IE 598: Big Data Optimization **Fall 2016**

Lecture 20: Splitting Algorithms

Lecturer: Niao He *Scribers: Juho Kim*

In this lecture, we discuss splitting algorithms for convex minimization problems with objective given by the sum of two nonsmooth functions. We start with the fixed point property of such problems and derive a general scheme of splitting algorithm based on fixed point iteration. This covers Douglas-Rachford splitting and Peaceman-Rachford splitting algorithms. We also discuss the convergence rate of the KM algorithm for general fixed point problem.

20.1 Introduction: Proximal Algorithms

Previously we have considered several proximal algorithms for convex problems under different settings.

- **Nonsmooth Minimization:** $\min_{x\in\mathbb{R}^n} f(x)$, where $f(x)$ is convex and nonsmooth. We show the optimal solution has the following fixed point property:

$$x_* \text{ is optimal } \iff 0 \in \partial f(x_*) \iff \forall \lambda > 0,\ x_* = \text{prox}_{\lambda f}(x_*) \tag{20.1}$$

  The fixed point iteration gives rise to the *proximal point algorithm*:

$$x_{t+1} = \text{prox}_{\lambda_t f}(x_t)$$

- **Smooth + Nonsmooth Minimization:** $\min_{x\in\mathbb{R}^n} f(x) + g(x)$, where $f(x)$ is smooth and convex, and $g(x)$ is nonsmooth and convex. In this case,

$$x_* \text{ is optimal } \iff 0 \in \nabla f(x_*) + \partial g(x_*). \iff \forall \lambda > 0,\ x_* = \text{prox}_{\lambda g}(x_* - \lambda \nabla f(x_*)) \tag{20.2}$$

  The fixed point iteration gives rise to the *proximal gradient algorithm*:

$$x_{t+1} = \text{prox}_{\lambda g}(x_t - \lambda_t \nabla f(x_t))$$

In this lecture, we consider the problem

- **Nonsmooth + Nonsmooth Minimization:** $\min_{x\in\mathbb{R}^n} f(x) + g(x)$, where both $f(x)$ and $g(x)$ are nonsmooth and convex functions. Apparently, we still have the fixed point property:

$$x_* \text{ is optimal } \iff 0 \in \partial f(x_*) + \partial g(x_*) \iff \forall \lambda > 0,\ x_* = \text{prox}_{\lambda(f+g)}(x_*) \tag{20.3}$$

  The fixed point iteration leads to

$$x_{t+1} = \text{prox}_{\lambda_t(f+g)}(x_t)$$

However, this would require the proximal operator of the sum of two convex function, which is not always easy to compute, even if the proximal operators of both functions separately may be easy to compute. For example, let $f(x) = \|x\|_1$ and $g(x) = \|Ax\|_2^2$. The proximal operators of f, g are given by

$$\text{prox}_f(y) = \text{argmin}_x \left\{ \frac{1}{2}\|x - y\|_2^2 + \|x\|_1 \right\}$$

$$\text{prox}_g(y) = \text{argmin}_x \left\{ \frac{1}{2}\|x-y\|_2^2 + \|Ax\|_2^2 \right\}$$

both are easy to compute. However, the proximal operator of the sum of the two functions $f+g$ is given by

$$\text{prox}_{(f+g)}(x) = \text{argmin}_x \left\{ \frac{1}{2}\|x-y\|_2^2 + \|x\|_1 + \|Ax\|_2^2 \right\}$$

is not easy to calculate.

Therefore, we need the above fixed point property (20.4) is not really useful. Intuitively, one might guess that $x_* = \text{prox}_{\lambda f}\left(\text{prox}_{\lambda g}(x)\right)$ and one can update $x_{t+1} = \text{prox}_{\lambda f}\left(\text{prox}_{\lambda g}(x_t)\right)$ by alternatively computing the proximal operators. For instance, if $f = \delta_{X_1}(\cdot)$ and $g = \delta_{X_2}(\cdot)$ are two indicator functions of convex sets X_1, X_2, this would imply an alternative projection on set X_1 and X_2. But this is not always true.

We show that instead, we have the following fixed point property:

$$x_* \text{ is optimal } \iff 0 \in \partial f(x_*) + \partial g(x_*) \iff \forall \lambda > 0,\ x_* = \text{prox}_{\lambda f}(y_*) \text{ and } y_* = \text{refl}_{\lambda g}(\text{refl}_{\lambda f}(y_*)) \quad (20.4)$$

where $\text{refl}_f = 2\text{prox}_f(x) - x$ is the reflection of the proximal operator. The corresponding fixed point iteration then leads to the so-called **splitting algorithms**. We now discuss the details.

20.2 Fix Point Theorem for Nonsmooth + Nonsmooth Problems

Consider the convex optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) + g(x) \quad (20.5)$$

where both $f(x)$ and $g(x)$ are proper convex (perhaps nonsmooth) functions.

Lemma 20.1 *x_* is optimal to (20.5) if and only if for any $\lambda > 0$ and $\rho \in \mathbb{R}$,*

$$x_* = prox_{\lambda f}(y_*) \text{ and } y_* = y_* + \rho[prox_{\lambda g}(2prox_{\lambda f}(y_*) - y_*) - prox_{\lambda f}(y_*)] \quad (20.6)$$

Proof: Suppose that x_* is optimal.

$$\begin{aligned}
x_* \text{ is optimal } &\iff 0 \in \partial f(x_*) + \partial g(x_*) \\
&\iff \forall \lambda > 0, \text{there exists } z \text{ such that } z \in \partial(\lambda f)(x_*) \text{ and } -z \in \partial(\lambda g)(x_*) \\
&\iff \forall \lambda > 0, \text{there exists } y \text{ such that } y - x_* \in \partial(\lambda f)(x_*) \text{ and } x_* - y \in \partial(\lambda g)(x_*) \\
&\iff \forall \lambda > 0, \text{there exists } y \text{ such that } y - x_* \in \partial(\lambda f)(x_*) \text{ and } 2x_* - y \in x_* + \partial(\lambda g)(x_*) \\
&\iff x_* = \text{prox}_{\lambda f}(y) \text{ and } 2\text{prox}_{\lambda f}(y) - y \in x_* + \partial(\lambda g)(x_*) \\
&\iff x_* = \text{prox}_{\lambda f}(y) \text{ and } x_* = \text{prox}_{\lambda g}(2\text{prox}_{\lambda f}(y) - y) \\
&\iff x_* = \text{prox}_{\lambda f}(y) \text{ and } y = y + \rho[\text{prox}_{\lambda g}(2\text{prox}_{\lambda f}(y) - y) - \text{prox}_{\lambda f}(y)], \forall \rho
\end{aligned}$$

Thus, both statements are equivalent. ■

Remark

1. when $\rho = 1$: we have
$$y_* = \frac{1}{2}[y_* + 2\text{prox}_{\lambda g}(2\text{prox}_{\lambda f}(y_*) - y_*) - (2\text{prox}_{\lambda f}(y_*) - y_*)] = \frac{1}{2}[y_* + \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(y_*)].$$
Hence, $y_* = \mathcal{T}_{\lambda,f,g}(y_*)$ with the operator
$$\mathcal{T}_{\lambda,f,g} = \frac{1}{2}[I + \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}]$$
this is known as the **Douglas-Rachford operator** [Lions and Mercier, 1979].

2. when $\rho = 2$: we have
$$y_* = y_* + 2\text{prox}_{\lambda g}(2\text{prox}_{\lambda f}(y_*) - y_*) - (2\text{prox}_{\lambda f}(y_*) - y_*) = y_* + \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(y_*).$$
Hence, $y_* = \mathcal{T}_{\lambda,f,g}(y_*)$ with the operator
$$\mathcal{T}_{\lambda,f,g} = \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}$$
this is known as the **Peaceman-Rachford operator** [Lions and Mercier, 1979].

Previously in Lecture 17, we show that the proximal operator is firmly nonexpansive, i.e.,
$$\|\text{prox}_f(x) - \text{prox}_f(y)\|_2^2 \leq \langle \text{prox}_f(x) - \text{prox}_f(y), x - y\rangle.$$
Indeed, both the Douglas-Rachford operator and the Peaceman-Rachford operator are both non-expansive operators, i.e. $\|\mathcal{T}_{\lambda,f,g}(x) - \mathcal{T}_{\lambda,f,g}(y)\|_2 \leq \|x - y\|_2, \forall x, y$.

First, we see that

Lemma 20.2 *The reflection operator $\text{refl}_{\lambda f}(\cdot)$ is non-expansive for any $\lambda > 0$.*

Proof: This is because
$$\begin{aligned}
\|\text{refl}_{\lambda f}(x) - \text{refl}_{\lambda f}(y)\|_2^2 &= \|2\text{prox}_{\lambda f}(x) - 2\text{prox}_{\lambda f}(y) - (x - y)\|_2^2 \\
&= 4\|\text{prox}_{\lambda f}(x) - \text{prox}_{\lambda f}(y)\|_2^2 - 4\langle \text{prox}_{\lambda f}(x) - \text{prox}_{\lambda f}(y), x - y\rangle + \|x - y\|_2^2 \\
&\leq 4\|\text{prox}_{\lambda f}(x) - \text{prox}_{\lambda f}(y)\|_2^2 - 4\|\text{prox}_{\lambda f}(x) - \text{prox}_{\lambda f}(y)\|_2^2 + \|x - y\|_2^2 \\
&= \|x - y\|_2^2
\end{aligned}$$
■

We are now able to show that

Lemma 20.3 *(i) The Peaceman-Rachford operator $T_1 = \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}$ is non-expansive.*

(ii) The Douglas-Rachford operator $T_2 = \frac{1}{2}[I + \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}]$ is non-expansive.

Proof:
$$\|T_1 x - T_1 y\|_2 = \|\text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(x) - \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(y)\|_2 \leq \|\text{refl}_{\lambda f}(x) - \text{refl}_{\lambda f}(y)\|_2 \leq \|x - y\|_2$$
where the first inequality is due to the non-expansiveness of $\text{prox}_{\lambda g}$ and the second inequality is due to the non-expansiveness of $\text{prox}_{\lambda f}$. Since $T_2 = \frac{1}{2}[I + T_1]$, then
$$\|T_2 x - T_2 y\|_2 \leq \frac{1}{2}\|x - y\|_2 + \frac{1}{2}\|T_1 x - T_1 y\|_2 \leq \|x - y\|_2$$
Thus, we have shown that both operators are non-expansive. ■

20.3 Splitting algorithms

The fixed point iterations corresponding to these non-expansive operators lead to the following algorithms:

1. *Douglas-Rachford splitting algorithm:*

$$y_{t+1} = \frac{1}{2}[y_t + \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(y_t)]$$

2. *Peaceman-Rachford splitting algorithm:*

$$y_{t+1} = \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(y_t)$$

3. *Relaxed Peaceman-Rachford splitting algorithm:* let $\gamma_t \in (0, 1]$,

$$y_{t+1} = (1 - \gamma_t)y_t + \gamma_t \cdot \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(y_t)$$

   (Special cases) $\gamma_t = 1$: Peaceman-Rachford splitting and $\gamma_t = \frac{1}{2}$: Douglas-Rachford splitting.

In the following, we illustrate the details of the Douglas-Rachford splitting algorithm and demonstrate that indeed it is a special case of the well-known Alternating Direction Method of Multipliers (ADMM).

20.3.1 Douglas-Rachford splitting

Let us initialize y_1 and $x_1 = \text{prox}_{\lambda f}(y_1)$, the Douglas-Rachford splitting algorithm can be rewritten as

$$\begin{cases} y_{t+1} = y_t + \text{prox}_{\lambda g}(2x_t - y_t) - x_t \\ x_{t+1} = \text{prox}_{\lambda f}(y_t) \end{cases}$$

Let $z_{t+1} = \text{prox}_{\lambda g}(2x_t - y_t)$, this can be further formulated as

$$\begin{cases} z_{t+1} = \text{prox}_{\lambda g}(2x_t - y_t) \\ x_{t+1} = \text{prox}_{\lambda f}(y_t + z_{t+1} - x_t) \\ y_{t+1} = y_t + z_{t+1} - x_t \end{cases}$$

Let $u_t = x_t - y_t$.

$$\begin{cases} z_{t+1} = \text{prox}_{\lambda g}(x_t + u_t) \\ x_{t+1} = \text{prox}_{\lambda f}(z_{t+1} - u_t) \\ u_{t+1} = u_t + (x_{t+1} - z_{t+1}) \end{cases}$$

which is a special case of the Alternating Direction Methods of Multipliers (ADMM).

20.3.2 Alternating Direction Methods of Multipliers (ADMM)

We consider the following optimization problem.

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \end{aligned} \tag{20.7}$$

Let $\rho > 0$, the augmented Lagrangian for this optimization is given as follows.

$$L_\rho(x, z, \lambda) = f(x) + g(z) + \lambda^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2.$$

ADMM consists of the iterations.

$$\begin{cases} z_{t+1} = \text{argmin}_z L_\rho(x_t, z, \lambda_t) \\ x_{t+1} = \text{argmin}_x L_\rho(x, z_{t+1}, \lambda_t) \\ \lambda_{t+1} = \lambda_t + \rho(Ax_{t+1} + Bz_{t+1} - c) \end{cases}$$

Letting $u_t = \frac{\lambda_t}{\rho}$, the iterations above are the followings

$$\begin{cases} z_{t+1} = \text{argmin}_z\{g(z) + \frac{\rho}{2}\|Ax_t + Bz - c + u_t\|_2^2\} \\ x_{t+1} = \text{argmin}_x\{f(x) + \frac{\rho}{2}\|Ax + Bz_{t+1} - c + u_t\|_2^2\} \\ u_{t+1} = u_t + (Ax_{t+1} + Bz_{t+1} - c) \end{cases}$$

A general convex optimization problem such as $\min_x f(x) + g(x)$ can be converted as the form that ADMM can be applied to.

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{s.t.} \quad & x - z = 0 \end{aligned}$$

This is a special case of (20.7) with $A = I, B = -I, c = 0$. Let $\rho = \frac{1}{\lambda}$, one can see that the ADMM algorithm is exactly the same as the Douglas-Rachford splitting algorithm in this case.

20.4 Convergence Analysis

Here we provide a unified analysis for fixed point iterative algorithms.

Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be a nonexpansive operator. The goal is to find a fixed point x_* such that $x_* = Tx_*$. We consider the relaxed fixed point algorithm

$$x_{t+1} = (1 - \gamma_t)x_t + \gamma_t \cdot Tx_t, \ \textit{for all } t \geq 0$$

where $\gamma_t \in (0, 1]$. This is known as the **Krasnosel'skii-Mann(KM) algorithm**. Note that when $\gamma_t = 1$, this reduces to the usual fixed point algorithm.

We point out that this algorithm covers most of the algorithms we discussed so far,

- Gradient descent: $x_{t+1} = Tx_t$, where $Tx = x - \frac{1}{L}\nabla f(x)$
- Projected gradient descent: $x_{t+1} = Tx_t$, where $Tx = \Pi_X(x - \frac{1}{L}\nabla f(x))$
- Proximal gradient descent: $x_{t+1} = Tx_t$, where $Tx = \text{prox}_{\frac{g}{L}}(x - \frac{1}{L}\nabla f(x))$
- Proximal point algorithm: $x_{t+1} = Tx_t$, where $Tx = \text{prox}_{\lambda f}(x)$
- Douglas-Rachford algorihtm : $x_{t+1} = Tx_t$, where $Tx = \frac{1}{2}[y + \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(x)]$
- Relaxed Peaceman-Rachford algorihtm : $x_{t+1} = (1 - \gamma_t)x_t + \gamma_t \cdot Tx_t$, where $Tx = \text{refl}_{\lambda g} \circ \text{refl}_{\lambda f}(x)$

Theorem 20.4 *Let T be a nonexpansive operator. The KM algorithm satisfies that*

$$\|Tx_t - x_t\|_2^2 \leq \frac{\|x_0 - x_*\|_2^2}{\sum_{\tau=0}^{t} \gamma_\tau(1-\gamma_\tau)}.$$

Proof: We first show that $\|x_t - Tx_t\|_2$ is non-increasing.

$$\begin{aligned}
\|x_{t+1} - Tx_{t+1}\|_2 &= \|(1-\gamma_t)x_t + \gamma_t Tx_t - Tx_{t+1}\|_2 \\
&= \|(1-\gamma_t)(x_t - Tx_t) + Tx_t - Tx_{t+1}\|_2 \\
&\leq (1-\gamma_t)\|x_t - Tx_t\|_2 + \|Tx_t - Tx_{t+1}\|_2 \quad \text{(Triangular inequality)} \\
&\leq (1-\gamma_t)\|x_t - Tx_t\|_2 + \|x_t - x_{t+1}\|_2 \quad \text{(T is a nonexpansive operator)} \\
&= (1-\gamma_t)\|x_t - Tx_t\|_2 + \gamma_t\|x_t - Tx_t\|_2 \\
&= \|x_t - Tx_t\|_2
\end{aligned}$$

We now show that

$$\begin{aligned}
\|x_{t+1} - x_*\|_2^2 &= \|(1-\gamma_t)x_t + \gamma_t Tx_t - (1-\gamma_t)x_* - \gamma_t Tx_*\|_2^2 \\
&= \|(1-\gamma_t)(x_t - x_*) + \gamma_t(Tx_t - Tx_*)\|_2^2 \\
&= (1-\gamma_t)\|x_t - x_*\|_2^2 + \gamma_t\|Tx_t - Tx_*\|_2^2 - \gamma_t(1-\gamma_t)\|x_t - Tx_t\|_2^2
\end{aligned}$$

The last equality is due to the fact that for any $\gamma \in [0,1]$ and u, v,

$$\|(1-\gamma)u + \gamma v\|_2^2 = (1-\gamma)\|u\|_2^2 + \gamma\|v\|_2^2 - \gamma(1-\gamma)\|u - v\|_2^2$$

Therefore,

$$\|x_{t+1} - x_*\|_2^2 \leq \|x_t - x_*\|_2^2 - \gamma_t(1-\gamma_t)\|x_t - Tx_t\|_2^2$$

Taking summation over t leads to

$$\left(\sum_{\tau=0}^{t} \gamma_\tau(1-\gamma_\tau)\right) \cdot \|x_t - Tx_t\|_2^2 \leq \sum_{\tau=0}^{t} \gamma_\tau(1-\gamma_\tau)\|x_\tau - Tx_\tau\|_2^2 \leq \|x_0 - x_*\|_2^2.$$

Thus,

$$\|Tx_t - x_t\|_2^2 \leq \frac{\|x_0 - x_*\|_2^2}{\sum_{\tau=0}^{t} \gamma_\tau(1-\gamma_\tau)}.$$

■

Remark. In particular, if we set $\gamma_t = \gamma \in (0,1)$, we have

$$\|Tx_t - x_t\|_2^2 \leq \frac{\|x_0 - x_*\|_2^2}{\gamma(1-\gamma)(t+1)}.$$

Thus, this indicates that the KM algorithm achieves an overall $O(1/t)$ rate of convergence for solving a fixed point problem. As an immediate fact, the relaxed Peaceman-Rachford algorithm also attains the same $O(1/t)$ rate. It remains interesting to investigate the acceleration of such algorithms; we will leave out the details here.

20.5 Example: LASSO

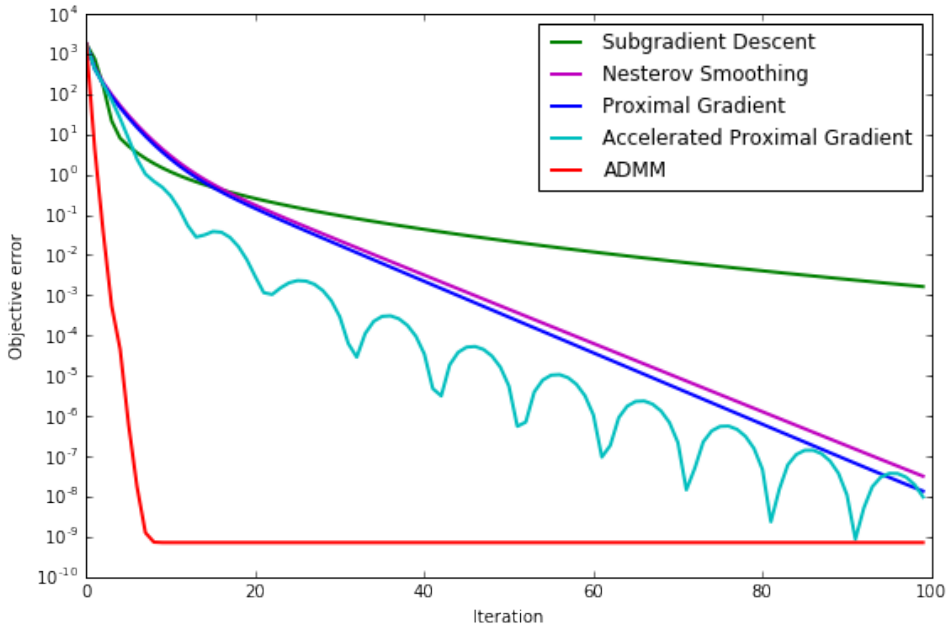
We illustrate the Douglas-Rachford/ADMM algorithm for solving the LASSO problem:

$$\min_x \underbrace{\frac{1}{2}||Ax-b||_2^2}_{f(x)} + \underbrace{\lambda||x||_1}_{g(x)}$$

The ADMM algorithm works as follows:

$$\begin{cases} z_{t+1} = S_{\lambda/\rho}(x_t + y_t/\rho) \\ x_{t+1} = (A^TA + \rho I)^{-1}(A^Tb + \rho z_{t+1} - \lambda_t) \\ \lambda_{t+1} = \lambda_t + \rho(x_{t+1} - z_{t+1}) \end{cases}$$

Below is the overall comparison of ADMM and the other five methods we implemented on the LASSO problem in the previous lecture:



We observe that ADMM significantly outperforms all other algorithms. One should note that in this case, ADMM treats the first data-fidelity term $f(x) = \frac{1}{2}\|Ax-b\|_2^2$ as a nonsmooth term and compute exactly its proximal operator at each iteration (which requires computing the inverse of a matrix), while other algorithms simply treat this as a smooth term and use only its gradient information at each iteration (which requires much cheaper computation cost).

References

[BPCPE11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, 2011.

[PB14] N. Parikh and S. Boyd, "Proximal Algorithms," *Foundations and Trends in Optimization*, 2014.

[RSV14] R. Cominetti, Roberto, J.A. Soto, and J. Vaisman, "On the rate of convergence of Krasnosel?ski?-Mann iterations and their connection with sums of Bernoullis." *Israel Journal of Mathematics* 199.2: 757-772, 2014.

[LM79] P.L. Lions, and B. Mercier, " Splitting algorithms for the sum of two nonlinear operators". *SIAM Journal on Numerical Analysis*, 16(6), 964-979, 1979.

[V16] L. Vandenberghe " Lecture Notes for EE236C: Optimization Methods for Large-Scale Systems ", Spring 2016.

[N16] He, Niao , "Python demo for LASSO " `http://niaohe.ise.illinois.edu/IE598/lasso_demo/index.html`, 2016.