Stochastic optimization:
Beyond stochastic gradients and convexity
Part I

Francis Bach

INRIA - Ecole Normale Supérieure, Paris, France

Joint tutorial with Suvrit Sra, MIT - *NIPS - 2016*

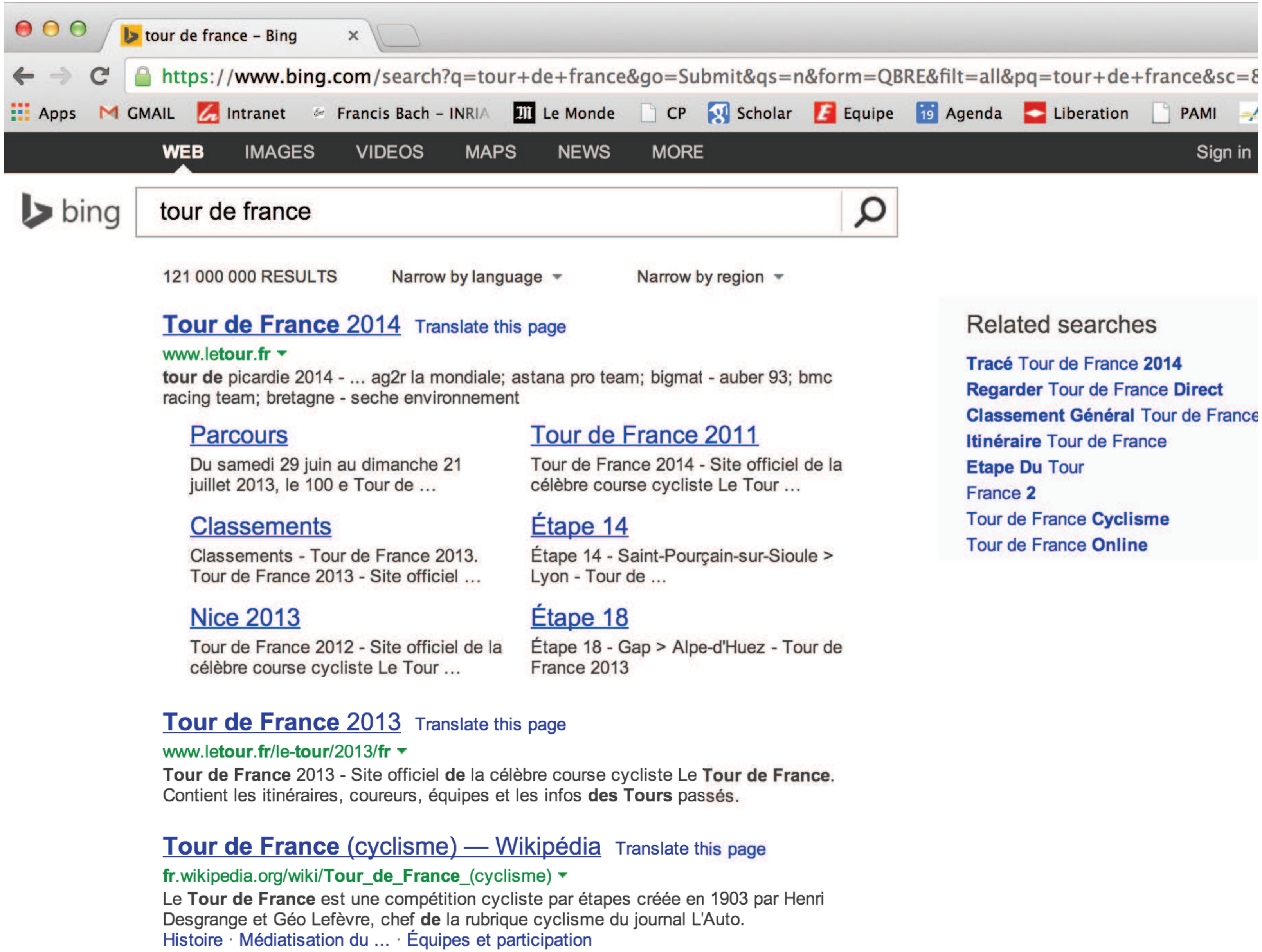
Context

Machine learning for large-scale data

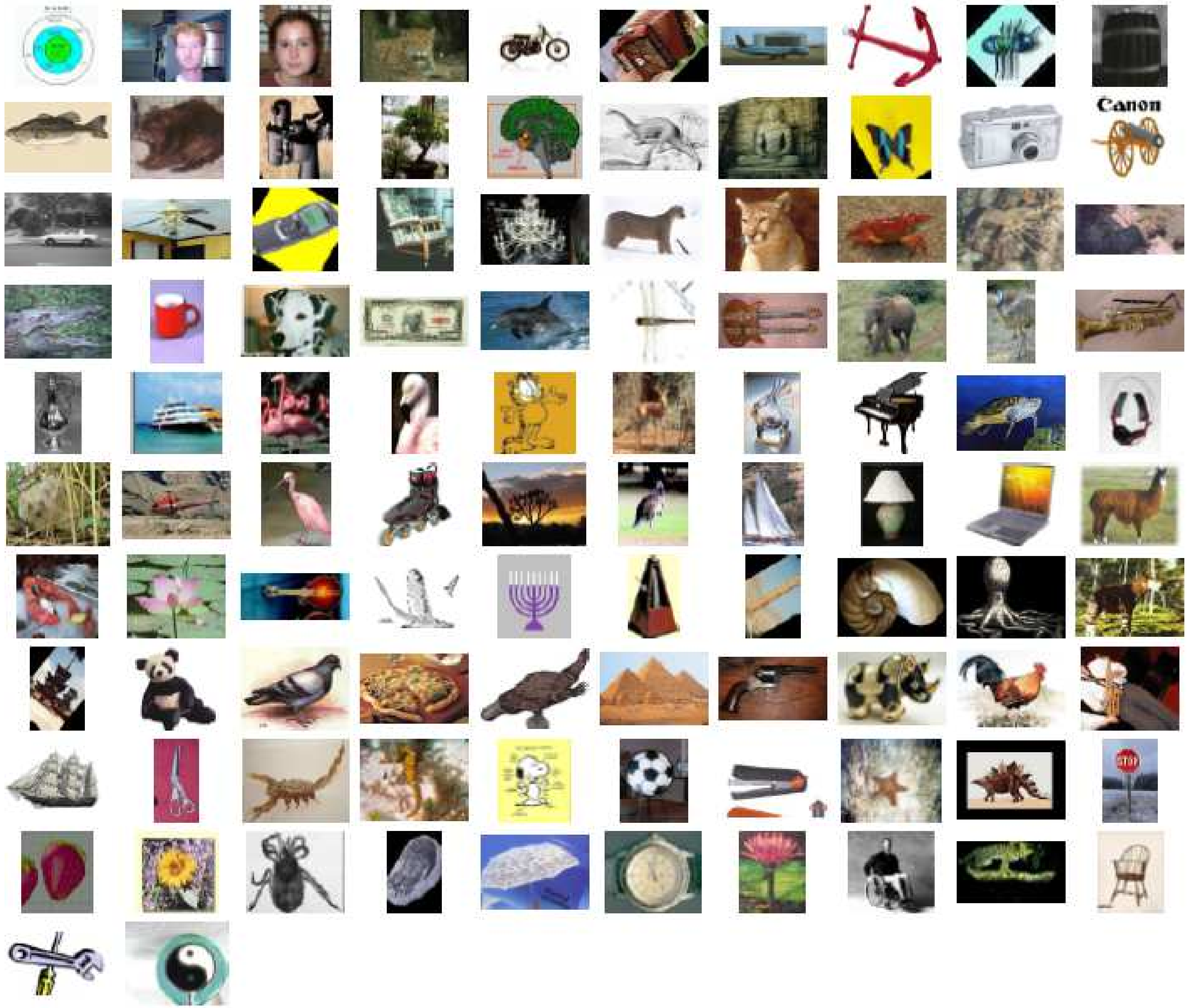
- **Large-scale supervised machine learning**: **large d, large n**
  - d : dimension of each observation (input) or number of parameters
  - n : number of observations

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

Search engines - Advertising - Marketing



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. **Convex finite-sum problems**
   - Linearly-convergent stochastic gradient method
   - SAG, SAGA, SVRG, SDCA, MISO, etc.
   - From lazy gradient evaluations to variance reduction
3. **Non-convex problems**
4. **Parallel and distributed settings**
5. **Perspectives**

References

- **Textbooks and tutorials**
  - Nesterov (2004): *Introductory lectures on convex optimization*
  - Bubeck (2015): *Convex Optimization: Algorithms and Complexity*
  - Bertsekas (2016): *Nonlinear programming*
  - Bottou et al. (2016): *Optimization methods for large-scale machine learning*

- **Research papers**
  - See end of slides
  - Slides available at `www.ens.fr/~fbach/`

Parametric supervised machine learning

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

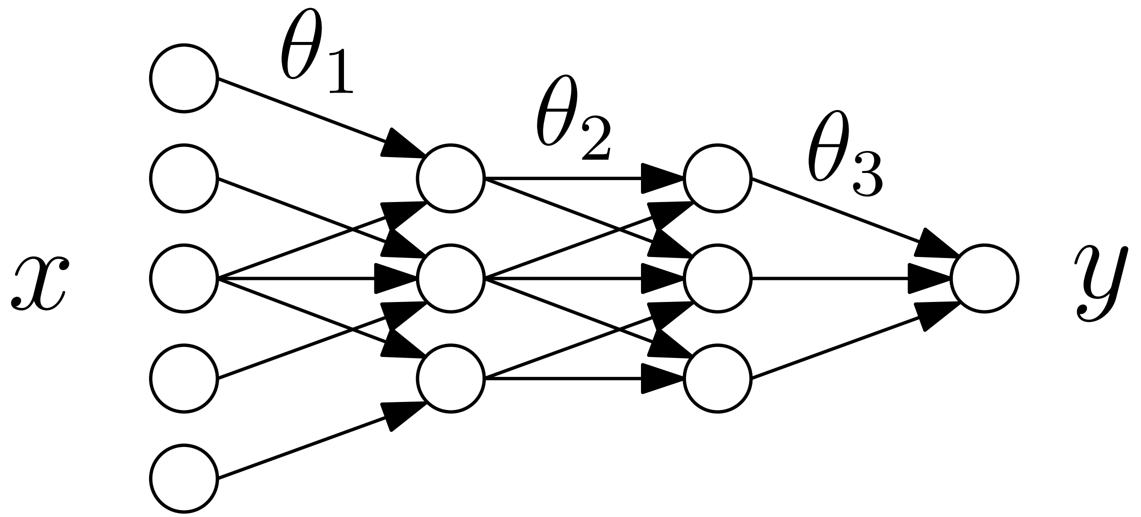
- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

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

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

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

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

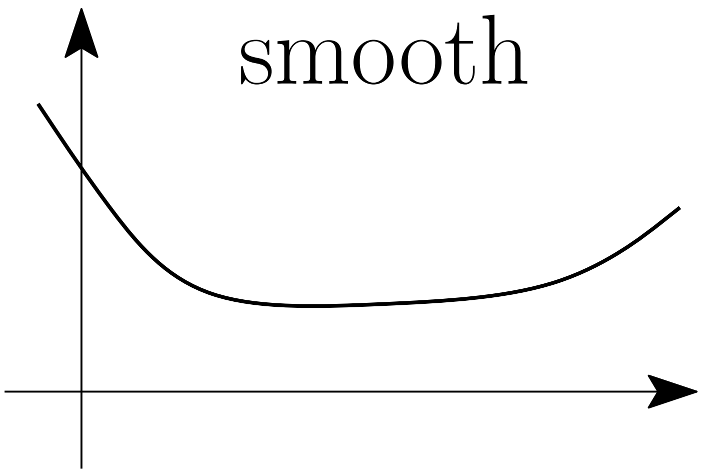
data fitting term + regularizer

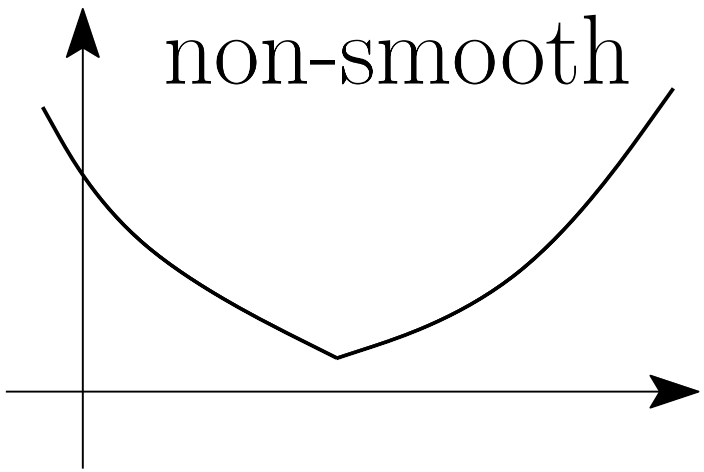
- Optimization: optimization of regularized risk — training cost
- Statistics: guarantees on $\mathbb{E}_{p(x,y)} \ell(y, h(x, \theta))$ — testing cost

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

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





Smoothness and (strong) convexity

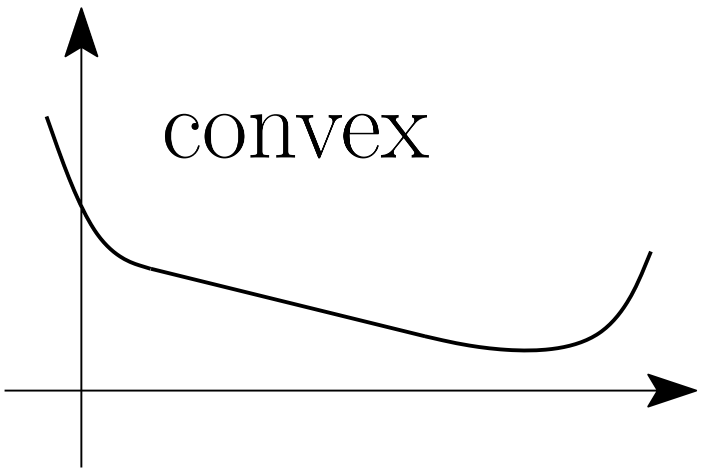
- A function $g : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if and only if it is twice differentiable and

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

- **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

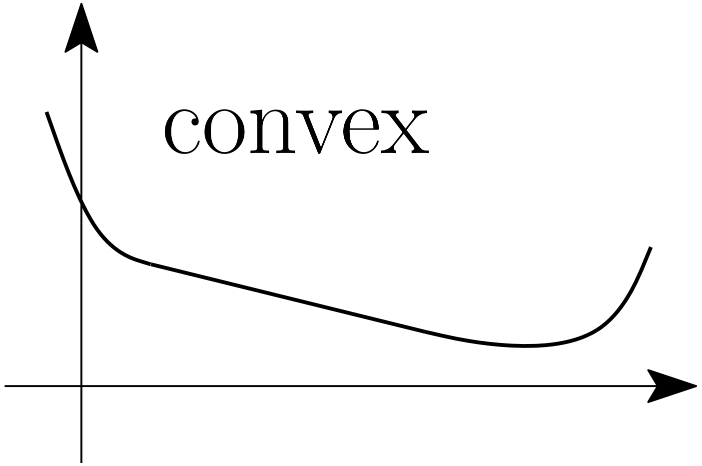
- A twice differentiable function $g : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if
$$\forall \theta \in \mathbb{R}^d, \text{ eigenvalues}\big[g''(\theta)\big] \geqslant 0$$

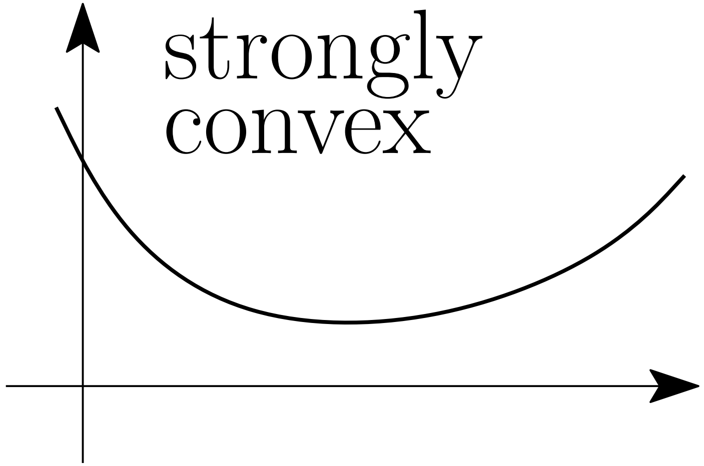


Smoothness and (strong) convexity

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

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



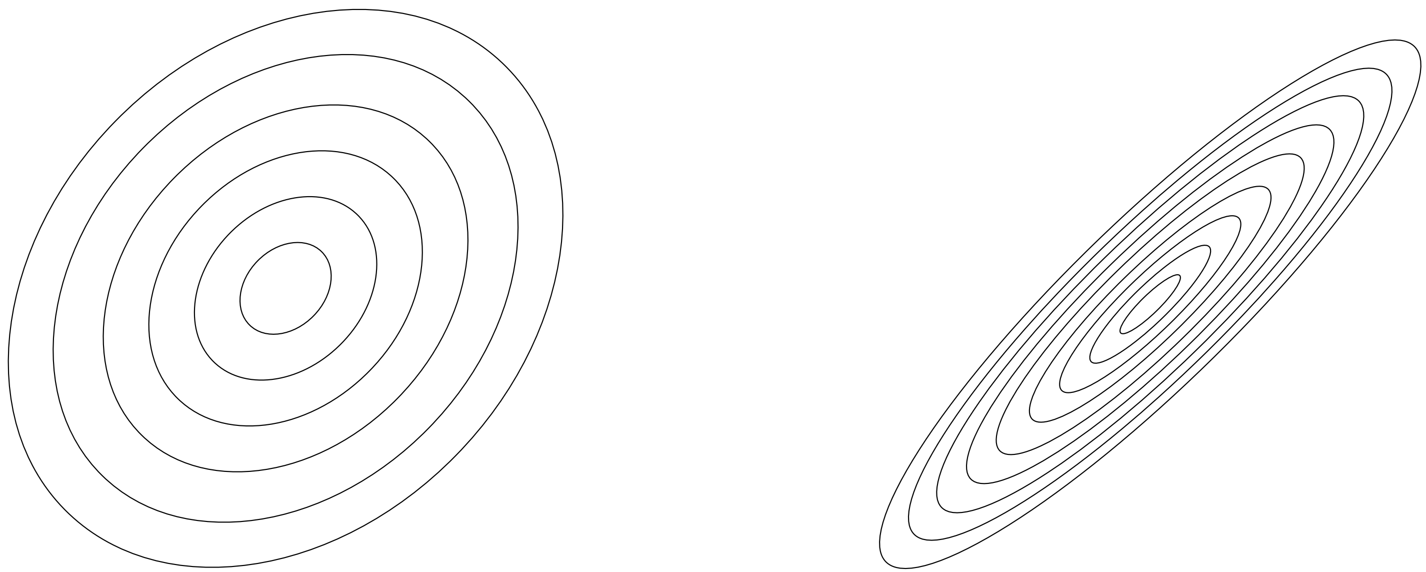


Smoothness and (strong) convexity

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

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

  - 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 $g : \mathbb{R}^d \to \mathbb{R}$ 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 $h(x, \theta) = \theta^\top \Phi(x)$

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

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

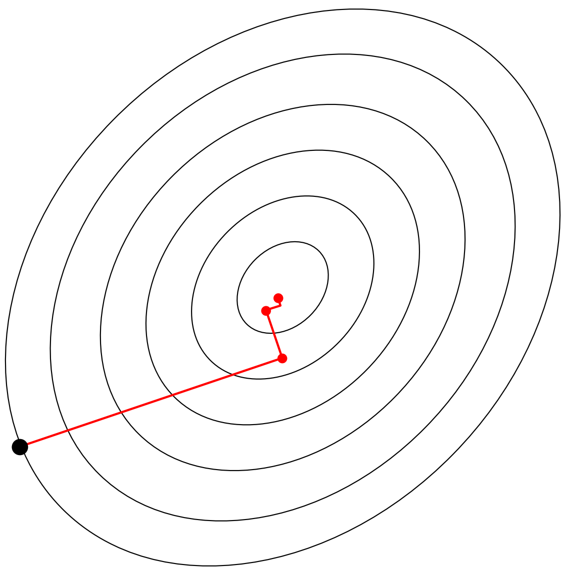
- **Strong convexity in machine learning**
  - With $g(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i, h(x_i, \theta))$
  - Strongly convex loss and linear predictions $h(x, \theta) = \theta^\top \Phi(x)$
  - Invertible covariance matrix $\frac{1}{n}\sum_{i=1}^n \Phi(x_i)\Phi(x_i)^\top \Rightarrow n \geqslant d$
  - 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 $L/\sqrt{n} \geqslant \mu \geqslant L/n$

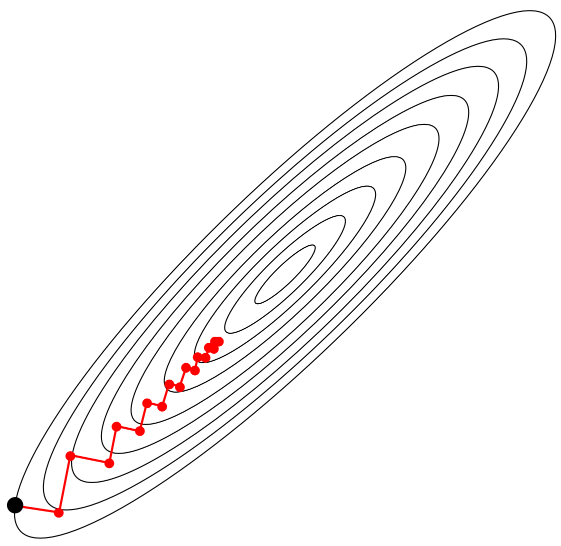
Iterative methods for minimizing smooth functions

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

- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t \, g'(\theta_{t-1})$



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



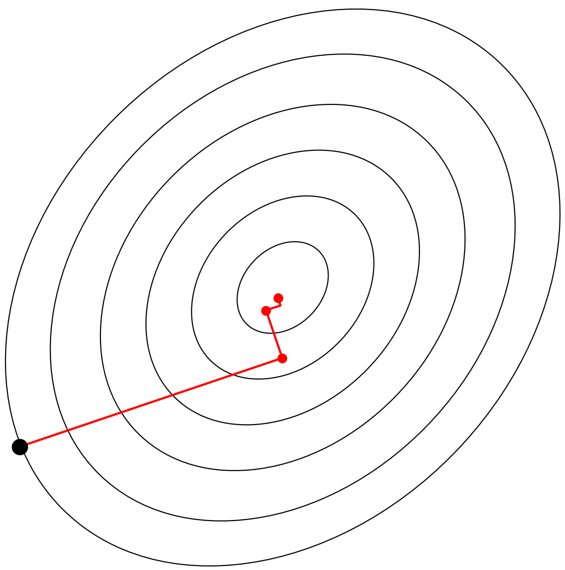
(large $\kappa = L/\mu$)

Iterative methods for minimizing smooth functions

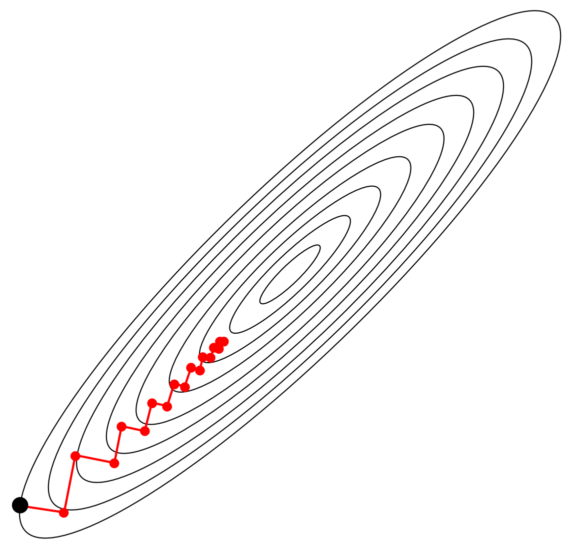
- **Assumption**: g convex and L-smooth on $\mathbb{R}^d$

- **Gradient descent**: $\theta_t = \theta_{t-1} - \gamma_t\, g'(\theta_{t-1})$

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



(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

- **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
  - Adaptivity with self-concordance assumption (Bach, 2014)
  - 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. **Convex finite-sum problems**
   - Linearly-convergent stochastic gradient method
   - SAG, SAGA, SVRG, SDCA, etc.
   - From lazy gradient evaluations to variance reduction

3. **Non-convex problems**

4. **Parallel and distributed settings**

5. **Perspectives**

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

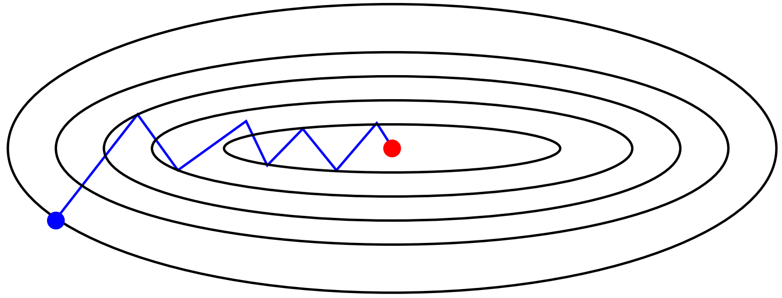
- Minimizing $g(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$ with $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$

- 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 $f_i(\theta) = \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$

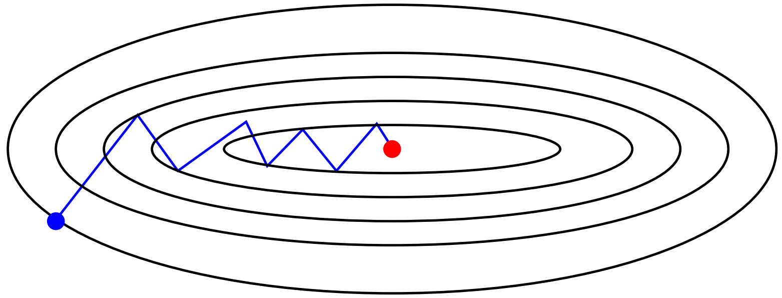
- Batch gradient descent: $\theta_t = \theta_{t-1} - \gamma_t g'(\theta_{t-1}) = \theta_{t-1} - \frac{\gamma_t}{n}\sum_{i=1}^{n} f_i'(\theta_{t-1})$
  - Linear (e.g., exponential) convergence rate in $O(e^{-t/\kappa})$
  - Iteration complexity is linear in n

- 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

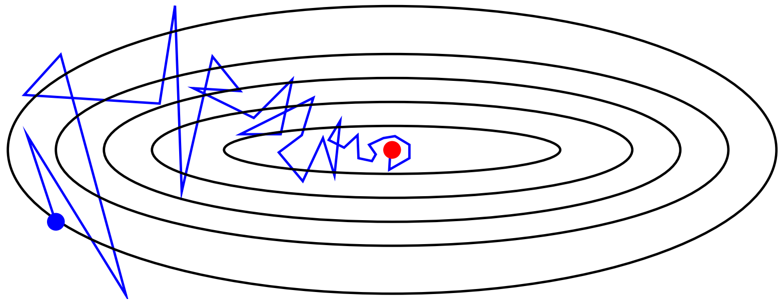
Stochastic vs. deterministic methods

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

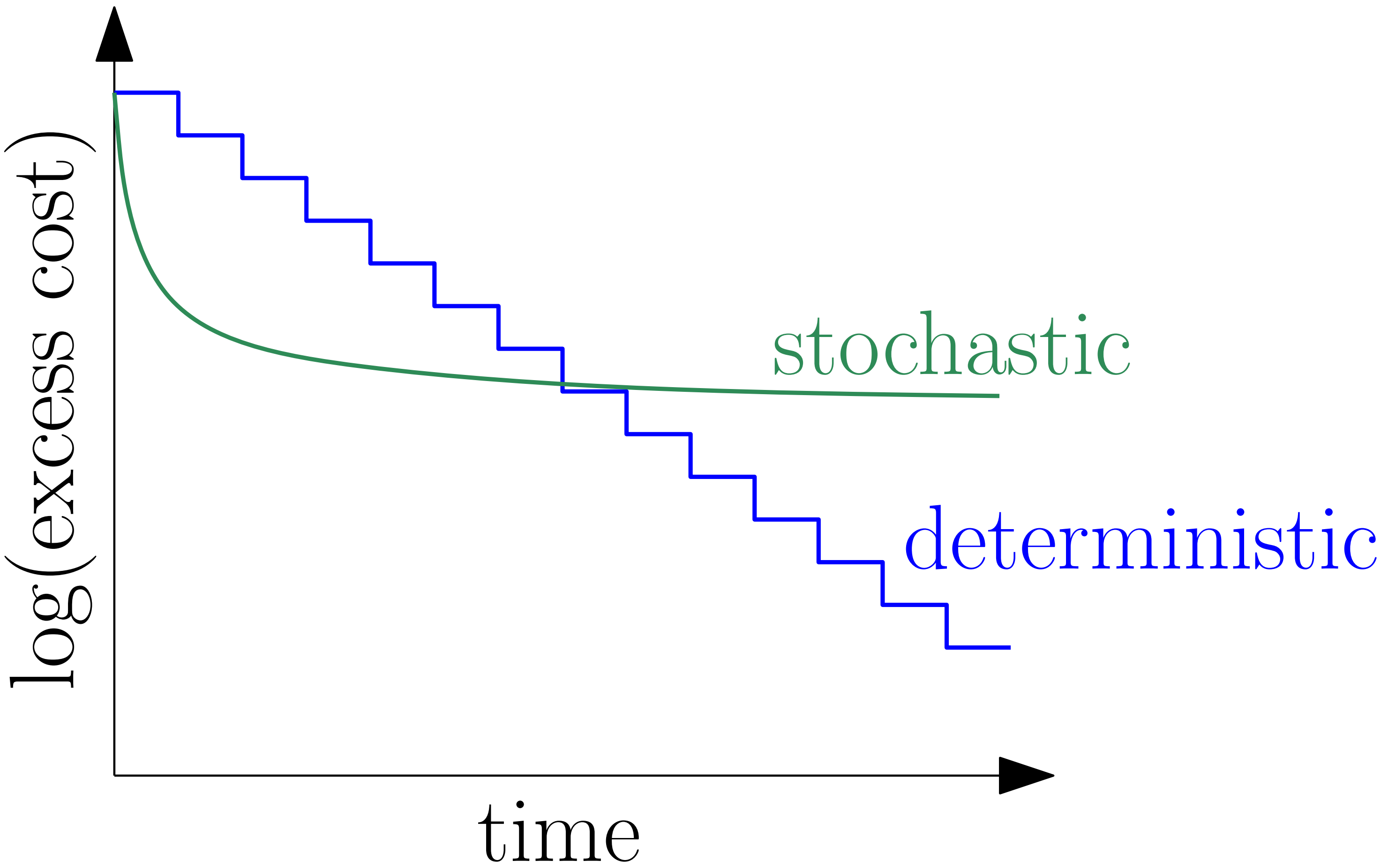


- Stochastic gradient descent: $\theta_t = \theta_{t-1} - \gamma_t f'_{i(t)}(\theta_{t-1})$



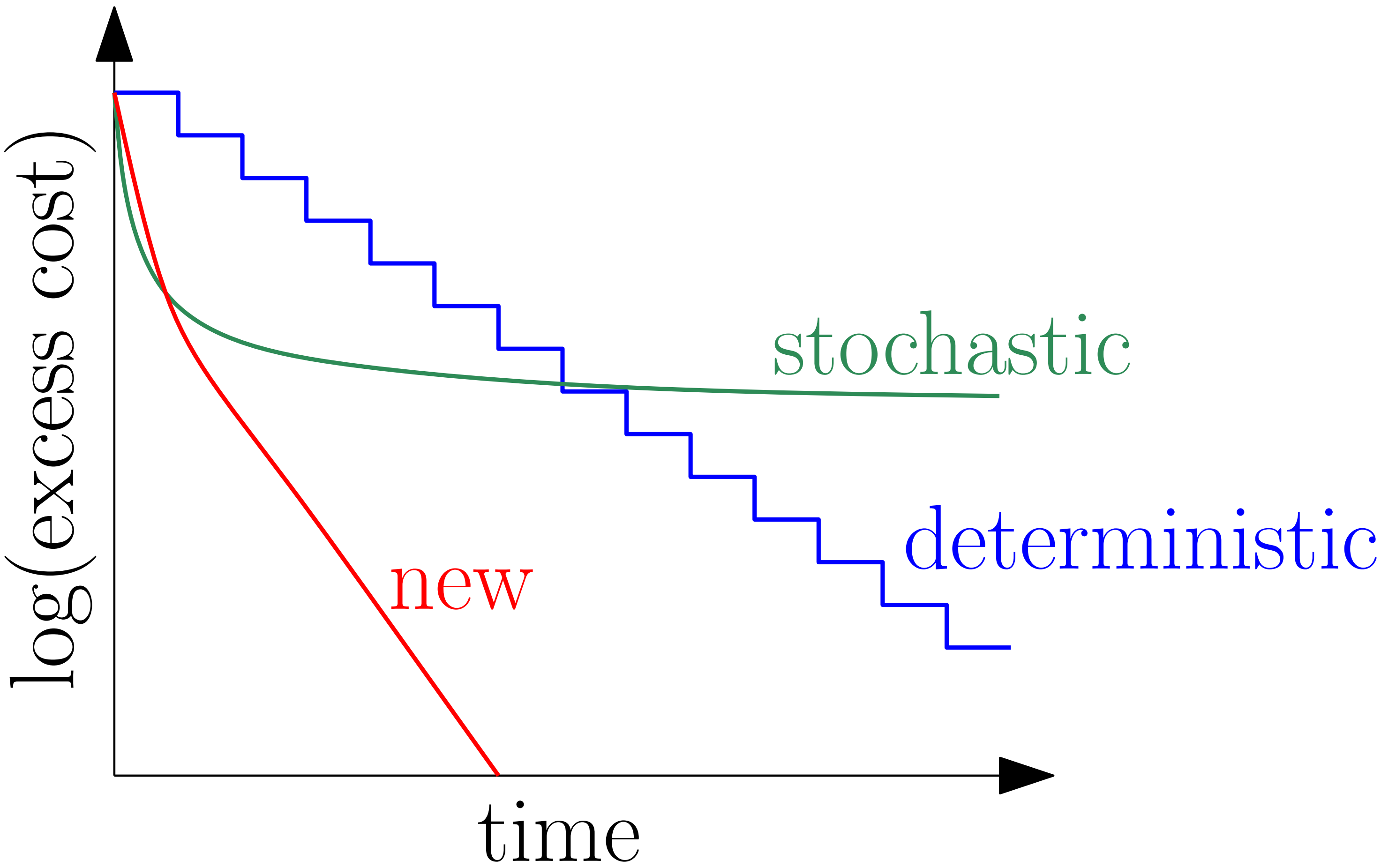
Stochastic vs. deterministic methods

- **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



Accelerating gradient methods - Related work

- **Generic acceleration** (Nesterov, 1983, 2004)

$$\theta_t = \eta_{t-1} - \gamma_t g'(\eta_{t-1}) \text{ and } \eta_t = \theta_t + \delta_t(\theta_t - \theta_{t-1})$$

Accelerating gradient methods - Related work

- **Generic acceleration** (Nesterov, 1983, 2004)

$$\theta_t = \eta_{t-1} - \gamma_t g'(\eta_{t-1}) \text{ and } \eta_t = \theta_t + \delta_t(\theta_t - \theta_{t-1})$$

  – Good choice of momentum term $\delta_t \in [0, 1)$

  $g(\theta_t) - g(\theta_*) \leqslant O(1/t^2)$

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

  – Optimal rates after $t = O(d)$ iterations (Nesterov, 2004)

Accelerating gradient methods - Related work

- **Generic acceleration** (Nesterov, 1983, 2004)

$$\theta_t = \eta_{t-1} - \gamma_t g'(\eta_{t-1}) \text{ and } \eta_t = \theta_t + \delta_t(\theta_t - \theta_{t-1})$$

  - Good choice of momentum term $\delta_t \in [0, 1)$

    $g(\theta_t) - g(\theta_*) \leqslant O(1/t^2)$

    $g(\theta_t) - g(\theta_*) \leqslant O(e^{-t\sqrt{\mu/L}}) = O(e^{-t/\sqrt{\kappa}})$ if μ-strongly convex
  - Optimal rates after $t = O(d)$ iterations (Nesterov, 2004)
  - Still $O(nd)$ iteration cost: complexity $= O(nd \cdot \sqrt{\kappa} \log \frac{1}{\varepsilon})$

Accelerating gradient methods - Related work

- **Constant step-size stochastic gradient**
  - Solodov (1998); Nedic and Bertsekas (2000)
  - Linear convergence, but only up to a fixed tolerance

Accelerating gradient methods - Related work

- **Constant step-size stochastic gradient**
  - Solodov (1998); Nedic and Bertsekas (2000)
  - Linear convergence, but only up to a fixed tolerance
- **Stochastic methods in the dual (SDCA)**
  - Shalev-Shwartz and Zhang (2013)
  - Similar linear rate but limited choice for the f_i's
  - Extensions without duality: see Shalev-Shwartz (2016)

Accelerating gradient methods - Related work

- **Constant step-size stochastic gradient**
  - Solodov (1998); Nedic and Bertsekas (2000)
  - Linear convergence, but only up to a fixed tolerance

- **Stochastic methods in the dual (SDCA)**
  - Shalev-Shwartz and Zhang (2013)
  - Similar linear rate but limited choice for the f_i's
  - Extensions without duality: see Shalev-Shwartz (2016)

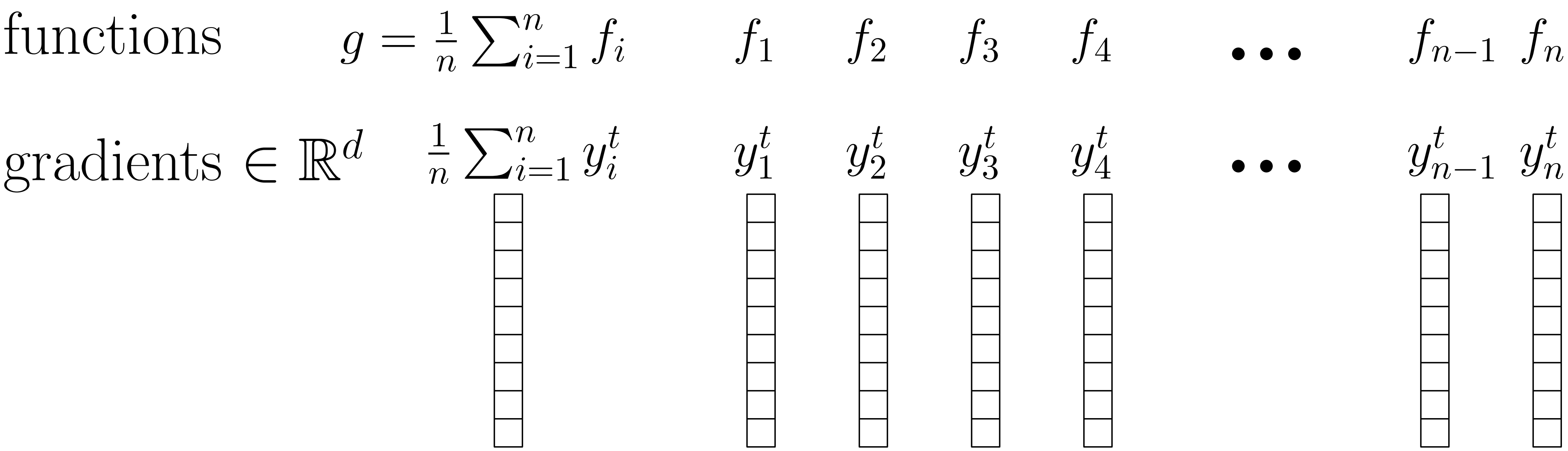
- **Stochastic version of accelerated batch gradient methods**
  - Tseng (1998); Ghadimi and Lan (2010); Xiao (2010)
  - Can improve constants, but still have sublinear $O(1/t)$ rate

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 $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$

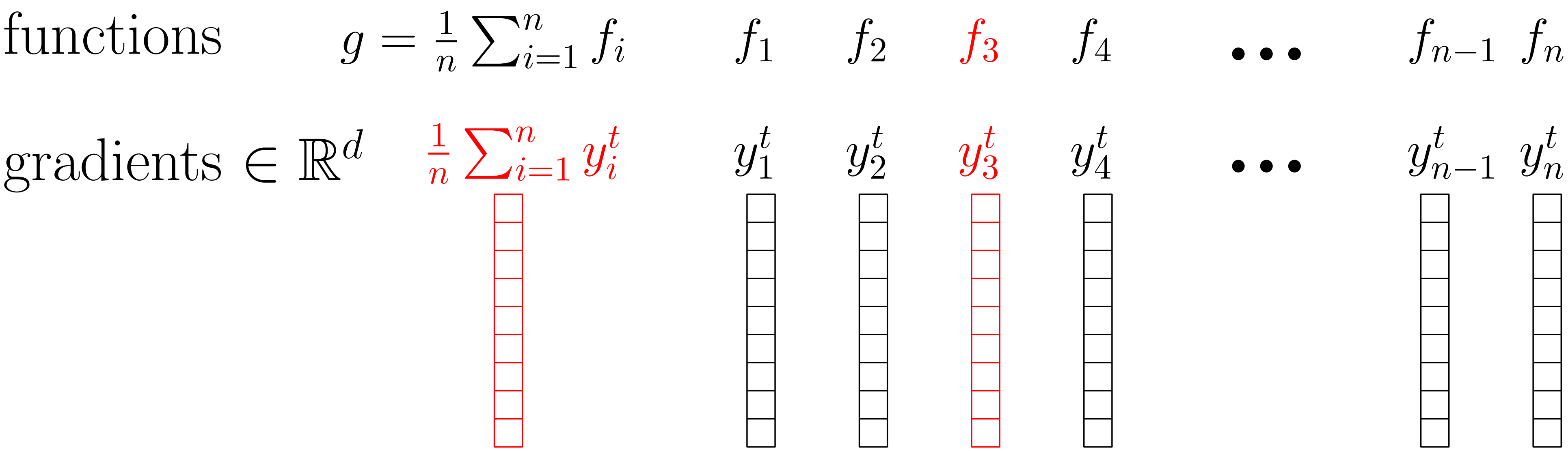
Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **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 $y_i^t = \begin{cases} f_i'(\theta_{t-1}) & \text{if } i = i(t) \\ y_i^{t-1} & \text{otherwise} \end{cases}$



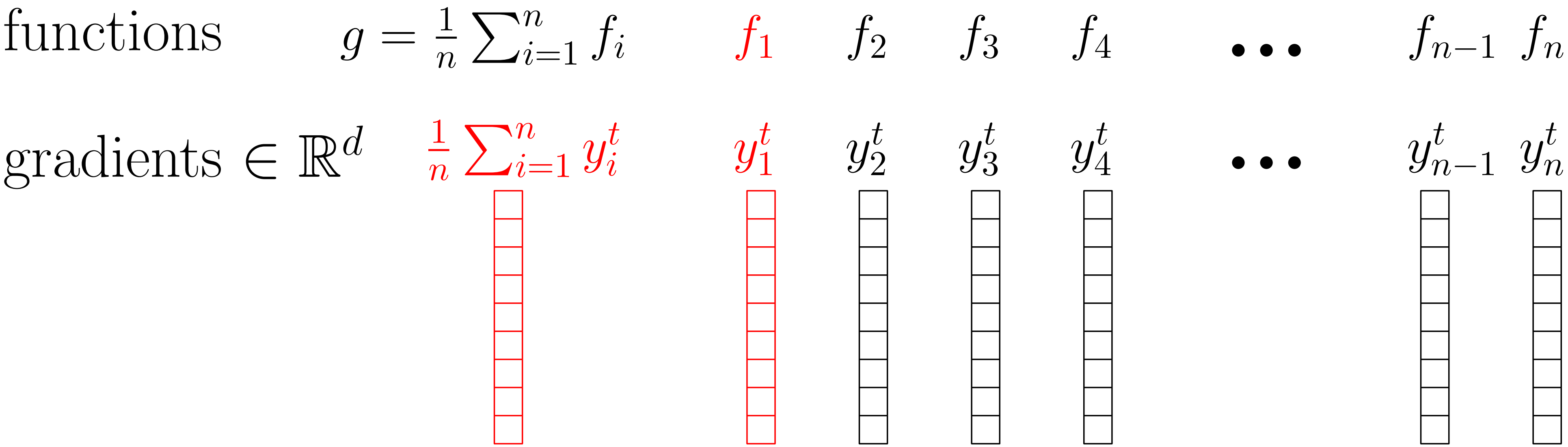
Stochastic average gradient (Le Roux, Schmidt, and Bach, 2012)

- **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 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 $f_i'(\theta) = \ell'(y_i, \Phi(x_i)^\top \theta)\, \Phi(x_i)$

Stochastic average gradient - Convergence analysis

- **Assumptions**
  - Each f_i is L-smooth, $i = 1, \dots, n$
  - $g = \frac{1}{n}\sum_{i=1}^n f_i$ is μ-strongly convex
  - constant step size $\gamma_t = 1/(16L)$ - no need to know μ

Stochastic average gradient - Convergence analysis

- **Assumptions**
  - Each f_i is L-smooth, $i = 1, \dots, n$
  - $g = \frac{1}{n} \sum_{i=1}^n f_i$ is μ-strongly convex
  - constant step size $\gamma_t = 1/(16L)$ - no need to know μ

- **Strongly convex case** (Le Roux et al., 2012; Schmidt et al., 2016)

$$\mathbb{E}\big[g(\theta_t) - g(\theta_*)\big] \leqslant \text{ cst } \times \left(1 - \min\left\{\frac{1}{8n}, \frac{\mu}{16L}\right\}\right)^t$$

  - Linear (exponential) convergence rate with $O(d)$ iteration cost
  - After one pass, reduction of cost by $\exp\left(-\min\left\{\frac{1}{8}, \frac{n\mu}{16L}\right\}\right)$
  - NB: in machine learning, may often restrict to $\mu \geqslant L/n$
    $\Rightarrow$ constant error reduction after each effective pass

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	$d\times$	$\sqrt{n}/\varepsilon$

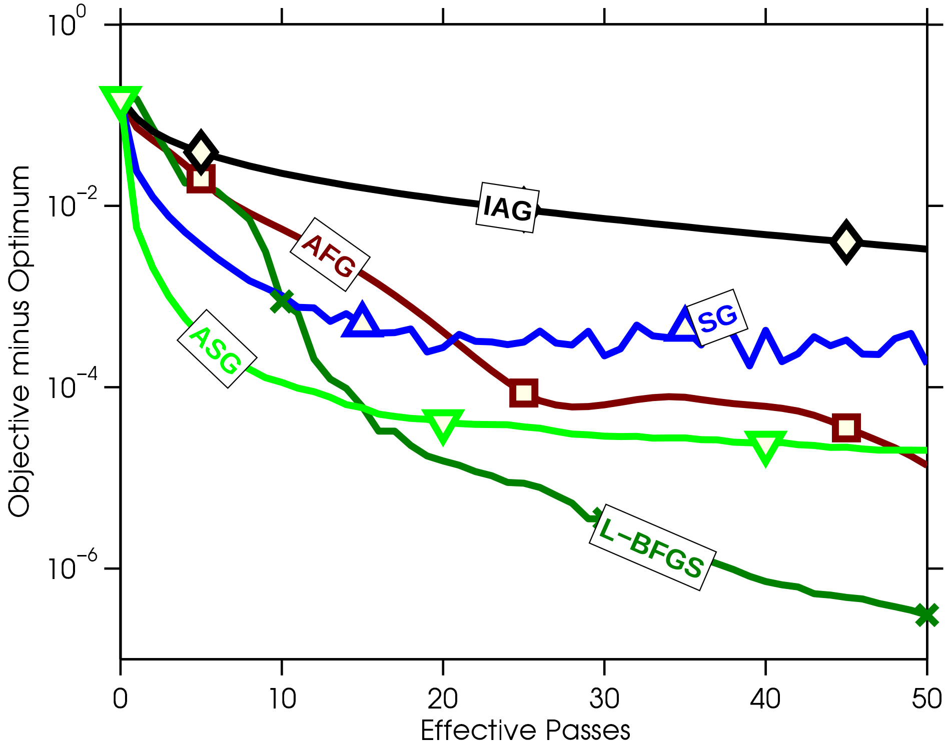
- Adaptivity to potentially hidden strong convexity
- No need to know the local/global strong-convexity constant

Stochastic average gradient
Implementation details and extensions

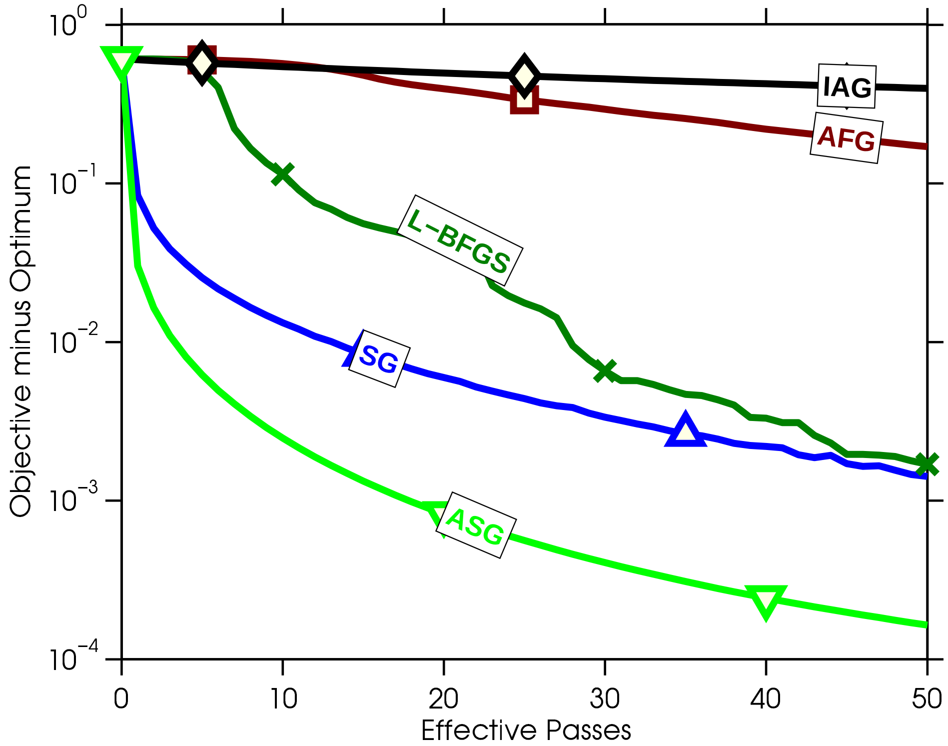
- **Sparsity in the features**
  - Just-in-time updates $\Rightarrow$ replace $O(d)$ by number of non zeros
  - See also Leblond, Pedregosa, and Lacoste-Julien (2016)
- **Mini-batches**
  - Reduces the memory requirement $+$ block access to data
- **Line-search**
  - Avoids knowing L in advance
- **Non-uniform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

Experimental results (logistic regression)

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



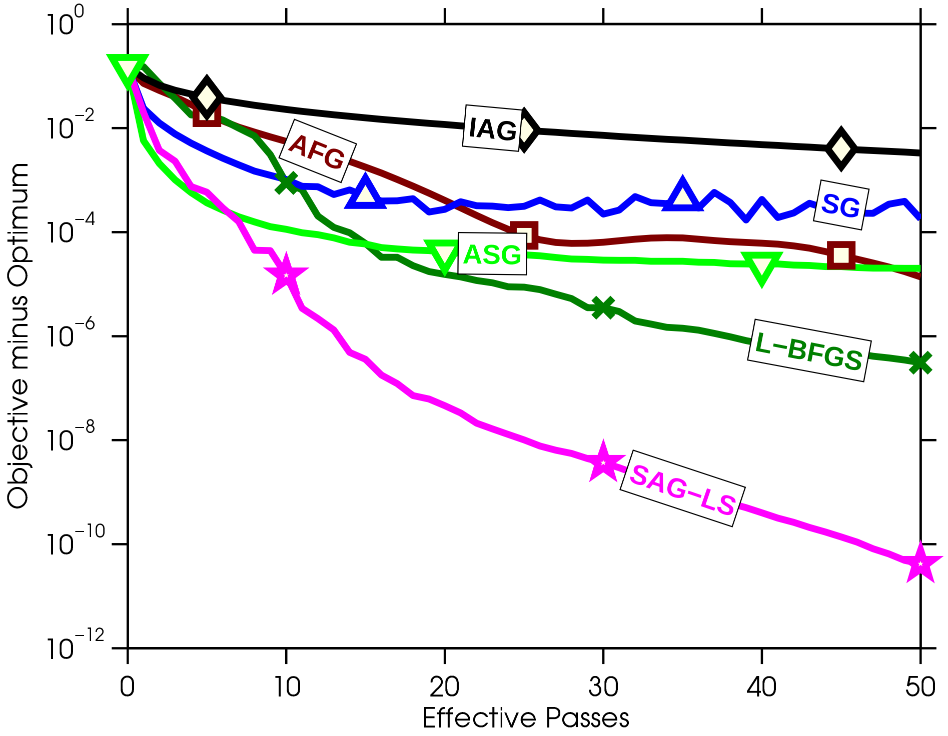
rcv1 dataset
$(n = 697\ 641,\ d = 47\ 236)$

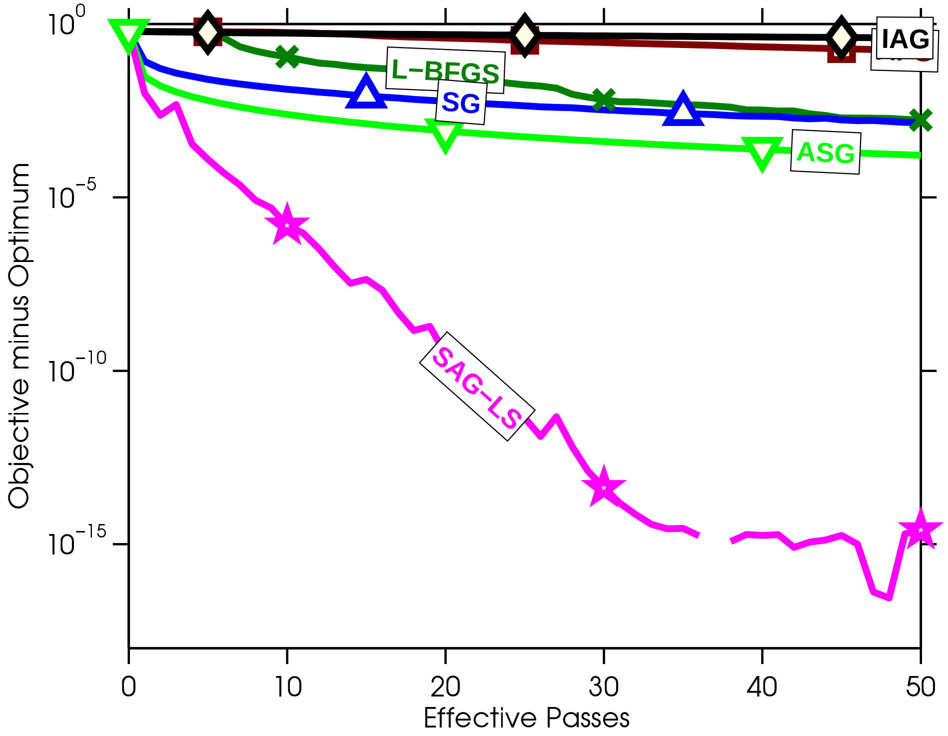


Experimental results (logistic regression)

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

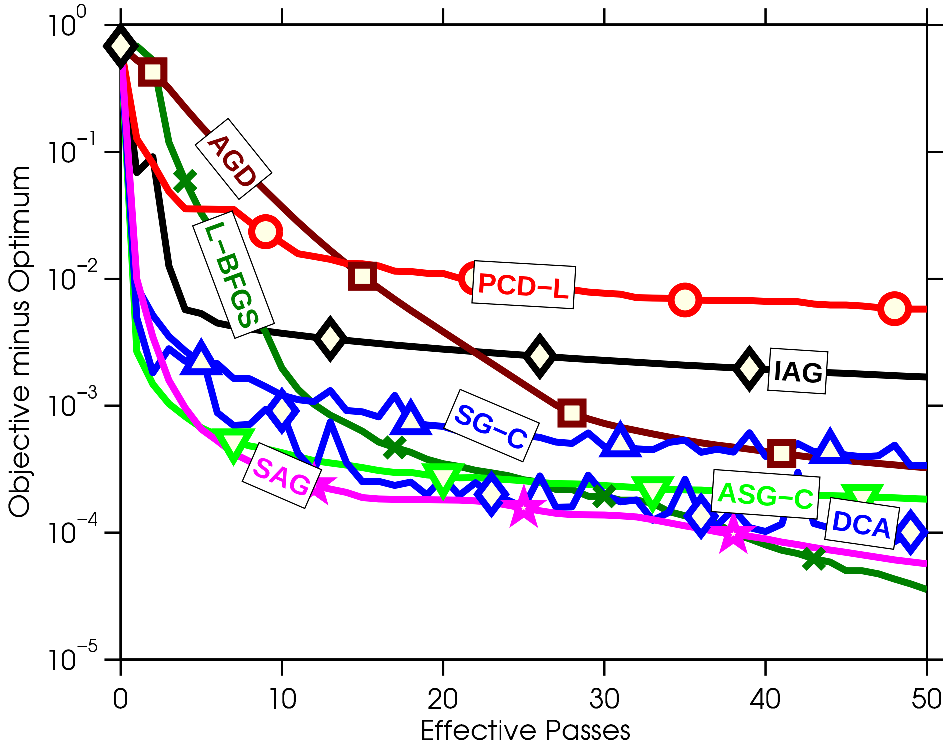
`rcv1` dataset
$(n = 697\,641,\ d = 47\,236)$



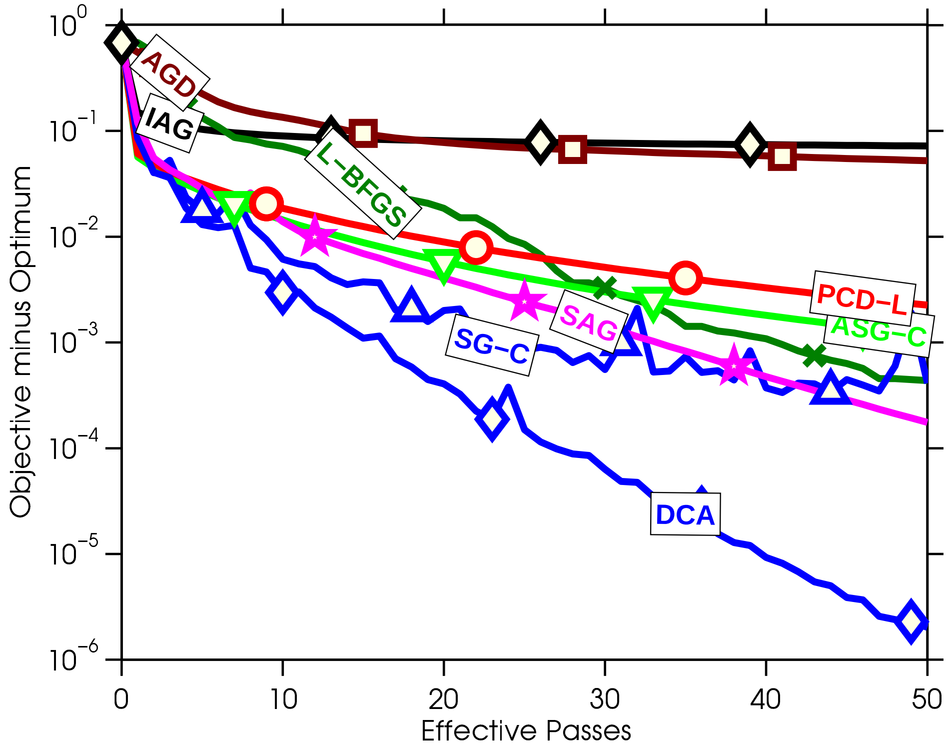


Before non-uniform sampling

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

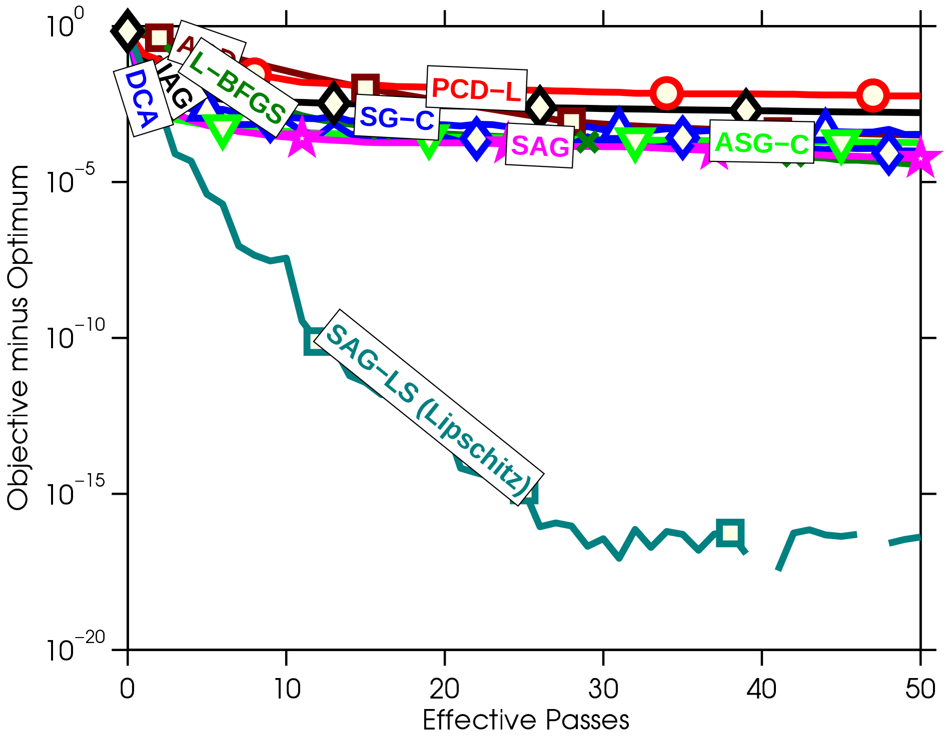


`sido` dataset
$(n = 12\ 678,\ d = 4\ 932)$

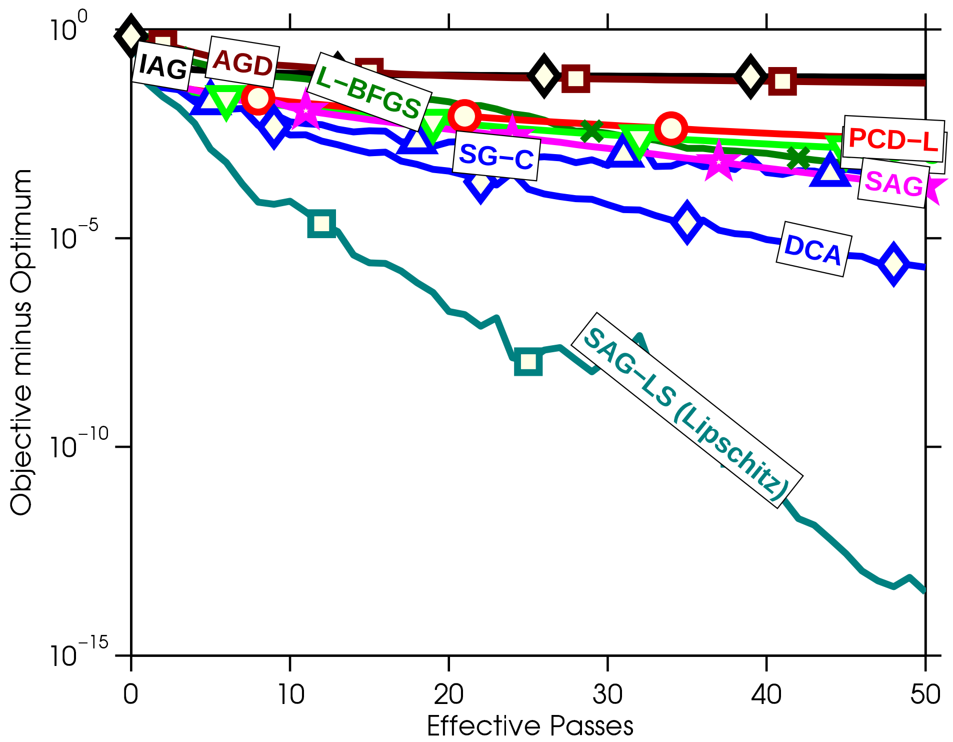


After non-uniform sampling

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



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



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

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

  - $\mathbb{E}Z_\alpha = \alpha\mathbb{E}X + (1-\alpha)\mathbb{E}Y$
  - $\mathrm{var}(Z_\alpha) = \alpha^2\big[\,\mathrm{var}(X) + \mathrm{var}(Y) - 2\mathrm{cov}(X,Y)\big]$
  - $\alpha = 1$: 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 $i_{\text{epoch}} = 1$ 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: $\tilde{\theta}$

– 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**: $\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) + h(\theta)$
  - 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**: SAG, SDCA, SVRG (Xiao and Zhang, 2014), SAGA, MISO (Mairal, 2015)

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**: SAG, SDCA, SVRG (Xiao and Zhang, 2014), SAGA, MISO (Mairal, 2015)

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	$d\times \quad (n+\sqrt{n\frac{L}{\mu}}) \quad \times \log\frac{1}{\varepsilon}$

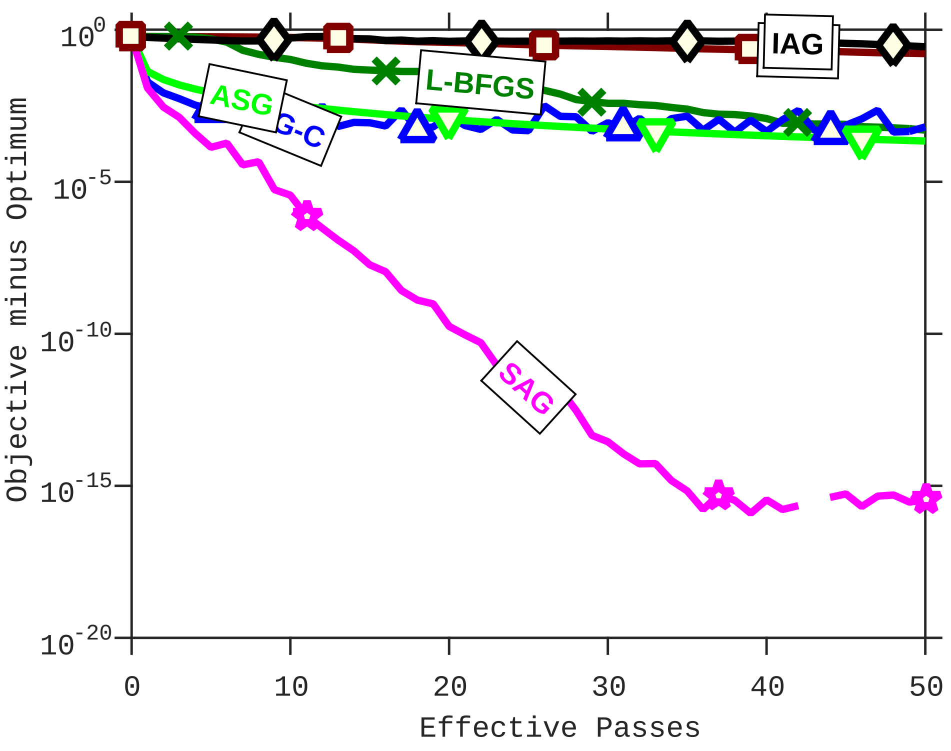
- **Acceleration for special algorithms** (e.g., Shalev-Shwartz and Zhang, 2014; Nitanda, 2014; Lan, 2015)

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

From training to testing errors

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

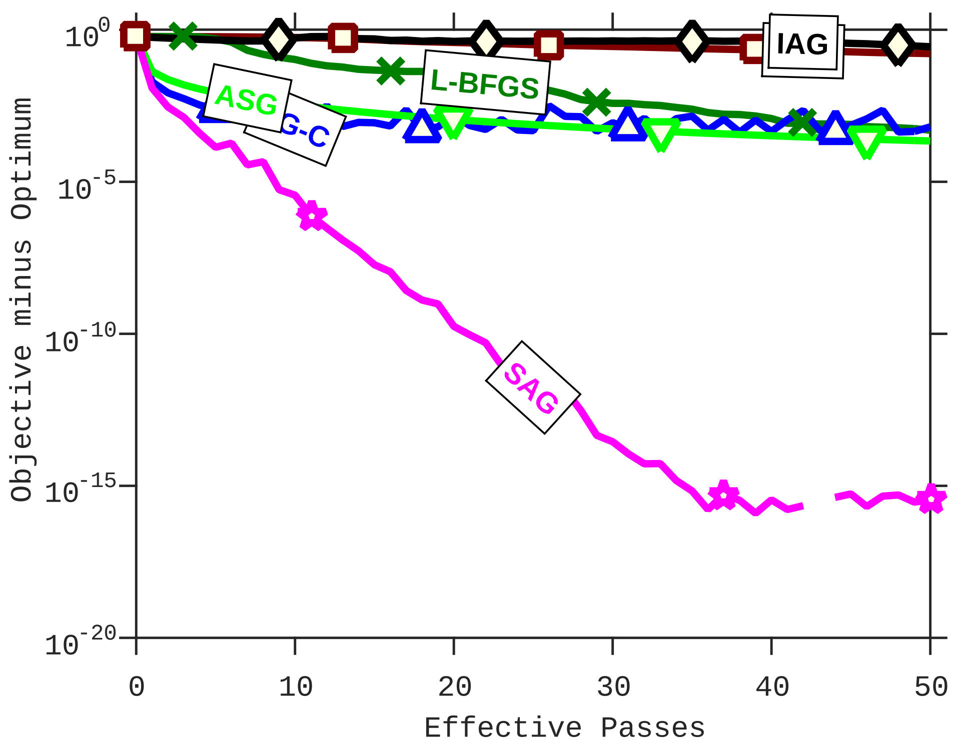
Training cost



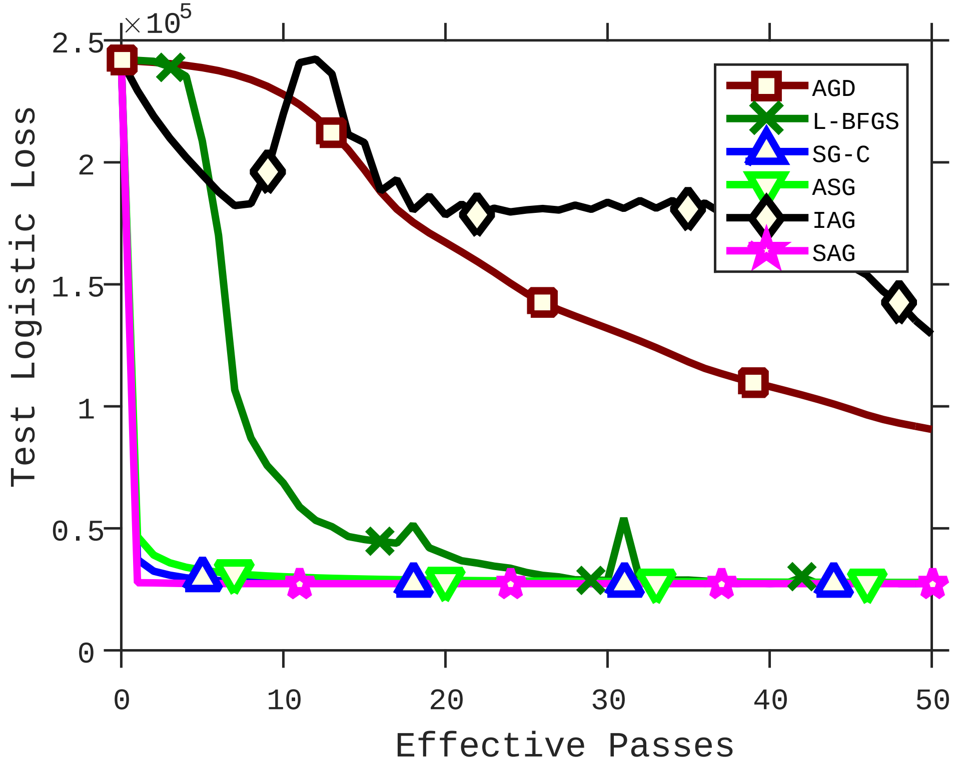
From training to testing errors

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

Training cost



Testing cost



SGD minimizes the testing cost!

- **Goal**: minimize $f(\theta) = \mathbb{E}_{p(x,y)} \ell(y, \theta^\top \Phi(x))$
  - Given n independent samples (x_i, y_i), $i = 1, \dots, n$ from $p(x, y)$
  - Given a single pass of stochastic gradient descent
  - Bounds on the excess testing cost $\mathbb{E} f(\bar{\theta}_n) - \inf_{\theta \in \mathbb{R}^d} f(\theta)$

SGD minimizes the testing cost!

- **Goal**: minimize $f(\theta) = \mathbb{E}_{p(x,y)} \ell(y, \theta^\top \Phi(x))$
  - Given n independent samples (x_i, y_i), $i = 1, \dots, n$ from $p(x, y)$
  - Given a single pass of stochastic gradient descent
  - Bounds on the excess testing cost $\mathbb{E} f(\bar{\theta}_n) - \inf_{\theta \in \mathbb{R}^d} f(\theta)$

- **Optimal convergence rates**: $O(1/\sqrt{n})$ and $O(1/(n\mu))$
  - Optimal for non-smooth losses (Nemirovski and Yudin, 1983)
  - Attained by averaged SGD with decaying step-sizes

SGD minimizes the testing cost!

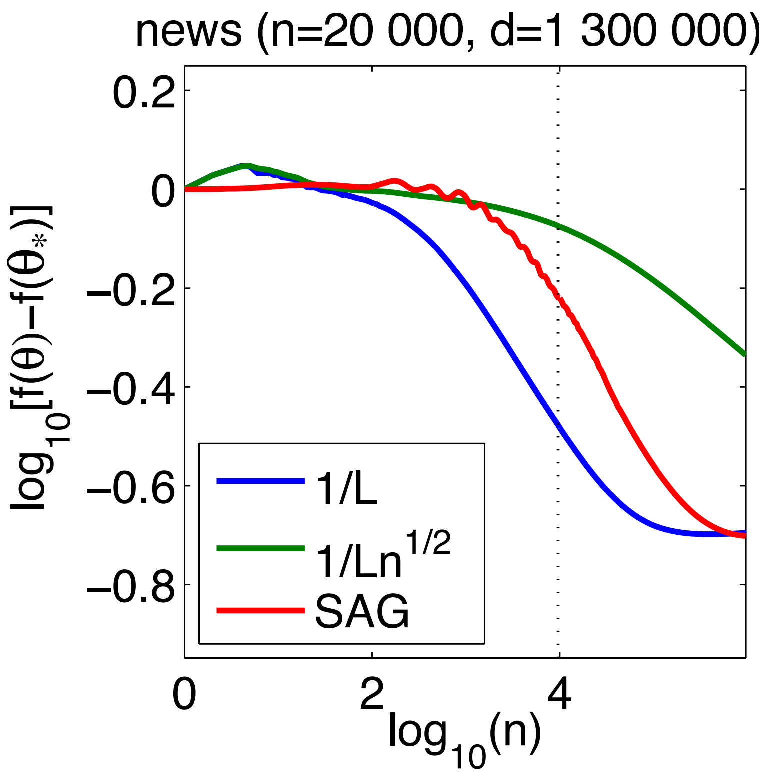
- **Goal**: minimize $f(\theta) = \mathbb{E}_{p(x,y)} \ell(y, \theta^\top \Phi(x))$
  - Given n independent samples (x_i, y_i), $i = 1, \dots, n$ from $p(x, y)$
  - Given a single pass of stochastic gradient descent
  - Bounds on the excess testing cost $\mathbb{E} f(\bar{\theta}_n) - \inf_{\theta \in \mathbb{R}^d} f(\theta)$

- **Optimal convergence rates**: $O(1/\sqrt{n})$ and $O(1/(n\mu))$
  - Optimal for non-smooth losses (Nemirovski and Yudin, 1983)
  - Attained by averaged SGD with decaying step-sizes

- **Constant-step-size SGD**
  - Linear convergence up to the noise level for strongly-convex problems (Solodov, 1998; Nedic and Bertsekas, 2000)
  - Full convergence and robustness to ill-conditioning?

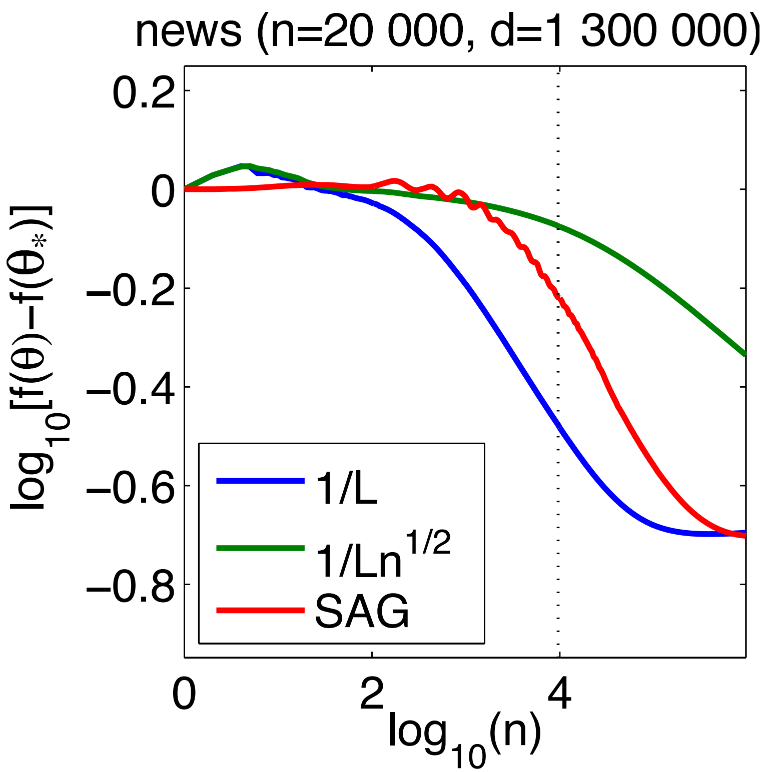
Robust averaged stochastic gradient (Bach and Moulines, 2013)

- **Constant-step-size SGD is convergent for least-squares**
  - Convergence rate in $O(1/n)$ without any dependence on μ
  - Simple choice of step-size (equal to $1/L$)



Robust averaged stochastic gradient (Bach and Moulines, 2013)

- **Constant-step-size SGD is convergent for least-squares**
  - Convergence rate in $O(1/n)$ without any dependence on μ
  - Simple choice of step-size (equal to $1/L$)



- Convergence in $O(1/n)$ for smooth losses with $O(d)$ online Newton step

Conclusions - Convex optimization

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

Conclusions - Convex optimization

- **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**
  - Extension to saddle-point problems (Balamurugan and Bach, 2016)
  - Lower bounds for finite sums (Agarwal and Bottou, 2015; Lan, 2015; Arjevani and Shamir, 2016)
  - Sampling without replacement (Gurbuzbalaban et al., 2015; Shamir, 2016)

Conclusions - Convex optimization

- **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**
  - Extension to saddle-point problems (Balamurugan and Bach, 2016)
  - Lower bounds for finite sums (Agarwal and Bottou, 2015; Lan, 2015; Arjevani and Shamir, 2016)
  - Sampling without replacement (Gurbuzbalaban et al., 2015; Shamir, 2016)
  - Bounds on testing errors for incremental methods (Frostig et al., 2015; Babanezhad et al., 2015)

Conclusions - Convex optimization

- **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**
  - Extension to saddle-point problems (Balamurugan and Bach, 2016)
  - Lower bounds for finite sums (Agarwal and Bottou, 2015; Lan, 2015; Arjevani and Shamir, 2016)
  - Sampling without replacement (Gurbuzbalaban et al., 2015; Shamir, 2016)
  - Bounds on testing errors for incremental methods (Frostig et al., 2015; Babanezhad et al., 2015)
- **What's next: non-convexity, parallelization, extensions/perspectives**

Postdoc opportunities in downtown Paris

- **Machine learning group at INRIA - Ecole Normale Supérieure**
  - Two postdoc positions (2 years)
  - One junior researcher position (4 years)

References

A. Agarwal and K. Bottou. A lower bound for the optimization of finite sums. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015.

Y. Arjevani and O. Shamir. Dimension-free iteration complexity of finite sum optimization problems. In *Advances In Neural Information Processing Systems*, 2016.

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*, 2015.

F. Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *Journal of Machine Learning Research*, 15(1):595–627, 2014.

F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Adv. NIPS*, 2013.

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. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 2016. 3rd edition.

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.

L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. Technical Report 1606.04838, arXiv, 2016.

S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

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, 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.

R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Competing with the empirical risk minimizer in a single pass. In *Proceedings of the Conference on Learning Theory*, 2015.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization. *Optimization Online*, July, 2010.

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.

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

A. Nedic and D. Bertsekas. Convergence rate of incremental subgradient algorithms. *Stochastic Optimization: Algorithms and Applications*, pages 263–304, 2000.

A. S. Nemirovski and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley & Sons, 1983.

Y. Nesterov. A method for solving a convex programming problem with rate of convergence $O(1/k^2)$. *Soviet Math. Doklady*, 269(3):543–547, 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.

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

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, In Press, 2016.

S. Shalev-Shwartz. Sdca without duality, regularization, and individual convexity. Technical Report 1602.01582, arXiv, 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.

O. Shamir. Without-replacement sampling for stochastic gradient methods: Convergence results and application to distributed optimization. Technical Report 1603.00570, arXiv, 2016.

M. V. Solodov. Incremental gradient algorithms with stepsizes bounded away from zero. *Computational Optimization and Applications*, 11(1):23–35, 1998.

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

P. Tseng. An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

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. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal* *of Machine Learning Research*, 9:2543–2596, 2010.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

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.