Linearly-Convergent Stochastic Gradient Algorithms

Francis Bach

INRIA - Ecole Normale Supérieure, Paris, France

Joint work with M. Schmidt, N. Le Roux, A. Defazio, S. Lacoste-Julien, P. Balamurugan

SIAM Conference on Imaging Science, June 2018

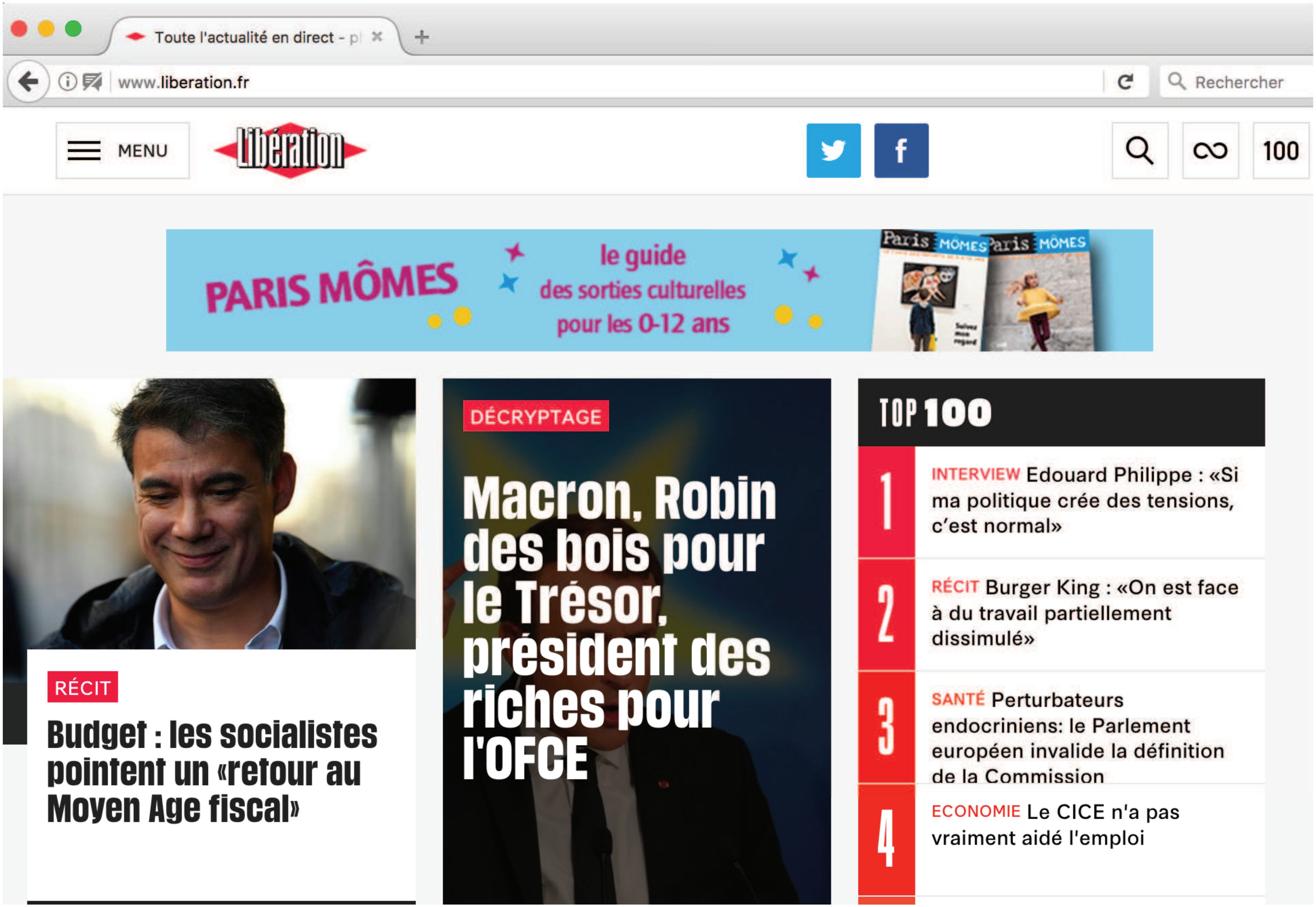
Context

Machine learning for large-scale data

- **Large-scale supervised machine learning**: **large d, large n**
  - d : dimension of each observation (input) or number of parameters
  - n : number of observations

- **Examples**: computer vision, advertising, bioinformatics, etc.

Advertising



Visual object recognition

Context
Machine learning for large-scale data

- **Large-scale supervised machine learning**: **large d, large n**
  - d : dimension of each observation (input), or number of parameters
  - n : number of observations

- **Examples**: computer vision, advertising, bioinformatics, etc.

- **Ideal running-time complexity**: $O(dn)$

Context
Machine learning for large-scale data

- **Large-scale supervised machine learning**: **large d, large n**
  - d : dimension of each observation (input), or number of parameters
  - n : number of observations
- **Examples**: computer vision, advertising, bioinformatics, etc.
- **Ideal running-time complexity**: $O(dn)$
- **Going back to simple methods**
  - Stochastic gradient methods (Robbins and Monro, 1951)
- **Goal: Present recent progress**

Outline

1. **Introduction/motivation: Supervised machine learning**
   – Optimization of finite sums
   – Existing optimization methods for finite sums

2. **Stochastic average gradient (SAG)**
   – Linearly-convergent stochastic gradient method
   – Precise convergence rates

3. **Extensions**
   – Link with variance reduction
   – Acceleration
   – Saddle-point problems

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$

- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$

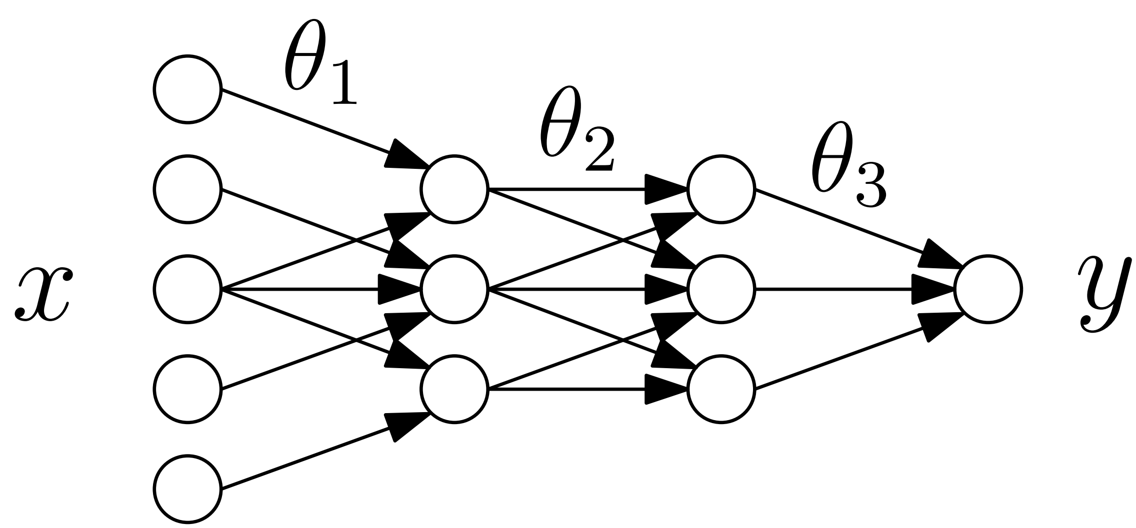
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

- **Motivating examples**

  – Linear predictions: $h(x, \theta) = \theta^\top \Phi(x)$ with features $\Phi(x) \in \mathbb{R}^d$

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **Motivating examples**
  - Linear predictions: $h(x, \theta) = \theta^\top \Phi(x)$ with features $\Phi(x) \in \mathbb{R}^d$
  - Neural networks: $h(x, \theta) = \theta_m^\top \sigma(\theta_{m-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x))$



Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \ \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta)$$

data fitting term + regularizer

Usual losses

- **Regression**: $y \in \mathbb{R}$
  - Quadratic loss $\ell\big(y, h(x,\theta)\big) = \frac{1}{2}(y - h(x,\theta))^2$

Usual losses

- **Regression**: $y \in \mathbb{R}$
  - Quadratic loss $\ell\big(y, h(x, \theta)\big) = \frac{1}{2}(y - h(x, \theta))^2$
- **Classification** : $y \in \{-1, 1\}$
  - Logistic loss $\ell\big(y, h(x, \theta)\big) = \log(1 + \exp(-yh(x, \theta)))$

Usual losses

- **Regression**: $y \in \mathbb{R}$
  - Quadratic loss $\ell\big(y, h(x,\theta)\big) = \frac{1}{2}(y - h(x,\theta))^2$

- **Classification** : $y \in \{-1, 1\}$
  - Logistic loss $\ell\big(y, h(x,\theta)\big) = \log(1 + \exp(-yh(x,\theta)))$

- **Structured prediction**
  - Complex outputs y (k classes/labels, graphs, trees, or $\{0,1\}^k$, etc.)
  - Prediction function $h(x,\theta) \in \mathbb{R}^{k}$
  - Conditional random fields (Lafferty et al., 2001)
  - Max-margin (Taskar et al., 2003; Tsochantaridis et al., 2005)

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \quad \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta)$$

data fitting term $+$ regularizer

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \Big\{ \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta) \Big\} \qquad = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

data fitting term + regularizer

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \Big\{ \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta) \Big\} \qquad = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

data fitting term + regularizer

- Optimization: optimization of regularized risk training cost

Parametric supervised machine learning

- **Data**: n observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \Big\{ \ell\big(y_i, h(x_i, \theta)\big) \quad + \quad \lambda \Omega(\theta) \Big\} \qquad = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

data fitting term $+$ regularizer

- Optimization: optimization of regularized risk — training cost
- Statistics: guarantees on $\mathbb{E}_{p(x,y)} \ell(y, h(x, \theta))$ — testing cost

Finite sums in signal/image processing

- **Model fitting**
  - *Same optimization problem*: $\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) + \lambda \Omega(\theta)$

Finite sums in signal/image processing

- **Model fitting**
  - *Same optimization problem*: $\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) + \lambda \Omega(\theta)$
  - *Differences:* (1) Typically need high precision for θ
    (2) Data (x_i, y_i) may not be i.i.d.

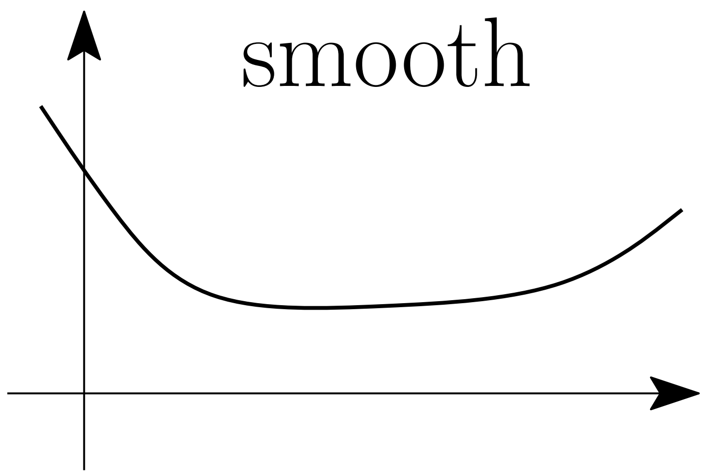
Finite sums in signal/image processing

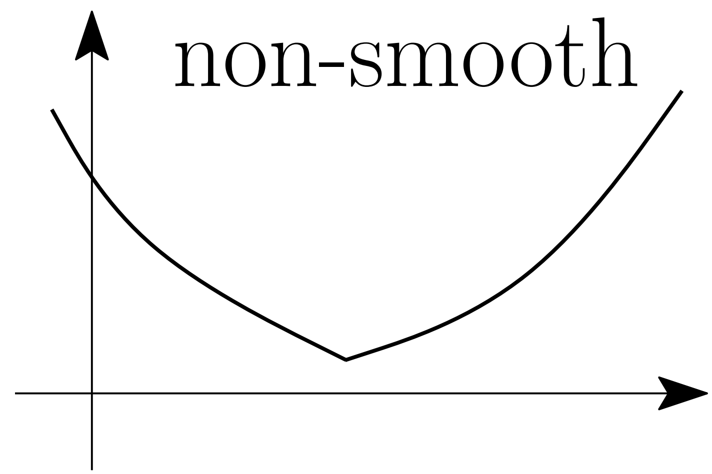
- **Model fitting**
  - *Same optimization problem*: $\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) + \lambda \Omega(\theta)$
  - *Differences:* (1) Typically need high precision for θ
    (2) Data (x_i, y_i) may not be i.i.d.
- **Structured regularization**
  - E.g., total variation $\sum_{i \sim j} |\theta_i - \theta_j|$

Smoothness and (strong) convexity

- A function $g : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if and only if it is twice differentiable and

$$\forall \theta \in \mathbb{R}^d, \ \left|\text{eigenvalues}\left[g''(\theta)\right]\right| \leqslant L$$





Smoothness and (strong) convexity

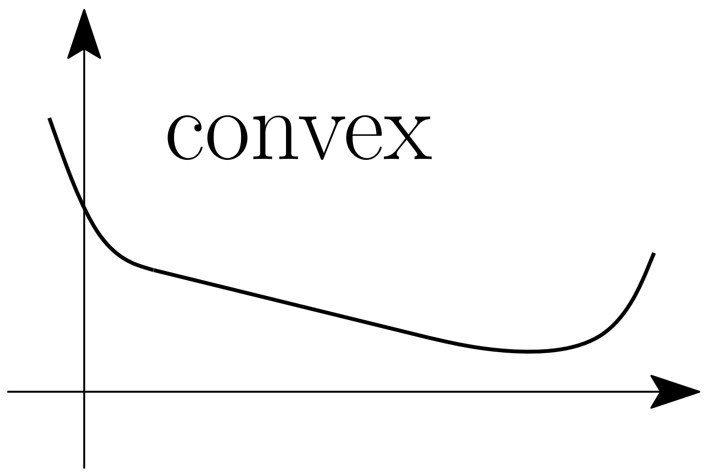
- A function $g : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if and only if it is twice differentiable and

$$\forall \theta \in \mathbb{R}^d, \ \big|\text{eigenvalues}\big[g''(\theta)\big]\big| \leqslant L$$

- **Machine learning**
  - with $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i, \theta))$
  - Smooth prediction function $\theta \mapsto h(x_i, \theta)$ + smooth loss

Smoothness and (strong) convexity

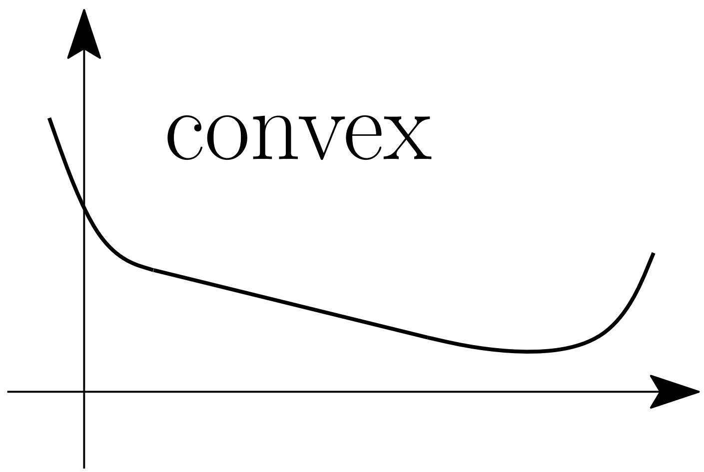
- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if
$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant 0$$

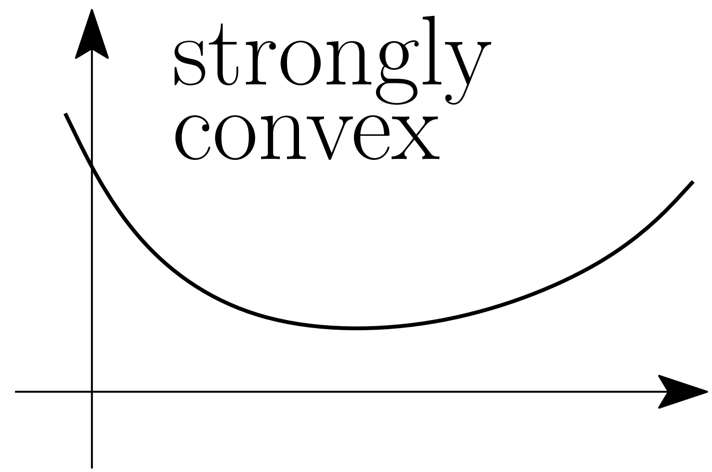


Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \ \text{eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$



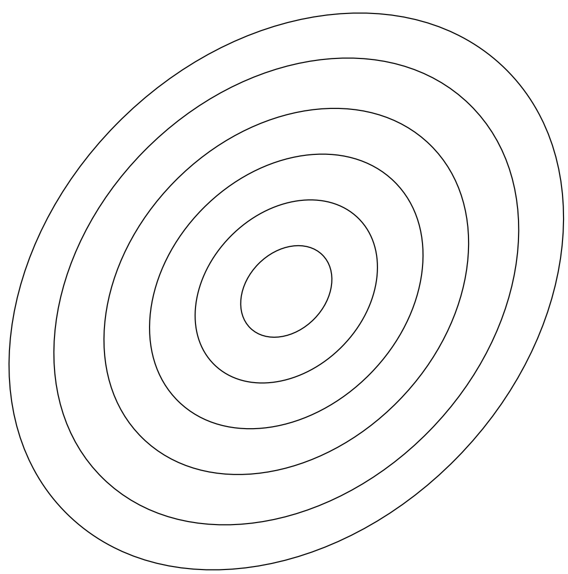


Smoothness and (strong) convexity

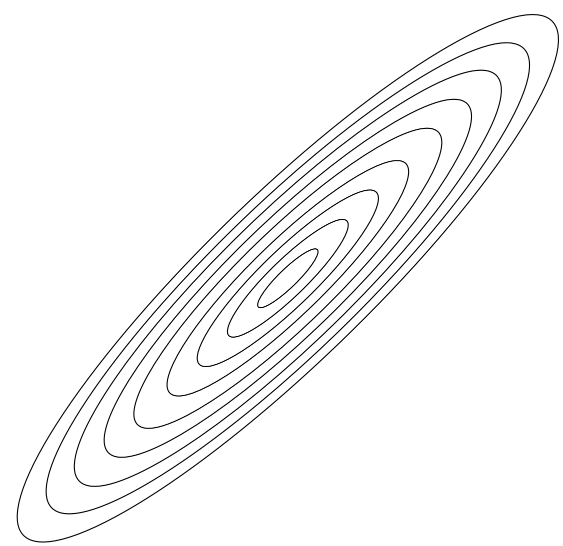
- A twice differentiable function $g : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

  – Condition number $\kappa = L/\mu \geqslant 1$



(small $\kappa = L/\mu$)



(large $\kappa = L/\mu$)

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \ \text{eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Convexity in machine learning**
  - With $g(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Convexity in machine learning**
  - With $g(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$

- **Relevance of convex optimization**
  - Easier design and analysis of algorithms
  - Global minimum vs. local minimum vs. stationary points
  - Gradient-based algorithms only need convexity for their analysis

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ-strongly convex if and only if
$$\forall \theta \in \mathbb{R}^d, \ \text{eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Strong convexity in machine learning**
  - With $g(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Strongly convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if
$$\forall \theta \in \mathbb{R}^d, \ \text{eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Strong convexity in machine learning**
  - With $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i, \theta))$
  - Strongly convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$
  - Invertible covariance matrix $\frac{1}{n}\sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^\top \Rightarrow n \geqslant d$
  - Even when $\mu > 0$, μ may be arbitrarily small!

Smoothness and (strong) convexity

- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if and only if

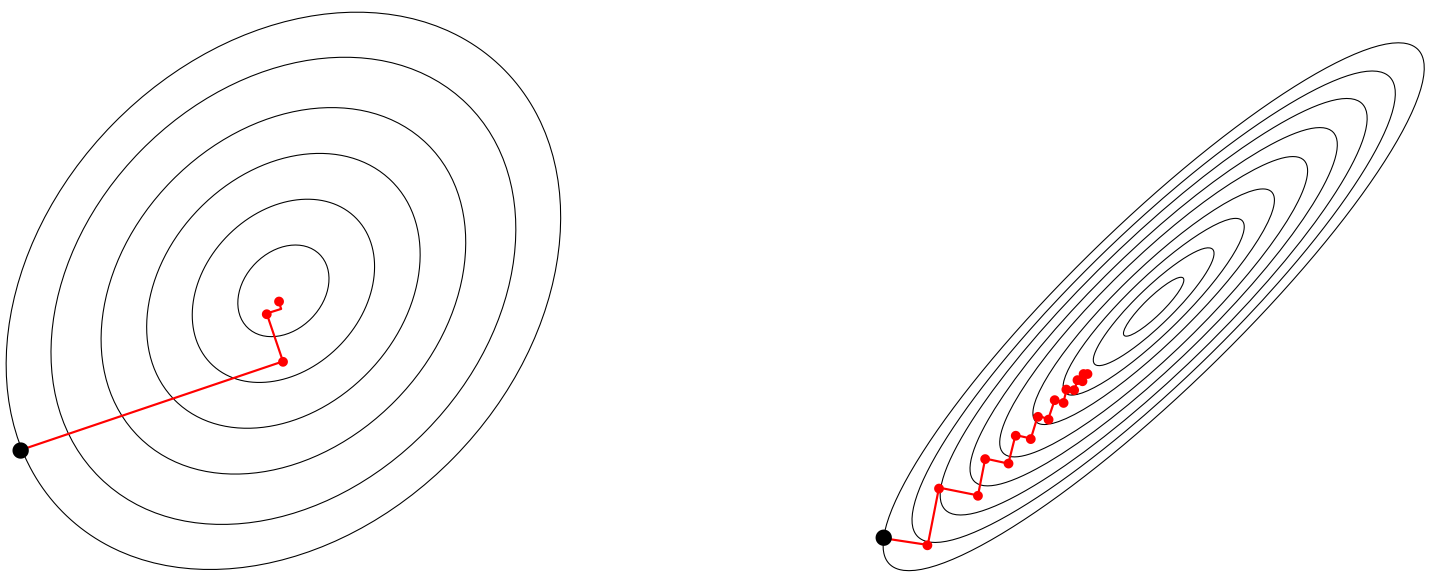
$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant \mu$$

- **Strong convexity in machine learning**
  - With $g(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Strongly convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$
  - Invertible covariance matrix $\frac{1}{n}\sum_{i=1}^n \Phi(x_i)\Phi(x_i)^\top \Rightarrow n \geqslant d$
  - Even when $\mu > 0$, μ may be arbitrarily small!

- **Adding regularization by $\frac{\mu}{2}\|\theta\|^2$**
  - creates additional bias unless μ is small, but reduces variance
  - Typically $L/\sqrt{n} \geqslant \mu \geqslant L/n$

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$



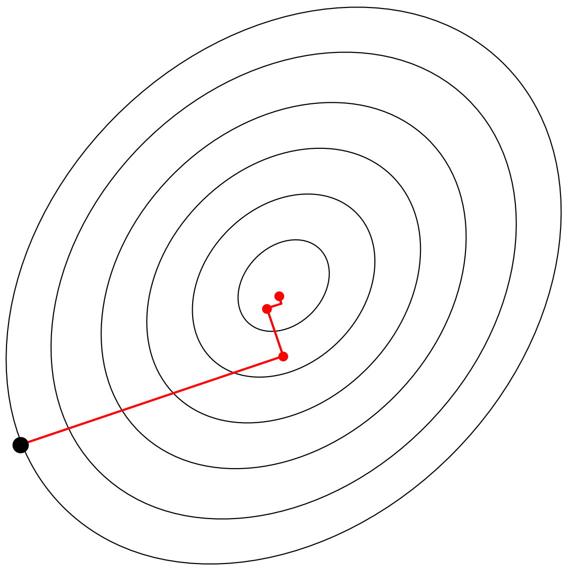
(small $\kappa = L/\mu$) (large $\kappa = L/\mu$)

Iterative methods for minimizing smooth functions

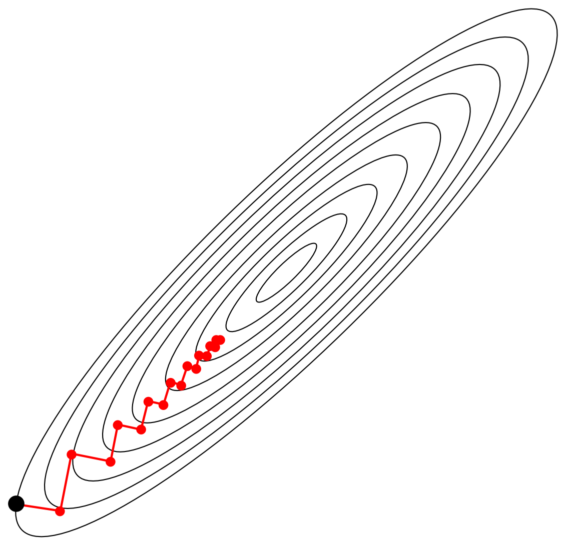
- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$

$g(\theta_t) - g(\theta_*) \leqslant O(1/t)$

$g(\theta_t) - g(\theta_*) \leqslant O((1-\mu/L)^t) = O(e^{-t(\mu/L)})$ if μ-strongly convex



(small $\kappa = L/\mu$)



(large $\kappa = L/\mu$)

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$

- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex
- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$

- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ $O(\kappa \log \frac{1}{\varepsilon})$ iterations

- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ $O(\log \log \frac{1}{\varepsilon})$ iterations

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$

- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$

- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ complexity $= O((nd^2 + d^3) \cdot \log\log \frac{1}{\varepsilon})$

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$
- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$
- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ complexity $= O((nd^2 + d^3) \cdot \log\log \frac{1}{\varepsilon})$
- **Key insights for machine learning (Bottou and Bousquet, 2008)**

1. No need to optimize below statistical error
2. Cost functions are averages
3. Testing error is more important than training error

Iterative methods for minimizing smooth functions

- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$

- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$
  - $O(1/t)$ convergence rate for convex functions
  - $O(e^{-t/\kappa})$ *linear* if strongly-convex $\Leftrightarrow$ complexity $= O(nd \cdot \kappa \log \frac{1}{\varepsilon})$

- **Newton method**: $\theta_t = \theta_{t-1} - g''(\theta_{t-1})^{-1} g'(\theta_{t-1})$
  - $O\big(e^{-\rho 2^t}\big)$ *quadratic* rate $\Leftrightarrow$ complexity $= O((nd^2 + d^3) \cdot \log \log \frac{1}{\varepsilon})$

- **Key insights for machine learning (Bottou and Bousquet, 2008)**
  1. No need to optimize below statistical error
  2. Cost functions are averages
  3. Testing error is more important than training error

Stochastic gradient descent (SGD) for finite sums

$$\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

- **Iteration**: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$
  - Sampling with replacement: $i(t)$ random element of $\{1, \dots, n\}$
  - Polyak-Ruppert averaging: $\bar{\theta}_t = \frac{1}{t+1} \sum_{u=0}^{t} \theta_u$

Stochastic gradient descent (SGD) for finite sums

$$\min_{\theta \in \mathbb{R}^d} g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)$$

- **Iteration**: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$
  - Sampling with replacement: $i(t)$ random element of $\{1, \dots, n\}$
  - Polyak-Ruppert averaging: $\bar{\theta}_t = \frac{1}{t+1} \sum_{u=0}^{t} \theta_u$

- **Convergence rate** if each f_i is convex L-smooth and g μ-strongly-convex:

$$\mathbb{E} g(\bar{\theta}_t) - g(\theta_*) \leqslant \begin{cases} O(1/\sqrt{t}) & \text{if } \gamma_t = 1/(L\sqrt{t}) \\ O(L/(\mu t)) = O(\kappa/t) & \text{if } \gamma_t = 1/(\mu t) \end{cases}$$

  - No adaptivity to strong-convexity in general
  - Running-time complexity: $O(d \cdot \kappa/\varepsilon)$

Outline

1. **Introduction/motivation: Supervised machine learning**
   - Optimization of finite sums
   - Existing optimization methods for finite sums

2. **Stochastic average gradient (SAG)**
   - Linearly-convergent stochastic gradient method
   - Precise convergence rates

3. **Extensions**
   - Link with variance reduction
   - Acceleration
   - Saddle-point problems

Stochastic vs. deterministic methods

- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$

Stochastic vs. deterministic methods

- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$

- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$

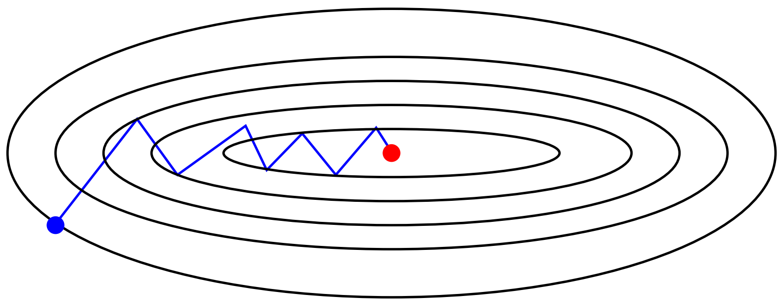
  – Linear (e.g., exponential) convergence rate in $O(e^{-t/\kappa})$

  – Iteration complexity is linear in n

Stochastic vs. deterministic methods

- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$

- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$



Stochastic vs. deterministic methods

- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$

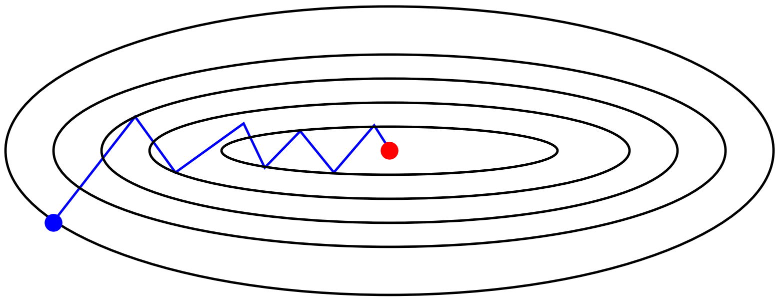
- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$
  - Linear (e.g., exponential) convergence rate in $O(e^{-t/\kappa})$
  - Iteration complexity is linear in n

- Stochastic gradient descent: $\theta_t = \theta_{t-1} - \gamma_t f_{i(t)}'(\theta_{t-1})$
  - Sampling with replacement: $i(t)$ random element of $\{1, \dots, n\}$
  - Convergence rate in $O(\kappa/t)$
  - Iteration complexity is independent of n

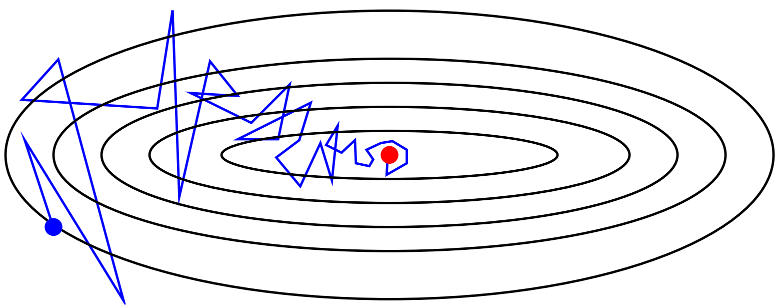
Stochastic vs. deterministic methods

- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i,\theta)\big) + \lambda\Omega(\theta)$

- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$

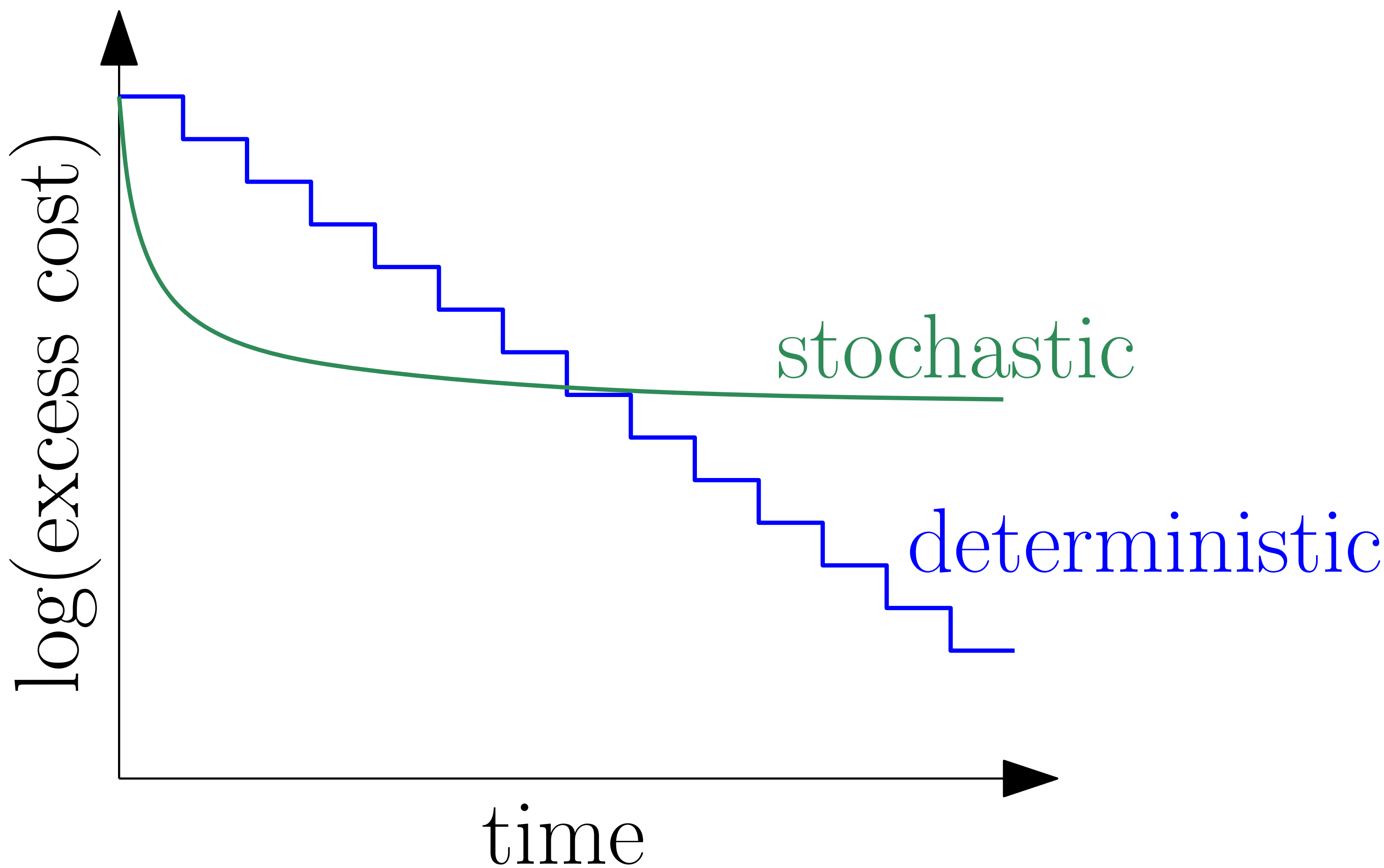


- Stochastic gradient descent: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$



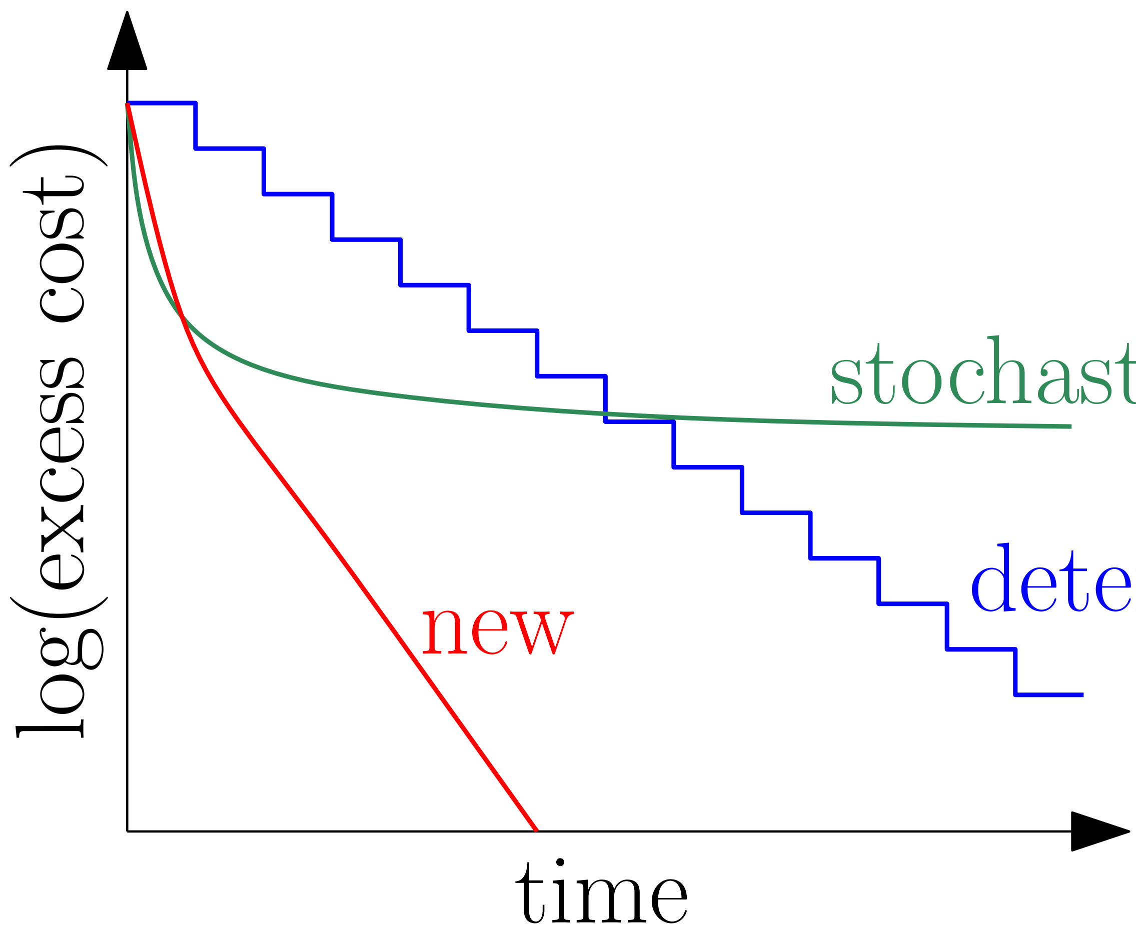
Stochastic vs. deterministic methods

- **Goal** = **best of both worlds**: Linear rate with $O(d)$ iteration cost
  Simple choice of step size



Stochastic vs. deterministic methods

- **Goal = best of both worlds**: Linear rate with $O(d)$ iteration cost
  Simple choice of step size



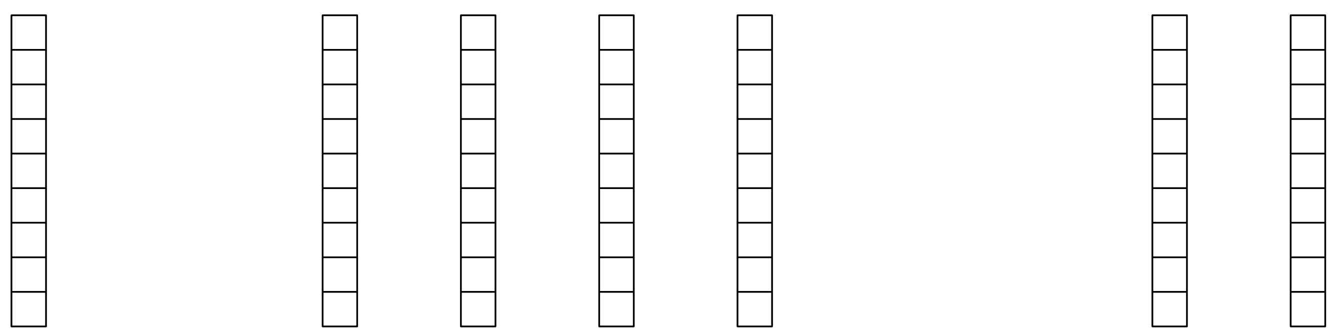
Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \dots, n$
  - Random selection $i(t) \in \{1, \dots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \frac{\gamma_t}{n} \sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$

Stochastic average gradient
(Le Roux, Schmidt, and Bach, 2012)

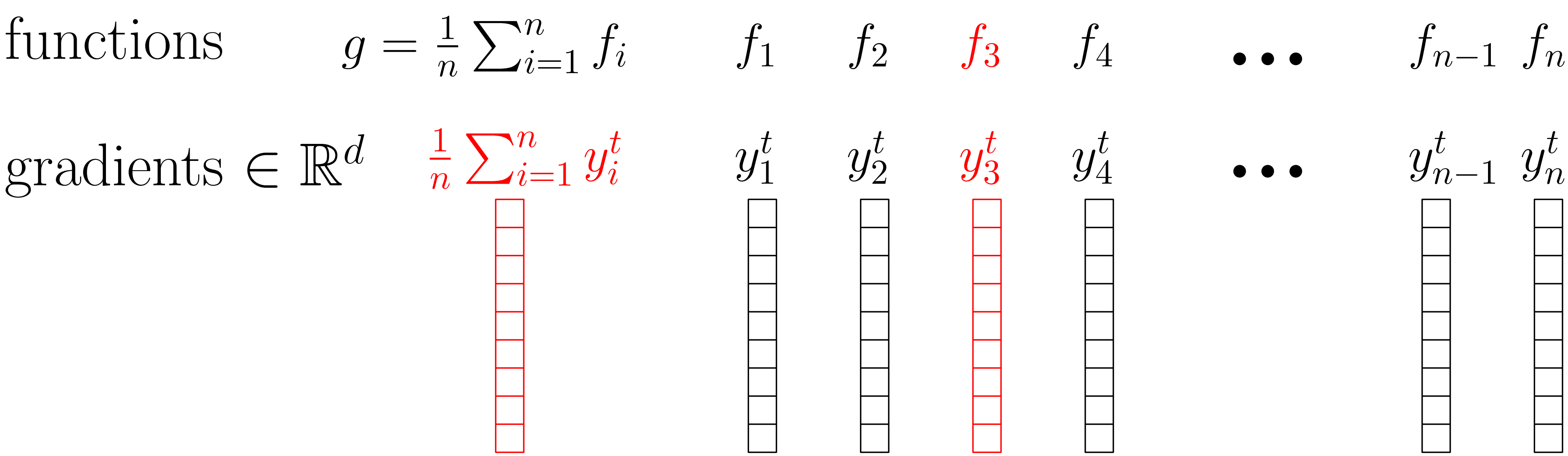
- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \dots, n$
  - Random selection $i(t) \in \{1, \dots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \dfrac{\gamma_t}{n} \sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$

functions	$g = \frac{1}{n}\sum_{i=1}^{n} f_i$	f_1	f_2	f_3	f_4	$\bullet\bullet\bullet$	f_{n-1}	f_n
gradients $\in \mathbb{R}^d$	$\frac{1}{n}\sum_{i=1}^{n} y_i^t$	y_1^t	y_2^t	y_3^t	y_4^t	$\bullet\bullet\bullet$	y_{n-1}^t	y_n^t



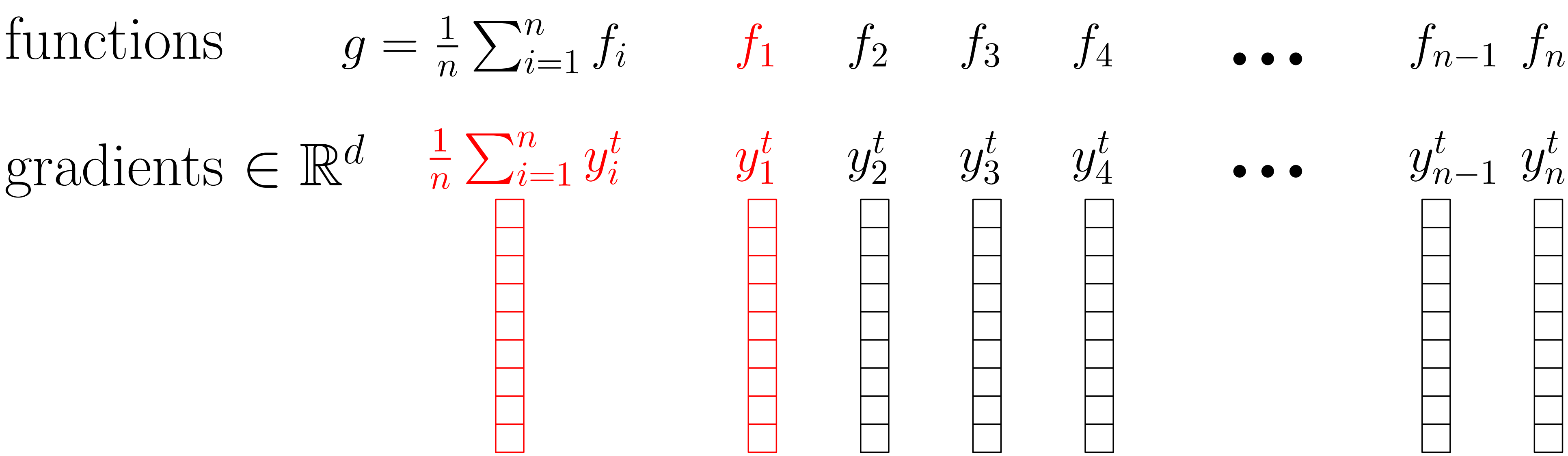
Stochastic average gradient
(Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \dots, n$
  - Random selection $i(t) \in \{1, \dots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \frac{\gamma_t}{n} \sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$



Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \dots, n$
  - Random selection $i(t) \in \{1, \dots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \frac{\gamma_t}{n} \sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$



Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \dots, n$
  - Random selection $i(t) \in \{1, \dots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \frac{\gamma_t}{n} \sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$
- Stochastic version of incremental average gradient (Blatt et al., 2008)

Stochastic average gradient
(Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient** (SAG) iteration
  - Keep in memory the gradients of all functions f_i, $i = 1, \dots, n$
  - Random selection $i(t) \in \{1, \dots, n\}$ with replacement
  - Iteration: $\theta_t = \theta_{t-1} - \frac{\gamma_t}{n} \sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$

- Stochastic version of incremental average gradient (Blatt et al., 2008)

- **Extra memory requirement**: n gradients in $\mathbb{R}^d$ in general

- **Linear supervised machine learning**: only n real numbers
  - If $f_i(\theta) = \ell(y_i, \Phi(x_i)^\top \theta)$, then $f_i'(\theta) = \ell'(y_i, \Phi(x_i)^\top \theta)\, \Phi(x_i)$

Running-time comparisons (strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each f_i convex L-smooth and g μ-strongly convex

Stochastic gradient descent	$d\times$	$\frac{L}{\mu}$	$\times$	$\frac{1}{\varepsilon}$
Gradient descent	$d\times$	$n\frac{L}{\mu}$	$\times \log$	$\frac{1}{\varepsilon}$
Accelerated gradient descent	$d\times$	$n\sqrt{\frac{L}{\mu}}$	$\times \log$	$\frac{1}{\varepsilon}$

Running-time comparisons (strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each f_i convex L-smooth and g μ-strongly convex

Stochastic gradient descent	$d\times$	$\frac{L}{\mu}$	$\times$	$\frac{1}{\varepsilon}$
Gradient descent	$d\times$	$n\frac{L}{\mu}$	$\times \log$	$\frac{1}{\varepsilon}$
Accelerated gradient descent	$d\times$	$n\sqrt{\frac{L}{\mu}}$	$\times \log$	$\frac{1}{\varepsilon}$
SAG	$d\times$	$(n + \frac{L}{\mu})$	$\times \log$	$\frac{1}{\varepsilon}$

  - NB-1: for (accelerated) gradient descent, $L =$ smoothness constant of g
  - NB-2: with non-uniform sampling, $L =$ average smoothness constants of all f_i's

Running-time comparisons (strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each f_i convex L-smooth and g μ-strongly convex

Stochastic gradient descent	$d\times$	$\frac{L}{\mu}$	$\times$	$\frac{1}{\varepsilon}$
Gradient descent	$d\times$	$n\frac{L}{\mu}$	$\times \log$	$\frac{1}{\varepsilon}$
Accelerated gradient descent	$d\times$	$n\sqrt{\frac{L}{\mu}}$	$\times \log$	$\frac{1}{\varepsilon}$
SAG	$d\times$	$(n+\frac{L}{\mu})$	$\times \log$	$\frac{1}{\varepsilon}$

- **Beating two lower bounds** (Nemirovski and Yudin, 1983; Nesterov, 2004): with additional assumptions

(1) stochastic gradient: exponential rate for finite sums
(2) full gradient: better exponential rate using the sum structure

Running-time comparisons (non-strongly-convex)

- **Assumptions**: $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$
  - Each f_i convex L-smooth
  - Ill conditioned problems: g may not be strongly-convex ($\mu = 0$)

Stochastic gradient descent	$d\times$	$1/\varepsilon^2$
Gradient descent	$d\times$	n/ε
Accelerated gradient descent	$d\times$	$n/\sqrt{\varepsilon}$
SAG	$d\times$	$\sqrt{n}/\varepsilon$

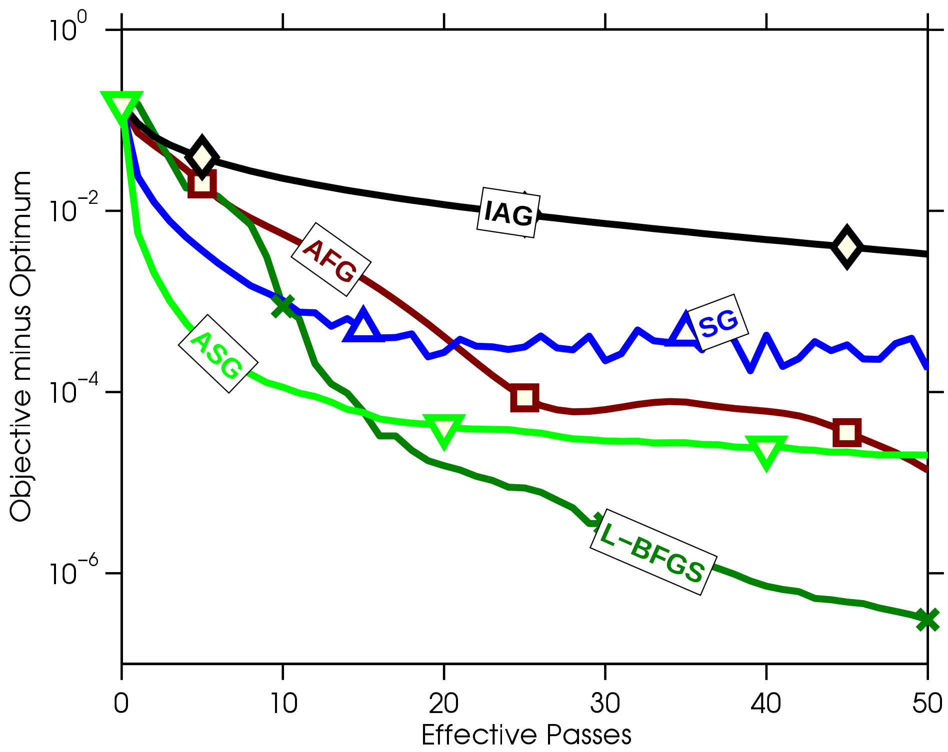
- Adaptivity to potentially hidden strong convexity
- No need to know the local/global strong-convexity constant

Stochastic average gradient
Implementation details and extensions

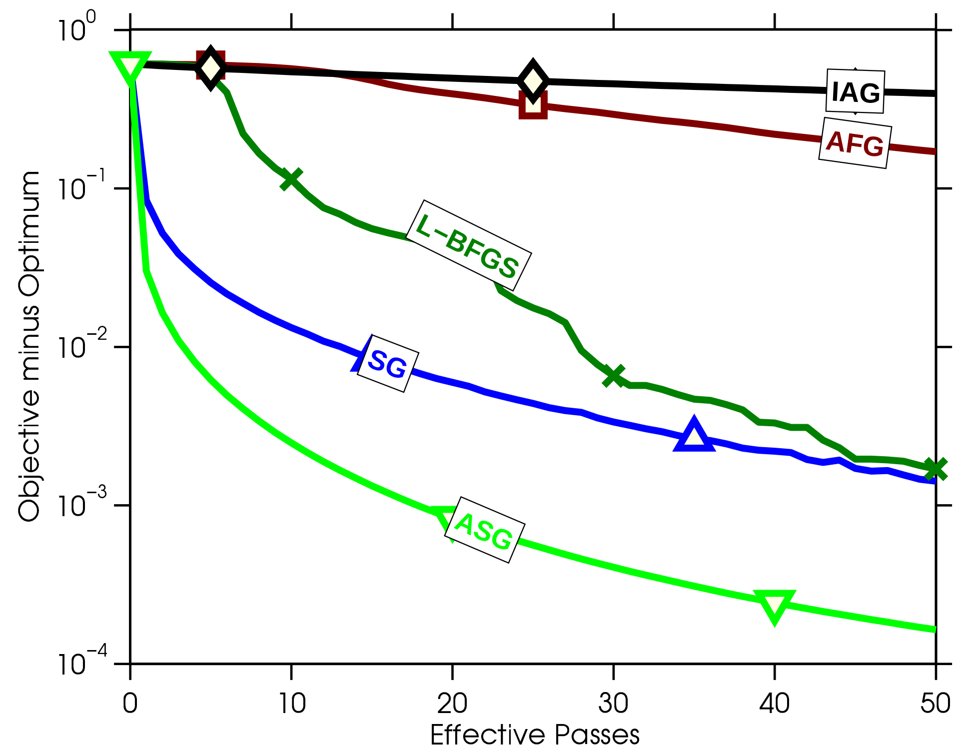
- **Sparsity in the features**
  - Just-in-time updates $\Rightarrow$ replace $O(d)$ by number of non zeros
  - See also Leblond, Pedregosa, and Lacoste-Julien (2016)
- **Mini-batches**
  - Reduces the memory requirement + block access to data
- **Line-search**
  - Avoids knowing L in advance
- **Non-uniform sampling**
  - Favors functions with large variations
- See `www.cs.ubc.ca/~schmidtm/Software/SAG.html`

Experimental results (logistic regression)

`quantum` dataset
$(n = 50\ 000,\ d = 78)$



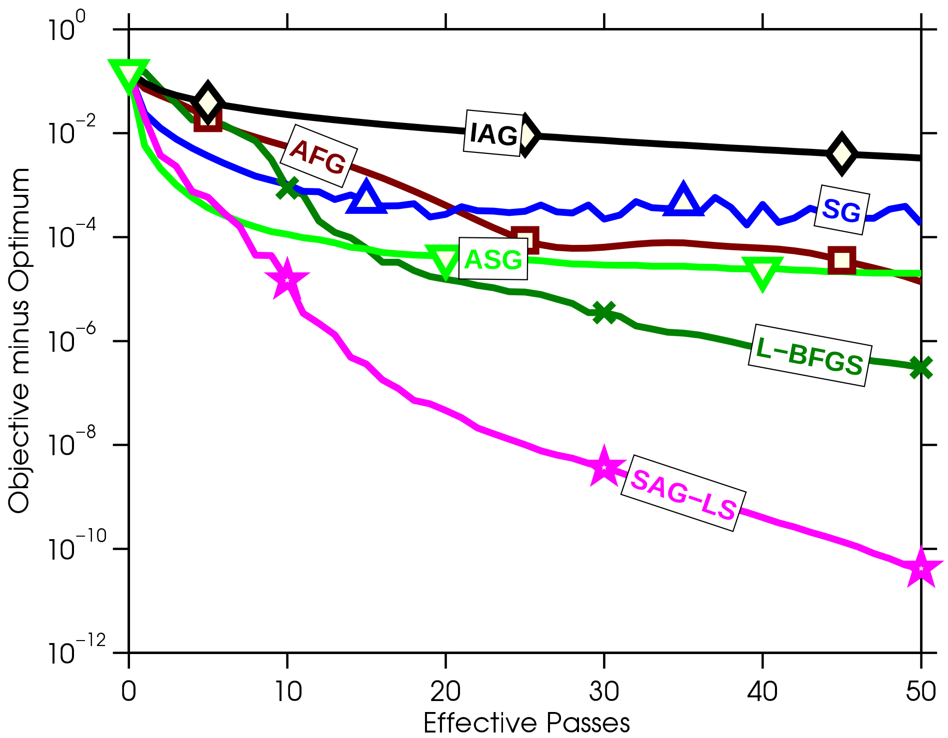
`rcv1` dataset
$(n = 697\ 641,\ d = 47\ 236)$

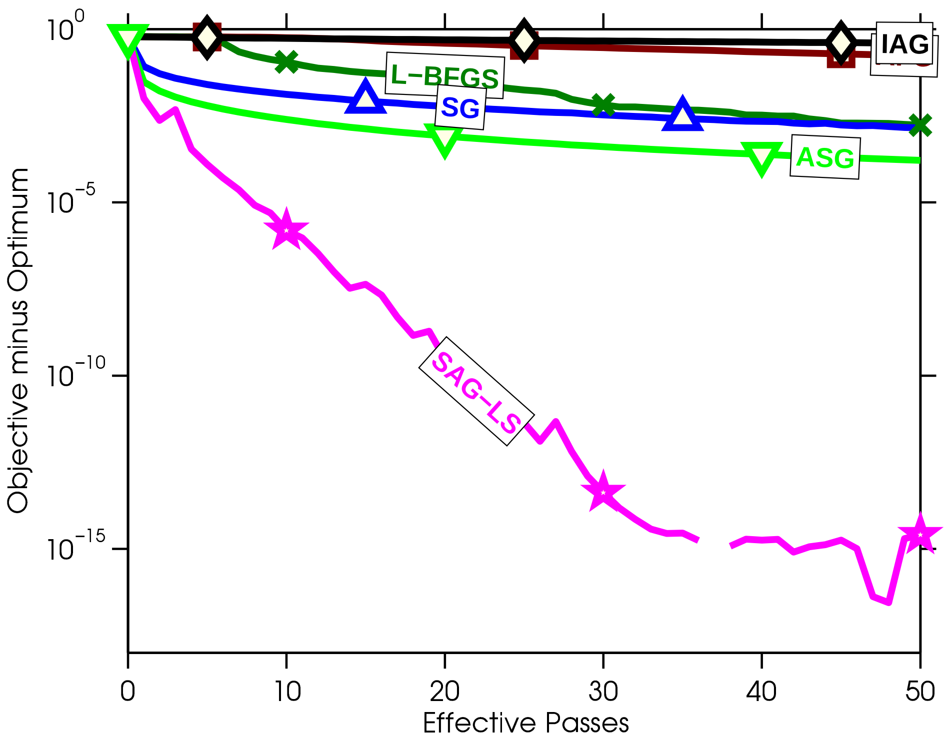


Experimental results (logistic regression)

`quantum` dataset
$(n = 50\ 000,\ d = 78)$

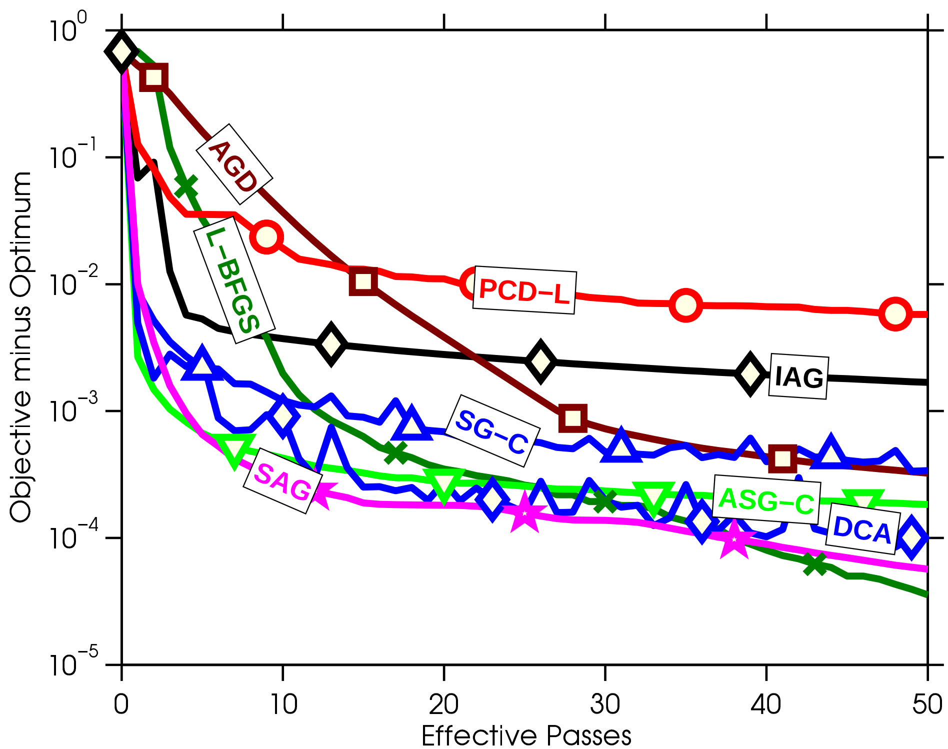
`rcv1` dataset
$(n = 697\ 641,\ d = 47\ 236)$



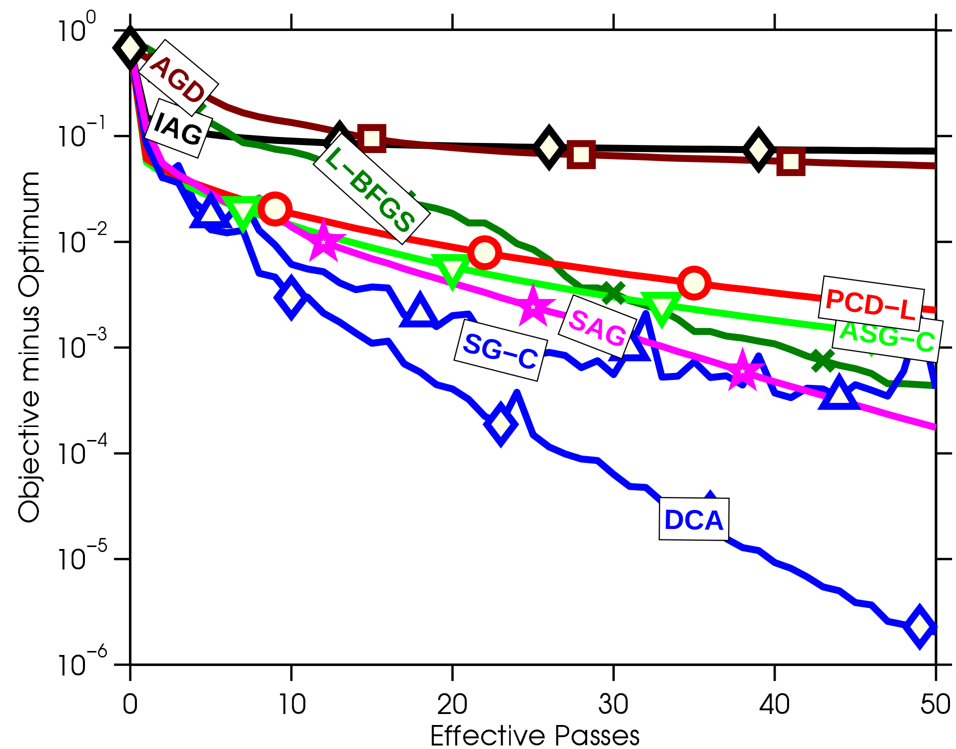


Before non-uniform sampling

protein dataset
$(n = 145\ 751,\ d = 74)$

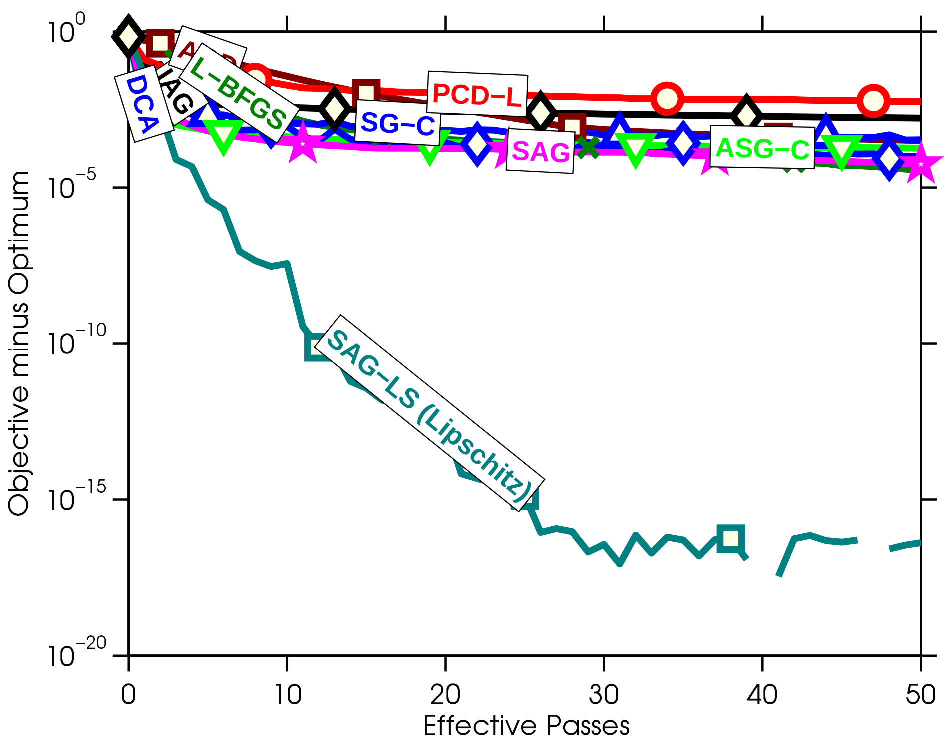


sido dataset
$(n = 12\ 678,\ d = 4\ 932)$

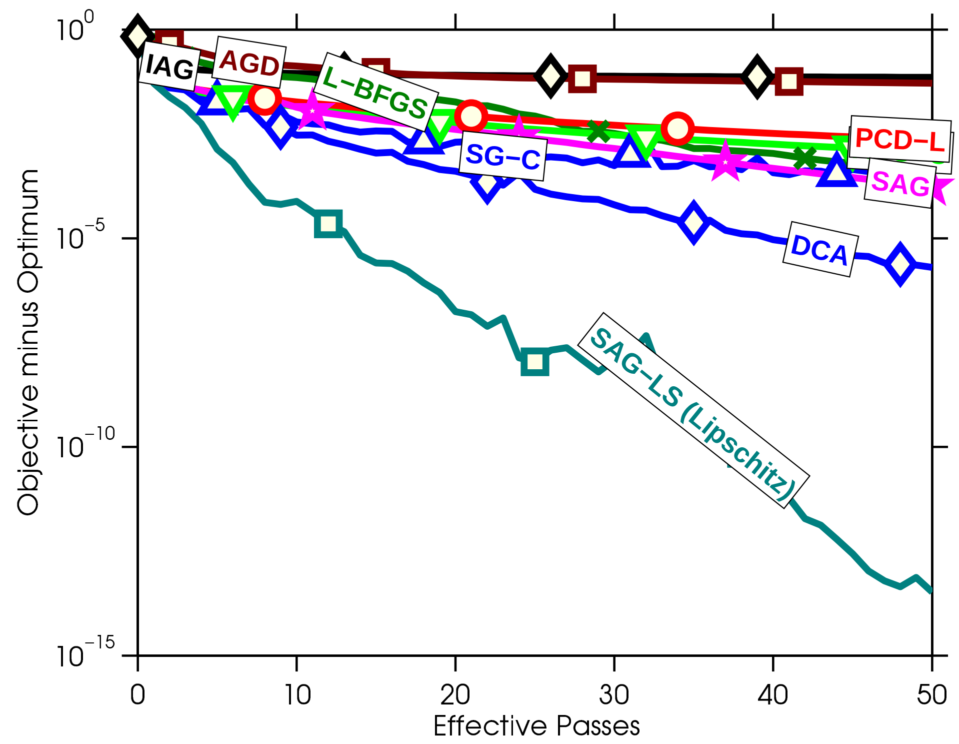


After non-uniform sampling

`protein` dataset
$(n = 145\ 751,\ d = 74)$



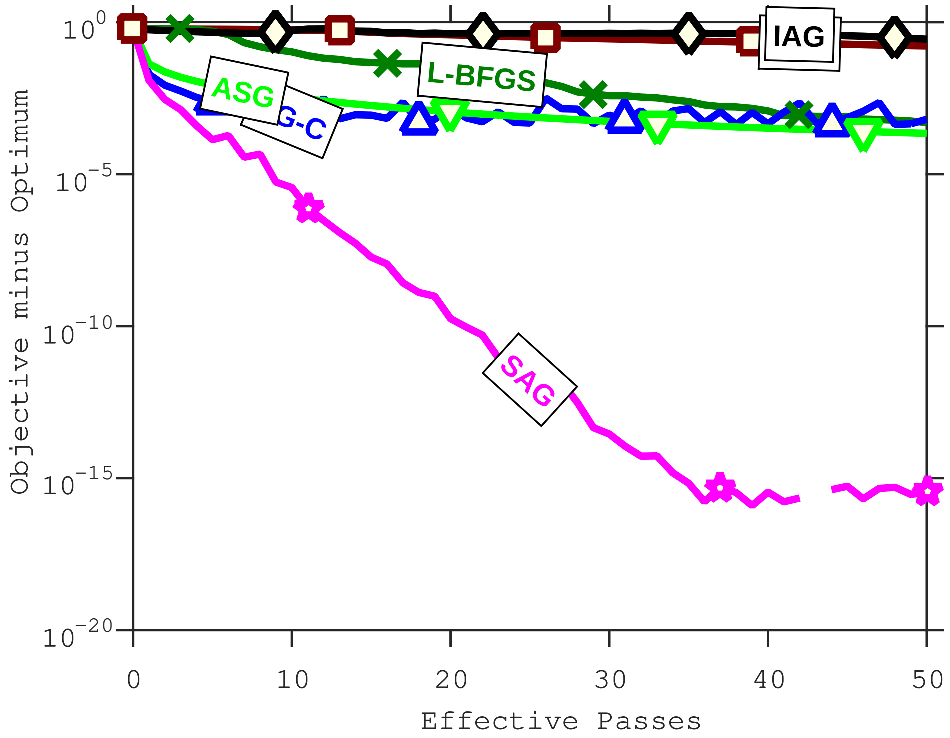
`sido` dataset
$(n = 12\ 678,\ d = 4\ 932)$



From training to testing errors

- `rcv1` dataset ($n = 697\,641$, $d = 47\,236$)
  - NB: IAG, SG-C, ASG with optimal step-sizes in hindsight

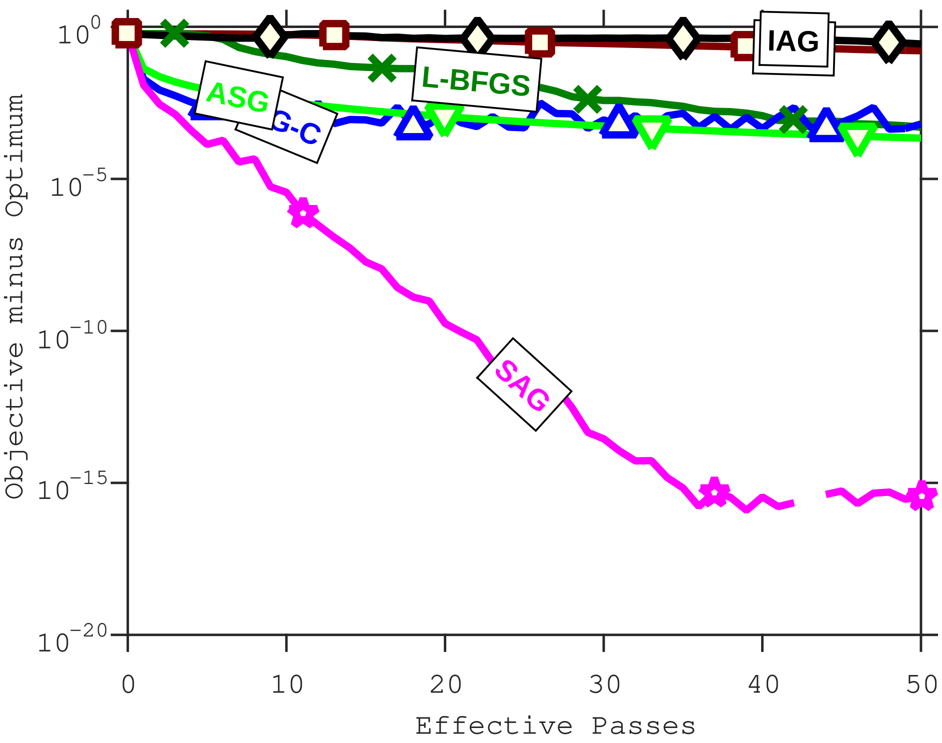
Training cost



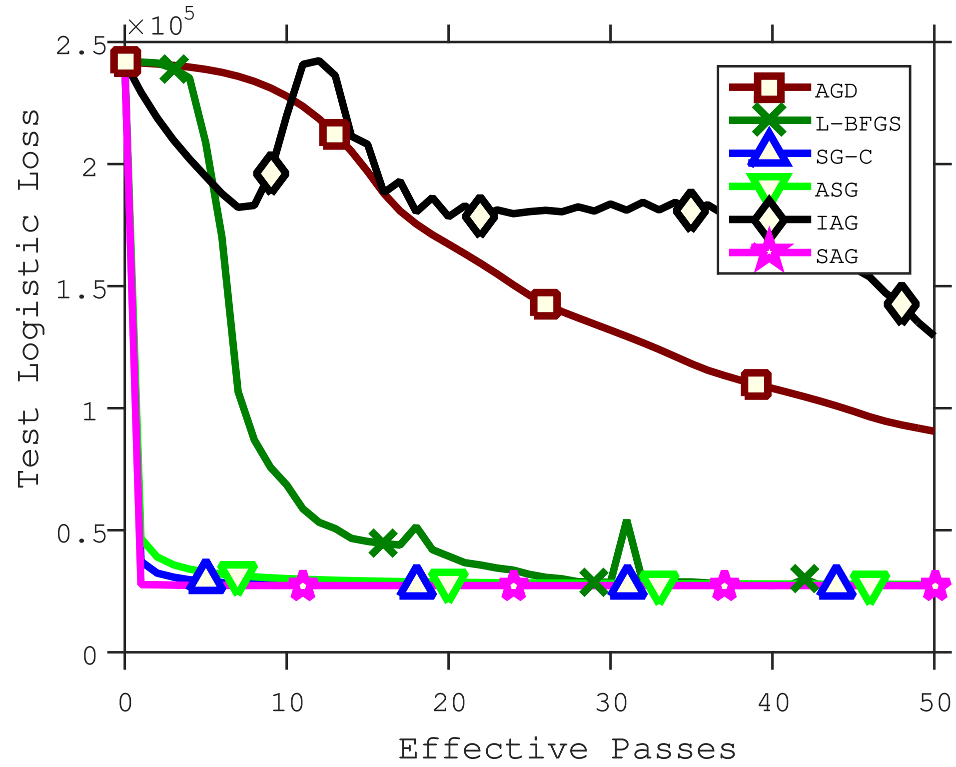
From training to testing errors

- `rcv1` dataset ($n = 697\,641$, $d = 47\,236$)
  - NB: IAG, SG-C, ASG with optimal step-sizes in hindsight

Training cost



Testing cost



Outline

1. **Introduction/motivation: Supervised machine learning**
   - Optimization of finite sums
   - Existing optimization methods for finite sums

2. **Stochastic average gradient (SAG)**
   - Linearly-convergent stochastic gradient method
   - Precise convergence rates

3. **Extensions**
   - Link with variance reduction
   - Acceleration
   - Saddle-point problems

Linearly convergent stochastic gradient algorithms

- **Many related algorithms**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SDCA (Shalev-Shwartz and Zhang, 2013)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - MISO (Mairal, 2015)
  - Finito (Defazio et al., 2014b)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014a)
  - ...

- **Similar rates of convergence and iterations**

Linearly convergent stochastic gradient algorithms

- **Many related algorithms**
  - SAG (Le Roux, Schmidt, and Bach, 2012)
  - SDCA (Shalev-Shwartz and Zhang, 2013)
  - SVRG (Johnson and Zhang, 2013; Zhang et al., 2013)
  - MISO (Mairal, 2015)
  - Finito (Defazio et al., 2014b)
  - SAGA (Defazio, Bach, and Lacoste-Julien, 2014a)
  - ...

- **Similar rates of convergence and iterations**

- **Different interpretations and proofs / proof lengths**
  - Lazy gradient evaluations
  - Variance reduction

Variance reduction

- **Principle**: reducing variance of sample of X by using a sample from another random variable Y with known expectation

$$Z_\alpha = \alpha(X - Y) + \mathbb{E}Y$$

  - $\mathbb{E}Z_\alpha = \alpha\mathbb{E}X + (1-\alpha)\mathbb{E}Y$
  - $\text{var}(Z_\alpha) = \alpha^2\big[\,\text{var}(X) + \text{var}(Y) - 2\text{cov}(X,Y)\big]$
  - $\alpha = 1$: no bias, $\alpha < 1$: potential bias (but reduced variance)
  - Useful if Y positively correlated with X

Variance reduction

- **Principle**: reducing variance of sample of X by using a sample from another random variable Y with known expectation

$$Z_\alpha = \alpha(X - Y) + \mathbb{E}Y$$

  - $\mathbb{E}Z_\alpha = \alpha\mathbb{E}X + (1-\alpha)\mathbb{E}Y$
  - $\text{var}(Z_\alpha) = \alpha^2\big[\,\text{var}(X) + \text{var}(Y) - 2\text{cov}(X,Y)\big]$
  - $\alpha = 1$: no bias, $\alpha < 1$: potential bias (but reduced variance)
  - Useful if Y positively correlated with X

- **Application to gradient estimation** (Johnson and Zhang, 2013; Zhang, Mahdavi, and Jin, 2013)

  - SVRG: $X = f'_{i(t)}(\theta_{t-1})$, $Y = f'_{i(t)}(\tilde{\theta})$, $\alpha = 1$, with $\tilde{\theta}$ stored
  - $\mathbb{E}Y = \frac{1}{n}\sum_{i=1}^n f'_i(\tilde{\theta})$ full gradient at $\tilde{\theta}$, $X - Y = f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\tilde{\theta})$

Stochastic variance reduced gradient (SVRG) (Johnson and Zhang, 2013; Zhang et al., 2013)

- Initialize $\tilde{\theta} \in \mathbb{R}^d$
- For $i_{\text{epoch}} = 1$ to $\#$ of epochs
  - Compute all gradients $f_i'(\tilde{\theta})$; store $g'(\tilde{\theta}) = \frac{1}{n}\sum_{i=1}^{n} f_i'(\tilde{\theta})$
  - Initialize $\theta_0 = \tilde{\theta}$
  - For $t = 1$ to length of epochs

$$\theta_t = \theta_{t-1} - \gamma \Big[g'(\tilde{\theta}) + \big(f_{i(t)}'(\theta_{t-1}) - f_{i(t)}'(\tilde{\theta}) \big) \Big]$$

  - Update $\tilde{\theta} = \theta_t$
- Output: $\tilde{\theta}$

Stochastic variance reduced gradient (SVRG) (Johnson and Zhang, 2013; Zhang et al., 2013)

- Initialize $\tilde{\theta} \in \mathbb{R}^d$
- For $i_{\rm epoch} = 1$ to $\#$ of epochs
  - Compute all gradients $f'_i(\tilde{\theta})$; store $g'(\tilde{\theta}) = \frac{1}{n}\sum_{i=1}^n f'_i(\tilde{\theta})$
  - Initialize $\theta_0 = \tilde{\theta}$
  - For $t = 1$ to length of epochs

$$\theta_t = \theta_{t-1} - \gamma\Big[g'(\tilde{\theta}) + \big(f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\tilde{\theta})\big)\Big]$$

  - Update $\tilde{\theta} = \theta_t$
- Output: $\tilde{\theta}$

– No need to store gradients - two gradient evaluations per inner step

– Two parameters: length of epochs + step-size γ

– Same linear convergence rate as SAG, simpler proof

Interpretation of SAG as variance reduction

- **SAG update**: $\theta_t = \theta_{t-1} - \frac{\gamma}{n}\sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$
  - Interpretation as lazy gradient evaluations

Interpretation of SAG as variance reduction

- **SAG update**: $\theta_t = \theta_{t-1} - \frac{\gamma}{n} \sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$
  - Interpretation as lazy gradient evaluations
- **SAG update**: $\theta_t = \theta_{t-1} - \gamma \Big[\frac{1}{n} \sum_{i=1}^{n} y_i^{t-1} + \frac{1}{n} \big(f'_{i(t)}(\theta_{t-1}) - y_{i(t)}^{t-1} \big) \Big]$
  - Biased update (expectation w.r.t. to $i(t)$ not equal to full gradient)

Interpretation of SAG as variance reduction

- **SAG update**: $\theta_t = \theta_{t-1} - \frac{\gamma}{n}\sum_{i=1}^{n} y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$
  - Interpretation as lazy gradient evaluations
- **SAG update**: $\theta_t = \theta_{t-1} - \gamma\Big[\frac{1}{n}\sum_{i=1}^{n} y_i^{t-1} + \frac{1}{n}\big(f'_{i(t)}(\theta_{t-1}) - y_{i(t)}^{t-1}\big)\Big]$
  - Biased update (expectation w.r.t. to $i(t)$ not equal to full gradient)
- **SVRG update**: $\theta_t = \theta_{t-1} - \gamma\Big[\frac{1}{n}\sum_{i=1}^{n} f_i'(\tilde{\theta}) + \big(f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\tilde{\theta})\big)\Big]$
  - Unbiased update

Interpretation of SAG as variance reduction

- **SAG update**: $\theta_t = \theta_{t-1} - \frac{\gamma}{n}\sum_{i=1}^n y_i^t$ with $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$
  - Interpretation as lazy gradient evaluations

- **SAG update**: $\theta_t = \theta_{t-1} - \gamma\Big[\frac{1}{n}\sum_{i=1}^n y_i^{t-1} + \frac{1}{n}\big(f_{i(t)}'(\theta_{t-1}) - y_{i(t)}^{t-1}\big)\Big]$
  - Biased update (expectation w.r.t. to $i(t)$ not equal to full gradient)

- **SVRG update**: $\theta_t = \theta_{t-1} - \gamma\Big[\frac{1}{n}\sum_{i=1}^n f_i'(\tilde\theta) + \big(f_{i(t)}'(\theta_{t-1}) - f_{i(t)}'(\tilde\theta)\big)\Big]$
  - Unbiased update

- **SAGA update**: $\theta_t = \theta_{t-1} - \gamma\Big[\frac{1}{n}\sum_{i=1}^n y_i^{t-1} + \big(f_{i(t)}'(\theta_{t-1}) - y_{i(t)}^{t-1}\big)\Big]$
  - Defazio, Bach, and Lacoste-Julien (2014a)
  - Unbiased update without epochs

SVRG vs. SAGA

- **SAGA update**: $\theta_t = \theta_{t-1} - \gamma \Big[\frac{1}{n} \sum_{i=1}^n y_i^{t-1} + \big(f'_{i(t)}(\theta_{t-1}) - y_{i(t)}^{t-1} \big) \Big]$

- **SVRG update**: $\theta_t = \theta_{t-1} - \gamma \Big[\frac{1}{n} \sum_{i=1}^n f'_i(\tilde{\theta}) + \big(f'_{i(t)}(\theta_{t-1}) - f'_{i(t)}(\tilde{\theta}) \big) \Big]$

	SAGA	SVRG
Storage of gradients	**yes**	**no**
Epoch-based	no	yes
Parameters	step-size	step-size & epoch lengths
Gradient evaluations per step	1	at least 2
Adaptivity to strong-convexity	yes	no
Robustness to ill-conditioning	yes	no

– See Babanezhad et al. (2015)

Proximal extensions

- **Composite optimization problems**: $\displaystyle\min_{\theta\in\mathbb{R}^d} \ \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) + h(\theta)$
  - f_i smooth and convex
  - h convex, potentially non-smooth

Proximal extensions

- **Composite optimization problems**: $\min_{\theta \in \mathbb{R}^d} \ \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) + h(\theta)$
  - f_i smooth and convex
  - h convex, potentially non-smooth
  - Constrained optimization: $h(\theta) = 0$ if $\theta \in K$, and $+\infty$ otherwise
  - Sparsity-inducing norms, e.g., $h(\theta) = \|\theta\|_1$

Proximal extensions

- **Composite optimization problems**: $\min_{\theta \in \mathbb{R}^d} \ \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) + h(\theta)$
  - f_i smooth and convex
  - h convex, potentially non-smooth
  - Constrained optimization: $h(\theta) = 0$ if $\theta \in K$, and $+\infty$ otherwise
  - Sparsity-inducing norms, e.g., $h(\theta) = \|\theta\|_1$

- **Proximal methods (a.k.a. splitting methods)**
  - Extra projection / soft thresholding step after gradient update
  - See, e.g., Combettes and Pesquet (2011); Bach, Jenatton, Mairal, and Obozinski (2012); Parikh and Boyd (2014)

Proximal extensions

- **Composite optimization problems**: $\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) + h(\theta)$
  - f_i smooth and convex
  - h convex, potentially non-smooth
  - Constrained optimization: $h(\theta) = 0$ if $\theta \in K$, and $+\infty$ otherwise
  - Sparsity-inducing norms, e.g., $h(\theta) = \|\theta\|_1$

- **Proximal methods (a.k.a. splitting methods)**
  - Extra projection / soft thresholding step after gradient update
  - See, e.g., Combettes and Pesquet (2011); Bach, Jenatton, Mairal, and Obozinski (2012); Parikh and Boyd (2014)

- **Directly extends to variance-reduced gradient techniques**
  - Same rates of convergence

Acceleration

- **Similar guarantees for finite sums**

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

Acceleration

- **Similar guarantees for finite sums**

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

- **Acceleration for special algorithms** (e.g., Shalev-Shwartz and Zhang, 2014; Nitanda, 2014; Lan, 2015; Defazio, 2016)
  - Achieves lower bounds for finite sums (Lan, 2015)

- **Catalyst** (Lin, Mairal, and Harchaoui, 2015)
  - Widely applicable generic acceleration scheme

Saddle-point problems
(Balamurugan and Bach, 2016)

- **Lazy evaluation / variance reduction beyond gradient descent**
  - As soon as an iterative algorithm uses a large finite sum

Saddle-point problems (Balamurugan and Bach, 2016)

- **Goal**: Solve $$\min_{\theta \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}^m} L(\theta, \alpha) = \frac{1}{n} \sum_{i=1}^{n} K_i(\alpha, \theta)$$
  - L convex/concave
  - Example: $\min_{\theta \in \mathbb{R}^d} \max_{\alpha \in \mathbb{R}^m} h(\theta) - f^*(\alpha) + \alpha^\top K\theta = \min_{\theta \in \mathbb{R}^d} h(\theta) + f(K\theta)$

Saddle-point problems (Balamurugan and Bach, 2016)

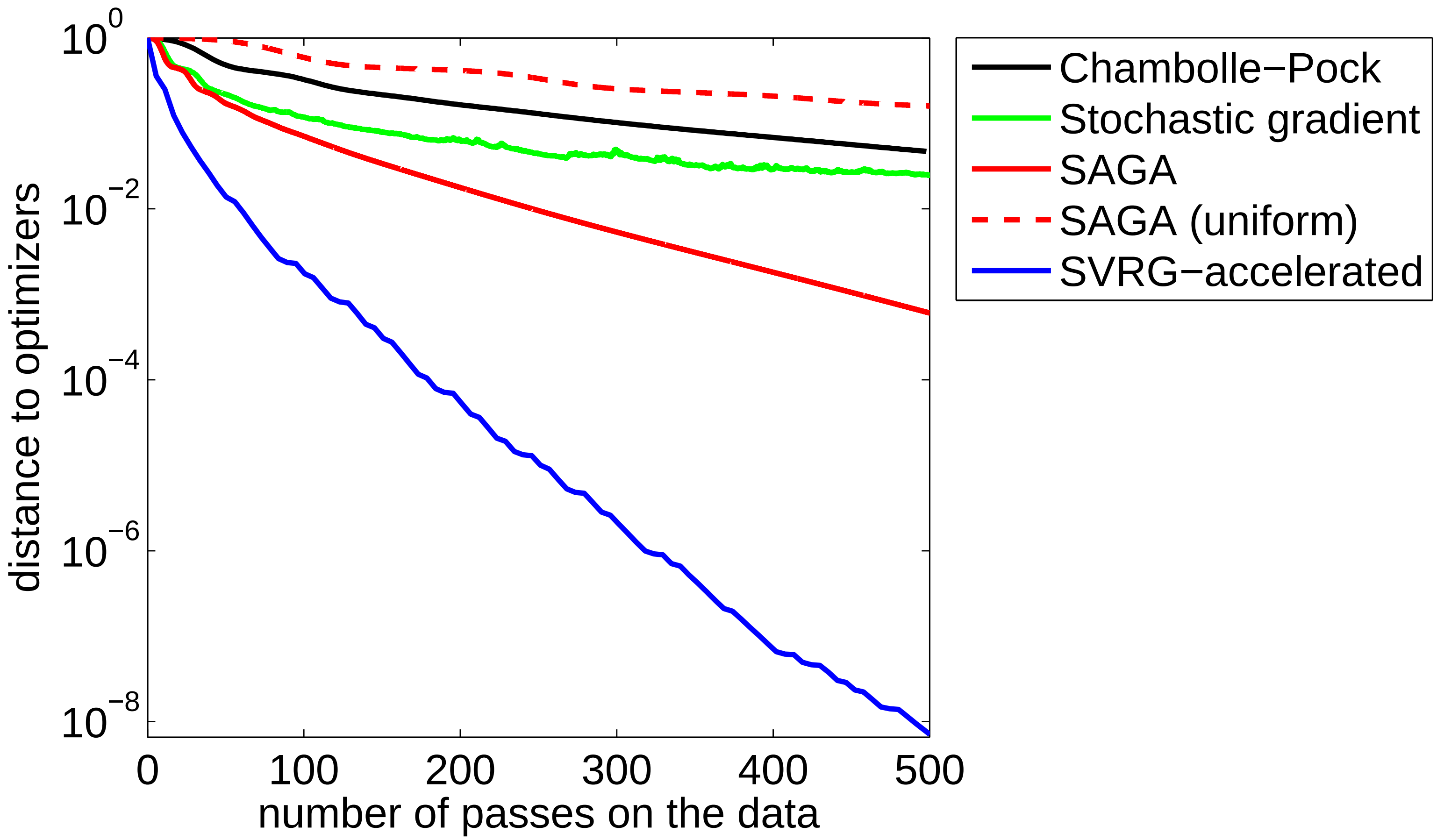
- **Goal**: Solve $\displaystyle \min_{\theta\in\mathbb{R}^d} \max_{\alpha\in\mathbb{R}^m} L(\theta,\alpha) = \frac{1}{n}\sum_{i=1}^n K_i(\alpha,\theta)$
  - L convex/concave
  - Example: $\displaystyle \min_{\theta\in\mathbb{R}^d} \max_{\alpha\in\mathbb{R}^m} h(\theta) - f^*(\alpha) + \alpha^\top K\theta = \min_{\theta\in\mathbb{R}^d} h(\theta) + f(K\theta)$
- **Forward method**: $\left\{ \begin{array}{rcl} \theta_t & = & \theta_{t-1} - \gamma\frac{\partial L}{\partial \theta}(\theta_{t-1},\alpha_{t-1}) \\ \alpha_t & = & \alpha_{t-1} + \gamma\frac{\partial L}{\partial \alpha}(\theta_{t-1},\alpha_{t-1}) \end{array} \right.$

Saddle-point problems
(Balamurugan and Bach, 2016)

- **Goal**: Solve $\displaystyle \min_{\theta\in\mathbb{R}^d} \max_{\alpha\in\mathbb{R}^m} L(\theta,\alpha) = \frac{1}{n}\sum_{i=1}^{n} K_i(\alpha,\theta)$
  - L convex/concave
  - Example: $\displaystyle \min_{\theta\in\mathbb{R}^d} \max_{\alpha\in\mathbb{R}^m} h(\theta) - f^*(\alpha) + \alpha^\top K\theta = \min_{\theta\in\mathbb{R}^d} h(\theta) + f(K\theta)$
- **Forward method**: $\left\{ \begin{array}{rcl} \theta_t & = & \theta_{t-1} - \gamma \frac{\partial L}{\partial \theta}(\theta_{t-1},\alpha_{t-1}) \\ \alpha_t & = & \alpha_{t-1} + \gamma \frac{\partial L}{\partial \alpha}(\theta_{t-1},\alpha_{t-1}) \end{array} \right.$
- **SAG(A) / SVRG can be straightforwardly applied**
  - Convergence proof applies to all monotone operators
  - Strongly convex/concave problems with proximal operators
  - No need for convex/concavity of each K_i
  - Catalyst acceleration is particularly simple

Saddle-point problems
(Balamurugan and Bach, 2016)

- `sido` dataset ($n = 12\ 678$, $d = 4\ 932$)
  - Convex surrogate to area under the ROC curve (AUC)



Conclusions

- **Linearly-convergent stochastic gradient methods**
  - Provable and precise rates
  - Improves on two known lower-bounds (by using structure)
  - Several extensions / interpretations / accelerations

Conclusions

- **Linearly-convergent stochastic gradient methods**
  - Provable and precise rates
  - Improves on two known lower-bounds (by using structure)
  - Several extensions / interpretations / accelerations

- **Extensions and future work**
  - Sampling without replacement (Gurbuzbalaban et al., 2015)
  - Parallelization (Leblond et al., 2016)
  - Non-convex problems (Reddi et al., 2016)

Conclusions

- **Linearly-convergent stochastic gradient methods**
  - Provable and precise rates
  - Improves on two known lower-bounds (by using structure)
  - Several extensions / interpretations / accelerations

- **Extensions and future work**
  - Sampling without replacement (Gurbuzbalaban et al., 2015)
  - Parallelization (Leblond et al., 2016)
  - Non-convex problems (Reddi et al., 2016)
  - Other forms of acceleration (Scieur, d'Aspremont, and Bach, 2016)
  - Exponential convergence of testing errors (Pillaud-Vivien, Rudi, and Bach, 2017)
  - Bounds on stochastic gradient with multiple passes (Lin and Rosasco, 2017; Pillaud-Vivien, Rudi, and Bach, 2018)

References

R. Babanezhad, M. O. Ahmed, A. Virani, M. W. Schmidt, J. Konecný, and S. Sallinen. Stopwasting my gradients: Practical SVRG. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

P. Balamurugan and F. Bach. Stochastic variance reduction methods for saddle-point problems. Technical Report 01319293, HAL, 2016.

D. Blatt, A. O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2008.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Adv. NIPS*, 2008.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

A. Defazio. A simple practical accelerated method for finite sums. In *Advances In Neural Information Processing Systems (NIPS)*, 2016.

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014a.

A. Defazio, J. Domke, and T. S. Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *Proc. ICML*, 2014b.

M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo. On the convergence rate of incremental aggregated gradient algorithms. Technical Report 1506.02081, arXiv, 2015.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, 2001.

G. Lan. An optimal randomized incremental gradient method. Technical Report 1507.02000, arXiv, 2015.

N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

R. Leblond, F. Pedregosa, and S. Lacoste-Julien. Asaga: Asynchronous parallel Saga. Technical Report 1606.04809, arXiv, 2016.

H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Junhong Lin and Lorenzo Rosasco. Optimal rates for multi-pass stochastic gradient methods. *Journal of Machine Learning Research*, 18(97):1–47, 2017. URL `http://jmlr.org/papers/v18/17-176.html`.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

A. S. Nemirovski and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley

& Sons, 1983.

Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Kluwer, 2004.

A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Exponential convergence of testing error for stochastic gradient methods. Technical Report 1712.04755, arXiv, 2017.

Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. Technical Report 1805.10074, arXiv, 2018.

S. J. Reddi, A. Hefny, S. Sra, B. Póczós, and A. Smola. Stochastic variance reduction for nonconvex optimization. Technical Report 1603.06160, arXiv, 2016.

H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statistics*, 22:400–407, 1951.

D. Scieur, A. d'Aspremont, and F. Bach. Regularized nonlinear acceleration. In *Advances in Neural Information Processing Systems*, 2016.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *Proc. ICML*, 2014.

B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. *Adv. NIPS*, 2003.

I. Tsochantaridis, Thomas Joachims, T., Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

L. Zhang, M. Mahdavi, and R. Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, 2013.