A characterization of nonexpansive frugal splitting methods

Anton Åkerman, PhD student, Lund University
Dept. of Automatic Control
Supervisors: Pontus Giselsson (LU) and Sebastian Banert (Bremen)

LUND UNIVERSITY

Motivation & Research Goals

Proximal splitting methods are a simple class of first-order methods suitable high-dimensional, convex and non-smooth optimization problems. We find a simple parameterization and a condition guaranteeing convergence for a certain class of proximal splitting methods. This completely characterization all methods in this class that are nonexpansive.
Joint work with Enis Chenchene and Emanuele Naldi.

Problem description

We consider methods for solving the optimization problem

$$\text{minimize}_{x\in\mathbb{R}^d} \sum_{i=1}^{F} f_i(x) + \sum_{i=1}^{B} g_i(x), \tag{1}$$

for $B \geq 2$, $F \in \mathbb{N}$ all functions convex, with each f_i differentiable with β_i-Lipschitz gradient.
Frugal splitting methods use the following building blocks

- One gradients for each differentiable functions f_i per iteration
- One evaluation of the proximal operator, defined as $\text{prox}_{\gamma g}(x) = \text{argmin}_z \left(g(z) + \frac{1}{2\gamma}\|z-x\|^2\right)$, for each function g_i per iteration
- Linear combinations

Why splitting?

- Proximal operator potentially much cheaper to evaluate for the individual function, functions if prox friendly
- Possible to solve problems distributively

Consider an optimization problem on the form $\text{minimize}_{x\in C} f(x) = \min_{x\in\mathbb{R}^d} f(x) + \iota_C(x)$, which can be solved with the proximal gradient method. The iterates of the projected gradient method, $x_{k+1} = \Pi_C(x_k - \gamma\nabla f(x_k))$, converge to a minimizer.

Results

For a splitting method we want to guarantee the following:

- Fixed-point encoding, i.e. algorithm fixed-points always correspond to solutions of our optimization problem.
- Guaranteed convergence (usually by showing the update is so called averaged nonexpansive)

We show that

> All frugal splitting methods (that are fixed-point encoding) (1), in which the iterates $\boldsymbol{z}^k \in \mathbb{R}^{n\times d}$ have minimal dimension $n = B-1$, can be parameterized as
>
> $$x_i^k = \text{prox}_{\gamma_i g_i}\left(\gamma_i L_i \boldsymbol{x}^k + \gamma_i M_i \boldsymbol{z}^k - \gamma_i \sum_{j=1}^{F} N_{ij}\nabla f_j(G_j \boldsymbol{x}^k)\right)$$
> $$\boldsymbol{z}^{k+1} = \boldsymbol{z}^k - M^T \boldsymbol{x}^k$$
>
> for algorithm points
>
> $$\begin{cases} \boldsymbol{x}^k = (x_1^k, x_2^k, \dots, x_B^k)^T \in \mathbb{R}^{B\times d} \\ \boldsymbol{z}^k \in \mathbb{R}^{(B-1)\times d}. \end{cases}$$
>
> and for $\mathbf{0} < \gamma \in \mathbb{R}^B$, $L \in \mathbb{R}^{B\times B}$ strictly lower triangular, with $G, N^T \in \mathbb{R}^{F\times B}$ and $M \in \mathbb{R}^{B\times(B-1)}$. Additionally we require that $\mathbb{1}^T\gamma = \mathbb{1}^T L\mathbb{1}$, $M^T\mathbb{1} = 0$ and $N^T\mathbb{1} = G\mathbb{1} = \mathbb{1}$.

Our convergence result is as follows.

> Let $\beta \in \mathbb{R}^f$ be the vector of smoothness constants for the functions f_i. A method on this form is nonexpansive if and only if
>
> $$2\,\text{diag}(\gamma)^{-1} - L - L^T - MM^T \succeq \left(K^T - N\right)\text{diag}(\beta)\left(K - N^T\right) \tag{2}$$

The same results hold also for the more general monotone inclusion problem, with operators either $\frac{1}{\beta}$-cocoercive or maximally monotone.

In summary:

- We give a simple, complete characterization of all nonexpansive frugal splitting methods with minimal lifting.
- Condition (2) gives standard parameter bounds for some existing splitting methods.
- Enables simple algorithm design. The choice of sparse matrices simplifies parallelization and use for distributed optimization.

References

[1] Frugal Splitting Operators: Representation, Minimal Lifting, and Convergence
M. Morin, S. Banert, P. Giselsson
SIAM Journal on Optimization, 2024

[2] Resolvent splitting for sums of monotone operators with minimal lifting
Y. Malitsky, M. Tam
Mathematical Programming, 2023

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM