UMONS
Université de Mons

Faculté Polytechnique

Optimisation Non Linéaire
Projet

GUILY Thomas, MAGAÑA LÓPEZ Gustavo

POLYTECH MONS

Sous la direction du professeur VANDAELE Arnaud

Année Académique 2018-2019

ACADÉMIE UNIVERSITAIRE WALLONIE-BRUXELLES

Contents

Chapter 1

Étude du problème NNLS (non-negative least squares)

1.1 Développement du problème NNLS

Le problème des moindres carrés non-négatifs est un problème de type *moindres carrés* où les variables ne sont pas autorisées à devenir négatives. L'objectif est de trouver le vecteur x minimisant la quantité suivante:

$$\min_x \ \frac{1}{2}\|Ax \ - \ b\|_2^2 \quad tel \ \ que \quad x \geq 0 \tag{1.1}$$

$$A \in \mathbb{R}^{m \times n}, \ \ b \in \mathbb{R}^m, \ \ x \in \mathbb{R}^n \tag{1.2}$$

Nous pouvons écrire le problème sous la forme quadratique comme nous avions fait dans le premier devoir, ce qui nous donne:

$$\min \ \frac{1}{2}x^T A^T Ax \ - \ b^T Ax \ + \ \frac{1}{2}b^T b \tag{1.3}$$

$$\nabla f(x) \ = \ A^T Ax \ - \ A^T b \tag{1.4}$$

$$\nabla^2 f(x) \ = \ A^T A \tag{1.5}$$

Le lagrangien de notre fonction et son gradient:

$$\mathcal{L}(x, \lambda) \ = \ \frac{1}{2}x^T A^T Ax \ - \ b^T Ax \ + \ \frac{1}{2}b^T b \ - \ \lambda^T x \tag{1.6}$$

$$\nabla_x \mathcal{L}(x, \lambda) \ = \ A^T Ax \ - \ A^T b \ - \ \lambda \tag{1.7}$$

$$\nabla_x \mathcal{L}(x, \lambda) \ = \ \nabla f(x) \ - \ \lambda \tag{1.8}$$

1.2 Conditions d'optimalité du premier ordre

Nous prennons les conditions KKT et nous les développons pour le probème traité:

$$h_i(x^*) \geq 0 \ \ \forall i \in I \ \ \implies x^* \geq 0 \tag{1.9}$$

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0 \ \ \implies \nabla f(x^*) = \lambda^* \tag{1.10}$$

$$\lambda_i^* \geq 0 \ \ \forall i \in I \tag{1.11}$$

$$\lambda_i^* h_i(x^*) = 0 \ \ \forall i \in I \tag{1.12}$$

En utilisant (1.10) nous pouvons réécrire les conditions d'optimalité de premier ordre:

$$x^* \geq 0 \tag{1.13}$$

$$\nabla f(x^*) \geq 0 \tag{1.14}$$

$$\nabla f(x^*)^T x^* = 0 \tag{1.15}$$

L'équation (1.15) implique que chaque variable est "conjuguée" par rapport au gradient. C'est-à-dire: Lorsqu'une variable ne vaut pas zéro le gradient doit valoir zéro et viceversa. Par définition les conditions KKT sont nécessaires mais pas suffisantes. Néamoins, s'il s'agit d'un problème convexe (matrice hessienne définie positive, $A^T A$ dans ce cas) les conditions deviennent nécessaires et suffisantes.

1.3 Calcul du paramètre γ

Etant donné un vecteur $x \in \mathbb{R}^n$, quelle valeur faut-il donner au paramètre $\gamma \in \mathbb{R}$ afin de minimiser la quantité $\|A(\gamma x) \ - \ b\|_2^2$?

Nous commençons en mettant le problème sous la forme quadratique, ce qui nous donne:

$$\min \ \gamma^2 x^T A^T A x \ - \ 2\gamma b^T A x \ + \ b^T b \tag{1.16}$$

En sachant que tous les éléments de la somme sont scalaires, nous pouvons introduire des définitions qui nous permettront trouver facilement la valeur optimale de γ.

$$p = x^T A^T A x, \quad q = -2b^T A x, \quad r = b^T b \tag{1.17}$$

$$p\gamma^2 \ + \ q\gamma \ + \ r \tag{1.18}$$

Pour une équation de la forme (1.18), nous pouvons calculer directement le γ optimal, ce qui nous donne $\gamma = \frac{-q}{2p}$. En remplaçant les définitions de p et q, nous trouvons la valeur de gamma qui permet de minimiser la fonction objectif:

$$\gamma \ = \ \frac{b^T A x}{x^T A^T A x} \tag{1.19}$$

Chapter 2

Factorisation nonnégative de matrices (NMF)

2.1 Introduction :

Étant donné une matrice $X \in \mathbb{R}^{m\times n}$ et un naturel $r < \min(m,n)$, le problème consiste à trouver des matrices nonnégatives $W \in \mathbb{R}^{m\times r}$ et $H \in \mathbb{R}^{r\times n}$ minimisant la quantité suivante:

$$\min_{W,H} \|X - WH\|_F^2 \quad tel\ que \quad W \geq 0\ et\ H \geq 0 \tag{2.1}$$

Cette forme est équivalente à la suivante:

$$\min_{W,H} trace[(X - WH)(X - WH)^T] \tag{2.2}$$

Si on considère le cas scalaire, en prennant $m = n = r = 1$, le problème devient:

$$\min_{w,h}(x - wh)^2 = \min_{w,h} x^2 - 2xyw + y^2w^2 \quad tel\ que \quad w, h \geq 0 \tag{2.3}$$

Dont le gradient et la matrice Hessienne sont les suivantes:

$$\nabla f(w,h) = \begin{bmatrix} 2wh^2 - 2xh \\ 2w^2h - 2xw \end{bmatrix} \tag{2.4}$$

$$\nabla^2 f(w,h) = \begin{bmatrix} 2h^2 & 4wh - 2x \\ 4wh - 2x & 2w^2 \end{bmatrix} \tag{2.5}$$

Si nous prennons $x = 2$, $w = 2$ et $h = 2$ on peut aiément trouver une matrice Hessienne qui n'est pas sémi-défini positive ce qui nous montre que la convexité n'est pas une caractéristique que l'on peut assumer même pas pour le cas le plus simple.

$$\nabla^2 f(2,2) = \begin{bmatrix} 8 & 12 \\ 12 & 8 \end{bmatrix} \quad \lambda_1 = -4, \ \ \lambda_2 = 20 \tag{2.6}$$

2.2 Convexité

Le problème (lorsque H et W ne sont pas fixés) ne peut pas être convexe.

$$\min_{W,H} \|X \ - \ WH\|_F^2 \quad tel\ que \quad W \geq 0\ et\ H \geq 0 \tag{2.7}$$

En effet, la solution d'un tel problême ne peut être que locale. De plus la solution n'est pas unique, car toute matrice D fournit des résultats équivalent (ajustement) :

$$X \simeq WDD^{-1}H \tag{2.8}$$

Par contre, il est possible de résoudre deux sous problèmes d'optimisations convexes en fixant à tour de rôles W et H, on obtient :

Pour W fixe, on optimise colonne par colonne H, le sous problème est convexe :

$$\min_{H(:,i)} \|X(:,i) \ - \ WH(:,i)\|_F^2 \quad tel\ que \quad H(:,i) \geq 0 \quad avec \quad i = 1, ..., n \tag{2.9}$$

Pour H fixe, on optimise ligne par ligne W, le sous problème est convexe :

$$\min_{W(j,:)} \|X(j,:) \ - \ W(j,:)H\|_F^2 \quad tel\ que \quad W(j,:) \geq 0 \quad avec \quad j = 1, ..., m \tag{2.10}$$

2.3 Programme

Notre algorithme (nmf Guily Magana.m) utilise la méthode du gradient pour optimiser de manière alternée H puis W. La colonne de H et la ligne de W qui seront optimisées sont choisies au hasard à chaque itération. A la fin de chaque mise à jour, le résultat est projeté pour prendre en compte les contraintes de nonégativité imposées aux matrices W et H.

2.4 RunMeNMF

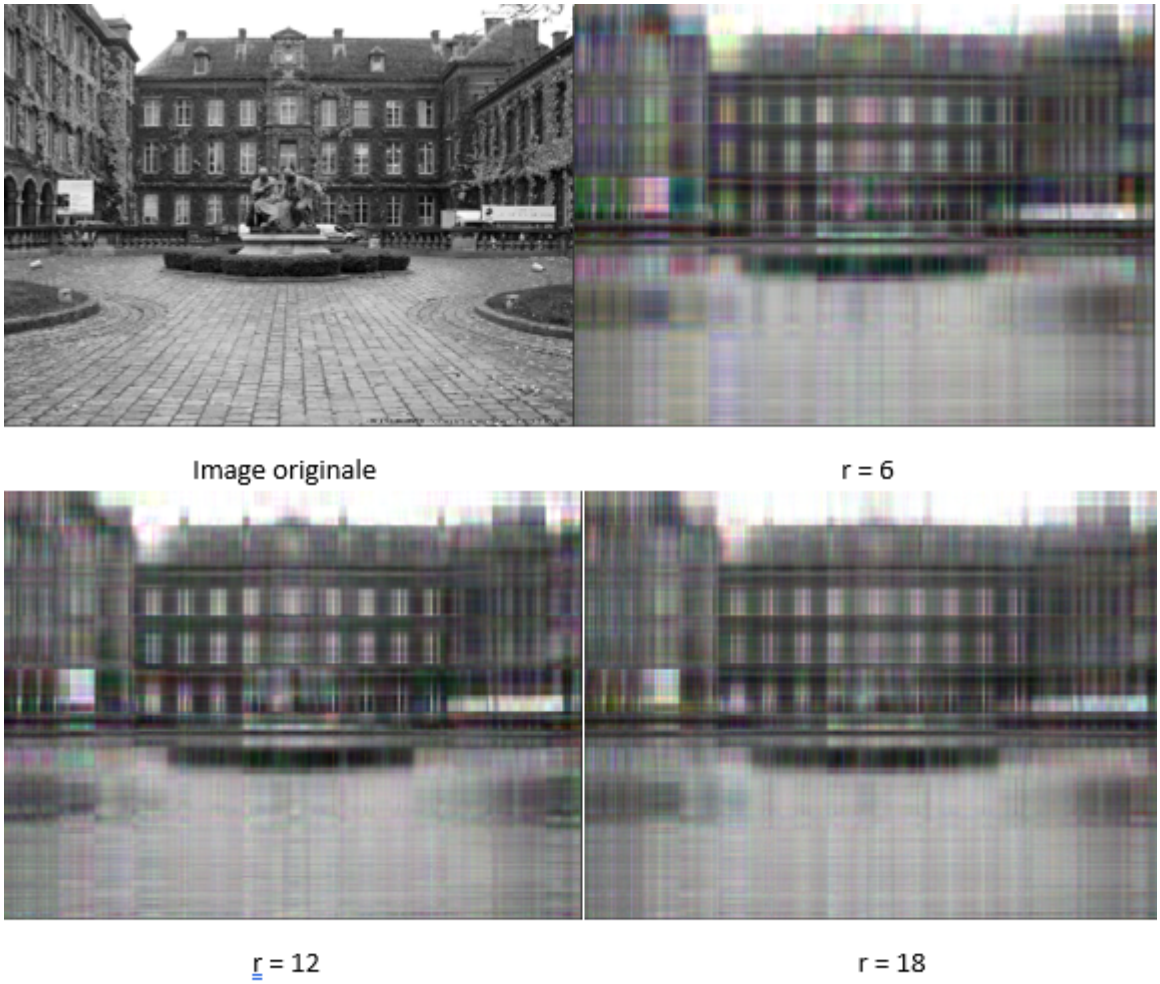
Voici la matrice $X =$

$$\begin{pmatrix} 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 2 & 1 \\ 1 & 0 & 0 & 1 & 2 & 2 \\ 2 & 1 & 0 & 0 & 1 & 2 \\ 2 & 2 & 1 & 0 & 0 & 1 \\ 1 & 2 & 2 & 1 & 0 & 0 \end{pmatrix}$$

Gràce à notre code, nous arrivons à une factorisation exacte sans tenir compte de la nonégativité jusqu'à une valeur $r = 3$, en dessous il s'agit d'une approximation grossière. En prenant en compte les contraintes de nonnégativité imposées sur les matrices W et H, la factorisation est possible jusqu'à une valeur de $r = \min(n, m)$, hors ici $n = m = 6$, donc la factorisation est possible jusque $r = 6$, en dessous, X est approximée très grossièrement.

2.5 RunMeNMF2 (compression d'image)

Observons cette fois-ci le résultat d'une telle opération sur les matrices d'une image(.jpeg). Il faut résoudre ce même problème pour chaque couleur (RGB), soit trois fois. Voici quelques résultats pour différentes valeurs de r :



On peut ici bien se représenter le comportement de notre algorithme pour de grandes dimensions de départ: si r est trop petit, l'approximation contient trop peu d'informations et l'image est très floue. Au contraire, si r est trop grand, la quantité de calculs augmentant exponentiellement, les mises à jour mettent beaucoup plus de temps. Il faut donc trouver un juste milieu, la valeur de r la plus stable semble être située entre 10 et 15.

2.6 RunMeNMF3

La factorisation de la matrice donnée nous permet d'afficher le facteur W et le facteur H sous format.jpg (ici pour r = 8) :

On peut observer que la matrice W rassemble plutôt les informations tirant aux traits et à la forme des visages tandis que H rassemble les pigments et contrastes des différents visages.

Si on augmente la valeur de r, on observe plus de types de visages en affichant W, en toute logique, la précision devrait donc augmenter avec r.