An Overview of Stochastic Methods for Solving Optimization Problems

Émilie Chouzenoux

Center for Visual Computing, CentraleSupelec, INRIA Saclay

19 Dec. 17

Introduction

STOCHASTIC PROBLEM

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}}\ \mathbb{E}(\varphi_j(\boldsymbol{h}_j^\top \boldsymbol{x}, y_j)) + g(\boldsymbol{D}\boldsymbol{x})$$

where $j \in \mathbb{N}^*$, $\boldsymbol{h}_j \in \mathbb{R}^N$, $y_j \in \mathbb{R}$, $\varphi_j : \mathbb{R}\times\mathbb{R} \to]-\infty,+\infty]$ is a loss function, and $g \circ \boldsymbol{D}$ is a regularization function, with $g : \mathbb{R}^P \to]-\infty,+\infty]$ and $\boldsymbol{D} \in \mathbb{R}^{P\times N}$.

Introduction

STOCHASTIC PROBLEM

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \mathbb{E}(\varphi_j(\boldsymbol{h}_j^\top \boldsymbol{x}, y_j)) + g(\boldsymbol{D}\boldsymbol{x})$$

where $j \in \mathbb{N}^*$, $\boldsymbol{h}_j \in \mathbb{R}^N$, $y_j \in \mathbb{R}$, $\varphi_j \colon \mathbb{R}\times\mathbb{R} \to]-\infty,+\infty]$ is a loss function, and $g \circ \boldsymbol{D}$ is a regularization function, with $g\colon \mathbb{R}^P \to]-\infty,+\infty]$ and $\boldsymbol{D} \in \mathbb{R}^{P\times N}$.

BATCH PROBLEM

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \frac{1}{M}\sum_{i=1}^{M} \varphi_i(\boldsymbol{h}_i^\top \boldsymbol{x}, y_i) + g(\boldsymbol{D}\boldsymbol{x})$$

where, for all $i \in \{1,\dots,M\}$, $\varphi_i\colon \mathbb{R}\times\mathbb{R} \to]-\infty,+\infty]$, $\boldsymbol{h}_i \in \mathbb{R}^N$ and $y_i \in \mathbb{R}$.

Link between stochastic and batch problems

STOCHASTIC PROBLEM

$j \in \mathbb{N}^*$ is deterministic,

$(\forall i \in \{2, \dots, M\})\ \varphi_i = \varphi_1$,

and $(\boldsymbol{h}_j)_{j \geqslant 1}$, $(y_j)_{j \geqslant 1}$ are

i.i.d random variables.

BATCH PROBLEM

Link between stochastic and batch problems

STOCHASTIC PROBLEM

$\boldsymbol{y}$ and $\boldsymbol{H}$ are deterministic,
and j is uniformly distributed
over $\{1, \ldots, M\}$.

BATCH PROBLEM

Introduction

Numerous examples:

- supervised classification
- inverse problems
- system identification, channel equalization
- linear prediction/interpolation
- echo cancellation, interference removal
- ...

In the context of large scale problems, how to find an optimization algorithm able to deliver a reliable numerical solution in a reasonable time, with low memory requirement?

Outline

* FUNDAMENTAL TOOLS IN CONVEX ANALYSIS

* OPTIMIZATION ALGORITHMS FOR SOLVING STOCHASTIC PROBLEM
  - Stochastic forward-backward algorithm
  - A brief focus on sparse adaptive filtering

* STOCHASTIC ALGORITHMS FOR SOLVING BATCH PROBLEM
  - Incremental gradient algorithms
  - Block coordinate approaches

Fundamental tools in convex analysis

Notation and definitions

Let $f\colon \mathbb{R}^N \to]-\infty,+\infty]$.

- The domain of function f is
$$\operatorname{dom} f = \{\boldsymbol{x} \in \mathbb{R}^N \mid f(\boldsymbol{x}) < +\infty\}$$
  If $\operatorname{dom} f \neq \varnothing$, function f is said to be proper.
- Function f is convex if
$$(\forall(\boldsymbol{x},\boldsymbol{y}) \in (\mathbb{R}^N)^2)(\forall \lambda \in [0,1]) \quad f(\lambda\boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leqslant \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}).$$
- Function f is lower semi-continuous (lsc) on $\mathbb{R}^N$ if, for all $\boldsymbol{x} \in \mathbb{R}^N$, for all sequence $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ of $\mathbb{R}^N$,
$$\boldsymbol{x}_k \longrightarrow \boldsymbol{x} \quad \Rightarrow \quad \liminf f(\boldsymbol{x}_k) \geqslant f(\boldsymbol{x}).$$

Notation and definitions

Let $f\colon \mathbb{R}^N \to]-\infty, +\infty]$. Function f is said ν-strongly convex if

$$(\forall (\boldsymbol{x}, \boldsymbol{y}) \in (\mathbb{R}^N)^2)(\forall \lambda \in [0,1])$$

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leqslant \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}) - \frac{1}{2}\nu\lambda(1-\lambda)\|\boldsymbol{x}-\boldsymbol{y}\|^2,$$

with $\nu \in]0, +\infty[$.

Notation and definitions

Let $f\colon \mathbb{R}^N \to]-\infty, +\infty]$. Function f is said ν-strongly convex if

$$(\forall(\boldsymbol{x},\boldsymbol{y}) \in (\mathbb{R}^N)^2)(\forall \lambda \in [0,1])$$
$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leqslant \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}) - \frac{1}{2}\nu\lambda(1-\lambda)\|\boldsymbol{x}-\boldsymbol{y}\|^2,$$

with $\nu \in]0, +\infty[$.

Let $f\colon \mathbb{R}^N \to]-\infty, +\infty[$. Function f is said β-Lipschitz differentiable if it is differentiable over $\mathbb{R}^N$ and its gradient fulfills

$$(\forall(\boldsymbol{x},\boldsymbol{y}) \in (\mathbb{R}^N)^2) \quad \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leqslant \beta\|\boldsymbol{x}-\boldsymbol{y}\|,$$

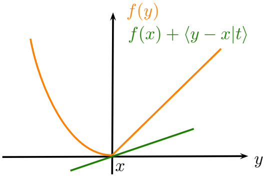
with $\beta \in]0, +\infty[$.

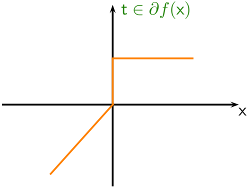
Subdifferential

The subdifferential of a convex function $f\colon \mathbb{R}^N \to \left]-\infty,+\infty\right]$ at $\boldsymbol{x}$ is the set

$$\partial f(\boldsymbol{x}) = \left\{\boldsymbol{t} \in \mathbb{R}^N \mid (\forall \boldsymbol{y} \in \mathbb{R}^N)\;\; f(\boldsymbol{y}) \geqslant f(\boldsymbol{x}) + \langle \boldsymbol{t} \mid \boldsymbol{y} - \boldsymbol{x}\rangle\right\}$$

An element $\boldsymbol{t}$ of $\partial f(\boldsymbol{x})$ is called a subgradient of f at $\boldsymbol{x}$.





- If f is differentiable at $\boldsymbol{x} \in \mathbb{R}^N$ then $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$.

Proximity operator

Let $f : \mathbb{R}^N \mapsto]-\infty, +\infty]$ a proper, convex, l.s.c function.

Characterization of proximity operator

$$(\forall \boldsymbol{x} \in \mathbb{R}^N) \quad \widehat{\boldsymbol{y}} = \operatorname{prox}_f(\boldsymbol{x}) \Leftrightarrow \boldsymbol{x} - \widehat{\boldsymbol{y}} \in \partial f(\widehat{\boldsymbol{y}}).$$

The proximity operator $\operatorname{prox}_f(\boldsymbol{x})$ of f at $\boldsymbol{x} \in \mathbb{R}^N$ is the unique vector $\widehat{\boldsymbol{y}} \in \mathbb{R}^N$ such that

$$f(\widehat{\boldsymbol{y}}) + \frac{1}{2}\|\widehat{\boldsymbol{y}} - \boldsymbol{x}\|^2 = \inf_{\boldsymbol{y} \in \mathbb{R}^N} f(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|.$$

Properties of proximal operator

	$f(\boldsymbol{x})$	$\operatorname{prox}_f(\boldsymbol{x})$
translation $\boldsymbol{z} \in \mathbb{R}^N$	$f(\boldsymbol{x}-\boldsymbol{z})$	$\boldsymbol{z} + \operatorname{prox}_f(\boldsymbol{x}-\boldsymbol{z})$
quadratic perturbation $\boldsymbol{z} \in \mathbb{R}^N$, $\alpha > 0$, $\gamma \in \mathbb{R}$	$f(\boldsymbol{x}) + \alpha\\|\boldsymbol{x}\\|^2/2 + \langle \boldsymbol{x} \mid \boldsymbol{z}\rangle + \gamma$	$\operatorname{prox}_{\frac{f}{\alpha+1}}\left(\frac{\boldsymbol{x}-\boldsymbol{z}}{\alpha+1}\right)$
scaling $\rho \in \mathbb{R}^*$	$f(\rho\boldsymbol{x})$	$\frac{1}{\rho}\operatorname{prox}_{\rho^2 f}(\rho\boldsymbol{x})$
quadratic function $\boldsymbol{L} \in \mathbb{R}^{M\times N}$, $\gamma > 0$, $\boldsymbol{z} \in \mathbb{R}^M$	$\gamma\\|\boldsymbol{L}\boldsymbol{x} - \boldsymbol{z}\\|^2/2$	$(\mathrm{Id} + \gamma\boldsymbol{L}\boldsymbol{L}^*)^{-1}(\mathsf{x} - \gamma\boldsymbol{L}^*\boldsymbol{z})$
semi-unitary transform $\boldsymbol{L} \in \mathbb{R}^{M\times N}$, $\boldsymbol{L}\boldsymbol{L}^* = \mu\mathrm{Id}$, $\mu > 0$	$f(\boldsymbol{L}\boldsymbol{x})$	$\boldsymbol{x} - \mu^{-1}\boldsymbol{L}^*\big(\boldsymbol{x} - \operatorname{prox}_{\mu f}(\boldsymbol{L}\boldsymbol{x})\big)$
reflexion	$f(-\boldsymbol{x})$	$-\operatorname{prox}_f(-\boldsymbol{x})$
separability	$\sum_{i=1}^{N}\varphi_i(x^{(i)})$ $\boldsymbol{x} = (x^{(i)})_{1\leqslant i\leqslant N}$	$\left(\operatorname{prox}_{\varphi_i}(x^{(i)})\right)_{1\leqslant i\leqslant N}$
indicator function	$\iota_C(\boldsymbol{x})$	$\mathsf{P}_C(\boldsymbol{x})$
support function	$\iota_C^*(\boldsymbol{x}) = \sigma_C(\boldsymbol{x})$	$\boldsymbol{x} - \mathsf{P}_C(\boldsymbol{x})$

See more on http://proximity-operator.net/

Optimization algorithms for solving stochastic problem

Stochastic forward-backward algorithm

STOCHASTIC PROBLEM

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \mathbb{E}(\varphi_j(\boldsymbol{h}_j^\top \boldsymbol{x}, y_j)) + g(\boldsymbol{D}\boldsymbol{x})$$

$\Rightarrow$ At each iteration $j \geqslant 1$, assume that an estimate $\boldsymbol{u}_j$ of the gradient of $\Phi(\cdot) = \mathbb{E}(\varphi_j(\boldsymbol{h}_j^\top \cdot, y_j))$ at $\boldsymbol{x}_j$ is available.

Stochastic forward-backward algorithm

STOCHASTIC PROBLEM

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \;\; \mathbb{E}(\varphi_j(\boldsymbol{h}_j^\top \boldsymbol{x}, y_j)) + g(\boldsymbol{D}\boldsymbol{x})$$

$\Rightarrow$ At each iteration $j \geqslant 1$, assume that an estimate $\boldsymbol{u}_j$ of the gradient of $\Phi(\cdot) = \mathbb{E}(\varphi_j(\boldsymbol{h}_j^\top \cdot, y_j))$ at $\boldsymbol{x}_j$ is available.

The SFB algorithm reads:

$(\gamma_j)_{j\geqslant 1} \in]0,+\infty[, (\lambda_j)_{j\geqslant 1} \in]0,1]$
for $j = 1, 2, \ldots$

$$\boldsymbol{z}_j = \operatorname{prox}_{\gamma_j g\circ \boldsymbol{D}}\left(\boldsymbol{x}_j - \gamma_j \boldsymbol{u}_j\right)$$
$$\boldsymbol{x}_{j+1} = (1-\lambda_j)\boldsymbol{x}_j + \lambda_j \boldsymbol{z}_j$$

- When $g \equiv 0$, the stochastic gradient descent (SGD) algorithm is recovered.

Convergence theorem [Rosasco *et al.*, 2014]

Let F $\neq \varnothing$ denote the set of minimizers of $\Phi + g \circ \boldsymbol{D}$. Assume that:

(i) Φ has a β-Lipschitzian gradient with $\beta \in]0, +\infty[$, g is a proper, lower-semicontinuous convex function, and $\Phi + g \circ \boldsymbol{D}$ is strongly convex.

(ii) For every $j \geqslant 1$,

$$\mathbb{E}(\{\|\boldsymbol{u}_j\|^2\}) < +\infty, \ \mathbb{E}\{\boldsymbol{u}_j \mid \mathfrak{X}_{j-1}\} = \nabla\Phi(\boldsymbol{x}_j),$$
$$\mathbb{E}\{\|\boldsymbol{u}_j - \nabla\Phi(\boldsymbol{x}_j)\|^2 \mid \mathfrak{X}_{j-1}\} \leqslant \sigma^2(1 + \alpha_j\|\nabla\Phi(\boldsymbol{x}_j)\|^2)$$

where $\mathfrak{X}_j = (y_i, \boldsymbol{h}_i)_{1 \leqslant i \leqslant j}$, and α_j and σ are positive values such that $\gamma_j \leqslant (2-\epsilon)/(\beta(1+2\sigma^2\alpha_j))$ with $\epsilon > 0$.

(iii) We have

$$\sum_{j \geqslant 1} \lambda_j \gamma_j = +\infty \quad \text{and} \quad \sum_{j \geqslant 1} \chi_j^2 < +\infty$$

where, for every $j \geqslant 1$, $\chi_j^2 = \lambda_j \gamma_j^2 (1 + 2\alpha_j \|\nabla\Phi(\overline{\boldsymbol{x}})\|^2)$ and $\overline{\boldsymbol{x}} \in$ F.

Then, $(\boldsymbol{x}_j)_{j \geqslant 1}$ converges almost surely to an element of F.

Bibliographical remarks

Related approaches

- Methods relying on subgradient steps [Shalev-Shwartz *et al.*, 2007],
- Regularized dual averaging methods [Xiao, 2010],
- Composite mirror descent methods [Duchi *et al.*, 2010].

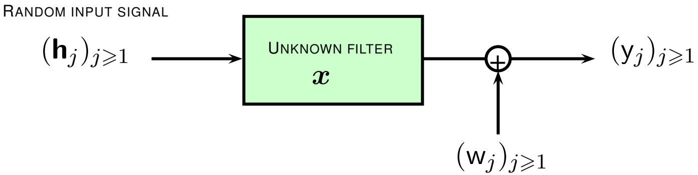
What if prox of $g \circ \boldsymbol{D}$ is not simple?

- Stochastic proximal averaging strategy [Zhong *et al.*, 2014],
- Conditional gradient ($\sim$ Franck-Wolfe) techniques [Lafond, 2015],
- Stochastic ADMM [Ouyang *et al.*, 2013],
- Block alternating strategy [Xu *et al.*, 2014],
- Stochastic proximal primal-dual methods (also for varying g) [Combettes *et al.*, 2015].

How to accelerate convergence?

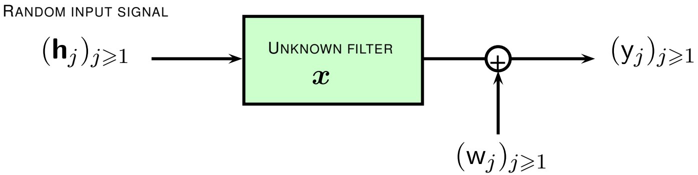
- Subspace acceleration techniques [Hu *et al.*, 2009][Atchadé *et al.*, 2014],
- Preconditioning techniques [Duchi *et al.*, 2011],
- Mixing both strategies (smooth case) [Chouzenoux *et al.*, 2014].

A brief focus on sparse adaptive filtering



$\Rightarrow$ Previous stochastic problem, with $(\forall j \geqslant 1)\ \varphi_j(\mathbf{h}_j^\top \boldsymbol{x}, \mathrm{y}_j) = (\mathbf{h}_j^\top \boldsymbol{x} - \mathrm{y}_j)^2$.

A brief focus on sparse adaptive filtering



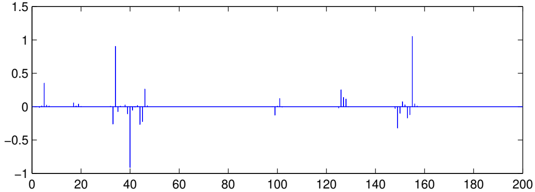
⇒ Previous stochastic problem, with $(\forall j \geqslant 1)\,\varphi_j(\mathbf{h}_j^\top \boldsymbol{x}, \mathrm{y}_j) = (\mathbf{h}_j^\top \boldsymbol{x} - \mathrm{y}_j)^2$.

EXISTING WORKS IN CASE OF SPARSE PRIOR:

- Proportionate least mean square methods ($\sim$ Preconditioned SGD) [Paleologu *et al.*, 2010],
- Zero-attracting algorithms ($\sim$ subgradient descent) [Chen *et al*, 2010],
- Proximal-like algorithms: SFB [Yamagashi *et al.*, 2011] or primal-dual approach [Ono *et al.*, 2013],
- Penalized versions of recursive least squares [Angelosante *et al.*,2011],
- Over-relaxed projection algorithms [Kopsinis *et al.*, 2011],
- Time-varying filters $\rightsquigarrow$ affine projection strategy ($\sim$ mini-batch in machine learning) [Markus *et al.*, 2014].

Simulation results

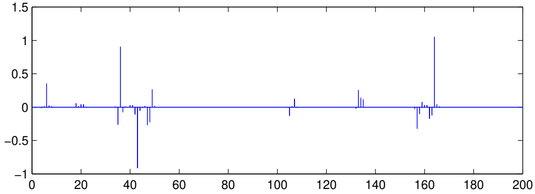
- $\boldsymbol{x}$: Time-variant linear system with 200 sparse coefficients,
- $\boldsymbol{h}$: Input sequence of 5000 random independent variables uniformly distributed on $\{-1, +1\}$,
- $\boldsymbol{w}$: White Gaussian noise with zero mean and variance 0.05.



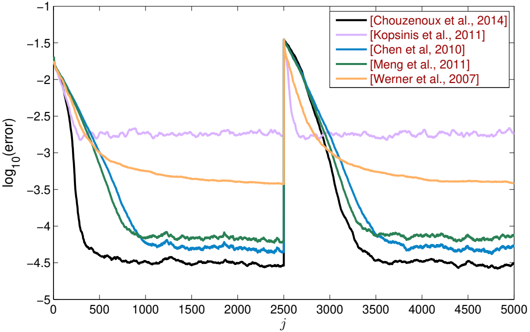
Values of the coefficients of the true sparse filter $\boldsymbol{x}$ for $1 \leqslant j \leqslant 2500$

Simulation results

- $\boldsymbol{x}$: Time-variant linear system with 200 sparse coefficients,
- $\boldsymbol{h}$: Input sequence of 5000 random independent variables uniformly distributed on $\{-1, +1\}$,
- $\boldsymbol{w}$: White Gaussian noise with zero mean and variance 0.05.



Values of the coefficients of the true sparse filter $\boldsymbol{x}$ for $2501 \leqslant j \leqslant 5000$



Estimation error along time, for various sparse adaptive filtering strategies

- The parameters of each tested method (forgetting factor, stepsize, regularization weight, affine projection blocksize) are optimized manually,
- The Stochastic Majorize-Minimize Memory gradient (S3MG) algorithm from [Chouzenoux *et al.*, 2014] leads to a minimal estimation error, while benefiting from good tracking properties.

Stochastic algorithms for solving batch problem

Incremental gradient algorithms

BATCH PROBLEM

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \quad \frac{1}{M}\sum_{i=1}^{M}\varphi_i(\boldsymbol{h}_i^\top\boldsymbol{x}, y_i) + g(\boldsymbol{D}\boldsymbol{x})$$

$\Rightarrow$ At each iteration $n \geqslant 0$, some $j_n \in \{1,\ldots,M\}$ is randomly chosen, and only the gradient of $\varphi_{j_n}(\boldsymbol{h}_{j_n}^\top\cdot, y_{j_n})$ at $\boldsymbol{x}_n$ is computed.

Incremental gradient algorithms

BATCH PROBLEM

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \quad \frac{1}{M}\sum_{i=1}^{M}\varphi_i(\boldsymbol{h}_i^\top \boldsymbol{x}, y_i) + g(\boldsymbol{D}\boldsymbol{x})$$

⇒ At each iteration $n \geqslant 0$, some $j_n \in \{1,\ldots,M\}$ is randomly chosen, and only the gradient of $\varphi_{j_n}(\boldsymbol{h}_{j_n}^\top \cdot, y_{j_n})$ at $\boldsymbol{x}_n$ is computed.

For instance, the SAGA algorithm [Defazio *et al.*, 2014] reads:

$\gamma \in]0,+\infty[$, and $(\forall i \in \{1,\ldots,M\})\boldsymbol{z}_{i,0} = \boldsymbol{x}_0 \in \mathbb{R}^N$.
for $n = 0, 1, \ldots$

Select randomly $j_n \in \{1,\ldots,M\}$,

$$\boldsymbol{u}_n = \boldsymbol{h}_{j_n}\nabla\varphi_{j_n}(\boldsymbol{h}_{j_n}^\top \boldsymbol{x}_n, y_{j_n}) - \boldsymbol{h}_{j_n}\nabla\varphi_{j_n}(\boldsymbol{h}_{j_n}^\top \boldsymbol{z}_{j_n,n}, y_{j_n}) + \tfrac{1}{M}\textstyle\sum_{i=1}^{M} \boldsymbol{h}_i\nabla\varphi_i(\boldsymbol{h}_i^\top \boldsymbol{z}_{i,n}, y_i)$$

$$\boldsymbol{x}_{n+1} = \text{prox}_{\gamma g\circ \boldsymbol{D}}\Big(\boldsymbol{x}_n - \gamma \boldsymbol{u}_n\Big)$$

$\boldsymbol{z}_{j_n,n+1} = \boldsymbol{x}_{n+1}$, and $(\forall i \in \{1,\ldots,M\})\, \boldsymbol{z}_{i,n+1} = \boldsymbol{z}_{i,n}$

Convergence theorem [Defazio *et al.*, 2014]

Let $\Phi(\cdot) = \frac{1}{M}\sum_{i=1}^{M} \varphi_i(\boldsymbol{h}_i^\top \cdot, y_i)$. Denote by F $\neq \varnothing$ the set of minimizers of $\Phi + g \circ \boldsymbol{D}$. If:

(i) Φ is convex, β-Lipschitz differentiable on $\mathbb{R}^N$, and g is proper, lower-semicontinuous convex on $\mathbb{R}^N$,

(ii) For every $n \in \mathbb{N}$, j_n is drawn from an i.i.d. uniform distribution on $\{1, \dots, M\}$,

Then, for $\gamma = 1/3\beta$, for $n \in \mathbb{N}^*$,

$$\mathrm{E}\left((\Phi + g \circ \boldsymbol{D})(\overline{\boldsymbol{x}}_n)\right) - (\Phi + g \circ \boldsymbol{D})(\widehat{\boldsymbol{x}}) \leqslant \frac{4M}{n}\Big(\frac{2\beta}{M}\|\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}\|^2 + \Phi(\boldsymbol{x}_0) - \nabla\Phi(\widehat{\boldsymbol{x}})^\top(\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}) - \Phi(\widehat{\boldsymbol{x}})\Big),$$

where $\widehat{\boldsymbol{x}} \in$ F and $\overline{\boldsymbol{x}}_n = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{x}_j$.

If, additionally, Φ is ν-strongly convex then, for $\gamma = 1/(2(\nu M + \beta))$,

$$\mathrm{E}\left(\|\boldsymbol{x}_n - \widehat{\boldsymbol{x}}\|^2\right) \leqslant \left(1 - \frac{\nu}{\gamma}\right)^n \Big(\|\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}\|^2 + 2\gamma M\big(\Phi(\boldsymbol{x}_0) - \nabla\Phi(\widehat{\boldsymbol{x}})^\top(\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}) - \Phi(\widehat{\boldsymbol{x}})\big)\Big).$$

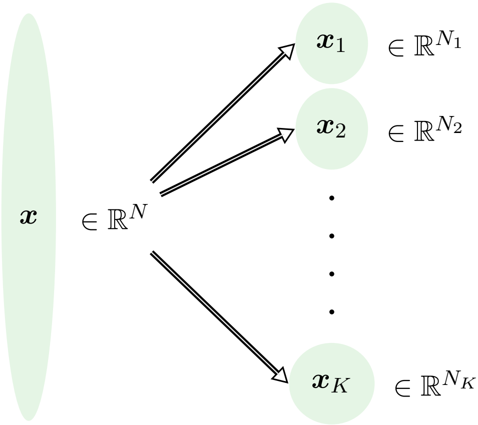
Bibliographical remarks

⇒ Links between stochastic incremental methods existing in the literature:

ALGORITHM	GENERAL IDEA	PROS/CONS	REFS
Standard incremental gradient	$\boldsymbol{u}_n = \boldsymbol{h}_{j_n} \nabla \varphi_{j_n}(\boldsymbol{h}_{j_n}^\top \boldsymbol{x}_n, y_{j_n})$	simplicity / decreasing stepsize required	[Bertsekas, 2010]
Variance reduction approaches (SVRG / mSGD)	At every $K \geqslant 0$ iterations, perform a full gradient step ($\sim$ mini-batch strategy)	reduced memory / more gradient evaluations	[Konečný, 2014], [Johnson *et al*, 2014]
Gradient averaging (SAG / SAGA)	Factor $1/M$ in front of gradient difference term	lower variance / increasing bias (in gradient estimates)	[Schmidt *et al*, 2014], [Defazio *et al*, 2014]
Proximal averaging (FINITO)	$\boldsymbol{x}_{n+1} = \operatorname{prox}_{\gamma g \circ \boldsymbol{D}}\left(\overline{\boldsymbol{z}}_n - \gamma \boldsymbol{u}_n\right)$ with $\overline{\boldsymbol{z}}_n$ average of $(\boldsymbol{z}_{i,n})_{1 \leqslant i \leqslant M}$	extra storage cost / less gradient evaluations	[Defazio *et al.*, 2014]
Majorization-Minimization (MISO)	$\boldsymbol{x}_{n+1}$ minimizer of a majorant function of $\varphi_{j_n}(\boldsymbol{h}_{j_n}^\top \cdot, y_{j_n}) + g \circ \boldsymbol{D}$ at $\overline{\boldsymbol{z}}_n$	extra storage cost / less gradient evaluations	[Mairal, 2015]

Block coordinate approaches

- Idea: variable splitting.



Assumption: $g(\boldsymbol{D}\boldsymbol{x}) = \sum_{k=1}^{K} g_{1,k}(\boldsymbol{x}_k) + g_{2,k}(\boldsymbol{D}_k\boldsymbol{x}_k)$ where, for every $k \in \{1,\ldots,K\}$, $\boldsymbol{D}_k \in \mathbb{R}^{P_k \times N_k}$.

Stochastic primal-dual proximal algorithm [Pesquet *et al.*, 2015]

$\tau \in]0,+\infty[, \gamma \in]0,+\infty[,$

for $n = 1, 2, \ldots$

for $k = 1, 2, \ldots, K$

with probability $\varepsilon_k \in]0,1]$ do

$$\boldsymbol{v}_{k,n+1} = (\mathrm{Id} - \mathrm{prox}_{\tau^{-1} g_{2,k}})(\boldsymbol{v}_{k,n} + \boldsymbol{D}_k \boldsymbol{x}_{k,n})$$

$$\boldsymbol{x}_{k,n+1} = \mathrm{prox}_{\gamma g_{1,k}}\Big(\boldsymbol{x}_{k,n} - \gamma\big(\tau \boldsymbol{D}_k^\top (2\boldsymbol{v}_{k+1,n} - \boldsymbol{v}_{k,n}) + \frac{1}{M}\sum_{i=1}^M \boldsymbol{h}_{i,k} \nabla\varphi_i(\textstyle\sum_{k'=1}^K \boldsymbol{h}_{i,k'}^T \boldsymbol{x}_{k',n}, y_i)\big)\Big)$$

otherwise

$$\boldsymbol{v}_{k,n+1} = \boldsymbol{v}_{k,n}, \ \boldsymbol{x}_{k,n+1} = \boldsymbol{x}_{k,n}.$$

- When $g_{2,k} \equiv 0$, the random block coordinate forward-backward algorithm is recovered [Combettes *et al.*, 2015],
- When $g_{1,k} \equiv 0$ and $g_{2,k} \equiv 0$, the random block coordinate descent algorithm is obtained [Nesterov, 2012].

Convergence theorem [Pesquet *et al.*, 2015]

Set, for every $n \in \mathbb{N}^*$, $\boldsymbol{\mathcal{X}}_n = (\boldsymbol{x}_{n'}, \boldsymbol{v}_{n'})_{1 \leqslant n' \leqslant n}$.
Let F $\neq \varnothing$ denote the set of minimizers of $\Phi + g \circ \boldsymbol{D}$.
Assume that:

(i) Φ is convex, β-Lipschitz differentiable on $\mathbb{R}^N$, g is lower-semicontinuous convex on $\mathbb{R}^N$,

(ii) The blocks activation is performed at each iteration n independently of $\boldsymbol{\mathcal{X}}_n$, with positive probabilities $(\varepsilon_1, \ldots, \varepsilon_K)$,

(iv) The primal and dual stepsizes (τ, γ) satisfy $\frac{1}{\tau} - \gamma \max_{1 \leqslant k \leqslant K} \|\boldsymbol{D}_k\|^2 > \frac{\beta}{2}$,

Then, $(\boldsymbol{x}_n)_{n \in \mathbb{N}^*}$ converges weakly almost surely to an F-valued random variable.

Bibliographical remarks

CONVERGENCE ANALYSIS

- Almost sure convergence [Pesquet *et al.*, 2015],
- Worst case convergence rates [Richtarik *et al.*, 2014] [Necoara *et al.*, 2014] [Lu *et al.*, 2015].

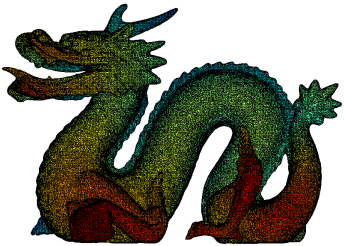
VARIANTS OF THE METHOD

- Improved convergence conditions in some specific cases [Fercoq *et al.*, 2015],
- Dual ascent strategies in the strongly convex case ($\sim$ dual forward-backward) [Shalev-Shwartz *et al.*, 2014] [Jaggi *et al.*, 2014] [Qu *et al.*, 2014],
- Douglas-Rachford/ADMM approaches [Combettes *et al.*, 2015] [Iutzeler *et al.*, 2013],
- Asynchronous distributed algorithms [Pesquet *et al.*, 2014] [Bianchi *et al.*, 2014].

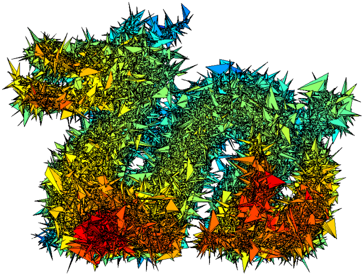
$\Rightarrow$ Dual ascent strategies and asynchronous distributed methods are closely related to **incremental gradient algorithms**.

Simulation results

(ANR GRAPHSIP)



Original mesh, $N = 100250$.

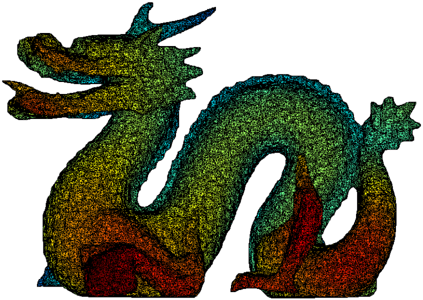


Noisy mesh, MSE = 2.89×10^{-6}.

Goal: Restore the nodes positions of an original mesh corrupted through an additive i.i.d. zero-mean Gaussian mixture noise model,

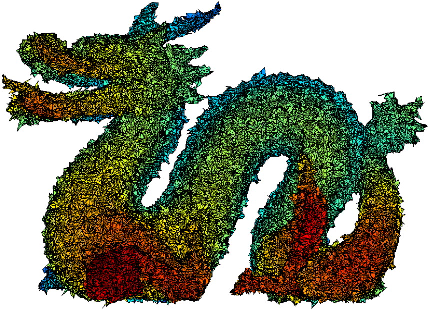
Limited memory available $\Rightarrow$ The mesh is decomposed into K/r non-overlapping blocks with size $r \leqslant K$, and ϵ is such that only one block is updated at each iteration.

- Reconstruction results using the stochastic primal-dual proximal algorithm for 3D mesh denoising from [Repetti *et al.*, 2015]:



Proposed reconstruction

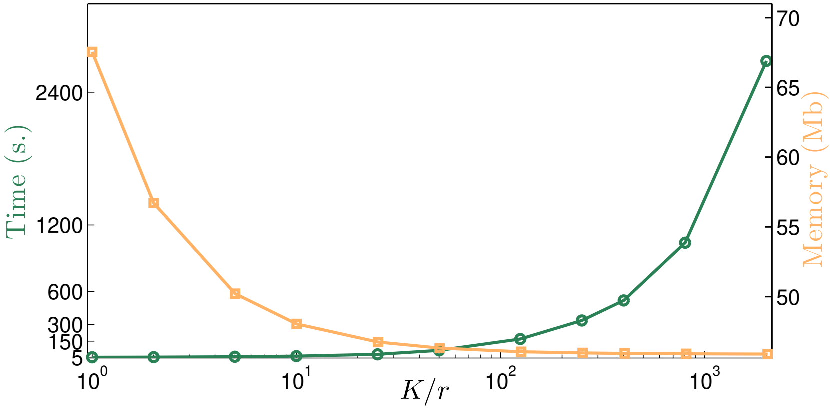
MSE = 8.09×10^{-8}



Laplacian smoothing

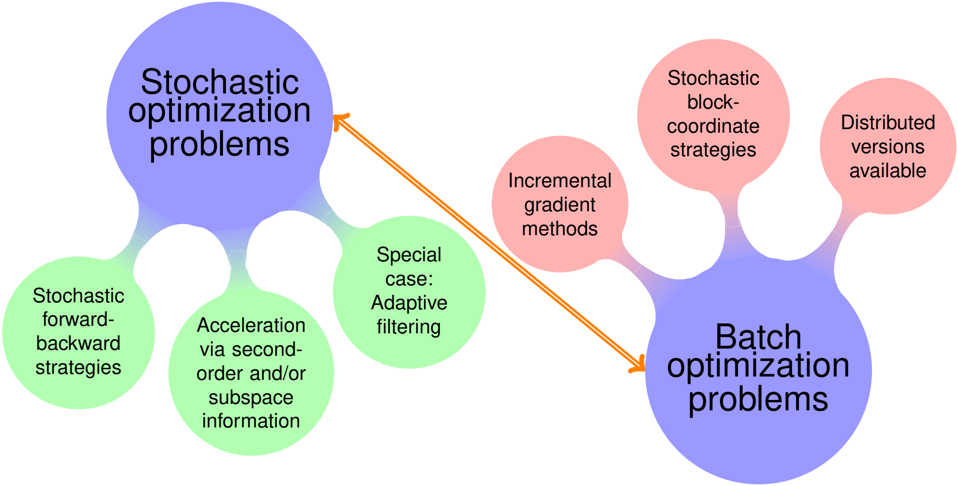
MSE = 5.23×10^{-7}

► Reconstruction results using the stochastic primal-dual proximal algorithm for 3D mesh denoising from [Repetti *et al.*, 2015]:



Memory requirement, and computation time, for different number of blocks.

Conclusion


Stochastic optimization problems
Stochastic forward-backward strategies
Acceleration via second-order and/or subspace information
Special case: Adaptive filtering
Incremental gradient methods
Stochastic block-coordinate strategies
Distributed versions available
Batch optimization problems

M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. O. Hero and S. McLaughlin.
A Survey of Stochastic Simulation and Optimization Methods in Signal Processing.
To appear in IEEE Journal of Selected Topics in Signal Processing. Available at http://arxiv.org/abs/1505.00273

P. Combettes and J.-C Pesquet
Stochastic Quasi-Fejér Block-Coordinate Fixed Point Iterations with Random Sweeping
SIAM Journal on Optimization, 25(2), pp. 1221-1248, 2015.

J.-C. Pesquet and A. Repetti
A Class of Randomized Primal-Dual Algorithms for Distributed Optimization
to appear in *Journal of Nonlinear and Convex Analysis*, 2015.

A. Repetti, E. Chouzenoux and J.-C. Pesquet
A Random Block-Coordinate Primal-Dual Proximal Algorithm with Application to 3D Mesh Denoising
IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2015), pp. 3561-3565, Brisbane, Australia, Apr. 19-24, 2015.

E. Chouzenoux, J.-C. Pesquet and A. Florescu.
A Stochastic 3MG Algorithm with Application to 2D Filter Identification
European Signal Processing Conference (EUSIPCO 2014), pp. 1587-1591, Lisboa, Portugal, 1-5 Sept. 2014.