Operator Splitting of Three or More Operators

Sun Zheng , Lin Kaixiang , Deng Ruiyang

July 25, 2013

Outline

- Introduction.
- Review of Some Operator Splitting Methods for Two Operators.
- From n=2 to n=3.
- From n=3 to n=m.
- Numerical Experiments and Results.
- Another Attempt to Derive Operator Splitting Method for Three Operators.

- **Operator Splitting**:
  To decompose two or more operators into simpler ones in the sense that they will be involved individually.
- **Goal**:To solve the following optimization problem:

$$\min_{\mathbf{x}} \sum_{i=1}^{n} f_i(\mathbf{x}), \quad n \geq 3$$

by operator splitting method. $\{f_i(\mathbf{x})\}_{i=1}^{n}$ are :

1. proper;
2. convex;
3. not necessarily smooth.

We assume that all of the functions we analyze in this report are of this kind without further notification.

Review, n=2

To solve:

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$$

- *Forward-backward splitting* method: if f is differentiable.

$$\mathbf{x}^{k+1} = \mathrm{prox}_{cg}(I - c\nabla f)\mathbf{x}^k$$

- *Peaceman-Rachford splitting* method: if neither of them are differentiable.

$$\mathbf{z}^{k+1} = \mathrm{refl}_{cg}\mathrm{refl}_{cf}\mathbf{z}^k$$
$$\mathbf{x}^{k+1} = \mathrm{prox}_{cf}\mathbf{z}^{k+1}$$

Review, n=2

- *Douglas-Rachford splitting* method.

$$\mathbf{z}^{k+1} = \lambda \mathrm{refl}_{cg} \mathrm{refl}_{cf} \mathbf{z}^k + (1-\lambda)\mathbf{z}^k$$
$$\mathbf{x}^{k+1} = \mathrm{prox}_{cf} \mathbf{z}^{k+1}$$

It's a damping version of *Peaceman-Rachford splitting* method. Convergence is guaranteed if $0 < \lambda < 1$.

From n=2 to n=3: Motivations

- We want to reduce the problem to two splitting.
- *Peaceman-Rachford splitting* for $\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$

$$\begin{cases} \mathbf{z}^{k+1} = \mathrm{refl}_{cf}(2\mathbf{x}^k - \mathbf{z}^k) \\ \mathbf{x}^{k+1} = \mathrm{prox}_{cg}\mathbf{z}^{k+1} \end{cases}$$

- The basic motivation:
  We drop one of the functions for the moment, and apply *Peaceman -Rachford splitting* to the remaining two functions in turns.

From n=2 to n=3

- By the above motivation, we have:

$$\begin{cases} \mathbf{z}_1^{k+1} = \mathrm{refl}_{cf}(2\mathbf{x}^k - \mathbf{z}_3^k) \\ \mathbf{x}^{k+\frac{1}{3}} = \mathrm{prox}_{cg}\mathbf{z}_1^{k+1} \end{cases}$$

$$\begin{cases} \mathbf{z}_2^{k+1} = \mathrm{refl}_{cg}(2\mathbf{x}^{k+\frac{1}{3}} - \mathbf{z}_1^{k+1}) \\ \mathbf{x}^{k+\frac{2}{3}} = \mathrm{prox}_{ch}\mathbf{z}_2^{k+1} \end{cases}$$

$$\begin{cases} \mathbf{z}_3^{k+1} = \mathrm{refl}_{ch}(2\mathbf{x}^{k+\frac{2}{3}} - \mathbf{z}_2^{k+1}) \\ \mathbf{x}^{k+1} = \mathrm{prox}_{cf}\mathbf{z}_3^{k+1} \end{cases}$$

Remarks: This is a *Gauss-Seidel* like method, which renews $\mathbf{x}$ as soon as it's updated. But the variables are coupled together. This routine inspires us for further derivation, but itself may not be convergent.

From n=2 to n=3

- We could also give a *Jacobi* version of the method:

$$\mathbf{z}_1^{k+1} = \mathrm{refl}_{cf}(2\mathbf{x}^k - \mathbf{z}_1^k)$$
$$\mathbf{z}_2^{k+1} = \mathrm{refl}_{cg}(2\mathbf{x}^k - \mathbf{z}_2^k)$$
$$\mathbf{z}_3^{k+1} = \mathrm{refl}_{ch}(2\mathbf{x}^k - \mathbf{z}_3^k)$$

  We have introduced three auxiliary variables $\{\mathbf{z}_i\}_{i=1}^3$ for different functions, so it's not fair to just use some of them in renewing $\mathbf{x}$. Therefore, the following "weighted average" may be a good idea:

$$\mathbf{x}^{k+1} = w_1\mathbf{z}_1 + w_2\mathbf{z}_2 + w_3\mathbf{z}_3$$

  where $\sum_{i=1}^3 w_i = 1$, $0 < w_i < 1$.

- Just like the *Douglas-Rachford splitting*, we could also give a damping version of the above algorithm:

$$
\begin{aligned}
\mathbf{z}_1^{k+1} &= \lambda(\mathrm{refl}_{cf}(2\mathbf{x}^k - \mathbf{z}_1^k)) + (1-\lambda)\mathbf{z}_1^k \\
\mathbf{z}_2^{k+1} &= \lambda(\mathrm{refl}_{cg}(2\mathbf{x}^k - \mathbf{z}_2^k)) + (1-\lambda)\mathbf{z}_1^k \\
\mathbf{z}_3^{k+1} &= \lambda(\mathrm{refl}_{ch}(2\mathbf{x}^k - \mathbf{z}_3^k)) + (1-\lambda)\mathbf{z}_1^k \\
\mathbf{x}^{k+1} &= w_1\mathbf{z}_1 + w_2\mathbf{z}_2 + w_3\mathbf{z}_3
\end{aligned}
$$

Remarks: The objective function is splitted into several parts, we attempt to operate on each sub-objective function, and then obtain the optimal by synthesizing and revising the solutions from the sub-problems. This algorithm is paralellizable.

From n=3 to m

- Based on the above algorithm, We could propose an algorithm for a more general case:

$$\min_{\mathbf{x}} \sum_{i=1}^{m} f_i(\mathbf{x})$$

- Since

$$\mathbf{x}^* = \mathrm{argmin}_{\mathbf{x}} \sum_{i=1}^{m} f_i(\mathbf{x}) \Leftrightarrow \mathbf{0} \in \sum_{i=1}^{m} \partial f_i(\mathbf{x}^*),$$

we could derive more equivalent forms of this problem.

From n=3 to m: Equivalent Transformation

- The above problem is equivalent to:

$$\exists\{\mathbf{y}_i\}_{i=1}^m, \text{s.t} \begin{cases} \mathbf{y}_i \in -\frac{\gamma}{w_i}\partial f_i(\mathbf{x}) & i = 1, ..., m \\ \sum_{i=1}^m w_i\mathbf{y}_i = \mathbf{0} \end{cases}$$

$$\Leftrightarrow \begin{cases} \mathbf{y}_i + \mathbf{x} \in -\frac{\gamma}{w_i}\partial f_i(\mathbf{x}) + \mathbf{x} & i = 1, ..., m \\ \sum_{i=1}^m w_i\mathbf{y}_i = \mathbf{0} \end{cases}$$

Let $\mathbf{z}_i = \mathbf{y}_i + \mathbf{x}$, then the above problem is equivalent to:

$$\begin{cases} w_i(\mathbf{x} - \mathbf{z}_i) \in \gamma\partial f_i(\mathbf{x}) & i = 1, ..., m \\ \sum_{i=1}^m w_i\mathbf{z}_i = \mathbf{x} \end{cases}$$

- Using the same trick as that in deriving *Peaceman-Rachford splitting*, we have:

$$
\begin{aligned}
& w_i(\mathbf{x} - \mathbf{z}_i) \in \gamma \partial f_i(\mathbf{x}) \\
\Leftrightarrow & 2\mathbf{x} - \mathbf{z}_i \in (I + \frac{\gamma}{w_i} f_i)\mathbf{x} \\
\Leftrightarrow & \mathbf{x} = (I + \frac{\gamma}{w_i} \partial f_i)^{-1}(2\mathbf{x} - \mathbf{z}_i) = \mathrm{prox}_{\frac{\gamma}{w_i} f_i}(2\mathbf{x} - \mathbf{z}_i) \\
\Leftrightarrow & 2\mathbf{x} - \mathbf{z}_i = 2\mathrm{prox}_{\frac{\gamma}{w_i} f_i}(2\mathbf{x} - \mathbf{z}_i) - \mathbf{z}_i \\
\Leftrightarrow & \mathbf{z}_i = \mathrm{refl}_{\frac{\gamma}{w_i} f_i}(2\mathbf{x} - \mathbf{z}_i)
\end{aligned}
$$

- Therefore,

$$
\mathbf{0} \in \sum_{i=1}^{m} \partial f_i(\mathbf{x}) \Leftrightarrow \begin{cases} \mathbf{z}_i = \mathrm{refl}_{\frac{\gamma}{w_i} f_i}(2\mathbf{x} - \mathbf{z}_i) \ i = 1, \ldots, m \\ \mathbf{x} = \sum_{i=1}^{m} w_i \mathbf{z}_i \end{cases}
$$

From n=3 to m: Damping Algorithm

- We could also split z_i to allow damping strategies in the algorithm, that is:

$$\begin{cases} \mathbf{z}_i = \lambda_i \mathrm{refl}_{\frac{\gamma}{w_i} f_i}(2\mathbf{x} - \mathbf{z}_i) + (1 - \lambda_i)\mathbf{z}_i \ \ i = 1, ..., m \\ \mathbf{x} = \sum_{i=1}^{m} w_i \mathbf{z}_i \end{cases}$$

- The form above indicates that $\mathbf{z}_i$ is the fixed point of the mapping $\lambda_i \mathrm{refl}_{\frac{\gamma}{w_i} f_i}(2\mathbf{x} - \bullet) + (1 - \lambda_i)\bullet$, which inspires us to design the iteration algorithm to find them:

$$\begin{cases} \mathbf{z}_i^{t+1} = \lambda_i^t \mathrm{refl}_{\frac{\gamma}{w_i} f_i}(2\mathbf{x}^t - \mathbf{z}_i^t) + (1 - \lambda_i^t)\mathbf{z}_i^t \ \ i = 1, ..., m \\ \mathbf{x}^{t+1} = \sum_{i=1}^{m} w_i \mathbf{z}_i^{t+1} \end{cases}$$

Abstract Descriptions

- For simplification, we denote $\lambda_i^t = \frac{1}{2}$.
- Since we have splitted the objective functions and introduced $\mathbf{z}_i$ to investigate the decreasing process on each subproblem, we attempt to describe the relationship between $\mathbf{z}_i$ and $\mathbf{x}$. Therefore, we introduce the product space for illustration.

Abstract Descriptions

- $\mathbf{x}, \mathbf{z}_i \in \mathcal{H}$, $\mathcal{H}^n = \prod_{i=1}^n \mathcal{H}$.
- The inner product in $\mathcal{H}$ is defined as:
  $\langle \mathbf{A}, \mathbf{B} \rangle_{\mathcal{H}^n} = \sum_{i=1}^n w_i \langle \mathbf{a}_i, \mathbf{b}_i \rangle_{\mathcal{H}}$,
  in which $\mathbf{A} = (\mathbf{a}_1, ..., \mathbf{a}_n)$, $\mathbf{B} = (\mathbf{b}_1, ..., \mathbf{b}_n)$.
- $\mathcal{S} = \{(\mathbf{x}_1, ..., \mathbf{x}_n) : \mathbf{x}_1 = ... = \mathbf{x}_n \in \mathcal{H}\}$ is a subspace of $\mathcal{H}^n$, which is isomorphic to $\mathcal{H}$, Hence, $\mathcal{S}$ is the subspace for solution.
- The algorithm is equivalent to:

$$\begin{cases} \mathbf{Z}^{t+1} = \frac{1}{2}(\mathbf{refl}_{\mu\mathbf{F}}(2\mathbf{X}^t - \mathbf{Z}^t) + \mathbf{Z}^t) \\ \mathbf{X}^{t+1} = (\sum_{i=1}^n w_i \mathbf{z}_i^{t+1}, ..., \sum_{i=1}^n w_i \mathbf{z}_i^{t+1}) = \langle \mathbf{I}, \mathbf{Z}^{t+1} \rangle_{\mathcal{H}^n} \mathbf{I} \end{cases}$$

  in which, $\mathbf{refl}_{\mu\mathbf{F}} = (\mathrm{refl}_{\frac{\gamma}{w_i} f_i})_i$, $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_n)$, $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$
  .
- $\mathbf{X}^t = \langle \mathbf{I}, \mathbf{Z}^t \rangle_{\mathcal{H}^n} \mathbf{I}$ is the projection onto $\mathcal{S}$.
  $2\mathbf{X}^t - \mathbf{Z}^t$ is the mirror image of $\mathbf{Z}^t$.

Abstract Descriptions

- $$\begin{cases} \mathbf{Z}^{t+1} = \frac{1}{2}(\mathbf{refl}_{\mu\mathbf{F}}(2\mathbf{X}^t - \mathbf{Z}^t) + \mathbf{Z}^t) \\ \mathbf{X}^{t+1} = \langle \mathbf{I}, \mathbf{Z}^{t+1} \rangle_{\mathcal{H}^n} \mathbf{I} \end{cases}$$

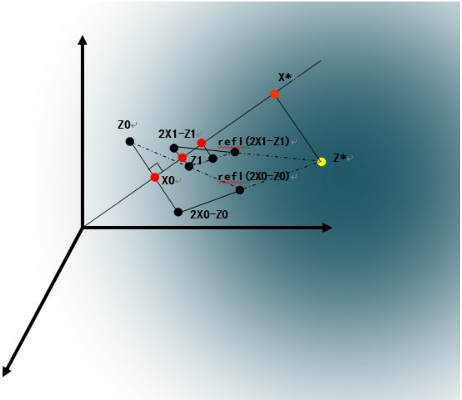

Figure: Algorithm

Convergence Discussion 1

Definition (*α-average*)

Let D be a nonempty subset of $\mathcal{H}$, let $T : D \to \mathcal{H}$ be nonexpansive, and let $\alpha \in (0, 1)$. Then T is *averaged* with constant α, or *α-averaged*, if there exists a nonexpansive operator $R : D \to \mathcal{H}$ such that $T = (1 - \alpha)\mathrm{Id}+\alpha R$.

Lemma

If $T : D \to \mathcal{H}$ is α-averaged with $\alpha \in (0, 0.5]$, then T is firmly nonexpansive.

Reference: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Heinz H. Bauschke, Patrick L. Combettes.

- By diminishing the variable $\mathbf{X}$, the iteration could be written as:

$$\mathbf{Z}^{t+1} = \frac{1}{2}(\mathbf{refl}_{\mu\mathbf{F}} R_{N_S} + I)\mathbf{Z}^t$$

where R_{N_s} is the mirror reflexion operator of $\mathcal{S}$. Since $\mathbf{refl}_{\mu\mathbf{F}} R_{N_S}$ is a nonexpansive operator, by definition, $\frac{1}{2}(\mathbf{refl}_{\mu\mathbf{F}} R_{N_S} + I)$ is $\frac{1}{2} -$ *averaged*, and therefore a firmly non-expansive operator.

- Hence, the algorithm above is convergent.

Numerical Experiments

Solve

$$\min_{\mathbf{x}} \sum_{i=1}^{3} \|\mathbf{A}_i\mathbf{x} - \mathbf{b}_i\|_1$$

Define $f_i(\mathbf{x}) = \|\mathbf{A}_i\mathbf{x} - \mathbf{b}_i\|_1, i = 1, 2, 3.$

- Relaxed-Douglas-Rachford Splitting Algorithm for ℓ_1 functions.
- Computing Prox
- Results

Require: $\mathbf{A}, \mathbf{z}, \mathbf{b}, \lambda_N, c, \{\omega_i\}_{i=1}^m$
Ensure: $\mathbf{x}$
  while $N < N_{max}$ **do**
    for all $i \in 1, \ldots, m$ **do**
      $\mathbf{z_i} \leftarrow \mathbf{z_i} + 2\lambda_N (Prox_{\{cf_i\}}(2\mathbf{x} - \mathbf{z}_i) - \mathbf{x})$
    end for
    $\mathbf{x} \leftarrow \sum_{i=1}^{m} \omega_i \mathbf{z}_i$
    $N \leftarrow N + 1$
  end while

where $\mathbf{z}_i, \mathbf{x} \in \mathbb{R}^{n \times 1}, \mathbf{z}_i$ denotes the i^{th} column of $\mathbf{z}$.

We can see at once that the main computation comes from calculating *Prox*. Therefore, we use both *ADM* and *CVX* to compute *Prox* and compare their results.

$$Prox_{cf_i}(\mathbf{y}) = \text{argmin}_{\mathbf{x}} f_i(\mathbf{x}) + \frac{1}{2c}\|\mathbf{x}-\mathbf{y}\|_2^2.$$

$$\begin{aligned} & \min \quad \|\mathbf{A}_i\mathbf{x}-\mathbf{b}_i\|_1 + \frac{1}{2c}\|\mathbf{x}-\mathbf{y}\|_2^2 \\ \Leftrightarrow \quad & \min \quad \|\mathbf{z}\|_1 + \frac{1}{2c}\|\mathbf{x}-\mathbf{y}\|_2^2 \quad s.t. \mathbf{A}_i\mathbf{x}-\mathbf{b}_i-\mathbf{z}=0 \end{aligned}$$

The augmented Lagrangian is:

$$\mathcal{L}_A(\mathbf{x},\mathbf{z},\mu) = \|\mathbf{z}\|_1 + \frac{1}{2c}\|\mathbf{x}-\mathbf{y}\|_2^2 + \mu^T(\mathbf{A}_i\mathbf{x}-\mathbf{b}_i-\mathbf{z}) + \frac{\beta}{2}\|\mathbf{A}_i\mathbf{x}-\mathbf{b}_i-\mathbf{z}\|_2^2$$

ADMM for computing Prox

$$
\begin{aligned}
\mathbf{x}^{k+1} &= \text{argmin}_{\mathbf{x}} \left(\frac{1}{2c}\|\mathbf{x}-\mathbf{y}\|_2^2 + {\mu^{\mathbf{k}}}^T(\mathbf{A}_i\mathbf{x}-\mathbf{b}_i) + \frac{\beta}{2}\|\mathbf{A}_i\mathbf{x}-\mathbf{b}_i-\mathbf{z}^k\|_2^2 \right) \\
&\Rightarrow \mathbf{x}^{k+1} = (\frac{1}{c}\mathbf{I} + \beta\mathbf{A}_i^T\mathbf{A}_i)^{-1}\left(\mathbf{A}_i^T(\beta\mathbf{b}_i+\beta\mathbf{z}-\mu^{\mathbf{k}}) + \frac{1}{c}\mathbf{y}\right) \\
\mathbf{z}^{k+1} &= \text{argmin}_{\mathbf{z}}\|\mathbf{z}\|_1 - {\mu^{\mathbf{k}}}^T\mathbf{z} + \frac{\beta}{2}\|\mathbf{A}_i\mathbf{x}^{k+1}-\mathbf{b}_i-\mathbf{z}\|_2^2 \\
&= \text{argmin}_{\mathbf{z}}\|\mathbf{z}\|_1 + \frac{\beta}{2}\|\mathbf{z}-(\frac{1}{\beta}\mu^{\mathbf{k}}+\mathbf{A}_i\mathbf{x}^{k+1}-\mathbf{b}_i)\|_2^2 \\
&\Rightarrow \mathbf{z} = \textit{shrink}(\frac{1}{\beta}\mu^{\mathbf{k}}+\mathbf{A}_i\mathbf{x}^{k+1}-\mathbf{b}_i, \frac{1}{\beta}) \\
\mu^{\mathbf{k+1}} &= \mu^{\mathbf{k}} + (\mathbf{A}_i\mathbf{x}^{k+1}-\mathbf{b}_i-\mathbf{z}^{k+1})
\end{aligned}
$$

Results

- Results of signal
- Compare of ADM and CVX
- More results

Results of signals

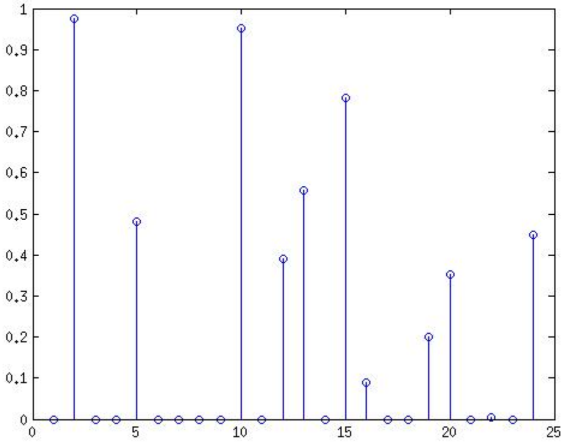

Figure: The Illustration of The Input Signal with ℓ_1 norm about 5.7

Results of signals

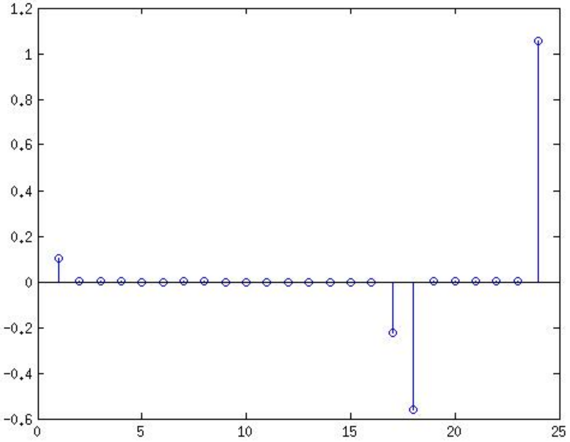

Figure: The Illustration of Signal We Got after the Alg., with ℓ_1 norm about 1.9.

ADM vs. CVX

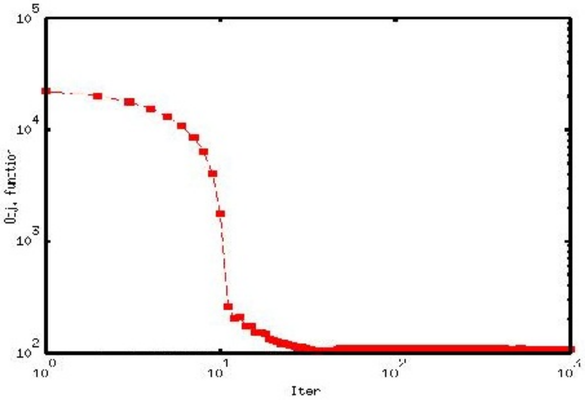

Figure: The illustration of ADM descending curve

ADM vs. CVX

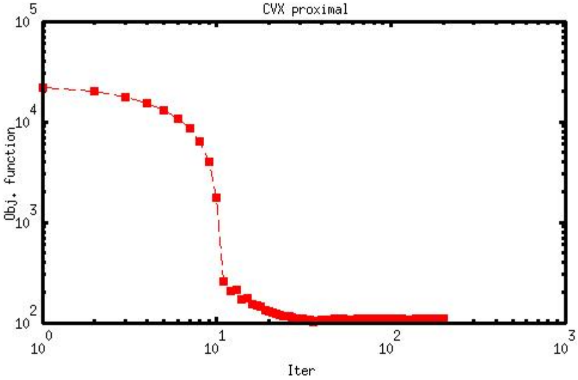

Figure: The illustration of CVX descending curve

ADM vs. CVX

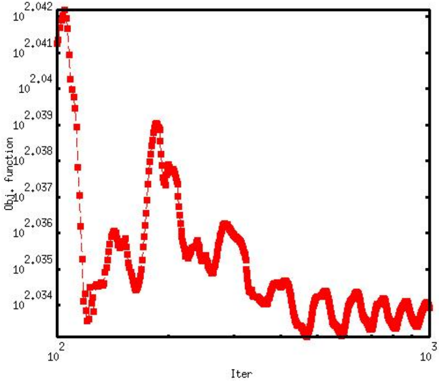

Figure: The illustration of ADM descending curve tail

ADM vs. CVX

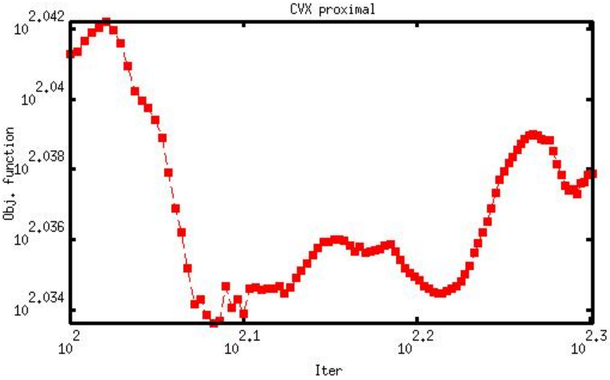

Figure: The illustration of CVX descending curve tail

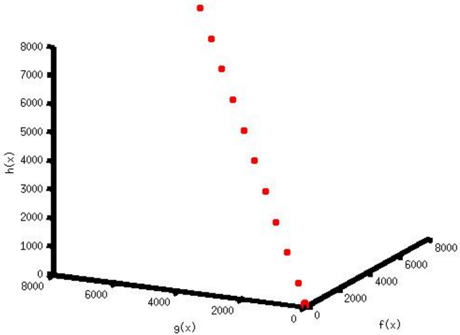

Figure: The illustration of ADM descending curve(respect to 3 functions)

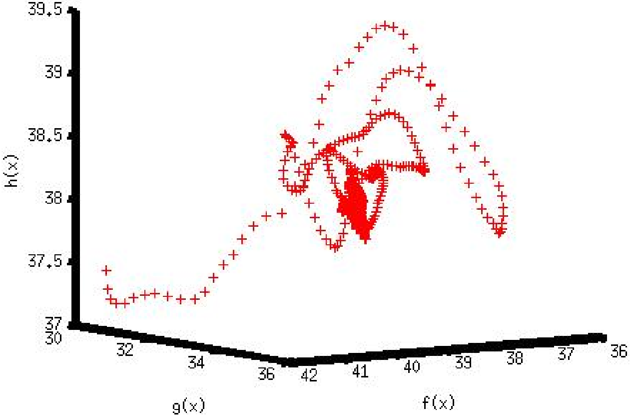

Figure: The illustration of ADM descending curve(respect to 3 functions tail)

ADM vs. CVX

Time complexity

- CVX takes 3 mins for 200 iteration
- ADM only takes 15s for 1000 iteration

From n=2 to n=3: Another Attempt

- Goal: $\text{minimize}_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) + h(\mathbf{x})$

$$\begin{aligned} \mathbf{x} &= \text{minimize}_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) + h(\mathbf{x}) \\ \Leftrightarrow \mathbf{0} &\in c(\partial f + \partial g + \partial h)\mathbf{x} \end{aligned}$$

We may firstly assume f and g are differentiable.

$$\begin{aligned} &\Leftrightarrow (I - c\nabla f - c\nabla g)\mathbf{x} \in (I + c\partial h)\mathbf{x} \\ &\Leftrightarrow \mathbf{x} = (I + c\partial h)^{-1}(I - c\nabla f - c\nabla g)\mathbf{x} \end{aligned}$$

By Peaceman-Rachford spltting, we can introduce $\mathbf{w}$ to simplify the problem:

$$\begin{cases} \mathbf{w} = \text{refl}_{c(f+g)}\text{refl}_{ch}\mathbf{w} = (2\text{prox}_{c(f+g)} - I)(2\mathbf{x} - \mathbf{w}) \\ \mathbf{x} = \text{prox}_{ch}\mathbf{w} \end{cases}$$

- Strategy to compute $\operatorname{prox}_{c(f+g)}(\mathbf{u})$.
  At the moment, we assume f is differentiable.

$$\operatorname{prox}_{c(f+g)}(\mathbf{u}) = \operatorname{argmin}_{\mathbf{z}} c(f(\mathbf{z}) + g(\mathbf{z})) + \frac{1}{2}\|\mathbf{z} - \mathbf{u}\|^2$$

  Equivalently, we should look for $\mathbf{z}$ that satisfies:

$$\mathbf{0} \in (c\nabla f + c\partial g)\mathbf{z} + \mathbf{z} - \mathbf{u}$$

  We introduce a new variable $\mathbf{v}$, s.t.$\mathbf{v} = \mathbf{z} - \mathbf{u}$.
  Denote

$$\begin{aligned} f^u(\mathbf{v}) &= f(\mathbf{u} + \mathbf{v}) \\ g^u(\mathbf{v}) &= g(\mathbf{u} + \mathbf{v}) \\ \tilde{f}^u(\mathbf{v}) &= f^u(\mathbf{v}) + \frac{1}{2c}\|\mathbf{v}\|^2. \end{aligned}$$

From n=2 to n=3: Another Attempt

- Therefore,

$$\mathbf{0} \in (c\nabla f + c\partial g)\mathbf{z} + \mathbf{z} - \mathbf{u}$$

is equivalent to:

$$\mathbf{0} \in (c\nabla \tilde{f}^u + c\partial g^u)\mathbf{v}$$

Applying Peaceman-Rachford splitting again, introducing $\mathbf{r}$, we have a equivalent form:

$$\begin{cases} \mathbf{r} = \mathrm{refl}_{cg^u}\mathrm{refl}_{c\tilde{f}^u}\mathbf{r} \\ \mathbf{v} = \mathrm{prox}_{c\tilde{f}^u}\mathbf{r} \\ \mathbf{z} = \mathrm{prox}_{c(f+g)}(\mathbf{u}) = \mathbf{u} + \mathbf{v} \end{cases}$$

From n=2 to n=3: Another Attempt–Algorithm

- Therefore, we could decouple $\mathrm{prox}_{c(f+g)}$. And here are the main formulas derived above:

$$\begin{cases} \mathbf{r} = \mathrm{refl}_{cg^u}\mathrm{refl}_{c\tilde{f}^u}\mathbf{r} \\ \mathbf{v} = \mathrm{prox}_{c\tilde{f}^u}\mathbf{r} \\ \mathbf{z} = \mathrm{prox}_{c(f+g)}(\mathbf{u}) = \mathbf{u} + \mathbf{v} \end{cases}$$

$$\begin{cases} \mathbf{w} = \mathrm{refl}_{c(f+g)}\mathrm{refl}_{ch}\mathbf{w} = (2\mathrm{prox}_{c(f+g)} - I)(2\mathbf{x} - \mathbf{w}) \\ \mathbf{x} = \mathrm{prox}_{ch}\mathbf{w} \end{cases}$$

- By combining the formulas above, we derive the following iterative process:

$$\begin{cases} \mathbf{r}^{k+1} & = \operatorname{refl}_{cg^{2x^k - w^k}} \operatorname{refl}_{c\tilde{f}^{2x^k - w^k}} \mathbf{r}^k \\ \mathbf{v}^{k+1} & = \operatorname{prox}_{c\tilde{f}^{2x^k - w^k}} \mathbf{r}^{k+1} \end{cases}$$

$$\mathbf{w}^{k+1} = 2\mathbf{z} - (2\mathbf{x}^k - \mathbf{w}^k) = 2\mathbf{v}^{k+1} + 2\mathbf{x}^k - \mathbf{w}^k$$
$$\mathbf{x}^{k+1} = \operatorname{prox}_{ch}(\mathbf{w}^{k+1})$$

References

- Patric L Combettes, Jean-Christophe Pesquet *A proximal decomposition method for solving convex variational inverse problems.*
- Hugo Raguet, Jalal Fadili, Gabriel Peyre, *Generalized Forward-Backward Splitting*.
- Heinz H. Bauschke, Patrick L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*.