凸优化和单调变分不等式的收缩算法

第六讲：为线性约束凸优化定制的 PPA 算法及其应用

Customized PPA for linearly constrained Optimization and its applications

南京大学数学系　　何炳生
hebma@nju.edu.cn

The context of this lecture is based on the publications [6, 7]

1 Introduction

从这一讲开始的四讲, 讨论的问题是

$$\min\{\theta(x) \mid Ax = b \,(or\ Ax \ge b)\ x \in \mathcal{X}\}$$

其中 $\theta(x)$ 是凸函数, $A \in \Re^{m\times n}$, $b \in \Re^m$, $\mathcal{X}$ 是 $\Re^n$ 中的闭凸集.

对任意给定的 $r > 0$ 和 $a \in \Re^n$, 通篇我们假设子问题

$$\min\left\{\theta(x) + \frac{r}{2}\|x - a\|^2 \mid x \in \mathcal{X}\right\}$$

的求解是简单的.

目的 说明将线性约束的凸优化问题转换成混合单调变分不等式, 选择适当的正定矩阵 H, 采用 H-模下的 PPA 方法. 在上述假设条件下, 线性约束的凸优化问题就变得非常容易求解.

For the analysis in this paper, we need only the basic property which is described in the following lemma.

Lemma 1.1 *Let* $\mathcal{X} \subset \Re^n$ *be a closed convex set,* $\theta(x)$ *and* $f(x)$ *be convex functions and* $f(x)$ *is differentiable. Assume that the solution set of the minimization problem* $\min\{\theta(x) + f(x) \,|\, x \in \mathcal{X}\}$ *is nonempty. Then,*

$$x^* \in \arg\min\{\theta(x) + f(x) \,|\, x \in \mathcal{X}\} \tag{1.1a}$$

if and only if

$$x^* \in \mathcal{X}, \quad \theta(x) - \theta(x^*) + (x - x^*)^T \nabla f(x^*) \geq 0, \quad \forall\, x \in \mathcal{X}. \tag{1.1b}$$

Proof : First, if (1.1a) is true, then for any $x \in \mathcal{X}$, we have

$$\frac{\theta(x_\alpha) - \theta(x^*)}{\alpha} + \frac{f(x_\alpha) - f(x^*)}{\alpha} \geq 0, \tag{1.2}$$

where

$$x_\alpha = (1 - \alpha)x^* + \alpha x, \quad \forall \alpha \in (0, 1].$$

Because $\theta(\cdot)$ is convex, it follows that

$$\theta(x_\alpha) \le (1-\alpha)\theta(x^*) + \alpha\theta(x),$$

and thus

$$\theta(x) - \theta(x^*) \ge \frac{\theta(x_\alpha) - \theta(x^*)}{\alpha}, \;\; \forall \alpha \in (0,1].$$

Substituting the last inequality in the left hand side of (1.2), we have

$$\theta(x) - \theta(x^*) + \frac{f(x_\alpha) - f(x^*)}{\alpha} \ge 0, \quad \forall \alpha \in (0,1].$$

Using $f(x_\alpha) = f(x^* + \alpha(x - x^*))$ and letting $\alpha \to 0_+$, from the above inequality we get

$$\theta(x) - \theta(x^*) + \nabla f(x^*)^T(x - x^*) \ge 0, \;\; \forall\, x \in \mathcal{X}.$$

Thus (1.1b) follows from (1.1a). Conversely, since f is convex, it follow that

$$f(x_\alpha) \le (1-\alpha) f(x^*) + \alpha f(x)$$

and it can be rewritten as

$$f(x_\alpha) - f(x^*) \le \alpha(f(x) - f(x^*)).$$

Thus, we have

$$f(x) - f(x^*) \ge \frac{f(x_\alpha) - f(x^*)}{\alpha} = \frac{f(x^* + \alpha(x - x^*)) - f(x^*)}{\alpha},$$

for all $\alpha \in (0, 1]$. Letting $\alpha \to 0_+$, we get

$$f(x) - f(x^*) \ge \nabla f(x^*)^T (x - x^*).$$

Substituting it in the left hand side of (1.1b), we get

$$x^* \in \mathcal{X}, \quad \theta(x) - \theta(x^*) + f(x) - f(x^*) \ge 0, \quad \forall\, x \in \mathcal{X},$$

and (1.1a) is true. The proof is complete. $\square$

1.1 Linear constrained convex optimization and VI

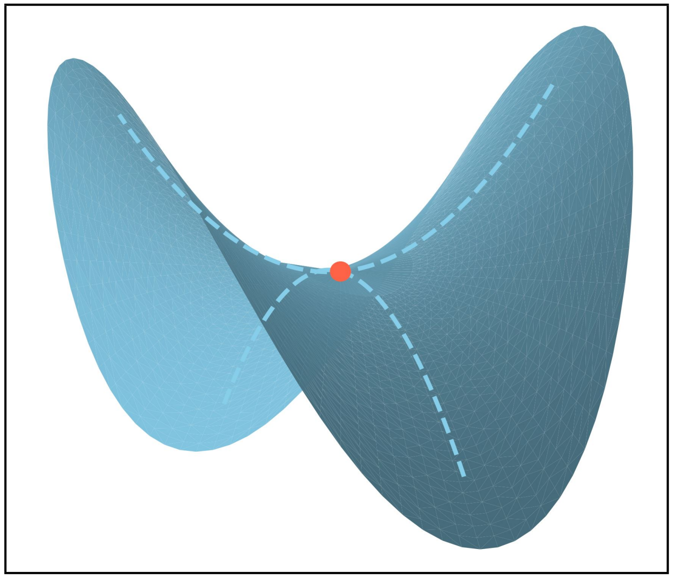
We consider the linearly constrained convex optimization problem

$$\min\{\theta(x) | Ax = (\text{or } \geq)\, b, x \in \mathcal{X}\}. \quad (1.3)$$

The Lagrangian function of the problem (1.3) is

$$L(x, \lambda) = \theta(x) - \lambda^T(Ax - b),$$

which is defined on $\mathcal{X} \times \Lambda$ ($\Re^m$ or $\Re^m_+$).



A pair of (x^*, λ^*) is called a saddle point of the Lagrange function, if

$$L_{\lambda \in \Lambda}(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L_{x \in \mathcal{X}}(x, \lambda^*).$$

An equivalent expression of the saddle point is the following variational inequality:

$$\begin{cases} x^* \in \mathcal{X}, & \theta(x) - \theta(x^*) + (x - x^*)^T(-A^T\lambda^*) \geq 0, \quad \forall\, x \in \mathcal{X}, \\ \lambda^* \in \Lambda, & (\lambda - \lambda^*)^T(Ax^* - b) \geq 0, \quad \forall\, \lambda \in \Lambda. \end{cases}$$

The optimal condition can be characterized as a monotone variational inequality:

$$w^* \in \Omega, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega. \qquad \text{(1.4a)}$$

where

$$w = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A^T\lambda \\ Ax - b \end{pmatrix} \quad \text{and} \quad \Omega = \mathcal{X} \times \Lambda. \qquad \text{(1.4b)}$$

Note that the operator F is monotone, because

$$(w - \tilde{w})^T (F(w) - F(\tilde{w})) \geq 0, \ \text{Here } (w - \tilde{w})^T (F(w) - F(\tilde{w})) = 0. \qquad \text{(1.5)}$$

Example 1 of the problem (1.3): Finding the nearest correlation matrix

A positive semi-definite matrix, whose each diagonal element is equal 1, is called the correlation matrix. For given symmetric $n \times n$ matrix C, the mathematical form of finding the nearest correlation matrix X is

$$\min\{\tfrac{1}{2}\|X - C\|_F^2 \,|\, \mathsf{diag}(X) = e, \ X \in S_+^n\}, \qquad \text{(1.6)}$$

where S_+^n is the positive semi-definite cone and e is a n-vector whose each element is equal 1. The problem (1.6) is a concrete problem of type (1.3).

Example 2 of the problem (1.3): The matrix completion problem

Let M be a given $m \times n$ matrix, Π is the elements indices set of M,

$$\Pi \subset \{(ij) | i \in \{1, \dots, m\},\ j \in \{1, \dots, n\}\}.$$

The mathematical form of the matrix completion problem is relaxed to

$$\min\{\|X\|_* \mid X_{ij} = M_{ij},\ (ij) \in \Pi\}, \qquad (1.7)$$

where $\|\cdot\|_*$ is the nuclear norm–the sum of the singular values of a given matrix. The problem (3.5) is a convex optimization of form (1.3). The matrix A in (1.3) for the linear constraints

$$X_{ij} = M_{ij},\ (ij) \in \Pi,$$

is a projection matrix, and thus $\|A^T A\| = 1$.

M is low Rank, only some elements of M are known.

*			*		*			*	
		*		*			*		
*			*		*		*		*
		*			*			*	
	*			*		*			*
*			*		*			*	
	*			*		*			*
*		*		*	*		*	*	
					*				*
*			*			*		*	
	*		*		*		*		
			*		*			*	
		*		*			*		*
	*		*			*		*	

1.2 Proximal point algorithms for convex optimization

Lemma 1.2 *Let the vectors* $a, b \in \Re^n$, $H \in \Re^{n\times n}$ *be a positive definite matrix. If* $b^T H(a-b) \geq 0$, *then we have*

$$\|b\|_H^2 \leq \|a\|_H^2 - \|a-b\|_H^2. \tag{1.8}$$

The assertion follows from $\|a\|^2 = \|b+(a-b)\|^2 \;\geq\; \|b\|^2 + \|a-b\|^2$.

Convex Optimization

Now, let us consider the *simple* convex optimization

$$\min\{\theta(x) + f(x) \mid x \in \mathcal{X}\}, \tag{1.9}$$

where $\theta(x)$ and $f(x)$ are convex but $\theta(x)$ is not necessary smooth, $\mathcal{X}$ is a closed convex set.

For solving (1.9), the k-th iteration of the proximal point algorithm (abbreviated to PPA) [9, 11] begins with a given x^k, offers the new iterate x^{k+1} via the recursion

$$x^{k+1} = \text{Argmin}\{\theta(x) + f(x) + \frac{r}{2}\|x - x^k\|^2 \mid x \in \mathcal{X}\}. \tag{1.10}$$

Since x^{k+1} is the optimal solution of (1.10), it follows from Lemma 1.1 that

$$\theta(x)-\theta(x^{k+1})+(x-x^{k+1})^T\{\nabla f(x^{k+1})+r(x^{k+1}-x^k)\}\geq 0,\ \forall\, x\in\mathcal{X}. \tag{1.11}$$

Setting $x=x^*$ in the above inequality, it follows that

$$(x^{k+1}-x^*)^T r(x^k-x^{k+1})\geq\theta(x^{k+1})-\theta(x^*)+(x^{k+1}-x^*)^T\nabla f(x^{k+1}). \tag{1.12}$$

Since

$$(x^{k+1}-x^*)^T\nabla f(x^{k+1})\geq(x^{k+1}-x^*)^T\nabla f(x^*),$$

it follows that the right hand side of (1.12) is nonnegative. And consequently,

$$(x^{k+1}-x^*)^T(x^k-x^{k+1})\geq 0. \tag{1.13}$$

Let $a=x^k-x^*$ and $b=x^{k+1}-x^*$ and using Lemma 1.2, we obtain

$$\|\boldsymbol{x}^{\boldsymbol{k+1}}-\boldsymbol{x}^*\|^{\mathbf{2}}\leq\|\boldsymbol{x}^{\boldsymbol{k}}-\boldsymbol{x}^*\|^{\mathbf{2}}-\|\boldsymbol{x}^{\boldsymbol{k}}-\boldsymbol{x}^{\boldsymbol{k+1}}\|^{\mathbf{2}}, \tag{1.14}$$

which is the nice convergence property of Proximal Point Algorithm.

1.3 Preliminaries of PPA for Variational Inequalities

The optimal condition of the problem (1.3) is characterized as a mixed monotone variational inequality:

$$w^* \in \Omega, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall\, w \in \Omega. \tag{1.15}$$

PPA for VI in Euclidean-norm

For given w^k and $H \succ 0$, find w^{k+1}

$$\begin{aligned} \tilde{w}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) &+ (w - \tilde{w}^k)^T F(\tilde{w}^k) \\ &\geq (w - \tilde{w}^k)^T H(w^k - \tilde{w}^k), \ \forall\, w \in \Omega. \end{aligned} \tag{1.16}$$

$\tilde{w}^k$ is called the proximal point of the k-th iteration for the problem (1.15).

✠ **w^k is the solution of (1.15) if and only if $w^k = \tilde{w}^k$** ✠

Setting $w = w^*$ in (1.16), we obtain

$$(\tilde{w}^k - w^*)^T H(w^k - \tilde{w}^k) \geq \theta(\tilde{x}^k) - \theta(x^*) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k)$$

Note that (see the structure of $F(w)$ in (1.4b))

$$(\tilde{w}^k - w^*)^T F(\tilde{w}^k) = (\tilde{w}^k - w^*)^T F(w^*),$$

and consequently (by using (1.15)) we obtain

$$(\tilde{w}^k - w^*)^T r(w^k - \tilde{w}^k) \geq \theta(\tilde{x}^k) - \theta(x^*) + (\tilde{w}^k - w^*)^T F(w^*) \geq 0,$$

and thus

$$(\tilde{w}^k - w^*)^T H(w^k - \tilde{w}^k) \geq 0. \tag{1.17}$$

If we take $w^{k+1} = \tilde{w}^k$, and set $a = w^k - w^*$ and $b = w^{k+1} - w^*$ in the inequality (1.17), it yields that $b^T H(a - b) \geq 0$. Using Lemma1.2, we get the nice convergence property of the PPA in H-norm:

$$\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - \|w^k - w^{k+1}\|_H^2. \tag{1.18}$$

Extended PPA From the inequality (1.17) consequently

$$(w^k - w^*)^T H(w^k - \tilde{w}^k) \geq \|w^k - \tilde{w}^k\|_H^2. \tag{1.19}$$

Usually, we take

$$w^{k+1} = w^k - \gamma(w^k - \tilde{w}^k), \quad \gamma = 1.5 \in (0, 2) \tag{1.20}$$

as the new iterate. Then

$$\|w^{k+1} - w^*\|_H^2 \le \|w^k - w^*\|_H^2 - \gamma(2-\gamma)\|w^k - \tilde{w}^k\|_H^2. \qquad (1.21)$$

Theorem 1.1 *For given w^k, setting the solution of* (1.16) *by $\tilde{w}^k$ and let the new iterate w^{k+1} be updated by* (1.20)*. Then, the generated sequence $\{w^k\}$ converges to a solution point of* (1.15)*.*

Sketch of the proof .

- Since (1.21) is true for every solution point, the sequence $\{w^k\}$ is bounded and $\lim_{k\to\infty} \|w^k - \tilde{w}^k\|_H = 0$.
- Because $\lim_{k\to\infty} \|w^k - \tilde{w}^k\|_H = 0$ and $\{w^k\}$ is bounded, $\{\tilde{w}^k\}$ is also bounded. Moreover, $\{w^k\}$ and $\{\tilde{w}^k\}$ have the same cluster points.
- According to (1.16), every cluster point of $\{\tilde{w}^k\}$ which is also the cluster point of $\{w^k\}$, is a solution point of the VI (1.15).
- Finally, according to the contraction inequality (1.21), the generated sequence $\{w^k\}$ can only have a cluster point and the method is convergent.

2 变分不等式框架下凸优化的 PPA 算法

对等式约束 (或不等式约束) 的凸优化问题 (1.3),

$$\min\{\theta(x) \mid Ax = b\,(or\ Ax \ge b),\ x \in \mathcal{X}\}$$

我们用 Customized PPA 求解与它们等价的变分不等式 (1.4). 说 Customized PPA, 是因为后面的 H-模下的 PPA 方法可以看作按需定制的.

采用 Customized PPA 方法, 要求 A^TA 的模是容易估计的. 这一讲的最后一节会说明一些有应用背景的问题, 恰能满足这些要求.

2.1 Primal-dual hybrid gradient algorithm

Since the objective is to find a saddle point of the Lagrange function, a natural idea is to use the primal-dual hybrid gradient algorithm [12]. For given (x^k, λ^k), by using the primal-dual hybrid gradient algorithm,

$$x^{k+1} = \text{Argmin}\{L(x, \lambda^k) + \frac{r}{2}\|x - x^k\|^2 \mid x \in \mathcal{X}\}, \tag{2.1}$$

after getting x^{k+1}, we obtain

$$\lambda^{k+1} = \text{Argmax}\{L(x^{k+1}, \lambda) - \frac{s}{2}\|\lambda - \lambda^k\|^2 \mid \lambda \in \Re^m\}. \tag{2.2}$$

Combining (2.1) and (2.2), we get

$$\theta(x) - \theta(x^{k+1}) + \begin{pmatrix} x - x^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\lambda^{k+1} \\ Ax^{k+1} - b \end{pmatrix} \right.$$

$$\left. + \begin{pmatrix} r(x^{k+1} - x^k) + A^T(\lambda^{k+1} - \lambda^k) \\ s(\lambda^{k+1} - \lambda^k) \end{pmatrix} \right\} \geq 0, \quad \forall (x, \lambda) \in \Omega.$$

The compact form is

$$\theta(x) - \theta(x^{k+1}) + (w - w^{k+1})^T\{F(w^{k+1}) + Q(w^{k+1} - w^k)\} \geq 0, \ \forall w \in \Omega, \tag{2.3}$$

where

$$Q = \begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix}$$ is not symmetric.

Remark For general, min-max problem, the primal-dual hybrid gradient algorithm is not convergent. For example, we consider a pair of the primal-dual linear programming:

$$\text{(Primal)} \quad \begin{array}{ll} \min & c^T x \\ \text{s. t.} & Ax = b \\ & x \geq 0. \end{array} \qquad \text{(Dual)} \quad \begin{array}{ll} \max & b^T y \\ \text{s. t.} & A^T y \leq c. \end{array}$$

We take the following example

$$\text{(P)} \quad \begin{array}{ll} \min & x_1 + 2x_2 \\ \text{s. t.} & x_1 + x_2 = 1 \\ & x_1, \; x_2 \geq 0. \end{array} \qquad \text{(D)} \quad \begin{array}{ll} \max & y \\ \text{s. t.} & \begin{bmatrix} 1 \\ 1 \end{bmatrix} y \leq \begin{bmatrix} 1 \\ 2 \end{bmatrix} \end{array}$$

where $A = [1, 1], \; b = 1, c = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and the vector $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

The optimal solutions of this pair of linear programming are $x^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $y^* = 1$.

Note that its Lagrange function is

$$L(x,y) = c^T x - y^T (Ax - b) \tag{2.4}$$

which defined on $R^2_+ \times R$. (x^*, y^*) is the unique saddle point of the Lagrange function.

Using the Primal-dual hybrid gradient method with $r = s = 1$. For given (x^k, y^k), Zhu and Chan's procedure is

$$x^{k+1} = \text{Argmin}\{L(x, y^k) + \frac{1}{2}\|x - x^k\|^2 \,|\, x \geq 0\} = \max\{(x^k + A^T y^k - c), 0\},$$

and then

$$y^{k+1} = \text{Argmax}\{L(x^{k+1}, y) - \frac{1}{2}\|y - y^k\|^2\} = y^k - (Ax^{k+1} - 1).$$

In other words, the iteration formula is

$$\begin{cases} x^{k+1} = \max\{(x^k + \frac{1}{r}(A^T y^k - c)), 0\}, \\ y^{k+1} = y^k - \frac{1}{s}(Ax^{k+1} - b). \end{cases}$$

We use $(x_1^0, x_2^0; y^0) = (0, 0; 0)$ as the start point. For this example, the method is not convergent.

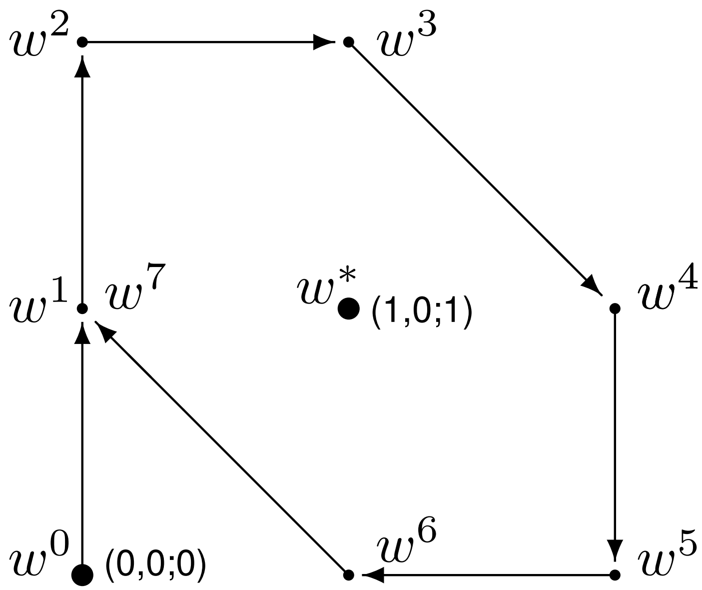


Fig. 2.1 The sequence generated by PDHG Method with $r = s = 1$

$$
\begin{aligned}
w^0 &= (0, 0; 0) \\
w^1 &= (0, 0; 1) \\
w^2 &= (0, 0; 2) \\
w^3 &= (1, 0; 2) \\
w^4 &= (2, 0; 1) \\
w^5 &= (2, 0; 0) \\
w^6 &= (1, 0; 0) \\
w^7 &= (0, 0; 1) \\
\boldsymbol{w^{k+6}} &= \boldsymbol{w^k}
\end{aligned}
$$

2.2 Customized PPA

If we can change Q to a symmetric matrix H such that

$$
Q = \begin{pmatrix} rI_n & A^T \\ 0 & sI_m \end{pmatrix} \quad \Rightarrow \quad H = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix},
$$

then $\tilde{w}^k$ is the solution of the following variational inequality:

$$\tilde{w}^k \in \Omega, \;\; \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T \{F(\tilde{w}^k) + H(\tilde{w}^k - w^k)\} \geq 0, \;\; \forall w \in \Omega. \tag{2.5}$$

For this purpose, we need only change $\quad (A\tilde{x}^k - b) + s(\tilde{\lambda}^k - \lambda^k) = 0 \quad$ to

$$(A\tilde{x}^k - b) + A(\tilde{x}^k - x^k) + s(\tilde{\lambda}^k - \lambda^k) = 0. \tag{2.6}$$

Because $\tilde{x}^k$ is known, with the given x^k and λ^k, $\tilde{\lambda}^k$ is given by

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A(2\tilde{x}^k - x^k) - b].$$

Thus, for given (x^k, λ^k), produce a proximal point $(\tilde{x}^k, \tilde{\lambda}^k)$ via (2.1) and (2.6) can be summarized as:

$$\tilde{x}^k = \text{argmin}\Big\{ L(x, \lambda^k) + \frac{r}{2}\big\|x - x^k\big\|^2 \,\big|\, x \in \mathcal{X} \Big\}. \tag{2.7a}$$

$$\tilde{\lambda}^k = \text{argmax}\Big\{ L\big([2\tilde{x}^k - x^k], \lambda\big) - \frac{s}{2}\big\|\lambda - \lambda^k\big\|^2 \Big\} \tag{2.7b}$$

We call the point $(\tilde{x}^k, \tilde{\lambda}^k)$ generated by (2.7) as the predictor. The subproblem (2.7a) is equivalent to

$$\tilde{x}^k = \text{argmin}\Big\{\theta(x) + \frac{r}{2}\big\|x - \big[x^k + \frac{1}{r}A^T\lambda^k\big]\big\|^2 \,\Big|\, x \in \mathcal{X}\Big\}.$$

The solution of (2.7b) is given by

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}[A(2\tilde{x}^k - x^k) - b].$$

Indeed, under the assumption, the sub-problem (2.7a) is simple.

In the case that $rs > \|A^T A\|$, the matrix

$$H = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix} \quad \text{is positive definite}$$

and the $\tilde{w}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ generated by (2.7) is a proximal point. Based on the predictor, the new iterate is given by

$$w^{k+1} := w^k - \gamma(w^k - w^{k+1}), \quad \gamma \in (0, 2).$$

Theorem 2.1 *The sequence* $\{w^k = (x^k, \lambda^k)\}$ *generated by the customized PPA satisfies*

$$\|w^{k+1} - w^*\|_H^2 \le \|w^k - w^*\|_H^2 - \gamma(2-\gamma)\|w^k - \tilde{w}^k\|_H^2. \tag{2.8}$$

定理 2.1 中的 (2.8) 是方法总体收敛的关键不等式. 如同 §1.3，可以由此证明算法的总体收敛性. 计算中实际困难往往在于如何选择参数 r 和 s.

我们也可以用另外的顺序产生预测点 $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$. 对给定的 (x^k, λ^k), 求 $(\tilde{x}^k, \tilde{\lambda}^k) \in \Omega$, 使得

$$\theta(x) - \theta(\tilde{x}^k) + \begin{pmatrix} x - \tilde{x}^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{\lambda}^k \\ A\tilde{x}^k - b \end{pmatrix} \right.$$

$$\left. + \begin{pmatrix} r(\tilde{x}^k - x^k) - A^T(\tilde{\lambda}^k - \lambda^k) \\ -A(\tilde{x}^k - x^k) + s(\tilde{\lambda}^k - \lambda^k) \end{pmatrix} \right\} \ge 0, \quad \forall (x, \lambda) \in \Omega. \tag{2.9}$$

这相当于求得变分不等式 (1.4) 的 PPA 子问题 (1.16) 的解, 其中对称矩阵

$$H = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix}. \tag{2.10}$$

同样, 当 $rs > \|A^TA\|$ 时 G 是正定的. 注意到(2.9) 的下半部分相当于

$$\tilde{\lambda}^k \in \Lambda, \quad (\lambda - \tilde{\lambda}^k)^T \{(Ax^k - b) + s(\tilde{\lambda}^k - \lambda^k)\} \geq 0, \quad \forall\, \lambda \in \Lambda,$$

它可以由

$$\tilde{\lambda}^k = P_\Lambda[\lambda^k - \frac{1}{s}(Ax^k - b)] \tag{2.11}$$

给出. 因为有了 $\tilde{\lambda}^k$, 变分不等式 (2.9) 的上半部分中未知的只有 $\tilde{x}^k$, 它可以通过求解极小化问题

$$\min\left\{\theta(x) + \frac{r}{2}\left\|x - \left[x^k + \frac{1}{r}\, A^T(2\tilde{\lambda}^k - \lambda^k)\right]\right\|^2 \,\Big|\, x \in \mathcal{X}\right\} \tag{2.12}$$

得到. 这是一个容易求解的问题. 生成预测点以后, 给出新迭代点 w^{k+1} 的公式是

$$w^{k+1} = w^k - \gamma(w^k - \tilde{w}^k), \quad \gamma \in (0, 2).$$

3 Applications in optimization problems

3.1 Correcting the correlation matrices

在统计学中, 一个对角元均为 1 的对称半正定矩阵称为 (Correlation Matrix) 相关性矩阵. 对给定的对称矩阵 C, 求 F-模下与 C 距离最近的相关性矩阵, 其数学表达式是

$$\min\{\frac{1}{2}\|X-C\|_F^2 \mid \text{diag}(X)=e,\ X\in S_+^n\}, \tag{3.1}$$

其中 e 表示每个分量都为 1 的 n-维向量, S_+^n 表示 $n\times n$ 正半定锥的集合. 问题 (3.1) 是形如 (1.3) 的等式约束凸优化问题, 其中 $\|A^TA\|=1$.

我们用 $z\in\Re^n$ 作为等式约束 $\text{diag}(X)=e$ 的 Lagrange 乘子.

Dual-Primal Customized PPA 求解问题(3.1)

对给定的 (X^k,z^k), 用 (2.11)–(2.12) 产生$(\tilde{X}^k,\tilde{z}^k)$:

1. Producing $\tilde{z}^k$ by

$$\tilde{z}^k = z^k - \frac{1}{s}(\text{diag}(X^k)-e).$$

2. Finding $\tilde{X}^k$ which is the solution of the following minimization problem

$$\min\{\frac{1}{2}\|X - C\|_F^2 + \frac{r}{2}\|X - [X^k + \frac{1}{r}\operatorname{diag}(2\tilde{z}^k - z^k)]\|_F^2 | X \in S_+^n\}. \quad (3.2)$$

子问题 (3.2) 求解的具体做法: 化为等价问题

$$\min\{\frac{1}{2}\|X - \frac{1}{1+r}[rX^k + \operatorname{diag}(2\tilde{z}^k - z^k) + C]\|_F^2 | X \in S_+^n\}.$$

记 $A = \frac{1}{1+r}[rX^k + \operatorname{diag}(2\tilde{z}^k - z^k) + C]$, 我们只要考虑如何求解

$$\tilde{X}^k = \text{Argmin}\{\frac{1}{2}\|X - A\|_F^2 \mid X \in S_+^n\}. \quad (3.3)$$

实际上, 将对称矩阵 A 做标准特征值– 特征向量分解

$$A = V\Lambda V^T, \qquad \Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n). \quad (3.4)$$

其中 V 是正交矩阵. 注意到正交变换下矩阵的 Frobenius-模是不变的, 我们有

$$\|X - A\|_F = \|X - V\Lambda V^T\|_F = \|V^T XV - \Lambda\|_F.$$

要使 $\tilde{X} \succeq 0$ 并且上式右端最小, 应该有

$$V^T \tilde{X} V = \tilde{\Lambda},$$

其中

$$\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_n), \qquad \tilde{\lambda}_j = \max\{0, \lambda_j\}.$$

最后通过

$$\tilde{X} = V \tilde{\Lambda} V^T$$

得到 $\tilde{X}$. 因此, 每次迭代的主要工作是做 (3.4) 中的特征值 (特征向量) 分解.

数值试验 为生成试验例子, 只要给定对称矩阵 C.

C=rand(n,n); C=(C'+C)-ones(n,n) + eye(n)

这样的矩阵 C 的对角元在 $(0, 2)$ 之间, 非对角元在 $(-1, 1)$ 之间。

Code 4.a. Matlab code for Creating the test examples

```
clear;  close all;
n = 1000;     tol=1e-5;     r=2.0;     s=1.01/r;     gamma=1.5;
rand('state',0);   C=rand(n,n);    C=(C'+C)-ones(n,n) + eye(n);
```

Code 4.1. Matlab code of the classical PPA

```
%%%     Classical PPA for calibrating correlation matrix        %(1)
function PPAC(n,C,r,s,tol)                                      %(2)
X=eye(n);      z=zeros(n,1);    tic;      %%  The initial iterate %(3)
stopc=1;                         k=0;                           %(4)
while (stopc>tol && k<=100)        %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(' k=%4d   epsm=%9.3e  \n',k,stopc); end; %(6)
    X0=X;    z0=z;    k=k+1;                                    %(7)
    zt=z0 - (diag(X0)-ones(n,1))/s;             EZ=z0-zt;       %(8)
    A=(X0*r + C + diag(zt*2-z0))/(1+r);                         %(9)
    [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V';     EX=X0-XT;     %(10)
    ex=max(max(abs(EX)));  ez=max(abs(EZ));   stopc=max(ex,ez);  %(11)
    X=XT;                  z=zt;                                %(12)
end;                                      % End of an Iteration %(13)
toc;                       TB = max(abs(diag(X-eye(n))));       %(14)
fprintf(' k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n',k,stopc,TB); %%
```

做 (3.4) 中的特征值 (特征向量) 分解, 在上述程序中的第 (10) 行用 Matlab 中的语句 [V,D]=eig(A) 实现的, 这是一个计算量大概 $9n^3$ 的运算.

将 Classical PPA 改成 Extended PPA, 只要将第 (12) 行改一下。

Code 4.2 Matlab Code of the Extended PPA

```
%%%     Extended PPA for calibrating correlation matrix          %(1)
function PPAE(n,C,r,s,tol,gamma)                                 %(2)
X=eye(n);      z=zeros(n,1);    tic;     %%  The initial iterate  %(3)
stopc=1;                       k=0;                               %(4)
while (stopc>tol && k<=100)        %% Beginning of an Iteration  %(5)
 if mod(k,20)==0 fprintf(’ k=%4d   epsm=%9.3e  \n’,k,stopc); end; %(6)
    X0=X;    z0=z;    k=k+1;                                     %(7)
    zt=z0 - (diag(X0)-ones(n,1))/s;               EZ=z0-zt;      %(8)
    A=(X0*r + C + diag(zt*2-z0))/(1+r);                          %(9)
    [V,D]=eig(A);   D=max(0,D);  XT=(V*D)*V’;     EX=X0-XT;      %(10)
    ex=max(max(abs(EX)));  ez=max(abs(EZ));   stopc=max(ex,ez);  %(11)
    X=X0-EX*gamma;         z=z0-EZ*gamma;                        %(12)
end;                                          % End of an Iteration %(13)
toc;                       TB = max(abs(diag(X-eye(n))));        %(14)
fprintf(’ k=%4d  epsm=%9.3e   max|X_jj - 1|=%8.5f \n’,k,stopc,TB); %%
```

两个不同的方法, 程序都很简单, 用不了几行. 两个程序不同的地方仅仅是第 12 行有些差别, 取 $\gamma = 1.5$ 的方法效果却有明显的提高。

由于在这个问题中, A 是投影矩阵, $\|A^T A\| = 1$, 我们只需要选 $rs > 1$ 就能确保 $G \succ 0$. 注意到, 为了平衡 primal 和 dual 残量, 我们取 $r = 2$, $s = 1.01/r$.

特征值分解用 Matlab 中的 eig 的迭代次数和计算时间

$n \times n$ Matrix	Classical PPA		Extended PPA	
$n =$	No. It	CPU Sec.	No. It	CPU Sec.
100	31	0.35	23	0.26
200	34	2.00	25	1.48
500	39	17.28	27	13.14
800	41	72.82	29	51.97
1000	47	153.14	31	102.12
2000	62	1344.50	40	881.38

The extended PPA converges faster than the classical PPA.

$$\frac{\text{It. No. of Extended PPA}}{\text{It. No. of Classical PPA}} \approx 65\%.$$

♣ 关于相关系数矩阵校正的程序在附件的 Codes-04 的文件夹"矩阵校正"中. 只要运行 demo.m, 输入 n 就可以了. 其中的 PPAC.m 和 PPAE.m 分别是 Classical PPA 和 Extended PPA 的子程序.

确实, 用这一讲介绍的 PPA 方法求解相关矩阵矩阵校正问题, 每步迭代的主要计算工作量是对一个对称矩阵用 Matlab 中的标准子程序做 [V,D]=eig(A). 如果改用 Kim TOH 写的 mexeig 做 [V,D]=mexeig(A), 计算时间大为节省.

特征值分解使用 mexeig 的迭代次数和计算时间

$n \times n$ Matrix	Classical PPA		Extended PPA	
$n =$	No. It	CPU Sec.	No. It	CPU Sec.
100	30	0.12	23	0.10
200	33	0.54	25	0.40
500	38	7.99	26	6.25
800	38	37.44	28	27.04
1000	45	94.32	30	55.32
2000	62	723.40	38	482.18

3.2 矩阵完整化方面的应用

设 M 是一个 $m \times n$ 矩阵, Π 是矩阵元素的指标集.

$$\Pi = \{(ij) \,|\, i \in \{1, 2, \ldots, m\},\ j \in \{1, 2, \ldots, n\}\}.$$

矩阵完整化问题是 **由部分信息获取全部信息**

- 显然, 没有其他信息, 恢复一个一般的矩阵是不可能的.
- 幸运的是, 在许多情况下, 我们要恢复的矩阵是一个低秩矩阵.

恢复低秩矩阵的一般数学模型是:

$$\min\{\text{rank}(X) \ :\ X_{ij} = M_{ij},\ (ij) \in \Pi\}.$$

然而, 这个问题是NP-hard 的. 根据 **Candés, Recht, Tao** 最近的工作

- E. J. Candés and B. Recht, Exact Matrix Completion via Convex Optimization, 2008.
- E. J. Candés and T. Tao, The Power of Convex Relaxation: Near-Optimial Matrix Completion, 2009,

在适当(实际问题具备的)条件下, 大多数不完整信息的低秩矩阵可以通过求解松弛问题

$$\min\{\|X\|_* \mid X_{ij} = M_{ij},\ (ij) \in \Pi\} \tag{3.5}$$

得到精确恢复. 其中 $\|X\|_*$ 表示矩阵 X 的奇异值的和. 通常称为矩阵 X 的核模— Nuclear Norm. 这类问题的商业应用, 可见 [10].

问题 (3.5) 是形如 (1.3) 的等式约束凸优化问题, 其中 $\|A^T A\| = 1$. 我们将 (3.5) 的等式约束记为 $X_\Pi = M_\Pi$，并用 $Z \in \Re^{m\times n}$ 作为相应的 Lagrange 乘子.

Dual-Primal PPA 求解问题(3.5)

对给定的 (X^k, Z^k), 用 (2.11)–(2.12) 产生$(\tilde{X}^k, \tilde{Z}^k)$:

1. Producing $\tilde{Z}^k$ by
$$\tilde{Z}^k_\Pi = Z^k_\Pi - \frac{1}{s}(X^k_\Pi - M_\Pi).$$

2. Finding $\tilde{X}^k$ which is the solution of the following linear variational inequality
$$\min\ \Big\{\|X\|_* + \frac{r}{2}\Big\|X - \big[X^k + \frac{1}{r}(2\tilde{Z}^k_\Pi - Z^k_\Pi)\big]\Big\|_F^2\Big\} \tag{3.6}$$

子问题 (3.6) 求解的具体做法: 只需考虑如何求解

$$\tilde{X}^k = \text{Argmin}\Big\{\frac{1}{r}\|X\|_* + \frac{1}{2}\|X - A\|_F^2\Big\}. \tag{3.7}$$

我们将 A 做 SVD 分解

$$A = U\Lambda V^T,$$

并代入 (3.7), 得到

$$\frac{1}{r}\|\tilde{X}\|_* + \frac{1}{2}\|\tilde{X} - U\Lambda V^T\|_F^2 = \frac{1}{r}\|U^T\tilde{X}V\|_* + \frac{1}{2}\|U^T\tilde{X}V - \Lambda\|_F^2.$$

上面的等式是由于矩阵的奇异值及F-范数在正交变换下不变的原因. 因此, $U^T\tilde{X}V$ 应该是一个非负的对角矩阵. 设 $U^T\tilde{X}V = \tilde{\Lambda}$, 换句话说,

$$\tilde{X} = U\tilde{\Lambda}V^T. \tag{3.8}$$

对给定的非负对角矩阵 Λ,

$$\min\Big\{\frac{1}{r}\|\tilde{\Lambda}\|_* + \frac{1}{2}\|\tilde{\Lambda} - \Lambda\|_F^2\Big\}$$

的解对角矩阵 $\tilde{\Lambda}$, 其对角元通过

$$\tilde{\lambda}_j = \lambda_j - \min(\lambda_j, \frac{1}{r}), \tag{3.9}$$

就能得到. 代入 (3.8) 就得到 (3.7) 的解 $\tilde{X}^k$. 因此, 每次迭代的主要工作量是做一个矩阵的 SVD 分解.

如果以 λ 和 $\tilde{\lambda}$ 分别表示对角矩阵 Λ 和 $\tilde{\Lambda}$ 的对角元生成的向量, 由于 λ 是非负向量, 关系式 (3.9) 也可以写成

$$\tilde{\lambda} = \lambda - P_{B_{\infty}^{1/r}}[\lambda],$$

其中 $B_{\infty}^{1/r}$ 是无穷模下半径为 $1/r$ 的“圆”（一个立方体）. 上述运算在文献中常常称作 **Shrinkage**.

结论: 将线性约束的凸优化问题转换成单调变分不等式, 再用本文介绍的 PPA 方法求解, 每次迭代中要求解的子问题, 数值代数中都有确定的成熟的方法求解！

数值试验 数值试验例子取自[1]

一个秩为 ra 的 $n \times n$ 的自由度是 $d_{ra} := ra(2n - ra)$.

生成试验问题:

- 先用高斯同分布 (Gaussian i.i.d) 独立生成两个 $n \times ra$ 的矩阵 M_1 和 M_2, 然后令$M = M_1 M_2^T$, 则 $n \times n$ 矩阵 M 的秩为 ra.
- 随机选定 M 的 m 个元素作为已知元素, 这些元素的下标集为 Π.

计算结果:

- 矩阵完整化问题的难度与比率 m/d_{ra} 和 m/n^2 都有关系.
- 分别用 Classical PPA 和 Extended PPA ($\gamma = 1.5$) 进行计算.
- 正定矩阵 G (see (2.10)) 中的参数 $r,\ s$ 分别取 $rs = 1.01$ 和 $r = 0.005$.
- 停机准则采用相对误差 $\|X_\Pi^k - M_\Pi\|_F / \|M_\Pi\|_F \le 10^{-4}$.

我们用 $\gamma = 1.5$ 的 Extended PPA 求解, 注意到 KKT 条件 Primal 部分((2.9) 的上半部分) 的不满足量是

$$r(\tilde{X}^k - X^k) - (\tilde{Z}^k - Z^k).$$

♣ 矩阵完整化的程序在附件的 Codes-04 的文件夹“矩阵完整化”中. 只要运行 demo.m 就可以了. 要对不同情形试验, 只要在 demo.m 中用 % 做适当选择. 其中的 PPAC.m 和 PPAE.m 分别是 Classical PPA 和 Extended PPA 的子程序.

Code 4.b. Creating the test examples of Matrix Completion

```
%%  Creating the test examples of the matrix Completion problem     %(1)
clear all;  clc                                                     %(2)
maxIt=100;          tol = 1e-4;                                     %(3)
r=0.005;            s=1.01/r;          gamma=1.5;                   %(4)
  n=200;              ra = 10;           oversampling = 5;          %(5)
% n=1000;   ra=100;       oversampling = 3; %%  Iteration No. 31     %(6)
% n=1000;   ra=50;        oversampling = 4; %%  Iteration No. 36     %(7)
% n=1000;   ra=10;        oversampling = 6; %%  Iteration No. 78     %(8)
%% Generating the test problem                                      %(9)
rs = randseed;          randn('state',rs);                         %(10)
M=randn(n,ra)*randn(ra,n);             %% The matrix will be completed %(11)
df =ra*(n*2-ra);                       %% The freedom of the matrix   %(12)
mo=oversampling;                                                   %(13)
m =min(mo*df,round(.99*n*n));           %% No. of the known elements  %(14)
Omega= randsample(n^2,m);               %% Define the subset Omega    %(15)
fprintf('Matrix: n=%4d  Rank(M)=%3d  Oversampling=%2d \n',n,ra,mo);%(16)
```

Code 4.3. Extended PPA for Matrix Completion Problem

```
function PPAE(n,r,s,M,Omega,maxIt,tol,gamma)     %  Ititial Process %%(1)
X=zeros(n);      Y=zeros(n);      YT=zeros(n);                        %(2)
  nM0=norm(M(Omega),’fro’);    eps=1;  VioKKT=1;  k=0;   tic;         %(3)
%% Minimum nuclear norm solution by PPA method                        %(4)
while  (eps > tol && k<= maxIt)                                       %(5)
 if mod(k,5)==0                                                       %(6)
 fprintf(’It=%3d |X-M|/|M|=%9.2e VioKKT=%9.2e\n’,k,eps,VioKKT); end;%(7)
  k=k+1;       X0=X;       Y0=Y;                                      %(8)
  YT(Omega)=Y0(Omega)-(X0(Omega)-M(Omega))/s;            EY=Y-YT;    %(9)
  A = X0 + (YT*2-Y0)/r;                      [U,D,V]=svd(A,0);       %(10)
  D=D-eye(n)/r;     D=max(D,0);      XT=(U*D)*V’;        EX=X-XT;    %(11)
  DXM=XT(Omega)-M(Omega);           eps = norm(DXM,’fro’)/nM0;       %(12)
  VioKKT = max( max(max(abs(EX)))*r, max(max(abs(EY))) );            %(13)
   if (eps <= tol)   gamma=1;   end;                                 %(14)
  X = X0 - EX*gamma;                                                 %(15)
  Y(Omega) = Y0(Omega) - EY(Omega)*gamma;                            %(16)
end;                                                                 %(17)
 fprintf(’It=%3d  |X-M|/|M|=%9.2e  Vi0KKT=%9.2e \n’,k,eps,VioKKT); %(18)
   RelEr=norm((X-M),’fro’)/norm(M,’fro’);   toc;                     %(19)
 fprintf(’ Relative error = %9.2e  Rank(X)=%3d \n’,RelEr,rank(X)); %(20)
 fprintf(’ Violation of KKT Condition = %9.2e  \n’,VioKKT);         %(21)
```

矩阵完整化问题：用 Matlab 中标准 SVD 求解结果

Unknown $n \times n$ matrix M				Computational Results			
n	rank(ra)	m/d_{ra}	m/n^2	#iters	times(Sec.)	relative error	KKT-Violation
1000	10	6	0.12	76	841.59	9.38E-5	9.31E-6
1000	50	4	0.39	37	406.24	1.21E-4	2.11E-5
1000	100	3	0.58	31	362.58	1.50E-4	2.88E-5

♣ 用 Matlab 中的 SVD, 做一次 SVD 的花费很大, 总耗时与迭代次数成比例.

♣ 用 PROPACK [8] 中的 SVD, 快许多, 总耗时主要与问题性质有关. 我们主要对迭代次数感兴趣, 只报道用 Matlab 做 SVD 的结果, 也附上所需要的子程序.

对矩阵完整化问题, (2.5) 和 (2.9) 的下半部分误差用 $\|X_\Pi^k - M_\Pi\|_F / \|M_\Pi\|_F$ 控制. 由于 (2.5) 和 (2.9) 的上半部分是

$$\theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T \{(-A^T\tilde{\lambda}^k) + [r(\tilde{x}^k - x^k) \pm A^T(\tilde{\lambda}^k - \lambda^k)]\},$$

因此上半部分误差项是 $[r(\tilde{x}^k - x^k) \mp A^T(\tilde{\lambda}^k - \lambda^k)]$. 对矩阵完整化问题, A是投影矩阵, 其模为 1, 所以我们在计算结果中也列出

$$\text{KKT-Violation} := \max\{r \max_{ij} |X_{ij}^k - \tilde{X}_{ij}^k|, \ \max_{ij} |Z_{ij}^k - \tilde{Z}_{ij}^k|\}.$$

♣ 论文 [1] 是最早发表在 SIAM J. Optimization 上的求解矩阵完整化问题的文章. 对以上试验的三个例子, [1] 中的方法达到同样精度要求的迭代次数分别是 117, 114 和 129 (See the first three examples in Table 5.1 of [1], pp. 1974), 每次迭代的主要工作量也是做一次 SVD 分解. 由于采用了不完全分解技术, [1] 中节省了 SVD 分解的时间. 我们调用的是 Matlab 中标准SVD, 虽然花费了更少的迭代次数, 但没有节省总的运行时间.

3.3 压缩传感(Compressed Sensing)问题

- 压缩传感问题是利用信号的稀疏性质, 只对其进行相对较少的测量, 然后能以较小的误差或者精确地恢复原始信号。
- CS 问题: 求满足 $Ax = b$ 的非零元个数最小的解

$$\min\{\|x\|_0 \mid Ax = b\}.$$

其中$\|x\|_0$ 表示向量 x 的非零元的个数, 它不是常规意义下的模。是一个很难求解的组合优化问题.

$$\|x\|_1 \leq \|x\|_0,\ \forall\ x \in \Omega = \{x \in R^n \mid |x_j| \leq 1,\ j = 1, \ldots, n\}.$$

Candés 和 Tao 证明了在一定条件下, 问题

$$\min\{\|x\|_0\ :\ Ax = b\} \quad \text{与问题} \quad \min\{\|x\|_1\ :\ Ax = b\}$$

的解在概率意义下是相同的.

极小化问题

$$\min_x \{ \mu\|x\|_1 \mid Ax = b \} \qquad \text{(BP)} \tag{3.10}$$

是形如 (1.3) 的等式约束凸优化问题.

PPA 算法求解问题(3.10)

对给定的 (x^k, λ^k), 用 (2.11)–(2.12) 产生$(\tilde{x}^k, \tilde{\lambda}^k)$:

1. Producing $\tilde{\lambda}^k$ by

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}(Ax^k - b).$$

2. Finding $\tilde{x}^k$ which is the solution of the following minimization problem

$$\min\{ \mu\|x\|_1 + \frac{r}{2}\|x - [x^k + \frac{1}{r}A^T(2\tilde{\lambda}^k - \lambda^k)]\|^2\}. \tag{3.11}$$

子问题 (3.11) 求解的具体做法: 我们只要考虑如何求解

$$\min_{x\in\Re^n} \{\tau\|x\|_1 + \frac{1}{2}\|x - a\|^2\}. \tag{3.12}$$

设 $x^* \in \Re^n$ 是 (3.12) 的解,

- 如果 $x_j^* > 0$, 对 $\tau x + \frac{1}{2}\|x - a\|^2$ 求导, 有

$$\tau + x_j^* - a_j = 0, \quad 即 \quad x_j^* = a_j - \tau, \qquad (这隐含了 \quad a_j > \tau).$$

- 如果 $x_j^* < 0$, 对 $-\tau x + \frac{1}{2}\|x - a\|^2$ 求导, 有

$$-\tau + x_j^* - a_j = 0, \quad 即 \quad x_j^* = a_j + \tau, \qquad (这隐含了 \quad a_j < -\tau).$$

- 为什么 $x_j^* = 0$, 是因为 $-\tau \le a_j \le \tau$.

综上所述, 问题 (3.12) 的解由

$$x_j^* = \begin{cases} a_j - \tau, & \text{if} \quad a_j > \tau \\ a_j + \tau, & \text{if} \quad a_j < -\tau \\ 0, & \text{if} \quad -\tau \le a_j \le \tau. \end{cases}$$

给出. 这也可以写成 Shrinkage 的形式

$$x^* = a - P_{B_\infty^\tau}[a], \qquad \text{where} \qquad B_\infty^\tau = \{\xi \in \Re^n \mid -\tau e \le \xi \le \tau e\}.$$

实际 CS 问题的计算, 要加进一些 Continuation 技术, 限于篇幅, 不详细介绍.

3.4 Min-Max 问题上的应用

第一讲 §4.2 中提到用全变差极小处理图像去模糊 [2], 经离散化以后, 问题的数学模型是的 min-max 问题,

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \ \Phi(x,y) := \theta_1(x) + y^T Ax - \theta_2(y). \tag{3.13}$$

它可以转换成等价的变分不等式: 求 $(x^*, y^*) \in \mathcal{X}\times\mathcal{Y}$, 使得

$$\begin{pmatrix} x - x^* \\ y - y^* \end{pmatrix}^T \begin{pmatrix} f(x^*) + A^T y^* \\ g(y^*) - Ax^* \end{pmatrix} \geq 0, \quad \forall (x,y)\in\mathcal{X}\times\mathcal{Y}, \tag{3.14}$$

其中 $f(x)\in\partial\theta_1(x)$, $g(y)\in\partial\theta_2(y)$. 文献 [6] 中提到用这里的 PPA 方法去求解. 对给定的 (x^k, y^k), 求 $(\tilde{x}^k, \tilde{y}^k)\in\mathcal{X}\times\mathcal{Y}$, 对一切 $(x,y)\in\mathcal{X}\times\mathcal{Y}$, 都有

$$\begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} f(\tilde{x}^k) + A^T\tilde{y}^k \\ g(\tilde{y}^k) - A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) + A^T(\tilde{y}^k - y^k) \\ +A(\tilde{x}^k - x^k) + s(\tilde{y}^k - y^k) \end{pmatrix} \right\} \geq 0. \tag{3.15}$$

注意到 (3.15) 的下半部分中只含有未知的 $\tilde{y}^k$. 它可以通过求解 y 的子问题:

$$\tilde{y}^k = \text{Argmin}\{\theta_2(y) + y^T A x^k + \frac{s}{2}\|y - y^k\|^2 \mid y \in \mathcal{Y}\}. \tag{3.16a}$$

得到. 然后, 为求得 (3.15) 上半部分得 $\tilde{x}^k$, 我们只要求解一个关于 x 的问题:

$$\tilde{x}^k = \text{Argmin}\Big\{\theta_1(x) + \frac{r}{2}\Big\|x - \big[x^k - \frac{1}{r}A^T(2\tilde{y}^k - y^k)\big]\Big\|^2 \,\Big|\, x \in \mathcal{X}\Big\}. \tag{3.16b}$$

当然, PPA 预测点 $(\tilde{x}^k, \tilde{y}^k) \in \mathcal{X} \times \mathcal{Y}$ 也可以通过先 x, 后 y 的顺序完成.

对给定的 (x^k, y^k), 通过求解 x 的子问题

$$\tilde{x}^k = \text{Argmin}\{\theta_1(x) + x^T A^T y^k + \frac{r}{2}\|x - x^k\|^2 \mid x \in \mathcal{X}\}. \tag{3.17a}$$

得到 $\tilde{x}^k$, 然后求 $\tilde{y}^k$ 通过一个关于 y 的子问题:

$$\tilde{y}^k = \text{Argmin}\Big\{\theta_2(y) + \frac{s}{2}\Big\|y - \big[y^k + \frac{1}{s}A(2\tilde{x}^k - x^k)\big]\Big\|^2 \,\Big|\, y \in \mathcal{Y}\Big\}. \tag{3.17b}$$

由 (3.17) 生成的 $(\tilde{x}^k, \tilde{y}^k) \in \mathcal{X} \times \mathcal{Y}$, 对一切 求 $(x, y) \in \mathcal{X} \times \mathcal{Y}$, 都有

$$\begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} f(\tilde{x}^k) + A^T \tilde{y}^k \\ g(\tilde{y}^k) - A\tilde{x}^k \end{pmatrix} + \begin{pmatrix} r(\tilde{x}^k - x^k) - A^T(\tilde{y}^k - y^k) \\ -A(\tilde{x}^k - x^k) + s(\tilde{y}^k - y^k) \end{pmatrix} \right\} \geq 0. \tag{3.18}$$

无论是 (3.15) 还是 (3.18), 都可以写成

$$\tilde{u}^k \in \Omega, \qquad (u - \tilde{u}^k)^T \{F(\tilde{u}^k) + G(\tilde{u}^k - u^k)\} \geq 0, \quad \forall\, u \in \Omega, \tag{3.19}$$

的形式, 所不同的只是

$$G = \begin{pmatrix} rI_n & A^T \\ A & sI_m \end{pmatrix} \quad \text{和} \quad G = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix}.$$

它们都是当 $rs > \|A^T A\|$ 时候正定. 我们建议新的迭代点u^{k+1} 用迭代式

$$u^{k+1} = u^k - \gamma(u^k - \tilde{u}^k), \quad \gamma \in [1, 2),$$

生成, 一般取 $\gamma = 1.5$. 对一些工程技术上的问题, 采取 $\gamma > 1$ 的迭代满足停机准则以后, 最后加一次 $\gamma = 1$ 的迭代. 这样既提高了速度, 又满足实际问题对诸如 Low-Rank 这样的要求. 这类方法在图像处理中的应用可见论文 [6, 7].

定制的 PPA 算法具有 $O(1/k)$ 的收敛速率, 证明可以在论文 [4] 中找到.

References

[1] J. F. Cai, E. J. Candès and Z. W. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. **20**, 1956-1982, 2010.

[2] A. Chambolle and T. Pock, A first-order primal-dual algorithms for convex problem with applications to imaging, J. Math. Imaging Vison, **40**, 120-145, 2011.

[3] B. S. He, X. L. Fu, Z.K. Jiang, Proximal point algorithm using a linear proximal term, JOTA **141**, 209-239, 2009.

[4] B. S. He and Y. Shen, On the convergence rate of customized proximal point algorithm for convex optimization and saddle-point problem (in Chinese). Sic Sin Math, **42**, 515-525, 2012. doi: 10.1360/012011-1049

[5] B. S. He and M.-H. Xu, A general framework of contraction methods for monotone variational inequalities, Pacific J. Optimization **4**, 195-212, 2008.

[6] B.S. He and X. M. Yuan, Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective, SIAM Journal on Imaging Science **5**, pp. 119-149, 2012.

[7] B. S. He, X. M. Yuan and W.X. Zhang, A customized proximal point algorithm for convex minimization with linear constraints, Comput. Optim. Appl., **56**, pp. 559-572, 2013.

[8] R. M. Larsen, Propack − software for large and sparse svd calculation, http://sun.stanford.edu/rmunk/PROPACK/, 2005.

[9] B. Martinet, Regularisation, d'inéquations variationelles par approximations succesives, Rev. Francaise d'Inform. Recherche Oper., **4**, 154-159, 1970.

[10] Netflix, Netflix Prize, Available at http://www.netflixprize.com/. 2006.

[11] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Cont. Optim. **14**, 877-898, 1976.

[12] M. Zhu and T. F. Chan, An efficient primal-dual hybrid gradient algorithm for total variation image restoration, CAM Reports 08-34, UCLA, 2008.