Stochastic gradient methods for machine learning

Francis Bach

INRIA - Ecole Normale Supérieure, Paris, France

Joint work with Eric Moulines, Nicolas Le Roux and Mark Schmidt - January 2013
Context

Machine learning for “big data”

- **Large-scale machine learning**: large \( p \), large \( n \), large \( k \)
  - \( p \) : dimension of each observation (input)
  - \( k \) : number of tasks (dimension of outputs)
  - \( n \) : number of observations

- **Examples**: computer vision, bioinformatics, signal processing

- **Ideal running-time complexity**: \( O(pn + kn) \)
Context

Machine learning for “big data”

- **Large-scale machine learning**: large $p$, large $n$, large $k$
  - $p$: dimension of each observation (input)
  - $k$: number of tasks (dimension of outputs)
  - $n$: number of observations

- **Examples**: computer vision, bioinformatics, signal processing

- **Ideal running-time complexity**: $O(pn + kn)$

- **Going back to simple methods**
  - Stochastic gradient methods (Robbins and Monro, 1951)
  - Mixing statistics and optimization
  - It is possible to improve on the sublinear convergence rate?
Outline

• Introduction
  – Supervised machine learning and convex optimization
  – Beyond the separation of statistics and optimization

• Stochastic approximation algorithms (Bach and Moulines, 2011)
  – Stochastic gradient and averaging
  – Strongly convex vs. non-strongly convex

• Going beyond stochastic gradient (Le Roux, Schmidt, and Bach, 2012)
  – More than a single pass through the data
  – Linear (exponential) convergence rate for strongly convex functions
Supervised machine learning

- **Data**: \( n \) observations \((x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n, \text{ i.i.d.}\)

- Prediction as a linear function \( \theta^\top \Phi(x) \) of features \( \Phi(x) \in \mathcal{F} = \mathbb{R}^p \)

- **(regularized) empirical risk minimization**: find \( \hat{\theta} \) solution of

\[
\min_{\theta \in \mathcal{F}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i)) + \mu \Omega(\theta)
\]

convex data fitting term + regularizer
Supervised machine learning

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

- Prediction as a linear function $\theta^\top \Phi(x)$ of features $\Phi(x) \in \mathcal{F} = \mathbb{R}^p$

- **(regularized) empirical risk minimization:** find $\hat{\theta}$ solution of

  $$\min_{\theta \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i)) + \mu \Omega(\theta)$$

  convex data fitting term + regularizer

- Empirical risk: $\hat{f}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$ training cost

- Expected risk: $f(\theta) = \mathbb{E}_{(x,y)} \ell(y, \theta^\top \Phi(x))$ testing cost

- **Two fundamental questions:** (1) computing $\hat{\theta}$ and (2) analyzing $\hat{\theta}$
Supervised machine learning

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

- Prediction as a linear function $\theta^\top \Phi(x)$ of features $\Phi(x) \in \mathcal{F} = \mathbb{R}^p$

- **(regularized) empirical risk minimization**: find $\hat{\theta}$ solution of

  $$\min_{\theta \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i)) + \mu \Omega(\theta)$$

  convex data fitting term + regularizer

- Empirical risk: $\hat{f}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$ training cost

- Expected risk: $f(\theta) = \mathbb{E}_{(x,y)} \ell(y, \theta^\top \Phi(x))$ testing cost

- **Two fundamental questions**: (1) computing $\hat{\theta}$ and (2) analyzing $\hat{\theta}$
  - May be tackled simultaneously
Smoothness and strong convexity

- A function \( g : \mathbb{R}^p \rightarrow \mathbb{R} \) is \( L \)-smooth if and only if it is differentiable and its gradient is \( L \)-Lipschitz-continuous

\[
\forall \theta_1, \theta_2 \in \mathbb{R}^p, \|g'(\theta_1) - g'(\theta_2)\| \leq L\|\theta_1 - \theta_2\|
\]

- If \( g \) is twice differentiable: \( \forall \theta \in \mathbb{R}^p, g''(\theta) \preceq L \cdot Id \)

\[
\text{smooth} \quad \text{non-smooth}
\]
Smoothness and strong convexity

- A function $g : \mathbb{R}^p \to \mathbb{R}$ is $L$-smooth if and only if it is differentiable and its gradient is $L$-Lipschitz-continuous

\[ \forall \theta_1, \theta_2 \in \mathbb{R}^p, \|g'(\theta_1) - g'(\theta_2)\| \leq L\|\theta_1 - \theta_2\| \]

- If $g$ is twice differentiable: $\forall \theta \in \mathbb{R}^p, g''(\theta) \preceq L \cdot I_d$

- Machine learning
  - with $g(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$
  - Hessian $\approx$ covariance matrix $\frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \Phi(x_i)^\top$
  - Bounded data
Smoothness and strong convexity

- A function $g: \mathbb{R}^p \to \mathbb{R}$ is $\mu$-strongly convex if and only if

  $$\forall \theta_1, \theta_2 \in \mathbb{R}^p, \ g(\theta_1) \geq g(\theta_2) + \langle g'(\theta_2), \theta_1 - \theta_2 \rangle + \frac{\mu}{2} \|\theta_1 - \theta_2\|^2$$

- Equivalent definition: $\theta \mapsto g(\theta) - \frac{\mu}{2} \|\theta\|^2$ is convex

- If $g$ is twice differentiable: $\forall \theta \in \mathbb{R}^p, \ g''(\theta) \succeq \mu \cdot Id$

---

**convex**

**strongly convex**
Smoothness and strong convexity

- A function $g : \mathbb{R}^p \to \mathbb{R}$ is $\mu$-strongly convex if and only if
  \[
  \forall \theta_1, \theta_2 \in \mathbb{R}^p, \ g(\theta_1) \geq g(\theta_2) + \langle g'(\theta_2), \theta_1 - \theta_2 \rangle + \frac{\mu}{2} \|\theta_1 - \theta_2\|^2
  \]

- Equivalent definition: $\theta \mapsto g(\theta) - \frac{\mu}{2} \|\theta\|^2$ is convex

- If $g$ is twice differentiable: $\forall \theta \in \mathbb{R}^p, \ g''(\theta) \succeq \mu \cdot Id$

- Machine learning
  - with $g(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$
  - Hessian $\approx$ covariance matrix $\frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \Phi(x_i)^\top$
  - Data with invertible covariance matrix (low correlation/dimension)
  - ... or with added regularization by $\frac{\mu}{2} \|\theta\|^2$
**Stochastic approximation**

- **Goal**: Minimizing a function $f$ defined on a Hilbert space $\mathcal{H}$
  - given only unbiased estimates $f_n'(\theta_n)$ of its gradients $f'(\theta_n)$ at certain points $\theta_n \in \mathcal{H}$

- **Stochastic approximation**
  - Observation of $f_n'(\theta_n) = f'(\theta_n) + \varepsilon_n$, with $\varepsilon_n =$ i.i.d. noise
  - Non-convex problems
Stochastic approximation

- **Goal**: Minimizing a function $f$ defined on a Hilbert space $\mathcal{H}$
  - given only unbiased estimates $f'_n(\theta_n)$ of its gradients $f'(\theta_n)$ at certain points $\theta_n \in \mathcal{H}$

- **Stochastic approximation**
  - Observation of $f'_n(\theta_n) = f'(\theta_n) + \varepsilon_n$, with $\varepsilon_n = \text{i.i.d. noise}$
  - Non-convex problems

- **Machine learning - statistics**
  - loss for a single pair of observations: $f_n(\theta) = \ell(y_n, \theta^\top \Phi(x_n))$
  - $f(\theta) = \mathbb{E} f_n(\theta) = \mathbb{E} \ell(y_n, \theta^\top \Phi(x_n)) = \text{generalization error}$
  - Expected gradient: $f'(\theta) = \mathbb{E} f'_n(\theta) = \mathbb{E} \{ \ell'(y_n, \theta^\top \Phi(x_n)) \Phi(x_n) \}$
Convex smooth stochastic approximation

- Key properties of $f$ and/or $f_n$
  - Smoothness: $f_n$ $L$-smooth
  - Strong convexity: $f$ $\mu$-strongly convex
Convex smooth stochastic approximation

- **Key properties of** $f$ and/or $f_n$
  - **Smoothness:** $f_n$ $L$-smooth
  - **Strong convexity:** $f$ $\mu$-strongly convex

- **Key algorithm:** Stochastic gradient descent (a.k.a. Robbins-Monro)

  $$
  \theta_n = \theta_{n-1} - \gamma_n f'_n(\theta_{n-1})
  $$

  - Polyak-Ruppert averaging: $\bar{\theta}_n = \frac{1}{n} \sum_{k=0}^{n-1} \theta_k$
  - Which learning rate sequence $\gamma_n$? Classical setting: $\gamma_n = Cn^{-\alpha}$
Convex smooth stochastic approximation

- **Key properties of** $f$ **and/or** $f_n$
  - **Smoothness**: $f_n$ $L$-smooth
  - **Strong convexity**: $f$ $\mu$-strongly convex

- **Key algorithm**: Stochastic gradient descent (a.k.a. Robbins-Monro)
  \[
  \theta_n = \theta_{n-1} - \gamma_n f'_n(\theta_{n-1})
  \]
  - Polyak-Ruppert averaging: $\bar{\theta}_n = \frac{1}{n} \sum_{k=0}^{n-1} \theta_k$
  - Which learning rate sequence $\gamma_n$? Classical setting: $\gamma_n = C n^{-\alpha}$

- **Desirable practical behavior**
  - Applicable (at least) to least-squares and logistic regression
  - Robustness to (potentially unknown) constants ($L$, $\mu$)
  - Adaptivity to difficulty of the problem (e.g., strong convexity)
Convex stochastic approximation
Related work

• Machine learning/optimization
  – Known minimax rates of convergence (Nemirovski and Yudin, 1983; Agarwal et al., 2010)
  – Strongly convex: $O(n^{-1})$
  – Non-strongly convex: $O(n^{-1/2})$
  – Achieved with and/or without averaging (up to log terms)
  – Non-asymptotic analysis (high-probability bounds)
  – Online setting and regret bounds
  – Bottou and Le Cun (2005); Bottou and Bousquet (2008); Hazan et al. (2007); Shalev-Shwartz and Srebro (2008); Shalev-Shwartz et al. (2007, 2009); Xiao (2010); Duchi and Singer (2009)
  – Nesterov and Vial (2008); Nemirovski et al. (2009)
Convex stochastic approximation

Related work

- Stochastic approximation
  - Asymptotic analysis
  - Non convex case with strong convexity around the optimum
  - $\gamma_n = Cn^{-\alpha}$ with $\alpha = 1$ is not robust to the choice of $C$
  - $\alpha \in (1/2, 1)$ is robust with averaging
  - Broadie et al. (2009); Kushner and Yin (2003); Kul’chitskii and Mozgovoï (1991); Fabian (1968)
  - Polyak and Juditsky (1992); Ruppert (1988)
Problem set-up - General assumptions

- **Unbiased gradient estimates:**
  - $f_n(\theta)$ is of the form $h(z_n, \theta)$, where $z_n$ is an i.i.d. sequence
  - e.g., $f_n(\theta) = h(z_n, \theta) = \ell(y_n, \theta^\top \Phi(x_n))$ with $z_n = (x_n, y_n)$
  - NB: can be generalized

- **Variance of estimates:** There exists $\sigma^2 \geq 0$ such that for all $n \geq 1$,
  $\mathbb{E}(\|f'_n(\theta^*) - f'(\theta^*)\|^2) \leq \sigma^2$, where $\theta^*$ is a global minimizer of $f$
Problem set-up - Smoothness/convexity assumptions

- **Smoothness of** $f_n$: For each $n \geq 1$, the function $f_n$ is a.s. convex, differentiable with $L$-Lipschitz-continuous gradient $f'_n$:
  - Bounded data
Problem set-up - Smoothness/convexity assumptions

- **Smoothness of** $f_n$: For each $n \geq 1$, the function $f_n$ is a.s. convex, differentiable with $L$-Lipschitz-continuous gradient $f'_n$:
  
  - Bounded data

- **Strong convexity of** $f$: The function $f$ is strongly convex with respect to the norm $\| \cdot \|$, with convexity constant $\mu > 0$:
  
  - Invertible population covariance matrix
  - or regularization by $\frac{\mu}{2} \| \theta \|^2$
Summary of new results (Bach and Moulines, 2011)

- Stochastic gradient descent with learning rate $\gamma_n = Cn^{-\alpha}$

- Strongly convex smooth objective functions
  - Old: $O(n^{-1})$ rate achieved without averaging for $\alpha = 1$
  - New: $O(n^{-1})$ rate achieved with averaging for $\alpha \in [1/2, 1]$
  - Non-asymptotic analysis with explicit constants
  - Forgetting of initial conditions
  - Robustness to the choice of $C$
Summary of new results (Bach and Moulines, 2011)

- Stochastic gradient descent with learning rate $\gamma_n = Cn^{-\alpha}$

- Strongly convex smooth objective functions
  - Old: $O(n^{-1})$ rate achieved without averaging for $\alpha = 1$
  - New: $O(n^{-1})$ rate achieved with averaging for $\alpha \in [1/2, 1]$
  - Non-asymptotic analysis with explicit constants
  - Forgetting of initial conditions
  - Robustness to the choice of $C$

- Proof technique
  - Derive deterministic recursion for $\delta_n = \mathbb{E} \| \theta_n - \theta^* \|^2$
    
    \[
    \delta_n \leq (1 - 2\mu \gamma_n + 2L^2 \gamma_n^2) \delta_{n-1} + 2\sigma^2 \gamma_n^2
    \]
  - Mimic SA proof techniques in a non-asymptotic way
Summary of new results (Bach and Moulines, 2011)

- Stochastic gradient descent with learning rate $\gamma_n = C n^{-\alpha}$

- **Strongly convex smooth objective functions**
  - Old: $O(n^{-1})$ rate achieved without averaging for $\alpha = 1$
  - New: $O(n^{-1})$ rate achieved with averaging for $\alpha \in [1/2, 1]$
  - Non-asymptotic analysis with explicit constants
  - Forgetting of initial conditions
  - Robustness to the choice of $C$

- **Convergence rates** for $\mathbb{E}\|\theta_n - \theta^*\|^2$ and $\mathbb{E}\|\bar{\theta}_n - \theta^*\|^2$
  - no averaging: $O\left(\frac{\sigma^2 \gamma_n}{\mu}\right) + O(e^{-\mu n \gamma_n})\|\theta_0 - \theta^*\|^2$
  - averaging: $\frac{\text{tr} \ H(\theta^*)^{-1}}{n} + \mu^{-1} O(n^{-2\alpha} + n^{-2+\alpha}) + O\left(\frac{\|\theta_0 - \theta^*\|^2}{\mu^2 n^2}\right)$
Summary of new results (Bach and Moulines, 2011)

• Stochastic gradient descent with learning rate $\gamma_n = C n^{-\alpha}$

• Strongly convex smooth objective functions
  – Old: $O(n^{-1})$ rate achieved without averaging for $\alpha = 1$
  – New: $O(n^{-1})$ rate achieved with averaging for $\alpha \in [1/2, 1]$
  – Non-asymptotic analysis with explicit constants
Summary of new results (Bach and Moulines, 2011)

• Stochastic gradient descent with learning rate $\gamma_n = Cn^{-\alpha}$

• Strongly convex smooth objective functions
  – Old: $O(n^{-1})$ rate achieved without averaging for $\alpha = 1$
  – New: $O(n^{-1})$ rate achieved with averaging for $\alpha \in [1/2, 1]$
  – Non-asymptotic analysis with explicit constants

• Non-strongly convex smooth objective functions
  – Old: $O(n^{-1/2})$ rate achieved with averaging for $\alpha = 1/2$
  – New: $O(\max\{n^{1/2-3\alpha/2}, n^{-\alpha/2}, n^{\alpha-1}\})$ rate achieved without averaging for $\alpha \in [1/3, 1]$

• Take-home message
  – Use $\alpha = 1/2$ with averaging to be adaptive to strong convexity
Conclusions / Extensions

Stochastic approximation for machine learning

• Mixing convex optimization and statistics
  – Non-asymptotic analysis through moment computations
  – Averaging with longer steps is (more) robust and adaptive
Conclusions / Extensions

Stochastic approximation for machine learning

• Mixing convex optimization and statistics
  – Non-asymptotic analysis through moment computations
  – Averaging with longer steps is (more) robust and adaptive

• Future/current work - open problems
  – High-probability through all moments $\mathbb{E}\|\theta_n - \theta^*\|^{2d}$
  – Analysis for logistic regression using self-concordance (Bach, 2010)
  – Including a non-differentiable term (Xiao, 2010; Lan, 2010)
  – Non-random errors (Schmidt, Le Roux, and Bach, 2011)
  – Line search for stochastic gradient
  – Non-parametric stochastic approximation
  – Online estimation of uncertainty
  – Going beyond a single pass through the data
Going beyond a single pass over the data

- **Stochastic approximation**
  - Assumes infinite data stream
  - Observations are used only once
  - Directly minimizes testing cost $\mathbb{E}_zh(\theta, z) = \mathbb{E}_{(x,y)} \ell(y, \theta^\top \Phi(x))$
Going beyond a single pass over the data

- **Stochastic approximation**
  - Assumes infinite data stream
  - Observations are used only once
  - Directly minimizes testing cost $\mathbb{E}_zh(\theta, z) = \mathbb{E}_{(x,y)} \ell(y, \theta^\top \Phi(x))$

- **Machine learning practice**
  - Finite data set $(z_1, \ldots, z_n)$
  - Multiple passes
  - Minimizes training cost $\frac{1}{n} \sum_{i=1}^{n} h(\theta, z_i) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \theta^\top \Phi(x_i))$
  - Need to regularize (e.g., by the $\ell_2$-norm) to avoid overfitting
Stochastic vs. deterministic methods

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

- 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
    - Iteration complexity is linear in $n$
Stochastic vs. deterministic methods

- Minimizing \( g(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta) \) with \( f_i(\theta) = \ell(y_i, \theta^\top \Phi(x_i)) + \mu \Omega(\theta) \)

- 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 \( f_i(\theta) = \ell(y_i, \theta^T \Phi(x_i)) + \mu \Omega(\theta) \)

- **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
  - 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 \( \{1, \ldots, n\} \)
  - Convergence rate in \( O(1/t) \)
  - Iteration complexity is independent of \( n \)
**Stochastic vs. deterministic methods**

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

- **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 $O(1)$ iteration cost
Stochastic vs. deterministic methods

- **Goal** = best of both worlds: linear rate with $O(1)$ iteration cost
Accelerating gradient methods - Related work

• **Nesterov acceleration**
  - Better linear rate but still $O(n)$ iteration cost

• **Hybrid methods, incremental average gradient, increasing batch size**
  - Bertsekas (1997); Blatt et al. (2008); Friedlander and Schmidt (2011)
  - Linear rate, but iterations make full passes through the data.
**Accelerating gradient methods - Related work**

- **Momentum, gradient/iterate averaging, stochastic version of accelerated batch gradient methods**
  - Polyak and Juditsky (1992); Tseng (1998); Sunehag et al. (2009); Ghadimi and Lan (2010); Xiao (2010)
  - Can improve constants, but still have sublinear $O(1/t)$ rate

- **Constant step-size stochastic gradient (SG), accelerated SG**
  - Kesten (1958); Delyon and Juditsky (1993); Solodov (1998); Nedic and Bertsekas (2000)
  - Linear convergence, but only up to a fixed tolerance.

- **Stochastic methods in the dual**
  - Shalev-Shwartz and Zhang (2012)
  - Linear rate but limited choice for the $f_i$'s
Stochastic average gradient
(Le Roux, Schmidt, and Bach, 2012)

- **Stochastic average gradient (SAG) iteration**
  - Keep in memory the gradients of all functions \( f_i, i = 1, \ldots, n \)
  - Random selection \( i(t) \in \{1, \ldots, n\} \) 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 $f_i$, $i = 1, \ldots, n$
  - Random selection $i(t) \in \{1, \ldots, n\}$ 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**
  - Supervised machine learning
    - If $f_i(\theta) = \ell_i(y_i, \Phi(x_i)^\top \theta)$, then $f_i'(\theta) = \ell_i'(y_i, \Phi(x_i)^\top \theta) \Phi(x_i)$
    - Only need to store $n$ real numbers
Stochastic average gradient
Convergence analysis - I

• Assume each $f_i$ is $L$-smooth and $\hat{f} = \frac{1}{n} \sum_{i=1}^{n} f_i$ is $\mu$-strongly convex

• Constant step size $\gamma_t = \frac{1}{2nL}$:

$$\mathbb{E}[\|\theta_t - \theta^*\|^2] \leq \left(1 - \frac{\mu}{8Ln}\right)^t \left[3\|\theta_0 - \theta^*\|^2 + \frac{9\sigma^2}{4L^2}\right]$$

  – Linear rate with iteration cost independent of $n$ ...
  – ... but, same behavior as batch gradient and IAG (cyclic version)

• Proof technique

  – Designing a quadratic Lyapunov function for a $n$-th order non-linear stochastic dynamical system
Stochastic average gradient
Convergence analysis - II

• Assume each $f_i$ is $L$-smooth and $\hat{f} = \frac{1}{n} \sum_{i=1}^{n} f_i$ is $\mu$-strongly convex

• Constant step size $\gamma_t = \frac{1}{2n\mu}$, if $\frac{\mu}{L} \geq \frac{8}{n}$

$$\mathbb{E}[\hat{f}(\theta_t) - \hat{f}(\theta^*)] \leq C\left(1 - \frac{1}{8n}\right)^t$$

with $C = \left[\frac{16L}{3n} \|\theta_0 - \theta^*\|^2 + \frac{4\sigma^2}{3n\mu} \left(8 \log \left(1 + \frac{\mu n}{4L}\right) + 1\right)\right]$

- Linear rate with iteration cost independent of $n$
- Linear convergence rate “independent” of the condition number
- After each pass through the data, constant error reduction
Rate of convergence comparison

• Assume that $L = 100$, $\mu = .01$, and $n = 80000$
  
  – Full gradient method has rate
    $$\left( 1 - \frac{\mu}{L} \right) = 0.9999$$
  
  – Accelerated gradient method has rate
    $$\left( 1 - \sqrt{\frac{\mu}{L}} \right) = 0.9900$$
  
  – Running $n$ iterations of SAG for the same cost has rate
    $$\left( 1 - \frac{1}{8n} \right)^n = 0.8825$$
  
  – Fastest possible first-order method has rate
    $$\left( \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2 = 0.9608$$

• Beating two lower bounds (with additional assumptions)
  
  – (1) stochastic gradient and (2) full gradient
Stochastic average gradient
Implementation details and extensions

- The algorithm can use sparsity in the features to reduce the storage and iteration cost

- Grouping functions together can further reduce the memory requirement

- We have obtained good performance when $L$ is not known with a heuristic line-search

- Algorithm allows non-uniform sampling

- Possibility of making proximal, coordinate-wise, and Newton-like variants
Stochastic average gradient
Simulation experiments

- protein dataset \( (n = 145751, p = 74) \)
- Dataset split in two (training/testing)

![Graph showing training and testing cost with various algorithms: Steepest, AFG, L-BFGS, pegasos, RDA, SAG \((2/(L+n\mu))\), SAG-LS.]
Stochastic average gradient
Simulation experiments

- cover type dataset \((n = 581012, p = 54)\)

- Dataset split in two (training/testing)
Conclusions / Extensions

Stochastic average gradient

- Going beyond a single pass through the data
  - Keep memory of all gradients for finite training sets
  - Linear convergence rate with $O(1)$ iteration complexity
  - Randomization leads to easier analysis and faster rates
  - Beyond machine learning
Conclusions / Extensions
Stochastic average gradient

- Going beyond a single pass through the data
  - Keep memory of all gradients for finite training sets
  - Linear convergence rate with $O(1)$ iteration complexity
  - Randomization leads to easier analysis and faster rates
  - Beyond machine learning

- Future/current work - open problems
  - Including a non-differentiable term
  - Line search
  - Using second-order information or non-uniform sampling
  - Going beyond finite training sets (bound on testing cost)
  - Non strongly-convex case
References


