Scalable Robust Matrix Recovery: Frank-Wolfe Meets Proximal Methods

Cun Mu,Yuqian Zhang,John Wright,Donald Goldfarb （2017）

2020.6.19

王佳文

- 1.文章摘要
- 2.问题背景
- 3.问题描述
- 4.问题求解
- 5.数值实验

1. 文章摘要

从压缩和严重破坏的观测数据中恢复矩阵是稳健统计中的一个基本问题，在计算机视觉和机器学习中有着丰富的应用。理论上，在一定的条件下，这个问题可以通过一个自然的凸松弛在多项式时间内求解，称为压缩主成分追踪(CPCP)。现有的许多可证明收敛的CPCP算法每次都存在**超线性**的迭代代价，这严重限制了它们对大规模问题的适用性。在本文中提出了可证明的收敛性、可伸缩性和有效的方法来求解具有每次迭代代价为**线性**的CPCP。主要结合了**FW算法**和**临近法**。在每次迭代中，利用**FW法以及秩一SVD来更新低秩项**，并利用**临近梯度法更新稀疏项**。讨论了收敛结果和实现细节。通过对可视化数据的数值实验，证明了该方法的实用性和可扩展性。

2.问题背景

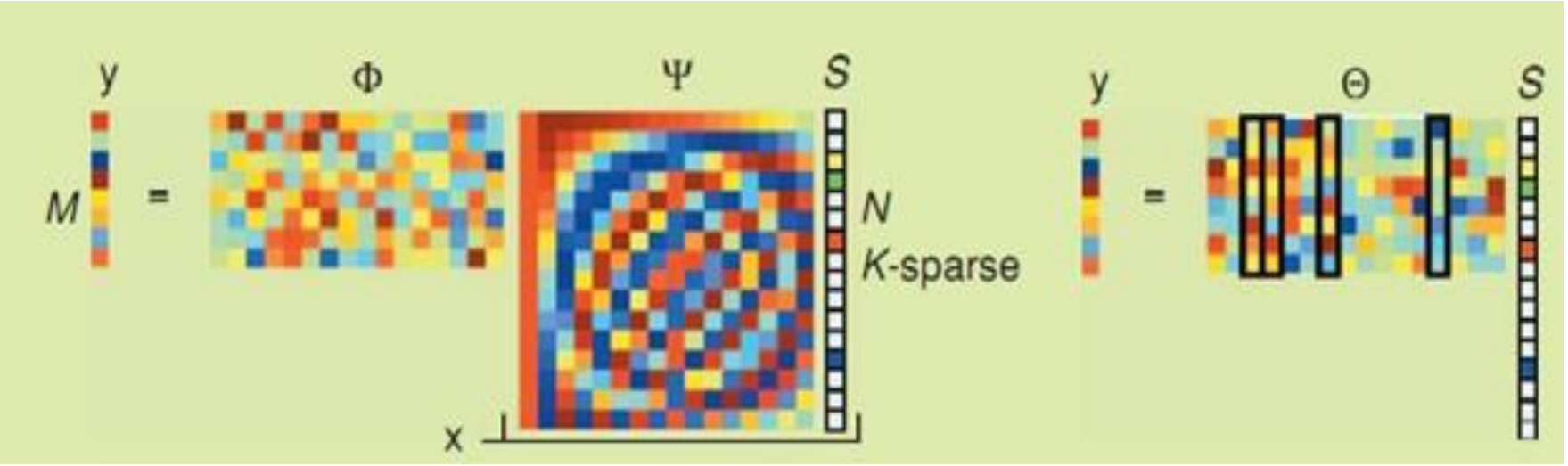
背景建模的最简单情形是从固定摄相机拍摄的视频中分离背景和前景。将视频图像序列的每一帧图像像素值拉成一个列向量，那么多个帧也就是多个列向量就组成了一个观测矩阵。由于背景比较稳定，图像序列帧与帧之间具有极大的相似性，所以仅由背景像素组成的矩阵具有低秩特性；同时由于前景是移动的物体，占据像素比例较低，故前景像素组成的矩阵具有稀疏特性。视频观测矩阵就是这两种特性矩阵的叠加，因此，可以说视频背景建模实现的过程就是低秩矩阵恢复的过程。

3. 问题描述

3.1 压缩感知简介

- 设$\mathbf{x} \in \mathbb{R}^{n\times 1}$, 稀疏度为$k$(即含有$k$个非零值)，$\boldsymbol{\phi} \in \mathbb{R}^{m\times n}(m < n)$, $\mathbf{y} = \boldsymbol{\phi}\mathbf{x}$为长度为$m$的一维测量值。
- 压缩感知问题就是已知测量值$\mathbf{y}$和测量矩阵$\boldsymbol{\phi}$的基础上，求解欠定方程组$\mathbf{y} = \boldsymbol{\phi}\mathbf{x}$得到原信号$\mathbf{x}$。
- $\boldsymbol{\phi}$的每一行可以看作是一个传感器，它与信号相乘，拾取了信号的一部分信息。而这一部分信息足以代表原信号，并能找到一个算法来高概率恢复原信号。

- 一般的自然信号$\mathbf{x}$本身并不是稀疏的，需要在某种稀疏基上进行稀疏表示，$\mathbf{x}=\boldsymbol{\psi}\mathbf{s}$，$\boldsymbol{\psi}$为稀疏基矩阵，$\mathbf{s}$为稀疏系数($\mathbf{s}$只有$k$个是非零值($k << m$)。压缩感知方程为$\mathbf{y}=\boldsymbol{\phi}\mathbf{x}=\boldsymbol{\phi}\boldsymbol{\psi}\mathbf{s}=\boldsymbol{\Theta}\mathbf{s}$。将原来的测量矩阵$\boldsymbol{\phi}$变换为$\boldsymbol{\Theta}=\boldsymbol{\phi}\boldsymbol{\psi}$(称之为传感矩阵)，解出$\mathbf{s}$的逼近值$\mathbf{s}'$，则原信号$\mathbf{x}'=\boldsymbol{\psi}\mathbf{s}'$。



3.2. 问题

- 假设 $\boldsymbol{M_0} \in \mathbb{R}^{m\times n}$，$\boldsymbol{M}_0 = \boldsymbol{L}_0 + \boldsymbol{S}_0 + \boldsymbol{N}_0$，$\boldsymbol{L}_0$是一个低秩矩阵，$\boldsymbol{S_0}$是一个稀疏矩阵，$\boldsymbol{N_0}$是一个密度噪声矩阵。
- 定义：

$$(1.1)\quad \mathbf{b} = \boldsymbol{A}[\boldsymbol{M}_0] = \left(\langle \boldsymbol{A}_1, \boldsymbol{M}_0\rangle, \langle \boldsymbol{A}_2, \boldsymbol{M}_0\rangle, \cdots, \langle \boldsymbol{A}_{\mathrm{p}}, \boldsymbol{M}_0\rangle\right)^T \in \mathbb{R}^p$$

其中 $\boldsymbol{A}: \mathbb{R}^{m\times n} \to \mathbb{R}^p$ 是感知算子，$\mathbf{A}_k$ 表示感知矩阵，且：

$$\langle \boldsymbol{A}_{\mathrm{k}}, \boldsymbol{M}_0\rangle = \mathrm{Tr}\left(\boldsymbol{M}_0{}^{\boldsymbol{T}}\, \boldsymbol{A}_{\mathrm{k}}\right)$$

- 在给定 $\boldsymbol{A}$ 时，我们可以有效的从 $\mathbf{b}$ 中恢复 $\boldsymbol{L}_0$和 $\boldsymbol{S}_0$ 吗？

3.3 压缩主成分追踪(CPCP)问题

- 自然的，可以用如下方式进行优化：

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ \frac{1}{2}\|\boldsymbol{b}-\mathcal{A}[\boldsymbol{L}+\boldsymbol{S}]\|_2^2+\lambda_L \mathrm{rank}(\boldsymbol{L})+\lambda_S\|\boldsymbol{S}\|_0\,. \tag{1.2}$$

缺点：非凸，无法直接处理
转化为：

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ \frac{1}{2}\|\boldsymbol{b}-\mathcal{A}[\boldsymbol{L}+\boldsymbol{S}]\|_2^2+\lambda_L\|\boldsymbol{L}\|_*+\lambda_S\|\boldsymbol{S}\|_1 \tag{1.3}$$

其中$\|\boldsymbol{L}\|_*$定义为$\boldsymbol{L}$的奇异值之和，$\|\mathbf{S}\|_1=\sum_{i=1}\sum_{j=1}|\mathbf{S}_{i,j}|$
(1.3)式有时被称为压缩主成分追踪(CPCP)问题.

- 等价的，由于：

$$\left\{ \boldsymbol{M} \in \mathbb{R}^{m\times n} \mid \boldsymbol{b} = \mathcal{A}[\boldsymbol{M}] \right\} = \left\{ \boldsymbol{M} \in \mathbb{R}^{m\times n} \mid \mathcal{P}_Q[\boldsymbol{M}] = \mathcal{P}_Q[\boldsymbol{M}_0] \right\}$$

其中 $Q \subseteq \mathbb{R}^{m\times n}$ 是由感知矩阵 $\{\mathbf{A}_i\}_{i=1}^{p}$ 张成的子空间，$\mathcal{P}_Q$ 表示在子空间上的投影算子。

- 可将原问题转化为：

$$\min_{\boldsymbol{L},\boldsymbol{S}} \; f(\boldsymbol{L},\boldsymbol{S}) \doteq \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}_0]\right\|_F^2 + \lambda_L \left\|\boldsymbol{L}\right\|_* + \lambda_S \left\|\boldsymbol{S}\right\|_1 \tag{1.4}$$

注意问题的目标函数不可微且可行集无界。

Recently, CPCP and its close variants have been studied for different sensing operators $\mathcal{A}$ (or equivalently different subspaces $\mathcal{Q}$). In specific, [2, 3, 4, 5, 6] consider the case where a subset $\Omega \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$ of the entries of $\boldsymbol{M}_0$ is observed. Then CPCP can be reduced to

$$\min_{\boldsymbol{L},\boldsymbol{S}} \quad \frac{1}{2} \left\| \mathcal{P}_\Omega[\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{M}_0] \right\|_F^2 + \lambda_L \left\| \boldsymbol{L} \right\|_* + \lambda_S \left\| \boldsymbol{S} \right\|_1 , \tag{1.5}$$

where $\mathcal{P}_\Omega[\cdot]$ denotes the orthogonal projection onto the linear space of matrices supported on Ω, i.e., $\mathcal{P}_\Omega[\boldsymbol{M}_0](i,j) = (\boldsymbol{M}_0)_{ij}$ if $(i,j) \in \Omega$ and $\mathcal{P}_\Omega[\boldsymbol{M}_0](i,j) = 0$ otherwise. [1] studies the case where each $\mathcal{A}_k$ is an i.i.d. $\mathcal{N}(0,1)$ matrix, which is equivalent (in distribution) to saying that we choose a linear subspace $\mathcal{Q}$ uniformly at random from the set of all p-dimensional subspaces of $\mathbb{R}^{m \times n}$ and observe $\mathcal{P}_Q[\boldsymbol{M}_0]$. Accordingly, all the above provide theoretical guarantees for CPCP, under fairly mild conditions, to produce accurate estimates of $\boldsymbol{L}_0$ and $\mathcal{P}_\Omega[\boldsymbol{S}_0]$ (or $\boldsymbol{S}_0$), even when the number of measurements p is substantially less than mn.

4. 问题求解

4.1 FW算法

- 设h是一紧集上的可微凸函数.

$$\text{minimize} \quad h(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in \mathcal{D} \subseteq \mathbb{R}^n \tag{2.1}$$

假设∇h是L利普希茨,

在第k次迭代，在点 $\boldsymbol{x}^k$ 处线性化目标函数h得:

$$\forall\, \boldsymbol{x},\, \boldsymbol{y} \in \mathcal{D}, \qquad \|\nabla h(\boldsymbol{x}) - \nabla h(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\| \tag{2.2}$$

$$h(\boldsymbol{v}) \approx h(\boldsymbol{x}^k) + \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v} - \boldsymbol{x}^k \right\rangle \tag{2.3}$$

$$\boldsymbol{v}^k \in \arg\min_{\boldsymbol{v} \in \mathcal{D}} \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v} \right\rangle \tag{2.4}$$

然后在可行下降方向 $\boldsymbol{v}^k - \boldsymbol{x}^k$ 上采取:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \frac{2}{k+2}(\boldsymbol{v}^k - \boldsymbol{x}^k) \tag{2.5}$$

Algorithm 1 Frank-Wolfe method for problem (2.1)

1: **Initialization:** $\boldsymbol{x}^0 \in \mathcal{D}$;
2: **for** $k = 0,\ 1,\ 2,\ \ldots$ **do**
3: $\quad \boldsymbol{v}^k \in \operatorname{argmin}_{\boldsymbol{v}\in\mathcal{D}} \langle \boldsymbol{v}, \nabla h(\boldsymbol{x}^k)\rangle$;
4: $\quad \gamma = \frac{2}{k+2}$;
5: $\quad \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k)$;
6: **end for**

Algorithm 2 Frank-Wolfe method for problem (2.1) with general updating scheme

1: **Initialization:** $\boldsymbol{x}^0 \in \mathcal{D}$;
2: **for** $k = 0,\ 1,\ 2,\ \ldots$ **do**
3: $\quad \boldsymbol{v}^k \in \operatorname{argmin}_{\boldsymbol{v}\in\mathcal{D}} \langle \boldsymbol{v}, \nabla h(\boldsymbol{x}^k)\rangle$;
4: $\quad \gamma = \frac{2}{k+2}$;
5: $\quad$ Update $\boldsymbol{x}^{k+1}$ to some point in $\mathcal{D}$ such that $h(\boldsymbol{x}^{k+1}) \le h(\boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k))$;
6: **end for**

THEOREM 2.1. *Let $\boldsymbol{x}^\star$ be an optimal solution to (2.1). For $\{\boldsymbol{x}^k\}$ generated by Algorithm 2, we have for $k = 0,\ 1,\ 2,\ \ldots,$*

$$h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star) \leq \frac{2LD^2}{k+2}. \tag{2.9}$$

Proof. For $k = 0,\ 1,\ 2,\ \ldots,$ we have

$$\begin{aligned} h(\boldsymbol{x}^{k+1}) &\leq h(\boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k)) \\ &\leq h(\boldsymbol{x}^k) + \gamma \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + \frac{L\gamma^2}{2} \left\| \boldsymbol{v}^k - \boldsymbol{x}^k \right\|^2 \\ &\leq h(\boldsymbol{x}^k) + \gamma \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + \frac{\gamma^2 LD^2}{2} \qquad (2.10) \\ &\leq h(\boldsymbol{x}^k) + \gamma \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{x}^\star - \boldsymbol{x}^k \right\rangle + \frac{\gamma^2 LD^2}{2} \\ &\leq h(\boldsymbol{x}^k) + \gamma(h(\boldsymbol{x}^\star) - h(\boldsymbol{x}^k)) + \frac{\gamma^2 LD^2}{2}, \qquad (2.11) \end{aligned}$$

其中 $D = \max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}} \|\boldsymbol{x} - \boldsymbol{y}\|$ 表示可行集 $\mathcal{D}$ 的直径

Rearranging terms in (2.11), one obtains that for $k = 0,\ 1,\ 2,\ \ldots,$

$$h(\boldsymbol{x}^{k+1}) - h(\boldsymbol{x}^\star) \leq (1-\gamma)\left(h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star)\right) + \frac{\gamma^2 L D^2}{2}. \tag{2.12}$$

Therefore, by mathematical induction, it can be verified that

$$h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star) \leq \frac{2LD^2}{k+2}, \quad \text{for } k = 1,\ 2,\ 3,\ \ldots.$$

REMARK 1. *Note that the constant in the rate of convergence depends on the Lipschitz constant L of h and the diameter $\mathcal{D}$.*

4.2 对偶间隙

对偶间隙： $d(\boldsymbol{x}^k) = \langle \boldsymbol{x}^k - \boldsymbol{v}^k, \nabla h(\boldsymbol{x}^k) \rangle$

- 在 $h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star)$上提供一个有用的上界

$$
\begin{aligned}
h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star) &\leq -\langle \boldsymbol{x}^\star - \boldsymbol{x}^k, \nabla h(\boldsymbol{x}^k) \rangle \\
&\leq -\min_{\boldsymbol{v}} \langle \boldsymbol{v} - \boldsymbol{x}^k, \nabla h(\boldsymbol{x}^k) \rangle = \langle \boldsymbol{x}^k - \boldsymbol{v}^k, \nabla h(\boldsymbol{x}^k) \rangle = d(\boldsymbol{x}^k)
\end{aligned}
$$

THEOREM 2.2. *Let $\{\boldsymbol{x}^k\}$ be the sequence generated by Algorithm 2. Then for any $K \geq 1$, there exists $1 \leq \tilde{k} \leq K$ such that*

$$
d(\boldsymbol{x}^{\tilde{k}}) \leq \frac{6LD^2}{K+2}. \tag{2.15}
$$

REMARK 2. *The convergence rate for the duality gap matches the one for $h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star)$ (see (2.9)), which suggests that the upper bound $d(\boldsymbol{x}^k)$ can serve as a practical stopping criterion.*

4.3 几个重要的方法

Minimizing a linear function over the nuclear norm ball. Since the dual norm of the nuclear norm is the operator norm, i.e., $\|\boldsymbol{Y}\| = \max_{\|\boldsymbol{X}\|_* \le 1} \langle \boldsymbol{Y}, \boldsymbol{X} \rangle$, the optimization problem

$$\text{minimize}_{\boldsymbol{X}} \ \langle \boldsymbol{Y}, \boldsymbol{X} \rangle \qquad \text{subject to } \|\boldsymbol{X}\|_* \le 1 \tag{2.19}$$

has optimal value $-\|\boldsymbol{Y}\|$. One minimizer is the rank-one matrix $\boldsymbol{X}^\star = -\boldsymbol{u}\boldsymbol{v}^\top$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the left- and right- singular vectors corresponding to the leading singular value of $\boldsymbol{Y}$, and can be efficiently computed (e.g. using power method).

Minimizing a linear function over the ℓ_1 ball. Since the dual norm of the ℓ_1 norm is the ℓ_∞ norm, i.e., $\|\boldsymbol{Y}\|_\infty := \max_{(i,j)} |Y_{ij}| = \max_{\|\boldsymbol{X}\|_1 \le 1} \langle \boldsymbol{Y}, \boldsymbol{X} \rangle$, the optimization problem

$$\text{minimize}_{\boldsymbol{X}} \ \langle \boldsymbol{Y}, \boldsymbol{X} \rangle \qquad \text{subject to } \|\boldsymbol{X}\|_1 \le 1 \tag{2.20}$$

has optimal value $-\|\boldsymbol{Y}\|_\infty$. One minimizer is the one-sparse matrix

$$\boldsymbol{X}^\star = -\text{sgn}(Y_{i^\star j^\star}) \boldsymbol{e}_{i^\star} \boldsymbol{e}_{j^\star}^\top,$$

where $(i^\star, j^\star) \in \arg\max_{(i,j)} |Y_{ij}|$; i.e. $\boldsymbol{X}^\star$ has exactly one nonzero element.

Projection onto the ℓ_1-ball. To effectively handle the sparse term in the norm constrained problem (1.6), we will need to modify the Frank-Wolfe algorithm by incorporating additional projection steps. For any $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$ and $\beta > 0$, the projection onto the ℓ_1-ball:

$$\mathcal{P}_{\|\cdot\|_1\leq\beta}[\boldsymbol{Y}] = \arg\min_{\|\boldsymbol{X}\|_1\leq\beta} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}\|_F^2, \tag{2.21}$$

can be easily solved with $O(mn(\log m + \log n))$ cost [32]. Moreover, a divide and conquer algorithm, achieving linear cost in expectation to solve (2.21), has also been proposed in [32].

Proximal mapping of ℓ_1 norm. To effectively handle the sparse term arising in problem (1.4), we will need to modify the Frank-Wolfe algorithm by incorporating additional proximal steps. For any $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$ and $\lambda > 0$, the proximal mapping of ℓ_1 norm has the following closed-form expression

$$\mathcal{T}_\lambda[\boldsymbol{Y}] = \arg\min_{\boldsymbol{X}\in\mathbb{R}^{m\times n}} \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}\|_F^2 + \lambda\|\boldsymbol{X}\|_1, \tag{2.22}$$

where $\mathcal{T}_\lambda : \mathbb{R} \to \mathbb{R}$ denotes the soft-thresholding operator $\mathcal{T}_\lambda(x) = \operatorname{sgn}(x)\max\{|x| - \lambda, 0\}$, and extension to matrices is obtained by applying the scalar operator $\mathcal{T}_\lambda(\cdot)$ to each element.

4.4 范数约束问题的FW-P方法

- 考虑问题:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ l(\boldsymbol{L},\boldsymbol{S}) = \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\right\|_F^2 \quad \text{s.t.} \quad \|\boldsymbol{L}\|_* \le \tau_L,\ \|\boldsymbol{S}\|_1 \le \tau_S \tag{3.1}$$

注意到目标函数可微，有:

$$\nabla_{\boldsymbol{L}} l(\boldsymbol{L},\boldsymbol{S}) = \mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}] \tag{3.2}$$

$$\nabla_{\boldsymbol{S}} l(\boldsymbol{L},\boldsymbol{S}) = \mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}] \tag{3.3}$$

$$\nabla l(\boldsymbol{L},\boldsymbol{S}) = (\nabla_{\boldsymbol{L}} l, \nabla_{\boldsymbol{S}} l)$$

- 第k次FW算法产生的迭代点记为 $\boldsymbol{x}^k = (\boldsymbol{L}^k, \boldsymbol{S}^k)$, $\boldsymbol{v}^k = (\boldsymbol{V}_L^k, \boldsymbol{V}_S^k)$由下式得到:

$$\begin{pmatrix} \boldsymbol{V}_L^k \\ \boldsymbol{V}_S^k \end{pmatrix} \in \arg\min \left\langle \begin{pmatrix} \mathcal{P}_Q[\boldsymbol{L}^k+\boldsymbol{S}^k-\boldsymbol{M}] \\ \mathcal{P}_Q[\boldsymbol{L}^k+\boldsymbol{S}^k-\boldsymbol{M}] \end{pmatrix}, \begin{pmatrix} \boldsymbol{V}_L \\ \boldsymbol{V}_S \end{pmatrix} \right\rangle \tag{3.5}$$

$$\text{s.t.} \qquad \|\boldsymbol{V}_L\|_* \le \tau_L, \ \ \|\boldsymbol{V}_S\|_1 \le \tau_S,$$

- （3.5）是可分成两个独立的子问题：

$$\boldsymbol{V}_L^k \in \arg\min_{\|\boldsymbol{V}_L\|_* \le \tau_L} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{V}_L \rangle$$

$$\boldsymbol{V}_S^k \in \arg\min_{\|\boldsymbol{V}_S\|_1 \le \tau_S} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{V}_S \rangle.$$

- 容易解得：

$$\boldsymbol{V}_L^k = -\tau_L \boldsymbol{u}^k (\boldsymbol{v}^k)^\top, \tag{3.6}$$

$$\boldsymbol{V}_S^k = -\tau_S \cdot \delta_{i^\star j^\star}^k \cdot \boldsymbol{e}_{i^\star}^k (\boldsymbol{e}_{j^\star}^k)^\top \tag{3.7}$$

- 其中$\boldsymbol{u}^k$ 和 $\boldsymbol{v}^k$ 是 $\mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]$ 的对应最大奇异值的左右奇异向量。$(i^\star, j^\star)$是 $\mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]$ 中最大元对应的位置。

$$\delta_{ij}^k := \operatorname{sgn}\left[\left(\mathcal{P}_{\mathcal{Q}}\left[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}\right]\right)_{ij}\right]$$

Algorithm 3 Frank-Wolfe method for problem (3.1)

1: **Initialization:** $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$;
2: **for** $k = 0,\ 1,\ 2,\ \cdots$ **do**
3: $\quad \boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L \rangle$; $\boldsymbol{V}_L^k = \tau_L \boldsymbol{D}_L^k$;
4: $\quad \boldsymbol{D}_S^k \in \arg\min_{\|\boldsymbol{D}_S\|_1 \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S \rangle$; $\boldsymbol{V}_S^k = \tau_S \boldsymbol{D}_S^k$;
5: $\quad \gamma = \frac{2}{k+2}$;
6: $\quad \boldsymbol{L}^{k+1} = \boldsymbol{L}^k + \gamma(\boldsymbol{V}_L^k - \boldsymbol{L}^k)$;
7: $\quad \boldsymbol{S}^{k+1} = \boldsymbol{S}^k + \gamma(\boldsymbol{V}_S^k - \boldsymbol{S}^k)$;
8: **end for**

- 不足: $\boldsymbol{S}$ 每次迭代只进行一次稀疏更新,由于$\boldsymbol{V}_S^k = -\tau_S \boldsymbol{e}_{i^\star}^k (\boldsymbol{e}_{j^\star}^k)^\top$ 只有一个非零元，实际应用中是一个明显的缺点，作为最优的 $\boldsymbol{S}^\star$ 可能有相对较多的非零项

4.5 FW-P算法:将FW和投影梯度相结合

- 在第k次迭代, $(\boldsymbol{L}^{k+1/2}, \boldsymbol{S}^{k+1/2})$ 由FW算法产生
- 对稀疏项 $\boldsymbol{S}$ 增加一个额外的投影梯度步骤

$$\boldsymbol{S}^{k+1} = \mathcal{P}_{\|\cdot\|_1 \le \tau_S}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \nabla_{\boldsymbol{S}} l(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}})\right] \tag{3.8}$$

$$= \mathcal{P}_{\|\cdot\|_1 \le \tau_S}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}]\right] \tag{3.9}$$

Algorithm 4 FW-P method for problem (3.1)

Initialization: $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$;
for $k = 0,\ 1,\ 2,\ \cdots$ **do**
  $\boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L\rangle$; $\boldsymbol{V}_L^k = \tau_L \boldsymbol{D}_L^k$;
  $\boldsymbol{D}_S^k \in \arg\min_{\|\boldsymbol{D}_S\|_1 \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S\rangle$; $\boldsymbol{V}_S^k = \tau_S \boldsymbol{D}_S^k$;
  $\gamma = \frac{2}{k+2}$;
  $\boldsymbol{L}^{k+\frac{1}{2}} = \boldsymbol{L}^k + \gamma(\boldsymbol{V}_L^k - \boldsymbol{L}^k)$;
  $\boldsymbol{S}^{k+\frac{1}{2}} = \boldsymbol{S}^k + \gamma(\boldsymbol{V}_S^k - \boldsymbol{S}^k)$;
  $\boldsymbol{S}^{k+1} = \mathcal{P}_{\|\cdot\|_1 \le \tau_S}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}]\right]$;
  $\boldsymbol{L}^{k+1} = \boldsymbol{L}^{k+\frac{1}{2}}$;
end for

4.6 使用合成数据的例子

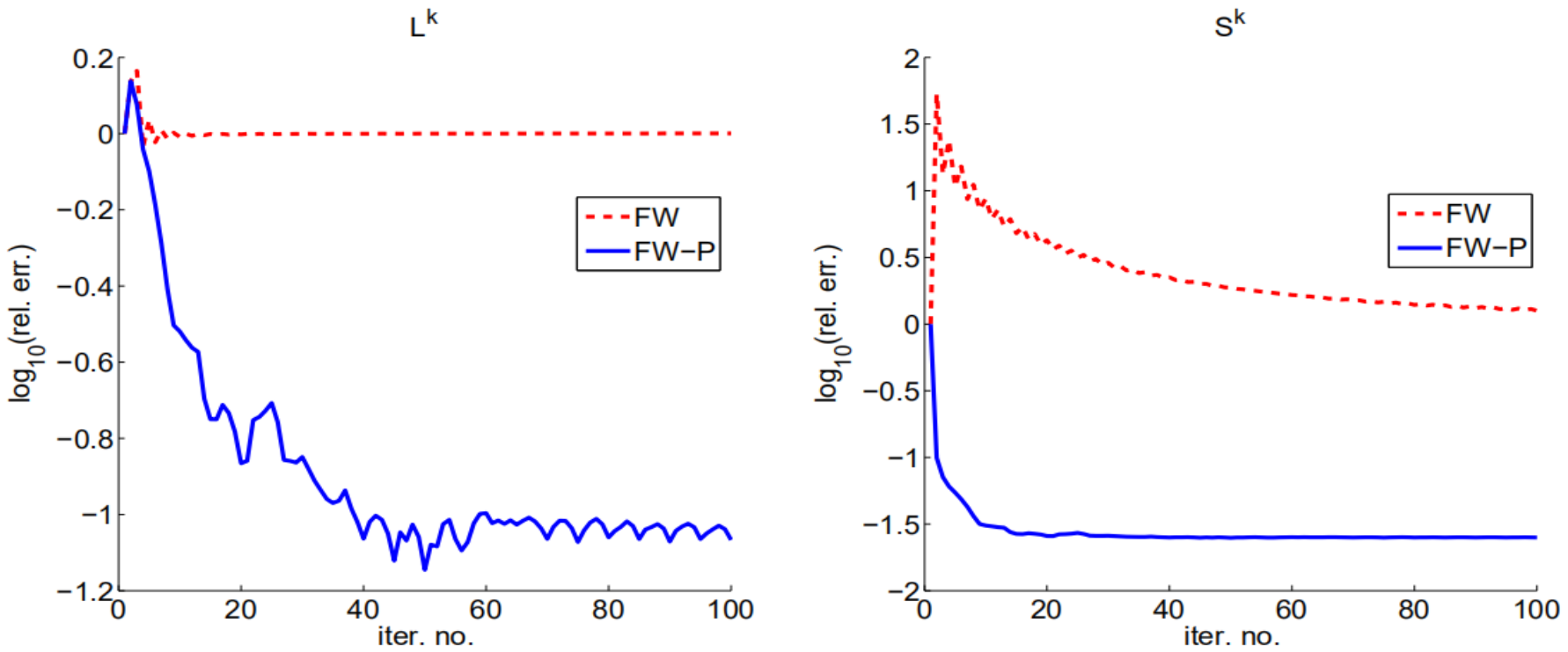


FIG. 1. ***Comparisons between Algorithms 3 and 4 for problem*** (3.1) ***on synthetic data.*** *The data are generated in Matlab as* $\mathtt{m} = 1000$; $\mathtt{n} = 1000$; $\mathtt{r} = 5$; $\mathtt{L0} = \mathtt{randn(m,r)} * \mathtt{randn(r,n)}$; $\mathtt{Omega} = \mathtt{ones(m,n)}$; $\mathtt{S0} = 100 * \mathtt{randn(m,n)}.*(\mathtt{rand(m,n)} < 0.01)$; $\mathtt{M} = \mathtt{L0} + \mathtt{S0} + \mathtt{randn(m,n)}$; $\tau_{\mathtt{L}} = \mathtt{norm\_nuc(L0)}$; $\tau_{\mathtt{S}} = \mathtt{norm(vec(S0),1)}$; *The left figure plots* $\log_{10}(\|\boldsymbol{L}^k - \boldsymbol{L}_0\|_F / \|\boldsymbol{L}_0\|_F)$ *versus the iteration number* k. *The right figure plots* $\log_{10}(\|\boldsymbol{S}^k - \boldsymbol{S}_0\|_F / \|\boldsymbol{S}_0\|_F)$ *versus* k. *The FW-P method is clearly more efficient than the straightforward FW method in recovering* $\boldsymbol{L}_0$ *and* $\boldsymbol{S}_0$.

LEMMA 3.2. *The feasible set* $\mathcal{D} = \{(\boldsymbol{L}, \boldsymbol{S}) \mid \|\boldsymbol{L}\|_* \le \tau_L,\ \|\boldsymbol{S}\|_1 \le \tau_S\}$ *has diameter* $D \le 2\sqrt{\tau_L^2 + \tau_S^2}$.

Proof. For any $\boldsymbol{Z} = (\boldsymbol{L}, \boldsymbol{S})$ and $\boldsymbol{Z}' = (\boldsymbol{L}', \boldsymbol{S}') \in \mathcal{D}$,

$$\begin{aligned} \|\boldsymbol{Z} - \boldsymbol{Z}'\|_F^2 &= \|\boldsymbol{L} - \boldsymbol{L}'\|_F^2 + \|\boldsymbol{S} - \boldsymbol{S}'\|_F^2 \le (\|\boldsymbol{L}\|_F + \|\boldsymbol{L}'\|_F)^2 + (\|\boldsymbol{S}\|_F + \|\boldsymbol{S}'\|_F)^2 \\ &\le (\|\boldsymbol{L}\|_* + \|\boldsymbol{L}'\|_*)^2 + (\|\boldsymbol{S}\|_1 + \|\boldsymbol{S}'\|_1)^2 \le 4\tau_L^2 + 4\tau_S^2. \end{aligned} \tag{3.4}$$

- $||\boldsymbol{L}||_F = \sqrt{\mathrm{Tr}(\boldsymbol{L}^T\boldsymbol{L})} = \sqrt{\sum \sigma_i^2} \le \sum \sigma_i = ||\boldsymbol{L}||_*$

4.7 FW-P算法的收敛性

THEOREM 3.3. *Let $l^\star$ be the optimal value to problem (3.1), $\boldsymbol{x}^k = (\boldsymbol{L}^k, \boldsymbol{S}^k)$ and $\boldsymbol{v}^k = (\boldsymbol{V}_L^k, \boldsymbol{V}_S^k)$ be the sequence produced by Algorithm 4. Then we have*

$$l(\boldsymbol{L}^k, \boldsymbol{S}^k) - l^\star \leq \frac{16(\tau_L^2 + \tau_S^2)}{k+2}. \tag{3.11}$$

Moreover, for any $K \geq 1$, there exists $1 \leq \tilde{k} \leq K$ such that the surrogate duality gap (defined in (2.13)) satisfies

$$d(\boldsymbol{x}^{\tilde{k}}) = \left\langle \boldsymbol{x}^{\tilde{k}} - \boldsymbol{v}^{\tilde{k}}, \nabla l(\boldsymbol{x}^{\tilde{k}}) \right\rangle \leq \frac{48(\tau_L^2 + \tau_S^2)}{K+2}. \tag{3.12}$$

Proof. Substituting $L = 2$ (Lemma 3.1) and $D \leq 2\sqrt{\tau_L^2 + \tau_S^2}$ (Lemma 3.2) into Theorems 2.1 and 2.2, we can easily obtain the above result. □

4.8 带有惩罚项的CPCP问题

- 考虑如下问题:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \quad f(\boldsymbol{L},\boldsymbol{S}) \doteq \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\right\|_F^2 + \lambda_L\left\|\boldsymbol{L}\right\|_* + \lambda_S\left\|\boldsymbol{S}\right\|_1 \tag{4.1}$$

- 将上述问题转化为复合优化问题:

$$\begin{aligned} \min \quad & g(\boldsymbol{L},\boldsymbol{S},t_L,t_S) \doteq \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\right\|_F^2 + \lambda_L t_L + \lambda_S t_S \\ \text{s.t.} \quad & \left\|\boldsymbol{L}\right\|_* \le t_L,\ \left\|\boldsymbol{S}\right\|_1 \le t_S, \end{aligned} \tag{4.2}$$

- 注意到目标函数 $g(\boldsymbol{L},\boldsymbol{S},t_L,t_S)$ 是可微的:

$$\nabla_L g(\boldsymbol{L},\boldsymbol{S},t_L,t_S) = \nabla_S g(\boldsymbol{L},\boldsymbol{S},t_L,t_S) = \mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}] \tag{4.3}$$

$$\nabla_{t_L} g(\boldsymbol{L},\boldsymbol{S},t_L,t_S) = \lambda_L, \quad \nabla_{t_S} g(\boldsymbol{L},\boldsymbol{S},t_L,t_S) = \lambda_S. \tag{4.4}$$

梯度 $\nabla g(\boldsymbol{L},\boldsymbol{S},t_L,t_S) = (\nabla_L g, \nabla_S g, \nabla_{t_L} g, \nabla_{t_S} g)$

- 但是此时Frank-Wolfe方法仍然不能处理，因为可行域是无界。如果能得到t_L和t_S的最优值的上界即可利用FW方法。此时有:

$$(4.6)\qquad \begin{aligned} \min \quad & \frac{1}{2}\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\|_F^2+\lambda_L t_L+\lambda_S t_S \\ \text{s.t.} \quad & \|\boldsymbol{L}\|_* \le t_L \le U_L,\ \|\boldsymbol{S}\|_1 \le t_S \le U_S, \end{aligned}$$

- 注意到$\mathbf{L}=\mathbf{0},\mathbf{S}=\mathbf{0},t_L=0,t_S=0$时，目标函数值为$\frac{1}{2}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2$，可令:

$$(4.7)\qquad t_L^\star \ \le\ \frac{1}{2\lambda_L}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2, \qquad t_S^\star \ \le\ \frac{1}{2\lambda_S}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2$$

LEMMA 4.2. *The feasible set* $\mathcal{D}=\{(\boldsymbol{L},\boldsymbol{S},t_L,t_S)\mid \|\boldsymbol{L}\|_*\le t_L\le U_L,\ \|\boldsymbol{S}\|_1\le t_S\le U_S\}$ *has diameter* $D\le\sqrt{5}\cdot\sqrt{U_L^2+U_S^2}$.

Proof. Since for any $\boldsymbol{Z}=(\boldsymbol{L},\boldsymbol{S},t_L,t_S)$, $\boldsymbol{Z}'=(\boldsymbol{L}',\boldsymbol{S}',t_L',t_S')\in\mathcal{D}$, we have

$$\begin{aligned} \|\boldsymbol{Z}-\boldsymbol{Z}'\|_F^2 &= \|\boldsymbol{L}-\boldsymbol{L}'\|_F^2+\|\boldsymbol{S}-\boldsymbol{S}'\|_F^2+(t_L-t_L')^2+(t_S-t_S')^2 \\ &\le (\|\boldsymbol{L}\|_F+\|\boldsymbol{L}'\|_F)^2+(\|\boldsymbol{S}\|_F+\|\boldsymbol{S}'\|_F)^2+(t_L-t_L')^2+(t_S-t_S')^2 \\ &\le (\|\boldsymbol{L}\|_*+\|\boldsymbol{L}'\|_*)^2+(\|\boldsymbol{S}\|_1+\|\boldsymbol{S}'\|_1)^2+(t_L-t_L')^2+(t_S-t_S')^2 \\ &\le (U_L+U_L)^2+(U_S+U_S)^2+U_L^2+U_S^2 \\ &= 5(U_L^2+U_S^2) \end{aligned}$$

4.9 问题(4.6)的FW算法

- 第k步，迭代点 $\boldsymbol{x}^k = (\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)$ 由FW算法产生
  令 $\boldsymbol{v}^k = (\boldsymbol{V}_L^k, \boldsymbol{V}_S^k, V_{t_L}^k, V_{t_S}^k)$ 是求解下述问题(FW)得到的:

$$(4.9) \qquad \boldsymbol{v}^k \in \arg\min_{\boldsymbol{v}\in\mathcal{D}} \quad \left\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}], \boldsymbol{V}_L + \boldsymbol{V}_S \right\rangle + \lambda_L V_{t_L} + \lambda_S V_{t_S}$$

- 上述问题可以解耦成两个独立的子问题:

$$(4.10) \quad (\boldsymbol{V}_L^k, V_{t_L}^k) \in \arg\min_{\|\boldsymbol{V}_L\|_* \le V_{t_L} \le U_L} g_L(\boldsymbol{V}_L, V_{t_L}) \doteq \left\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}], \boldsymbol{V}_L \right\rangle + \lambda_L V_{t_L}$$

$$(4.11) \quad (\boldsymbol{V}_S^k, V_{t_S}^k) \in \arg\min_{\|\boldsymbol{V}_S\|_1 \le V_{t_S} \le U_S} g_S(\boldsymbol{V}_S, V_{t_S}) \doteq \left\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}], \boldsymbol{V}_S \right\rangle + \lambda_S V_{t_S}$$

- 先考虑第一个子问题，令:

$$(4.12) \qquad \boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \quad \hat{g}_L(\boldsymbol{D}_L) \doteq \left\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}], \boldsymbol{D}_L \right\rangle + \lambda_L$$

- 利用[24,引理1]:

$$(4.13)\qquad (\boldsymbol{V}_L^k, V_{t_L}^k) \in \begin{cases} \{(\mathbf{0}, 0)\} & \text{if } \hat{g}_L(\boldsymbol{D}_L^k) > 0 \\ \text{conv}\{(\mathbf{0}, 0), U_L(\boldsymbol{D}_L^k, 1)\} & \text{if } \hat{g}_L(\boldsymbol{D}_L^k) = 0 \\ \{U_L(\boldsymbol{D}_L^k, 1)\} & \text{if } \hat{g}_L(\boldsymbol{D}_L^k) < 0 \end{cases}$$

COROLLARY 4.3. *Let* $\boldsymbol{x}^\star = (\boldsymbol{L}^\star, \boldsymbol{S}^\star, t_L^\star, t_S^\star)$ *be an optimal solution to* (4.6). *For* $\{\boldsymbol{x}^k\}$ *generated by Algorithm* 5, *we have for* $k = 0,\ 1,\ 2,\ \ldots,$

$$(4.14)\qquad g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star) \leq \frac{20(U_L^2 + U_S^2)}{k+2}.$$

[24] Z. Harchaoui, A. Juditsky, and A. Nemirovski, “Conditional gradient algorithms for norm-regularized smooth convex optimization,” Mathematical Programming, pp. 1–38, 2014

Proof. Applying Theorem 2.1 with parameters calculated in Lemmas 4.1 and 4.2 we directly have

$$g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star) \le \frac{2 \cdot 2 \cdot \left(\sqrt{5(U_L^2 + U_S^2)}\right)^2}{k+2} = \frac{20(U_L^2 + U_S^2)}{k+2}. \tag{4.15}$$

A more careful calculation below slightly improves the constant in (4.15).

$$\begin{aligned} g(\boldsymbol{x}^{k+1}) &= g(\boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k)) \\ &\le g(\boldsymbol{x}^k) + \gamma \left\langle \nabla g(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + \gamma^2 \left\| \boldsymbol{V}_L^k - \boldsymbol{L}^k \right\|_F^2 + \gamma^2 \left\| \boldsymbol{V}_S^k - \boldsymbol{S}^k \right\|_F^2 \\ &\le g(\boldsymbol{x}^k) + \gamma \left\langle \nabla g(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + 4\gamma^2 (U_L^2 + U_S^2), \end{aligned} \tag{4.16}$$

where the second line holds by noting that g is only linear in t_L and t_S; the last line holds as

$$\begin{aligned} \left\| \boldsymbol{V}_L^k - \boldsymbol{L}^k \right\|_F^2 &\le (\left\| \boldsymbol{V}_L^k \right\|_F + \left\| \boldsymbol{L}^k \right\|_F)^2 \le (U_L + U_L)^2 = 4U_L^2, \quad \text{and} \\ \left\| \boldsymbol{V}_S^k - \boldsymbol{S}^k \right\|_F^2 &\le (\left\| \boldsymbol{V}_S^k \right\|_F + \left\| \boldsymbol{S}^k \right\|_F)^2 \le (U_S + U_S)^2 = 4U_S^2. \end{aligned}$$

Following the arguments in the proof of Theorem 1 with (2.10) replaced by (4.16), we can easily obtain that

$$g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star) \le \frac{16(U_L^2 + U_S^2)}{k+2}.$$

Algorithm 5 Frank-Wolfe method for problem (4.6)

Initialization: $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \mathbf{0}$; $t_L^0 = t_S^0 = 0$;
for $k = 0,\ 1,\ 2,\ \ldots$ **do**
  $\boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L \rangle$;
  $\boldsymbol{D}_S^k \in \arg\min_{\|\boldsymbol{D}_S\|_1 \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S \rangle$;
  if $\lambda_L \ge -\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L^k \rangle$ **then**
    $\boldsymbol{V}_L^k = \mathbf{0}$; $V_{t_L}^k = 0$
  else
    $\boldsymbol{V}_L^k = U_L \boldsymbol{D}_L^k$, $V_{t_L}^k = U_L$;
  end if
  if $\lambda_S \ge -\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S^k \rangle$ **then**
    $\boldsymbol{V}_S^k = \mathbf{0}$; $V_{t_S}^k = 0$;
  else
    $\boldsymbol{V}_S^k = U_S \boldsymbol{D}_S^k$, $V_{t_S}^k = U_S$;
  end if
  $\gamma = \frac{2}{k+2}$;
  $\boldsymbol{L}^{k+1} = (1-\gamma)\boldsymbol{L}^k + \gamma \boldsymbol{V}_L^k$, $t_L^{k+1} = (1-\gamma)t_L^k + \gamma V_{t_L}^k$;
  $\boldsymbol{S}^{k+1} = (1-\gamma)\boldsymbol{S}^k + \gamma \boldsymbol{V}_S^k$, $t_S^{k+1} = (1-\gamma)t_S^k + \gamma V_{t_S}^k$;
end for

算法5的不足： 首先在范数约束的情况下，**S** 只有一个非零项，所以 $\mathbf{V}_S^k$在每次迭代中只有一个稀疏更新。其次，精确的收敛速度依赖于我们对U_L和U_S的估计。

4.10 问题(4.6)的FW-T算法:结合FW和临近梯度法

- 主要思想:利用FW算法更新$\mathbf{L}$，利用临近梯度法更新 $\mathbf{S}$，精确线搜索，自适应更新U_L和U_S

Proximal gradient step for $\boldsymbol{S}$. To update $\boldsymbol{S}$ in a more efficient way, we incorporate an additional proximal gradient step for $\boldsymbol{S}$. At iteration k, let $(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}})$ be the result produced by Frank-Wolfe step. To produce the next iterate, we retain the low-rank term $\boldsymbol{L}^{k+\frac{1}{2}}$, but execute a proximal gradient step for the function $f(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S})$ at the point $\boldsymbol{S}^{k+\frac{1}{2}}$, i.e.

$$\boldsymbol{S}^{k+1} \in \arg\min_{\boldsymbol{S}} \left\langle \nabla_{\boldsymbol{S}} f(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}),\ \boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}} \right\rangle + \frac{1}{2} \left\| \boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}} \right\|_F^2 + \lambda_S \|\boldsymbol{S}\|_1$$

(4.17)

$$= \arg\min_{\boldsymbol{S}} \left\langle \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}], \boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}} \right\rangle + \frac{1}{2} \left\| \boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}} \right\|_F^2 + \lambda_S \|\boldsymbol{S}\|_1$$

which can be easily computed using the soft-thresholding operator:

$$\boldsymbol{S}^{k+1} = \mathcal{T}_{\lambda_S} \left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}] \right]. \quad (4.18)$$

Exact line search. For the Frank-Wolfe step, instead of choosing the fixed step length $\frac{2}{k+2}$, we implement an exact line search by solving a two-dimensional quadratic problem (4.20), as in [24]. This modification turns out to be crucial to achieve a primal convergence result that only weakly depends on the tightness of our guesses U_L and U_S.

Adaptive updates of U_L ***and*** U_S. We initialize U_L and U_S using the crude bound (4.8). Then, at the end of the k-iteration, we respectively update

(4.19)
$$U_L^{k+1} = g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1})/\lambda_L, \quad U_S^{k+1} = g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1})/\lambda_S.$$

This scheme maintains the property that $U_L^{k+1} \geq t_L^\star$ and $U_S^{k+1} \geq t_S^\star$. Moreover, we prove (Lemma 4.4) that g is non-increasing through our algorithm, and so this scheme produces a sequence of tighter upper bounds for $U_L^\star$ and $U_S^\star$. Although this dynamic scheme does not improve the theoretical convergence result, some acceleration is empirically exhibited.

Algorithm 6 FW-T method for problem (4.1)

1: **Input:** data matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$; weights λ_L, $\lambda_S > 0$; max iteration number T;
2: **Initialization:** $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$; $t_L^0 = t_S^0 = 0$; $U_L^0 = g(\boldsymbol{L}^0, \boldsymbol{S}^0, t_L^0, t_S^0)/\lambda_L$; $U_S^0 = g(\boldsymbol{L}^0, \boldsymbol{S}^0, t_L^0, t_S^0)/\lambda_S$;
3: **for** $k = 0,\ 1,\ 2,\ \cdots,\ T$ **do**
4: *same as lines 3-14 in Algorithm 5;*
5: $\left(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}, t_L^{k+\frac{1}{2}}, t_S^{K+\frac{1}{2}}\right)$ is computed as an optimizer to

$$
\begin{aligned}
(4.20) \qquad \min \quad & \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\right\|_F^2 + \lambda_L t_L + \lambda_S t_S \\
\text{s.t.} \quad & \begin{pmatrix} \boldsymbol{L} \\ t_L \end{pmatrix} \in \operatorname{conv}\left\{ \begin{pmatrix} \boldsymbol{L}^k \\ t_L^k \end{pmatrix}, \begin{pmatrix} \boldsymbol{V}_L^k \\ V_{t_L}^k \end{pmatrix} \right\} \\
& \begin{pmatrix} \boldsymbol{S} \\ t_S \end{pmatrix} \in \operatorname{conv}\left\{ \begin{pmatrix} \boldsymbol{S}^k \\ t_S^k \end{pmatrix}, \begin{pmatrix} \boldsymbol{V}_S^k \\ V_{t_S}^k \end{pmatrix} \right\};
\end{aligned}
$$

6: $\boldsymbol{S}^{k+1} = \mathcal{T}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}], \lambda_S\right]$;
7: $\boldsymbol{L}^{k+1} = \boldsymbol{L}^{k+\frac{1}{2}}$, $t_L^{k+1} = t_L^{k+\frac{1}{2}}$; $t_S^{k+1} = \left\|\boldsymbol{S}^{k+1}\right\|_1$;
8: $U_L^{k+1} = g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1})/\lambda_L$;
9: $U_S^{k+1} = g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1})/\lambda_S$;
10: **end for**

4.11 FW-T算法的收敛性

THEOREM 4.5. *Let $r_L^\star$ and $r_S^\star$ be the smallest radii such that*

$$\left\{(\boldsymbol{L},\boldsymbol{S}) \,\middle|\, f(\boldsymbol{L},\boldsymbol{S}) \le g(\boldsymbol{L}^0,\boldsymbol{S}^0,t_L^0,t_S^0) = \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{M}]\right\|_F^2\right\} \subseteq \overline{B(r_L^\star)} \times \overline{B(r_S^\star)}, \tag{4.24}$$

where $\overline{B(r)} \doteq \{\boldsymbol{X} \in \mathbb{R}^{m\times n} | \left\|\boldsymbol{X}\right\|_F \le r\}$ for any $r \ge 0$.‡ Then for the sequence $\{(\boldsymbol{L}^k,\boldsymbol{S}^k,t_L^k,t_S^k)\}$ generated by Algorithm 6, we have

$$g(\boldsymbol{L}^k,\boldsymbol{S}^k,t_L^k,t_S^k) - g(\boldsymbol{L}^\star,\boldsymbol{S}^\star,t_L^\star,t_S^\star) \le \frac{\min\{4(t_L^\star+r_L^\star)^2+4(t_S^\star+r_S^\star)^2,\ 16(U_L^0)^2+16(U_S^0)^2\}}{k+2}. \tag{4.25}$$

THEOREM 4.6. *Let* $\boldsymbol{x}^k$ *denote* $(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)$ *generated by Algorithm 6. Then for any* $K \geq 1$, *there exists* $1 \leq \tilde{k} \leq K$ *such that*

$$g(\boldsymbol{x}^{\tilde{k}}) - g(\boldsymbol{x}^\star) \leq d(\boldsymbol{x}^{\tilde{k}}) \leq \frac{48\left((U_L^0)^2 + (U_S^0)^2\right)}{K+2}. \tag{4.29}$$

Proof. Define $\Delta^k = g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star)$. Following (4.16), we have

$$\Delta^{k+1} \leq \Delta^k + \gamma \left\langle \nabla g(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + 4\gamma^2 \left((U_L^0)^2 + (U_S^0)^2\right). \tag{4.30}$$

Then following the arguments in the proof of Theorem 2 with (2.17) replaced by (4.30), we can easily obtain the result. □

Stopping criterion. Compared to the convergence of $g(\boldsymbol{x}^k)$ (Theorem 4.5), the convergence result for $d(\boldsymbol{x}^k)$ can be much slower (Theorem 4.6). Therefore, here the surrogate duality gap $d(\cdot)$ is not that suitable to serve as a stopping criterion. Consequently, in our implementation, we terminate Algorithm 6 if

$$\left|g(\boldsymbol{x}^{k+1}) - g(\boldsymbol{x}^k)\right| / g(\boldsymbol{x}^k) \leq \varepsilon, \tag{4.31}$$

for five consecutive iterations.

5. 数值实验

5.1 ISTA算法与FISTA算法

- ISTA算法在第k步通过求解下式来更新 $(\boldsymbol{L}, \boldsymbol{S})$:

$$(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}) = \arg\min_{\boldsymbol{L},\boldsymbol{S}} \left\langle \begin{pmatrix} \nabla_{\boldsymbol{L}} l(\boldsymbol{L}^k, \boldsymbol{S}^k) \\ \nabla_{\boldsymbol{S}} l(\boldsymbol{L}^k, \boldsymbol{S}^k) \end{pmatrix}, \begin{pmatrix} \boldsymbol{L} - \boldsymbol{L}^k \\ \boldsymbol{S} - \boldsymbol{S}^k \end{pmatrix} \right\rangle + \frac{L_f}{2} \left\| \begin{pmatrix} \boldsymbol{L} \\ \boldsymbol{S} \end{pmatrix} - \begin{pmatrix} \boldsymbol{L}^k \\ \boldsymbol{S}^k \end{pmatrix} \right\|_F^2 + \lambda_L \|\boldsymbol{L}\|_* + \lambda_S \|\boldsymbol{S}\|_1 \tag{5.1}$$

Here $L_f = 2$ denotes the Lipschitz constant of $\nabla l(\boldsymbol{L}, \boldsymbol{S})$ with respect to $(\boldsymbol{L}, \boldsymbol{S})$, and $\nabla_{\boldsymbol{L}} l(\boldsymbol{L}^k, \boldsymbol{S}^k) = \nabla_{\boldsymbol{S}} l(\boldsymbol{L}^k, \boldsymbol{S}^k) = \mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]$. Since $\boldsymbol{L}$ and $\boldsymbol{S}$ are decoupled in (5.1), equivalently we have

$$\boldsymbol{L}^{k+1} = \arg\min_{\boldsymbol{L}} \left\| \boldsymbol{L} - \left(\boldsymbol{L}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \right) \right\|_F^2 + \lambda_L \|\boldsymbol{L}\|_* , \tag{5.2}$$

$$\boldsymbol{S}^{k+1} = \arg\min_{\boldsymbol{S}} \left\| \boldsymbol{S} - \left(\boldsymbol{S}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \right) \right\|_F^2 + \lambda_S \|\boldsymbol{S}\|_1 . \tag{5.3}$$

The solution to problem (5.3) can be given explicitly in terms of the proximal mapping of $\|\cdot\|_1$ as introduced in Section 2.2, i.e.,

$$\boldsymbol{S}^{k+1} = \mathcal{T}_{\lambda_S/2}\left[\boldsymbol{S}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \right].$$

For a matrix $\boldsymbol{X}$ and any $\tau \geq 0$, let $\mathcal{D}_\tau(\boldsymbol{X})$ denote the singular value thresholding operator $\mathcal{D}_\tau(\boldsymbol{X}) = \boldsymbol{U}\mathcal{T}_\tau(\boldsymbol{\Sigma})\boldsymbol{V}^\top$, where $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the singular value decomposition of $\boldsymbol{X}$. It is not difficult to show [35, 36] that the solution to problem (5.2) can be given explicitly by

$$\boldsymbol{L}^{k+1} = \mathcal{D}_{\lambda_L/2}\left[\boldsymbol{L}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \right].$$

Algorithm 7 ISTA for problem (1.5)

Initialization: $\boldsymbol{L}^0 = \boldsymbol{0}$, $\boldsymbol{S}^0 = \boldsymbol{0}$;
for $k = 0,\ 1,\ 2,\ \cdots$ **do**
  $\boldsymbol{L}^{k+1} = \mathcal{D}_{\lambda_L/2}\left[\boldsymbol{L}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]\right]$;
  $\boldsymbol{S}^{k+1} = \mathcal{T}_{\lambda_S/2}\left[\boldsymbol{S}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]\right]$;
end for

Algorithm 8 FISTA for problem (1.5)

Initialization: $\hat{\boldsymbol{L}}^0 = \boldsymbol{L}^0 = \boldsymbol{0}$, $\hat{\boldsymbol{S}}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$, $t_0 = 1$;
for $k = 0,\ 1,\ 2,\ \cdots$ **do**
  $\boldsymbol{L}^{k+1} = \mathcal{D}_{\lambda_L/2}\left[\hat{\boldsymbol{L}}^k - \frac{1}{2}\mathcal{P}_\Omega[\hat{\boldsymbol{L}}^k + \hat{\boldsymbol{S}}^k - \boldsymbol{M}]\right]$;
  $\boldsymbol{S}^{k+1} = \mathcal{T}_{\lambda_S/2}\left[\hat{\boldsymbol{S}}^k - \frac{1}{2}\mathcal{P}_\Omega[\hat{\boldsymbol{L}}^k + \hat{\boldsymbol{S}}^k - \boldsymbol{M}]\right]$;
  $t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}$;
  $\hat{\boldsymbol{L}}^{k+1} = \boldsymbol{L}^{k+1} + \frac{t^k-1}{t^{k+1}}(\boldsymbol{L}^{k+1} - \boldsymbol{L}^k)$;
  $\hat{\boldsymbol{S}}^{k+1} = \boldsymbol{S}^{k+1} + \frac{t^k-1}{t^{k+1}}(\boldsymbol{S}^{k+1} - \boldsymbol{S}^k)$;
end for

5.2 ISTA算法与FISTA算法

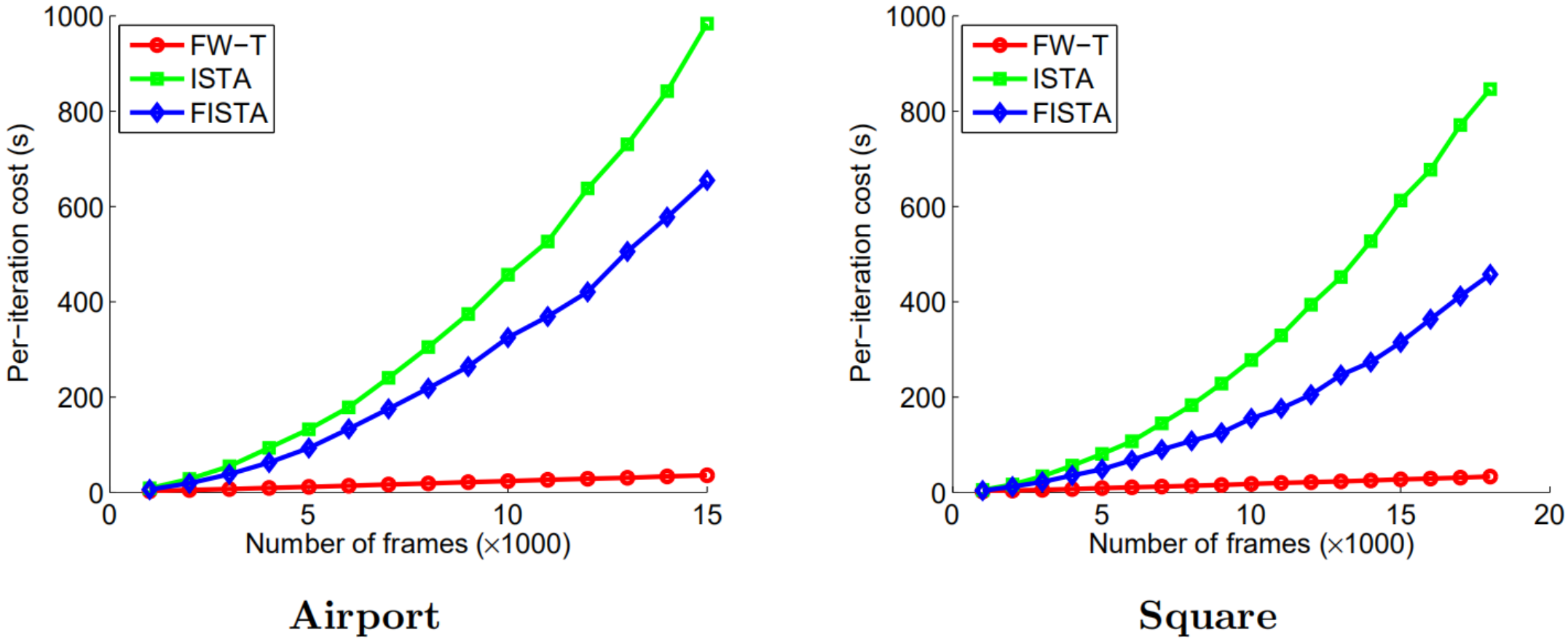


FIG. 2. ***Per-iteration cost vs. the number of frames in Airport and Square videos with full observation.*** *The per-iteration cost of our FW-T method grows linearly with the size of data, in contrast with the superlinear per-iteration cost of ISTA and FISTA. That makes the FW-T method more advantageous or may even be the only feasible choice for large problems.*

TABLE 1

Comparisons of FW-T, ISTA and FISTA on surveillance video data. *The advantage of our FW-T method becomes prominent when the data are at large scale and compressed (i.e. the small ρ scenarios).*

		FW-T		ISTA		FISTA	
Data	ρ	iter.	time	iter.	time	iter.	time
Lobby	1.0	96	1.94e+02	144	3.64e+02	41	**1.60e+02**
(20480 × 1000)	0.8	104	**2.33e+02**	216	1.03e+03	52	3.55e+02
	0.6	133	**3.12e+02**	380	1.67e+03	74	5.10e+02
Campus	1.0	45	**1.56e+02**	78	1.49e+03	23	4.63e+02
(20480 × 1439)	0.8	44	**1.57e+02**	122	2.34e+03	30	6.45e+02
	0.6	41	**1.39e+02**	218	4.27e+03	43	1.08e+03
Escalator	1.0	81	**7.40e+02**	58	4.19e+03	25	2.18e+03
(20800 × 3417)	0.8	80	**7.35e+02**	90	8.18e+03	32	3.46e+03
	0.6	82	**7.68e+02**	162	1.83e+04	43	5.73e+03
Mall	1.0	38	**4.70e+02**	110	5.03e+03	35	1.73e+03
(81920 × 1286)	0.8	35	**4.58e+02**	171	7.32e+03	44	2.34e+03
	0.6	44	**5.09e+02**	308	1.31e+04	62	3.42e+03
Restaurant	1.0	70	**5.44e+02**	52	3.01e+03	20	1.63e+03
(19200 × 3055)	0.8	74	**5.51e+02**	81	4.84e+03	26	1.82e+03
	0.6	76	**5.73e+02**	144	9.93e+03	38	3.31e+03

Hall	1.0	60	**6.33e+02**	52	2.98e+03	21	1.39e+03
(25344 × 3584)	0.8	62	**6.52e+02**	81	6.45e+03	28	2.90e+03
	0.6	70	**7.43e+02**	144	1.42e+04	39	4.94e+03
Airport	1.0	130	**6.42e+03**	29	2.37e+04	14	1.37e+04
(25344 × 15730)	0.8	136	**6.65e+03**	45	6.92e+04	18	4.27e+04
	0.6	154	**7.72e+03**	77	1.78e+05	24	7.32e+04
Square	1.0	179	**1.24e+04**	29	3.15e+04	13	1.51e+04
(19200 × 28181)	0.8	181	**1.26e+04**	44	1.04e+05	17	6.03e+04
	0.6	191	**1.31e+04**	78	2.63e+05	22	9.88e+05

谢谢！