Operator Splitting for Parallel and Distributed Optimization

Wotao Yin

(UCLA Math)

Shanghai Tech, SSDS'15 — June 23, 2015

URL: alturl.com/2z7tv

What is “splitting”?

- Sun-Tzu: “远交近攻”，“各个击破” (400 BC)
- Caesar: “divide-n-conquer” (100–44 BC)
- splitting in computing:
  - **break** a problem → separate parts
  - **solve** the separate parts → sub-solutions
  - **combine** the sub-solutions **in a controlled fashion**

Some basic principles of splitting

- split x/y directions
- split convection from diffusion in differential equations
- split linear from nonlinear
- domain decomposition
- Bender's decomposition, column generation
- split smooth from nonsmooth
- split objective functions and constraints in optimization
- split composite operators

Monotone operator-splitting pipeline

1. **recognize the simple parts** in your problem
2. **build an equivalent monotone-inclusion problem**: $0 \in Ax$
   (the simple parts are separately placed in A)
3. **apply an operator-splitting scheme**: $0 \in Ax \Longrightarrow z = Tz$
   (the simple parts become sub-operators of T)
4. run the Krasnosel'skiĭ–Mann (KM) iteration

$$z^{k+1} = z^k + \lambda(Tz^k - z^k), \quad \lambda \in (0, 1]$$

Example: LASSO (basis pursuit denoising)

- Tibshirani'96:

$$\text{minimize} \ \ \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1$$

  2 simple parts: smooth + simple

  - smooth function: $f(x) = \frac{1}{2}\|Ax - b\|^2$
  - simple nonsmooth function: $r(x) = \lambda\|x\|_1$

- **equivalent condition**: $0 \in \nabla f(x) + \partial r(x)$

- **forward-backward splitting** algorithm:

$$x^{k+1} = \underbrace{\mathbf{prox}_{\gamma r}(x^k - \gamma \nabla f(x^k))}_{Tx^k}$$

  also known as the Iterative Soft-Thresholding Algorithm (ISTA)

Example: total variation (TV) deblurring

- **Rudin-Osher-Fatemi'92:**

$$\begin{aligned} \underset{u}{\text{minimize}} \;\; & \frac{1}{2}\|Ku - b\|^2 + \lambda \|Du\|_1 \\ \text{subject to } & 0 \le u \le 255 \end{aligned}$$

4 simple parts: smooth + simple ∘ linear + simple

- smooth function: $f(u) = \frac{1}{2}\|Ku - b\|^2$
- linear operator: D
- simple nonsmooth function: $r_1 = \lambda \|\cdot\|_1$
- simple indicator function: $r_2 = \iota_{[0,255]}$

(We only show the results, skipping the details)

- **equivalent condition**:

$$0 \in \begin{bmatrix} \partial r_2 & D^* \\ -D & \partial r_1^* \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} + \begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix}$$

  (where w is the auxiliary (dual) variable)

- **forward-backward splitting** algorithm under a **special metric**:

$$\begin{aligned} u^{k+1} &= \mathbf{proj}_{[0,255]^n}(u^k - \gamma D^* w^k - \gamma \nabla f(u^k)) \\ w^{k+1} &= \mathbf{prox}_{\gamma \ell_\infty}(w^k + \gamma D(2u^{k+1} - u^k)) \end{aligned}$$

  every step is simple to implement

Simple parts

Simple parts

- linear maps (e.g., matrices, finite differences, orthogonal transforms)
- differentiable functions
- (non-differentiable) functions that have simple **proximal maps**
- sets that are easy to project to

Abstraction: we look for **monotone maps** which have certain simple operators

Monotone map

- $A : \mathcal{H} \to \mathcal{H}$ is **monotone** if

$$\langle Ax - Ay, x - y \rangle \geq 0, \quad \forall x, y \in \mathcal{H}$$

- extend to **set-valued** $A : \mathcal{H} \to 2^{\mathcal{H}}$:

$$\langle p - q, x - y \rangle \geq 0, \quad \forall x, y \in \mathcal{H},\ p \in Ax,\ q \in Ay$$

- **examples:**
  - a positive semi-definite linear map
  - a skew-symmetric linear map: $\langle Ax, x \rangle = 0, \quad \forall x \in \mathcal{H}$
  - ∂f: subdifferential of a proper closed convex function f

Forward operator

- **require**: A is **monotone** and either **Lipschitz** or **cocoercive**[1]
- **definition**: $\mathbf{fwd}_{\gamma A} := (I - \gamma A)$
- **examples**:
  - forward Euler for $\dot{y} + g(t, y) = 0$:

$$y^{t+1} = y^t - h \cdot g(t, y^t) = (I - h \cdot g(t, \cdot))y^t$$

  - gradient descent for $\min f(x)$:

$$x^{k+1} = x^k - \gamma \nabla f(x^k) = (I - \gamma \nabla f)x^k$$

- **equivalent conditions:** $0 \in Ax \iff x = (I - \gamma A)x$

[1] $\langle Ax - Ay, x - y \rangle \geq \beta \|Ax - Ay\|^2, \ \forall x, y \in \mathcal{H}$. If A is β-cocoercive, then A is $1/\beta$-Lipschitz. The reverse is generally untrue.

Backward operator

- **require**: A is monotone
- **definition**: $J_{\gamma A} := (I + \gamma A)^{-1}$
- **equivalent conditions:**: $0 \in Ax \;\Leftrightarrow\; x \in x + \gamma Ax \;\Leftrightarrow\; x = J_{\gamma A}x$

  even if A is set-valued, $J_{\gamma A}$ is single-valued
- **examples**:
  - regularized matrix inversion
  - backward Euler
  - **proximal map**, including projection as a special case

Proximal map

- **require**: a proper closed convex function f
- **definition**:
$$\mathbf{prox}_{\gamma f}(y) = \arg\min_{x} \; f(x) + \frac{1}{2\gamma}\|x - y\|^2$$
- the minimizer must satisfy:
$$0 \in \gamma \partial f(x^*) + (x^* - y) \iff x^* = (I + \gamma \partial f)^{-1}(y) = J_{\gamma \partial f}(y)$$
therefore, $\mathbf{prox}_{\gamma f} \equiv J_{\gamma \partial f} \equiv (I + \gamma \partial f)^{-1}$
- **proximal-point algorithm** (PPA):
$$x^{k+1} = \mathbf{prox}_{\gamma f}(x^k)$$
converges a minimizer of f, if it exists

Reflective backward operator

- **require**: A is monotone
- **definition**: $R_{\gamma A} := 2J_{\gamma A} - I$

  ““reflects” x through $J_{\gamma A}x$ by adding $J_{\gamma A}x - x$
- **examples**:
  - “mirror” or reflective projection: $\mathbf{refl}_C = 2\mathbf{proj}_C - I$
  - reflective proximal map: for closed proper convex function f

$$R_{\gamma \partial f} = 2J_{\gamma \partial f} - I = 2\mathbf{prox}_{\gamma f} - I$$

Operator splitting

Monotone inclusion

- $A_1, \ldots, A_m$ are monotone, either single- or set-valued, $m \geq 1$
- **operator-splitting** solves

$$\boxed{\mathbf{0} \in A_1 x + \cdots + A_m x}$$

by constructing an operator $T_{A_1,\ldots,A_m} : \mathcal{H} \to \mathcal{H}$, based on the simple operators of $A_1, \ldots, A_m$ and running the iteration

$$z^{k+1} = T_{A_1,\ldots,A_m}(z^k)$$

The "big three" operator-splitting schemes

$$0 \in Ax + Bx$$

- **Douglas-Rachford** (Lion-Mercier'79) for
  (maximally monotone) + (maximally monotone)
- **forward-backward** (Mercier'79) for
  (maximally monotone) + (cocoercive)
- **forward-backward-forward** (Tseng'00) for
  (maximally monotone) + (Lipschitz & monotone)
- all the schemes are built from **forward operators** and **backward operators**
- the first two have been reinvented many times
  (in some cases, the reduction not obvious and gone unnoticed)

Forward-backward splitting

- **require:** A maximally monotone, B cocoercive (thus single-valued)
- **forward-backward splitting (FBS) operator** (Lion-Mercier'79)

$$T_{\mathrm{FBS}} := J_{\gamma A} \circ (I - \gamma B)$$

- reduces to forward operator if $A = 0$, and backward operator if $B = 0$
- **equivalent conditions:**

$$0 \in Ax + Bx \iff x = T_{\mathrm{FBS}}(x)$$

- **backward-forward splitting (BFS)** operator $T_{\mathrm{BFS}} := (I - \gamma B) \circ J_{\gamma A}$, then

$$0 \in Ax + Bx \iff z = T_{\mathrm{BFS}}(z),\ x = J_{\gamma A} z$$

Douglas-Rachford splitting

- **require:** A, B both monotone
- **Douglas-Rachford splitting (DRS)** operator (Lion-Mercier'79)

$$T_{\mathrm{DRS}} := \frac{1}{2}I + \frac{1}{2}R_{\gamma A} \circ R_{\gamma B}$$

note: switching A and B gives a different DRS operator

- (relaxed) **Peaceman-Rachford splitting (PRS)** operator, $\lambda \in (0, 1]$:

$$T_{\mathrm{PRS}}^{\lambda} := (1 - \lambda)I + \lambda R_{\gamma A} \circ R_{\gamma B}$$

also, $T_{\mathrm{PRS}} = T_{\mathrm{PRS}}^{1}$

- **equivalent conditions**:

$$0 \in Ax + Bx \iff z = T_{\mathrm{PRS}}^{\lambda}(z), \; x = J_{\gamma B} z$$

Forward-backward-forward splitting

- **require:** A maximally monotone, B monotone and β-Lipschitz, $\beta > 0$
- useful when B is Lipschitz but not cocoercive (e.g., skew symmetric, convex combination of operators)
- **forward-backward-forward splitting (FBFS)** operator (Tseng'00)

$$T_{\mathrm{FBFS}} := I + (I - \gamma B) \circ J_{\gamma A} \circ (I - \gamma B) - (I - \gamma B)$$

- reduces to the backward operator if $B = 0$, and to $I - \gamma B \circ (I - \gamma B)$ if $A = 0$
- **equivalent conditions**: $\gamma \in (0, 1/\beta)$

$$0 \in Ax + Bx \iff x = T_{\mathrm{FBFS}}(x)$$

A three-operator splitting scheme

- **require:** A, B maximally monotone, C cocoercive
- Davis and Yin'15:

$$T_{\mathrm{DYS}} := I - J_{\gamma B} + J_{\gamma A} \circ (2J_{\gamma B} - I - \gamma C \circ J_{\gamma B})$$

(evaluating $T_3 z$ will evaluate $J_{\gamma A}$, $J_{\gamma B}$, and C only once each)

- reduces to BFS if $A = 0$, FBS if $B = 0$, and to DRS if $C = 0$
- **equivalent conditions**:

$$0 \in Ax + Bx + Cx \iff z = T_3(z),\ x = J_{\gamma B} z$$

Abstraction: KM (Krasnosel'skiĭ–Mann) iteration

- **require:** T is nonexpansive (1-Lipschitz)
- choose $\lambda > 0$ so that $T_\lambda = (1-\lambda) + \lambda T$ is an **averaged operator**
- **KM iteration**:
$$z^{k+1} = T_\lambda z^k$$
- **special cases**: $J_{\gamma A}$, $(I - \gamma A)$, T_{FBS}, T_{BFS}, T_{DRS}, T_{PRS}^{λ}, and T_{DYS}

  (the step-size of any cocoercive map therein must be bounded)
- **convergence**: if $\text{Fix}T \neq \emptyset$, then $z^k \rightharpoonup z^* \in \text{Fix}T$.
- **divergence**: if $\text{Fix}T = \emptyset$, then $(z^k)_{k\geq 0}$ goes unbounded.

Operator splitting:
Direct application

Regularization least squares

$$\underset{x}{\text{minimize}}\, r(x) + \underbrace{\frac{1}{2}\|Kx - b\|^2}_{f(x)}$$

- K: linear operator
- b: input data (observation)
- r: enforces a structure on x.

  examples: ℓ_2^2, ℓ_1, sorted ℓ_1, ℓ_2, TV, nuclear norm, ...
- **equivalent condition**: $0 \in \partial r(x) + \nabla f(x)$
- **forward-backward splitting** iteration:

$$x^{k+1} = \mathbf{prox}_{\gamma r} \circ (I - \gamma \nabla f) x^k$$

Constrained minimization

- C is a convex set. f is a proper close convex function.

$$\underset{x}{\text{minimize}} \;\; f(x)$$
$$\text{subject to } x \in C$$

- **equivalent condition**:

$$0 \in N_C(x) + \partial f(x)$$

- **if** f is Lipschitz differentiable, then apply **forward-backward splitting**

$$x^{k+1} = \mathbf{proj}_C \circ (I - \gamma \nabla f) x^k$$

recovers the **projected gradient method**

- **if** f is non-differentiable, then apply **Douglas-Rachford splitting (DRS)**

$$z^{k+1} = \Big(\frac{1}{2}I + \frac{1}{2}(2\mathbf{prox}_{\gamma f} - I) \circ (2\mathbf{proj}_C - I)\Big)z^k$$

(where $x^k = \mathbf{proj}_C z^k$)

- **dual approach**: introduce $x - y = 0$ and apply **ADMM** to

$$\begin{aligned} \underset{x,y}{\text{minimize}} \;\; & f(x) + \iota_C(y) \\ \text{subject to } & x - y = 0. \end{aligned}$$

(indicator function $\iota_C(y) = 0$, if $y \in C$, and ∞ otherwise.)

- **equivalence**: the ADMM iteration = the DRS iteration

Multi-function minimization

- $f_1, \ldots, f_m : \mathcal{H} \to (-\infty, \infty]$ are proper closed convex functions.

$$\underset{x}{\text{minimize}}\, f_1(x) + \cdots + f_N(x)$$

- **product-space trick**:
  - introduce copies $x_{(i)} \in \mathcal{H}$ of x; let $\mathbf{x} = (x_{(1)}, \ldots, x_{(N)}) \in \mathcal{H}^N$
  - let $\mathcal{C} = \{\mathbf{x} : x_{(1)} = \cdots = x_{(N)}\}$
  - equivalent problem in $\mathcal{H}^N$:

$$\underset{\mathbf{x}}{\text{minimize}}\ \iota_C(\mathbf{x}) + \sum_{i=1}^{N} f_i(x_{(i)})$$

then apply two-operator splitting scheme

Operator splitting:

Dual application

Duality

- convex (and some nonconvex) optimization problems have two perspectives: the primal problem and the dual problem
- **duality** brings us:
  - alternative or relaxed problems, lower bounds
  - certificates for optimality or infeasibility
  - economic interpretations
- **duality + operator splitting**:
  - decouples objective functions and constraints' components
  - gives rise to parallel and distributed algorithms

Lagrange duality

- original problem:

$$\underset{x}{\text{minimize}}\ f(x) \quad \text{subject to } Ax = b.$$

- relaxations:

$$\begin{aligned} \text{Lagrangian:} \quad & L(x; w) && := f(x) + w^T(Ax - b) \\ \text{augmented Lagrangian:} \quad & L(x; w, \gamma) && := f(x) + w^T(Ax - b) + \frac{\gamma}{2}\|Ax - b\|^2 \end{aligned}$$

- dual function:

$$d(w) = -\min_x L(x; w)$$

(d is always convex, even if f is not)

- dual problem:

$$\underset{w}{\text{minimize}}\ d(w)$$

Monotropic programs

- **definition**:

$$\underset{x_1,\ldots,x_m}{\text{minimize}} \;\; f_1(x_1) + \cdots + f_m(x_m)$$
$$\text{subject to } A_1x_1 + \cdots A_mx_m = b.$$

  where $f_i(x_i)$ may include $\iota_{C_i}(x)$ for constraint $x_i \in C_i$

- $x_1, \ldots, x_m$ are **separable in the objective** and **coupled in the constraints**

- dual problem has the form

$$\underset{w}{\text{minimize}} \; d_1(w) + \cdots + d_m(w)$$

  where $d_i(w) := -\min_{x_i} f_i(x_i) + w^T(A_ix_i - \frac{1}{m}b)$

Examples of monotropic programs

- linear programs
- $\min\{f(x) : Ax \in C\} \Leftrightarrow \min_{x,y}\{f(x) + \iota_C(y) : Ax - y = 0\}$
- consensus problem $\min\{f_1(x_1) + \cdots + f_n(x_n) : A\mathbf{x} = 0\}$, where
  - $Ax = 0 \Leftrightarrow x_1 = \cdots = x_m$
  - the structure of A enables distributed computing
- exchange problem

Dual (Lagrangian) decomposition

$$\underset{x_1,\ldots,x_m}{\text{minimize}} \;\; f_1(x_1) + \cdots + f_m(x_m)$$

$$\text{subject to } A_1 x_1 + \cdots + A_m x_m = b.$$

- the variables $x_1, \ldots, x_m$ are decoupled in the Lagrangian

$$L(x_1, \ldots, x_n; w) = \sum_{i=1}^{m} f_i(x_i) + w^T (A_i x_i - \frac{1}{m} b)$$

(but *not* so in the augmented Lagrangian for $\frac{\gamma}{2}\|A_1 x_1 + \cdots + A_m x_m - b\|^2$)

- let $\mathbf{A} = [A_1 \ \cdots \ A_m]$ and $\mathbf{x} = [x_1; \ldots; x_m]$
- the **dual gradient iteration**

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}'} L(\mathbf{x}'; w^k)$$
$$w^{k+1} = w - \gamma(b - \mathbf{A}\mathbf{x}^{k+1})$$

- the first step decouples to m separate subproblems

$$x_i^{k+1} = \arg\min_{x_i'} f_i(x_i') + w^{kT}(A_i x_i' - \frac{1}{m}b), \quad i = 1, \ldots, m$$

- this decomposition **requires strongly convex** $f_1, \ldots, f_m$ or, equivalently, Lipschitz differentiable $d_1, \ldots, d_m$

  (dual PPA doesn't have this requirement, but the first step doesn't decouple either)

Dual forward-backward splitting

- original problem:

$$\underset{x}{\text{minimize}}\ f_1(x) + f_2(y) \quad \text{subject to } A_1x_1 + A_2x_2 = b.$$

- **require**: strongly convex f_1 (thus Lipschitz ∇d_1)
- FBS iteration: $z^{k+1} = \mathbf{prox}_{\gamma d_2}(I - \gamma \nabla d_1)z^k$
- express in terms of (augmented) Lagrangian:

$$\begin{aligned}
x_1^{k+1} &= \underset{x_1'}{\arg\min}\ f_1(x_1') + w^{kT} A_1 x_1' \\
x_2^{k+1} &\in \underset{x_2'}{\arg\min}\ f_2(x_2') + w^{kT} A_2 x_2' + \frac{\gamma}{2}\|A_1 x_1^{k+1} + A_2 x_2' - b\|^2 \\
w^{k+1} &= w - \gamma(b - A_1 x_1^{k+1} - A_2 x_2^{k+1})
\end{aligned}$$

we have recovered Tseng's "Alternating Minimization Algorithm"

Dual Douglas-Rachford splitting

- original problem:

$$\underset{x}{\text{minimize}}\ f_1(x) + f_2(y) \quad \text{subject to } A_1 x_1 + A_2 x_2 = b.$$

- no strong-convexity requirement
- DRS iteration: $z^{k+1} = \left(\frac{1}{2}I + \frac{1}{2}(2\mathbf{prox}_{\gamma d_2} - I)(2\mathbf{prox}_{\gamma d_1} - I)\right)z^k$
- express in terms of augmented Lagrangian:

$$x_1^{k+1} \in \underset{x_1'}{\arg\min}\, f_1(x_1') + w^{kT} A_1 x_1' + \frac{\gamma}{2}\|A_1 x_1' + A_2 x_2^k - b\|^2$$

$$x_2^{k+1} \in \underset{x_2'}{\arg\min}\, f_2(x_2') + w^{kT} A_2 x_2' + \frac{\gamma}{2}\|A_1 x_1^{k+1} + A_2 x_2' - b\|^2$$

$$w^{k+1} = w^k - \gamma(b - A_1 x_1^{k+1} - A_2 x_2^{k+1})$$

recover **the Alternating Direction Method of Multipliers (ADMM)**

Dual Davis-Yin splitting

- original problem, $m \geq 3$:

$$\underset{x}{\text{minimize}}\ f_1(x_1) + \cdots + f_m(x_m) \quad \text{subject to } A_1 x_1 + \cdots + A_m x_m = b.$$

- **require**: strongly convex $f_1, \ldots, f_{m-2}$
- Davis-Yin'15 iteration: $z^{k+1} = T_3 z^k$ for $(d_1 + \cdots + d_{m-2}) + d_{m-1} + d_m$
- express in terms of (augmented) Lagrangian:

$$x_i^{k+1} = \underset{x_i'}{\arg\min}\, f_1(x_i') + w^{kT} A_i x_i', \quad i = 1, \ldots m-2, \text{ independently}$$

$$x_{m-1}^{k+1} \in \underset{x_j',\ j=m-1}{\arg\min}\ f_j(x_j') + w^{kT} A_j x_j' + \frac{\gamma}{2} \Big\| \sum_{i=1}^{m-2} A_i x_i^{k+1} + A_j x_j' + A_m x_m^k - b \Big\|^2$$

$$x_m^{k+1} \in \underset{x_m'}{\arg\min}\, f_m(x_m') + w^{kT} A_m x_m' + \frac{\gamma}{2} \Big\| \sum_{i=1}^{m-1} A_i x_i^{k+1} + A_m x_m' - b \Big\|^2$$

$$w^{k+1} = w^k - \gamma \Big(b - \sum_{i=1}^{m} A_i x_i^{k+1} \Big)$$

Dual operator splitting summary

- for problems with **separable objective** and **coupling linear constraints**
- each iteration: **separate f_i subproblems** + **multiplier update**
- **Lagrangian x_i-subproblems** require strongly convex f_i and can be solved in parallel
- **augmented Lagrangian x_i-subproblems** does not have the strong-convexity requirement but are solved in sequence

Operator splitting:

Primal-dual application

Nonsmooth ∘ linear composition

- **problem:** minimize nonsmooth ∘ linear + nonsmooth + smooth

$$\underset{x}{\text{minimize}}\; r_1(Lx) + r_2(x) + f(x)$$

- **equivalent condition:**

$$0 \in (L^T \circ \partial r_1 \circ L + \partial r_2 + \nabla f)x$$

- **decouple ∂r_1 from L:** introduce

$$\text{dual variable} \quad y \in \partial r_1 \circ Lx \iff Lx \in \partial r_1^*(y)$$

- **equivalent condition**:

$$0 \in \begin{bmatrix} 0 & L^T \\ -L & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \partial r_2(x) \\ \partial r_1^*(y) \end{bmatrix} + \begin{bmatrix} \nabla f(x) \\ 0 \end{bmatrix}$$

- **equivalent condition** (copied from last slide):

$$0 \in \underbrace{\begin{bmatrix} 0 & L^T \\ -L & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \partial r_2(x) \\ \partial r_1^*(y) \end{bmatrix}}_{Az} + \underbrace{\begin{bmatrix} \nabla f(x) \\ 0 \end{bmatrix}}_{Bz}$$

- **primal-dual variable**: $z = \begin{bmatrix} x \\ y \end{bmatrix}$.

- apply **forward-backward splitting** to $0 \in Az + Bz$:

$$\begin{aligned} & z^{k+1} = J_{\gamma A} \circ F_{\gamma B} z^k \\ \iff \ & z^{k+1} = (I + \gamma A)^{-1}(I - \gamma B) z^k \\ \iff \ & \text{solve } \begin{cases} x^{k+1} + \gamma L^T y^{k+1} + \gamma \widetilde{\nabla} r_2(x^{k+1}) = x^k - \gamma \nabla f(x^k) \\ y^{k+1} - \gamma L x^{k+1} \ + \gamma \widetilde{\nabla} r_1^*(y^{k+1}) = y^k \end{cases} \end{aligned}$$

issue: both x^{k+1} and y^{k+1} appear in both equations!

- **solution:** introduce the **metric**

$$U = \begin{bmatrix} I & -\gamma L^T \\ -\gamma L & I \end{bmatrix} \succ 0$$

- apply **forward-backward splitting** to $0 \in U^{-1}Az + U^{-1}Bz$:

$$\begin{aligned} & z^{k+1} = J_{\gamma U^{-1}A} \circ F_{\gamma U^{-1}B} z^k \\ \Longleftrightarrow \quad & z^{k+1} = (I + \gamma U^{-1}A)^{-1}(I - \gamma U^{-1}B) z^k \\ \Longleftrightarrow \quad & \text{solve } Uz^{k+1} + \gamma \tilde{A} z^{k+1} = Uz^k - \gamma B z^k \\ \Longleftrightarrow \quad & \begin{cases} x^{k+1} - \gamma L^T y^{k+1} + \gamma L^T y^{k+1} + \gamma \widetilde{\nabla} r_2(x^{k+1}) = x^k - \gamma L^T y^k - \gamma \nabla f(x^k) \\ y^{k+1} - \gamma L\, x^{k+1} - \gamma L x^{k+1} + \gamma \nabla r_1^*(y^{k+1}) = y^k - \gamma L\, x^k \end{cases} \end{aligned}$$

(like Gaussian elimination, y^{k+1} is cancelled from the first equation)

- **strategy**: obtain x^{k+1} from the first equation; plug in x^{k+1} as a constant into the second equation and then obtain y^{k+1}
- **final iteration:**

$$x^{k+1} = \mathbf{prox}_{\gamma r_2}(x^k - \gamma L^T y^k - \gamma \nabla f(x^k))$$
$$y^{k+1} = \mathbf{prox}_{\gamma r_1^*}(y^k + \gamma L(2x^{k+1} - x^k))$$

- **nice properties:**
  - apply L and ∇f explicitly
  - solve proximal-point subproblems of r_1 and r_2
  - convergence follows from standard forward-backward splitting

Example: total variation (TV) deblurring

- **Rudin-Osher-Fatemi'92:**

$$\underset{u}{\text{minimize}} \ \frac{1}{2}\|Ku - b\|^2 + \lambda\|Du\|_1$$
$$\text{subject to } 0 \le u \le 255$$

  4 simple parts: smooth + simple ∘ linear + simple

  - smooth function: $f(u) = \frac{1}{2}\|Ku - b\|^2$
  - linear operator: D
  - simple nonsmooth function: $r_1 = \lambda\|\cdot\|_1$
  - simple indicator function: $r_2 = \iota_{[0,255]}$

(We only show the results, skipping the details)

- **equivalent condition**:

$$0 \in \begin{bmatrix} \partial r_2 & D^* \\ -D & \partial r_1^* \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} + \begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix}$$

(where w is the auxiliary (dual) variable)

- **forward-backward splitting** algorithm under a **special metric**:

$$u^{k+1} = \mathbf{proj}_{[0,255]^n}(u^k - \gamma D^* w^k - \gamma \nabla f(u^k))$$
$$w^{k+1} = \mathbf{prox}_{\gamma \ell_\infty}(w^k + \gamma D(2u^{k+1} - u^k))$$

every step is simple to implement

Operator splitting:

Parallel and distributed applications

Huge matrix A

- **background**: you wish to distribute a huge matrix A in your problem
- **three schemes** to distribute A:

$$\begin{bmatrix} —A_{(1)}— \\ \vdots \\ —A_{(M)}— \end{bmatrix} \quad \begin{bmatrix} | & & | \\ A_1 & \cdots & A_N \\ | & & | \end{bmatrix} \quad \begin{bmatrix} A_{1,1} & \cdots & A_{1,N} \\ & \cdots & \\ A_{M,1} & \cdots & A_{M,N} \end{bmatrix}$$

scheme 1   scheme 2   scheme 3

- **broadcast then parallelize**: schemes 1 & 2:

$$Ax = \begin{bmatrix} A_{(1)}x \\ \vdots \\ A_{(M)}x \end{bmatrix} \quad \text{and} \quad A^T y = \begin{bmatrix} A_1^T y \\ \vdots \\ A_N^T y \end{bmatrix}$$

- **parallelize then reduce**: schemes 1 & 2:

$$A^T y = \sum_{i=1}^{M} A_{(i)}^T y_i \quad \text{and} \quad Ax = \sum_{j=1}^{N} A_j x_j$$

- **broadcast, parallelize, then reduce**: scheme 3:

$$Ax = \begin{bmatrix} \sum_{j=1}^{N} A_{1,j} x_j \\ \vdots \\ \sum_{j=1}^{N} A_{M,j} x_j \end{bmatrix} \quad \text{and} \quad A^T y = \begin{bmatrix} \sum_{i=1}^{M} A_{i,1}^T y_i \\ \vdots \\ \sum_{i=1}^{M} A_{i,N} y_i \end{bmatrix}$$

choose a scheme baesd on the structures of functions/operators

- **example**: convex and smooth f; convex r (possibly nonsmooth)

$$\underset{x}{\text{minimize}}\ r(x) + f(Ax)$$

- **case 1**: **scheme 1 for separable** f, that is, $f(Ax) = \sum_{i=1}^{m} f_i(A_{(i)}x)$

  apply forward-backward splitting

$$\begin{aligned} x^{k+1} &= \mathbf{prox}_{\gamma r}(x^k - \gamma A^T \nabla f(Ax^k)) \\ &= \mathbf{prox}_{\gamma r}\Big(x^k - \gamma \sum_{i=1}^{M} A_{(i)}^T \nabla f_i(A_{(i)} x^k)\Big) \end{aligned}$$

  we can distribute/parallelize $A_{(i)}^T \nabla f_i(A_{(i)} x^k)$

- **scenario 2: scheme 2 for separable r**

$$\mathbf{prox}_{\gamma r}(y) = \begin{bmatrix} \mathbf{prox}_{\gamma r_1}(y_1) \\ \vdots \\ \mathbf{prox}_{\gamma r_N}(y_N) \end{bmatrix}$$

apply forward-backward splitting:

$$x^{k+1} = \mathbf{prox}_{\gamma r}(x^k - \gamma A^T \nabla f(Ax^k))$$

$$\iff \begin{cases} \text{cache } g = \nabla f(Ax^k) \\ \begin{bmatrix} x_1^{k+1} \\ \vdots \\ x_N^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{prox}_{\gamma r_1}(x_1^k - \gamma A_1^T g) \\ \vdots \\ \mathbf{prox}_{\gamma r_N}(x_N^k - \gamma A_N^T g) \end{bmatrix} \\ \text{broadcast } g = \nabla f\left(\sum_{j=1}^M A_j x_j^{k+1}\right) \end{cases}$$

we distribute the computing of $A_j x_j^{k+1} = A_j \mathbf{prox}_{\gamma r_j}(x_j^k - \gamma A_j^T g)$

this is also known as *parallel coordinate descent*

- **scenario 3**: **scheme 3 for separable** f **and** r (we skip the details)
- **remarks**:
  - no introduction of extra variables, no sacrifice of convergence speed
  - the principle also applies to other operator splitting schemes
  - can be further accelerated by *asynchronous* parallelism

Duality and structure trade

- assume **scheme 1**:
$$A = \begin{bmatrix} —A_{(1)}— \\ \vdots \\ —A_{(M)}— \end{bmatrix}$$

- **primal problem**: strongly-convex r and convex nonsmooth $f_1, \ldots, f_m$.:
$$\underset{w}{\text{minimize}}\ r(w) + \textstyle\sum_{i=1}^{m} f_i(A_{(i)} w)$$
structure: strongly-convex $+ \sum$ nonsmooth $\circ$ linear

- let $f^*(y) = \sup_x \langle y, x\rangle - f(x)$ denote the convex conjugate of f

- **dual problem**:
$$\underset{y}{\text{minimize}}\ r^*\Big(- \textstyle\sum_{i=1}^{m} A_{(m)}^T y_i\Big) + \textstyle\sum_{i=1}^{m} f_i^*(y_i)$$
dual structure: smooth $\circ$ linear $+ \sum$ nonsmooth

Example: support vector machine (SVM)

- given sample-label pairs (x_i, y_i) where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, -1\}$
- **primal problem**:

$$\underset{w,b}{\text{minimize}}\ \tfrac{1}{2}\|w\|^2 + \textstyle\sum_{i=1}^m \iota_{\mathbb{R}_+}\big(y_i(x_{(i)}^T w - b) - 1\big)$$

- **dual problem**:

$$\underset{\alpha}{\text{maximize}}\ \text{quadratic}(\alpha) + \textstyle\sum_{i=1}^m \iota_{\mathbb{R}_+}(\alpha_i) + \iota_{\{y_1\alpha_1+\cdots+y_m\alpha_m=0\}}(\alpha)$$

apply scheme 1 and the three-operator DYS

Jacobi parallel ADMM

$$\begin{aligned} \underset{\mathbf{x}=(x_1,\ldots,x_m)}{\text{minimize}} \quad & f_1(x_1) + \cdots + f_m(x_m) + g(\mathbf{x}) \\ \text{subject to } & A_1 x_1 + \cdots + A_m x_m = b. \end{aligned}$$

- **require**: convex f_i (possible nonsmooth); convex and smooth g
- **examples**: LP, QP, basis pursuit, control, exchange problems, ...
- **equivalent condition**: with dual variable y,

$$0 \in \begin{bmatrix} \partial f_1 & & & -A_1^T \\ & \ddots & & \vdots \\ & & \partial f_m & -A_m^T \\ A_1 & \cdots & A_m & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ y \end{bmatrix} + \begin{bmatrix} \nabla_1 g(\mathbf{x}) \\ \vdots \\ \nabla_m g(\mathbf{x}) \\ b \end{bmatrix}$$

- **introduce** the metric

$$U = \begin{bmatrix} I - \sigma \mathbf{A}^T \mathbf{A} & 0 \\ 0 & I \end{bmatrix}$$

  where $\mathbf{A} = [A_1, \ldots, A_m]$

- **obtain** the algorithm

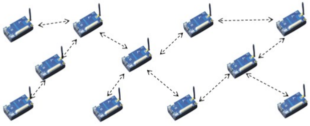
$$\begin{cases} x_i^{k+1} = \arg\min_{x_i} f_i(x_i) + \langle \nabla_i g(\mathbf{x}^k) - y + \sigma A_i^T(\mathbf{A}\mathbf{x} - b), x_i \rangle + \frac{1}{2}\|x_i - x_i^k\|^2 \\ \qquad \forall i = 1, \ldots, m \\ y^{k+1} = y^k - \sigma(\mathbf{A}\mathbf{x} - b) \end{cases}$$

- **nice properties**:
  - all x_i-subproblems can be solved in parallel
  - ∇g and A are applied in an explicit manner

Operator splitting:
Decentralized applications

Decentralized computing

- n agents in a connected network $G = (V, E)$ with bi-directional links E



- each agent i has a private function f_i
- **problem**: find a *consensus solution* x^* to

$$\underset{\mathbf{x}\in\mathbb{R}^p}{\text{minimize}}\ f(\mathbf{x}) := \sum_{i=1}^{n} f_i(x_i) \quad \text{subject to } x_i = x_j,\ \forall i, j.$$

- **challenges**: no center, only between-neighbor communication
- **benefits**: fault tolerance, no long-dist communication, privacy

Decentralized ADMM

- Decentralized consensus optimization problem:

$$\begin{aligned}\underset{x_i, i\in V}{\text{minimize}} \quad & \textstyle\sum_{i\in V} f_i(x_i)\\ \text{subject to } & x_i = x_j,\ \forall (i,j)\in E\end{aligned}$$

- **ADMM reformulation**:

$$\begin{aligned}\underset{x_i, i\in V,\, y_{ij}, (i,j)\in E}{\text{minimize}} \quad & \textstyle\sum_{i\in V} f_i(x_i)\\ \text{subject to} \quad & x_i = y_{ij},\ x_j = y_{ij},\ \forall (i,j)\in E\end{aligned}$$

- ADMM alternates between two steps
  - each agent: update x_i while related y_{ij} are fixed
  - each pair of agents (i,j): update y_{ij} and dual var while x_i, x_j are fixed

Primal-dual splitting

- **problem**:

$$\underset{x_i, i \in V}{\text{minimize}} \ \ \textstyle\sum_{i \in V} r_i(x_i) + f_i(x_i) \quad \text{subject to } W\mathbf{x} = \mathbf{x}.$$

  where r_i are convex and f_i are convex and smooth

- **the mixing matrix** $W \in \mathbb{R}^{|E| \times |E|}$:
  - $w_{ij} \neq 0$ only if $i = j$ or agents i, j are neighbors
  - symmetric $W = W^T$, doubly stochastic $W\mathbf{1} = \mathbf{1}$. thus $I - W \succeq 0$

- **consensus**: $x_i = x_j, \ \forall (i,j) \in E \ \Leftrightarrow \ W\mathbf{x} = \mathbf{x}$ where $\mathbf{x}$ stacks all x_i^T

- **equivalent problem**:

$$\underset{x_i, i \in V}{\text{minimize}} \ \ r(\mathbf{x}) + f(\mathbf{x}) = \textstyle\sum_{i \in V} r_i(x_i) + f_i(x_i) \quad \text{subject to } W\mathbf{x} = \mathbf{x}.$$

- let $V^T V = \frac{1}{2}(I - W)$
- **equivalent problem** (KKT conditions):

$$0 \in \begin{bmatrix} \partial r & V^T \\ -V & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} \nabla f(\mathbf{x}) \\ 0 \end{bmatrix}$$

- applying **forward-backward splitting** with a **special metric**, skipping details, we obtain

$$\mathbf{x}^{k+1+1/2} = W\mathbf{x}^{k+1} + \mathbf{x}^{k+1/2} - \frac{1}{2}(W + I)\mathbf{x}^k - \alpha\left[\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)\right]$$

$$\mathbf{x}^{k+2} = \arg\min r(\mathbf{x}) + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}^{k+1+1/2}\|_F^2$$

this recovers the PG-EXTRA decentralized algorithm

Operator-splitting analysis

How to analyze splitting algorithms?

- **problem**:

$$\text{find } \ x \ \text{ such that } \quad 0 \in (A+B)x \quad \text{and} \quad 0 \in (A+B+C)x$$

- **iteration**:

$$z^{k+1} = T(z^k)$$

- **require**:
  - fixed point z^* of T encodes a solution x^*
  - $\|z^{k+1} - z^*\| < \|z^k - z^*\|$; sufficiency: T is α-**averaged**, $\alpha \in (0,1)$

Averaged operator

- weaker than contractive operators; strong than nonexpansive operators
- T is α-averaged, $\alpha \in (0,1)$, if for any $z, \bar{z} \in \mathcal{H}$

$$\|Tz - T\bar{z}\|^2 \le \|z - \bar{z}\|^2 - \frac{1-\alpha}{\alpha}\|(I-T)z - (I-T)\bar{z}\|^2.$$

- assume $z^{k+1} = Tz^k$ and $\bar{z} = T\bar{z}$, then

$$\|z^{k+1} - \bar{z}\|^2 \le \|z^k - \bar{z}\|^2 - \frac{1-\alpha}{\alpha}\|z^{k+1} - z^k\|^2,$$

  consequences:
  - $\|z^{k+1} - z^k\| \to 0$
  - boundedness of $\{z^k\}$, subsequence $z^{k_j} \to z^*$ weakly
  - (by demiclossedness and monotonicity) $z^k \to z^*$ weakly and $z^* = Tz^*$
  - $\|z^{k+1} - z^k\|^2 = o(1/k)$ (Davis-Y'15)

Key examples

- A is monotone $\Rightarrow$ $J_{\gamma A} := (I + \gamma A)^{-1}$ is (1/2)-averaged[2]
- A is monotone $\Rightarrow$ $R_{\gamma A}$ is nonexpansive
- A is β-cocoercive $\Rightarrow$ $F_{\gamma A} := I - \gamma A$ is $(1 - \frac{\gamma}{2\beta})$-averaged
- Baillon-Haddad: if f is convex, ∇f is $\frac{1}{\beta}$-Lipschitz if and only if ∇f is β-cocoercive

  therefore, ∇f is $\frac{1}{\beta}$-Lipschitz $\Rightarrow$ $I - \gamma \nabla f$ is $(1 - \frac{\gamma}{2\beta})$-averaged

[2] also known as "firmly nonexpansive"

Key properties

- T_1 is nonexpansive $\Rightarrow$ $T_2 = (1-\alpha)I + \alpha T_1$ is α-averaged, $\alpha \in (0,1)$
- T_1, T_2 are nonexpansive $\Rightarrow$ $T_1 \circ T_2$ is nonexpansive
- T_1, T_2 are averaged $\Rightarrow$ $T_1 \circ T_2$ is averaged

Key consequences

- **assume** A, B are monotone
- B is β-cocoercive, $\gamma \in (0, 2\beta) \;\Rightarrow\;$ FBS $J_{\gamma A} \circ F_{\gamma B}$ is averaged
- PRS $R_{\gamma A} \circ R_{\gamma B}$ is nonexpansive
- DRS $\frac{1}{2}I + \frac{1}{2}R_{\gamma A} \circ R_{\gamma B}$ is (1/2)-averaged
- C is β-cocoercive, $\gamma \in (0, 2\beta)$
  $\Rightarrow\;$ DYS $I - J_{\gamma B} + J_{\gamma A} \circ (2J_{\gamma B} - I - \gamma C \circ J_{\gamma B})$ is $\frac{2\beta}{4\beta - \gamma}$-averaged

Open question

Find an operator-splitting scheme for

$$0 \in (T_1 + \cdots + T_m)x, \quad m \geq 4.$$

require:

- no use of auxiliary variable
- convergence is guaranteed under monotonic T_i's

Summary

- monotone operator splitting is a set of powerful and elegant tools for many problems in signal processing, machine learning, computer vision, etc.
- they give rise to parallel, distributed, and decentralized algorithms
- under the hood: fixed-point and nonexpansive-operator theory

not covered: the **convergence rates** of

- objective error: $f^k - f^*$
- point error: $\|z^k - z^*\|^2$
- accelerated rates by averaging and extrapolation

Thank you!

References:

- H. Bauschke and P. Combettes. Convex analysis and monotone operator theory in Hilbert spaces. Springer, 2011.
- N. Komodakis and J.-C. Pasquet. Playing with duality: an overview of recent primal-dual approaches for solving large-scale optimization problems. IEEE Signal Processing Magazine, 2014.
- D. Davis and Y, Convergence rate analysis of several splitting schemes, *UCLA CAM 14-51*, 2014.
- D. Davis and Y, A three-operator splitting method and its acceleration, *UCLA CAM 15-13*, 2015.