SPROJ Notes

MC (Vanilla Problem)

Input: $X_\Omega \in \mathbb{R}^{n_1 \times n_2}$

Ω: set of (i,j) indices where $X_{ij} = 0$

i.e. $X(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases}$

Objective: $\min\limits_{M \in \mathbb{R}^{n_1 \times n_2}} \operatorname{rank}(M)$ ①

user-specified threshold

$$\text{s.t } \|M_\Omega - X_\Omega\|_F^2 < \delta \text{ or } M_\Omega = X_\Omega$$

- Above objective is NP-hard as all algorithms are doubly exponential in dimension $\max(n_1, n_2)$

- Approximate ① by $\|\cdot\|_*$ (nucleus norm) relaxation. where $\|X\|_* = \sum_i \sigma_i(X)$; $\sigma_i = i^{th}$ singular value
- Since it's a norm (proof skipped) it's convex.

$\therefore$ New objective: $\min\limits_{M} \|M\|_*$ s.t $M_\Omega = X_\Omega$ ②

- Form Augmented Lagrangian:

$$~~L_H(M)~~ \; L_\mu(M, \Lambda) = \|M\|_* + \langle \Lambda, M_\Omega - X_\Omega \rangle + \frac{\mu}{2} \|M_\Omega - X_\Omega\|_F^2 \quad ③$$

- "Alternating" minimization strategy for ③ seeking a saddle point w.r.t to minimization of primal variable M and maximization w.r.t dual variable Λ.
- Steps:

$$M_{k+1} \in \arg\min_M L_\mu(M, \Lambda)$$
$$\Lambda_{k+1} = \Lambda_k + \mu [M_{k+1} - X]_\Omega$$

- let $g(M) = \|M\|_*$

Note the iterative step for M:

$$\min_M \underbrace{\|M\|_*}_{\substack{g(M) \\ \text{convex, non-smooth}}} + \underbrace{\langle \Lambda, X_\Omega - M_\Omega \rangle + \frac{\mu}{2}\|X_\Omega - M_\Omega\|_F^2}_{f(M)\text{: convex, smooth}}$$

$$\nabla f(M) = -\Lambda_\Omega + \mu[M - X]_\Omega$$ (μ-lipschitz proof skipped)

Note: problems when objective function is of the form:

$F(M) = g(M) + f(M)$ when $g(M)$ is convex (not neccessarily smooth) and $f(M)$ is convex + smooth, we can apply "proximal gradient method"

- Proof skipped but PG gives the following iteration scheme:

$$M_{k+1} = \underset{M}{\operatorname{argmin}} \left\{ g(M) + \frac{\mu}{2} \left\|M - \left(M_k - \frac{1}{\mu}\nabla f(M_k)\right)\right\|_F^2 \right.$$

let $Z_k = M_k - \frac{1}{\mu}\nabla f(M_k)$

we require to solve the sequence of "proximal problems":

$$\min_M \left\{ g(M) + \frac{\mu}{2}\|M - Z_k\|_F^2 \right\} \quad (4)$$

- Proof skipped but when $g(M)$ is nuclear norm, this (4) can be solved in "closed-form" from the SVD of Z_k. More specifically "SVT" defined as:

$$\mathcal{D}_\tau[M] = U S_\tau[\Sigma] V^T \; ; \; S_\tau[x] = \operatorname{sign}(x) \circ (|x| - \tau)_+$$

$(\cdot)_+ = \max(0, \cdot)$

$\therefore$ final unique solution to (4) is given by $M^* = \mathcal{D}_{\mu^{-1}}[Z_k]$.

ConvMC-Net:

we have following from ALM algorith for MC:

$$L^{K+1} = \Psi_{\mu^{-1}}\{L_{\Omega^c} + D_\Omega + \mu^{-1} Y_\Omega\}.$$

- Introduce measurement matrix H as follows:

$$\Rightarrow \min_L \frac{1}{\mu}\|L\|_* + \frac{1}{2}\|D_\Omega - (HL)_\Omega + \frac{Y_\Omega}{\mu}\|_F^2$$

- Replace $(HL)_\Omega$ with GL_Ω on the assumption such a matrix G exists s.t $\|GL_\Omega - (HL)_\Omega\| \le \delta/\epsilon$

- simplification:

$$L^{K+1} = \Psi_{\mu^{-1}}\{L^K + (G^T)D_\Omega + (-G^TG)L_\Omega^K + \frac{G^T}{\mu^K} Y_\Omega^K\}$$

- Replace G with convolutional kernels and use the subsitution: $G^T Y_\Omega = W \circ Y_\Omega + B$

$$\therefore L^{K+1} = \Psi_{\mu^{K^{-1}}}\{L^K + (C_1^K * D_\Omega) + (G_2^K * L_\Omega^K) + (W^K \circ Y_\Omega^K + B^K)\}$$

Some general observations:

- APG, TNNR, ~~FPC~~ WNNR etc are all SVT algorithms in requiring for solving MC. hence computationally demanding.

- To avoid SVD computation, matrix factorization (MF) is suggested.

- ~~Current~~ Most MF algorithms can work well when the observed matrix is noise-free or with white gaussian noise corruption.

- However, many practical scenarios contain impulsive Gaussian noise / salt-and-pepper noise.

- To resist / robustness against outliers in CMM case, l_p-norm with $0 < p < 2$ has been adopted.

- Lesser the p, the more robust.
- $0 < p < 1$ (non-convex, non-smooth - challenge)
- $1 < p < 2$ (convex but requires IRWLS)
- $p = 2$ (reduces to LS solution).

Convolution MC: $U \in \mathbb{R}^{n_1 \times r}$, $V \in \mathbb{R}^{r \times n_2}$

HubregV: $\beta \in \mathbb{R}^{r \times 1}$, $X \in \mathbb{R}^{(j,t,r)}$; $Y \in \mathbb{R}^{(j,t,1)}$

HubregU: $\beta \in \mathbb{R}^{1 \times r}$, $X \in \mathbb{R}^{(r,i,j)}$, $Y \in \mathbb{R}^{(1,i,j)}$