A Splitting Algorithm for Minimization under Stochastic Linear Constraints

Adil Salim
adil-salim.github.io

Telecom ParisTech

July 4th, 2018

Joint work with Pascal Bianchi and Walid Hachem

Outline

Stochastic Subgradient Algorithm

$$\min_{x \in \mathbb{R}^d} \mathbf{F}(x), \qquad \mathbf{F}(x) = \mathbb{E}_\xi(f(x, \xi))$$

where ξ is a r.v., for every s, $f(\cdot, s) \in \Gamma_0(\mathbb{R}^d)$ has a full domain, and for every x, $f(x, \cdot)$ is measurable.

Stochastic subgradient algorithm (generalizes the Law of Large Numbers)

$$x_{n+1} = x_n - \gamma_{n+1} \widetilde{\nabla} f(x_n, \xi_{n+1})$$

where

- (ξ_n) i.i.d copies of ξ
- $(\gamma_n) \in \ell^2 \setminus \ell^1$ is a sequence of positive numbers.
- $\widetilde{\nabla} f(x, s)$ is a subgradient of $f(\cdot, s)$ at point $x \in \mathbb{R}^d$.

Theorem : $x_n \to x_\star \in \arg\min \mathbf{F}$ a.s.

Example : Portfolio optimization

Define $\Delta = \{x \in \mathbb{R}^d, \quad \sum_{i=1}^{d} x(i) = 1, \quad \forall i, x(i) \geq 0\}$, $d \geq 1$.

Markowitz portfolio optimization

$$\min_{x \in \Delta} \mathbb{E}_\xi(\langle x, \xi \rangle^2) \quad \text{subject to} \quad \mathbb{E}_\xi(\langle x, \xi \rangle) = r$$

where $r > 0$[1] and ξ is a random variable (r.v.) in $\mathbb{R}^d$ with distribution μ.

The distribution μ is unknown but revealed across time through i.i.d realizations $(\xi_n)_{n \in \mathbb{N}}$ of ξ.

[1] Plenary talk of S. Ahmed this afternoon

The Problem

Solve

$$\min_{x\in\mathbb{R}^d, z\in\mathbb{R}^p} (\mathbf{F}+\mathbf{G})(x) + (\mathbf{P}+\mathbf{Q})(z) \quad \text{s.t.} \quad \mathbf{A}x + \mathbf{B}z = \mathbf{c} \qquad (1)$$

where

- $\mathbf{F}, \mathbf{G}, \mathbf{P}, \mathbf{Q}$ are proper, lsc, convex functions s.t. $\forall x \in \mathbb{R}^d, \mathbf{F}(x) < \infty$ and $\forall z \in \mathbb{R}^p, \mathbf{P}(z) < \infty$.
- $\mathbf{A}, \mathbf{B}$ are matrices
- $\mathbf{c} \in \mathbb{R}^q$ is a vector.

One can use Vu-Condat algorithm [Vu'13, Condat'13]

Stochastic Optimization Framework

- $\mathbf{F}(x) = \mathbb{E}_{\xi}(f(x,\xi))$ where ξ is a r.v., for every s, $f(\cdot, s)$ is a convex function over $\mathbb{R}^d$, and for every x, $f(x, \cdot)$ is measurable.
- Similar representation for $\mathbf{G}, \mathbf{P}, \mathbf{Q}$: $\mathbf{G}(x) = \mathbb{E}_{\xi}(g(x,\xi)), \mathbf{P}(x) = \mathbb{E}_{\xi}(p(x,\xi)), \mathbf{Q}(x) = \mathbb{E}_{\xi}(q(x,\xi))$.
- $\mathbf{A} = \mathbb{E}(A)$ where A is a random matrix.
- Similar representation for $\mathbf{B}, \mathbf{c}$: $\mathbf{B} = \mathbb{E}(B), \mathbf{c} = \mathbb{E}(c)$.

The distributions of ξ, A, B, c are unknown but revealed across time through i.i.d realizations ξ_n, A_n, B_n, c_n.

The Proposed Algorithm

At iteration $n+1$, the previous iterate is $(x_n, z_n, \lambda_n) \in \mathbb{R}^d \times \mathbb{R}^p \times \mathbb{R}^q$, and $(\xi_{n+1}, A_{n+1}, B_{n+1}, c_{n+1})$ is observed. Then,

$$
\begin{aligned}
x_{n+1} &= \mathrm{prox}_{\gamma_{n+1} g(\cdot,\xi_{n+1})}\left(x_n - \gamma_{n+1}(\widetilde{\nabla} f(x_n,\xi_{n+1}) + A_{n+1}^T \lambda_n)\right), \\
z_{n+1} &= \mathrm{prox}_{\gamma_{n+1} q(\cdot,\xi_{n+1})}\left(z_n - \gamma_{n+1}(\widetilde{\nabla} p(z_n,\xi_{n+1}) + B_{n+1}^T \lambda_n)\right), \\
\lambda_{n+1} &= \lambda_n + \gamma_{n+1}\left(A_{n+1} x_n + B_{n+1} z_n - c_{n+1}\right).
\end{aligned}
\tag{2}
$$

where

- $(\gamma_n) \in \ell^2 \setminus \ell^1$ is a sequence of positive numbers.
- $\widetilde{\nabla} f(x,s)$ is a subgradient of $f(\cdot,s)$ at point $x \in \mathbb{R}^d$.
- $\mathrm{prox}_{\gamma g}$ is the proximity operator[2] of g : $\forall x \in \mathbb{R}^d, \gamma > 0$,

$$
\mathrm{prox}_{\gamma g}(x) = \arg\min_{y \in \mathbb{R}^d} \frac{1}{2\gamma}\|x - y\|^2 + g(y).
$$

[2] Plenary talk of M. Teboulle yesterday

Convergence of the Algorithm

- If $\mathbf{G}, \mathbf{P}, \mathbf{Q}, \mathbf{A}, \mathbf{B}, \mathbf{c}$ are equal to zero, then Problem (1) is equivalent to $\min \mathbf{F}$ and Algorithm (2) boils down to the stochastic subgradient algorithm.
- If $\mathbf{P}, \mathbf{Q}, \mathbf{A}, \mathbf{B}, \mathbf{c}$ are equal to zero, then Problem (1) is equivalent to $\min \mathbf{F} + \mathbf{G}$ and Algorithm (2) boils down to the stochastic proximal gradient algorithm.

Theorem (BH'15)

In this case, a.s. $x_n \longrightarrow_{n \to +\infty} x_\star \in \arg\min \mathbf{F} + \mathbf{G}$.

- In the general case, define $\bar{x_n} = \frac{\sum_{k=1}^n \gamma_k x_k}{\sum_{k=1}^n \gamma_k}$ and similarly $\bar{z_n}, \bar{\lambda_n}$.

Theorem (SBH'18)

$(\bar{x_n}, \bar{z_n}, \bar{\lambda_n}) \longrightarrow_{n \to +\infty} (x_\star, z_\star, \lambda_\star)$ *a.s. where* $(x_\star, z_\star)$ *is a.s. a solution of Problem* (1) *and* $\lambda_\star$ *is a.s. a dual solution of* (1).

Outline

Maximal Monotone Operators[3]

Euclidean space X, operator $\mathbf{A} : \mathsf{X} \rightrightarrows \mathsf{X}$

- $\mathbf{A}$ is identified with its graph $\mathrm{gr}(\mathbf{A}) = \{(x, y) \in \mathsf{X} \times \mathsf{X}, y \in \mathbf{A}(x)\}$
- $\mathbf{A}^{-1} := \{(y, x) \in \mathsf{X} \times \mathsf{X}, x \in \mathbf{A}(y)\}$
- $\mathcal{Z}(\mathbf{A}) = \mathbf{A}^{-1}(0) = \{x \in \mathsf{X}, 0 \in \mathbf{A}(x)\}$
- $\mathbf{A}$ is **monotone** if $\forall (x_1, y_1), (x_2, y_2) \in \mathbf{A}, \langle y_1 - y_2, x_1 - x_2 \rangle \geq 0$
- $\mathbf{A}$ is **maximal monotone** if $\mathbf{A}$ is monotone and maximal among monotone operators (for $\subset$)
- In this case, the **resolvent** $J_{\gamma \mathbf{A}} = (I + \gamma \mathbf{A})^{-1} : \mathsf{X} \to \mathsf{X}$ is a contraction [Minty'62]

Examples :

- $\mathbf{A} = \partial \mathbf{G}, \mathbf{G} \in \Gamma_0(\mathsf{X})$, $\mathcal{Z}(\partial \mathbf{G}) = \arg\min \mathbf{G}$, $J_{\gamma \partial \mathbf{G}} = \mathrm{prox}_{\gamma \mathbf{G}}$
- $\mathbf{A}$ a skew-symmetric matrix

[3] Keynote of P.L. Combettes this morning, [Bauschke & Combettes '11]

Example

$$\mathbf{A} \in \mathscr{M}(\mathsf{X}) = \{\text{Maximal monotone operators over } \mathsf{X}\}$$

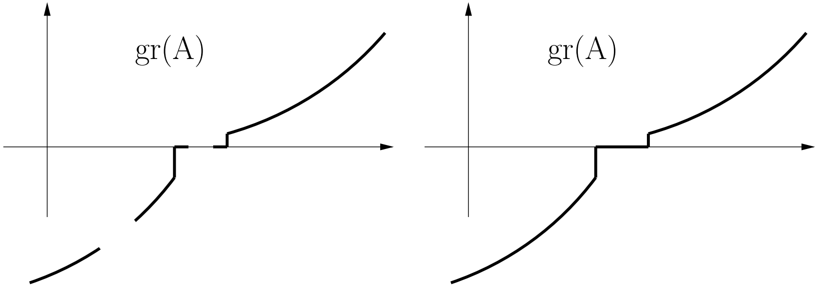

Figure 1: Left: A non maximal monotone operator over $\mathbb{R}$. Right: A maximal extension of the monotone operator

Write $\mathbf{A} = \mathbf{M} + \mathbf{M}'$ where $\mathbf{M}' : \mathsf{X} \to \mathsf{X}$.
Aim : Find $x_\star \in \mathcal{Z}(\mathbf{M} + \mathbf{M}')$

Forward Backward algorithm

Algorithm to find $x_\star \in \mathcal{Z}(\mathbf{M} + \mathbf{M}')$

$$x_{n+1} = J_{\gamma \mathbf{M}}(x_n - \gamma \mathbf{M}'(x_n))$$

Many examples like the proximal gradient algorithm, Chambolle-Pock, Vu-Condat...

If (**cocoercivity**) : $\langle \mathbf{M}'(x_1) - \mathbf{M}'(x_2), x_1 - x_2 \rangle \geq c\|x_1 - x_2\|^2$ and $\gamma < 2c$ then

$$x_n \longrightarrow_{n \to +\infty} x_\star \in \mathcal{Z}(\mathbf{M} + \mathbf{M}')$$

Random monotone operators

Random variable A with values in $\mathscr{M}(\mathsf{X})$ [Attouch'79]
Expectation : $x \in \mathsf{X}$

$$\mathbb{E}(A)(x) = \{\mathbb{E}(\varphi), \varphi \in A(x) \text{ a.s., } \varphi \text{ integrable}\}$$

Example : $A = \partial g(\cdot, \xi)$, $\mathbb{E}(\partial g(\cdot, \xi)) = \partial \mathbf{G}$ where
$\mathbf{G}(x) = \mathbb{E}_\xi(g(x, \xi))$ [Rockafellar & Wets'82]

Stochastic Forward Backward algorithm

M, M' random monotone operators with unknown distribution.
Denote $\mathbf{M} = \mathbb{E}(M), \mathbf{M}' = \mathbb{E}(M')$.
Algorithm to find $x_\star \in \mathcal{Z}(\mathbf{M} + \mathbf{M}')$.

$$x_{n+1} = J_{\gamma_{n+1} M_{n+1}}(x_n - \gamma_{n+1} M'_{n+1}(x_n))$$

where $(M_n)_n$ are i.i.d copies of M (similarly for M') and $(\gamma_n) \in \ell^2 \setminus \ell^1$.

Theorem (BH'15)

$\bar{x_n} \longrightarrow_{n\to+\infty} x_\star$ *where* $x_\star \in \mathcal{Z}(\mathbf{M} + \mathbf{M}')$ *a.s.*

No need of cocoercivity thanks to the decreasing step size.[4]

---

[4] Ad: If $\gamma_n \equiv \gamma$ is constant and cocoercivity holds then $\bar{x_n}$ converges to $\mathcal{Z}(\mathbf{M} + \mathbf{M}')$ in Probability as $n \to +\infty$ and $\gamma \to 0$, see [BHS'18]

Outline

Saddle Points

Recall Problem (1)

$$\min_{x\in\mathbb{R}^d, z\in\mathbb{R}^p} \mathbf{F}(x)+\mathbf{G}(x)+\mathbf{P}(z)+\mathbf{Q}(z) \quad \text{s.t.} \quad \mathbf{A}x+\mathbf{B}z=\mathbf{c}$$

We look for **saddle points** of the Lagrangian function

$$L(x,z,\lambda)=\mathbf{F}(x)+\mathbf{G}(x)+\mathbf{P}(z)+\mathbf{Q}(z)+\langle\lambda,\mathbf{A}x+\mathbf{B}z-\mathbf{c}\rangle$$

Then, (x,z,λ) is a saddle point iff

$$\begin{cases} 0 & \in & \partial\mathbf{F}(x)+\partial\mathbf{G}(x)+\mathbf{A}^T\lambda, \\ 0 & \in & \partial\mathbf{P}(z)+\partial\mathbf{Q}(z)+\mathbf{B}^T\lambda, \\ 0 & = & -\mathbf{A}x-\mathbf{B}z+\mathbf{c}\,. \end{cases} \tag{3}$$

which is equivalent to **finding zeros** of $\mathbf{M}+\mathbf{M}'$:

$$\boxed{\begin{bmatrix}0\\0\\0\end{bmatrix} \in \underbrace{\begin{bmatrix}\partial\mathbf{G}(x)\\ \partial\mathbf{Q}(z)\\ \mathbf{c}\end{bmatrix}}_{=\mathbf{M}(x,z,\lambda)} + \underbrace{\begin{bmatrix}\partial\mathbf{F}(x)+\mathbf{A}^T\lambda\\ \partial\mathbf{P}(z)+\mathbf{B}^T\lambda\\ -\mathbf{A}x-\mathbf{B}z\end{bmatrix}}_{=\mathbf{M}'(x,z,\lambda)}} \tag{4}$$

Apply Stochastic Forward Backward to the Saddle Point Problem

- $\mathbf{M}, \mathbf{M}' \in \mathscr{M}(\mathbb{R}^d \times \mathbb{R}^p \times \mathbb{R}^q)$
- $\mathbf{M}'(x, z, \lambda) = \mathbb{E}(M')(x, z, \lambda)$ where

$$M'(x, z, \lambda) = \begin{bmatrix} \partial f(x, \xi) + A^T \lambda \\ \partial p(z, \xi) + B^T \lambda \\ -Ax - Bz \end{bmatrix}$$

- $\mathbf{M}(x, z, \lambda) = \mathbb{E}(M)(x, z, \lambda)$ where

$$M(x, z, \lambda) = \begin{bmatrix} \partial g(x, \xi) \\ \partial q(z, \xi) \\ c \end{bmatrix} \quad \text{and} \quad J_{\gamma M}(x, z, \lambda) = \begin{bmatrix} \mathrm{prox}_{\gamma g(\cdot, \xi)}(x) \\ \mathrm{prox}_{\gamma q(\cdot, \xi)}(z) \\ \lambda - \gamma c \end{bmatrix}$$

- The iterations (2) are the iterations of the stochastic Forward Backward applied to solve (4) with i.i.d copies (M_n) and (M'_n) of M and M'.
- Theorem 2 is a consequence of Theorem 3.

Some questions

- An algorithm which is close to Algorithm (2) :

$$
\begin{aligned}
x_{n+1} &= \mathrm{prox}_{\gamma_{n+1} g(\cdot,\xi_{n+1})}\left(x_n - \gamma_{n+1}(\widetilde{\nabla} f(x_n,\xi_{n+1}) + A_{n+1}^T\lambda_n)\right), \\
z_{n+1} &= \mathrm{prox}_{\gamma_{n+1} q(\cdot,\xi_{n+1})}\left(z_n - \gamma_{n+1}(\widetilde{\nabla} p(z_n,\xi_{n+1}) + B_{n+1}^T\lambda_n)\right), \\
\lambda_{n+1} &= \lambda_n + \gamma_{n+1}\left(A_{n+1}(2x_{n+1} - x_n) + B_{n+1}(2z_{n+1} - z_n) - c_{n+1}\right).
\end{aligned}
\tag{5}
$$

  Can be rederive from Vu and Condat point of view [Vu'13, Condat'13]. Numerically more stable.

- Algorithm (2) can be seen as a noisy discretization of

$$(\dot{x}(t), \dot{z}(t), \dot{\lambda}(t)) \in -(\mathbf{M} + \mathbf{M}')(x(t), z(t), \lambda(t)).$$

  Is a Langevin version meaningful ?