Statistical Learning Models for Text and Graph Data Unconstrained Optimization Techniques

Yangqiu Song

Hong Kong University of Science and Technology

yqsong@cse.ust.hk

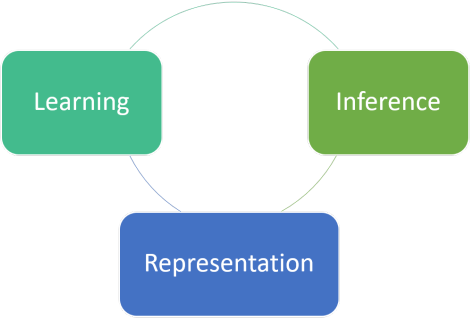
October 9, 2019

$*$Contents are based on materials created by Peter Richtérik, Mark Schmidt, Francis Bach, Tianbao Yang, Rong Jin, Shenghuo Zhu, and Qihang Lin

Reference Content

- Peter Richtérik and Mark Schmidt. ICML Tutorial on Modern Convex Optimization Methods for Large-scale Empirical Risk Minimization. https://icml.cc/2015/tutorials/2015_ICML_ConvexOptimization_I.pdf
- Francis Bach. NIPS 2016 Tutorial on Large-Scale Optimization: Beyond Stochastic Gradient Descent and Convexity. http://www.di.ens.fr/~fbach/fbach_tutorial_vr_nips_2016.pdf and http://www.di.ens.fr/~fbach/ssra_tutorial_vr_nips_2016.pdf
- Tianbao Yang, Qihang Lin, and Rong Jin. KDD Tutorial on Big Data Analytics: Optimization and Randomization. http://homepage.cs.uiowa.edu/~tyng/kdd15tutorial.html
- Tianbao Yang, Rong Jin and Shenghuo Zhu. SDM Tutorial on Stochastic Optimization for Big Data Analytics: Algorithms and Library. http://homepage.divms.uiowa.edu/~tyng/tutorial.html

Course Organization



- Representation: language models, word embeddings, topic models, knowledge graphs
- Learning: supervised learning, semi-supervised learning, distant supervision, indirect supervision, sequence models, deep learning, optimization techniques
- Inference: constraint modeling, joint inference, search algorithms

Overview

1. Introduction
2. Background
3. Unconstrained Convex Optimization
   - Gradient Based Optimization
   - Stochastic Subgradient
   - Finite-Sum Methods
   - Non-Smooth Objectives
4. Optimization for Neural Networks

Overview

1. Introduction

2. Background

3. Unconstrained Convex Optimization
   - Gradient Based Optimization
   - Stochastic Subgradient
   - Finite-Sum Methods
   - Non-Smooth Objectives

4. Optimization for Neural Networks

Big-N Problems

Recall the regularized empirical risk minimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{N} \underbrace{\sum_{i=1}^{N} \ell(\mathbf{w}, \mathbf{x}_i, y_i)}_{\text{Empirical Loss/Data Fitting}} + \underbrace{\lambda r(\mathbf{w})}_{\text{Regularization}}$$

- What if number of training examples N is very large?

Stochastic vs. Deterministic Gradient Methods

- We consider minimizing $f(\mathbf{w}) = \sum_{i=1}^{N} f_i(\mathbf{w})$
- Deterministic gradient method (Cauchy (1847)):

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta^t \nabla f(\mathbf{w}^t) = \mathbf{w}^t - \frac{\eta^t}{N} \sum_{i=1}^{N} \nabla f_i(\mathbf{w}^t)$$

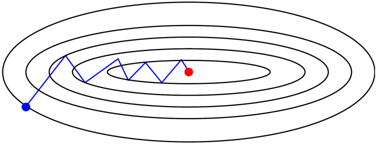
  - Iteration cost is linear in N
  - Convergence with constant η^t or line-search
- Stochastic gradient method (Robbins and Monro (1951)):
  - Random selection of i from $\{1, 2, \ldots, N\}$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta^t f_i'(\mathbf{w}^t)$$

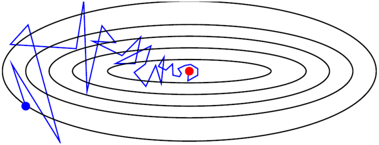
  - Gives unbiased estimate of true gradient, $\mathbb{E}[f_i'(\mathbf{w})] = \frac{1}{N}\nabla f_i(\mathbf{w}) = \nabla f(\mathbf{w})$
  - Iteration cost is independent of N
  - Convergence requires $\eta^t \to 0$

Stochastic vs. Deterministic Gradient Methods

- We consider minimizing $f(\mathbf{w}) = \sum_{i=1}^{N} f_i(\mathbf{w})$
- Deterministic gradient method (Cauchy (1847)):



- Stochastic gradient method (Robbins and Monro (1951)):



Stochastic vs. Deterministic Gradient Methods

Stochastic iterations are N times faster, but how many iterations?

Assumption	Deterministic	Stochastic
Convex	$O(\frac{1}{T^2})$	$O(\frac{1}{\sqrt{T}})$
Strongly-Convex	$O((1-\mu/L)^\top)$	$O(\frac{1}{T})$

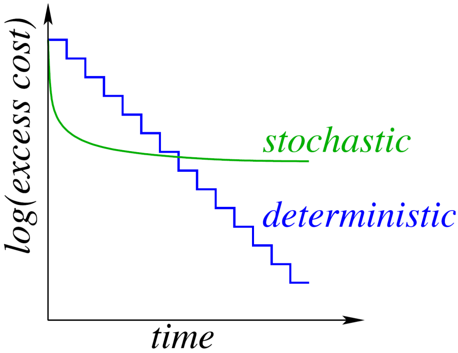
Proof:

`https://www.cs.rochester.edu/u/jliu/CSC-576/class-note-10.pdf`

- Stochastic has low iteration cost but slow convergence rate
    - Sublinear rate even in strongly-convex case

Stochastic vs. Deterministic Convergence Rates

Plot of convergence rates in strongly-convex case:



Stochastic will be superior for low-accuracy/time situations.

Stochastic vs. Deterministic for Non-Smooth

- Consider the binary support vector machine objective:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{N} \max\{0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i)\} + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

- Rates for subgradient methods for non-smooth objectives (Shalev-Shwartz et al. (2011)):

Assumption	Deterministic	Stochastic
Convex	$O(\frac{1}{\sqrt{T}})$	$O(\frac{1}{\sqrt{T}})$
Strongly-Convex	$O(\frac{1}{T})$	$O(\frac{1}{T})$

- Other black-box methods (cutting plane) are not faster
- For non-smooth problems:
  - Stochastic methods have same rate as smooth case
  - Deterministic methods are not faster than stochastic method
  - So use stochastic subgradient (iterations are n times faster)

Sub-Gradients and Sub-Differentials

Recall that for differentiable convex functions we have

$$f(\mathbf{w}') \geq f(\mathbf{w}) + \nabla f(\mathbf{w})^\top (\mathbf{w}' - \mathbf{w}), \forall \mathbf{w}, \mathbf{w}'$$

A vector **d** is a subgradient of a convex function f at **w** if

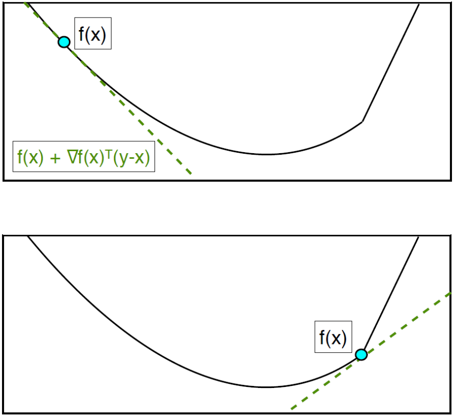
$$f(\mathbf{w}') \geq f(\mathbf{w}) + \mathbf{d}^\top (\mathbf{w}' - \mathbf{w}), \forall \mathbf{w}, \mathbf{w}'$$

- At differentiable **w**:
  - Only subgradient is $\nabla f(\mathbf{w})$
- At non-differentiable **w**:
  - We have a set of subgradients
  - Called the sub-differential, $\partial f(\mathbf{w})$
- Note that $\mathbf{0} \in \partial f(\mathbf{w})$ if **w** is a global minimum

Sub-Gradients and Sub-Differentials

A vector **d** is a subgradient of a convex function f at **w** if

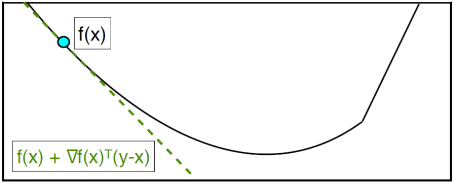
$$f(\mathbf{w}') \geq f(\mathbf{w}) + \mathbf{d}^\top(\mathbf{w}' - \mathbf{w}), \forall \mathbf{w}, \mathbf{w}'$$

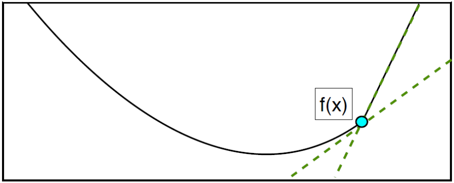


Sub-Gradients and Sub-Differentials

A vector **d** is a subgradient of a convex function f at **w** if

$$f(\mathbf{w}') \geq f(\mathbf{w}) + \mathbf{d}^\top(\mathbf{w}' - \mathbf{w}), \forall \mathbf{w}, \mathbf{w}'$$

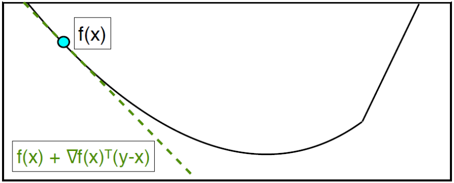


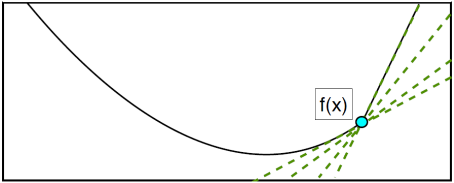


Sub-Gradients and Sub-Differentials

A vector **d** is a subgradient of a convex function f at **w** if

$$f(\mathbf{w}') \geq f(\mathbf{w}) + \mathbf{d}^\top(\mathbf{w}' - \mathbf{w}), \forall \mathbf{w}, \mathbf{w}'$$

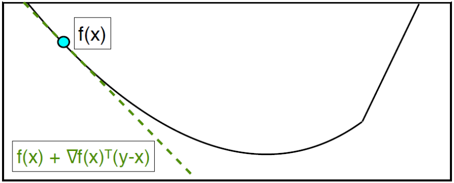


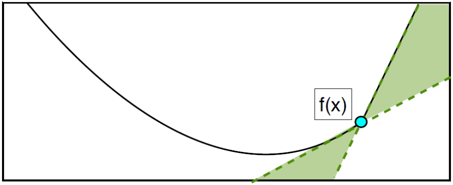


Sub-Gradients and Sub-Differentials

A vector **d** is a subgradient of a convex function f at **w** if

$$f(\mathbf{w}') \geq f(\mathbf{w}) + \mathbf{d}^\top(\mathbf{w}' - \mathbf{w}), \forall \mathbf{w}, \mathbf{w}'$$

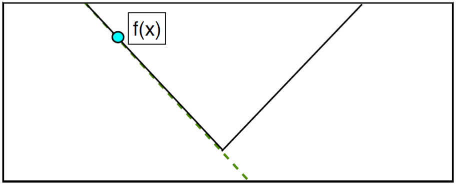




Sub-Differential of Absolute Value and Max Functions

- Sub-differential of absolute value function (sign of the variable if non-zero, anything in [-1, 1] at 0):

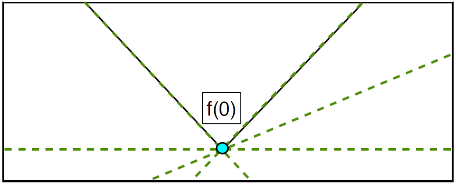
$$\partial|x| = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \\ [-1,1] & x = 0 \end{cases}$$



Sub-Differential of Absolute Value and Max Functions

- Sub-differential of absolute value function (sign of the variable if non-zero, anything in [-1, 1] at 0):

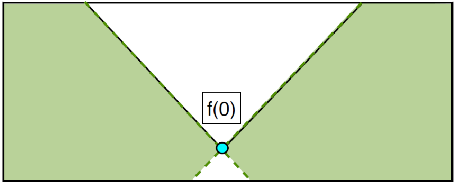
$$\partial|x| = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \\ [-1, 1] & x = 0 \end{cases}$$



Sub-Differential of Absolute Value and Max Functions

- Sub-differential of absolute value function (sign of the variable if non-zero, anything in [-1, 1] at 0):

$$\partial|x| = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \\ [-1,1] & x = 0 \end{cases}$$



Sub-Differential of Absolute Value and Max Functions

- Sub-differential of absolute value function (sign of the variable if non-zero, anything in [-1, 1] at 0):

$$\partial|x| = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \\ [-1, 1] & x = 0 \end{cases}$$

- Sub-differential of max function (any convex combination of the gradients of the argmax):

$$\partial \max\{f_1(x), f_2(x)\} = \begin{cases} \nabla f_1(x) & f_1(x) > f_2(x) \\ \nabla f_2(x) & f_1(x) < f_2(x) \\ \theta \nabla f_1(x) + (1-\theta)\nabla f_2(x) & f_1(x) = f_2(x) \end{cases}$$

Subgradient and Stochastic Subgradient methods

- The basic subgradient method:

  $$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \mathbf{d}$$

  for some $\mathbf{d} \in \partial f(\mathbf{w}^t)$
- For convergence, we require $\eta \to 0$
- The basic stochastic subgradient method:

  $$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \mathbf{d}$$

  for some $\mathbf{d} \in \partial f_i(\mathbf{w}^t)$ for some random $i \in \{1, 2, \ldots, N\}$

Stochastic Subgradient Methods in Practice

- The theory says to use decreasing sequence $\eta^t = 1/\lambda t$

$$i_t = \text{RAND}(1, \ldots, N), \eta^t = 1/\lambda t$$
$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta^t f'_{i_t}(\mathbf{w}^t)$$

  - $O(1/t)$ for smooth objectives
  - $O(\log(t)/t)$ for non-smooth objectives (Shamir and Zhang (2013))
- Except for some special cases, you should not do this
  - Initial steps are huge: usually $\lambda = O(1/N)$ or $O(1/\sqrt{N})$
  - Later steps are tiny: $1/t$ gets small very quickly
  - Convergence rate is not robust to mis-specification of λ
- Tricks that can improve theoretical and practical properties
  1. Use smaller initial step-sizes, that go to zero more slowly
  2. Take a (weighted) average of the iterations or gradients:

$$\bar{\mathbf{w}}^t = \sum_{i=1}^{t} \omega^t \mathbf{w}^t \quad \bar{\mathbf{d}}^t = \sum_{i=1}^{t} \delta^t \mathbf{d}^t$$

Speeding up Stochastic Subgradient Methods

Works that support using large steps and averaging:

- Gradient averaging all previous steps improves constants ("dual averaging") (Nesterov (2007)); Finds non-zero variables with sparse regularizers. (Xiao (2010))
- Averaging later iterations achieves $O(1/t)$ in non-smooth case. (Rakhlin et al. (2012))
- $\eta^t = O(1/t^\beta)$ for $\beta \in (0.5, 1)$ more robust than $\eta^t = O(1/t)$ (Bach and Moulines (2011))

Overview

1 Introduction

2 Background

3 Unconstrained Convex Optimization

- Gradient Based Optimization
- Stochastic Subgradient
- Finite-Sum Methods
- Non-Smooth Objectives

4 Optimization for Neural Networks

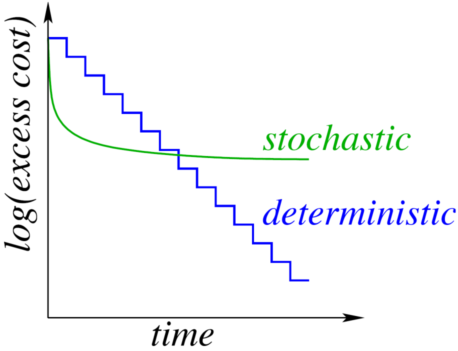
Big-N Problems

Recall the regularized empirical risk minimization problem:

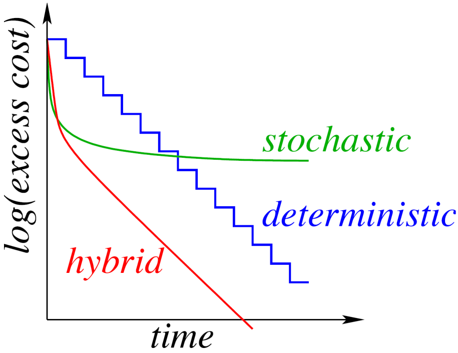
$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{N} \underbrace{\sum_{i=1}^{N} \ell(\mathbf{w}, \mathbf{x}_i, y_i)}_{\text{Empirical Loss/Data Fitting}} + \underbrace{\lambda r(\mathbf{w})}_{\text{Regularization}}$$

- Stochastic methods:
  - $O(1/t)$ convergence but requires 1 gradient per iterations
  - Rates are unimprovable for general stochastic objectives
- Deterministic methods:
  - $O(\rho^t)$ convergence but requires N gradients per iteration
  - The faster rate is possible because N is finite
- For minimizing finite sums, can we design a better method?

Motivation for Hybrid Methods



Motivation for Hybrid Methods



Hybrid Deterministic-Stochastic

- Control the sample size
- The FG method uses all N gradients,

$$\nabla f(\mathbf{w}^t) = \frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\mathbf{w}^t)$$

- The SG method approximates it with 1 sample,

$$\nabla f_{i_t}(\mathbf{w}^t) \approx \frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\mathbf{w}^t)$$

- A common variant is to use larger sample $\mathcal{B}^t$,

$$\sum_{i=1}^{|\mathcal{B}^t|}\nabla f_i(\mathbf{w}^t) \approx \frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\mathbf{w}^t)$$

Batching

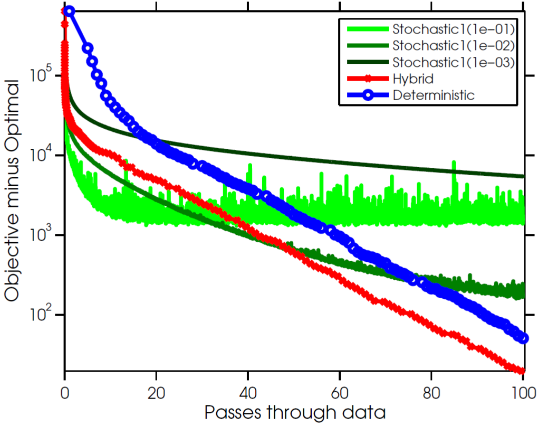
- The SG method with a sample $\mathcal{B}^t$ uses iterations

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \frac{\eta^t}{|\mathcal{B}^t|} \sum_{i \in \mathcal{B}^t}^{|\mathcal{B}^t|} \nabla f_i(\mathbf{w}^t)$$

- For a fixed sample size $|\mathcal{B}^t|$, the rate is sublinear
- Gradient error decreases as sample size $|\mathcal{B}^t|$ increases
- Common to gradually increase the sample size $|\mathcal{B}^t|$ (Bertsekas and Tsitsiklis (1996))
- We can choose $|\mathcal{B}^t|$ to achieve a linear convergence rate:
    - Early iterations are cheap like SG iterations
    - Later iterations can use a Newton-like method

Evaluation on Chain-Structured CRFs

Results on chain-structured conditional random field:



Stochastic Average Gradient (SAG)

- Growing $|\mathcal{B}^t|$ eventually requires $O(N)$ iteration cost
- Can we have a rate of $O(\mu^t)$ with only 1 gradient evaluation per iteration?
- YES! The stochastic average gradient (SAG) algorithm (Roux et al. (2012)):
  - Randomly select i_t from $\{1, 2, \ldots, N\}$ and compute $\nabla f_{i_t}(\mathbf{w}^t)$

    $$\mathbf{w}^{t+1} = \mathbf{w}^t - \frac{\eta^t}{N} \sum_{i=1}^{N} g_i^t$$

    where

    $$g_i^t = \begin{cases} \nabla f_{i_t}(\mathbf{w}^t) & \text{if } i_t \text{ is selected} \\ g_i^{t-1} & \textit{otherwise} \end{cases}$$

  - **Memory**: $g^t = \nabla f_{i_t}(\mathbf{w}^t)$ from the last t where i_t was selected
    - Keep in memory the gradients of all functions f_i
    - Extra memory requirement: same size as original data

Convergence Rate of SAG

- If each f_i' is Lipschitz continuous and f is strongly-convex, with $\eta^t = 1/16L$, SAG has:

$$\mathbb{E}[f(\mathbf{w}^t) - f(\mathbf{w}^*)] \leq \left(1 - \min\left\{\frac{\mu}{16L}, \frac{1}{8N}\right\}\right)^t C$$

  where

$$C = [f(\mathbf{w}^0) - f(\mathbf{w}^*)] + \frac{4L}{N}\|\mathbf{w}^0 - \mathbf{w}^*\|^2 + \frac{\sigma^2}{16L}$$

- Linear convergence rate but only 1 gradient per iteration
  - For well-conditioned problems, constant reduction per pass:

$$\left(1 - \frac{1}{8N}\right)^N \leq \exp\left(-\frac{1}{8}\right) = 0.8825$$

  - For ill-conditioned problems, almost same as deterministic method (but N times faster).

Rate of Convergence Comparison

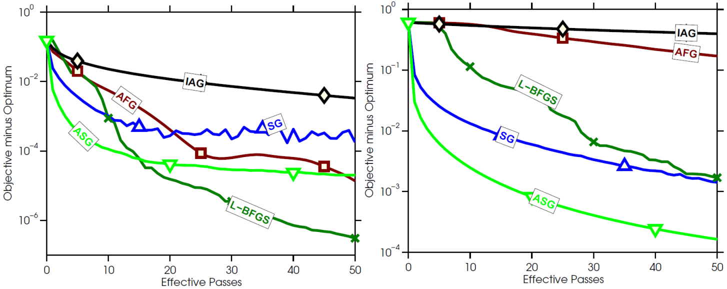
(Assuming Strongly-convex)

- Assume that $N = 700,000$, $L = 0.25$, $\mu = 1/N$:
  - Gradient method has rate $\left(\frac{L+\mu}{L-\mu}\right) = 0.99999$
  - Accelerated gradient method has rate $\left(1 - \sqrt{\frac{\mu}{L}}\right) = 0.99761$
  - SAG (N iterations) has rate $\left(1 - \min\left\{\frac{\mu}{16L}, \frac{1}{8N}\right\}\right)^N = 0.88250$
  - Fastest possible first-order method: $\left(1 - \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2 = 0.99048$
- SAG beats two lower bounds:
  - Stochastic gradient bound (of $O(1/T)$)
  - Deterministic gradient bound (for typical L, μ, and N)
- Number of f_i evaluations to reach ϵ: (iteration complexity)

Stochastic	$O(\frac{L}{\mu}(1/\epsilon))$
Gradient	$O(N\frac{L}{\mu}\log(1/\epsilon))$
Accelerated	$O(N\sqrt{\frac{L}{\mu}}\log(1/\epsilon))$
SAG	$O(\max\left\{N, \frac{L}{\mu}\right\}\log(1/\epsilon))$

Comparing Deterministic and Stochatic Methods

quantum ($N = 50{,}000$, $d = 78$) and rcv1 ($N = 697{,}641$, $d = 47{,}236$)



ASG: The average of the iterations generated by the SG method

AFG: Accelerated Full Gradient

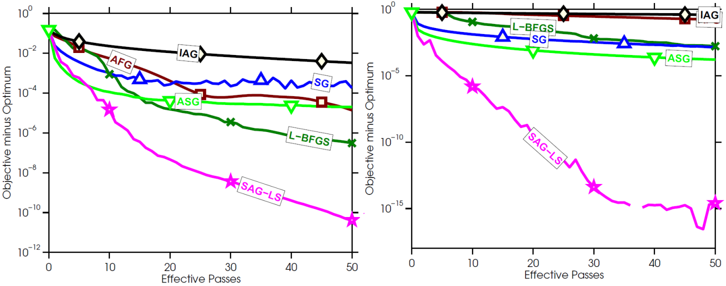
IAG: increment average gradient (Blatt et al. (2007))

SAG-LS: SAG with line search for step sizes

More results: `https://hal.inria.fr/hal-00674995v3/document`

Comparing Deterministic and Stochatic Methods

quantum ($N = 50{,}000$, $d = 78$) and rcv1 ($N = 697{,}641$, $d = 47{,}236$)



ASG: The average of the iterations generated by the SG method

AFG: Accelerated Full Gradient

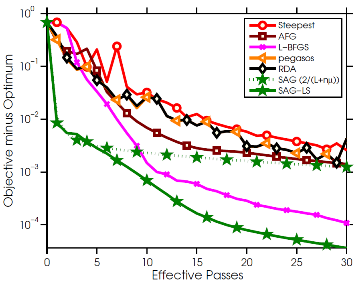
IAG: increment average gradient (Blatt et al. (2007))

SAG-LS: SAG with line search for step sizes

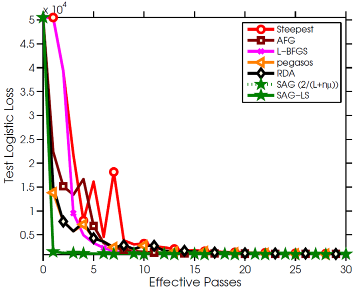
More results: `https://hal.inria.fr/hal-00674995v3/document`

Comparing Deterministic and Stochastic Methods

protein dataset ($N = 145{,}751$, $d = 74$), dataset split in two (training/testing)



Training cost



Testing cost

ASG: The average of the iterations generated by the SG method

AFG: Accelerated Full Gradient

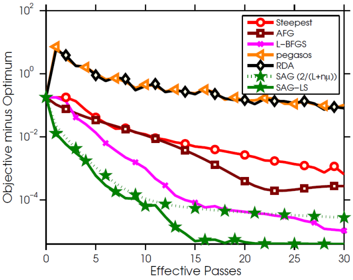
RDA: Dual Averaging for Regularized Stochastic Learning (Xiao (2010))

pegasos: SGD for SVM (Shalev-Shwartz et al. (2011))

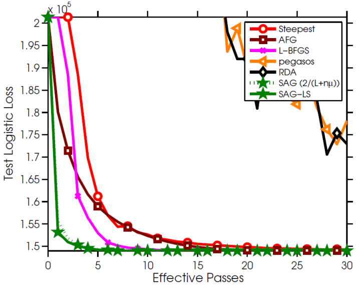
More results: `https://hal.inria.fr/hal-00674995v3/document`

Comparing Deterministic and Stochastic Methods

cover type dataset ($N = 581{,}012$, $d = 54$), dataset split in two (training/testing)



Training cost



Testing cost

ASG: The average of the iterations generated by the SG method

AFG: Accelerated Full Gradient

RDA: Dual Averaging for Regularized Stochastic Learning (Xiao (2010))

pegasos: SGD for SVM (Shalev-Shwartz et al. (2011))

More results: `https://hal.inria.fr/hal-00674995v3/document`

Minimizing Finite Sums: Dealing with the Memory

- A major disadvantage of SAG is the memory requirement
    - Use mini-batches (only store gradient of the mini-batch)
    - Use structure in the objective, e.g.,
        - For $f_i(\mathbf{w}) = L(\mathbf{x}_i^\top \mathbf{w})$, only need to store N values of $\mathbf{x}_i^\top \mathbf{w}$
    - If the above don't work, use stochastic variance-reduced gradient (SVRG)... (Johnson and Zhang (2013); Mahdavi et al. (2013))

Stochastic Variance-Reduced Gradient (SVRG)

- For $s = 0, 1, \ldots,$ do
  - (Maintain an estimate $\tilde{\mathbf{w}}_s$ that is close to the optimal $\mathbf{w}^*$)
  - Compute the gradient at $\tilde{\mathbf{w}}_s$: $\mathbf{d}_s = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\tilde{\mathbf{w}}_s)$
  - Let $\mathbf{w}^0 = \tilde{\mathbf{w}}_s$
  - For $t = 1, \ldots, m$ (e.g., choose $m = 2N$ for convex problem and $m = 5N$ in non-convex problems), do
    - Randomly select i_t from $\{1, 2, \ldots, N\}$

$$\mathbf{w}^t = \mathbf{w}^t - \eta^t(\nabla f_{i_t}(\mathbf{w}^{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{w}}_s) + \mathbf{d}_s)$$

  - Option 1: set $\tilde{\mathbf{w}}_s = \bar{\mathbf{w}}$
  - Option 2: set $\tilde{\mathbf{w}}_s = \mathbf{w}^t$ for randomly chosen t from $t = 1, \ldots, m$

- Requires 2 gradients per iteration but only requires storing $\mathbf{d}_s$ and $\tilde{\mathbf{w}}_s$
- $\mathbb{E}[\nabla f_{i_t}(\mathbf{w}^{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{w}}_s) + \mathbf{d}_s)] = \frac{1}{N}\nabla f_i(\mathbf{w}^{t-1}) = \nabla f(\mathbf{w}^{t-1})$
  - When $\tilde{\mathbf{w}}_s$ and $\mathbf{w}^{t-1}$ converged to the same parameter $\mathbf{w}^*$, then $\mathbf{d}_s \to 0$
  - Therefore if $\nabla f_{i_t}(\tilde{\mathbf{w}}_s) \to \nabla f_{i_t}(\mathbf{w}^*)$, then $\nabla_{i_t} f(\mathbf{w}^{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{w}}_s) + \mathbf{d}_s \to \nabla_{i_t} f(\mathbf{w}^{t-1}) - \nabla f_{i_t}(\mathbf{w}^*) \to 0$
  - Unlike SGD, the learning rate η^t for SVRG does not have to decay, which leads to faster convergence as one can use a relatively large learning rate.

References I

Bach, F. R. and Moulines, E. (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *NIPS*, pages 451–459.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, 1st edition.

Blatt, D., III, A. O. H., and Gauchman, H. (2007). A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51.

Cauchy, A. (1847). Méthode générale pour la résolution des systémes d'équations simultanées. *C. R. Acad. Sci. Paris*, 25:536–538.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323.

Mahdavi, M., Zhang, L., and Jin, R. (2013). Mixed optimization for smooth functions. In *NIPS*, pages 674–682.

Nesterov, Y. (2007). Gradient methods for minimizing composite objective function. Technical report, Center for operations research and econometrics (CORE), Catholic University of Louvein (UCL).

Rakhlin, A., Shamir, O., and Sridharan, K. (2012). Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*.

References II

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Roux, N. L., Schmidt, M. W., and Bach, F. R. (2012). A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, pages 2672–2680.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2011). Pegasos: primal estimated sub-gradient solver for SVM. *Math. Program.*, 127(1):3–30.

Shamir, O. and Zhang, T. (2013). Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML*, pages 71–79.

Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596.