right)||^2\right] \leq \sigma^s \cdot ||\nabla P\left(\tilde{w}^{(0)}\right)||^2$

SARAH Convergence

- Choose $\eta < \frac{1}{L}$ and m such that

$$\sigma := \frac{1}{\mu\eta(m+1)} + \frac{L\eta}{2 - L\eta} < 1$$

Then, $\mathbb{E}\left[||\nabla \boldsymbol{P}\left(\widetilde{\boldsymbol{w}}^{(s)}\right)||^2\right] \leq \boldsymbol{\sigma}^{\boldsymbol{s}} \cdot ||\nabla \boldsymbol{P}\left(\widetilde{\boldsymbol{w}}^{(0)}\right)||^2$

It is a little bit better than SVRG (since SARAH could use the fixed learning rate, whose size is **larger than** that of SVRG and $\boldsymbol{\sigma} < \boldsymbol{\alpha}$ with the same η and m).

SARAH Convergence

- Choose $\eta < \frac{1}{L}$ and m such that

$$\sigma := \frac{1}{\mu\eta(m+1)} + \frac{L\eta}{2 - L\eta} < 1$$

Then, $\mathbb{E}\left[||\nabla P(\tilde{w}^{(s)})||^2\right] \leq \sigma^s \cdot ||\nabla P(\tilde{w}^{(0)})||^2$

It is a little bit better than SVRG (since SARAH could use the fixed learning rate, whose size is **larger than** that of SVRG and $\sigma < \alpha$ with the same η and m).

But, they are still considered as **the same rate of convergence (linear)**

SARAH Convergence

- Choose $\eta < \frac{1}{L}$ and m such that

$$\sigma := \frac{1}{\mu\eta(m+1)} + \frac{L\eta}{2 - L\eta} < 1$$

Then, $\mathbb{E}\left[||\nabla P(\tilde{w}^{(s)})||^2\right] \leq \sigma^s \cdot ||\nabla P(\tilde{w}^{(0)})||^2$

It is a little bit better than SVRG (since SARAH could use the fixed learning rate, whose size is **larger than** that of SVRG and $\sigma < \alpha$ with the same η and m).

But, they are still considered as **the same rate of convergence (linear)**

What is the main difference between SARAH and SVRG?

SARAH One Outer Loop

Recall the update: $w_{t+1} = w_t - \eta v_t$

SARAH One Outer Loop

Recall the update: $w_{t+1} = w_t - \eta v_t$

- P is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \left(\frac{2}{\eta L} - 1\right)\mu^2\eta^2 < 1, \qquad \eta < \frac{2}{L}$$

SARAH One Outer Loop

Recall the update: $w_{t+1} = w_t - \eta v_t$

- P is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \left(\frac{2}{\eta L} - 1\right)\mu^2\eta^2 < 1, \qquad \eta < \frac{2}{L}$$

- Each $f_i, \forall i$, is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \frac{2\mu L\eta}{\mu + L} < 1, \qquad \eta \leq \frac{2}{L + \mu}$$

SARAH One Outer Loop

Recall the update: $w_{t+1} = w_t - \eta v_t$

- P is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \left(\frac{2}{\eta L} - 1\right)\mu^2\eta^2 < 1, \qquad \eta < \frac{2}{L}$$

- Each $f_i, \forall i$, is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \frac{2\mu L\eta}{\mu + L} < 1, \qquad \eta \leq \frac{2}{L + \mu}$$

Hence,

$$\mathbb{E}\left[||\boldsymbol{v_t}||^2\right] \to \mathbf{0} \Rightarrow \mathbb{E}\left[||\boldsymbol{w_{t+1}} - \boldsymbol{w_t}||^2\right] \to \mathbf{0}$$

SARAH One Outer Loop

Recall the update: $w_{t+1} = w_t - \eta v_t$

- P is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \left(\frac{2}{\eta L} - 1\right)\mu^2\eta^2 < 1, \qquad \eta < \frac{2}{L}$$

- Each $f_i, \forall i$, is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \frac{2\mu L\eta}{\mu + L} < 1, \qquad \eta \leq \frac{2}{L + \mu}$$

Hence,

$$\mathbb{E}\left[||\boldsymbol{v_t}||^2\right] \to \mathbf{0} \Rightarrow \mathbb{E}\left[||\boldsymbol{w_{t+1}} - \boldsymbol{w_t}||^2\right] \to \mathbf{0}$$

SARAH is converging (somewhere) within a single outer loop with fixed "large" learning rate

SARAH One Outer Loop

Recall the update: $w_{t+1} = w_t - \eta v_t$

- P is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \left(\frac{2}{\eta L} - 1\right)\mu^2\eta^2 < 1, \qquad \eta < \frac{2}{L}$$

- Each $f_i, \forall i$, is L-smooth and μ-strongly convex

$$\mathbb{E}[||v_t||^2] \leq \rho^t \cdot \mathbb{E}[||\nabla P(w_0)||^2]$$

$$\rho = 1 - \frac{2\mu L\eta}{\mu + L} < 1, \qquad \eta \leq \frac{2}{L + \mu}$$

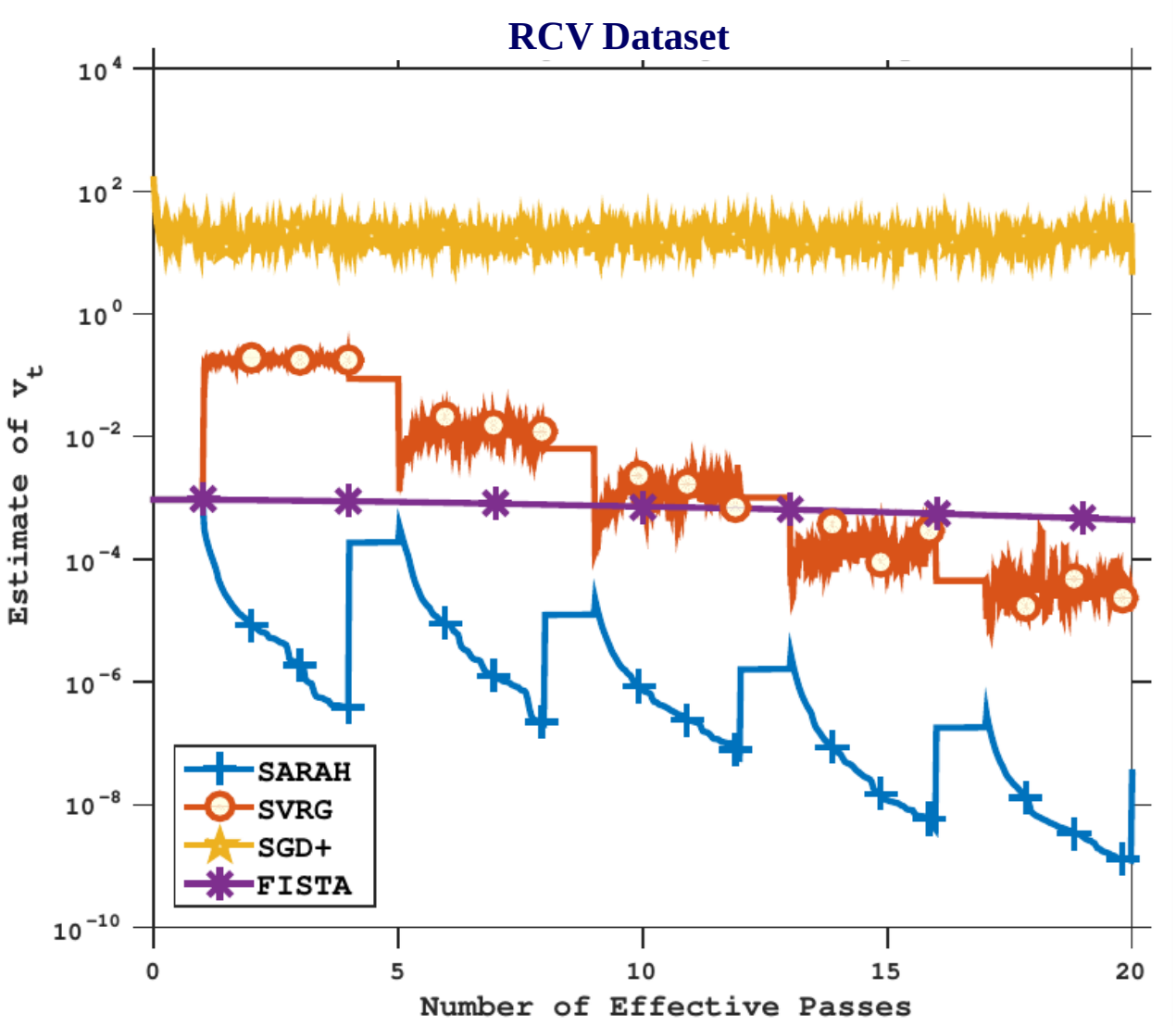
Hence,

$$\mathbb{E}\left[||\boldsymbol{v_t}||^2\right] \to \mathbf{0} \Rightarrow \mathbb{E}\left[||\boldsymbol{w_{t+1}} - \boldsymbol{w_t}||^2\right] \to \mathbf{0}$$

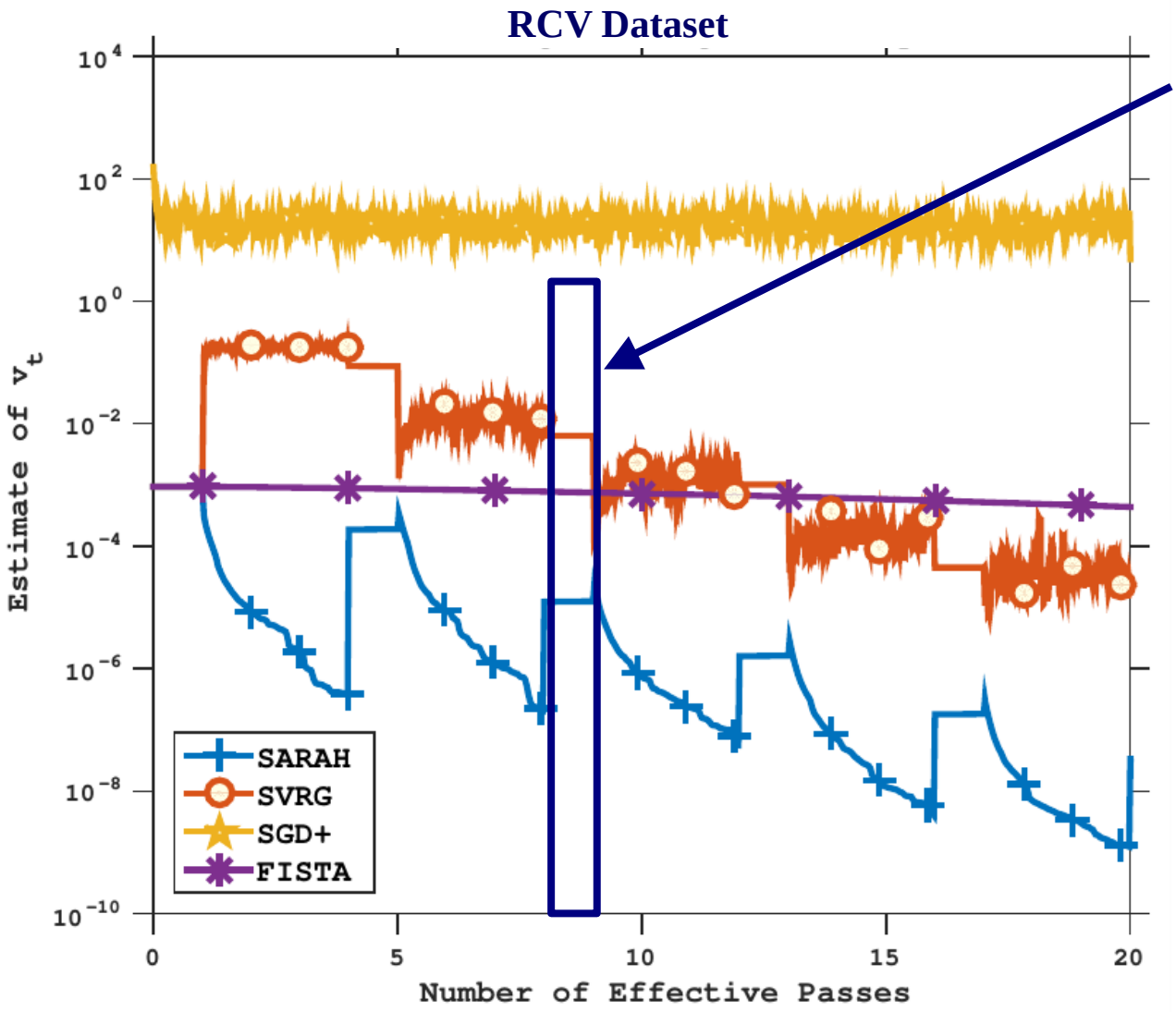
SARAH is converging (somewhere) within a single outer loop with fixed "large" learning rate

SARAH converges to ϵ-accurate solution within a single outer loop with fixed "small" learning rate for general convex and nonconvex cases (Results and Proofs in the papers)

SARAH Demonstration

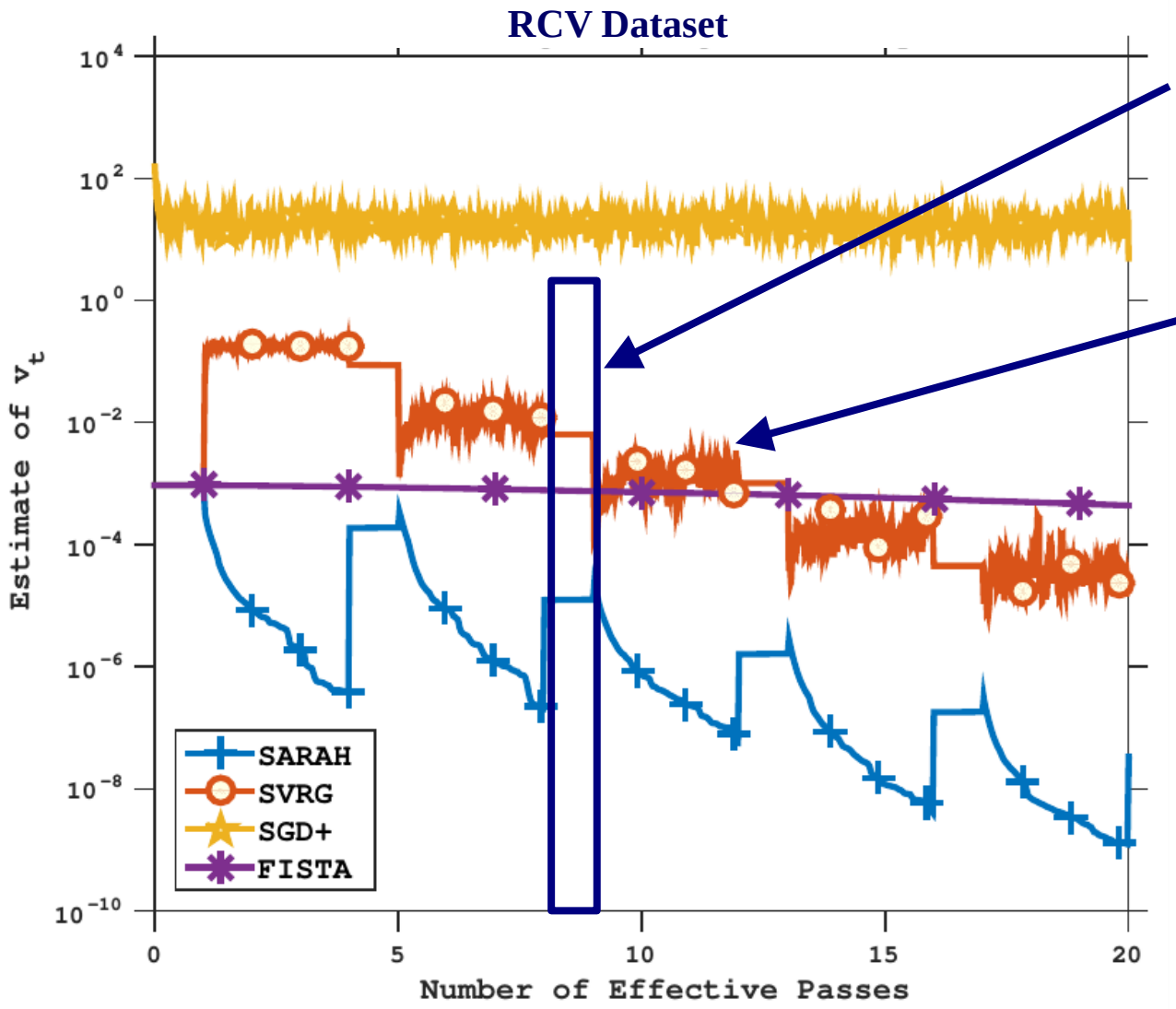


SARAH Demonstration



SVRG and SARAH need full gradient after restart

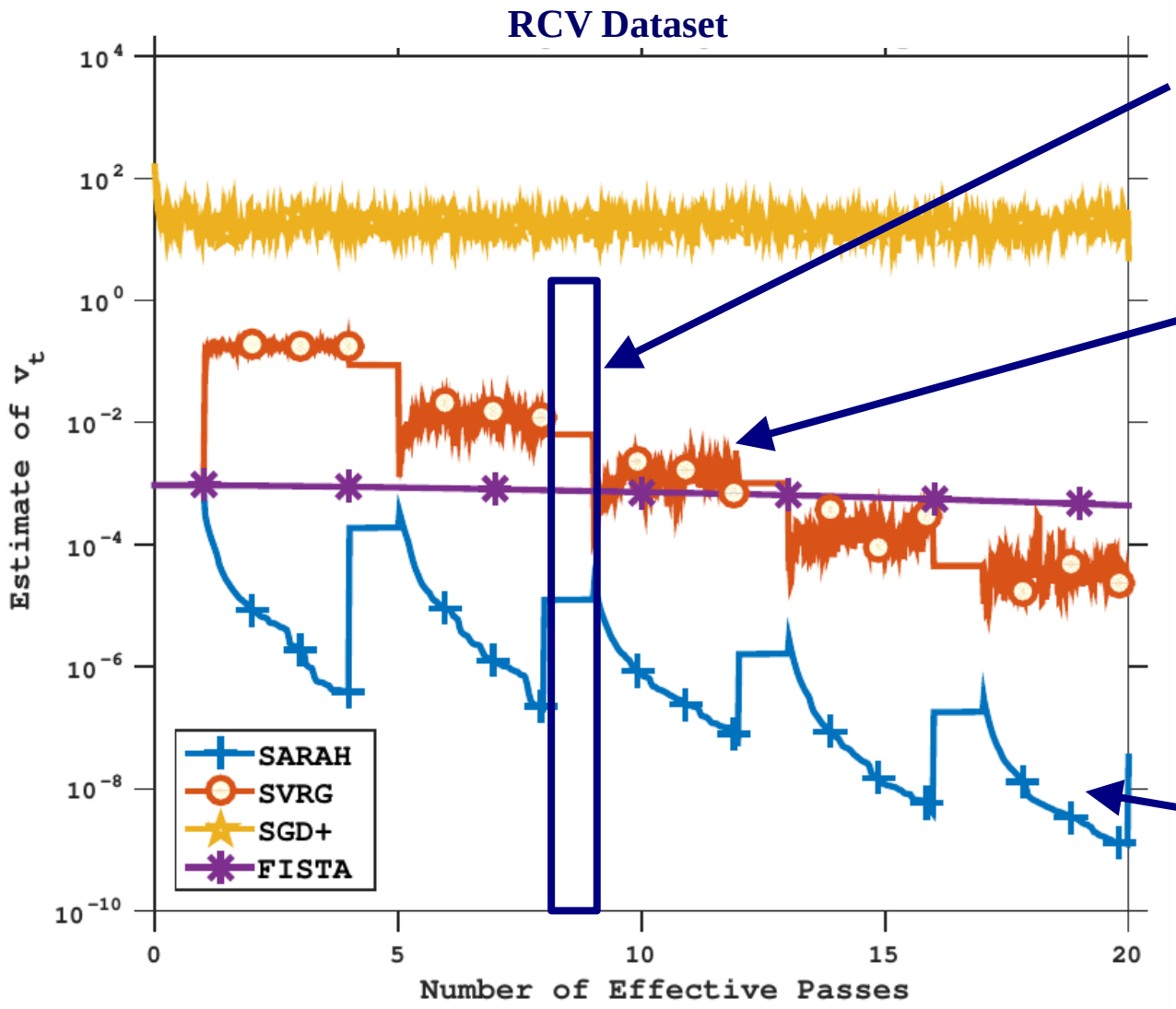
SARAH Demonstration



SVRG and SARAH need full gradient after restart

Variance of SVRG is decreased after each restart

SARAH Demonstration

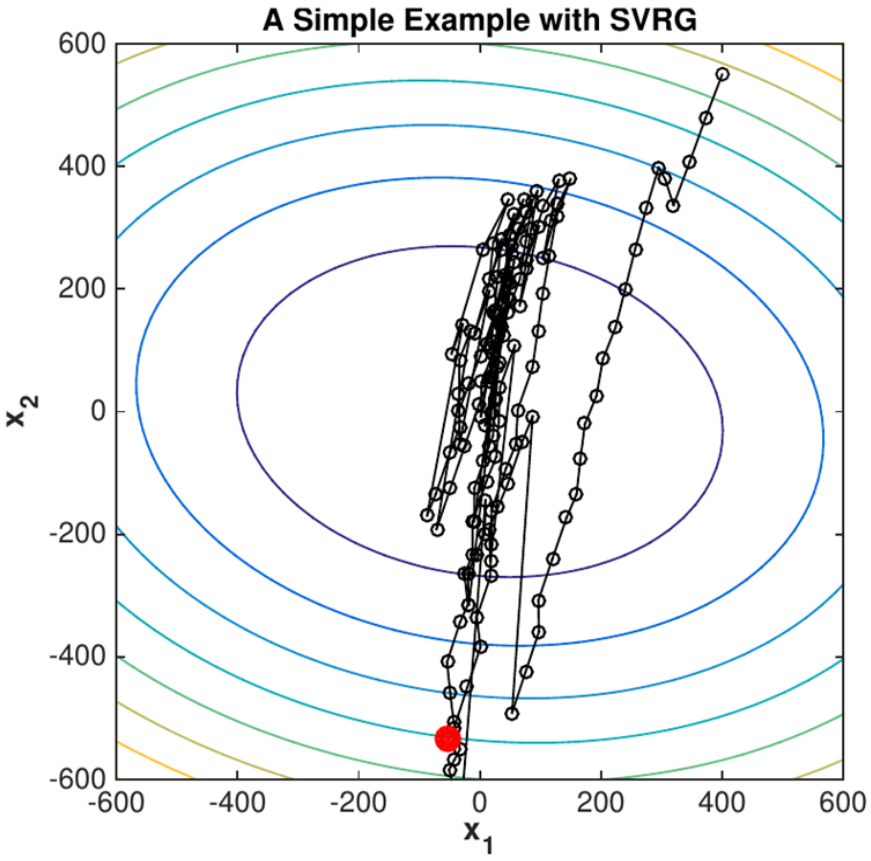


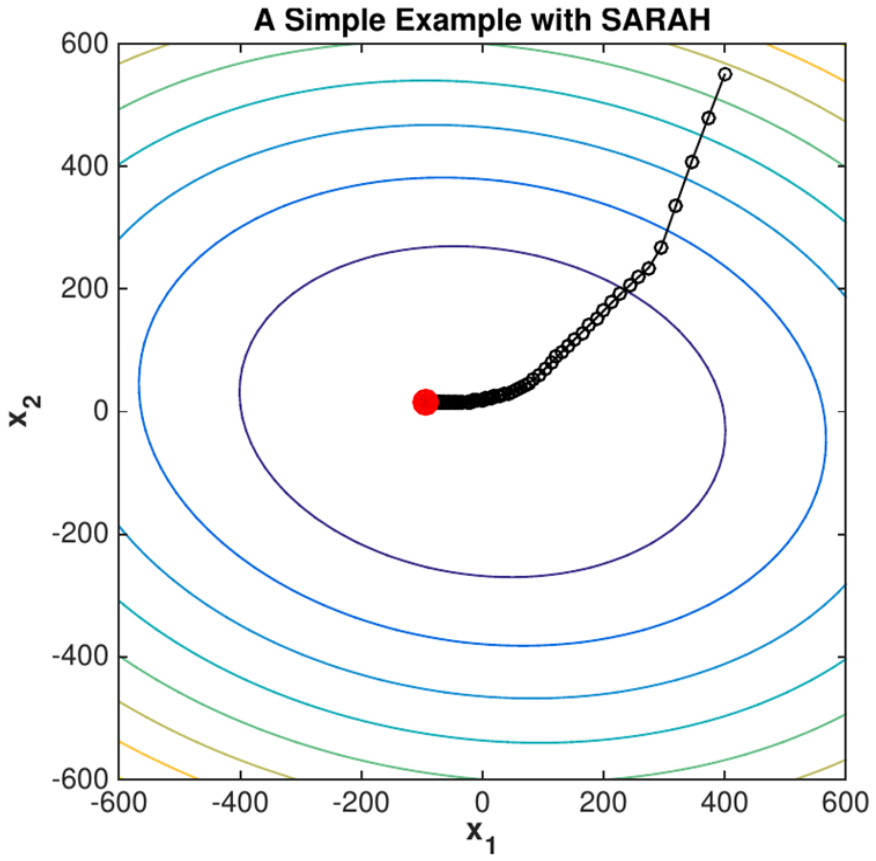
SVRG and SARAH need full gradient after restart

Variance of SVRG is decreased after each restart

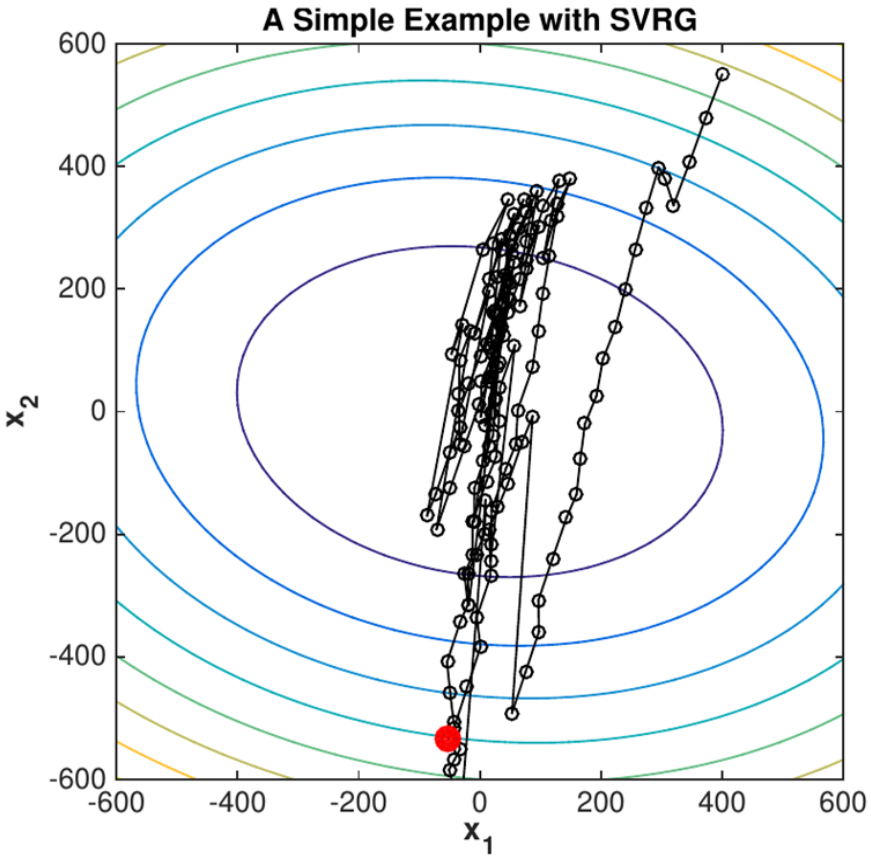
Variance of SARAH goes to zero

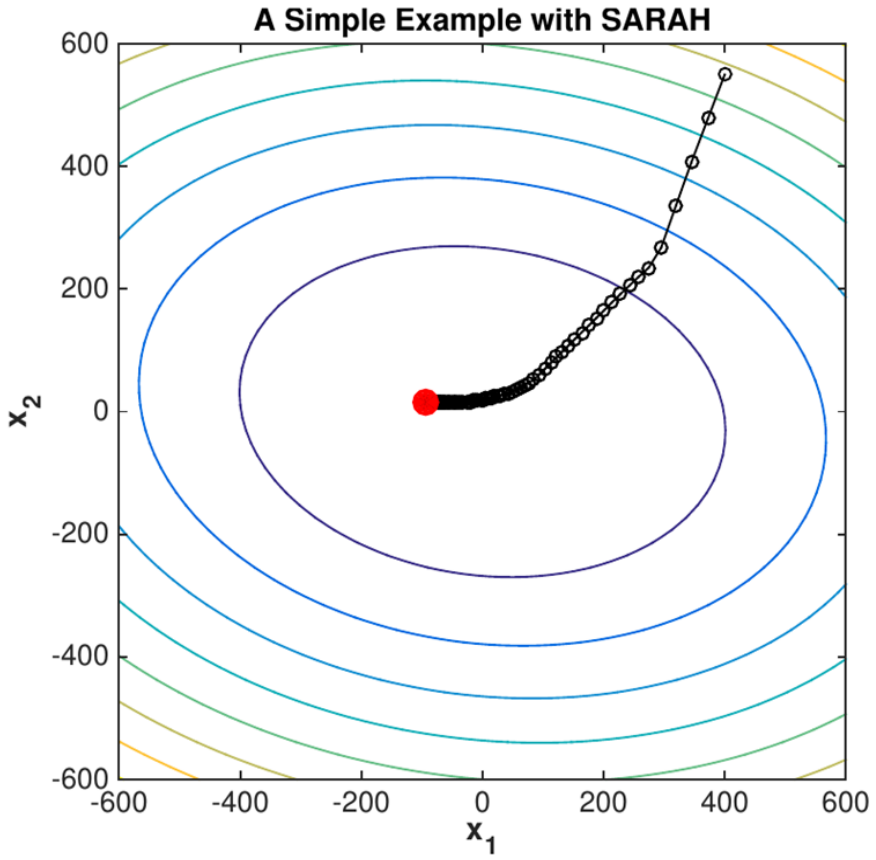
One Outer Loop Behavior





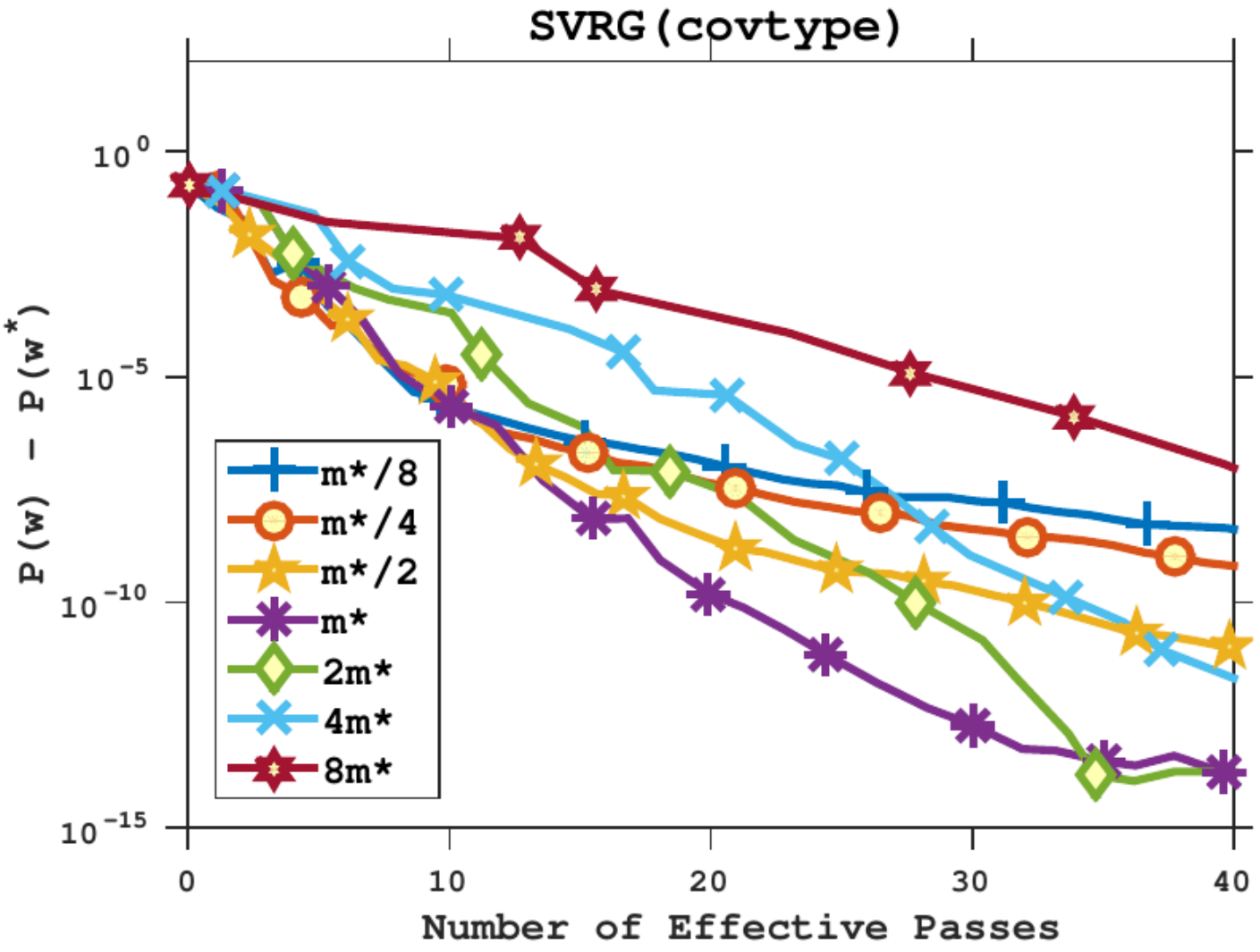
One Outer Loop Behavior



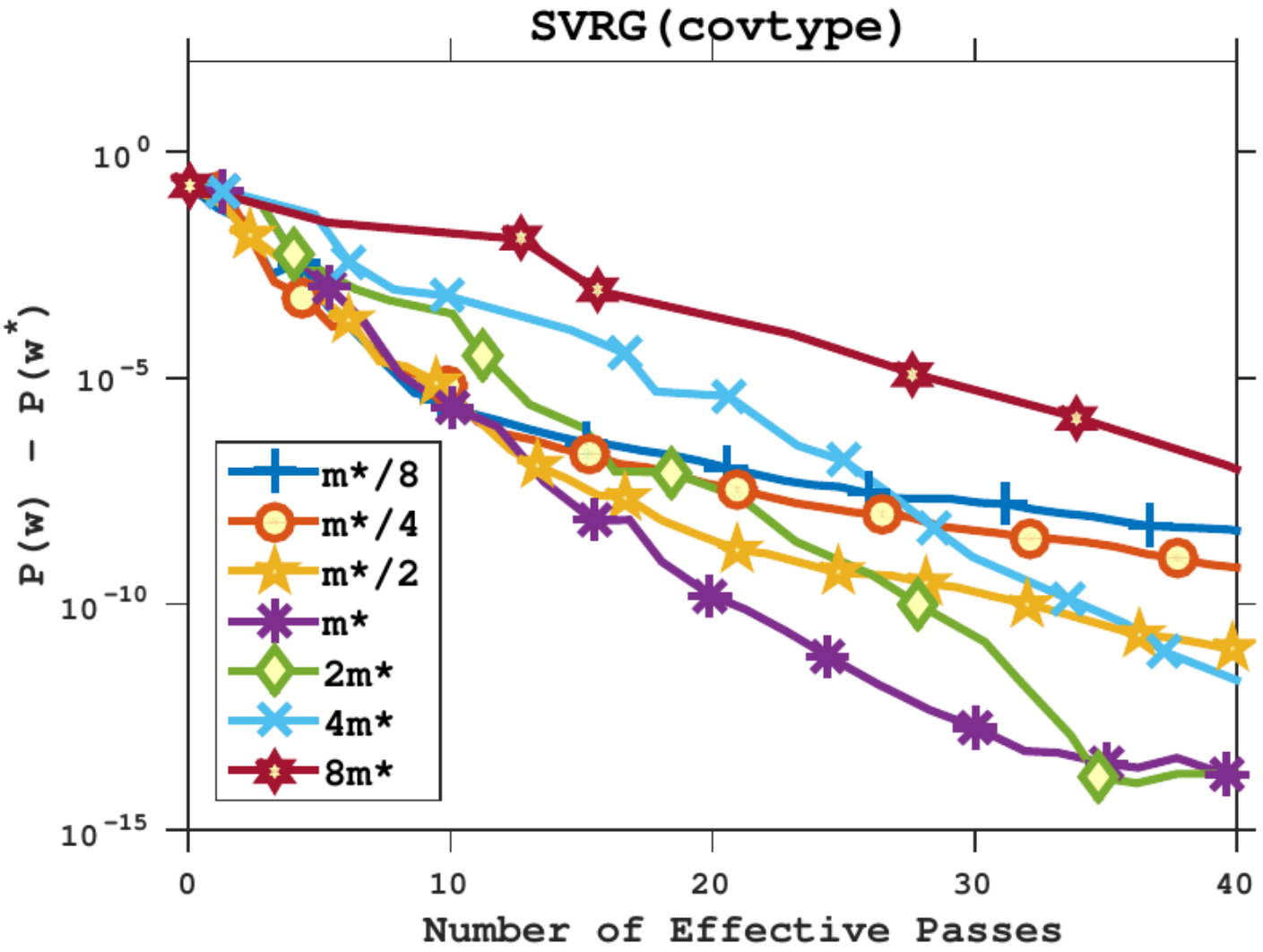


SARAH is more stable than SVRG!

Sensitivity of SVRG and SARAH on "*m*"

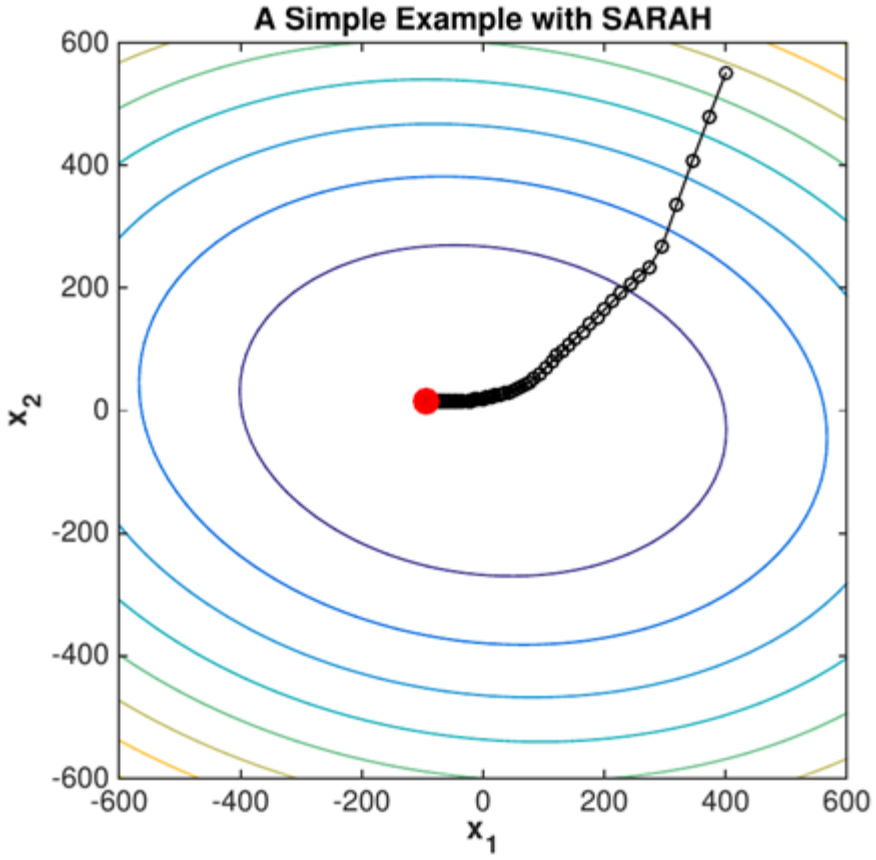


Sensitivity of SVRG and SARAH on "*m*"

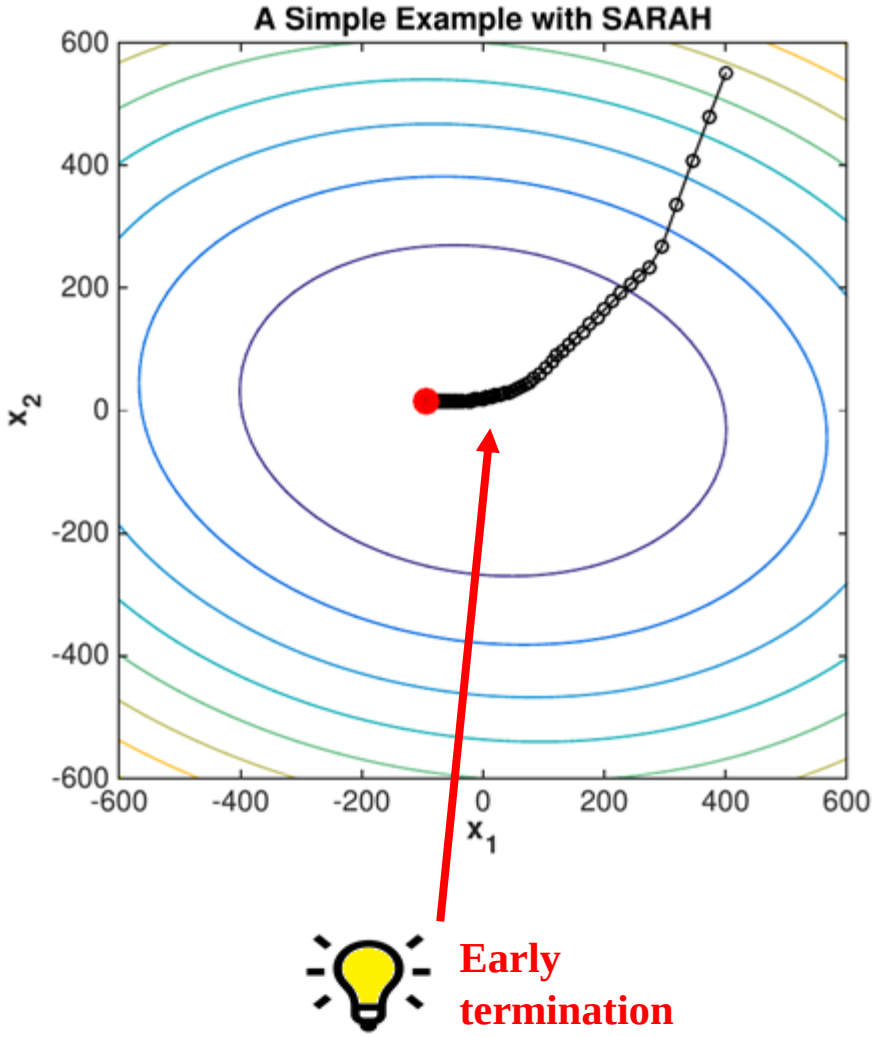


SARAH has a similar behavior!

SARAH One Outer Loop Behavior



SARAH One Outer Loop Behavior



SARAH+

- L. Nguyen, J. Liu, K. Scheinberg, and M. Takac. **SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient**, 2017

SARAH+ (Practical Variant)

Fact #1: Size of update is shrinking

It doesn’t make sense to do many tiny steps!

SARAH+ (Practical Variant)

Fact #1: Size of update is shrinking

It doesn't make sense to do many tiny steps!

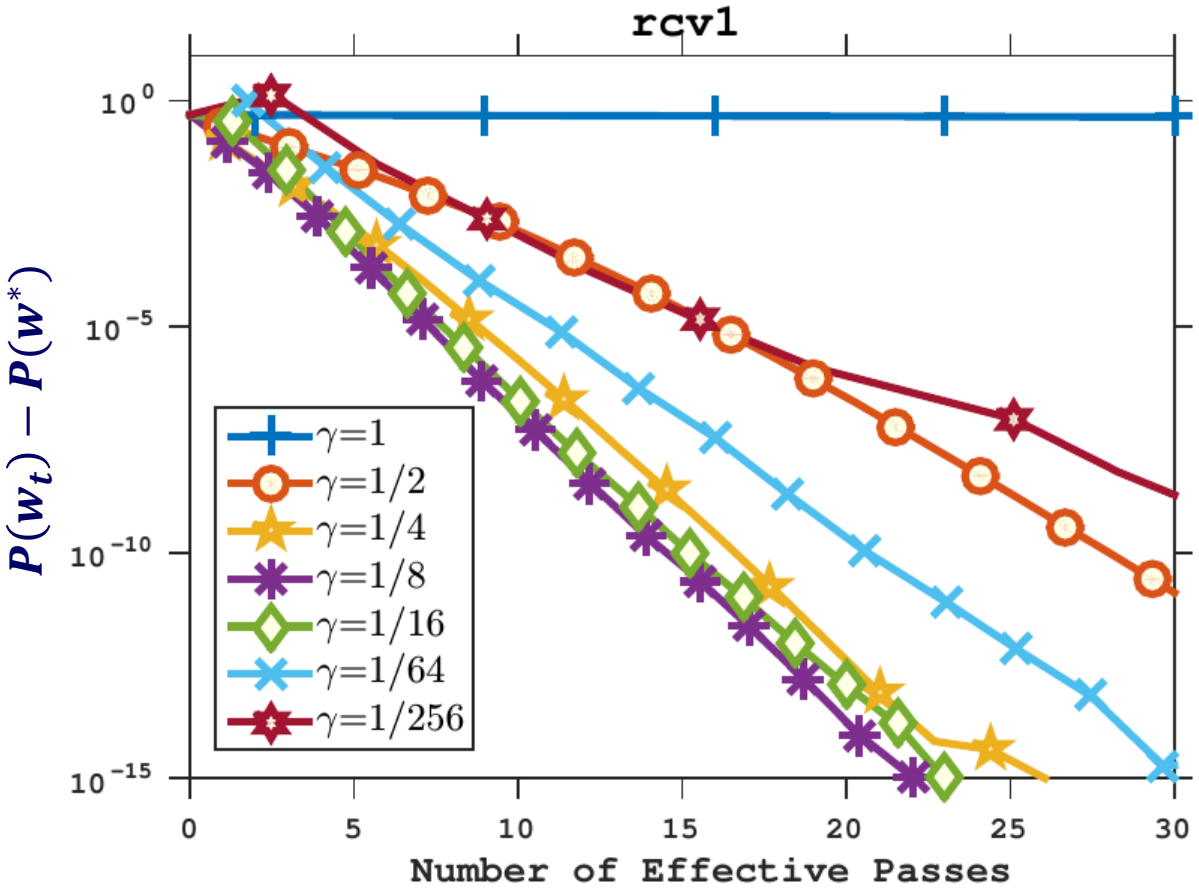
Heuristic: **Restart** algorithm when $\|v_t\|^2 \leq \gamma \|v_0\|^2$

SARAH+ (Practical Variant)

Fact #1: Size of update is shrinking

It doesn't make sense to do many tiny steps!

Heuristic: **Restart** algorithm when $\|v_t\|^2 \leq \gamma \|v_0\|^2$

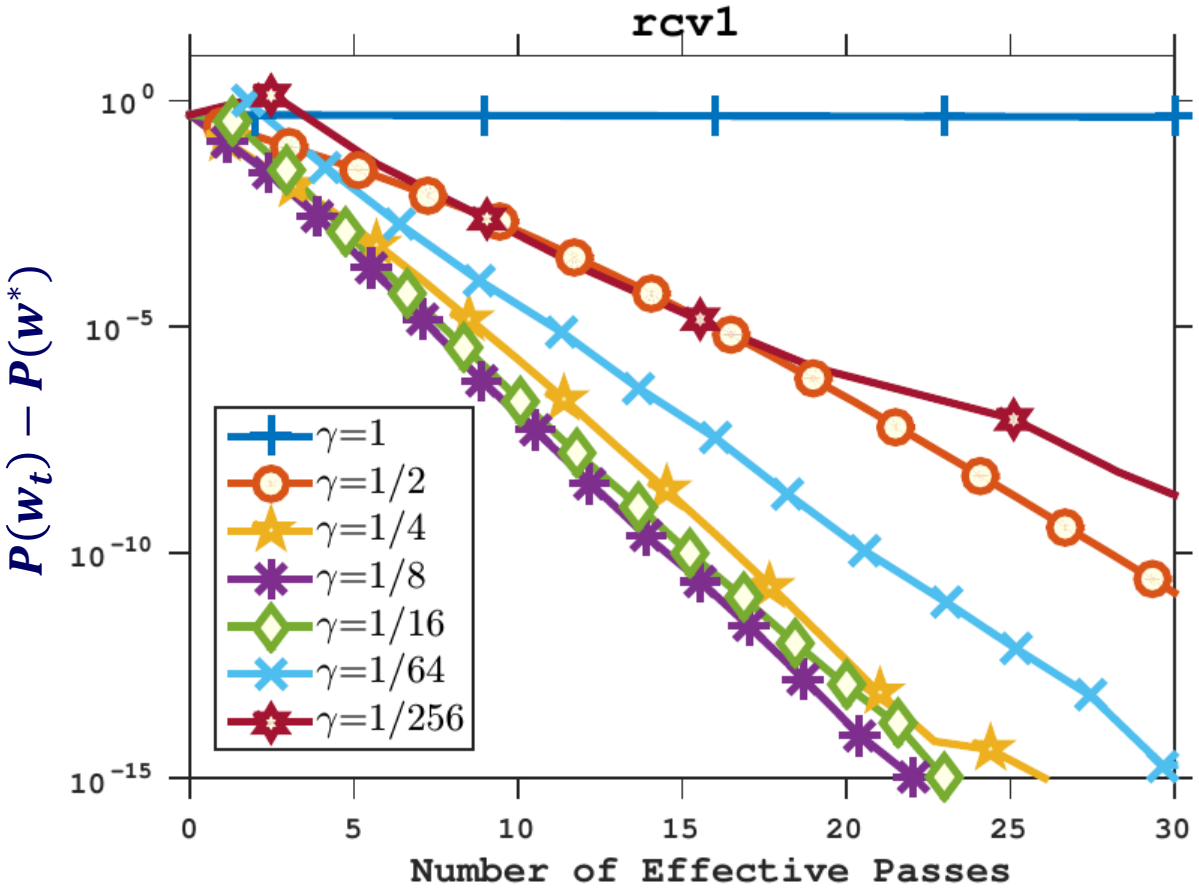


SARAH+ (Practical Variant)

Fact #1: Size of update is shrinking

It doesn't make sense to do many tiny steps!

Heuristic: **Restart** algorithm when $\|v_t\|^2 \leq \gamma \|v_0\|^2$

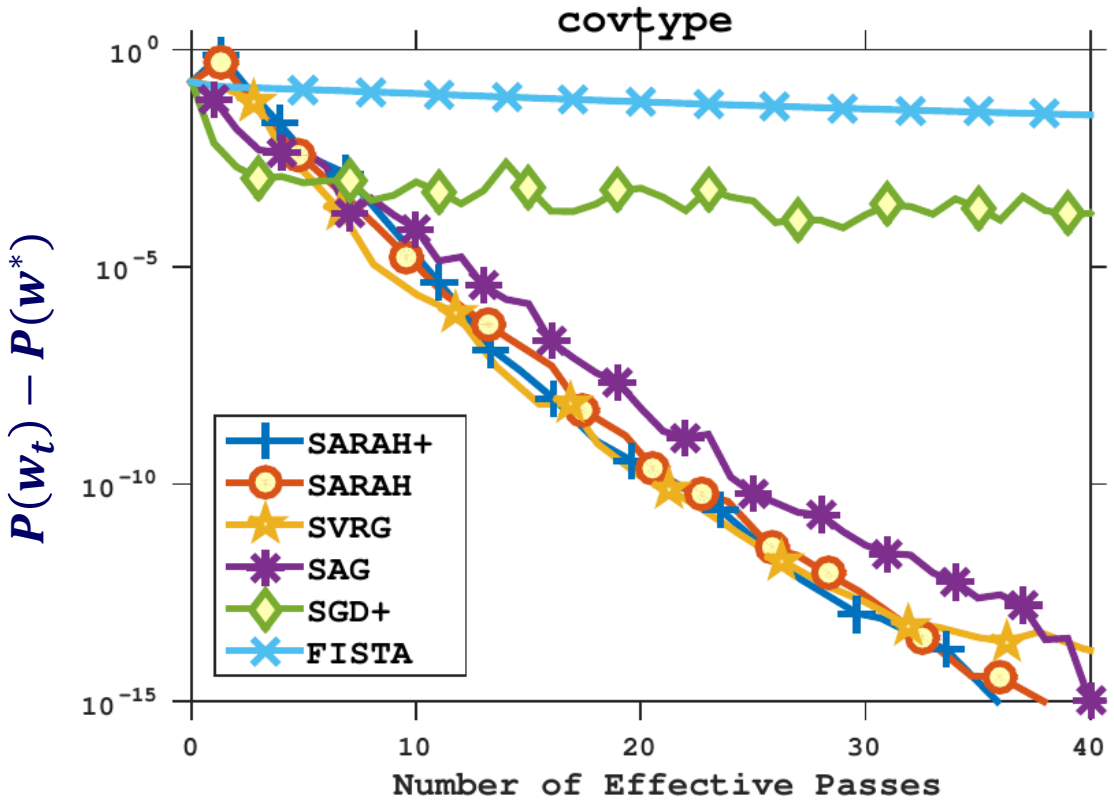


$\gamma \approx 1/10$

good performance
across many datasets

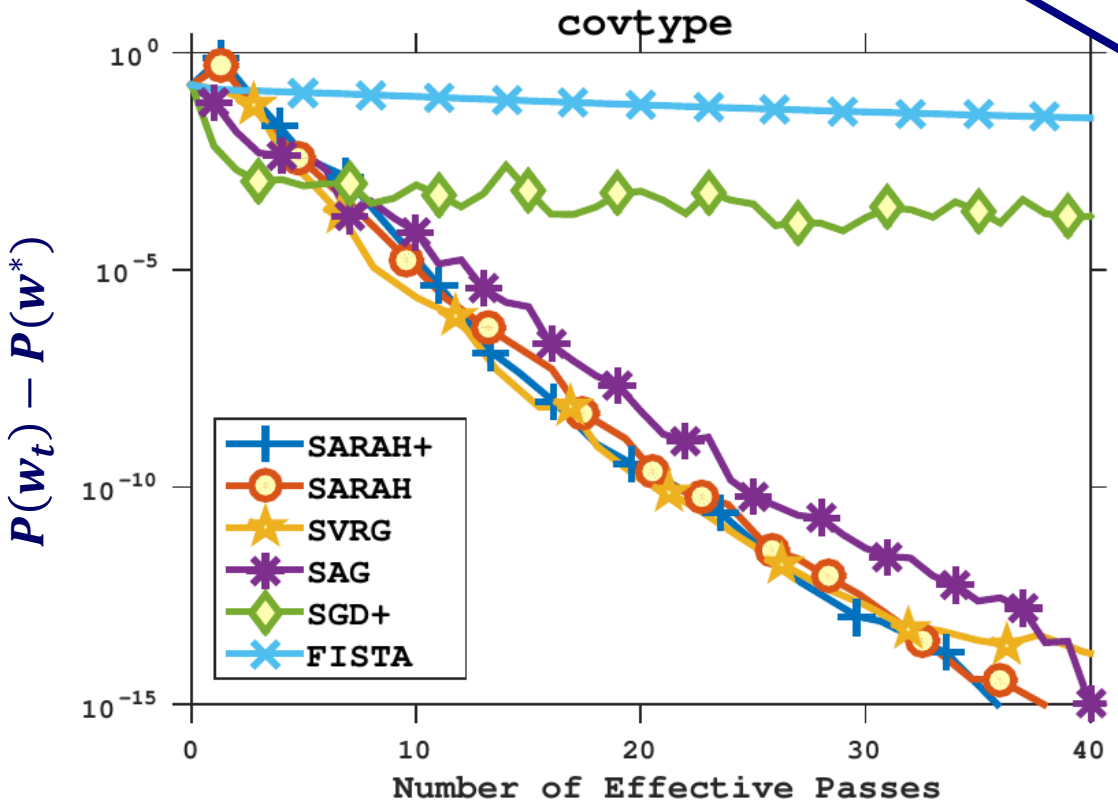
Numerical Experiments

Dataset	SARAH (m^*, η^*)	SVRG (m^*, η^*)	SAG (η^*)	SGD+ (η^*)	FISTA (η^*)
covtype	(2n, 0.9/L)	(n, 0.8/L)	0.3/L	0.06/L	50/L
ijcnn1	(0.5n, 0.8/L)	(n, 0.5/L)	0.7/L	0.1/L	90/L
news20	(0.5n, 0.9/L)	(n, 0.5/L)	0.1/L	0.2/L	30/L
rcv1	(0.7n, 0.7/L)	(0.5n, 0.9/L)	0.1/L	0.1/L	120/L



Numerical Experiments

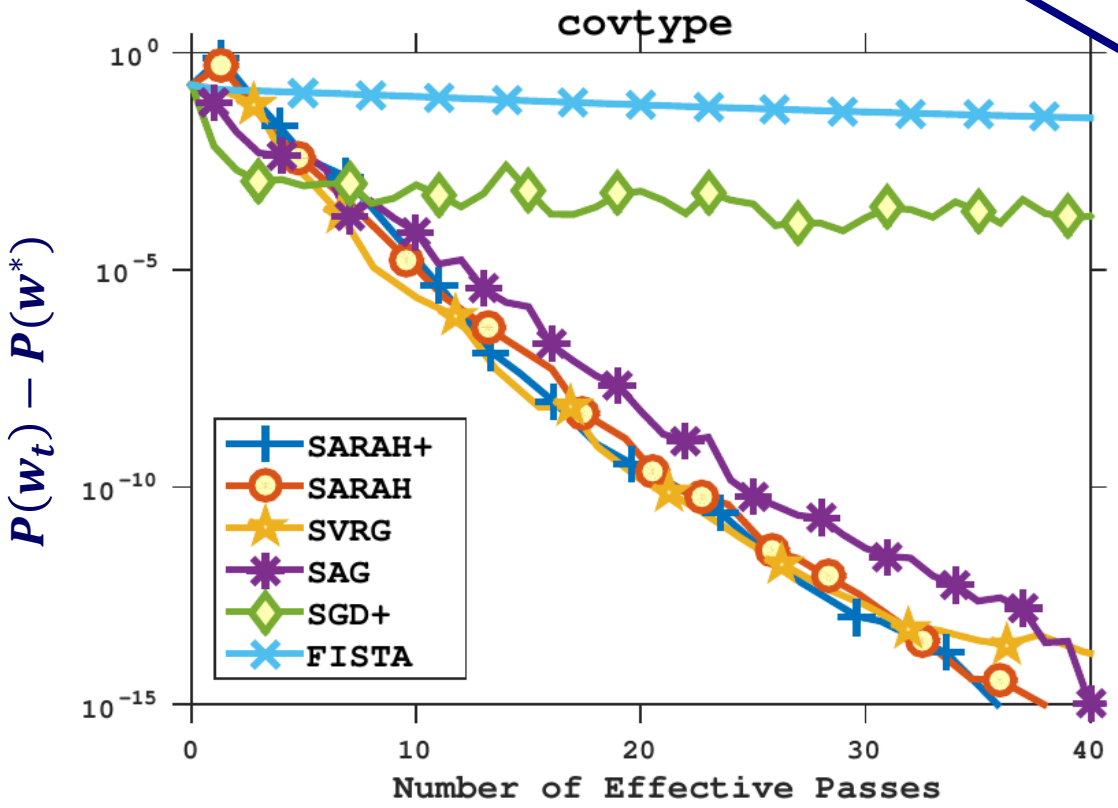
Dataset	SARAH (m^*, η^*)	SVRG (m^*, η^*)	SAG (η^*)	SGD+ (η^*)	FISTA (η^*)
covtype	(2n, 0.9/L)	(n, 0.8/L)	0.3/L	0.06/L	50/L
ijcnn1	(0.5n, 0.8/L)	(n, 0.5/L)	0.7/L	0.1/L	90/L
news20	(0.5n, 0.9/L)	(n, 0.5/L)	0.1/L	0.2/L	30/L
rcv1	(0.7n, 0.7/L)	(0.5n, 0.9/L)	0.1/L	0.1/L	120/L



One has to tune parameters to get a good performance!

Numerical Experiments

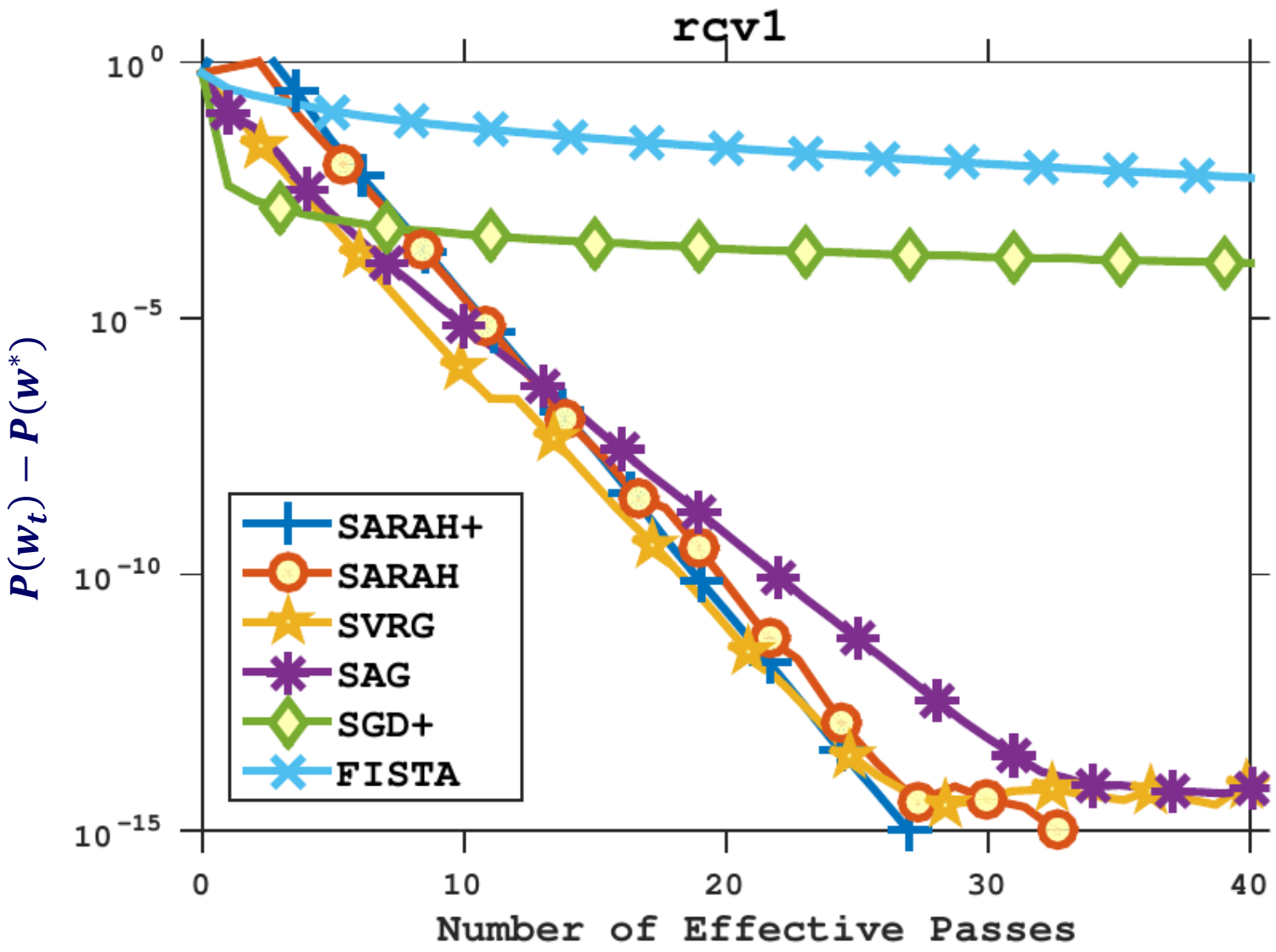
Dataset	SARAH (m^*, η^*)	SVRG (m^*, η^*)	SAG (η^*)	SGD+ (η^*)	FISTA (η^*)
covtype	(2n, 0.9/L)	(n, 0.8/L)	0.3/L	0.06/L	50/L
ijcnn1	(0.5n, 0.8/L)	(n, 0.5/L)	0.7/L	0.1/L	90/L
news20	(0.5n, 0.9/L)	(n, 0.5/L)	0.1/L	0.2/L	30/L
rcv1	(0.7n, 0.7/L)	(0.5n, 0.9/L)	0.1/L	0.1/L	120/L



One has to tune parameters to get a good performance!

Not for SARAH+!

Numerical Experiments



Convergence Rates Comparisons

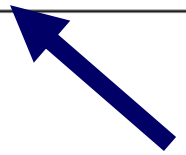
Strongly convex case: $\kappa = L/\mu$ is a condition number

Method	Complexity	Fixed Learning Rate	Low Storage Cost
GD	$\mathcal{O}\left(n\kappa \log\left(1/\epsilon\right)\right)$	✓	✓
SGD	$\mathcal{O}\left(1/\epsilon\right)$	✗	✓
SVRG	$\mathcal{O}\left((n+\kappa)\log\left(1/\epsilon\right)\right)$	✓	✓
SAG/SAGA	$\mathcal{O}\left((n+\kappa)\log\left(1/\epsilon\right)\right)$	✓	✗
SARAH	$\mathcal{O}\left((n+\kappa)\log\left(1/\epsilon\right)\right)$	✓	✓

Convergence Rates Comparisons

Strongly convex case: $\kappa = L/\mu$ is a condition number

Method	Complexity	Fixed Learning Rate	Low Storage Cost
GD	$\mathcal{O}\left(n\kappa \log\left(1/\epsilon\right)\right)$	✓	✓
SGD	$\mathcal{O}\left(1/\epsilon\right)$	✗	✓
SVRG	$\mathcal{O}\left((n+\kappa)\log\left(1/\epsilon\right)\right)$	✓	✓
SAG/SAGA	$\mathcal{O}\left((n+\kappa)\log\left(1/\epsilon\right)\right)$	✓	✗
SARAH	$\mathcal{O}\left((n+\kappa)\log\left(1/\epsilon\right)\right)$	✓	✓



Practical variant available

Convergence Rates Comparisons

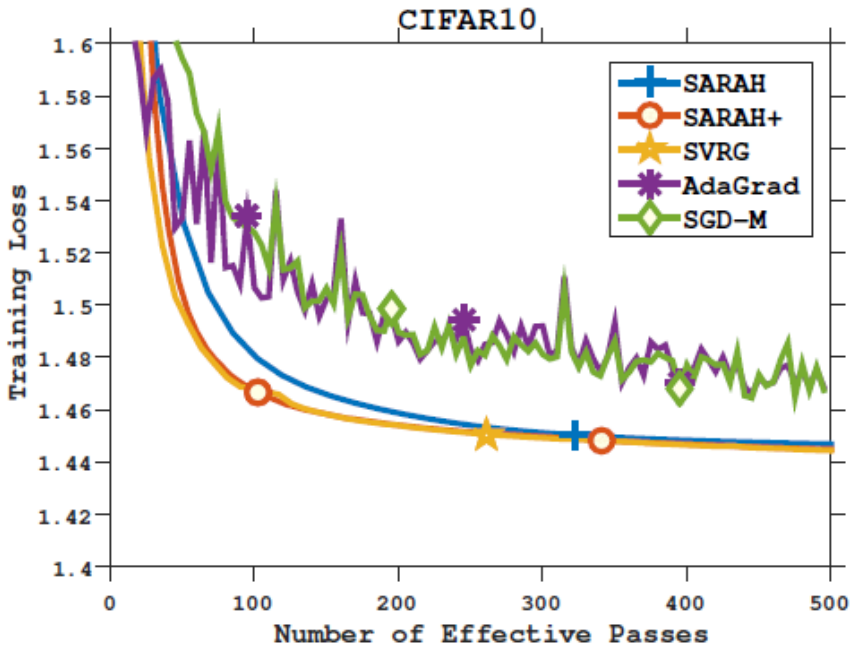
For smooth (general) convex functions

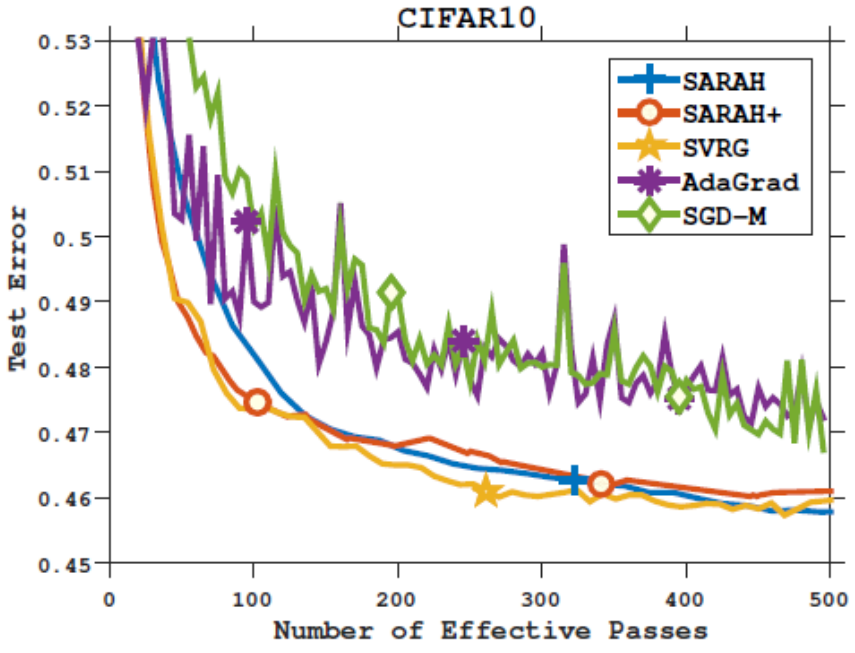
Method	Complexity
GD	$\mathcal{O}(n/\epsilon)$
SGD	$\mathcal{O}\left(1/\epsilon^2\right)$
SVRG	$\mathcal{O}(n+(\sqrt{n}/\epsilon))$
SAGA	$\mathcal{O}(n+(n/\epsilon))$
SARAH	$\mathcal{O}((n+(1/\epsilon))\log(1/\epsilon))$

For smooth nonconvex functions

Method	Complexity
GD	$\mathcal{O}\left(\frac{n}{\epsilon}\right)$
SGD	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2}\right)$
SVRG	$\mathcal{O}\left(n+\frac{n^{2/3}}{\nu\epsilon}\right)$
SARAH	$\mathcal{O}\left(n+\frac{1}{\epsilon^2}\right)$

Numerical Experiments





Convergence Rates Comparisons

For smooth (general) convex functions

Method	Complexity
GD	$\mathcal{O}(n/\epsilon)$
SGD	$\mathcal{O}\left(1/\epsilon^2\right)$
SVRG	$\mathcal{O}\left(n+(\sqrt{n}/\epsilon)\right)$
SAGA	$\mathcal{O}\left(n+(n/\epsilon)\right)$
SARAH	$\mathcal{O}\left((n+(1/\epsilon))\log(1/\epsilon)\right)$

For smooth nonconvex functions

Method	Complexity
GD	$\mathcal{O}\left(\frac{n}{\epsilon}\right)$
SGD	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2}\right)$
SVRG	$\mathcal{O}\left(n+\frac{n^{2/3}}{\nu\epsilon}\right)$
SARAH	$\mathcal{O}\left(n+\frac{1}{\epsilon^2}\right)$

SARAH converges to ϵ-accurate solution within a single outer loop with fixed "small" learning rate for nonconvex case (Results and Proofs in the papers)

Convergence Rates Comparisons

For smooth (general) convex functions

Method	Complexity
GD	$\mathcal{O}(n/\epsilon)$
SGD	$\mathcal{O}(1/\epsilon^2)$
SVRG	$\mathcal{O}(n + (\sqrt{n}/\epsilon))$
SAGA	$\mathcal{O}(n + (n/\epsilon))$
SARAH	$\mathcal{O}((n + (1/\epsilon))\log(1/\epsilon))$

For smooth nonconvex functions

Method	Complexity
GD	$\mathcal{O}\left(\frac{n}{\epsilon}\right)$
SGD	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2}\right)$
SVRG	$\mathcal{O}\left(n + \frac{n^{2/3}}{\nu\epsilon}\right)$
SARAH	$\mathcal{O}\left(n + \frac{1}{\epsilon^2}\right)$

SARAH converges to ϵ-accurate solution within a single outer loop with fixed "small" learning rate for nonconvex case (Results and Proofs in the papers)

Can we get rid of dependence on "*n*"?

Convergence Rates Comparisons

For smooth (general) convex functions

Method	Complexity
GD	$\mathcal{O}(n/\epsilon)$
SGD	$\mathcal{O}(1/\epsilon^2)$
SVRG	$\mathcal{O}(n+(\sqrt{n}/\epsilon))$
SAGA	$\mathcal{O}(n+(n/\epsilon))$
SARAH	$\mathcal{O}((n+(1/\epsilon))\log(1/\epsilon))$

For smooth nonconvex functions

Method	Complexity
GD	$\mathcal{O}\left(\frac{n}{\epsilon}\right)$
SGD	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2}\right)$
SVRG	$\mathcal{O}\left(n+\frac{n^{2/3}}{\nu\epsilon}\right)$
SARAH	$\mathcal{O}\left(n+\frac{1}{\epsilon^2}\right)$

SARAH converges to ϵ-accurate solution within a single outer loop with fixed "small" learning rate for nonconvex case (Results and Proofs in the papers)

Can we get rid of dependence on "*n*"?

NOT computing Full gradient

iSARAH

- L. Nguyen, K. Scheinberg, and M. Takac. **Inexact SARAH for Large Scale Machine Learning Problems**. *In preparation.*

Inexact SARAH (iSARAH)

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  Choose a subset $I \subseteq [n]$ of size b uniformly at random (without replacement)
  $v_0 = \frac{1}{b} \sum_{i \in I} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w} = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
  Output: $\tilde{w}_s = \tilde{w}$
end for

Inexact SARAH (iSARAH)

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  Choose a subset $I \subseteq [n]$ of size b uniformly at random (without replacement)
  $v_0 = \frac{1}{b} \sum_{i \in I} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w} = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
  Output: $\tilde{w}_s = \tilde{w}$
end for

NOT computing Full gradient

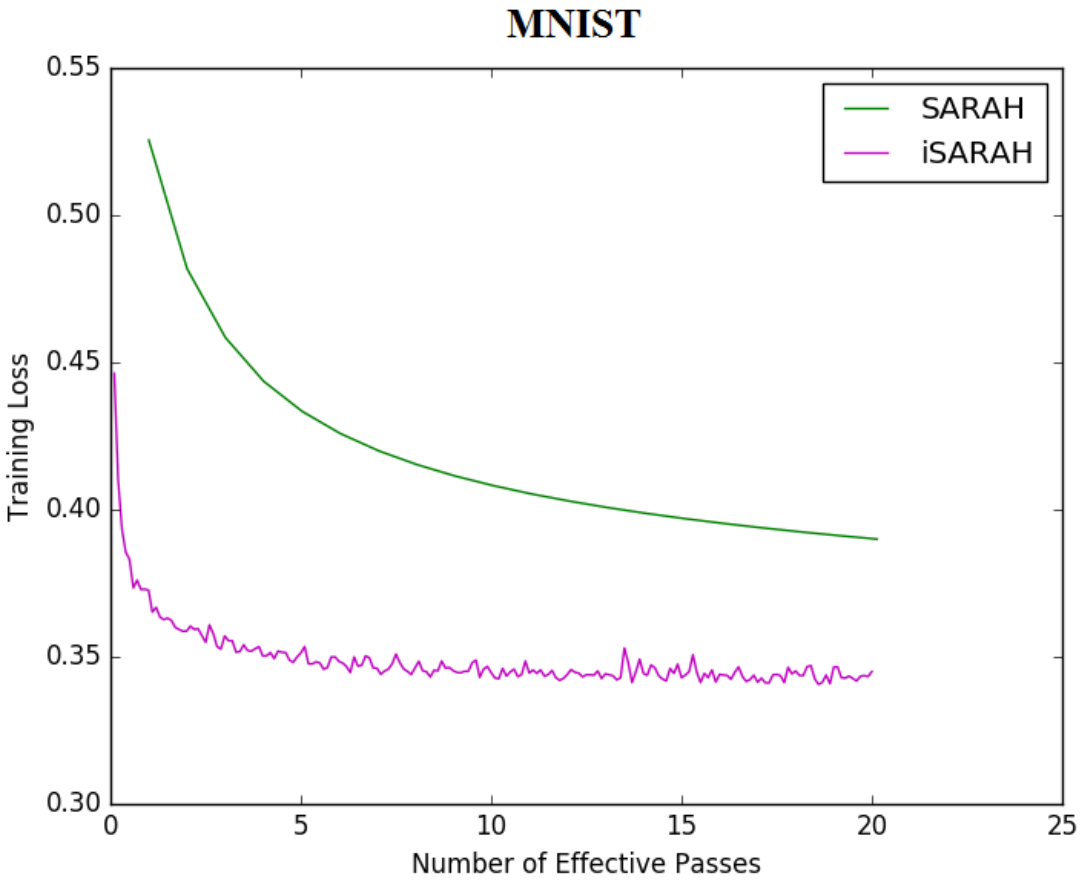
Inexact SARAH (iSARAH)

Parameters: the learning rate $\eta > 0$ and the inner loop size m.
Initialize: $\tilde{w}_0$
Iterate:
for $s = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{s-1}$
  Choose a subset $I \subseteq [n]$ of size b uniformly at random (without replacement)
  $v_0 = \frac{1}{b}\sum_{i \in I} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  Iterate:
  for $t = 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random from $[n]$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  Set $\tilde{w} = w_t$ with t chosen uniformly at random from $\{0, 1, \ldots, m\}$
  Output: $\tilde{w}_s = \tilde{w}$
end for

NOT computing Full gradient

- **For smooth general convex functions:** $b = m$
- **For smooth nonconvex functions:** $b = \sqrt{m}$

Numerical Experiments



Convergence Rates Comparisons

For smooth (general) convex functions

Method	Complexity
GD	$\mathcal{O}\left(n/\epsilon\right)$
SGD	$\mathcal{O}\left(1/\epsilon^2\right)$
SVRG	$\mathcal{O}\left(n+\left(\sqrt{n}/\epsilon\right)\right)$
SAGA	$\mathcal{O}\left(n+\left(n/\epsilon\right)\right)$
SARAH	$\mathcal{O}\left(\left(n+\left(1/\epsilon\right)\right)\log(1/\epsilon)\right)$
iSARAH	$\mathcal{O}\left(\left(1/\epsilon\right)\log(1/\epsilon)\right)$

For smooth nonconvex functions

Method	Complexity
GD	$\mathcal{O}\left(\frac{n}{\epsilon}\right)$
SGD	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2}\right)$
SVRG	$\mathcal{O}\left(n+\frac{n^{2/3}}{\nu\epsilon}\right)$
SARAH	$\mathcal{O}\left(n+\frac{1}{\epsilon^2}\right)$
iSARAH	$\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$

Convergence Rates Comparisons

For smooth (general) convex functions

Method	Complexity
GD	$\mathcal{O}(n/\epsilon)$
SGD	$\mathcal{O}(1/\epsilon^2)$
SVRG	$\mathcal{O}(n + (\sqrt{n}/\epsilon))$
SAGA	$\mathcal{O}(n + (n/\epsilon))$
SARAH	$\mathcal{O}((n + (1/\epsilon))\log(1/\epsilon))$
iSARAH	$\mathcal{O}((1/\epsilon)\log(1/\epsilon))$

For smooth nonconvex functions

Method	Complexity
GD	$\mathcal{O}\left(\frac{n}{\epsilon}\right)$
SGD	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2}\right)$
SVRG	$\mathcal{O}\left(n + \frac{n^{2/3}}{\nu\epsilon}\right)$
SARAH	$\mathcal{O}\left(n + \frac{1}{\epsilon^2}\right)$
iSARAH	$\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$

Total complexity of iSARAH does not depend on "*n*"

Convergence Rates Comparisons

For smooth (general) convex functions

Method	Complexity
GD	$\mathcal{O}(n/\epsilon)$
SGD	$\mathcal{O}(1/\epsilon^2)$
SVRG	$\mathcal{O}(n + (\sqrt{n}/\epsilon))$
SAGA	$\mathcal{O}(n + (n/\epsilon))$
SARAH	$\mathcal{O}((n + (1/\epsilon))\log(1/\epsilon))$
iSARAH	$\mathcal{O}((1/\epsilon)\log(1/\epsilon))$

For smooth nonconvex functions

Method	Complexity
GD	$\mathcal{O}\left(\frac{n}{\epsilon}\right)$
SGD	$\mathcal{O}\left(\frac{\sigma^2}{\epsilon^2}\right)$
SVRG	$\mathcal{O}\left(n + \frac{n^{2/3}}{\nu\epsilon}\right)$
SARAH	$\mathcal{O}\left(n + \frac{1}{\epsilon^2}\right)$
iSARAH	$\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$

Total complexity of iSARAH does not depend on "*n*"

VERY USEFUL for large scale machine learning problems!!!

SARAH

THANK YOU !!!

LamNguyen.MLTD@gmail.com

Lam M. Nguyen – Lehigh University

http://coral.ise.lehigh.edu/lmn214/