TP de méthodes variationnelles - IMA203

Júlia Togashi de Miranda

Décembre 2020

1 Débruitage par régularisation quadratique

1. *Comment utiliser l'outil resoud_quad_fourier pour trouver le minimiseur de cette énergie (voir le programme minimisation_quadratique) ?*

Nous avons le problème:

$$\underset{x}{\operatorname{argmin}} ||u-v||^2 + \lambda||k*u-w||^2 + \int ||\nabla u||^2 \quad ;(v->cte)$$

Nous pouvons récrire:

$$||u-v||^2 + ||(\sqrt{\lambda}k)*u-\sqrt{\lambda}w||^2 + \int ||\nabla u||^2$$

$$||\delta*u-v||^2 + ||(\sqrt{\lambda}k)*u-\sqrt{\lambda}w||^2 + ||K_x*u-0||^2 + ||K_y*u-0||^2$$

Donc:

$$\sum_i ||K_i*u-V_i||^2$$

Avec Fourier:

$$\sum_i \sum_{f\in freq} |(K_i*\hat{u}-V_i)(f))|^2$$

Les deux sommes sont indépendants, donc nous pouvons échanger f et i:

$$\sum_{f\in freq} \sum_i |\hat{K}_i(f)\hat{u}(f)-\hat{V}_i(f)|^2$$

Pour simplifier: $\hat{u}\in\mathbb{R}; \hat{K}_i\in\mathbb{R}; \hat{V}_i\in\mathbb{R}$

$$= \sum_{f\in freq} [\hat{u}(f)^2(\sum_i \hat{K}_i(f))^2 - 2\hat{u}(f)(\sum_i \hat{V}_i(f)\hat{K}_i) + cte]$$

Minimum est attendu pour $\hat{u}(f)$ (c'est un fonction quadratique):

$$\hat{u}(f) = \frac{\sum_i \hat{V}_i(f)\hat{K}_i(f)}{\sum_i \hat{K}_i(f)^2}$$

La fonction minimisation_quadratique range tout pour minimiser u_1 donne des fonctions génériques que sont résulte de resoud_quad_fourier.

* (Le caractère fini des images traitées implique de faire une hypothèse sur la continuation des images hors de leur support connu: on fera l'hypothèse de périodicité pour simplifier la présentation, ça veut dire symétriser avant traiter.)

2. *Décrire le résultat de ce débruitage lorsque λ est très grand ou très petit.*

Comme nous pouvons observer dans 1, quand nous avons un λ très petit, le débruitage est très faible. À mesure que nous augmentons λ, l'image restaurée s'approche de l'image parfaite u. Si nous continuons à l'augmenter, l'image résultant devient floue.

À mesure que nous augmentons λ, nous augmentons le poids du terme de régularisation. Cela signifie que nous privilégions de plus en plus une image résultante avec une statistique d'image naturelle, en place de l'attache aux données.

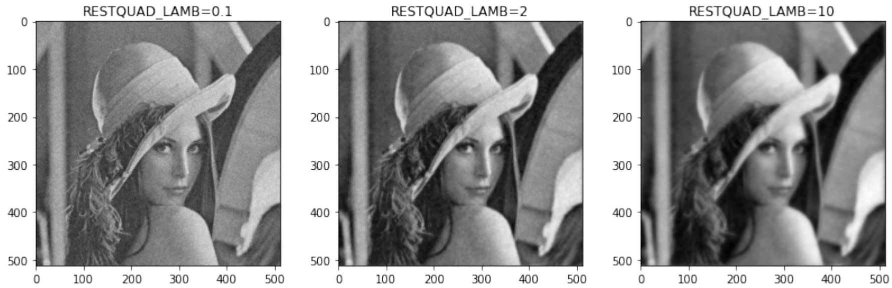

Figure 1: Résultat de la débruitage de l'image Lena par régularisation quadratique, avec λ respectivement égale a 0.1, 2 et 10.

3. *Après avoir ajouté un bruit d'écart type $\sigma = 5$ à l'image de lena, trouver (par dichotomie) le paramètre λ pour lequel $||\tilde{u} - v||^2 \sim ||u - v||^2$. C'est-à-dire le paramètre pour lequel l'image reconstruite $\tilde{u}$ est à la même distance de l'image parfaite u que ne l'est l'image dégradée.*

Par dichotomie, nous avons commencé avec un intervalle de λ suffisamment grande. À chaque fois nous faisons $\frac{\lambda_{min}+\lambda_{max}}{2}$ et nous remplaçons l'une des valeurs. Avec l'algorithme décrit par 2, nous obtenons $\lambda = 0.331$

```
lamb=0
restau=minimisation_quadratique(imb,lamb)
x=np.linalg.norm(restau-imb)**2
y=np.linalg.norm(im-imb)**2
lamb1=0
lamb2=1000

while np.abs(x-y) >0.0001:
  lamb=(lamb1+lamb2)/2
  restau=minimisation_quadratique(imb,lamb)
  x=np.linalg.norm(restau-imb)**2
  y=np.linalg.norm(im-imb)**2
  #print(x,y)
  #print(lamb1,lamb2)
  if (x<y):
    lamb1=lamb
  else:
    lamb2=lamb

print("lambda égale à: ",lamb)
viewimage(restau,titre='RESTQUAD_LAMB='+str(lamb))
```

Figure 2: Le code pour le calcul de la dichotomie et l'image résultant avec λ égale au trouvé par dichotomie

4. *Ecrire un algorithme pour trouver le paramètre λ tel que $||\tilde{u} - u||^2$ soit minimale. (dans le cadre de ce TP on connait l'image parfaite u). Commentaires ?*

Sur 3, nous pouvons observer l'algorithme qu'il s'agit de calculer l'image débruité pour un intervalle de λ et calculer la distance de l'image parfait. Aprés, on cherche la norme minimale.

```
#%% COMPARAISON des methodes
# vous pouvez vous inspirer de ce qui suit pour trouver les meilleurs
# parametres de regularisation


errvt=[]
erreur=[]
vk=np.arange(0,0.25,0.005);
for k in vk:
    print (k)
    restq=minimisation_quadratique(imb,10**(k))
    erreur.append(norm2(im-restq))
    #restva=vartotale_Chambolle(imb,10**(k));
    #errvt.append(norm2(restva-myim));



plt.plot(10**vk,erreur)
plt.title("norme par lambda")
plt.show()
```

Figure 3: L'algorithme pour trouver le λ optimale.

Nous avons sur le graphe que le λ que minimise la distance est égale à 1.16.

Nous pouvons voir qu'en augmentant λ, nous rapprochons le résultat de l'image originale, car nous la régularisons de plus en plus. C'est-à-dire en rapprochant ses statistiques d'une image naturelle, ce qui rend l'image débruité. A partir d'un moment, cependant, la régularisation commence à dé-caractériser l'image (nous perdons l'attache aux données), obtenant une image floue plus éloignée de l'original.

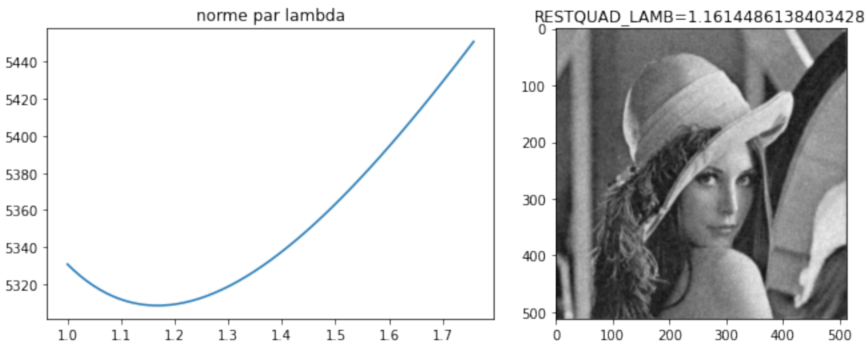

Figure 4: $||\tilde{u} - u||^2$ pour chaque lambda et l'image avec lambda que nous donne le minimal.

2 Débruitage par variation totale

2.1 Descente de gradient

Atteignez-vous toujours le même minimum d'énergie?

Non, comme nous pouvons avoir sur 5, nous n'avons toujours pas le même énergie.

Comme nous l'avons vu en classe, nous avons des problèmes numériques avec la minimisation de la VT par descente de gradient à pas constant. Si nous choisissons ρ très grand, c'est possible que la variation totale augment (à cause du valor absolut, nous sommes entrés dans une boucle entre deux points). Même en choisissant ρ pour que l'énergie totale diminue, nous avons toujours des valeurs qui nous donneront de meilleurs résultats compte tenu d'un certain nombre d'interactions.

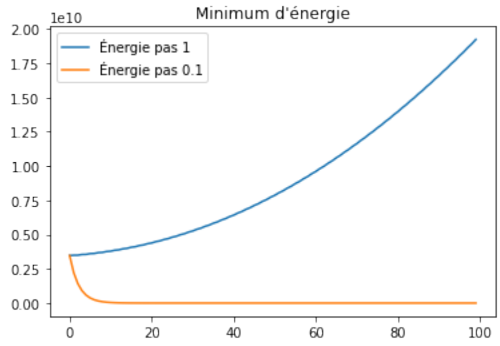

Figure 5: Minimum d'énergie trouvé par pas de 0.1 et 1

2.2 Projection Chambolle

Utilisez ce programme et que constatez-vous quant à la vitesse de cette algorithme et sa précision (minimisation effective de E2) par rapport à la descente de gradient.

Comme nous l'avons déjà remarqué dans la question précédente, le problème avec la méthode du gradient, c'est qu'en fonction de la choisi, nous obtenons des énergies différentes. C'est aussi une méthode beaucoup plus lente.

Pour obtenir des résultats énergétiques similaires entre les deux,il faudrait faire 1000 pas de descente de gradient avec un pas de 0.01. Cela nous donne un runtime (dans google colab), pour le descente de gradient de 14.5 secondes, en place de les 0.28 secondes pour la projection Chambolle.

En place, par exemple se nous faisons 100 pas de descente de gradient avec un pas de 0.1, nous obtenons un runtime meilleur, égale à 1.5 secondes, mais l'énergie résultant c'est 1.3 fois plus grande que de la projection Chambolle.

2.3 Comparaison

Après avoir fixé une image bruitée par un bruit de 25. Trouver pour chacune des deux méthodes le meilleur paramètre et comparez qualitativement le résultat obtenu par les deux méthodes pour le débruitage.

Nous avons obtenu pour la régularisation quadratique, le meilleur $\lambda = 1.12$ et ça nous donnée $||\tilde{u} - u|| = 5308$. Pour la projection Chambolle,le meilleur $\lambda = 42.6$ et ça nous donnée $||\tilde{u} - u|| = 4423$. Donc nous pouvons constater que nous obtenons un meilleur résultat quantitativement et qualitativement pour la projection Chambolle (visuellement l'image se ressemble plus à l'image originale).

* Nous devons faire un addendum, qui en fonction des paramètres choisis, il est possible que quantitativement, le résultat du la régularisation quadratique soit plus proche de l'image d'origine. Mais, visuellement, la deuxième méthode est meilleure (le résultat pour le quadratique nous donne des bordés floues).

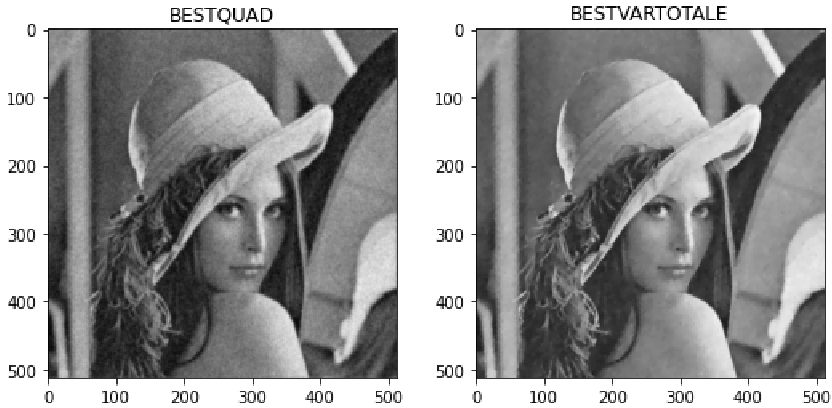

Figure 6: Image résultante pour le meilleur paramètre de régularisation quadratique et de la Projection Chambolle, respectivement.