An Overview of Stochastic Methods for Solving Optimization Problems

Émilie Chouzenoux

Center for Visual Computing, CentraleSupelec, INRIA Saclay

19 Dec. 17
Introduction

STOCHASTIC PROBLEM

\[
\minimize_{x \in \mathbb{R}^N} \mathbb{E}(\varphi_j(h_j^\top x, y_j)) + g(Dx)
\]

where \( j \in \mathbb{N}^* \), \( h_j \in \mathbb{R}^N \), \( y_j \in \mathbb{R} \), \( \varphi_j : \mathbb{R} \times \mathbb{R} \to ]-\infty, +\infty[ \) is a loss function, and \( g \circ D \) is a regularization function, with \( g : \mathbb{R}^P \to ]-\infty, +\infty[ \) and \( D \in \mathbb{R}^{P \times N} \).
**Introduction**

**STOCHASTIC PROBLEM**

\[
\min_{x \in \mathbb{R}^N} \mathbb{E}(\varphi_j(h_j^T x, y_j)) + g(Dx)
\]

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

**BATCH PROBLEM**

\[
\min_{x \in \mathbb{R}^N} \frac{1}{M} \sum_{i=1}^{M} \varphi_i(h_i^T x, y_i) + g(Dx)
\]

where, for all \( i \in \{1, \ldots, M\} \), \( \varphi_i : \mathbb{R} \times \mathbb{R} \to ] - \infty, +\infty] \), \( h_i \in \mathbb{R}^N \) and \( y_i \in \mathbb{R} \).
Link between stochastic and batch problems

\[ j \in \mathbb{N}^* \text{ is deterministic,} \]
\[ (\forall i \in \{2, \ldots, M\}) \ \varphi_i = \varphi_1, \]
and \( (h_j)_{j \geq 1}, (y_j)_{j \geq 1} \) are i.i.d random variables.
Link between stochastic and batch problems

STOCHASTIC PROBLEM

$y$ and $H$ 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 \( f : \mathbb{R}^N \rightarrow ]-\infty, +\infty] \).

- **The domain** of function \( f \) is
  \[
  \text{dom } f = \{ x \in \mathbb{R}^N \mid f(x) < +\infty \}
  \]

  If \( \text{dom } f \neq \emptyset \), function \( f \) is said to be **proper**.

- **Function** \( f \) is **convex** if
  \[
  (\forall (x, y) \in (\mathbb{R}^N)^2)(\forall \lambda \in [0, 1])
  \]
  \[
  f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).
  \]

- **Function** \( f \) is **lower semi-continuous** (lsc) on \( \mathbb{R}^N \) if, for all \( x \in \mathbb{R}^N \), for all sequence \( (x_k)_{k \in \mathbb{N}} \) of \( \mathbb{R}^N \),
  \[
  x_k \rightarrow x \quad \Rightarrow \quad \lim \inf f(x_k) \geq f(x).
  \]
Let $f : \mathbb{R}^N \to ]-\infty, +\infty]$. Function $f$ is said $\nu$-strongly convex if

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

\[
f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) - \frac{1}{2} \nu \lambda (1 - \lambda) \|x - y\|^2,
\]

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

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

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

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

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

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

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

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

The subdifferential of a convex function \( f: \mathbb{R}^N \rightarrow ]-\infty, +\infty] \) at \( x \) is the set

\[
\partial f(x) = \{ t \in \mathbb{R}^N \mid (\forall y \in \mathbb{R}^N) \ f(y) \geq f(x) + \langle t \mid y - x \rangle \}
\]

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

▶ If \( f \) is differentiable at \( x \in \mathbb{R}^N \) then \( \partial f(x) = \{ \nabla f(x) \} \).
Proximity operator

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

**Characterization of Proximity Operator**

\[
(\forall x \in \mathbb{R}^N) \quad \hat{y} = \text{prox}_f(x) \iff x - \hat{y} \in \partial f(\hat{y}).
\]

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

\[
f(\hat{y}) + \frac{1}{2}\|\hat{y} - x\|^2 = \inf_{y \in \mathbb{R}^N} f(y) + \frac{1}{2}\|y - x\|.
\]
## Properties of proximal operator

<table>
<thead>
<tr>
<th>Translation</th>
<th>$z \in \mathbb{R}^N$</th>
<th>$f(x - z)$</th>
<th>$z + \text{prox}_f(x - z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadratic perturbation</td>
<td>$z \in \mathbb{R}^N$, $\alpha &gt; 0$, $\gamma \in \mathbb{R}$</td>
<td>$f(x) + \alpha |x|^2/2 + \langle x \mid z \rangle + \gamma$</td>
<td>$\text{prox}_{\frac{f}{\alpha+1}}\left(\frac{x-z}{\alpha+1}\right)$</td>
</tr>
<tr>
<td>Scaling</td>
<td>$\rho \in \mathbb{R}^*$</td>
<td>$f(\rho x)$</td>
<td>$\frac{1}{\rho} \text{prox}_{\rho^2 f}(\rho x)$</td>
</tr>
<tr>
<td>Quadratic function</td>
<td>$L \in \mathbb{R}^{M \times N}$, $\gamma &gt; 0$, $z \in \mathbb{R}^M$</td>
<td>$\gamma |Lx - z|^2/2$</td>
<td>$\left(\text{Id} + \gamma LL^*\right)^{-1}(x - \gamma L^*z)$</td>
</tr>
<tr>
<td>Semi-unitary transform</td>
<td>$L \in \mathbb{R}^{M \times N}$, $LL^* = \mu \text{Id}$, $\mu &gt; 0$</td>
<td>$f(Lx)$</td>
<td>$x - \mu^{-1}L^*(x - \text{prox}_{\mu f}(Lx))$</td>
</tr>
<tr>
<td>Reflexion</td>
<td></td>
<td>$f(-x)$</td>
<td>$-\text{prox}_f(-x)$</td>
</tr>
<tr>
<td>Separability</td>
<td>$\sum_{i=1}^N \varphi_i(x^{(i)})$</td>
<td>$\left(\text{prox}<em>{\varphi_i}(x^{(i)})\right)</em>{1 \leq i \leq N}$</td>
<td></td>
</tr>
<tr>
<td>Indicator function</td>
<td>$\iota_C(x)$</td>
<td>$\pi_C(x)$</td>
<td></td>
</tr>
<tr>
<td>Support function</td>
<td>$\iota_C^*(x) = \sigma_C(x)$</td>
<td>$x - \pi_C(x)$</td>
<td></td>
</tr>
</tbody>
</table>

See more on [http://proximity-operator.net/](http://proximity-operator.net/)
Optimization algorithms for solving stochastic problem
Stochastic forward-backward algorithm

\[
\text{minimize} \quad \mathbb{E}(\phi_j(h_j^T x, y_j)) + g(Dx)
\]

⇒ At each iteration \( j \geq 1 \), assume that an estimate \( u_j \) of the gradient of \( \Phi(\cdot) = \mathbb{E}(\phi_j(h_j^T \cdot, y_j)) \) at \( x_j \) is available.
Stochastic forward-backward algorithm

**STOCHASTIC PROBLEM**

\[
\min_{x \in \mathbb{R}^N} \mathbb{E}(\varphi_j(h_j^T x, y_j)) + g(Dx)
\]

⇒ At each iteration \( j \geq 1 \), assume that an estimate \( u_j \) of the gradient of \( \Phi(\cdot) = \mathbb{E}(\varphi_j(h_j^T \cdot, y_j)) \) at \( x_j \) is available.

The SFB algorithm reads:

\[
\begin{align*}
(\gamma_j)_{j \geq 1} & \in ]0, +\infty[, (\lambda_j)_{j \geq 1} \in ]0, 1] \\
\text{for } j = 1, 2, \ldots \\
& \quad z_j = \text{prox}_{\gamma_j g \circ D}(x_j - \gamma_j u_j) \\
& \quad x_{j+1} = (1 - \lambda_j)x_j + \lambda_j z_j
\end{align*}
\]

- When \( g \equiv 0 \), the stochastic gradient descent (SGD) algorithm is recovered.
Convergence theorem [Rosasco et al., 2014]

Let $F \neq \emptyset$ denote the set of minimizers of $\Phi + g \circ D$. Assume that:

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

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

\[
\mathbb{E}(\{\|u_j\|^2\}) < +\infty, \quad \mathbb{E}\{u_j \mid X_{j-1}\} = \nabla \Phi(x_j),
\]

\[
\mathbb{E}\{|u_j - \nabla \Phi(x_j)|^2 \mid X_{j-1}\} \leq \sigma^2 (1 + \alpha_j \|\nabla \Phi(x_j)\|^2)
\]

where $X_j = (y_i, h_i)_{1 \leq i \leq j}$, and $\alpha_j$ and $\sigma$ are positive values such that $\gamma_j \leq (2 - \epsilon)/\left(\beta(1 + 2\sigma^2\alpha_j)\right)$ with $\epsilon > 0$.

(iii) We have

\[
\sum_{j \geq 1} \lambda_j \gamma_j = +\infty \quad \text{and} \quad \sum_{j \geq 1} \chi_j^2 < +\infty
\]

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

Then, $(x_j)_{j \geq 1}$ 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 $g \circ D$ IS NOT SIMPLE?**
- Stochastic proximal averaging strategy [Zhong et al., 2014],
- Conditional gradient (∼ 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

\[ \begin{align*}
(\mathbf{h}_j)_{j \geq 1} & \quad \rightarrow \quad \text{UNKNOWN FILTER} \quad \mathbf{x} \\
(\mathbf{w}_j)_{j \geq 1} & \quad \rightarrow \quad (y_j)_{j \geq 1}
\end{align*} \]

⇒ Previous stochastic problem, with \((\forall j \geq 1) \varphi_j(\mathbf{h}_j^\top \mathbf{x}, y_j) = (\mathbf{h}_j^\top \mathbf{x} - y_j)^2\).
A brief focus on sparse adaptive filtering

\[
(h_j)_{j \geq 1} \rightarrow \text{Unknown filter} \rightarrow (y_j)_{j \geq 1}
\]

\[
(w_j)_{j \geq 1}
\]

⇒ Previous stochastic problem, with \((\forall j \geq 1) \varphi_j(h_j^\top x, y_j) = (h_j^\top x - y_j)^2\).

**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 \(\sim\) affine projection strategy (\(\sim\) mini-batch in machine learning) [Markus et al., 2014].
Simulation results

\( \mathbf{x} \) : Time-variant linear system with 200 sparse coefficients,

\( \mathbf{h} \) : Input sequence of 5000 random independent variables uniformly distributed on \( \{-1, +1\} \),

\( \mathbf{w} \) : White Gaussian noise with zero mean and variance 0.05.

Values of the coefficients of the true sparse filter \( \mathbf{x} \) for \( 1 \leq j \leq 2500 \)
Simulation results

\( x \) : Time-variant linear system with 200 sparse coefficients,

\( h \) : Input sequence of 5000 random independent variables uniformly distributed on \([-1, +1]\),

\( w \) : White Gaussian noise with zero mean and variance 0.05.

Values of the coefficients of the true sparse filter \( x \) for \( 2501 \leq j \leq 5000 \)
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

\[
\begin{align*}
\text{minimize} \quad & \frac{1}{M} \sum_{i=1}^{M} \varphi_i(h_i^\top x, y_i) + g(Dx) \\
\text{subject to} \quad & x \in \mathbb{R}^N
\end{align*}
\]

⇒ At each iteration \( n \geq 0 \), some \( j_n \in \{1, \ldots, M\} \) is randomly chosen, and only the gradient of \( \varphi_{j_n}(h_{j_n}^\top \cdot, y_{j_n}) \) at \( x_n \) is computed.
**Incremental gradient algorithms**

**Batch problem**

\[
\text{minimize } \quad \frac{1}{M} \sum_{i=1}^{M} \varphi_i(h_i^\top x, y_i) + g(Dx)
\]

⇒ At each iteration \( n \geq 0 \), some \( j_n \in \{1, \ldots, M\} \) is randomly chosen, and only the gradient of \( \varphi_{j_n}(h_{j_n}^\top \cdot, y_{j_n}) \) at \( x_n \) is computed.

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

\[
\gamma \in ]0, +\infty[, \text{ and } (\forall i \in \{1, \ldots, M\}) z_{i,0} = x_0 \in \mathbb{R}^N.
\]

for \( n = 0, 1, \ldots \)

- Select randomly \( j_n \in \{1, \ldots, M\} \),

\[
\begin{align*}
 u_n &= h_{j_n} \nabla \varphi_{j_n}(h_{j_n}^\top x_n, y_{j_n}) - h_{j_n} \nabla \varphi_{j_n}(h_{j_n}^\top z_{j_n,n}, y_{j_n}) \\
 &\quad + \frac{1}{M} \sum_{i=1}^{M} h_i \nabla \varphi_i(h_i^\top z_{i,n}, y_i) \\
 x_{n+1} &= \text{prox}_{\gamma g \circ D}(x_n - \gamma u_n) \\
 z_{j_n,n+1} &= x_{n+1}, \text{ and } (\forall i \in \{1, \ldots, M\}) z_{i,n+1} = z_{i,n}
\end{align*}
\]
Convergence theorem [Defazio et al., 2014]

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

(i) $\Phi$ is convex, $\beta$-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 $\{1, \ldots, M\}$,

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

$$
\mathbb{E} \left( (\Phi + g \circ D)(\overline{x}_n) \right) - (\Phi + g \circ D)(\hat{x}) \leq \frac{4M}{n} \left( \frac{2\beta}{M} \|x_0 - \hat{x}\|^2 + \Phi(x_0) - \nabla \Phi(\hat{x})^T (x_0 - \hat{x}) - \Phi(\hat{x}) \right),
$$

where $\hat{x} \in F$ and $\overline{x}_n = \frac{1}{n} \sum_{j=1}^{n} x_j$.

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

$$
\mathbb{E} \left( \|x_n - \hat{x}\|^2 \right) \leq \left(1 - \frac{\nu}{\gamma}\right)^n \left( \|x_0 - \hat{x}\|^2 + 2\gamma M (\Phi(x_0) - \nabla \Phi(\hat{x})^T (x_0 - \hat{x}) - \Phi(\hat{x})) \right).
$$
Bibliographical remarks

⇒ Links between stochastic incremental methods existing in the literature:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>General idea</th>
<th>Pros/Cons</th>
<th>Refs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard incremental gradient</strong></td>
<td>$u_n = h_{jn} \nabla \varphi_{jn} (h_{jn}^\top x_n, y_{jn})$</td>
<td>simplicity / decreasing stepsize required</td>
<td>[Bertsekas, 2010]</td>
</tr>
<tr>
<td><strong>Variance reduction approaches</strong></td>
<td>At every $K \geq 0$ iterations, perform a full gradient step ($\sim$ mini-batch strategy)</td>
<td>reduced memory / more gradient evaluations</td>
<td>[Konečný, 2014], [Johnson et al, 2014]</td>
</tr>
<tr>
<td><strong>Gradient averaging</strong></td>
<td>Factor $1/M$ in front of gradient difference term</td>
<td>lower variance / increasing bias (in gradient estimates)</td>
<td>[Schmidt et al, 2014], [Defazio et al, 2014]</td>
</tr>
<tr>
<td><strong>Proximal averaging</strong></td>
<td>$x_{n+1} = \text{prox}_{\gamma g \circ D} \left( \bar{z}<em>n - \gamma u_n \right)$ with $\bar{z}<em>n$ average of $(z</em>{i,n})</em>{1 \leq i \leq M}$</td>
<td>extra storage cost / less gradient evaluations</td>
<td>[Defazio et al., 2014]</td>
</tr>
<tr>
<td><strong>Majorization-Minimization</strong></td>
<td>$x_{n+1}$ minimizer of a majorant function of $\varphi_{jn} (h_{jn}^\top \cdot, y_{jn}) + g \circ D$ at $\bar{z}_n$</td>
<td>extra storage cost / less gradient evaluations</td>
<td>[Mairal, 2015]</td>
</tr>
</tbody>
</table>
Block coordinate approaches

▶ Idea: variable splitting.

**Assumption:** \( g(Dx) = \sum_{k=1}^{K} g_{1,k}(x_k) + g_{2,k}(D_k x_k) \) where, for every \( k \in \{1, \ldots, K\} \), \( D_k \in \mathbb{R}^{P_k \times N_k} \).
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

\[
\begin{align*}
\mathbf{v}_{k,n+1} &= (\text{Id} - \text{prox}_{\tau^{-1}g_{2,k}})(\mathbf{v}_{k,n} + D_k \mathbf{x}_{k,n}) \\
\mathbf{x}_{k,n+1} &= \text{prox}_{\gamma g_{1,k}} \left( \mathbf{x}_{k,n} - \gamma \left( \tau D_k^T (2\mathbf{v}_{k+1,n} - \mathbf{v}_{k,n}) \\
&\quad + \frac{1}{M} \sum_{i=1}^{M} h_{i,k} \nabla \varphi_i \left( \sum_{k'=1}^{K} h_{i,k'}^T \mathbf{x}_{k',n}, y_i \right) \right) \right)
\end{align*}
\]

otherwise

\[
\begin{align*}
\mathbf{v}_{k,n+1} &= \mathbf{v}_{k,n}, \quad \mathbf{x}_{k,n+1} = \mathbf{x}_{k,n}.
\end{align*}
\]

- 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 $n \in \mathbb{N}^*$, $\mathcal{X}_n = (x_{n'}, v_{n'})_{1 \leq n' \leq n}$.
Let $F \neq \emptyset$ denote the set of minimizers of $\Phi + g \circ D$.
Assume that:

(i) $\Phi$ is convex, $\beta$-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 $\mathcal{X}_n$, with positive probabilities $(\varepsilon_1, \ldots, \varepsilon_K)$,

(iv) The primal and dual stepsizes $(\tau, \gamma)$ satisfy
$$\frac{1}{\tau} \geq \gamma \max_{1 \leq k \leq K} \|D_k\|^2 > \frac{\beta}{2},$$

Then, $(x_n)_{n \in \mathbb{N}^*}$ 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 (∼ dual forward-backward) [Shalev-Shwartz et al., 2014] [Jaggi et al., 2014] [Qu et al., 2014],
- Douglas-Rachford/ADMM approaches [Combettes et al., 2015] [lutzeler et al., 2013],
- Asynchronous distributed algorithms [Pesquet et al., 2014] [Bianchi et al., 2014].

⇒ 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 = \( 2.89 \times 10^{-6} \).

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 \( r \leq K \), and \( \epsilon \) 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 \times 10^{-8}

Laplacian smoothing
MSE = 5.23 \times 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

Batch optimization problems

- Incremental gradient methods
- Stochastic block-coordinate strategies
- Distributed versions available

Stochastic problem

Batch problem


P. Combettes and J.-C Pesquet

Stochastic Quasi-Fejér Block-Coordinate Fixed Point Iterations with Random Sweeping


J.-C. Pesquet and A. Repetti

A Class of Randomized Primal-Dual Algorithms for Distributed Optimization

A. Repetti, E. Chouzenoux and J.-C. Pesquet

A Random Block-Coordinate Primal-Dual Proximal Algorithm with Application to 3D Mesh Denoising


E. Chouzenoux, J.-C. Pesquet and A. Florescu.

A Stochastic 3MG Algorithm with Application to 2D Filter Identification