IALM

求解方法

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0730ac36-9699-487f-a06a-f8b7a44fac89/S_1_2___.pdf

增广拉格朗日乘子法

Robust PCA 的模型②. $\begin{cases} X=(A,E). & h(X)=D-A-E \\ f(X)=\|A\|_* + \lambda\|E\|_1 \end{cases}$

则 增广拉格朗日函数 为

$$L(A,E,Y,\mu)=f(X)+\langle Y,h(X)\rangle+\frac{\mu}{2}\|Y-A-E\|_F^2$$

$\langle\cdot\rangle$ 标准内积

μ: 惩罚因子.

$\|\cdot\|_F$: Frobenius 范数

最小化采用交替投影更新的思想,

精准 EALM

①. 先固定 A和Y 求一个使 L 最小化的 E

②. 固定 E 和 Y, 求一个使 L最小化的 A.

若 $E=E^j_{k+1}$, 则 $A^{j+1}_{k+1}=\arg\min\limits_A L(A,E^j_{k+1},Y_k,\mu_k)$

$$=\arg\min_A \|A\|_* + \mu_k\|A-(D-E^j_{k+1}+Y_k/\mu_k)\|/2$$

$$=D_{1/\mu_k}(D-E^j_{k+1}+Y_k/\mu_k)$$

更新E时

$$E^{j+1}_{k+1}=\arg\min_E L(A^{j+1}_{k+1},E,Y_k,\mu_k)=\arg\min_E \lambda\|E\|_1 + \mu_k\|E-(D-A^{j+1}_{k+1}+Y_k/\mu_k)\|^2/2$$

$$=S_{\lambda/\mu_k}(D-A^{j+1}_{k+1}+Y_k/\mu_k)$$

$A^{j+1}_{k+1}\to A^*_{k+1}$, $E^{j+1}_{k+1}\longrightarrow E^*_{k+1}$

则最后 Y: $Y_{k+1}=Y_k+\mu_k(D-A^*_{k+1}-E^*_{k+1})$

更新 μ: $\mu_{k+1}=\begin{cases}\rho\mu_k & 若\ \mu_k\|E^*_{k+1}-E^*_k\|_F/\|D\|_F<\varepsilon \\ \mu_k & 否则\end{cases}$

ρ为大于1的常数, ε为较小的正数.

实际求解中只需要更新A或E各一次得到子问题的一个近似解即可，即非精确增广拉格朗日乘子法ADM。

$A_{k+1} = \arg\min_A L\left(A, E_{k+1}, Y_k, \mu_k\right), E_{k+1} = \arg\min_E L\left(A_{k+1}, E, Y_k, \mu_k\right)$

更多详细的算法内容参见下方论文的Algorithm 4、5

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/3ab0728a-ceb9-4553-854e-1b0ec4206a03/1009.5055.pdf

Algorithm 4 (RPCA via the Exact ALM Method)

Input: Observation matrix $D \in \mathbb{R}^{m \times n}$, λ.
1: $Y_0^* = \operatorname{sgn}(D)/J(\operatorname{sgn}(D))$; $\mu_0 > 0$; $\rho > 1$; $k = 0$.
2: **while** not converged **do**
3: // Lines 4-12 solve $(A_{k+1}^*, E_{k+1}^*) = \arg\min_{A,E} L(A, E, Y_k^*, \mu_k)$.
4: $A_{k+1}^0 = A_k^*$, $E_{k+1}^0 = E_k^*$, $j = 0$;
5: **while** not converged **do**
6: // Lines 7-8 solve $A_{k+1}^{j+1} = \arg\min_{A} L(A, E_{k+1}^j, Y_k^*, \mu_k)$.
7: $(U, S, V) = \operatorname{svd}(D - E_{k+1}^j + \mu_k^{-1} Y_k^*)$;
8: $A_{k+1}^{j+1} = U \mathcal{S}_{\mu_k^{-1}}[S] V^T$;
9: // Line 10 solves $E_{k+1}^{j+1} = \arg\min_{E} L(A_{k+1}^{j+1}, E, Y_k^*, \mu_k)$.
10: $E_{k+1}^{j+1} = \mathcal{S}_{\lambda\mu_k^{-1}}[D - A_{k+1}^{j+1} + \mu_k^{-1} Y_k^*]$;
11: $j \leftarrow j + 1$.
12: **end while**
13: $Y_{k+1}^* = Y_k^* + \mu_k(D - A_{k+1}^* - E_{k+1}^*)$.
14: Update μ_k to μ_{k+1}.
15: $k \leftarrow k + 1$.
16: **end while**
Output: (A_k^*, E_k^*).

Algorithm 5 (RPCA via the Inexact ALM Method)

Input: Observation matrix $D \in \mathbb{R}^{m \times n}$, λ.
1: $Y_0 = D/J(D)$; $E_0 = 0$; $\mu_0 > 0$; $\rho > 1$; $k = 0$.
2: **while** not converged **do**
3: // Lines 4-5 solve $A_{k+1} = \arg\min_{A} L(A, E_k, Y_k, \mu_k)$.
4: $(U, S, V) = \operatorname{svd}(D - E_k + \mu_k^{-1} Y_k)$;
5: $A_{k+1} = U \mathcal{S}_{\mu_k^{-1}}[S] V^T$.
6: // Line 7 solves $E_{k+1} = \arg\min_{E} L(A_{k+1}, E, Y_k, \mu_k)$.
7: $E_{k+1} = \mathcal{S}_{\lambda\mu_k^{-1}}[D - A_{k+1} + \mu_k^{-1} Y_k]$.
8: $Y_{k+1} = Y_k + \mu_k(D - A_{k+1} - E_{k+1})$.
9: Update μ_k to μ_{k+1}.
10: $k \leftarrow k + 1$.
11: **end while**
Output: (A_k, E_k).

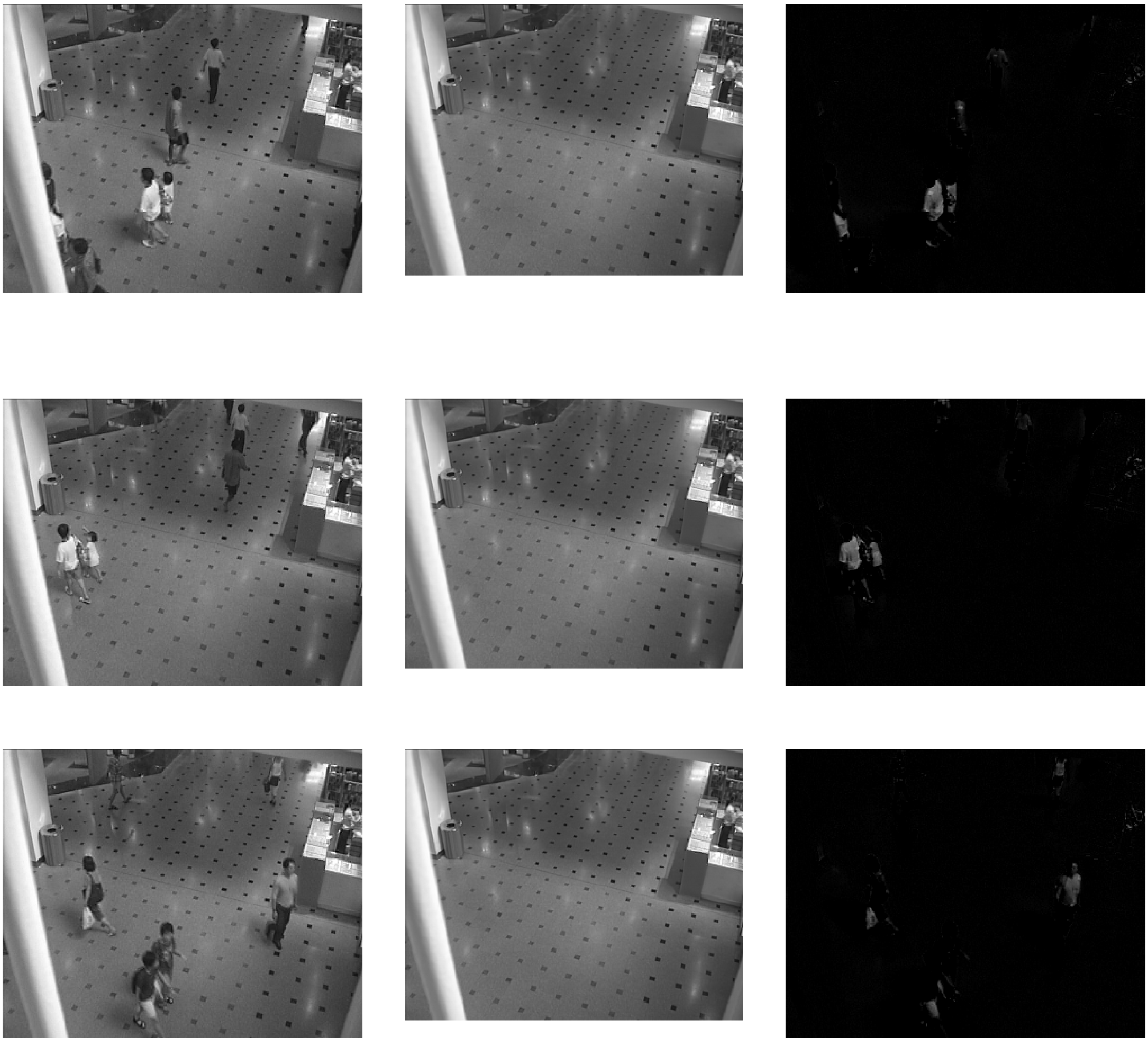
其中的$\langle A, B\rangle = \operatorname{tr}\left(A^T B\right), \quad J(Y) = \max\left(\|Y\|_2, \lambda^{-1}\|Y\|_\infty\right)$

$$\mathcal{S}_\varepsilon[x] \doteq \begin{cases} x - \varepsilon, \text{ if } x > \varepsilon \\ x + \varepsilon, \text{ if } x < -\varepsilon \\ 0, \quad \text{otherwise} \end{cases}$$

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

增广拉格朗日算法中每一轮迭代最主要计算量在于奇异值分解，寻找一种高效的部分奇异值分解方法对于视频背景建模这种较大规模的问题是非常重要的。

根据算法5得到的视频结果：部分帧（从左到右依次是原始帧、背景、前景）



我的代码地址：https://github.com/yxhuang7538/Video_background_modeling/tree/main/IALM