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

$$R_A = (I + \lambda A)^{-1}, \qquad R_B = (I + \lambda B)^{-1}$$

  (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

$$\begin{aligned} x^{k+1/2} &:= R_B(z^k) \\ z^{k+1/2} &:= 2x^{k+1/2} - z^k \\ x^{k+1} &:= R_A(z^{k+1/2}) \\ z^{k+1} &:= z^k + x^{k+1} - x^{k+1/2} \end{aligned}$$

last update follows from

$$\begin{aligned} z^{k+1} &:= (1/2)(2x^{k+1} - z^{k+1/2}) + (1/2)z^k \\ &= x^{k+1} - (1/2)(2x^{k+1/2} - z^k) + (1/2)z^k \\ &= z^k + x^{k+1} - x^{k+1/2} \end{aligned}$$

- can consider $x^{k+1} - x^{k+1/2}$ as a residual
- 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

$$\begin{aligned} x^{k+1/2} &:= \operatorname*{argmin}_x \left(f(x) + (1/2\lambda)\|x - z^k\|_2^2\right) \\ z^{k+1/2} &:= 2x^{k+1/2} - z^k \\ x^{k+1} &:= \operatorname*{argmin}_x \left(g(x) + (1/2\lambda)\|x - z^{k+1/2}\|_2^2\right) \\ z^{k+1} &:= z^k + x^{k+1} - x^{k+1/2} \end{aligned}$$

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

Constrained optimization

- constrained convex problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

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

$$\begin{aligned} x^{k+1/2} &:= \mathbf{prox}_f(z^k) \\ z^{k+1/2} &:= 2x^{k+1/2} - z^k \\ x^{k+1} &:= \Pi_C(z^{k+1/2}) \\ z^{k+1} &:= z^k + x^{k+1} - x^{k+1/2} \end{aligned}$$

Dykstra's alternating projections

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

$$\begin{aligned}
x^{k+1/2} &:= \Pi_C(z^k) \\
z^{k+1/2} &:= 2x^{k+1/2} - z^k \\
x^{k+1} &:= \Pi_D(z^{k+1/2}) \\
z^{k+1} &:= z^k + x^{k+1} - x^{k+1/2}
\end{aligned}$$

- 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

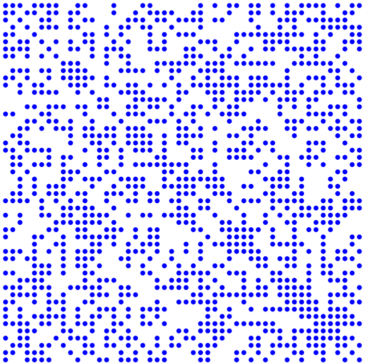
- some entries of matrix in $\mathbf{S}^n$ known; find values for others so completed matrix is PSD
- $C = \mathbf{S}^n_+$, $D = \{X \mid X_{ij} = X_{ij}^{\mathrm{known}},\ (i,j) \in \mathcal{K}\}$
- projection onto C: find eigendecomposition $X = \sum_{i=1}^n \lambda_i q_i q_i^T$; then

$$\Pi_C(X) = \sum_{i=1}^n \max\{0, \lambda_i\} q_i q_i^T$$

- projection onto D: set specified entries to known values

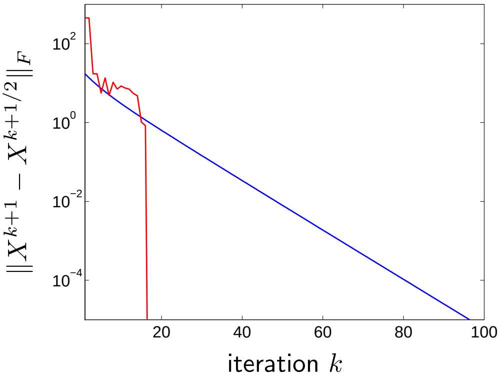
Positive semidefinite matrix completion

specific example: 50×50 matrix missing about half of its entries



- initialize $Z^0 = 0$

Positive semidefinite matrix completion



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

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*

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} & x \in C = \{(x_1, \ldots, x_N) \mid x_1 = \cdots = x_N\} \end{array}$$

- D-R consensus optimization:

$$\begin{aligned} x^{k+1/2} &:= \mathbf{prox}_f(z^k) \\ z^{k+1/2} &:= 2x^{k+1/2} - z^k \\ x^{k+1} &:= \Pi_C(z^{k+1/2}) \\ z^{k+1} &:= z^k + x^{k+1} - x^{k+1/2} \end{aligned}$$

Douglas-Rachford consensus

- $x^{k+1/2}$-update splits into N separate (parallel) problems:

$$x_i^{k+1/2} \quad := \quad \operatorname*{argmin}_{z_i} \left(f_i(z_i) + (1/2\lambda)\|z_i - z_i^k\|_2^2 \right), \quad i = 1, \ldots, N$$

- x^{k+1}-update is averaging:

$$x_i^{k+1} \quad := \quad \overline{z}^{k+1/2} = (1/N) \sum_{i=1}^{N} z_i^{k+1/2}, \quad i = 1, \ldots, N$$

- z^{k+1}-update becomes

$$\begin{aligned} z_i^{k+1} &= z_i^k + \overline{z}^{k+1/2} - x_i^{k+1/2} \\ &= z_i^k + 2\overline{x}^{k+1/2} - \overline{z}^k - x_i^{k+1/2} \\ &= z_i^k + (\overline{x}^{k+1/2} - x_i^{k+1/2}) + (\overline{x}^{k+1/2} - \overline{z}^k) \end{aligned}$$

- taking average of last equation, we get $\overline{z}^{k+1} = \overline{x}^{k+1/2}$

Douglas-Rachford consensus

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

$$\begin{aligned} x_i^{k+1} &:= \mathbf{prox}_{f_i}(z_i^k) \\ z_i^{k+1} &:= z_i^k + (\overline{x}^{k+1} - x_i^{k+1}) + (\overline{x}^{k+1} - \overline{x}^k) \end{aligned}$$

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

Distributed QP

- we use D-R consensus to solve QP

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

  with variable $x \in \mathbf{R}^n$

- 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

$$\begin{aligned} x_i^{k+1} &:= \underset{F_i x_i \le g_i}{\operatorname{argmin}} \left((1/2)\|A_i x_i - b_i\|_2^2 + (1/2\lambda)\|x_i - z_i^k\|_2^2 \right) \\ z_i^{k+1} &:= z_i^k + (\overline{x}^{k+1} - x_i^{k+1}) + (\overline{x}^{k+1} - \overline{x}^k), \end{aligned}$$

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

Distributed QP

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