Lecture on ADMM

Acknowledgement: this slides is based on Prof. Wotao Yin's lecture notes

Outline

Dual problem

- convex equality constrained optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

- Lagrangian: $L(x, y) = f(x) + y^T(Ax - b)$
- dual function: $g(y) = \inf_x L(x, y)$
- dual problem: maximize $g(y)$
- recover $x^\star = \operatorname{argmin}_x L(x, y^\star)$

Dual ascent

- gradient method for dual problem: $y^{k+1} = y^k + \alpha^k \nabla g(y^k)$
- $\nabla g(y^k) = A\tilde{x} - b$, where $\tilde{x} = \operatorname{argmin}_x L(x, y^k)$
- dual ascent method is

$$
\begin{array}{lcll}
x^{k+1} & := & \operatorname{argmin}_x L(x, y^k) & \textit{// } x\textit{-minimization} \\
y^{k+1} & := & y^k + \alpha^k (Ax^{k+1} - b) & \textit{// dual update}
\end{array}
$$

- works, with lots of strong assumptions

Dual decomposition

- suppose f is separable:

$$f(x) = f_1(x_1) + \cdots + f_N(x_N), \quad x = (x_1, \ldots, x_N)$$

- then L is separable in x: $L(x, y) = L_1(x_1, y) + \cdots + L_N(x_N, y) - y^T b$,

$$L_i(x_i, y) = f_i(x_i) + y^T A_i x_i$$

- x-minimization in dual ascent splits into N separate minimizations

$$x_i^{k+1} \quad := \quad \underset{x_i}{\operatorname{argmin}}\, L_i(x_i, y^k)$$

  which can be carried out in parallel

Dual decomposition

- dual decomposition (Everett, Dantzig, Wolfe, Benders 1960–65)

$$\begin{aligned} x_i^{k+1} &:= \operatorname{argmin}_{x_i} L_i(x_i, y^k), \quad i = 1, \ldots, N \\ y^{k+1} &:= y^k + \alpha^k (\textstyle\sum_{i=1}^N A_i x_i^{k+1} - b) \end{aligned}$$

- scatter y^k; update x_i in parallel; gather $A_i x_i^{k+1}$

- solve a large problem
  - by iteratively solving subproblems (in parallel)
  - dual variable update provides coordination

- works, with lots of assumptions; often slow

Outline

Method of multipliers

- a method to robustify dual ascent
- use **augmented Lagrangian** (Hestenes, Powell 1969), $\rho > 0$

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + (\rho/2)\|Ax - b\|_2^2$$

- method of multipliers (Hestenes, Powell; analysis in Bertsekas 1982)

$$\begin{aligned} x^{k+1} &:= \operatorname*{argmin}_x L_\rho(x, y^k) \\ y^{k+1} &:= y^k + \rho(Ax^{k+1} - b) \end{aligned}$$

(note specific dual update step length ρ)

Method of multipliers dual update step

- optimality conditions (for differentiable f):

$$Ax^\star - b = 0, \qquad \nabla f(x^\star) + A^T y^\star = 0$$

  (primal and dual feasibility)

- since x^{k+1} minimizes $L_\rho(x, y^k)$

$$\begin{array}{rcl} 0 & = & \nabla_x L_\rho(x^{k+1}, y^k) \\ & = & \nabla_x f(x^{k+1}) + A^T \left(y^k + \rho(Ax^{k+1} - b)\right) \\ & = & \nabla_x f(x^{k+1}) + A^T y^{k+1} \end{array}$$

- dual update $y^{k+1} = y^k + \rho(x^{k+1} - b)$ makes (x^{k+1}, y^{k+1}) *dual feasible*
- *primal feasibility* achieved in limit: $Ax^{k+1} - b \to 0$

Method of multipliers

(compared to dual decomposition)

- *good news*: converges under much more relaxed conditions (f can be nondifferentiable, take on value $+\infty$, ...)

- *bad news*: quadratic penalty destroys splitting of the x-update, so can't do decomposition

Outline

典型问题形式

考虑如下凸问题：

$$\begin{aligned} \min_{x_1,x_2} \quad & f_1(x_1)+f_2(x_2), \\ \text{s.t.} \quad & A_1x_1+A_2x_2=b, \end{aligned} \tag{1}$$

- f_1, f_2是适当的闭凸函数，但不要求是光滑的，$x_1 \in \mathbb{R}^n, x_2 \in \mathbb{R}^m$, $A_1 \in \mathbb{R}^{p\times n}, A_2 \in \mathbb{R}^{p\times m}$, $b \in \mathbb{R}^p$.
- 问题特点：目标函数可以分成彼此分离的两块，但是变量被线性约束结合在一起．常见的一些无约束和带约束的优化问题都可以表示成这一形式。

问题形式举例

- 可以分成两块的无约束优化问题

$$\min_{x} \quad f_1(x) + f_2(x).$$

引入一个新的变量z并令$x = z$，将问题转化为

$$\begin{aligned} \min_{x,z} \quad & f_1(x) + f_2(z), \\ \text{s.t.} \quad & x - z = 0. \end{aligned}$$

- 带线性变换的无约束优化问题

$$\min_{x} \quad f_1(x) + f_2(Ax).$$

可以引入一个新的变量z，令$z = Ax$，则问题变为

$$\begin{aligned} \min_{x,z} \quad & f_1(x) + f_2(z), \\ \text{s.t.} \quad & Ax - z = 0. \end{aligned}$$

- 凸集$C \subset \mathbb{R}^n$上的约束优化问题

$$\begin{aligned} \min_x \quad & f(x), \\ \text{s.t.} \quad & Ax \in C, \end{aligned}$$

$I_C(z)$是集合C的示性函数，引入约束$z = Ax$，那么问题转化为

$$\begin{aligned} \min_{x,z} \quad & f(x) + I_C(z), \\ \text{s.t.} \quad & Ax - z = 0. \end{aligned}$$

- 全局一致性问题

$$\min_x \quad \sum_{i=1}^N \phi_i(x).$$

令$x = z$，并将x复制N份，分别为x_i，那么问题转化为

$$\begin{aligned} \min_{x_i,z} \quad & \sum_{i=1}^N \phi_i(x_i), \\ \text{s.t.} \quad & x_i - z = 0,\ i = 1, 2, \cdots, N. \end{aligned}$$

增广拉格朗日函数法

- 首先写出问题(1)的增广拉格朗日函数

$$
\begin{aligned}
L_\rho(x_1,x_2,y) =& f_1(x_1)+f_2(x_2)+y^{\mathrm{T}}(A_1x_1+A_2x_2-b) \\
&+\frac{\rho}{2}\|A_1x_1+A_2x_2-b\|_2^2,
\end{aligned} \tag{2}
$$

  其中$\rho>0$是二次罚项的系数.

- 常见的求解带约束问题的增广拉格朗日函数法为如下更新:

$$
(x_1^{k+1},x_2^{k+1}) = \operatorname*{argmin}_{x_1,x_2} L_\rho(x_1,x_2,y^k), \tag{3}
$$

$$
y^{k+1} = y^k+\tau\rho(A_1x_1^{k+1}+A_2x_2^{k+1}-b), \tag{4}
$$

  其中τ为步长.

Alternating direction method of multipliers, ADMM

- 交替方向乘子法的基本思路: 第一步迭代(3)同时对x_1和x_2进行优化有时候比较困难，而固定一个变量求解关于另一个变量的极小问题可能比较简单，因此我们可以考虑对x_1和x_2交替求极小
- 其迭代格式可以总结如下：

$$x_1^{k+1} = \operatorname*{argmin}_{x_1} L_\rho(x_1, x_2^k, y^k), \tag{5}$$

$$x_2^{k+1} = \operatorname*{argmin}_{x_2} L_\rho(x_1^{k+1}, x_2, y^k), \tag{6}$$

$$y^{k+1} = y^k + \tau\rho(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b), \tag{7}$$

其中τ 为步长，通常取值于$\left(0, \frac{1+\sqrt{5}}{2}\right]$

原问题最优性条件

- 因为f_1,f_2均为闭凸函数，约束为线性约束，所以当Slater 条件成立时，可以使用凸优化问题的KKT 条件来作为交替方向乘子法的收敛准则．问题(1)的拉格朗日函数为

$$L(x_1,x_2,y)=f_1(x_1)+f_2(x_2)+y^{\mathrm{T}}(A_1x_1+A_2x_2-b).$$

- 根据最优性条件定理，若x_1^*,x_2^*为问题(1)的最优解，y^*为对应的拉格朗日乘子，则以下条件满足：

$$0\in\partial_{x_1}L(x_1^*,x_2^*,y^*)=\partial f_1(x_1^*)+A_1^{\mathrm{T}}y^*, \tag{8a}$$
$$0\in\partial_{x_2}L(x_1^*,x_2^*,y^*)=\partial f_2(x_2^*)+A_2^{\mathrm{T}}y^*, \tag{8b}$$
$$A_1x_1^*+A_2x_2^*=b. \tag{8c}$$

在这里条件(8c)又称为原始可行性条件，条件(8a)和条件(8b)又称为对偶可行性条件．

ADMM单步迭代最优性条件

- 由x_2的更新步骤

$$x_2^k = \mathop{\mathrm{argmin}}_x \left\{ f_2(x) + \frac{\rho}{2} \left\| A_1 x_1^k + A_2 x - b + \frac{y^{k-1}}{\rho} \right\|^2 \right\},$$

根据最优性条件不难推出

$$0 \in \partial f_2(x_2^k) + A_2^{\mathrm{T}}[y^{k-1} + \rho(A_1 x_1^k + A_2 x_2^k - b)]. \tag{9}$$

当$\tau = 1$时，根据(7)可知上式方括号中的表达式就是y^k，最终有

$$0 \in \partial f_2(x_2^k) + A_2^{\mathrm{T}} y^k,$$

- 由x_1的更新公式

$$x_1^k = \mathop{\mathrm{argmin}}_x \left\{ f_1(x) + \frac{\rho}{2} \| A_1 x + A_2 x_2^{k-1} - b + \frac{y^{k-1}}{\rho} \|^2 \right\},$$

假设子问题能精确求解，根据最优性条件

$$0 \in \partial f_1(x_1^k) + A_1^{\mathrm{T}}[\rho(A_1 x_1^k + A_2 x_2^{k-1} - b) + y^{k-1}].$$

ADMM单步迭代最优性条件

- 根据ADMM 的第三式(7)取$\tau=1$有

$$0 \in \partial f_1(x_1^k) + A_1^{\mathrm{T}}(y^k + A_2(x_2^{k-1} - x_2^k)). \tag{10}$$

对比条件(8a)可知多出来的项为$A_1^{\mathrm{T}}A_2(x_2^{k-1}-x_2^k)$。因此要检测对偶可行性只需要检测残差

$$s^k = A_1^{\mathrm{T}}A_2(x_2^{k-1} - x_2^k)$$

- 综上当x_2更新取到精确解且$\tau=1$时，判断ADMM 是否收敛只需要检测前述两个残差r^k, s^k是否充分小：

$$\begin{aligned} 0 \approx \|r^k\| &= \|A_1x_1^k + A_2x_2^k - b\| \quad &\text{（原始可行性）}, \\ 0 \approx \|s^k\| &= \|A_1^{\mathrm{T}}A_2(x_2^{k-1} - x_2^k)\| \quad &\text{（对偶可行性）}. \end{aligned} \tag{11}$$

Outline

线性化

- 线性化技巧使用近似点项对子问题目标函数进行二次近似.
- 不失一般性，我们考虑第一个子问题，即

$$\min_{x_1} \quad f_1(x_1) + \frac{\rho}{2}\|A_1x_1 - v^k\|^2, \tag{12}$$

其中$v^k = b - A_2x_2^k - \frac{1}{\rho}y^k$.

- 当子问题目标函数可微时，线性化将问题(12)变为

$$x_1^{k+1} = \operatorname*{argmin}_{x_1}\left\{\left(\nabla f_1(x_1^k) + \rho A_1^{\mathrm{T}}\left(A_1x_1^k - v^k\right)\right)^{\mathrm{T}} x_1 + \frac{1}{2\eta_k}\|x_1 - x^k\|_2^2\right\},$$

其中η_k是步长参数，这等价于做一步梯度下降.

- 当目标函数不可微时，可以考虑只将二次项线性化，即

$$x_1^{k+1} = \operatorname*{argmin}_{x_1}\left\{f(x_1) + \rho\left(A_1^{\mathrm{T}}(A_1x_1^k - v^k)\right)^{\mathrm{T}} x_1 + \frac{1}{2\eta_k}\|x_1 - x^k\|_2^2\right\},$$

这等价于做一步近似点梯度步.

缓存分解

- 如果目标函数中含二次函数，例如$f_1(x_1) = \frac{1}{2}\|Cx_1 - d\|_2^2$，那么针对$x_1$的更新(5)等价于求解线性方程组

$$(C^TC + \rho A_1^T A_1)x_1 = C^Td + \rho A_1^T v^k.$$

- 虽然子问题有显式解，但是每步求解的复杂度仍然比较高，这时候可以考虑用**缓存分解**的方法．首先对$C^TC + \rho A_1^T A_1$进行Cholesky 分解并缓存分解的结果，在每步迭代中只需要求解简单的三角形方程组
- 当ρ发生更新时，就要重新进行分解．特别地，当$C^TC + \rho A_1^T A_1$一部分容易求逆，另一部分是低秩的情形时，可以用SMW公式来求逆．

优化转移

- 有时候为了方便求解子问题，可以用一个性质好的矩阵D近似二次项$A_1^{\mathrm{T}}A_1$，此时子问题(12)替换为

$$
\begin{aligned}
x_1^{k+1} \quad = \quad & \operatorname*{argmin}_{x_1} \left\{ f_1(x_1) + \frac{\rho}{2}\|A_1x_1 - v^k\|_2^2 \right. \\
& \left. + \frac{\rho}{2}(x_1 - x^k)^{\mathrm{T}}(D - A_1^{\mathrm{T}}A_1)(x_1 - x^k) \right\}.
\end{aligned}
$$

这种方法也称为**优化转移**.

- 通过选取合适的D，当计算$\operatorname*{argmin}_{x_1} \left\{ f_1(x_1) + \frac{\rho}{2}x_1^{\mathrm{T}}Dx_1 \right\}$明显比计算$\operatorname*{argmin}_{x_1}\{f_1(x_1) + \frac{\rho}{2}x_1^{\mathrm{T}}A_1^{\mathrm{T}}A_1x_1\}$要容易时，优化转移可以极大地简化子问题的计算．特别地，当$D = \frac{\eta_k}{\rho}I$时，优化转移等价于做单步的近似点梯度步.

二次罚项系数的动态调节

- 原始可行性和对偶可行性分别用$\|r^k\|$和$\|s^k\|$度量.
- 求解过程中二次罚项系数ρ太大会导致原始可行性$\|r^k\|$下降很快，但是对偶可行性$\|s^k\|$下降很慢；二次罚项系数太小，则会有相反的效果．这样都会导致收敛比较慢或得到的解的可行性很差．
- 一个自然的想法是在每次迭代时动态调节惩罚系数ρ的大小，从而使得原始可行性和对偶可行性能够以比较一致的速度下降到零．一个简单有效的方式是令

$$
\rho^{k+1} = \begin{cases} \gamma_p \rho^k, & \|r^k\| > \mu\|s^k\|, \\ \dfrac{\rho^k}{\gamma_d} & \|s^k\| > \mu\|r^k\|, \\ \rho^k, & \text{其他}, \end{cases}
$$

  其中$\mu > 1, \gamma_p > 1, \gamma_d > 1$是参数．常见的选择为$\mu = 10, \gamma_p = \gamma_d = 2$．在迭代过程中将原始可行性$\|r^k\|$和对偶可行性$\|s^k\|$保持在彼此的$\mu$倍内．如果发现$\|r^k\|$或$\|s^k\|$下降过慢就应该相应增大或减小二次罚项系数$\rho^k$．

超松弛

- 在(6)式与(7)式中，$A_1x_1^{k+1}$可以被替换为

$$\alpha_k A_1 x_1^{k+1} + (1-\alpha_k)(A_2 x_2^k - b),$$

  其中$\alpha_k \in (0,2)$是一个松弛参数.
- 当$\alpha_k > 1$时，这种技巧称为超松弛；当$\alpha_k < 1$时，这种技巧称为欠松弛. 实验表明$\alpha_k \in [1.5, 1.8]$的超松弛可以提高收敛速度.

多块问题的ADMM

- 考虑有多块变量的情形

$$\begin{aligned} \min_{x_1,x_2,\cdots,x_N} \quad & f_1(x_1)+f_2(x_2)+\cdots+f_N(x_N), \\ \text{s.t.} \quad & A_1x_1+A_2x_2+\cdots+A_Nx_N=b. \end{aligned} \tag{13}$$

这里$f_i(x_i)$是闭凸函数，$x_i \in \mathbb{R}^{n_i}, A_i \in \mathbb{R}^{m\times n_i}$．

- 同样写出增广拉格朗日函数$L_\rho(x_1,x_2,\cdots,x_N,y)$，相应的多块ADMM 迭代格式为

$$\begin{aligned} x_1^{k+1} &= \operatorname*{argmin}_x L_\rho(x,x_2^k,\cdots,x_N^k,y^k), \\ x_2^{k+1} &= \operatorname*{argmin}_x L_\rho(x_1^{k+1},x,\cdots,x_N^k,y^k), \\ &\cdots\cdots\cdots\cdots\cdots \\ x_N^{k+1} &= \operatorname*{argmin}_x L_\rho(x_1^{k+1},x_2^{k+1},\cdots,x,y^k), \\ y^{k+1} &= y^k+\tau\rho(A_1x_1^{k+1}+A_2x_2^{k+1}+\cdots+A_Nx_N^{k+1}-b), \end{aligned}$$

其中$\tau \in \left(0,\frac{1}{2}(\sqrt{5}+1)\right)$为步长参数．

Outline

LASSO 问题的Primal 形式

- LASSO 问题

$$\min \quad \mu\|x\|_1 + \frac{1}{2}\|Ax-b\|^2.$$

转换为标准问题形式：

$$\begin{aligned}\min_{x,z} \quad & \frac{1}{2}\|Ax-b\|^2 + \mu\|z\|_1, \\ \text{s.t.} \quad & x = z.\end{aligned}$$

- 交替方向乘子法迭代格式为

$$\begin{aligned}x^{k+1} &= \operatorname*{argmin}_x \left\{\frac{1}{2}\|Ax-b\|^2 + \frac{\rho}{2}\|x - z^k + \frac{1}{\rho}y^k\|_2^2\right\}, \\ &= (A^\mathrm{T}A + \rho I)^{-1}(A^\mathrm{T}b + \rho z^k - y^k), \\ z^{k+1} &= \operatorname*{argmin}_z \left\{\mu\|z\|_1 + \frac{\rho}{2}\|x^{k+1} - z + \frac{1}{\rho}y^k\|^2\right\}, \\ &= \operatorname{prox}_{(\mu/\rho)\|\cdot\|_1}\left(x^{k+1} + \frac{1}{\rho}y^k\right), \\ y^{k+1} &= y^k + \tau\rho(x^{k+1} - z^{k+1}).\end{aligned}$$

LASSO 问题的Primal 形式

- 注意，因为$\rho > 0$，所以$A^\mathrm{T}A + \rho I$总是可逆的．x迭代本质上是计算一个岭回归问题（ℓ_2范数平方正则化的最小二乘问题）；而对z的更新为ℓ_1范数的邻近算子，同样有显式解．在求解x迭代时，若使用固定的罚因子ρ，我们可以缓存矩阵$A^\mathrm{T}A + \rho I$的初始分解，从而减小后续迭代中的计算量．

- 需要注意的是，在LASSO 问题中，矩阵$A \in \mathbb{R}^{m\times n}$通常有较多的列（即$m \ll n$），因此$A^\mathrm{T}A \in \mathbb{R}^{n\times n}$是一个低秩矩阵，二次罚项的作用就是将$A^\mathrm{T}A$增加了一个正定项．该ADMM 主要运算量来自更新$x$变量时求解线性方程组，复杂度为$O(n^3)$（若使用缓存分解技术或SMW 公式则可进一步降低每次迭代的运算量）

LASSO 问题的对偶形式

- 考虑LASSO 问题的对偶问题

$$\begin{aligned} \min \quad & b^{\mathrm{T}}y + \tfrac{1}{2}\|y\|^2, \\ \text{s.t.} \quad & \|A^{\mathrm{T}}y\|_\infty \le \mu. \end{aligned} \tag{14}$$

- 引入约束$A^{\mathrm{T}}y + z = 0$，可以得到如下等价问题：

$$\begin{aligned} \min \quad & \underbrace{b^{\mathrm{T}}y + \frac{1}{2}\|y\|^2}_{f(y)} + \underbrace{I_{\|z\|_\infty \le \mu}(z)}_{h(z)}, \\ \text{s.t.} \quad & A^{\mathrm{T}}y + z = 0. \end{aligned} \tag{15}$$

- 对约束$A^{\mathrm{T}}y + z = 0$引入乘子x，对偶问题的增广拉格朗日函数为

$$L_\rho(y, z, x) = b^{\mathrm{T}}y + \frac{1}{2}\|y\|^2 + I_{\|z\|_\infty \le \mu}(z) - x^{\mathrm{T}}(A^{\mathrm{T}}y + z) + \frac{\rho}{2}\|A^{\mathrm{T}}y + z\|^2.$$

LASSO 问题的对偶形式

- 当固定y,x时，对z的更新即向无穷范数球$\{z \mid \|z\|_\infty \le \mu\}$做欧几里得投影，即将每个分量截断在区间$[-\mu,\mu]$中；当固定$z,x$时，对$y$的更新即求解线性方程组

$$(I+\rho AA^{\mathrm{T}})y = A(x^k - \rho z^{k+1}) - b.$$

- 因此得到ADMM 迭代格式为

$$\begin{aligned}
z^{k+1} &= \mathcal{P}_{\|z\|_\infty \le \mu}\left(\frac{x^k}{\rho} - A^{\mathrm{T}}y^k\right),\\
y^{k+1} &= (I+\rho AA^{\mathrm{T}})^{-1}\left(A(x^k - \rho z^{k+1}) - b\right),\\
x^{k+1} &= x^k - \tau\rho(A^{\mathrm{T}}y^{k+1} + z^{k+1}).
\end{aligned}$$

- 虽然ADMM 应用于对偶问题也需要求解一个线性方程组，但由于LASSO 问题的特殊性（$m \ll n$），求解y更新的线性方程组需要的计算量是$O(m^3)$，使用缓存分解技巧后可进一步降低至$O(m^2)$，这大大小于针对原始问题的ADMM．

广义LASSO 问题

- 对许多问题x 本身不稀疏，但在某种变换下是稀疏的:

$$\min_x \quad \mu\|Fx\|_1 + \frac{1}{2}\|Ax - b\|^2. \tag{16}$$

- 一个重要的例子是当$F \in \mathbb{R}^{(n-1)\times n}$ 是一阶差分矩阵

$$F_{ij} = \begin{cases} 1, & j = i+1, \\ -1, & j = i, \\ 0, & \text{其他}, \end{cases}$$

且$A = I$时，广义LASSO问题为

$$\min_x \quad \frac{1}{2}\|x - b\|^2 + \mu \sum_{i=1}^{n-1} |x_{i+1} - x_i|,$$

这个问题就是图像去噪问题的TV 模型；当$A = I$且F是二阶差分矩阵时，问题(16)被称为一范数趋势滤波.

- 通过引入约束$Fx=z$:

$$\begin{aligned}\min_{x,z} \quad & \frac{1}{2}\|Ax-b\|^2+\mu\|z\|_1, \\ \text{s.t.} \quad & Fx-z=0,\end{aligned} \tag{17}$$

- 引入乘子y，其增广拉格朗日函数为

$$L_\rho(x,z,y)=\frac{1}{2}\|Ax-b\|^2+\mu\|z\|_1+y^{\mathrm{T}}(Fx-z)+\frac{\rho}{2}\|Fx-z\|^2.$$

- 此问题的x迭代是求解方程组

$$(A^{\mathrm{T}}A+\rho F^{\mathrm{T}}F)x=A^{\mathrm{T}}b+\rho F^{\mathrm{T}}\left(z^k-\frac{y^k}{\rho}\right),$$

而z迭代依然通过ℓ_1范数的邻近算子.

- 因此交替方向乘子法所产生的迭代为

$$
\begin{aligned}
x^{k+1} &= (A^\mathrm{T}A + \rho F^\mathrm{T}F)^{-1}\left(A^\mathrm{T}b + \rho F^\mathrm{T}\left(z^k - \frac{y^k}{\rho}\right)\right),\\
z^{k+1} &= \mathrm{prox}_{(\mu/\rho)\|\cdot\|_1}\left(Fx^{k+1} + \frac{y^k}{\rho}\right),\\
y^{k+1} &= y^k + \tau\rho(Fx^{k+1} - z^{k+1}).
\end{aligned}
$$

- 对于全变差去噪问题，$A^\mathrm{T}A + \rho F^\mathrm{T}F$是三对角矩阵，所以此时$x$迭代可以在$\mathcal{O}(n)$的时间复杂度内解决；对于图像去模糊问题，$A$是卷积算子，则利用傅里叶变换可将求解方程组的复杂度降低至$O(n\log n)$；对于一范数趋势滤波问题，$A^\mathrm{T}A + \rho F^\mathrm{T}F$是五对角矩阵，所以$x$迭代仍可以在$\mathcal{O}(n)$的时间复杂度内解决

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

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

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

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

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

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

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

- 该问题的基本形式是

$$\min_{X} \quad \langle S, X\rangle - \ln\det X + \mu\|X\|_1, \tag{18}$$

其中S是已知的对称矩阵，通常由样本协方差矩阵得到．变量$X \in \mathcal{S}^n_{++}$，$\|\cdot\|_1$定义为矩阵所有元素绝对值的和．

- 目标函数由光滑项和非光滑项组成，因此引入约束$X = Z$将问题的两部分分离：

$$\begin{aligned} \min \quad & \underbrace{\langle S, X\rangle - \ln\det X}_{f(X)} + \underbrace{\mu\|Z\|_1}_{h(Z)}, \\ \text{s.t.} \quad & X = Z. \end{aligned}$$

引入乘子U作用在约束$X - Z = 0$上，可得增广拉格朗日函数为

$$L_\rho(X, Z, U) = \langle S, X\rangle - \ln\det X + \mu\|Z\|_1 + \langle U, X - Z\rangle + \frac{\rho}{2}\|X - Z\|_F^2.$$

- 首先，固定Z^k, U^k，则X子问题是凸光滑问题，对X求矩阵导数并令其为零，

$$S - X^{-1} + U^k + \rho(X - Z^k) = 0.$$

这是一个关于X的矩阵方程，可以求出满足上述矩阵方程的唯一正定的X为

$$X^{k+1} = Q\mathrm{Diag}(x_1, x_2, \cdots, x_n)Q^{\mathrm{T}},$$

其中Q包含矩阵$S - \rho Z^k + U^k$的所有特征向量，x_i的表达式为

$$x_i = \frac{-d_i + \sqrt{d_i^2 + 4\rho}}{2\rho},$$

d_i为矩阵$S - \rho Z^k + U^k$的第i个特征值.

- 固定X^{k+1}, U^k，则Z的更新为矩阵ℓ_1 范数的邻近算子.
- 最后是常规的乘子更新.

矩阵分离问题

- 考虑矩阵分离问题：

$$\begin{aligned} \min_{X,S} \quad & \|X\|_* + \mu\|S\|_1, \\ \text{s.t.} \quad & X + S = M, \end{aligned} \tag{19}$$

  其中$\|\cdot\|_1$与$\|\cdot\|_*$分别表示矩阵ℓ_1范数与核范数.

- 引入乘子Y作用在约束$X+S=M$上，我们可以得到此问题的增广拉格朗日函数

$$L_\rho(X,S,Y) = \|X\|_* + \mu\|S\|_1 + \langle Y, X+S-M\rangle + \frac{\rho}{2}\|X+S-M\|_F^2. \tag{20}$$

矩阵分离问题

- 对于X子问题，

$$
\begin{aligned}
X^{k+1} &= \underset{X}{\operatorname{argmin}}\ L_\rho(X, S^k, Y^k) \\
&= \underset{X}{\operatorname{argmin}} \left\{ \|X\|_* + \frac{\rho}{2}\left\| X + S^k - M + \frac{Y^k}{\rho} \right\|_F^2 \right\}, \\
&= \underset{X}{\operatorname{argmin}} \left\{ \frac{1}{\rho}\|X\|_* + \frac{1}{2}\left\| X + S^k - M + \frac{Y^k}{\rho} \right\|_F^2 \right\}, \\
&= U \mathrm{Diag}\left(\mathrm{prox}_{(1/\rho)\|\cdot\|_1}(\sigma(A)) \right) V^{\mathrm{T}},
\end{aligned}
$$

其中$A = M - S^k - \frac{Y^k}{\rho}$，$\sigma(A)$为$A$的所有非零奇异值构成的向量并且$U\mathrm{Diag}(\sigma(A))V^{\mathrm{T}}$为$A$的约化奇异值分解.

- 对于S子问题，

$$
\begin{aligned}
S^{k+1} &= \underset{S}{\operatorname{argmin}}\ L_\rho(X^{k+1}, S, Y^k) \\
&= \underset{S}{\operatorname{argmin}} \left\{ \mu\|S\|_1 + \frac{\rho}{2} \left\| X^{k+1} + S - M + \frac{Y^k}{\rho} \right\|_F^2 \right\} \\
&= \operatorname{prox}_{(\mu/\rho)\|\cdot\|_1} \left(M - X^{k+1} - \frac{Y^k}{\rho} \right).
\end{aligned}
$$

- 那么交替方向乘子法的迭代格式为

$$
\begin{aligned}
X^{k+1} &= U\mathrm{Diag}\Big(\operatorname{prox}_{(1/\rho)\|\cdot\|_1}(\sigma(A))\Big)V^{\mathrm{T}}, \\
S^{k+1} &= \operatorname{prox}_{(\mu/\rho)\|\cdot\|_1} \left(M - L^{k+1} - \frac{Y^k}{\rho} \right), \\
Y^{k+1} &= Y^k + \tau\rho(X^{k+1} + S^{k+1} - M).
\end{aligned}
$$

Image blurring model

$$b = Kx_t + w$$

- x_t is unknown image
- b is observed (blurred and noisy) image; w is noise
- $N \times N$-images are stored in column-major order as vectors of length N^2

blurring matrix K

- represents 2D convolution with space-invariant point spread function
- with periodic boundary conditions, block-circulant with circulant blocks
- can be diagonalized by multiplication with unitary 2D DFT matrix W:

$$K = W^H \mathbf{diag}(\lambda) W$$

equations with coefficient $I + K^T K$ can be solved in $O(N^2 \log N)$ time

Total variation deblurring with 1-norm

$$\begin{array}{ll} \min & \|Kx - b\|_1 + \gamma \|Dx\|_{tv} \\ \text{s.t.} & 0 \le x \le 1 \end{array}$$

second term in objective is **total variation penalty**

- Dx is discretized first derivative in vertical and horizontal direction

$$\left(\begin{array}{c} I \otimes D_1 \\ D_1 \otimes I \end{array} \right), \qquad \left(\begin{array}{ccccccc} -1 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{array} \right)$$

- $\|\cdot\|_{tv}$ is a sum of Euclidean norms: $\|(u,v)\|_{tv} = \sum_{i=1}^{n} \sqrt{u_i^2 + v_i^2}$

Image blurring model by ADMM

Consider an equivalent model by splitting:

$$\min \ \|u\|_1 + \gamma\|v\|_{tv}, \ \text{ s.t. } u = Kx - b, \ v = Dx, y = x, 0 \leq y \leq 1$$

ADMM requires:

- decoupled prox-evaluations of $\|u\|_1$ and $\|v\|_{tv}$, and projections on C
- solution of linear equations with coefficient matrix

$$I + K^T K + D^T D$$

solvable in $O(N^2 \log N)$ time

Image blurring: Example

- 1024×1024 image, periodic boundary conditions
- Gaussian blur
- salt-and-pepper noise (50% pixels randomly changed to 0/1)

original

noisy/blurred

restored

全局一致性优化问题

- 增广拉格朗日函数为

$$L_\rho(x_1,\cdots,x_N,z,y_1,\cdots,y_N)=\sum_{i=1}^N\phi_i(x_i)+\sum_{i=1}^N y_i^{\mathrm{T}}(x_i-z)+\frac{\rho}{2}\sum_{i=1}^N\|x_i-z\|^2.$$

- 固定z^k,y_i^k，更新x_i的公式为

$$x_i^{k+1}=\operatorname*{argmin}_x\left\{\phi_i(x)+\frac{\rho}{2}\left\|x-z^k+\frac{y_i^k}{\rho}\right\|^2\right\}. \tag{21}$$

- 注意，虽然表面上看增广拉格朗日函数有（$N+1$）个变量块，但本质上还是两个变量块．这是因为在更新某x_i时并没有利用其他x_i的信息，所有x_i可以看成一个整体．相应地，所有乘子y_i也可以看成一个整体．

- 迭代式(21)的具体计算依赖于ϕ_i的形式，在一般情况下更新x_i的表达式为

$$x_i^{k+1}=\operatorname{prox}_{\phi_i/\rho}\left(z^k-\frac{y_i^k}{\rho}\right).$$

- 固定x_i^{k+1}, y_i^k，问题关于z是二次函数，因此可以直接写出显式解：

$$z^{k+1} = \frac{1}{N}\sum_{i=1}^{N}\left(x_i^{k+1} + \frac{y_i^k}{\rho}\right).$$

- 综上，该问题的交替方向乘子法迭代格式为

$$\begin{aligned}
x_i^{k+1} &= \mathrm{prox}_{\phi_i/\rho}\left(z^k - \frac{y_i^k}{\rho}\right), \ i = 1, 2, \cdots, N, \\
z^{k+1} &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i^{k+1} + \frac{y_i^k}{\rho}\right), \\
y_i^{k+1} &= y_i^k + \tau\rho(x_i^{k+1} - z^{k+1}), \ i = 1, 2, \cdots, N.
\end{aligned}$$

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

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 I

New model:

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

- $\mathbf{x}_l$'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 I

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 $\mathbf{y}_l^{k+1}, \bar{\mathbf{y}}_l^{k+1}$

Distributed ADMM I

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 II

$\mathbf{A}_{ij}\mathbf{x}_j'$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 III

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

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

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.

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

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

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。

非凸约束问题

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

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

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：

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

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

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

Outline

Douglas-Rachford splitting algorithm

Consider

$$\min \quad f(x) = g(x) + h(x)$$

g and h are closed convex functions

Douglas-Rachford iteration: starting at any $z^{(0)}$, repeat

$$\begin{aligned} x^{(k)} &= \mathrm{prox}_{th}(z^{(k-1)}) \\ y^{(k)} &= \mathrm{prox}_{tg}(2x^{(k)} - z^{(k-1)}) \\ z^{(k)} &= z^{(k-1)} + y^{(k)} - x^{(k)} \end{aligned}$$

- t is a positive constant (simply scales the objective)
- useful when g and h have inexpensive prox-operators
- under weak conditions (existence of a minimizer), $x^{(k)}$ converges

Equivalent form

- start iteration at y-update

$$y^+ = \text{prox}_{tg}(2x - z); \quad z^+ = z + y^+ - x; \quad x^+ = \text{prox}_{th}(z^+)$$

- switch z- and x-updates

$$y^+ = \text{prox}_{tg}(2x - z); \quad x^+ = \text{prox}_{th}(z + y^+ - x); \quad z^+ = z + y^+ - x$$

- make change of variables $w = z - x$

alternate form of DR iteration: start at $x^{(0)} \in \textbf{dom}\, h, w^{(0)} \in t\partial h(x^{(0)})$

$$\begin{aligned} y^+ &= \text{prox}_{tg}(x - w) \\ x^+ &= \text{prox}_{th}(y^+ + w) \\ w^+ &= w + y^+ - x^+ \end{aligned}$$

Interpretation as fixed-point iteration

Douglas-Rachford iteration can be written as

$$z^{(k)} = F(z^{(k-1)})$$

where $F(z) = z + \mathrm{prox}_{tg}(2\mathrm{prox}_{th}(z) - z) - \mathrm{prox}_{th}(z)$

fixed points of F and minimizers of $g + h$

- if z is a fixed point, then $x = \mathrm{prox}_{th}(z)$ is a minimizer:

$$\begin{aligned} z = F(z), \quad x = prox_{th}(z) &\Rightarrow \mathrm{prox}_{tg}(2x - z) = x = \mathrm{prox}_{th}(z) \\ &\Rightarrow x - z \in t\partial g(x); z - x \in t\partial h(x) \\ &\Rightarrow 0 \in t\partial g(x) + t\partial h(x) \end{aligned}$$

- if x is a minimizer and $u \in t\partial g(x) \cap -t\partial h(x)$, then $x - u = F(x - u)$

Douglas-Rachford iteration with relaxation

fixed-point iteration with relaxation

$$z^+ = z + \rho(F(z) - z)$$

$1 < \rho < 2$ is overrelaxation, $0 < \rho < 1$ is underrelaxation

first version of DR method

$$\begin{aligned} x^+ &= \mathrm{prox}_{th}(z) \\ y^+ &= \mathrm{prox}_{tg}(2x^+ - z) \\ z^+ &= z + \rho(y^+ - x^+) \end{aligned}$$

alternate version

$$\begin{aligned} y^+ &= \mathrm{prox}_{tg}(x - w) \\ x^+ &= \mathrm{prox}_{th}((1-\rho)x + \rho y^+ + w) \\ w^+ &= w + \rho y^+ + (1-\rho)x - x^+ \end{aligned}$$

Dual application of Douglas-Rachford method

separable convex problem

$$\begin{array}{ll} \min & f_1(x_1) + f_2(x_2) \\ \text{s.t.} & A_1 x_1 + A_2 x_2 = b \end{array}$$

dual problem

$$\max \quad -b^T z - f_1^*(-A_1^T z) - f_2^*(-A_2^T z)$$

we apply the Douglas-Rachford method (page 3) to minimize

$$\underbrace{b^T z + f_1^*(-A_1^T z)}_{g(z)} + \underbrace{f_2^*(-A_2^T z)}_{h(z)}$$

Douglas Rachford on the dual

$$y^+ = \text{prox}_{tg}(z - w), \quad z^+ = \text{prox}_{th}(y^+ + w), \quad w^+ = w + y^+ - z^+$$

first line: use result in "lect-dualProxGrad.pdf" to compute $y^+ = \text{prox}_{tg}(z - w)$

$$\begin{aligned}
\hat{x}_1 &= \operatorname*{argmin}_{x_1}(f_1(x_1) + z^T(A_1x_1 - b) + \frac{t}{2}\|A_1x_1 - b - w/t\|_2^2) \\
y^+ &= z - w + t(A_1\hat{x}_1 - b)
\end{aligned}$$

second line: similarly, compute $z^+ = \text{prox}_{th}(z + t(A_1\hat{x}_1 - b))$

$$\begin{aligned}
\hat{x}_2 &= \operatorname*{argmin}_{x_1}(f_1(x_2) + z^TA_2x_2 + \frac{t}{2}\|A_1\hat{x}_1 + A_2x_2 - b\|_2^2) \\
z^+ &= z + t(A_1\hat{x}_1 + A_2\hat{x}_2 - b)
\end{aligned}$$

third line reduces to $w^+ = -tA_2\hat{x}_2$

Alternating direction method of multipliers

Define the augmented Lagrangian function:

$$L_t(x_1, x_2, z) = f_1(x_1) + f_2(x_2) + z^T(A_1x_1 + A_2x_2 - b) + \frac{t}{2}\|A_1x_1 + A_2x_2 - b\|_2^2$$

1. minimize augmented Lagrangian function over x_1

$$\begin{aligned} x_1^{(k)} &= \underset{x_1}{\operatorname{argmin}}\, L_t(x_1, x_2^{(k-1)}, z^{(k-1)}) \\ &= \underset{x_1}{\operatorname{argmin}} \left(f_1(x_1) + (z^{(k-1)})^T A_1x_1 + \frac{t}{2}\|A_1x_1 + A_2x_2^{(k-1)} - b\|_2^2\right) \end{aligned}$$

2. minimize augmented Lagrangian function over x_2

$$\begin{aligned} x_2^{(k)} &= \underset{x_2}{\operatorname{argmin}}\, L_t(x_1^{(k)}, x_2, z^{(k-1)}) \\ &= \underset{x_2}{\operatorname{argmin}} \left(f_2(x_2) + (z^{(k-1)})^T A_2x_2 + \frac{t}{2}\|A_1x_1^{(k)} + A_2x_2 - b\|_2^2\right) \end{aligned}$$

3. dual update $z^{(k)} = z^{(k-1)} + t(A_1x_1^{(k)} + A_2x_2^{(k)} - b)$

also known as split Bregman method

Outline

Nonexpansiveness

if $u = \mathrm{prox}_h(x), v = \mathrm{prox}_h(y)$, then

$$(u - v)^\top (x - y) \geq \|u - v\|2$$

prox_h is *firmly nonexpansive*, or *co-coercive* with constant 1

- follows from characterization of proximal mapping and monotonicity

$$x - u \in \partial h(u), y - v \in \partial h(v) \quad \Rightarrow \quad (x - u - y + v)^\top (u - v) \geq 0$$

- implies (from Cauchy-Schwarz inequality)

$$\|\mathrm{prox}_h(x) - prox_h(y)\|_2 \leq \|x - y\|_2$$

prox_h is *nonexpansive*, or *Lipschitz continuous* with constant 1

Douglas-Rachford iteration mappings

define iteration map F and negative step G

$$\begin{aligned} F(z) &= z + \mathrm{prox}_{tg}(2\mathrm{prox}_{th}(z) - z) - \mathrm{prox}_{th}(z) \\ G(z) &= z - F(z) \\ &= \mathrm{prox}_{th}(z) - \mathrm{prox}_{tg}(2\mathrm{prox}_{th}(z) - z) \end{aligned}$$

- F is firmly nonexpansive (co-coercive with parameter 1)

$$(F(z) - F(\hat{z}))^T(z - \hat{z}) \geq \|F(z) - F(\hat{z})\|_2^2 \quad \forall z, \hat{z}$$

- implies that G is firmly nonexpansive:

$$\begin{aligned} & (G(z) - G(\hat{z}))^T(z - \hat{z}) \\ = \;& \|G(z) - G(\hat{z})\|_2^2 + (F(z) - F(\hat{z}))^T(z - \hat{z}) - \|F(z) - F(\hat{z})\|_2^2 \\ \geq \;& \|G(z) - G(\hat{z})\|_2^2 \end{aligned}$$

Proof.

firm nonexpansiveness of F

- define $x = \mathrm{prox}_{th}(z), \hat{x} = \mathrm{prox}_{th}(\hat{z})$, and

$$y = \mathrm{prox}_{tg}(2x - z), \quad \hat{y} = \mathrm{prox}_{tg}(2\hat{x} - \hat{z})$$

- substitute expressions $F(z) = z + y - x$ and $F(\hat{z}) = \hat{z} + \hat{y} - \hat{x}$:

$$\begin{aligned}
&(F(z) - F(\hat{z}))^T(z - \hat{z}) \\
&\geq (z + y - x - \hat{z} - \hat{y} + \hat{x})^T(z - \hat{z}) - (x - \hat{z})^T(z - \hat{z}) + \|x - \hat{x}\|_2^2 \\
&= (y - \hat{y})^T(z - \hat{z}) + \|z - x - \hat{z} + \hat{x}\|_2^2 \\
&= (y - \hat{y})^T(2x - z - 2\hat{x} + \hat{z}) - \|y - \hat{y}\|_2^2 + \|F(z) - F(\hat{z})\|_2^2 \\
&\geq \|F(z) - F(\hat{z})\|_2^2
\end{aligned}$$

inequalities use firm nonexpansiveness of prox_{th} and prox_{tg}

$$(x - \hat{x})^T(z - \hat{z}) \geq \|x - \hat{x}\|_2^2, \quad (2x - z - 2\hat{x} + \hat{z})^T(y - \hat{y}) \geq \|y - \hat{y}\|_2^2$$

Convergence result

$$\begin{aligned} z^{(k)} &= (1-\rho_k)z^{(k-1)} + \rho_k F(z^{(k-1)}) \\ &= z^{(k-1)} - \rho_k G(z^{(k-1)}) \end{aligned}$$

assumptions

- optimal value $f^* = \inf_x (g(x) + h(x))$ is finite and attained
- $\rho_k \in [\rho_{\min}, \rho_{\max}]$ with $0 < \rho_{\min} < \rho_{\max} < 2$

result

- $z^{(k)}$ converges to a fixed point z^* of F
- $x^{(k)} = \mathrm{prox}_{th}(z^{(k-1)})$ converges to a minimizer $x^* = \mathrm{prox}_{th}(z^*)$ (follows from continuity of prox_{th})

Proof.

Let z^* be any fixed point of $F(z)$ (zero of $G(z)$). Consider iteration k (with $z = z^{(k-1)}, \rho = \rho_k, z^+ = z^{(k)}$):

$$\begin{aligned} \|z^+ - z^*\|_2^2 - \|z - z^*\|_2^2 &= 2(z^+ - z)^T(z - z^*) + \|z^+ - z\|_2^2 \\ &= -2\rho G(z)^T(z - z^*) + \rho^2\|G(z)\|_2^2 \\ &\leq -\rho(2-\rho)\|G(z)\|_2^2 \\ &\leq -M\|G(z)\|_2^2 \end{aligned} \tag{22}$$

where $M = \rho_{\min}(2 - \rho_{\max})$ (line 3 is firm nonexpansiveness of G)

- (22) implies that

$$M \sum_{k=0}^{\infty} \|G(z^{(k)})\|_2^2 \leq \|z^{(0)} - z^*\|_2^2, \quad \|G(z^{(k)})\|_2 \to 0$$

- (22) implies that $\|z^{(k)} - z^*\|_2$ is nonincreasing; $z^{(k)}$ bounded
- since $\|z^{(k)} - z^*\|_2$ is nonincreasing, the limit $\lim_{k\to\infty} \|z^{(k)} - z^*\|_2$ exists

continued.

- since the sequence $z^{(k)}$ is bounded, it has a convergent subsequence
- let $\bar{z_k}$ be a convergent subsequence with limit $\bar{z}$; by continuity of G,

$$0 = \lim_{k\to\infty} G(\bar{z_k}) = G(\bar{z})$$

hence, $\bar{z}$ is a zero of G and the limit $\lim_{k\to\infty} \|z^{(k)} - \bar{z}\|_2$ exists

- let $\bar{z_1}$ and $\bar{z_2}$ be two limit points; the limits

$$\lim_{k\to\infty} \|z^{(k_{j_1})} - \bar{z_1}\|_2, \quad \lim_{k\to\infty} \|z^{(k_{j_2})} - \bar{z_2}\|_2$$

exist, and subsequences of $z^{(k)}$ converge to $\bar{z_1}$, resp. $\bar{z_2}$; therefore

$$\|\bar{z_2} - \bar{z_1}\|_2 = \lim_{k\to\infty} \|z^{(k)} - \bar{z_1}\|_2 = \lim_{k\to\infty} \|z^{(k)} - \bar{z_2}\|_2 = 0$$

$\square$

- 考虑最优化问题

$$\begin{aligned} \min \quad & 0, \\ \text{s.t.} \quad & A_1x_1 + A_2x_2 + A_3x_3 = 0, \end{aligned} \tag{23}$$

其中$A_i \in \mathbb{R}^3$, $i = 1, 2, 3$为三维空间中的非零向量，$x_i \in \mathbb{R}$, $i = 1, 2, 3$是自变量．该问题实际上就是求解三维空间中的线性方程组，若A_1, A_2, A_3之间线性无关，则问题(23) 只有零解．此时容易计算出最优解对应的乘子为$y = (0, 0, 0)^{\mathrm{T}}$．

- 增广拉格朗日函数为

$$L_\rho(x, y) = 0 + y^{\mathrm{T}}(A_1x_1 + A_2x_2 + A_3x_3) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 + A_3x_3\|^2.$$

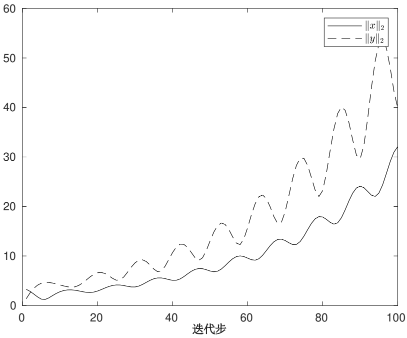
- 当固定x_2, x_3, y时，对x_1求最小可推出
$$A_1^{\mathrm{T}} y + \rho A_1^{\mathrm{T}}(A_1 x_1 + A_2 x_2 + A_3 x_3) = 0,$$
整理可得
$$x_1 = -\frac{1}{\|A_1\|^2}\left(A_1^{\mathrm{T}}\left(\frac{y}{\rho} + A_2 x_2 + A_3 x_3\right)\right).$$
可类似地计算x_2, x_3的表达式

- 因此多块交替方向乘子法的迭代格式可以写为
$$\begin{aligned}
x_1^{k+1} &= -\frac{1}{\|A_1\|^2} A_1^{\mathrm{T}}\left(\frac{y^k}{\rho} + A_2 x_2^k + A_3 x_3^k\right), \\
x_2^{k+1} &= -\frac{1}{\|A_2\|^2} A_2^{\mathrm{T}}\left(\frac{y^k}{\rho} + A_1 x_1^{k+1} + A_3 x_3^k\right), \\
x_3^{k+1} &= -\frac{1}{\|A_3\|^2} A_3^{\mathrm{T}}\left(\frac{y^k}{\rho} + A_1 x_1^{k+1} + A_2 x_2^{k+1}\right), \\
y^{k+1} &= y^k + \rho(A_1 x_1^{k+1} + A_2 x_2^{k+1} + A_3 x_3^{k+1}).
\end{aligned} \tag{24}$$

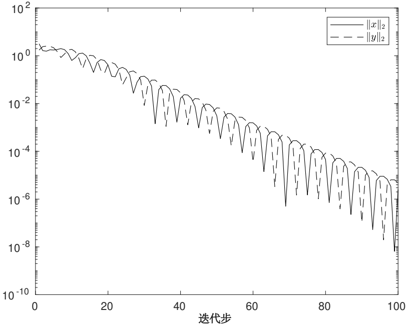
- 自变量初值初值选为$(1,1,1)$，乘子选为$(0,0,0)$. 选取A为

$$\widetilde{A}=\begin{bmatrix}1&1&2\\0&1&1\\0&0&1\end{bmatrix}\quad\text{或}\quad\widehat{A}=\begin{bmatrix}1&1&1\\1&1&2\\1&2&2\end{bmatrix}.$$

- 下图记录了在不同A下x和y的ℓ_2范数随迭代的变化过程.



(a) 系数矩阵为$\widetilde{A}$



(b) 系数矩阵为$\widehat{A}$

References

Douglas-Rachford method, ADMM, Spingarn's method

- J. E. Spingarn, *Applications of the method of partial inverses to convex programming: decomposition*, Mathematical Programming (1985)
- J. Eckstein and D. Bertsekas, *On the Douglas-Rachford splitting method and the proximal algorithm for maximal monotone operators*, Mathematical Programming (1992)
- P.L. Combettes and J.-C. Pesquet, *A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery*, IEEE Journal of Selected Topics in Signal Procesing (2007)
- S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers* (2010)
- N. Parikh, S. Boyd, *Block splitting for distributed optimization* (2013)

image deblurring: the example is taken from

D. O'Connor and L. Vandenberghe, *Primal-dual decomposition by operator splitting and applications to image deblurring* (2014)