Line-search second-order methods for optimization in noisy environments

Marco Viola

Department of Mathematics and Physics
University of Campania "L. Vanvitelli"
marco.viola@unicampania.it

Joint work with

Daniela di Serafino – University of Naples Federico II
Nataša Krejić, Nataša Krklec Jerinkić – University of Novi Sad

BOS/SOR2020 Conference
Palais Staszic, Warsaw

December 15, 2020

Università
degli Studi
della Campania
Luigi Vanvitelli

Outline

Outline

The problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\ \phi(\boldsymbol{x})$$

$\phi(\boldsymbol{x})$ twice continuously differentiable function in a noisy environment, i.e. $\phi(\boldsymbol{x})$, $\nabla\phi(\boldsymbol{x})$ and $\nabla^2\phi(\boldsymbol{x})$ are only accessible with some level of noise:

$$\begin{aligned} f(\boldsymbol{x}) &= \phi(\boldsymbol{x}) + \varepsilon_f(\boldsymbol{x}) \\ \boldsymbol{g}(\boldsymbol{x}) &= \nabla\phi(\boldsymbol{x}) + \boldsymbol{\varepsilon}_g(\boldsymbol{x}) \\ B(\boldsymbol{x}) &= \nabla^2\phi(\boldsymbol{x}) + \varepsilon_B(\boldsymbol{x}) \end{aligned}$$

$\varepsilon_f(\boldsymbol{x})$ random number, $\boldsymbol{\varepsilon}_g(\boldsymbol{x})$ random vector, $\varepsilon_B(\boldsymbol{x})$ symmetric random matrix

The problem (cont'd)

The error may derive from:

- uncertainty on data;
- measurement errors;
- communication errors;
- computational inaccuracy (data come from a simulation);
- ...

The problem (cont'd)

The error may derive from:

- uncertainty on data;
- measurement errors;
- communication errors;
- computational inaccuracy (data come from a simulation);
- ...

Special cases:

- mathematical expectation:

$$\phi(\boldsymbol{x}) = E_{\xi \sim \mathcal{D}}\left[v(\boldsymbol{x}, \xi)\right], \qquad \text{and} \qquad f(\boldsymbol{x}) = v(\boldsymbol{x}, \bar{\xi}), \text{ with } \bar{\xi} \sim \mathcal{D}$$

The problem (cont'd)

The error may derive from:

- uncertainty on data;
- measurement errors;
- communication errors;
- computational inaccuracy (data come from a simulation);
- ...

Special cases:

- mathematical expectation:

$$\phi(\boldsymbol{x}) = E_{\xi\sim\mathcal{D}}\left[v(\boldsymbol{x},\xi)\right], \quad \text{and} \quad f(\boldsymbol{x}) = v(\boldsymbol{x},\bar{\xi}), \text{ with } \bar{\xi}\sim\mathcal{D}$$

- (large) finite sum of functions:

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{N}\phi_i(\boldsymbol{x}), \quad \text{and} \quad f(\boldsymbol{x}) = \sum_{i\in\mathcal{S}}\phi_i(\boldsymbol{x}), \text{ with } \mathcal{S}\subseteq\{1,\dots,N\}$$

Stochastic optimization methods

First-order methods (NON-exhaustive list)

- Stochastic Approximation - SA (Stochastic Gradient - SG)
  [Robbins & Monro, Ann. Math. Statistics 1951] (convergence in probability with harmonic-type step length, also almost sure (a.s.) convergence with SA variants)

- In the "realm" of machine learning:
  - minibatch gradient methods, see e.g. [Bottou, Curtis & Nocedal, SIREV 2018] (convergence in expectation of obj fun error with constant or harmonic-type step length)
  - variance-reduction gradient methods, e.g. SVRG [Johnson & Zhang, NIPS 2013], SAGA [Defazio, Bach & Lacoste-Julien, NIPS 2014], JacSketch [Gower, Richtárik & Bach, Math Prog 2020]
    (linear convergence in expectation with constant step length)

Stochastic optimization methods (cont'd)

Methods using second-order info (NON-exhaustive list)

- Stochastic versions of Newton-type methods
  - Ruppert, Ann Statist 1985
  - Spall, Proc various IEEE Conferences 1994, 1995, 1005
  - Byrd, Chin, Neveitt & Nocedal, SIOPT 2011
  - Byrd, Chin, Nocedal & Wu, Math Program 2012
  - Bellavia, Krejić & Krklec Jerinkić, IMA JNA 2019
  - Bollapragada, Byrd & Nocedal, IMA JNA 2019
- Stochastic BFGS
  - Byrd, Chin, Neveitt & Nocedal, SIOPT 2011
  - Moktari & Ribeiro, IEEE TSP 2014
  - Byrd, Hansen, Nocedal & Singer, SIOPT 2016
  - Gower, Goldfarb & Richtárik, Proc ICML 2016
  - Moritz, Nishihara & Jordan, Proc MLR 2016

Our family of methods: LSOS

- **L**ine-search **S**econd-**O**rder **S**tochastic algorithmic framework, where Newton-type and quasi-Newton directions are used
- Almost sure convergence of the sequence of iterates generated by the methods fitting into the LSOS framework and effectiveness in practice
- For finite-sum objective functions (e.g. in machine learning)
  - stochastic L-BFGS for Hessian estimates + SAGA-type for gradient estimates + line search
  - almost sure convergence of the sequence of iterates (for state-of-the-art stochastic L-BFGS convergence in expectation of the obj function error)
  - linear convergence rate and worst-case $\mathcal{O}(log(\varepsilon^{-1}))$ complexity
  - practical efficiency (comparison with state-of-the-art stochastic optimization methods)

Outline

SOS: Second-Order Stochastic method

Sketch of SOS method

1: given $\boldsymbol{x}_0 \in \mathbb{R}^n$ and $\{\alpha_k\} \subset \mathbb{R}_+$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: compute $\boldsymbol{d}_k \in \mathbb{R}^n$
4: set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
5: **end for**

$\boldsymbol{d}_k$ specified later

SOS: Second-Order Stochastic method

Sketch of SOS method

1: given $\boldsymbol{x}_0 \in \mathbb{R}^n$ and $\{\alpha_k\} \subset \mathbb{R}_+$
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: compute $\boldsymbol{d}_k \in \mathbb{R}^n$
4: set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
5: **end for**

$\boldsymbol{d}_k$ specified later

Basic assumptions

1. ϕ strongly convex with Lipschitz-continuous gradient:
   - $\boldsymbol{x}_*$ unique solution
   - $\mu I \preceq \nabla^2 \phi(\boldsymbol{x}) \preceq LI$
2. Harmonic step-length sequence:
   $\alpha_k > 0, \;\; \sum_k \alpha_k = \infty, \;\; \sum_k \alpha_k^2 < \infty$
3. Unbiased gradient estimator and bounded variance of gradient errors:
   $\mathbb{E}(\boldsymbol{\varepsilon}_g(\boldsymbol{x}) | \mathcal{F}_k) = 0$ and $\mathbb{E}(\|\boldsymbol{\varepsilon}_g(\boldsymbol{x})\|^2 | \mathcal{F}_k) \leq M$
   ($\mathcal{F}_k = \sigma$-algebra generated by $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$)

Basic assumptions on the search directions

Deterministic case: $c_i > 0$ constants

3. "Sufficient" descent direction:

$$\nabla\phi(\boldsymbol{x}_k)^\top \boldsymbol{d}_k \leq -c_2 \left\|\nabla\phi(\boldsymbol{x}_k)\right\|^2$$

4. Direction norm bounded by gradient:

$$\|\boldsymbol{d}_k\| \leq c_3 \left\|\nabla\phi(\boldsymbol{x}_k)\right\|$$

Basic assumptions on the search directions

Stochastic case: $c_i > 0$ constants

3. Deviation from descent direction allowed:

$$\nabla\phi(\boldsymbol{x}_k)^\top \mathbb{E}\left(\boldsymbol{d}_k|\mathcal{F}_k\right) \le c_1\delta_k - c_2\left\|\nabla\phi(\boldsymbol{x}_k)\right\|^2, \quad \delta_k > 0, \quad \sum_k \alpha_k\delta_k < \infty$$

4. Direction norm bounded by noisy gradient:

$$\|\boldsymbol{d}_k\| \le c_3\left\|\boldsymbol{g}(\boldsymbol{x}_k)\right\| \quad \text{a.s.}$$

Basic assumptions on the search directions

Stochastic case: $c_i > 0$ constants

3. Deviation from descent direction allowed:

$$\nabla \phi(\boldsymbol{x}_k)^\top \mathbb{E}\left(\boldsymbol{d}_k | \mathcal{F}_k\right) \leq c_1 \delta_k - c_2 \left\|\nabla \phi(\boldsymbol{x}_k)\right\|^2, \quad \delta_k > 0, \quad \sum_k \alpha_k \delta_k < \infty$$

4. Direction norm bounded by noisy gradient:

$$\|\boldsymbol{d}_k\| \leq c_3 \|\boldsymbol{g}(\boldsymbol{x}_k)\| \quad \text{a.s.}$$

Theorem

Under the previous assumptions, the sequence $\{\boldsymbol{x}_k\}$ *converges to* $\boldsymbol{x}_*$ *a.s.*

Search directions using second-order information

Further (reasonable) assumptions

6. Positive definite and bounded approximate Hessians: $\mu I \preceq B(\boldsymbol{x}) \preceq LI$

7. Mutually independent noise terms $\varepsilon_f(\boldsymbol{x}), \boldsymbol{\varepsilon}_g(\boldsymbol{x})$ and $\varepsilon_B(\boldsymbol{x})$ (to be relaxed for finite-sum problems)

Search directions using second-order information

Further (reasonable) assumptions

6. Positive definite and bounded approximate Hessians: $\mu I \preceq B(\boldsymbol{x}) \preceq LI$
7. Mutually independent noise terms $\varepsilon_f(\boldsymbol{x}), \boldsymbol{\varepsilon}_g(\boldsymbol{x})$ and $\varepsilon_B(\boldsymbol{x})$ (to be relaxed for finite-sum problems)

Possible directions guaranteeing convergence:

- Newton directions:

$$B(\boldsymbol{x}_k)\boldsymbol{d}_k = -\boldsymbol{g}(\boldsymbol{x}_k)$$

- "Inexact" Newton directions:

$$\|B(\boldsymbol{x}_k)\boldsymbol{d}_k + \boldsymbol{g}(\boldsymbol{x}_k)\| \leq \delta_k \gamma_k$$

γ_k random variable with bounded variance

Search directions using second-order information

Further (reasonable) assumptions

6. Positive definite and bounded approximate Hessians: $\mu I \preceq B(\boldsymbol{x}) \preceq LI$
7. Mutually independent noise terms $\varepsilon_f(\boldsymbol{x}), \boldsymbol{\varepsilon}_g(\boldsymbol{x})$ and $\varepsilon_B(\boldsymbol{x})$ (to be relaxed for finite-sum problems)

Possible directions guaranteeing convergence:

- Newton directions:

$$B(\boldsymbol{x}_k)\boldsymbol{d}_k = -\boldsymbol{g}(\boldsymbol{x}_k)$$

- "Inexact" Newton directions:

$$\|B(\boldsymbol{x}_k)\boldsymbol{d}_k + \boldsymbol{g}(\boldsymbol{x}_k)\| \leq \delta_k(\omega_1\eta_k + \omega_2\|\boldsymbol{g}(\boldsymbol{x}_k)\|)$$

$\omega_1, \omega_2 \geq 0$ constant, η_k random variable with bounded variance

LSOS: Line-search SOS

- A harmonic step-length sequence $(\sum_k \alpha_k = \infty, \ \sum_k \alpha_k^2 < \infty)$ may make the algorithm slow (the steplength becomes too small soon)

- Tuning is necessary to ensure reasonable results; if the steplengths are not small enough the algorithm may diverge

LSOS: Line-search SOS

- A harmonic step-length sequence ($\sum_k \alpha_k = \infty, \;\; \sum_k \alpha_k^2 < \infty$) may make the algorithm slow (the steplength becomes too small soon)

- Tuning is necessary to ensure reasonable results; if the steplengths are not small enough the algorithm may diverge

IDEA: start with line search and move to harmonic step lengths only if the line search produces small step lengths

LSOS: Line-search SOS

- A harmonic step-length sequence ($\sum_k \alpha_k = \infty, \ \sum_k \alpha_k^2 < \infty$) may make the algorithm slow (the steplength becomes too small soon)
- Tuning is necessary to ensure reasonable results; if the steplengths are not small enough the algorithm may diverge

IDEA: start with line search and move to harmonic step lengths only if the line search produces small step lengths

- At each step the direction is not guaranteed to be a descent direction for $\phi(\boldsymbol{x})$

IDEA: use nonmonotone line search

LSOS: Line-search SOS (cont'd)

LSOS algorithm

given $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\eta \in (0,1)$, $t_{\min} > 0$ and $\{\alpha_k\}, \{\delta_k\}, \{\zeta_k\} \subset \mathbb{R}_+$
set LSphase = *active*
for $k = 0, 1, 2, \ldots$ **do**
    compute a search direction $\boldsymbol{d}_k$ such that

$$\|B(\boldsymbol{x}_k)\boldsymbol{d}_k + \boldsymbol{g}(\boldsymbol{x}_k)\| \leq \delta_k \|\boldsymbol{g}(\boldsymbol{x}_k)\|$$

end for

LSOS: Line-search SOS (cont'd)

LSOS algorithm

given $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\eta \in (0,1)$, $t_{\min} > 0$ and $\{\alpha_k\}, \{\delta_k\}, \{\zeta_k\} \subset \mathbb{R}_+$
set LSphase = *active*
for $k = 0, 1, 2, \dots$ **do**
    compute a search direction $\boldsymbol{d}_k$ such that

$$\|B(\boldsymbol{x}_k)\boldsymbol{d}_k + \boldsymbol{g}(\boldsymbol{x}_k)\| \le \delta_k \|\boldsymbol{g}(\boldsymbol{x}_k)\|$$

    find a step length t_k as follows:
        if LSphase = *active* **then** find t_k that satisfies

$$f(\boldsymbol{x}_k + t_k \boldsymbol{d}_k) \le f(\boldsymbol{x}_k) + \eta t_k \boldsymbol{g}(\boldsymbol{x}_k)^\top \boldsymbol{d}_k + \zeta_k$$

        if $t_k < t_{\min}$ **then** set LSphase = *inactive*
        if LSphase = *inactive* **then** set $t_k = \alpha_k$
    set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t_k \boldsymbol{d}_k$
end for

LSOS convergence

Theorem

Assume that $\{\zeta_k\}$ is summable and the objective function estimator f is unbiased, i.e.

$$\mathbb{E}(\varepsilon_f(\boldsymbol{x})|\mathcal{F}_k) = 0.$$

If the sequence $\{\boldsymbol{x}_k\}$ generated by LSOS is bounded, then $\boldsymbol{x}_k \to \boldsymbol{x}_*$ a.s..

Outline

Convex random problems (type 1)

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \left(e^{x_i} - x_i\right) + (\boldsymbol{x} - \boldsymbol{1})^\top A(\boldsymbol{x} - \boldsymbol{1})$$

Convex random problems (type 1)

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \left(e^{x_i} - x_i\right) + (\boldsymbol{x} - \mathbf{1})^\top A(\boldsymbol{x} - \mathbf{1})$$

- λ_i's logarithmically spaced between 1 and κ
- $A \in \mathbb{R}^{n \times n}$ spd with eigenvalues λ_i (generated by `sprandsym`)
- $n = 10^3$, $\kappa = 10^2, 10^3, 10^4$
- $\varepsilon_f(\boldsymbol{x}) \sim \mathcal{N}(0, \sigma)$, $(\varepsilon_g(\boldsymbol{x}))_i \sim \mathcal{N}(0, \sigma)$ and
  $\varepsilon_B(\boldsymbol{x}) = \mathrm{diag}\,(\mu_1, \ldots, \mu_n)$, $\quad \mu_i \sim \mathcal{N}(0, \sigma)$
- $\sigma = 0.1\%\,\kappa, 0.5\%\,\kappa, 1\%\,\kappa$
- x_* computed with high accuracy using deterministic L-BFGS
  (M. Schmidt, `https://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`)

Convex random problems (type 1)

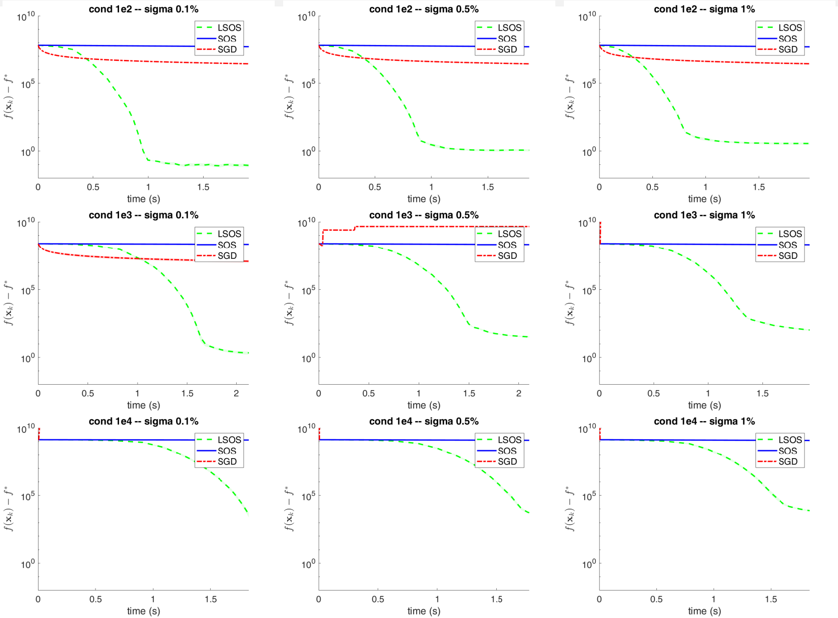
$$\phi(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \left(e^{x_i} - x_i\right) + (\boldsymbol{x} - \mathbf{1})^\top A(\boldsymbol{x} - \mathbf{1})$$

- λ_i's logarithmically spaced between 1 and κ
- $A \in \mathbb{R}^{n \times n}$ spd with eigenvalues λ_i (generated by `sprandsym`)
- $n = 10^3$, $\kappa = 10^2, 10^3, 10^4$
- $\varepsilon_f(\boldsymbol{x}) \sim \mathcal{N}(0,\sigma)$, $(\varepsilon_g(\boldsymbol{x}))_i \sim \mathcal{N}(0,\sigma)$ and
  $\varepsilon_B(\boldsymbol{x}) = \mathrm{diag}\,(\mu_1, \ldots, \mu_n), \quad \mu_i \sim \mathcal{N}(0,\sigma)$
- $\sigma = 0.1\%\,\kappa, 0.5\%\,\kappa, 1\%\,\kappa$
- x_* computed with high accuracy using deterministic L-BFGS
  (M. Schmidt, `https://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`)

Comparison of

- LSOS with exact solution of noisy Newton systems
- SOS with pre-defined step length $\alpha_k = \dfrac{1}{\|\boldsymbol{d}_0\|}\dfrac{T}{T+k}, \quad T = 10^6$
- Stochastic Gradient Descent (SGD) with step length α_k

Convex random problems (type 1): obj fun error vs time



Convex random problems (type 2)

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \left(e^{x_i} - x_i\right) + (\boldsymbol{x} - \mathbf{1})^\top A(\boldsymbol{x} - \mathbf{1})$$

$$A = V\,D\,V^T, \;\; D = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \;\; V = (I - 2\,\boldsymbol{v}_3\boldsymbol{v}_3^T)(I - 2\,\boldsymbol{v}_2\boldsymbol{v}_2^T)(I - 2\,\boldsymbol{v}_1\boldsymbol{v}_1^T),$$

$$\boldsymbol{v}_j \text{ random}, \;\; \|\boldsymbol{v}_j\| = 1$$

- $n = 2 \cdot 10^4$, $\kappa = 10^2, 10^3, 10^4$
- $\sigma = 0.1\%\,\kappa, 0.5\%\,\kappa, 1\%\,\kappa$
- Hessian in factorized form $\Longrightarrow$ (noisy) Newton system must be solved inexactly (e.g., by CG)

Convex random problems (type 2)

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \left(e^{x_i} - x_i\right) + (\boldsymbol{x} - \mathbf{1})^\top A(\boldsymbol{x} - \mathbf{1})$$

$$A = V\,D\,V^T, \;\; D = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \;\; V = (I - 2\,\boldsymbol{v}_3\boldsymbol{v}_3^T)(I - 2\,\boldsymbol{v}_2\boldsymbol{v}_2^T)(I - 2\,\boldsymbol{v}_1\boldsymbol{v}_1^T),$$

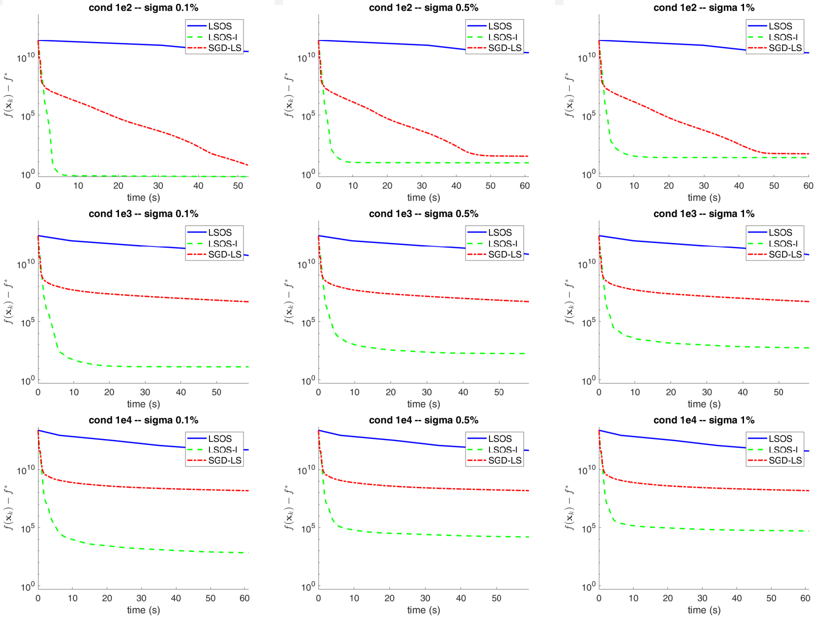
$$\boldsymbol{v}_j \text{ random}, \;\; \|\boldsymbol{v}_j\| = 1$$

- $n = 2 \cdot 10^4$, $\kappa = 10^2, 10^3, 10^4$
- $\sigma = 0.1\%\,\kappa, 0.5\%\,\kappa, 1\%\,\kappa$
- Hessian in factorized form $\Longrightarrow$ (noisy) Newton system must be solved inexactly (e.g., by CG)

Comparison of

- LSOS (“exact” solution of noisy Newton systems - CG tolerance 1e-6)
- LSOS-I (inexact solution of noisy Newton systems - decreasing tolerance sequence)
- SGD-LS (SGD with line search)

Convex random problems (type 2): obj fun error vs time



Outline

The finite sum case

$$\phi(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\phi_i(\boldsymbol{x})$$

$\phi_i(\boldsymbol{x}) \in \mathcal{C}^2$ $\overline{\mu}$-strongly convex, with Lipschitz-continuous gradient with constant $\overline{L}$

The finite sum case

$$\phi(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\phi_i(\boldsymbol{x})$$

$\phi_i(\boldsymbol{x}) \in \mathcal{C}^2$ $\overline{\mu}$-strongly convex, with Lipschitz-continuous gradient with constant $\overline{L}$

Subsampling: at each iter k, a sample $\mathcal{N}_k$ of size $N_k \ll N$ is chosen randomly and uniformly from $\mathcal{N} = \{1, ..., N\}$:

$$f_{\mathcal{N}_k}(\boldsymbol{x}) = \frac{1}{N_k}\sum_{i\in\mathcal{N}_k}\phi_i(\boldsymbol{x}), \quad \boldsymbol{g}_{\mathcal{N}_k}(\boldsymbol{x}) = \frac{1}{N_k}\sum_{i\in\mathcal{N}_k}\nabla\phi_i(\boldsymbol{x}),$$

$$B_{\mathcal{N}_k}(\boldsymbol{x}) = \frac{1}{N_k}\sum_{i\in\mathcal{N}_k}\nabla^2\phi_i(\boldsymbol{x})$$

(unbiased estimators of $\phi(\boldsymbol{x})$, $\nabla\phi(\boldsymbol{x})$ and $\nabla^2\phi(\boldsymbol{x})$)

Stochastic variant of L-BFGS

Hessian approximation from stochastic variant of Limited-memory BFGS (L-BFGS) [Byrd, Hansen, Nocedal & Singer, SIOPT 2016]

H_k defined by applying m BFGS updates to an initial matrix, using the m most recent correction pairs $(\boldsymbol{s}_j, \boldsymbol{y}_j)$ obtained averaging iterates over r steps ($j = k/r$):

$$H_k = H_k^{(m)}, \quad \text{where} \quad H_k^{(0)} = \frac{\boldsymbol{s}_m^\top \boldsymbol{y}_m}{\|\boldsymbol{y}_m\|^2} I$$

$$H_k^{(j)} = \left(I - \frac{\boldsymbol{s}_j\, \boldsymbol{y}_j^\top}{\boldsymbol{s}_j^\top \boldsymbol{y}_j}\right)^\top H_k^{(j-1)} \left(I - \frac{\boldsymbol{y}_j\, \boldsymbol{s}_j^\top}{\boldsymbol{s}_j^\top \boldsymbol{y}_j}\right) + \frac{\boldsymbol{s}_j\, \boldsymbol{s}_j^\top}{\boldsymbol{s}_j^\top \boldsymbol{y}_j}, \quad j = 1, \ldots, m$$

$$\boldsymbol{s}_j = \boldsymbol{w}_j - \boldsymbol{w}_{j-1}, \quad \boldsymbol{y}_j = B_{\mathcal{T}_j}(\boldsymbol{w}_j)\, \boldsymbol{s}_j, \quad \mathcal{T}_j \subset \{1, \ldots, N\}$$

$$\boldsymbol{w}_j = \frac{1}{r} \sum_{i=k-r+1}^{k} \boldsymbol{x}_i, \quad \boldsymbol{w}_{j-1} = \frac{1}{r} \sum_{i=k-2r+1}^{k-r} \boldsymbol{x}_i$$

Mini-batch SAGA

Subsampled gradient estimate by a a mini-batch variant of SAGA
[Defazio, Bach & Lacoste-Julien, NIPS 2014; Gower, Richtárik & Bach, Math Prog 2020]

$$\boldsymbol{g}_{\mathcal{N}_k}^{\mathrm{SAGA}}(\boldsymbol{x}_k) = \frac{1}{N_k}\sum_{i\in\mathcal{N}_k}\left(\nabla\phi_i(\boldsymbol{x}_k) - J_k^{(i)}\right) + \frac{1}{N}\sum_{r=1}^{N} J_k^{(r)}$$

$$J_{k+1}^{(i)} = \begin{cases} J_k^{(i)} & \text{if } i \notin \mathcal{N}_k \\ \nabla\phi_i(\boldsymbol{x}_{k+1}) & \text{if } i \in \mathcal{N}_k \end{cases}, \qquad J_0^{(i)} = \nabla\phi_i(\boldsymbol{x}_0)$$

$\{1,\ldots,N\}$ partitioned into a fixed number n_b of random mini-batches, which are used in order

Advantage of SAGA over SVRG: full gradient computation only at the beginning of the algorithm (SVRG: full gradient computation each n_b iterations)

LSOS-BFGS: Finite-Sum LSOS with L-BFGS

LSOS-BFGS

given $\boldsymbol{x}_0 \in \mathbb{R}^n,\ \ m, r \in \mathbb{N},\ \ \eta, \vartheta \in (0,1)$
for $k = 0, 1, 2, \ldots$ **do**
    compute a partition $\{\mathcal{K}_0, \mathcal{K}_1, \ldots, \mathcal{K}_{n_b-1}\}$ of $\{1, \ldots, N\}$
end for

LSOS-BFGS: Finite-Sum LSOS with L-BFGS

LSOS-BFGS

```
given $x_0 \in \mathbb{R}^n$,  $m, r \in \mathbb{N}$,  $\eta, \vartheta \in (0,1)$
for $k = 0, 1, 2, \dots$ do
    compute a partition $\{\mathcal{K}_0, \mathcal{K}_1, \dots, \mathcal{K}_{n_b-1}\}$ of $\{1, \dots, N\}$
    for $s = 0, \dots, n_b - 1$ do
        choose $\mathcal{N}_k = \mathcal{K}_s$ and compute $g(x_k) = g_{\mathcal{N}_k}^{\text{SAGA}}(x_k)$
        compute $d_k = -H_k\, g(x_k)$ with $H_k$ defined by stochastic L-BFGS

end for
```

LSOS-BFGS: Finite-Sum LSOS with L-BFGS

LSOS-BFGS

given $\boldsymbol{x}_0 \in \mathbb{R}^n$, $m, r \in \mathbb{N}$, $\eta, \vartheta \in (0,1)$
for $k = 0, 1, 2, \dots$ **do**
  compute a partition $\{\mathcal{K}_0, \mathcal{K}_1, \dots, \mathcal{K}_{n_b - 1}\}$ of $\{1, \dots, N\}$
  for $s = 0, \dots, n_b - 1$ **do**
    choose $\mathcal{N}_k = \mathcal{K}_s$ and compute $\boldsymbol{g}(\boldsymbol{x}_k) = \boldsymbol{g}_{\mathcal{N}_k}^{\text{SAGA}}(\boldsymbol{x}_k)$
    compute $\boldsymbol{d}_k = -H_k\, \boldsymbol{g}(\boldsymbol{x}_k)$ with H_k defined by stochastic L-BFGS
    find a step length t_k such that

$$f_{\mathcal{N}_k}(\boldsymbol{x}_k + t_k \boldsymbol{d}_k) \le f_{\mathcal{N}_k}(\boldsymbol{x}_k) + \eta t_k\, \boldsymbol{g}(x_k)^\top \boldsymbol{d}_k + \vartheta^k$$

    set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t_k \boldsymbol{d}_k$;
    if $\mod(k, r) = 0$ and $k \ge 2\,r$ **then**
      update the L-BFGS correction pairs
    end if
  end for
end for

FS-LSOS: convergence

Theorem (convergence)

Assume $\{t_k\}$ *is bounded away from zero. Then* $\{\boldsymbol{x}_k\}$ *converges a.s. to the unique minimizer of* ϕ*.*

FS-LSOS: convergence

Theorem (convergence)

Assume $\{t_k\}$ *is bounded away from zero. Then* $\{\boldsymbol{x}_k\}$ *converges a.s. to the unique minimizer of* ϕ*.*

Theorem (convergence rate)

Let $\{t_k\}$ *be bounded away from zero. Then there exist* $\rho \in (0,1)$ *and* $C > 0$ *such that*

$$\mathbb{E}(\phi(\boldsymbol{x}_k) - \phi(\boldsymbol{x}_*)) \leq C\rho^k.$$

Theorem (complexity bound)

In order to achieve $\mathbb{E}(\phi(\boldsymbol{x}_k) - \phi(\boldsymbol{x}_*)) \leq \varepsilon$ *for some* $\varepsilon \in (0, e^{-1})$*, LSOS-FS takes at most*

$$k_{\max} = \left\lceil \frac{|log(C)| + 1}{|log(\rho)|} log(\varepsilon^{-1}) \right\rceil = \mathcal{O}\left(\log(\varepsilon^{-1})\right)$$

with $\rho \in (0,1)$ *and* $C > 0$*.*

Outline

Linear classification problems

Training a linear classifier by minimizing the ℓ_2-regularized logistic regression

Given N pairs $(\boldsymbol{a}_i, b_i)$, $\boldsymbol{a}_i \in \mathbb{R}^n$ training point, $b_i \in \{-1, 1\}$ corresponding label, a hyperplane approximately separating the two classes can be found by minimizing

$$\phi(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \phi_i(\boldsymbol{x}), \quad \text{with } \phi_i(\boldsymbol{x}) = \log\left(1 + e^{-b_i\, \boldsymbol{a}_i^\top \boldsymbol{x}}\right) + \frac{\mu}{2} \|\boldsymbol{x}\|^2, \ \ \mu > 0$$

Linear classification problems

Training a linear classifier by minimizing the ℓ_2-regularized logistic regression

Given N pairs $(\boldsymbol{a}_i, b_i)$, $\boldsymbol{a}_i \in \mathbb{R}^n$ training point, $b_i \in \{-1, 1\}$ corresponding label, a hyperplane approximately separating the two classes can be found by minimizing

$$\phi(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \phi_i(\boldsymbol{x}), \quad \text{with } \phi_i(\boldsymbol{x}) = \log\left(1 + e^{-b_i\, \boldsymbol{a}_i^\top \boldsymbol{x}}\right) + \frac{\mu}{2}\|\boldsymbol{x}\|^2, \;\; \mu > 0$$

Note that

$$\nabla\phi_i(\boldsymbol{x}) = \frac{1 - z_i(\boldsymbol{x})}{z_i(\boldsymbol{x})} b_i\, \boldsymbol{a}_i + \mu\boldsymbol{x}, \;\; \nabla^2\phi_i(\boldsymbol{x}) = \frac{z_i(\boldsymbol{x}) - 1}{z_i^2(\boldsymbol{x})}\boldsymbol{a}_i\boldsymbol{a}_i^\top + \mu I, \;\; z_i(\boldsymbol{x}) = 1 + e^{-b_i\, \boldsymbol{a}_i^\top}$$

$$\Downarrow$$

$$\phi_i \;\mu\text{-strongly convex}, \quad \mu I \preceq \nabla^2\phi_i(\boldsymbol{x}) \preceq LI, \quad L = \mu + \max_{i=1,\ldots,N}\|a_i\|^2$$

Linear classification problems (cont'd)

LIBSVM datasets (`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/`)

name	N	n
covtype	406709	54
w8a	49749	300
epsilon	400000	2000
gisette	6000	5000
real-sim	50617	20958
rcv1	20242	47236

NOTE: $\mu = 1/N$, sample size $= \lceil \sqrt{N} \rceil$

Linear classification problems (cont'd)

LIBSVM datasets (`https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/`)

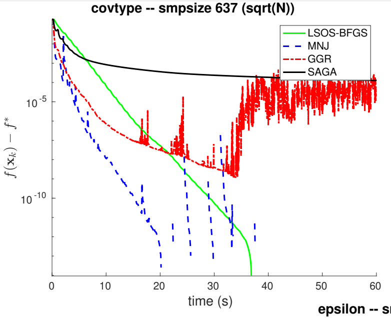
name	N	n
covtype	406709	54
w8a	49749	300
epsilon	400000	2000
gisette	6000	5000
real-sim	50617	20958
rcv1	20242	47236

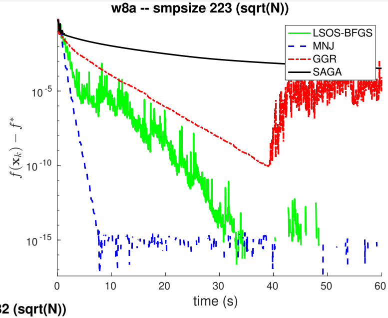
NOTE: $\mu = 1/N$, sample size $= \lceil \sqrt{N} \rceil$

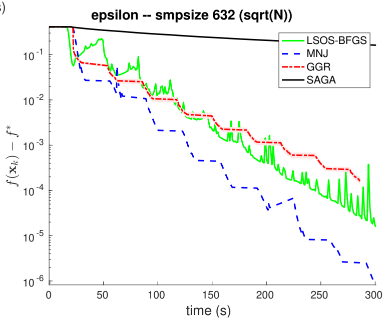
Comparison between

- LSOS-BFGS, with $m = 10$ and $r = 5$
- GGR [Gower, Goldfarb & Richtárik, Proc ICML 2016]
- MNJ [Moritz, Nishihara & Jordan, Proc MLR 2016]
- Mini-batch variant of SAGA, with the same line search as LSOS-BFGS

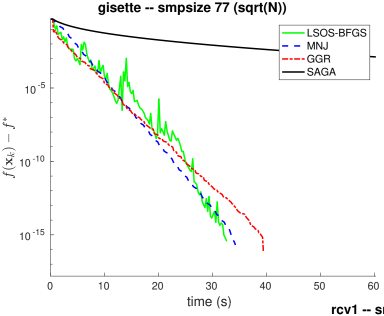
Classification problems: obj fun error vs time

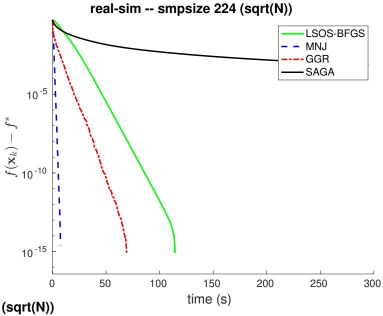


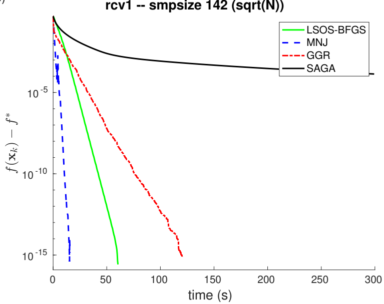




Classification problems: obj fun error vs time







Outline

Conclusions and future work

- We introduced LSOS a flexible second-order framework for optimization in noisy environments
- Almost sure convergence holds for the sequences generated by all the LSOS variants
- For finite-sum problems, we proved linear convergence rate on the obj. fun. error and worst-case complexity bound $\mathcal{O}(\log(\varepsilon^{-1}))$ for LSOS with stochastic L-BFGS Hessian and any Lipschitz-continuous unbiased gradient estimates are used

Conclusions and future work

- We introduced LSOS a flexible second-order framework for optimization in noisy environments
- Almost sure convergence holds for the sequences generated by all the LSOS variants
- For finite-sum problems, we proved linear convergence rate on the obj. fun. error and worst-case complexity bound $\mathcal{O}(\log(\varepsilon^{-1}))$ for LSOS with stochastic L-BFGS Hessian and any Lipschitz-continuous unbiased gradient estimates are used
- Numerical experiments confirm that line-search techniques in second-order stochastic methods yield a significant improvement over predefined step-length sequences
- For finite sum problems LSOS-BFGS highly competitive with state-of-the art second-order stochastic optimization methods

Conclusions and future work

- We introduced LSOS a flexible second-order framework for optimization in noisy environments
- Almost sure convergence holds for the sequences generated by all the LSOS variants
- For finite-sum problems, we proved linear convergence rate on the obj. fun. error and worst-case complexity bound $\mathcal{O}(\log(\varepsilon^{-1}))$ for LSOS with stochastic L-BFGS Hessian and any Lipschitz-continuous unbiased gradient estimates are used
- Numerical experiments confirm that line-search techniques in second-order stochastic methods yield a significant improvement over predefined step-length sequences
- For finite sum problems LSOS-BFGS highly competitive with state-of-the art second-order stochastic optimization methods
- **What's next?** Possible extension to problems not satisfying the strong convexity assumption and to constrained problems

Thanks for the attention!
Any questions?

Do you want to know more?

D. di Serafino, N. Krejić, N. Krklec Jerinkić, M. Viola, *LSOS: Line-search Second-Order Stochastic optimization methods*, submitted (also available on ArXiv and Optimization Online)