Monotone Operator Splitting Methods

Stephen Boyd (with help from Neal Parikh and Eric Chu)

EE364b, Stanford University
Outline

1. Operator splitting
2. Douglas-Rachford splitting
3. Consensus optimization
Operator splitting

• want to solve \( 0 \in F(x) \) with \( F \) maximal monotone
• main idea: write \( F \) as \( F = A + B \), with \( A \) and \( B \) maximal monotone
• called operator splitting
• solve using methods that require evaluation of resolvents

\[
R_A = (I + \lambda A)^{-1}, \quad R_B = (I + \lambda B)^{-1}
\]

(or Cayley operators \( C_A = 2R_A - I \) and \( C_B = 2R_B - I \))

• useful when \( R_A \) and \( R_B \) can be evaluated more easily than \( R_F \)
Main result

- $A, B$ maximal monotone, so Cayley operators $C_A, C_B$ nonexpansive
- hence $C_A C_B$ nonexpansive
- key result:

\[ 0 \in A(x) + B(x) \iff C_A C_B(z) = z, \quad x = R_B(z) \]

- so solutions of $0 \in A(x) + B(x)$ can be found from fixed points of nonexpansive operator $C_A C_B$
Proof of main result

- write $C_A C_B(z) = z$ and $x = R_B(z)$ as

$$x = R_B(z), \quad \tilde{z} = 2x - z, \quad \tilde{x} = R_A(\tilde{z}), \quad z = 2\tilde{x} - \tilde{z}$$

- subtract 2nd & 4th equations to conclude $x = \tilde{x}$

- 4th equation is then $2x = \tilde{z} + z$

- now add $x + \lambda B(x) \ni z$ and $x + \lambda A(x) \ni \tilde{z}$ to get

$$2x + \lambda(A(x) + B(x)) \ni \tilde{z} + z = 2x$$

- hence $A(x) + B(x) \ni 0$

- argument goes other way (but we don’t need it)
Outline

1 Operator splitting

2 Douglas-Rachford splitting

3 Consensus optimization
Peaceman-Rachford and Douglas-Rachford splitting

• Peaceman-Rachford splitting is undamped iteration

\[ z^{k+1} = C_A C_B(z^k) \]

doesn’t converge in general case; need \( C_A \) or \( C_B \) to be contraction

• Douglas-Rachford splitting is damped iteration

\[ z^{k+1} := (1/2)(I + C_A C_B)(z^k) \]

always converges when \( 0 \in A(x) + B(x) \) has solution

• these methods trace back to the mid-1950s (!!)
Douglas-Rachford splitting

write D-R iteration $z^{k+1} := (1/2)(I + C_A C_B)(z^k)$ as

\[
\begin{align*}
    x^{k+1/2} & := R_B(z^k) \\
    \hat{z}^{k+1/2} & := 2x^{k+1/2} - z^k \\
    x^{k+1} & := R_A(\hat{z}^{k+1/2}) \\
    \bar{z}^{k+1} & := \hat{z}^k + x^{k+1} - x^{k+1/2}
\end{align*}
\]

last update follows from

\[
\begin{align*}
    \bar{z}^{k+1} & := (1/2)(2x^{k+1} - \hat{z}^{k+1/2}) + (1/2)\hat{z}^k \\
    & = x^{k+1} - (1/2)(2x^{k+1/2} - z^k) + (1/2)\hat{z}^k \\
    & = \bar{z}^k + x^{k+1} - x^{k+1/2}
\end{align*}
\]

- can consider $x^{k+1} - x^{k+1/2}$ as a residual
- $\bar{z}^k$ is running sum of residuals
Douglas-Rachford algorithm

- *many* ways to rewrite/rearrange D-R algorithm
- equivalent to many other algorithms; often not obvious
- need very little: \( A, B \) maximal monotone; solution exists
- \( A \) and \( B \) are handled separately (via \( R_A \) and \( R_B \)); they are ‘uncoupled’
Alternating direction method of multipliers

to minimize $f(x) + g(x)$, we solve $0 \in \partial f(x) + \partial g(x)$

with $A(x) = \partial g(x)$, $B(x) = \partial f(x)$, D-R is

\[
\begin{align*}
x^{k+1/2} & := \arg\min_x \left( f(x) + \frac{1}{2\lambda} \|x - z^k\|_2^2 \right) \\
z^{k+1/2} & := 2x^{k+1/2} - z^k \\
x^{k+1} & := \arg\min_x \left( g(x) + \frac{1}{2\lambda} \|x - z^{k+1/2}\|_2^2 \right) \\
z^{k+1} & := z^k + x^{k+1} - x^{k+1/2}
\end{align*}
\]

a special case of the alternating direction method of multipliers (ADMM)

Douglas-Rachford splitting
Constrained optimization

- constrained convex problem:

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad x \in C
\end{align*}
\]

- take \( B(x) = \partial f(x) \) and \( A(x) = \partial I_C(x) = N_C(x) \)
- so \( R_B(z) = \text{prox}_f(z) \) and \( R_A(z) = \Pi_C(z) \)
- D-R is

\[
\begin{align*}
x^{k+1/2} & := \text{prox}_f(z^k) \\
z^{k+1/2} & := 2x^{k+1/2} - z^k \\
x^{k+1} & := \Pi_C(z^{k+1/2}) \\
z^{k+1} & := z^k + x^{k+1} - x^{k+1/2}
\end{align*}
\]
Dykstra’s alternating projections

• find a point in the intersection of convex sets $C, D$
• D-R gives algorithm

\[
\begin{align*}
x^{k+1/2} &= \Pi_C(z^k) \\
z^{k+1/2} &= 2x^{k+1/2} - z^k \\
x^{k+1} &= \Pi_D(z^{k+1/2}) \\
z^{k+1} &= z^k + x^{k+1} - x^{k+1/2}
\end{align*}
\]

• this is Dykstra’s alternating projections algorithm
• much faster than classical alternating projections
  (e.g., for $C, D$ polyhedral, converges in finite number of steps)
Positive semidefinite matrix completion

- some entries of matrix in $\mathbf{S}^n$ known; find values for others so completed matrix is PSD
- $C = \mathbf{S}^n_+$, $D = \{ X \mid X_{ij} = X_{ij}^\text{known}, (i, j) \in \mathcal{K} \}$
- projection onto $C$: find eigendecomposition $X = \sum_{i=1}^{n} \lambda_i q_i q_i^T$; then
  $$
  \Pi_C(X) = \sum_{i=1}^{n} \max\{0, \lambda_i\} q_i q_i^T
  $$
- projection onto $D$: set specified entries to known values
Positive semidefinite matrix completion

specific example: $50 \times 50$ matrix missing about half of its entries

- initialize $Z^0 = 0$

Douglas-Rachford splitting
Positive semidefinite matrix completion

\[ \| X_{k+1} - X_{k+1/2} \|_F \]

iteration \( k \)

- **blue**: alternating projections; **red**: D-R
- \( X^{k+1/2} \in C, \ X^{k+1} \in D \)

Douglas-Rachford splitting
Outline

1. Operator splitting
2. Douglas-Rachford splitting
3. Consensus optimization
Consensus optimization

- want to minimize $\sum_{i=1}^{N} f_i(x)$
- rewrite as consensus problem

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{N} f_i(x_i) \\
\text{subject to} & \quad x \in C = \{(x_1, \ldots, x_N) \mid x_1 = \cdots = x_N\}
\end{align*}
\]

- D-R consensus optimization:

\[
\begin{align*}
x^{k+1/2} & := \text{prox}_f(z^k) \\
z^{k+1/2} & := 2x^{k+1/2} - z^k \\
x^{k+1} & := \Pi_C(z^{k+1/2}) \\
z^{k+1} & := z^k + x^{k+1} - x^{k+1/2}
\end{align*}
\]
Douglas-Rachford consensus

- $x^{k+1/2}$-update splits into $N$ separate (parallel) problems:
  \[ x^{k+1/2}_i := \arg \min_{z_i} (f_i(z_i) + (1/2\lambda)\|z_i - z^{k}_i\|_2^2), \quad i = 1, \ldots, N \]

- $x^{k+1}$-update is averaging:
  \[ x^{k+1}_i := \bar{z}^{k+1/2} = (1/N) \sum_{i=1}^{N} z^{k+1/2}_i, \quad i = 1, \ldots, N \]

- $z^{k+1}$-update becomes
  \[ z^{k+1}_i = z^{k}_i + \bar{z}^{k+1/2} - x^{k+1/2}_i \]
  \[ = z^{k}_i + 2\bar{x}^{k+1/2} - \bar{z}^{k} - x^{k+1/2}_i \]
  \[ = z^{k}_i + (\bar{x}^{k+1/2} - x^{k+1/2}_i) + (\bar{x}^{k+1/2} - \bar{z}^{k}) \]

- Taking average of last equation, we get $\bar{z}^{k+1} = \bar{x}^{k+1/2}$
Douglas-Rachford consensus

- renaming $x^{k+1/2}$ as $x^{k+1}$, D-R consensus becomes

$$
x_i^{k+1} := \text{prox}_{f_i}(z_i^k)
$$

$$
z_i^{k+1} := z_i^k + (\bar{x}_i^{k+1} - x_i^{k+1}) + (\bar{x}_i^{k+1} - \bar{x}_i^k)
$$

- subsystem (local) state: $\bar{x}, z_i, x_i$
- gather $x_i$'s to compute $\bar{x}$, which is then scattered
Distributed QP

• we use D-R consensus to solve QP

\[
\begin{align*}
\text{minimize} \quad & f(x) = \sum_{i=1}^{N} (1/2) \|A_i x - b_i\|_2^2 \\
\text{subject to} \quad & F_i x \leq g_i, \quad i = 1, \ldots, N
\end{align*}
\]

with variable \( x \in \mathbb{R}^n \)

• each of \( N \) processors will handle an objective term, block of constraints

• coordinate \( N \) QP solvers to solve big QP
Distributed QP

- D-R consensus algorithm is

\[
\begin{align*}
    x_i^{k+1} &:= \arg\min_{F_i x_i \leq g_i} \left( \frac{1}{2} \| A_i x_i - b_i \|^2_2 + \frac{1}{2\lambda} \| x_i - z_i^k \|^2_2 \right) \\
    z_i^{k+1} &:= z_i^k + (\overline{x}_i^{k+1} - x_i^{k+1}) + (\overline{x}_i^{k+1} - \overline{x}_i^k),
\end{align*}
\]

- first step is \( N \) parallel QP solves
- second step gives coordination, to solve large problem
- inequality constraint residual is \( 1^T (F \overline{x}_i^k - g)_+ \)
Distributed QP

example with $n = 100$ variables, $N = 10$ subsystems

Consensus optimization