ADMM

Acknowledgement: slides are from Prof. zaiwen Wen and Prof. wotao yin.

Outline

Separable objective and coupling constraints

Consider a convex program with a separable objective and coupling constraints

$$\min_{x,z} f(\mathbf{x}) + g(\mathbf{z}) \ \text{ s.t. } \mathbf{Ax} + \mathbf{Bz} = \mathbf{b}$$

Examples:

- $\min f(\mathbf{x}) + g(\mathbf{x}) \Rightarrow \min_{\mathbf{x},\mathbf{z}} \{f(\mathbf{x}) + g(\mathbf{z}) : \mathbf{x} - \mathbf{z} = 0\}$
- $\min f(\mathbf{x}) + g(\mathbf{Ax}) \Rightarrow \min_{\mathbf{x},\mathbf{z}} \{f(\mathbf{x}) + g(\mathbf{z}) : \mathbf{Ax} - \mathbf{z} = 0\}$
- $min\{f(\mathbf{x}) : \mathbf{AX} \in \mathcal{C}\} \Rightarrow \min_{\mathbf{x},\mathbf{z}} \{f(\mathbf{x}) + l_{\mathcal{C}}(\mathbf{z}) : \mathbf{Ax} - \mathbf{z} = 0\}$
- $\min \Sigma_{i=1}^N f_i(\mathbf{x}) \Rightarrow \min_{\{\mathbf{x_i}\},\mathbf{z}} \{\Sigma_{i=1}^N f_i(\mathbf{x_i}) : \mathbf{x_i} - \mathbf{z} = 0, \forall i\}$
  each $\mathbf{x_i}$ is **a copy** of $\mathbf{x}$ for f_i, not a subvector of $\mathbf{x}$.

Alternating direction method of multipliers(ADMM)

Let f and g be **convex.** They may be **nonsmooth**, can take the **extended value.** Consider

$$\begin{aligned}\min_{\mathbf{x},\mathbf{z}}\ & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t. }\ & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}.\end{aligned}$$

Define the augmented Lagrangian function:

$$L_\beta(\mathbf{x}, \mathbf{z}, \mathbf{w}) = f(\mathbf{x}) + g(\mathbf{z}) - \mathbf{w}^\top(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{b}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{b}\|_2^2$$

Standard ADMM iteration

1. $\mathbf{x}^{k+1} = \operatorname{argmin}_{\mathbf{x}} L_\beta(\mathbf{x}, \mathbf{z}^{\mathbf{k}}, \mathbf{w}^k)$,
2. $\mathbf{z}^{k+1} = \operatorname{argmin}_{\mathbf{z}} L_\beta(\mathbf{x}^{k+1}, \mathbf{z}, \mathbf{w}^k)$,
3. $\mathbf{w}^{k+1} = \mathbf{w}^k - \beta(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b})$.

Be careful about the form of the augmented Lagrangian function

Alternating direction method of multipliers(ADMM)

Consider

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z})$$
$$\text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}.$$

ADMM variants:

1. $\mathbf{x}^{k+1} = \mathrm{argmin}_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{z}^k) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k\|_2^2,$
2. $\mathbf{z}^{k+1} = \mathrm{argmin}_{\mathbf{z}} f(\mathbf{x}^{k+1}) + g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z} - \mathbf{b} - \mathbf{y}^k\|_2^2,$
3. $\mathbf{y}^{k+1} = \mathbf{y}^k - (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b}).$

Dates back to Douglas, Peaceman, and Rachford (50s-70s, operator splitting for PDEs); Glowinsky et al.'80s, Gabay'83; Spingarn'85; Eckstein and Bertsekes'92, He et al.'02 in variational inequality.

Alternating direction method of multipliers(ADMM)

Comments:

- $\mathbf{y}$ is the **scaled dual variable**, $\mathbf{y} = \beta$ (Lagrange multipliers)
- $\mathbf{y}$-update can take a large step size $\gamma < \frac{1}{2}(\sqrt{5}+1)$

$$\mathbf{y}^{k+1} = \mathbf{y}^{k} - \gamma(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b}).$$

- Gauss-Seidel style update is applied to $\mathbf{x}$ and $\mathbf{z}$ of either order
- If $\mathbf{x}$ and $\mathbf{z}$ are minimized jointly, it reduces to augmented Lagrangian iteration:

$$\begin{aligned}(\mathbf{x}^{k+1}, \mathbf{z}^{k+1}) &= \operatorname*{argmin}_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{b} - \mathbf{y}^k\|_2^2 \\ \mathbf{y}^{k+1} &= \mathbf{y}^{k} - (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b}).\end{aligned}$$

- it extends to multiple blocks (a few questions remain open)
- it extends to Jacobian (parallel) updates with damping the update of $\mathbf{y}$

Why is ADMM liked

- Split awkward intersections and objectives to easy subproblems
  - $\mathbf{X} \succeq \mathbf{0}, \mathbf{X} \geq 0 \rightarrow$ seperate projections
  - $\|\mathbf{L}\|_* + \beta\|\mathbf{M} - \mathbf{L}\|_1 \rightarrow$ separate subproblems with $\|\cdot\|_*$ and $\|\cdot\|_1$
  - $\|\nabla \mathbf{x}\|_1 \rightarrow$ decouple $\|\cdot\|_1$ and ∇ to separable subproblems
  - $\Sigma_i \|\mathbf{x}_{[\mathcal{G}_i]}\|_2 \rightarrow$ decouple to subproblems of individual groups
  - $\Sigma_{i=1}^K f_i(\mathbf{x}) \rightarrow K$ parallel subproblems(coordinated by gather-scattering or gossiping between neighbors)
- $\#$iterations is comparable to those of other first-order methods, so the total time can be much smaller(not always though)
- Quite easy to implement, be (nearly) state-of-the-art for a few hours' work

Outline

KKT conditions

Recall KKT conditions (omitting the complementarity part):

$$
\begin{aligned}
&(\textit{primal feasibility}) && \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{z}^* = \mathbf{b} \\
&(\textit{dual feasibility I}) && 0 \in \partial f(\mathbf{x}^*) + \mathbf{A}^T \mathbf{y}^* \\
&(\textit{dual feasibility II}) && 0 \in \partial g(\mathbf{z}^*) + \mathbf{B}^T \mathbf{y}^*
\end{aligned}
$$

Recall $\mathbf{z}^{k+1} = \operatorname{argmin}_{\mathbf{z}} g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z} - \mathbf{b} - \mathbf{y}^k\|_2^2$
$\Rightarrow 0 \in \partial g(\mathbf{z}^{k+1}) + \mathbf{B}^T(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b} - \mathbf{y}^k) = \partial g(\mathbf{z}^{k+1}) + \mathbf{B}^T \mathbf{y}^{k+1}$
Therefore, dual feasibility II is maintained.
Dual feasibility I is not maintained since

$$
0 \in \partial f(\mathbf{x}^{k+1}) + \mathbf{A}^T(\mathbf{y}^{k+1} + \mathbf{B}(\mathbf{z}^k - \mathbf{z}^{k+1}))
$$

But, primal feasibility and dual feasibility I hold asymptotically as $k \to \infty$.

Convergence of ADMM

ADMM is neither purely-primal nor purely-dual. There is no known objective closely associated with the iterations. Recall via the transform

$$\mathbf{y}^k = \mathrm{prox}_{\beta d_1} \mathbf{w}^k,$$

ADMM is a fixed-point iteration

$$\mathbf{w}^{k+1} = (\frac{1}{2}I + \frac{1}{2}\mathbf{refl}_{\beta d_1}\mathbf{refl}_{\beta d_2})\mathbf{w}^k,$$

where the operator is firmly nonexpansive.

Convergence

- Assumptions: f and g convex, closed, proper, and $\exists$ KKT point
- $\mathbf{A}\mathbf{x}^k + \mathbf{B}\mathbf{z}^k \to \mathbf{b}, f(\mathbf{x}^k) + g(\mathbf{z}^k) \to p^*, \mathbf{y}^k$ converge
- Inaddition, if $(\mathbf{x}^k, \mathbf{y}^k)$ are bounded, they also converge

Rate of convergence

- simplified cases, exact updates, f smooth, and ∇f Lipschitz $\rightarrow$ objective$\sim O(1/k), O(1/k^2)$
- at least one update is exact $\rightarrow$
  ergodic: objective error $+(\tilde{\mathbf{u}}^k - \mathbf{u}^*)^T F(\mathbf{u}^*) \sim O(1/k)$
  non-ergodic: $\|\mathbf{u}^k - \mathbf{u}^{k+1}\| \sim O(1/k)$
- f strongly convex and ∇f Lipschitz $+$ some full rank conditions $\rightarrow$ both solution and objective $\sim O(1/c^k), c > 1$
- applied to LP and QP $\rightarrow$ (asymptotic) strongly convex

Outline

Variants of ADMM

- An ADMM subproblem is easy, if it has a closed-form solution;
- If a subproblem is difficult, it may be not worth solving it exactly.
  This motivates variants of ADMM.
  A few approaches of inexact ADMM subproblems:
  1.**Iteration limiter**: limited iterations of CG or L-BFGS applied to

$$\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{v}\|_2^2$$

  where $\mathbf{v} = \mathbf{b} - \mathbf{B}\mathbf{z}^k + \mathbf{y}^k$.
  - Applicable to quadratic f, perhaps other C^2 functions as well
  - Does not apply to nonsmooth subproblems
  - Practically efficient, but lacking theoretical guarantees for now

Variants of ADMM

2.**Cached factorization**: For quadratic subproblem $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{C}\mathbf{x} - \mathbf{d}\|_2^2$, $\mathbf{x}-$subproblem solves

$$(\mathbf{C}^T\mathbf{C} + \beta\mathbf{A}^T\mathbf{A})\mathbf{x}^{k+1} = (\cdots)$$

- cache the Cholesky or LDL^T decomposition to$(\mathbf{C}^T\mathbf{C} + \beta\mathbf{A}^T\mathbf{A})$
- later, in each iteration, solve simple triangle systems
- changing β generally requires re-factorizatio
- if $(\mathbf{C}^T\mathbf{C} + \beta\mathbf{A}^T\mathbf{A})$ has a (simple+low-rank) structure, apply the Woodbury matrix inversion formula

Variants of ADMM

3.**Single gradient-descent step.** Simplify $\mathbf{x}-$update from

$$\mathbf{x}^{k+1} = \operatorname{argmin} f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k\|_2^2$$

to

$$\mathbf{x}^{k+1} = \mathbf{x}^k - c^k(\nabla f(\mathbf{x}^k) + \beta\mathbf{A}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k))$$

- applicable to C^1 subproblems only
- convergence requires reduced update to $\mathbf{y}$
- gradient update c^k and $\mathbf{y}-$update step sizes γ depend on spectral properties of $\mathbf{A}$

Variants of ADMM

4.**Single prox-linear step.** Simplify $\mathbf{x}-$update from

$$\mathbf{x}^{k+1} = \operatorname{argmin} f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k\|_2^2$$

to

$$\mathbf{x}^{k+1} = \operatorname{argmin} f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{x} \rangle + \frac{1}{2t}\|\mathbf{x} - \mathbf{x}^k\|_2^2,$$

where

$$\mathbf{g} = \nabla_{\mathbf{x}}\left(\frac{\beta}{2}\|\mathbf{A}\mathbf{x}^k + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k\|_2^2\right)$$

- similar to the prox-linear iteration
- applicable to nonsmooth f
- convergence requires reduced $\mathbf{y}-$update
- t, β,step size γ of $\mathbf{y}-$update, and spectral properties of $\mathbf{A}$ are related
- also applicable to the other subproblem simultaneously

Variants of ADMM

5.**Approximating** $\mathbf{A}^T\mathbf{A}$ **by nice matrix** $\mathbf{D}$. As an example, repalce

$$\mathbf{x}^{k+1} = \operatorname{argmin} f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k\|_2^2$$

by

$$\mathbf{x}^{k+1} = \operatorname{argmin} f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k\|_2^2 + \frac{\beta}{2}(\mathbf{x} - \mathbf{x}^k)^T(\mathbf{D} - \mathbf{A}^T\mathbf{A})(\mathbf{x} - \mathbf{x}^k)$$

- also known as "optimization transfer"
- reduces to the prox-linear step if $\mathbf{D} = \frac{\beta}{t} I$
- useful if

  $$\min f(\mathbf{x}) + \frac{\beta}{2}\mathbf{x}^T\mathbf{D}\mathbf{x}$$

  is computationally easier than

  $$\min f(\mathbf{x}) + \frac{\beta}{2}\mathbf{x}^T(\mathbf{A}^T\mathbf{A})\mathbf{x}.$$

- applications: $\mathbf{A}$ is an off-the-grid Fourier transform

Outline

LASSO

考虑问题

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{x}\|_1,$$

将问题写为如下形式：

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{z}} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{z}\|_1,\\ \text{s.t.} \quad & \boldsymbol{x}-\boldsymbol{z}=\boldsymbol{0},\end{aligned}$$

对于此问题，交替方向乘子法迭代为：

$$\begin{aligned}\boldsymbol{x}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}}\left(\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+(\frac{\rho}{2})\|\boldsymbol{x}-\boldsymbol{z}^k+\boldsymbol{u}^k\|_2^2\right),\\ &= (\boldsymbol{A}^T\boldsymbol{A}+\rho\boldsymbol{I})^{-1}(\boldsymbol{A}^T\boldsymbol{b}+\rho(\boldsymbol{z}^k-\boldsymbol{u}^k)),\\ \boldsymbol{z}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{z}}\left(\lambda\|\boldsymbol{x}\|_1+(\frac{\rho}{2})\|\boldsymbol{x}^{k+1}-\boldsymbol{z}+\boldsymbol{u}^k\|_2^2\right),\\ &= prox_{(\lambda/\rho\|\cdot\|_1)}(\boldsymbol{x}^{k+1}+\boldsymbol{u}^k),\\ \boldsymbol{u}^{k+1} &= \boldsymbol{u}^k+\tau(\boldsymbol{x}^{k+1}-\boldsymbol{z}^{k+1}).\end{aligned}$$

LASSO

考虑问题

$$\min_{\boldsymbol{x}} \ \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2+\lambda\|\boldsymbol{x}\|_1,$$

对偶问题：

$$\min_{\boldsymbol{y}} \ \{-\boldsymbol{b}^T\boldsymbol{y}+\frac{\mu}{2}\|\boldsymbol{y}\|_2^2: \ \|\boldsymbol{A}^T\boldsymbol{y}\|_\infty \le 1\}$$

等价于：

$$\min_{\boldsymbol{y},\boldsymbol{z}} \ \{-\boldsymbol{b}^T\boldsymbol{y}+\frac{\mu}{2}\|\boldsymbol{y}\|_2^2+\mathbb{I}_{\{\|z\|_\infty\le 1\}}: \ \boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{z}=0\}$$

对偶问题的增广拉格朗日函数：

$$L(\boldsymbol{y},\boldsymbol{z};\boldsymbol{x})=\boldsymbol{b}^T\boldsymbol{y}+\frac{\mu}{2}\|\boldsymbol{y}\|_2^2+\mathbb{I}_{\|z\|_\infty\le 1}+\frac{\beta}{2}\|\boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{z}-\boldsymbol{x}\|_2^2,$$

应用交替方向乘子法迭代，可得：

$$\begin{aligned}
\boldsymbol{z}^{k+1} &= \mathcal{P}_{\|.\|_\infty\le 1}(\boldsymbol{x}^k-\boldsymbol{A}^T\boldsymbol{y}^k),\\
\boldsymbol{y}^{k+1} &= (\mu\boldsymbol{I}+\beta\boldsymbol{A}\boldsymbol{A}^T)^{-1}(\beta\boldsymbol{A}(\boldsymbol{x}^k-\boldsymbol{z}^{k+1})-\boldsymbol{b}),\\
\boldsymbol{x}^{k+1} &= \boldsymbol{z}^k-\gamma(\boldsymbol{A}^T\boldsymbol{y}^{k+1}+\boldsymbol{z}^{k+1}).
\end{aligned}$$

Consider

$$\begin{aligned} \min_{X \in S^n} \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \left\langle A^{(i)}, X \right\rangle = b_i, \quad i = 1, \cdots, m, \\ & X \succeq 0 \end{aligned}$$

The dual problem

$$(D) \quad \begin{cases} \min\limits_{y \in \mathbb{R}^m, S \in S^n} & -b^\top y \\ \quad \text{s.t.} & \mathcal{A}^*(y) + S = C, \quad S \succeq 0, \end{cases}$$

Augmented Lagrangian function:
$\mathcal{L}_\mu(X, y, S) = -b^\top y + \langle X, \mathcal{A}^*(y) + S - C \rangle + \frac{1}{2\mu} \|\mathcal{A}^*(y) + S - C\|_F^2$.

ADMM for SDP

$$
\begin{aligned}
y^{k+1} &:= \arg\min_{y\in\mathbb{R}^m} \mathcal{L}_\mu(X^k,\ y,\ S^k), \\
&= -(\mathcal{A}\mathcal{A}^*)^{-1}\left(\mu(\mathcal{A}(X^k)-b)+\mathcal{A}(S^k-C)\right) \\
S^{k+1} &:= \arg\min_{S\in\mathcal{S}^n} \mathcal{L}_\mu(X^k,\ y^{k+1},\ S), \quad S\succeq 0, \\
X^{k+1} &:= X^k + \frac{\mathcal{A}^*(y^{k+1})+S^{k+1}-C}{\mu}.
\end{aligned}
$$

- The S-subproblem:

$$
\min_{S\in\mathcal{S}^n} \quad \left\|S-V^{k+1}\right\|_F^2, \quad S\succeq 0,
$$

  where $V^{k+1} := V(S^k, X^k) = C - \mathcal{A}^*(y(S^k, X^k)) - \mu X^k$.
- Hence, the solution is

$$
S^{k+1} := V_\dagger^{k+1} := Q_\dagger \Sigma_+ Q_\dagger^\top
$$

  where $V^{k+1} = Q\Sigma Q^\top = \begin{pmatrix} Q_\dagger & Q_\ddagger \end{pmatrix} \begin{pmatrix} \Sigma_+ & \mathbf{0} \\ \mathbf{0} & \Sigma_- \end{pmatrix} \begin{pmatrix} Q_\dagger^\top \\ Q_\ddagger^\top \end{pmatrix}$

ADMM for SDP

Updating the Lagrange multiplier X^{k+1}

- Updating formula:

$$X^{k+1} := X^k + \frac{\mathcal{A}^*(y^{k+1}) + S^{k+1} - C}{\mu}$$

- Equivalent formulation:

$$X^{k+1} = \frac{1}{\mu}(S^{k+1} - V^{k+1}) = \frac{1}{\mu} V_{\ddagger}^{k+1},$$

  where $V_{\ddagger}^{k+1} := -Q_{\ddagger} \Sigma_{-} Q_{\ddagger}$.

- Note that X^{k+1} is also the optimal solution of

$$\min_{X \in S^n} \quad \left\| \mu X + V^{k+1} \right\|_F^2, \quad X \succeq 0.$$

Example: total variation

Let $\mathbf{x}$ represent a 2D image.

$$\min \; TV(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Applications

- Denoising: $\mathbf{A} = I$
- Deblurring and deconvolution:$\mathbf{A}$ is circulant matrix or convolution
- MRI CS: $\mathbf{A} = \mathbf{P}\mathbf{F}$ downsampled Fourier transform; $\mathbf{P}$ is a row selector,$\mathbf{F}$ is Fourier transform
- Circulant CS:$\mathbf{A} = \mathbf{P}\mathbf{C}$ downsampled convolution; $\mathbf{P}$ is a row selector, $\mathbf{C}$ is a circulant matrix or convolution operator

Challenge: TV is the composite of l_1 and ∇x, defined as

$$TV(\mathbf{x}) := \|\nabla \mathbf{x}\|_1 = \sum_{\text{pixels } (i,j)} \left\| \begin{bmatrix} x_{i+1,j} - x_{i,j} \\ x_{i,j+1} - x_{i,j} \end{bmatrix} \right\|_2 .$$

Opportunity: assuming the periodic boundary conditon, $\nabla\cdot$ is a convolution operator.

Example: total variation

Decouple l_1 from ∇x:

$$\min \frac{\mu}{2}\|\mathbf{Ax}-\mathbf{b}\|_2^2 + \|\mathbf{z}\|_1, \text{ s.t. } \nabla\mathbf{x}-\mathbf{z}=\mathbf{0}$$

where $\|\mathbf{z}\|_1 = \Sigma_i\|\mathbf{z}_i\|_2$.

ADMM

- $\mathbf{x}-$update is quadratic in the form of

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}} \ \mathbf{x}^T(\mu\mathbf{A}^T\mathbf{A}+\beta\nabla^T\nabla)\mathbf{x} + \text{linear terms}$$

  If $\mathbf{A}$ is identity, convolution, or partial Fourier, then

$$F(\mu\mathbf{A}^T\mathbf{A}+\beta\nabla^T\nabla)F^{-1}$$

  is a diagonal matrix. So, $\mathbf{x}-$update becomes closed-form.
- $\mathbf{z}-$subproblem is soft-thresholding

This splitting approach is often faster than the splitting

$$\min \ TV(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{Az}-\mathbf{b}\|_2^2, \text{ s.t. } \mathbf{x}-\mathbf{z}=\mathbf{0}$$

because the $\mathbf{x}-$update is not in closed form.

Example: transform l_1 minimization

Model

$$\min \|\mathbf{Lx}\|_1 + \frac{\mu}{2}\|\mathbf{Ax}-\mathbf{b}\|_2^2$$

where examples of $\mathbf{L}$ include

- anisotropic finite difference operators
- orthogonal transforms:DCT, orthogonal wavelets
- frames: curvelets, shearlets

New models

$$\min \frac{\mu}{2}\|\mathbf{Ax}-\mathbf{b}\|_2^2 + \|\mathbf{z}\|_1, \text{ s.t. } \mathbf{Lx}-\mathbf{z}=\mathbf{0},$$

or

$$\min \|\mathbf{Lx}\|_1 + \frac{\mu}{2}\|\mathbf{Az}-\mathbf{b}\|_2^2, \text{ s.t. } \mathbf{x}-\mathbf{z}=\mathbf{0}.$$

Example: l_1 fitting

Model

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$$

New model

$$\min_{\mathbf{x},\mathbf{z}} \|\mathbf{z}\|_1, \text{ s.t. } \mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{b}.$$

ADMM

- $\mathbf{x}-$update is quadratic
- $\mathbf{z}-$update is soft-thresholding

Example: robust(Huber-function) fitting

Model

$$\min_{\mathbf{x}} H(\mathbf{A}\mathbf{x}-\mathbf{b}) = \sum_{i=1}^{m} h(\mathbf{a}_i^T\mathbf{x}-b_i)$$

where

$$h(y) = \begin{cases} \frac{y^2}{2\mu}, & 0 \le |y| \le \mu, \\ |y| - \frac{\mu}{2}, & |y| > \mu. \end{cases}$$

Original model is differentiable, amenable to gradient descent. Split model

$$\min_{\mathbf{x},\mathbf{z}} \; H(\mathbf{z}), \text{ s.t. } \mathbf{A}\mathbf{x}+\mathbf{z}=\mathbf{b}.$$

ADMM

- $\mathbf{x}-$ update is quadratic, involving $\mathbf{A}\mathbf{A}^T$
- $\mathbf{z}-$ update is component-wise separable

考虑

$$\min_{\boldsymbol{X}} \ \mathrm{Tr}(\boldsymbol{S}\boldsymbol{X}) - \log\det \boldsymbol{X} + \lambda\|\boldsymbol{X}\|_1,$$

其中，变量$\boldsymbol{X} \in \boldsymbol{S}_+^n$，$\|\cdot\|_1$定义为矩阵所有元素绝对值的和。
引入拆分$X = Z$，则交替方向乘子法产生的迭代为：

$$\begin{aligned}
\boldsymbol{X}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{X}} \left(\mathrm{Tr}(\boldsymbol{S}\boldsymbol{X}) - \log\det \boldsymbol{X} + (\rho/2)\|\boldsymbol{X} - \boldsymbol{Z}^k + \boldsymbol{U}^k\|_F^2\right), \\
\boldsymbol{Z}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{Z}} \left(\lambda\|\boldsymbol{Z}\|_1 + (\rho/2)\|\boldsymbol{X}^{k+1} - \boldsymbol{Z} + \boldsymbol{U}^k\|_F^2\right) \\
&= prox_{(\lambda/\rho\|\cdot\|_1)}(\boldsymbol{X}^{k+1} + \boldsymbol{U}^k), \\
\boldsymbol{U}^{k+1} &= \boldsymbol{U}^k + (\boldsymbol{X}^{k+1} - \boldsymbol{Z}^{k+1}).
\end{aligned}$$

鲁棒主成分分析

考虑

$$\begin{aligned} \min_{\boldsymbol{L},\boldsymbol{S}} \quad & \|\boldsymbol{L}\|_* + \mu\|\boldsymbol{S}\|_1, \\ s.t. \quad & \boldsymbol{L}+\boldsymbol{S}=\boldsymbol{M}, \end{aligned}$$

此问题的增广拉格朗日函数为：

$$L_\rho(\boldsymbol{L},\boldsymbol{S};\boldsymbol{Y}) = \|\boldsymbol{L}\|_* + \mu\|\boldsymbol{S}\|_1 + \langle \boldsymbol{Y}, \boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}\rangle + \frac{\rho}{2}\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}\|_F^2.$$

交替方向乘子法迭代

$$\begin{aligned} \boldsymbol{L}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{L}} \ \|\boldsymbol{L}\|_* + \frac{\rho}{2}\|\boldsymbol{L}+\boldsymbol{S}^k-\boldsymbol{M}+\boldsymbol{Y}^k/\rho\|_F^2, \\ \boldsymbol{S}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{S}} \ \mu\|\boldsymbol{S}\|_1 + \frac{\rho}{2}\|\boldsymbol{L}^{k+1}+\boldsymbol{S}-\boldsymbol{M}+\boldsymbol{Y}^k/\rho\|_F^2, \\ \boldsymbol{Y}^{k+1} &= \boldsymbol{Y}^k + \rho(\boldsymbol{L}^{k+1}+\boldsymbol{S}^{k+1}-\boldsymbol{M}) \end{aligned}$$

Outline

Block separable ADMM

Suppose $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$ and f is separable, i.e.,

$$f(\mathbf{x}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \cdots + f_N(\mathbf{x}_N).$$

Model

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}. \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & & & \mathbf{0} \\ & \mathbf{A}_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{A}_N \end{bmatrix}$$

Block separable ADMM

The $\mathbf{x}-$update

$$\mathbf{x}^{k+1} \leftarrow \min\ f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k\|_2^2$$

is separable to N independent subproblems

$$\mathbf{x}_1^{k+1} \leftarrow \min\ f_1(\mathbf{x}_1) + \frac{\beta}{2}\|\mathbf{A}_1\mathbf{x}_1 + (\mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k)_1\|_2^2,$$

$$\vdots$$

$$\mathbf{x}_N^{k+1} \leftarrow \min\ f_N(\mathbf{x}_N) + \frac{\beta}{2}\|\mathbf{A}_N\mathbf{x}_N + (\mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k)_N\|_2^2.$$

No coordination is required.

Example: consensus optimization

Model

$$\min \sum_{i=1}^{N} f_i(\mathbf{x})$$

the objective is partially separable.
Introduce N copies $\mathbf{x}_1, \cdots, \mathbf{x}_N$ of $\mathbf{x}$. They have the same dimensions.
New model:

$$\min_{\{\mathbf{x}_i\},\mathbf{z}} \sum_{i=1}^{N} f_i(\mathbf{x}_i), \text{ s.t. } \mathbf{x}_i - \mathbf{z} = \mathbf{0}, \forall i.$$

A more general objective with function g is $\sum_{i=1}^{N} f_i(\mathbf{x}) + g(\mathbf{z})$.
New model:

$$\min_{\{\mathbf{x}_i\},\mathbf{y}} \sum_{i=1}^{N} f_i(\mathbf{x}_i) + g(\mathbf{z}), \text{ s.t. } \begin{bmatrix} I & & \\ & \ddots & \\ & & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} - \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \mathbf{z} = \mathbf{0}.$$

Example: consensus optimization

Lagrangian

$$L(\{\mathbf{x}_i\}, \mathbf{z}; \{\mathbf{y}_i\}) = \sum_i (f_i(\mathbf{x}_i) + \frac{\beta}{2}\|\mathbf{x}_i - \mathbf{z} - \mathbf{y}_i\|_2^2))$$

where $\mathbf{y}_i$ is the Lagrange multipliers to $\mathbf{x}_i - \mathbf{z} = 0$.
ADMM

$$\begin{aligned}
\mathbf{x}_i^{k+1} &= \operatorname*{argmin}_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \frac{\beta}{2}\|\mathbf{x}_i - \mathbf{z}^k - \mathbf{y}_i^k\|_2, i = 1, \cdots, N, \\
\mathbf{z}^{k+1} &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i^{k+1} - \beta^{-1}\mathbf{y}_i^k), \\
\mathbf{y}_i^{k+1} &= \mathbf{y}_i^k - (\mathbf{x}_i^{k+1} - \mathbf{z}^{k+1}), i = 1, \cdots, N.
\end{aligned}$$

The exchange problem

Model $\mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathbb{R}^n$,

$$\min \sum_{i=1}^{N} f_i(\mathbf{x}_i), \text{ s.t. } \sum_{i=1}^{N} \mathbf{x}_i = \mathbf{0}.$$

- it is the dual of the consensus problem
- exchanging n goods among N parties to minimize a total cost
- our goal: to decouple $\mathbf{x}_i-$updates

An equivalent model

$$\min \sum_{i=1}^{N} f_i(\mathbf{x}_i), \text{ s.t. } \mathbf{x}_i - \mathbf{x}_i' = \mathbf{0}, \forall i, \sum_{i=1}^{N} \mathbf{x}_i' = \mathbf{0}.$$

The exchange problem

ADMM after consolidating the $\mathbf{x}_i'$ update:

$$
\begin{aligned}
\mathbf{x}_i^{k+1} &= \operatorname*{argmin}_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \frac{\beta}{2}\|\mathbf{x}_i - (\mathbf{x}_i^k - \text{mean}\{\mathbf{x}_i^k\} - \mathbf{u}^k)\|_2^2, \\
\mathbf{u}^{k+1} &= \mathbf{u}^k + \text{mean}\{\mathbf{x}_i^{k+1}\}.
\end{aligned}
$$

Applications: distributed dynamic energy management

Distributed ADMM I

$$\min_{\{\mathbf{x}_i\},\mathbf{y}} \sum_{i=1}^{N} f_i(\mathbf{x}_i) + g(\mathbf{z}), \text{ s.t. } \begin{bmatrix} I & & \\ & \ddots & \\ & & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} - \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \mathbf{z} = \mathbf{0}.$$

Consider N computing nodes with MPI (message passing interface).

- $\mathbf{x}_i$ are local variables; $\mathbf{x}_i$ is stored and updated <u>on node i only</u>
- $\mathbf{z}$ is the global variable; computed and communicated <u>by MPI</u>
- $\mathbf{y}_i$ are dual variables, stored and updated <u>on node i only</u>

At each iteration, given $\mathbf{y}^k$ and $\mathbf{z}_i^k$

- each node i computes $\mathbf{x}_i^{k+1}$
- each node i computes $\mathbf{P}_i := (\mathbf{x}_i^{k+1} - \beta^{-1}\mathbf{y}_i^k)$
- MPI gathers $\mathbf{P}_i$ and scatters its mean, $\mathbf{z}^{k+1}$, to all nodes
- each node i computes $\mathbf{y}_i^{k+1}$

Example: distributed LASSO

Model

$$\min \|\mathbf{x}\|_1 + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Decomposition

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \mathbf{x}, \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_N \end{bmatrix}.$$

$\Rightarrow$

$$\frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \sum_{i=1}^{N} \frac{\beta}{2}\|\mathbf{A}_i\mathbf{x} - \mathbf{b}_i\|_2^2 =: \sum_{i=1}^{N} f_i(\mathbf{x}).$$

LASSO has the form

$$\min \sum_{i=1}^{N} f_i(\mathbf{x}) + g(\mathbf{x})$$

and thus can be solved by distributed ADMM.

Example: dual of LASSO

LASSO

$$\min \|\mathbf{x}\|_1 + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Lagrange dual

$$\min_{\mathbf{y}} \{\mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 : \|\mathbf{A}^T\mathbf{y}\|_\infty \leq 1\}$$

equivalently,

$$\min_{\mathbf{y},\mathbf{z}} \{-\mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 + l_{\{\|\mathbf{z}\|_\infty \leq 1\}} : \mathbf{A}^T\mathbf{y} + \mathbf{z} = \mathbf{0}\}$$

Standard ADMM:

- primal $\mathbf{x}$ is the multipliers to $\mathbf{A}^T\mathbf{y} + \mathbf{z} = \mathbf{0}$
- $\mathbf{z}-$update is projection to $l_\infty-$ball; easy and separable
- $\mathbf{y}-$update is quadratic

Example: dual of LASSO

- Dual augmented Lagrangian (the scaled form):

$$L(\mathbf{y}, \mathbf{z}; \mathbf{x}) = \mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 + l_{\|\mathbf{z}\|_\infty \le 1} + \frac{\beta}{2}\|\mathbf{A}^T\mathbf{y} + \mathbf{z} - \mathbf{x}\|_2^2$$

- Dual ADMM iterations:

$$\begin{aligned}
\mathbf{z}^{k+1} &= Proj_{\|\cdot\|_\infty \le 1}(\mathbf{x}^k - \mathbf{A}^T\mathbf{y}^k), \\
\mathbf{y}^{k+1} &= (\mu I + \beta \mathbf{A}\mathbf{A}^T)^{-1}(\beta \mathbf{A}(\mathbf{x}^k - \mathbf{z}^{k+1}) - \mathbf{b}), \\
\mathbf{x}^{k+1} &= \mathbf{z}^k - \gamma(\mathbf{A}^T\mathbf{y}^{k+1} + \mathbf{z}^{k+1}).
\end{aligned}$$

  and upon termination at step K, return primal solution

$$\mathbf{x}^* = \beta \mathbf{x}^K \text{ (de-scaling).}$$

- Computation bottlenecks:
  - $(\mu I + \beta \mathbf{A}\mathbf{A}^T)^{-1}$, unless $\mathbf{A}\mathbf{A}^T = I$ or $\mathbf{A}\mathbf{A}^T \approx I$
  - $\mathbf{A}(\mathbf{x}^k - \mathbf{z}^{k+1})$ and $\mathbf{A}^T\mathbf{y}^k$, unless $\mathbf{A}$ is small or has structures

Example: dual of LASSO

Observe

$$\min_{\mathbf{y},\mathbf{z}}\{\mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 + l_{\{\|\mathbf{z}_\infty \le 1\}} : \mathbf{A}^T\mathbf{y} + \mathbf{z} = \mathbf{0}\}$$

- All the objective terms are perfectly separable
- The constraints cause the computation bottlenecks
- We shall try to decouple the blocks of $\mathbf{A}^T$

Distributed ADMM II

A general form with inseparable f and separable g

$$\min_{\mathbf{x},\mathbf{z}} \sum_{l=1}^{L} (f_l(\mathbf{x}) + g_l(\mathbf{z}_l)), \text{ s.t. } \mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{b}$$

- Make L copies $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L$ of $\mathbf{x}$
- Decompose

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_L \end{bmatrix}, \mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_L \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_L \end{bmatrix}$$

- Rewrite $\mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{0}$ as

$$\mathbf{A}_l\mathbf{x}_l + \mathbf{z}_l = \mathbf{b}_l, \mathbf{x}_l - \mathbf{x} = \mathbf{0}, l = 1, \cdots, L.$$

Distributed ADMM II

New model:

$$\begin{array}{ll} \min\limits_{\mathbf{x},\{\mathbf{x}_l\},\mathbf{z}} & \sum_{l=1}^{L}(f_l(\mathbf{x}_l)+g_l(\mathbf{z}_l)) \\ \text{s.t.} & \mathbf{A}_l\mathbf{x}_l+\mathbf{z}_l=\mathbf{b}_l, \mathbf{x}_l-\mathbf{x}=\mathbf{0}, l=1,\cdots,L. \end{array}$$

- $\mathbf{x}_l$'s are copies of $\mathbf{x}$
- $\mathbf{z}_l$'s are sub-blocks of $\mathbf{z}$
- Group variables $\{\mathbf{x}_l\}, \mathbf{z}, \mathbf{x}$ into two sets
  - $\{\mathbf{x}_l\}$: given $\mathbf{z}$ and $\mathbf{x}$, the updates of $\mathbf{x}_l$ are separable
  - $(\mathbf{z}, \mathbf{x})$: given $\{\mathbf{x}_l\}$, the updates of $\mathbf{z}_l$ and $\mathbf{x}$ are separable

  Therefore, standard (2-block) ADMM applies.
- One can also add a simple regularizer $h(\mathbf{x})$

Distributed ADMM II

Consider L computing nodes with MPI.

- $\mathbf{A}_l$ is local data store on node l only
- $\mathbf{x}_l, \mathbf{z}_l$ are local variables; $\mathbf{x}_l$ is stored and updated on node l only
- $\mathbf{x}$ is the global variable; computed and dispatched by MPI
- $\mathbf{y}_l, \bar{\mathbf{y}}_l$ are Lagrange multipliers to $\mathbf{A}_l\mathbf{x}_l + \mathbf{z}_l = \mathbf{b}_l$ and $\mathbf{x}_l - \mathbf{x} = \mathbf{0}$,respectively, stored and updated on node l only

At each iteration,

- each node l computes $\mathbf{x}_l^{k+1}$, using data $\mathbf{A}_l$
- each node l computes $\mathbf{z}_l^{k+1}$, prepares $\mathbf{P}_l = (\cdots)$
- MPI gathers $\mathbf{P}_l$ and scatters its mean, $\mathbf{x}^{k+1}$, to all nodes l
- each node l computes $\mathbf{y}_l^{k+1}, \bar{\mathbf{y}}_l^{k+1}$

Example: distributed dual LASSO

Recall

$$\min_{\mathbf{y},\mathbf{z}}\{\mathbf{b}^T\mathbf{y}+\frac{\mu}{2}\|\mathbf{y}\|_2^2+l_{\{\|\mathbf{z}\|_\infty\leq 1\}}:\mathbf{A}^T\mathbf{y}+\mathbf{z}=\mathbf{0}\}$$

Apply distributed ADMM II

- decompose $\mathbf{A}^T$ to row blocks, equivalently, $\mathbf{A}$ to column blocks.
- make copies of $\mathbf{y}$
- parallel computing $+$ MPI(gathering and scatting vectors of size dim($\mathbf{y}$))

Recall distribute ADMM I

- decompose $\mathbf{A}$ to row blocks.
- make copies of $\mathbf{x}$
- parallel computing $+$ MPI (gathering and scatting vectors of size dim($\mathbf{x}$))

Between I and II, which is better?

- If $\mathbf{A}$ is fat
  - column decomposition in approach II is more efficient
  - the global variable of approach II is smaller
- If $\mathbf{A}$ is tall
  - row decomposition in approach I is more efficient
  - the global variable of approach I is smaller

Distributed ADMM II

A formulation with separable f and separable g

$$\min \sum_{j=1}^{N} f_j(\mathbf{x}_j) + \sum_{i=1}^{M} g_i(\mathbf{z}_i), \text{ s.t. } \mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{b},$$

where

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N), \mathbf{z} = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_M).$$

Decompose $\mathbf{A}$ in both directions as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1N} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2N} \\ & & \cdots & \\ \mathbf{A}_{M1} & \mathbf{A}_{M2} & \cdots & \mathbf{A}_{MN} \end{bmatrix}, also\ \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_M \end{bmatrix}.$$

Same model:

$$\min \sum_{j=1}^{N} f_j(\mathbf{x}_j) + \sum_{i=1}^{M} g_i(\mathbf{z}_i), \text{ s.t. } \sum_{j=1}^{N} \mathbf{A}_{ij}\mathbf{x}_j + \mathbf{z}_i = \mathbf{b}_i, i = 1, \cdots, M.$$

Distributed ADMM III

$\mathbf{A}_{ij}\mathbf{x}_j'$s are coupled in the constraints. Standard treatment:

$$\mathbf{p}_{ij} = \mathbf{A}_{ij}\mathbf{x}_j.$$

New model:

$$\min \sum_{j=1}^{N} f_j(\mathbf{x}_j) + \sum_{i=1}^{M} g_i(\mathbf{z}_i), \text{ s.t. } \begin{array}{l} \sum_{j=1}^{N} \mathbf{p}_{ij} + \mathbf{z}_i = \mathbf{b}_i, \forall i, \\ \mathbf{p}_{ij} - \mathbf{A}_{ij}\mathbf{x}_j = 0, \forall i, j. \end{array}$$

ADMM

- alternate between $\{\mathbf{p}_{ij}\}$ and $(\{\mathbf{x}_j\}, \{\mathbf{z}_i\})$
- $\mathbf{p}_{ij}-$subproblems have closed-form solutions
- $(\{\mathbf{x}_j\}, \{\mathbf{z}_i\})$-subproblem are separable over all $\mathbf{x}_j$ and $\mathbf{z}_i$
  - $\mathbf{x}_j-$update involves f_j and $\mathbf{A}_{1j}^T\mathbf{A}_{1j}, \cdots, \mathbf{A}_{Mj}^T\mathbf{A}_{Mj}$;
  - $\mathbf{z}_i-$update involves g_i.
- ready for distributed implementation

Question: how to further decouple f_j and $\mathbf{A}_{1j}^T\mathbf{A}_{1j}, \cdots, \mathbf{A}_{Mj}^T\mathbf{A}_{Mj}$?

Distributed ADMM IV

For each $\mathbf{x}_j$, make M identical copies:$\mathbf{x}_{1j}, \mathbf{x}_{2j}, \cdots, \mathbf{x}_{Mj}$.
New model:

$$\min \sum_{j=1}^{N} f_j(\mathbf{x}_j) + \sum_{i=1}^{M} g_i(\mathbf{z}_i), \text{ s.t. } \begin{array}{ll} \sum_{j=1}^{N} \mathbf{p}_{ij} + \mathbf{z}_i = \mathbf{b}_i, & \forall i, \\ \mathbf{p}_{ij} - \mathbf{A}_{ij}\mathbf{x}_{ij} = \mathbf{0}, & \forall i, j, \\ \mathbf{x}_j - \mathbf{x}_{ij} = \mathbf{0}, & \forall i, j. \end{array}$$

ADMM

- alternate between $(\{\mathbf{x}_j\}, \{\mathbf{p}_{ij}\})$ and $(\{\mathbf{x}_j\}, \{\mathbf{z}_i\})$
- $(\{\mathbf{x}_j\}, \{\mathbf{p}_{ij}\})$-subproblem are separable
  - $\mathbf{x}_j-$update involves f_j only; computes prox_{f_j}
  - $\mathbf{p}_{ij}-$update is in closed form
- $(\{\mathbf{x}_{ij}\}, \{\mathbf{z}_i\})$-subproblem are separable
  - $\mathbf{x}_{ij}-$update involves $(\alpha I + \beta \mathbf{A}_{ij}^T \mathbf{A}_{ij})$;
  - $\mathbf{y}_i-$update involves g_i only; computes prox_{g_i}.
- ready for distributed implementation

Outline

Decentralized ADMM

After making local copies $\mathbf{x}_i$ for $\mathbf{x}$, instead of imposing the consistency constraints like

$$\mathbf{x}_i - \mathbf{x} = 0, i = 1, \cdots, M,$$

consider graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$ where $\mathcal{V}$={nodes} and ε={edges}

and impose one type of the following consistency constraints

$$\begin{aligned} \mathbf{x}_i - \mathbf{x}_j = \mathbf{0}, \quad & \forall (i,j) \in \varepsilon, \quad \text{or} \\ \mathbf{x}_i - \mathbf{z}_{ij} = \mathbf{0}, \mathbf{x}_j - \mathbf{z}_{ij} = \mathbf{0} \quad & \forall (i,j) \in \varepsilon, \quad \text{or} \\ \text{mean}\{\mathbf{x}_j : (i,j) \in \varepsilon\} - \mathbf{x}_i = \mathbf{0}, \quad & \forall i \in \mathcal{V}. \end{aligned}$$

Decentralized ADMM

- Decentralized ADMM run on a connected network
- There is no data fusion / control center
- Applications:
  - wireless sensor networks
  - collaborative learning
- ADMM will alternative perform the followings
  - Local computation at each node
  - Communication between neighbors or broadcasting in neighborhood
- Since data is not shared or centrally store, data security is preserved
- Convergence rate depends on
  - the properties (e.g., convexity, condition number) of the objective function
  - the size, connectivity, and spectral properties of the graph

Outline

Example: latent variable graphical model selection

V. Chandrasekaran, P.Parrilo, A. Willsky

Model of regularized maximum normal likelihood

$$\min_{R,S,L} \langle R, \hat{\Sigma}_X \rangle - \text{log det}(R) + \alpha \|S\|_1 + \beta Tr(L), \text{ s.t. } R = S - L, R \succ 0, L \succeq 0,$$

where X are the observed variables, $\Sigma_X^{-1} \approx R = S - L$, S is spare, L is low rank. First two terms are from the log-likelihood function

$$l(K; \Sigma) = \text{log det}(K) - \text{tr}(K\Sigma).$$

Introduce indicator function

$$\mathcal{I}(L \succeq 0) := \begin{cases} 0, & \textit{if } L \succeq 0 \\ +\infty, & \textit{otherwise}. \end{cases}$$

Obtain the 3-block formulation

$$\min_{R,S,L} \langle R, \hat{\Sigma}_X \rangle - \text{log det}(R) + \alpha \|S\|_1 + \beta \text{Tr}(L) + \mathcal{I}(L \succeq 0), \text{ s.t. } R - S + L = 0.$$

Example: stable principle component pursuit

Model

$$\begin{aligned} \min_{L,S,Z} \quad & \|L\|_* + \rho\|S\|_1 \\ \text{s.t.} \quad & L + S + Z = M \\ & \|Z\|_F \le \sigma, \end{aligned}$$

$$M = \text{low-rank + sparse + noise.}$$

For quantities such as images and videos, add $L \ge 0$ component wise.
New model:

$$\begin{aligned} \min_{L,S,Z,K} \quad & \|L\|_* + \rho\|S\|_1 + \mathcal{I}(\|Z\|_F \le \sigma) + \mathcal{I}(K \ge 0) \\ \text{s.t.} \quad & L + S + Z = M \\ & L - K = 0. \end{aligned}$$

Block-form constraints:

$$\begin{pmatrix} I & I \\ I & 0 \end{pmatrix} \begin{pmatrix} L \\ S \end{pmatrix} + \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} \begin{pmatrix} Z \\ K \end{pmatrix} = \begin{pmatrix} M \\ 0 \end{pmatrix}.$$

Example: mixed TV and l_1 regularization

Model

$$\min_x \ TV(x) + \alpha\|Wx\|_1, \text{ s.t. } \|Rx - b\|_2 \le \sigma.$$

New model:

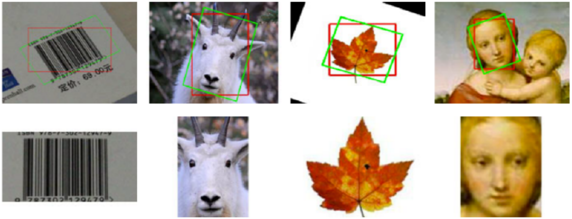
$$\begin{aligned} \min_x \quad & \Sigma_i \|z_i\|_2 + \alpha\|Wx\|_1 + \mathcal{I}(\|y\|_2 \le \sigma) \\ \text{s.t.} \quad & z_i = D_i x, \forall i = 1, \cdots, N \\ & y = Rx - b. \end{aligned}$$

If use two sets of variables, x vs $(y, \{z_i\})$

$$\begin{pmatrix} R \\ D_1 \\ \vdots \\ D_N \end{pmatrix} x - \begin{pmatrix} y \\ z_1 \\ \vdots \\ z_N \end{pmatrix} = \begin{pmatrix} b \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$x-$subproblem is not easy to solve.

Example: alignment for linearly correlated images



Model:

$$\min_{I^0,E,\tau} \|I^0\|_* + \lambda\|E\|_1, \text{ s.t. } I \circ \tau = I^0 + E$$

Linearize the non-convex term $I \circ \tau : I \circ (\tau + \delta\tau) \approx I \circ \tau + \nabla I \cdot \Delta\tau$.
New model

$$\min_{I^0,E,\Delta\tau} \|I^0\|_* + \lambda\|E\|_1, \text{ s.t. } I \circ \tau + \nabla I \Delta\tau = I^0 + E$$

Two solutions to decouple variables

To solve a subproblem with coupling variables
1. apply the prox-linear inexact update, or
2. introduce bridge variables, as done in distributed ADMM.

For example, consider

$$\min_{\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}} (f_1(\mathbf{x}_1)+f_2(\mathbf{x}_2))+g(\mathbf{y}), \text{ s.t. } (\mathbf{A}_1\mathbf{x}_1+\mathbf{A}_2\mathbf{x}_2)+\mathbf{B}\mathbf{y}=\mathbf{b}.$$

In the ADMM $(\mathbf{x}_1,\mathbf{x}_2)-$subproblem,$\mathbf{x}_1$ and $\mathbf{x}_2$ are coupled.
However, the prox-linear update is separable

$$\min_{\mathbf{x}_1,\mathbf{x}_2}(f_1(\mathbf{x}_1)+f_2(\mathbf{x}_2))+\left\langle \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \right\rangle + \frac{1}{2t}\left\| \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{x}_1^k \\ \mathbf{x}_2^k \end{bmatrix} \right\|_2^2.$$

Outline

非凸约束问题

考虑如下约束优化问题：

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}), \\ s.t. \quad & \boldsymbol{x} \in \mathcal{S}, \end{aligned}$$

其中f是凸的，但是$\mathcal{S}$是非凸的。可以将上述问题改写为：

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) + \mathbb{I}_{\mathcal{S}}(\boldsymbol{z}), \\ s.t. \quad & \boldsymbol{x} - \boldsymbol{z} = \mathbf{0}, \end{aligned}$$

交替方向乘子法产生如下迭代：

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \operatorname*{argmin}_{\boldsymbol{x}} \left(f(\boldsymbol{x}) + (\rho/2)\|\boldsymbol{x} - \boldsymbol{z}^k + \boldsymbol{u}^k\|_2^2 \right), \\ \boldsymbol{z}^{k+1} &= \Pi_{\mathcal{S}}(\boldsymbol{x}^{k+1} + \boldsymbol{u}^k), \\ \boldsymbol{u}^{k+1} &= \boldsymbol{u}^k + (\boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}) \end{aligned}$$

其中，$\Pi_{\mathcal{S}}(\boldsymbol{z})$是将$\boldsymbol{z}$投影到集合$\mathcal{S}$中。因为$f$是凸的，所以上述$\boldsymbol{x}$-极小化步是凸问题，但是$\boldsymbol{z}$-极小化步是向一个非凸集合的投影。

一般来说，这种投影很难计算，但是在下面列出的这些特殊情形中可以精确求解。

- 基数：如果$\mathcal{S}=\{\boldsymbol{x}|\boldsymbol{card}(\boldsymbol{x})\leq c\}$，其中$\boldsymbol{card}(\boldsymbol{v})$表示非零元素的数目，那么$\Pi_{\mathcal{S}}(\boldsymbol{v})$保持前$c$大的元素不变，其他元素变为0。
  例如回归选择（也叫特征选择）问题：

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2, \\ s.t. \quad & \boldsymbol{card}(\boldsymbol{x})\leq c. \end{aligned}$$

- 秩：如果$\mathcal{S}$是秩为c的矩阵的集合，那么$\boldsymbol{card}(\boldsymbol{V})$可以通过对$\boldsymbol{V}$做奇异值分解，$\boldsymbol{V}=\sum_i \sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^T$，然后保留前$c$大的奇异值及奇异向量，即$\Pi_{\mathcal{S}}(\boldsymbol{V})=\sum_{i=1}^{c}\sigma_i \boldsymbol{u}_i \boldsymbol{u}_i^T$。
- 布尔约束：如果$\mathcal{S}=\{\boldsymbol{x}|\boldsymbol{x}_i\in\{0,1\}\}$，那么$\Pi_{\mathcal{S}}(\boldsymbol{v})$就是简单地把每个元素变为0和1中离它更近的数。

非负矩阵分解和补全

非负矩阵分解和补全问题可以写成如下形式：

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{Y}} \quad & \|\mathcal{P}_\Omega(\boldsymbol{X}\boldsymbol{Y}-\boldsymbol{M})\|_F^2, \\ s.t. \quad & \boldsymbol{X}_{ij} \geq 0, \boldsymbol{Y}_{ij} \geq 0, \forall i,j, \end{aligned}$$

其中，Ω表示矩阵$\boldsymbol{M}$中的已知元素的下标集合，$\mathcal{P}_\Omega(\boldsymbol{A})$表示得到一个新的矩阵$\boldsymbol{A}'$，其下标在集合$\Omega$中的所对应的元素等于矩阵$\boldsymbol{A}$的对应元素，其下标不在集合$\Omega$中的所对应的元素为0。注意到，这个问题是非凸的。

为了利用交替方向乘子法的优势，我们考虑如下的等价形式：

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}} \quad & \frac{1}{2}\|\boldsymbol{X}\boldsymbol{Y}-\boldsymbol{Z}\|_F^2, \\ s.t. \quad & \boldsymbol{X}=\boldsymbol{U}, \boldsymbol{Y}=\boldsymbol{V}, \\ & \boldsymbol{U} \geq 0, \boldsymbol{V} \geq 0, \\ & \mathcal{P}_\Omega(\boldsymbol{Z}-\boldsymbol{M}) = 0. \end{aligned}$$

$$
\begin{aligned}
L_{\alpha,\beta}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z},\boldsymbol{U},\boldsymbol{V},\boldsymbol{\Lambda},\boldsymbol{\Pi}) =& \frac{1}{2}\|\boldsymbol{X}\boldsymbol{Y}-\boldsymbol{Z}\|_F^2 + \boldsymbol{\Lambda}\bullet(\boldsymbol{X}-\boldsymbol{U}) \\
&+ \boldsymbol{\Pi}\bullet(\boldsymbol{Y}-\boldsymbol{V}) + \frac{\alpha}{2}\|\boldsymbol{X}-\boldsymbol{U}\|_F^2 + \frac{\beta}{2}\|\boldsymbol{Y}-\boldsymbol{V}\|_F^2,
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{X}^{k+1} &= \mathop{\mathrm{argmin}}_{\boldsymbol{X}} \ L_{\alpha,\beta}(\boldsymbol{X},\boldsymbol{Y}^k,\boldsymbol{Z}^k,\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{\Lambda}^k,\boldsymbol{\Pi}^k), \\
\boldsymbol{Y}^{k+1} &= \mathop{\mathrm{argmin}}_{\boldsymbol{Y}} \ L_{\alpha,\beta}(\boldsymbol{X}^{k+1},\boldsymbol{Y},\boldsymbol{Z}^k,\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{\Lambda}^k,\boldsymbol{\Pi}^k), \\
\boldsymbol{Z}^{k+1} &= \mathop{\mathrm{argmin}}_{\mathcal{P}_\Omega(\boldsymbol{Z}-\boldsymbol{M})=0} \ L_{\alpha,\beta}(\boldsymbol{X}^{k+1},\boldsymbol{Y}^{k+1},\boldsymbol{Z},\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{\Lambda}^k,\boldsymbol{\Pi}^k), \\
\boldsymbol{U}^{k+1} &= \mathop{\mathrm{argmin}}_{\boldsymbol{U}\geq 0} \ L_{\alpha,\beta}(\boldsymbol{X}^{k+1},\boldsymbol{Y}^{k+1},\boldsymbol{Z}^{k+1},\boldsymbol{U},\boldsymbol{V}^k,\boldsymbol{\Lambda}^k,\boldsymbol{\Pi}^k), \\
\boldsymbol{V}^{k+1} &= \mathop{\mathrm{argmin}}_{\boldsymbol{V}\geq 0} \ L_{\alpha,\beta}(\boldsymbol{X}^{k+1},\boldsymbol{Y}^{k+1},\boldsymbol{Z}^{k+1},\boldsymbol{U}^{k+1},\boldsymbol{V},\boldsymbol{\Lambda}^k,\boldsymbol{\Pi}^k), \\
\Lambda^{k+1} &= \Lambda^k + \tau\alpha(\boldsymbol{X}^{k+1}-\boldsymbol{U}^{k+1}), \\
\Pi^{k+1} &= \Pi^k + \tau\beta(\boldsymbol{Y}^{k+1}-\boldsymbol{V}^{k+1}).
\end{aligned}
$$