Laborator 5: Supravegherea video.
Metoda multiplicatorilor cu direcție alternantă (ADMM)

In acest laborator rezolvam problema analizei componentelor principale (PCA) robusta utilizand ADMM. Aplicatia reprezinta o etapa din supravegherea video, mai exact identificarea prim-planului (*engl. foreground*) de fundal (*engl. background*).

1 Metoda multiplicatorilor cu direcție alternantă

Metoda multiplicatorilor cu direcție alternantă (ADMM) [1] este un algoritm care rezolvă problemele de optimizare convexă prin împărțirea lor în probleme mai mici, fiecare dintre acestea fiind mai ușor de gestionat. ADMM rezolva probleme de optimizare de forma:

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x) + g(z)$$
$$\text{s.l: } Ax + Bz = c$$

unde $A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{p \times m}$ si $c \in \mathbb{R}^p$ Observati ca functia obiectiv este separabila in x respectiv z, legatura intre varaibile avand loc prin intermediul constrangerii de egalitate. In prima etapa a algoritmului, se formuleaza **functia Lagrange augumentata**:

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

Observati ca la functia Lagrange (cea obisnuita) s-a adaugat si un *termen de regularizare*. Noul termen adaugat asigura robustetea metodei si o convergenta mai buna fara alte presupuneri cum ar fi convexitatea stricta sau continuitatea functiei. Regulile de actualizarea ale ADMM-ului sunt:

$$\begin{aligned} x^{k+1} &:= \underset{x}{\operatorname{argmin}} L_\rho\left(x, z^k, y^k\right) \\ z^{k+1} &:= \underset{z}{\operatorname{argmin}} L_\rho\left(x^{k+1}, z, y^k\right) \\ y^{k+1} &:= y^k + \rho\left(Ax^{k+1} + Bz^{k+1} - c\right), \end{aligned} \tag{1}$$

Funcția obiectiv $(f(x_{k+1}) + g(y_{k+1}))$ converge către $f^* + g^*$ (cu rata $O(\frac{1}{k})$).

2 Formularea problemei PCA robusta

Input: Fie o matrice $X \in \mathbb{R}^{n\times m}$ data.
Scopul: Dorim sa gasim o matrice $L \in \mathbb{R}^{n\times m}$ de **rang mic** si $S \in \mathbb{R}^{n\times m}$ **rara** astfel incat

$$X = L + S.$$

Considerand cerintele de mai sus putem formula urmatoarea problema de optimizare constransa:

$$\begin{aligned} \min_{L,S} \quad & \|L\|_* + \lambda\|S\|_1 \\ \text{s.l.:} \quad & X = L + S \end{aligned}$$

unde $\|\cdot\|_*$ este norma nucleara (suma de valori singulare) si $\lambda \in \mathbb{R}$ este un parametru cu care controlam nivelul de raritate din S.

$$\mathcal{L}_\rho(L,S,\Lambda) = \|L\|_* + \lambda\|S\|_1 + \frac{\rho}{2}\|L + S - X + \frac{1}{\rho}\Lambda\|_F^2 - \frac{1}{2\rho}\|\Lambda\|_F^2.$$

Aplicand ADMM gasim urmatoarele expresii:

$$\begin{aligned} L_{k+1} &= \arg\min_L \left\{ \|L\|_* + \frac{\rho}{2}\|L + S_k - X + \frac{1}{\rho}\Lambda_k\|_F^2 \right\} \\ S_{k+1} &= \arg\min_S \left\{ \lambda\|S\|_1 + \frac{\rho}{2}\|L_{k+1} + S - X + \frac{1}{\rho}\Lambda_k\|_F^2 \right\} \\ \Lambda_{k+1} &= \Lambda_k + \rho(L_{k+1} + S_{k+1} - X) \end{aligned}$$

Aplicam conditia de optimalitate pentru a gasi expresia explicita a lui L_{k+1}:

$$\nabla\left(\|L_{k+1}\|_* + \frac{\rho}{2}\|L_{k+1} + S_k - X + \frac{1}{\rho}\Lambda_k\|_F^2 \right) = 0$$

Amintim ca daca cunoastem descompunerea DVS $L_{k+1} = U_{k+1}\Sigma_{k+1}V_{k+1}^T$, atunci $\nabla\|L_{k+1}\|_* = U_{k+1}V_{k+1}^T$. Astfel avem:

$$\begin{aligned} U_{k+1}V_{k+1}^T + \rho\left(L_{k+1} + S_{k+1} - X + \frac{1}{\rho}\Lambda_k \right) &= 0 \\ U_{k+1}\left[\Sigma_{k+1} + \frac{1}{\rho}\mathbb{I} \right] V_{k+1}^T &= X - S_k - \frac{1}{\rho}\Lambda_k. \end{aligned}$$

Daca DVS-ul lui $X - S_k - \frac{1}{\rho}\Lambda_k = U_{k+1}\Upsilon V_{k+1}^T$ atunci:

$$\Sigma_{k+1} + \frac{1}{\rho}\mathbb{I} = \Upsilon$$

Ceea ce in final obtinem:

$$L_{k+1} = U\left[\Upsilon - \frac{1}{\rho}\mathbb{I} \right]_+ V^T,$$

unde $U\Upsilon V^T = SVD(X - S_k - \frac{1}{\rho}\Lambda_k)$.
Prin urmare formulele explicite ale ADMM-ului sunt:

$$L_{k+1} = SVT_{\frac{1}{\rho}}\left(X - S_k - \frac{1}{\rho}\Lambda_k\right) \tag{2}$$

$$S_{k+1} = ST_{\frac{\lambda}{\rho}}\left(X - L_{k+1} - \frac{1}{\rho}\Lambda_k\right) \tag{3}$$

$$\Lambda_{k+1} = \Lambda_k + \rho(L_{k+1} + S_{k+1} - X) \tag{4}$$

unde functiile $SVT_\alpha(A)$ si $ST_\alpha(A)$ au expresiile:

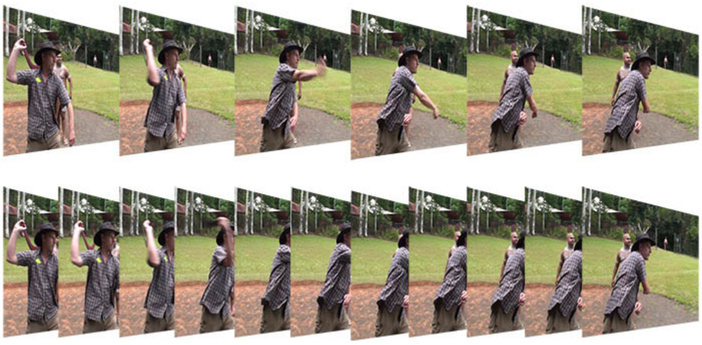
$$SVT_\alpha(A) = U[\Sigma - \alpha\mathbb{I}]_+ V^T, \text{ stiind } A = U\Sigma V^T$$

si

$$(ST_\alpha(A))_{i,j} = \begin{cases} A_{i,j} - \alpha & if \quad A_{i,j} > \alpha \\ 0 & if \quad |A_{i,j}| \leq \alpha \\ A_{i,j} + \alpha & if \quad A_{i,j} < -\alpha \end{cases}$$

3 Aplicatia: Monitorizarea video

Un video este o secventa de imagini:



Fiecare imagine (frame) din video este transformata intr-un vector si adaugata intr-o matrice $X^{n\times m}$, unde m este numarul de imagini si n este numarul de pixeli intr-o imagine.

Se stie ca un fundal static dintr-un video este reprezentat de o matrice de rang mic (mai exact rangul 1). Iar prim-planul, ce contine obiecte in miscare este o matrice rara:

Originale Frame

Low_Rank " L "

Sparse " S "

Prin urmare putem aplica formularea PCA robusta pentru a realiza aceasta descompunere a lui X.

4 Cerinte

Pentru video-ul *visiontraffic.avi*, implementati algoritmul ADMM.

- Un frame are dimensiunile 360×640 si selectam 30 de frame-uri $\Rightarrow X^{230400 \times 30}$.
- Critereiul de oprire este $rank(L_{k+1}) = 1$
- Initializati $\rho = 0.007$ si $\lambda = 0.0002$. Atentie! Daca ρ este prea mare algoritmul ADMM nu va converge.
- Afisati rezidul ($\|L_{k+1} + S_{k+1} - X\|$) de a lungul iteratiilor.
- Alegeti aleator un frame i si afisati descompunerea gasita, i.e fundalul si prim-planul asociat frame-ului i.
- Hint: Pentru o viteza buna de executie, evitati folosirea for-urilor in implementare.

References

[1] https://stanford.edu/~boyd/admm.html#:~:text=The%20alternating%20direction%20method%20of,are%20then%20easier%20to%20handle.