Projet d'optimisation linéaire

Récupération d'une image floutée (deblurring)

Deadline: lundi 13 novembre

1 Description du problème

Votre appareil photo ne fonctionne plus convenablement : au lieu de vous renvoyer précisément l'intensité de chaque pixel, il vous renvoie une moyenne pondérée de l'intensité de pixels voisins. De plus, un certain nombre de capteurs associés à un petit ensemble de pixels ne fonctionnent plus correctement; voir ci-dessous pour un exemple.

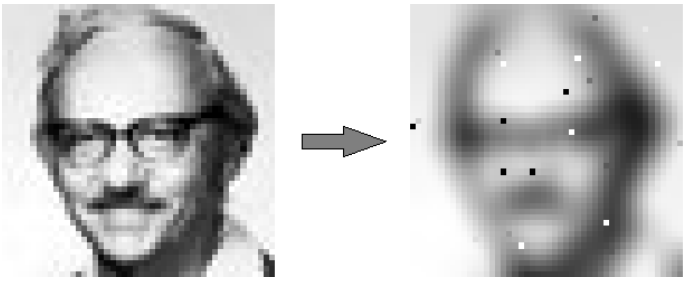

Figure 1: Image originale à gauche et image floutée et bruitée à droite.

Floutage Soit $\bar{x} \in [0,1]^n$ le vecteur contenant l'intensité de chaque pixel de l'image originale non floutée (entre 0 et 1). On observe un vecteur flouté $\tilde{x} \in [0,1]^n$: chaque entrée de $\tilde{x}$ est une combinaison linéaire des entrées de $\bar{x}$. Les poids de cette combinaison dépendent du type de floutage. Un floutage très simple consiste par exemple à avoir chaque entrée de $\tilde{x}$ égale à la moyenne des entrées de $\bar{x}$ correspondant à des pixels voisins.

De manière plus générale, en l'absence d'autres sources de bruit, on a une relation linéaire entre $\bar{x}$ et $\tilde{x}$:

$$\tilde{x} = A\bar{x},$$

où la matrice $A \in \mathbb{R}^{n \times n}$ est la matrice de floutage. Malheureusement, cette matrice A n'est pas inversible : étant donné $\tilde{x}$ et A, l'ensemble des solutions x du système $Ax = \tilde{x}$ est soit vide (s'il y a du bruit), soit contient un nombre infini de solutions (en particulier, si x est solution, $x + c$ en est également une où c est une constante).

En pratique, on va chercher à identifier la solution d'énergie minimale:

$$\min_{0 \leq x \leq 1} ||x||_1 \quad \text{tel que} \quad Ax = \tilde{x},$$

où $||x||_1 = \sum_i |x_i|$ est la norme 1 du vecteur x.

Bruit creux De plus, l'observation $\tilde{x}$ est également corrompue avec du bruit : $\tilde{x} = A\bar{x} + b$ où b est le vecteur contenant le bruit. On supposera ici que l'intensité d'un petit nombre de pixels est erronée (c'est ce qu'on appelle un bruit creux, ou parcimonieux). Ainsi, pour estimer $\bar{x}$, on considère le problème d'optimisation suivant

$$\min_{0 \leq x \leq 1} \; ||Ax - \tilde{x}||_1 + \lambda ||x||_1, \tag{1}$$

où λ est un paramètre positif qui dépend du niveau de bruit. En effet, la norme 1 $||Ax - \tilde{x}||_1$ est plus indiquée que la norme 2 dans le cas de bruit creux (la norme 2, $||x||_2^2 = \sum_i x_i^2$, est idéale pour du bruit Gaussien).

L'objectif de ce projet est l'étude du problème (1), et ainsi de reconstruire, de manière approchée, l'image originale $\bar{x}$ à partir de $\tilde{x}$ et A.

Remarque. En pratique, la matrice de floutage A n'est en général pas connue exactement. Cependant, il existe des techniques efficaces pour l'approcher; voir par exemple le livre *Deblurring images: matrices, spectra, and filtering* par Nagy et O'leary, SIAM, 2006. On supposera dans ce projet que cette matrice est connue (l'identifier peut aussi être formulé comme un problème d'optimisation!).

2 Questions

1. Modélisez le problème (1) comme un problème d'optimisation linéaire. Expliquez votre raisonnement.

2. Ecrivez ce problème linéaire sous forme standard.

3. Utilisez Octave[1] et la fonction ***glpk*** pour déflouter les images fournies (ExampleX.mat, $X = 0$, 1, et 2).

4. La solution obtenue est-elle un sommet du polyèdre correspondant? Justifiez.

5. Pour les images Example0.mat et Example1.mat, étudiez la sensibilité de la solution en fonction de la valeur de λ. L'image originale étant fournie, vous pouvez calculer l'erreur de reconstruction en fonction de λ. Quelles valeurs de λ semblent fonctionner le mieux pour chaque exemple? (Utilisez des échelles logarithmiques pour mieux visualiser ce comportement.)

Consignes. Le travail se réalise par groupe de 2 (*un* groupe de 3 est autorisé si nécessaire). Veuillez fournir avec le rapport les codes implémentés en annexe. Le rapport ne doit pas dépasser 6 pages (en dehors des annexes contenant le code, et la page de garde). Le tout est à envoyer pour le 13 novembre à arnaud.vandaele@umons.ac.be.

[1]Octave est un langague extrêment similaire à Matlab, excepté qu'Octave est un logiciel libre (=gratuit). Une version avec une interface graphique similaire à Matlab est disponible sur la page `https://github.com/octave-de/mxeoctave.osuv.de` pour Windows. Voir `http://wiki.octave.org/Octave_for_MacOS_X` pour les Mac's. Cependant, il est également possible de faire le projet en Matlab en utilisant la fonction ***linprog***.