Université Paris-Dauphine | Optimisation numérique
Département MIDO
Chargé de cours : F.-X. Vialard.
Chargés de TD/TP : G. Jankowiak, A. Frouvelle.
DUMI2E troisième année | année 2012/2013

TP n°3 : Inpainting

1. Intro

Le but de ce TP est d'implémenter deux méthodes de descente (gradient conjugué et méthode de Newton) pour résoudre le problème d'inpainting vu en TD. Comme pour le TP précédent, une archive (à extraire *intégralement*) est disponible à l'adresse suivante :

`http://gjankowiak.github.com/files/opt/2013/tp3.tar.gz`.

Elle contient un certain nombre de fonctions, déjà implémentées ou à compléter, ainsi que le fichier `TP3_NomsBinome.m` que vous devrez renommer et remplir avec toutes les commandes et les commentaires nécessaires pour répondre aux questions. *Votre travail sera en grande partie noté à partir de l'exécution directe de ce fichier, assurez vous qu'il tourne sans problème avant de le rendre (en particulier, n'oubliez pas de joindre tous les fichiers que vous auriez rajouté).*

2. Gradient conjugué

Question 1. Compléter la fonction `GradientConj`, qui implémente l'algorithme du gradient conjugué. On rappelle l'algorithme pour minimiser $\frac{1}{2}\langle A(x), x\rangle + \langle b, x\rangle$:

Initialisation de $k = 0$, $x_0 \in \mathbb{R}^n$, $r_0 \leftarrow A(x_0) + b$ and $p_0 \leftarrow -r_0$ et itération tant que $r_k \neq 0$ de

$$\begin{cases} \delta \leftarrow \frac{\langle r_k, r_k\rangle}{\langle p_k, A(p_k)\rangle} \\ x_{k+1} \leftarrow x_k + \delta\, p_k \\ r_{k+1} \leftarrow r_k + \delta\, A(p_k) \\ \gamma \leftarrow \frac{\langle r_{k+1}, r_{k+1}\rangle}{\langle r_k, r_k\rangle} \\ p_{k+1} \leftarrow -r_{k+1} + \gamma\, p_k \\ k \leftarrow k+1 \end{cases} \tag{1}$$

Attention, `A` est la *fonction* $x \mapsto A(x)$!

Question 2. Tirer une matrice $M \in \mathcal{M}_{100}(\mathbb{R})$ en utilisant la fonction `randn` de MATLAB. En former la matrice de Gram notée A, c'est-à-dire la matrice $M^T M$. Vérifier que la matrice ainsi obtenue est inversible. Pourquoi est-ce que le problème de minimisation a-t-il une solution (unique) ? Tirer un vecteur b et utiliser l'algorithme du gradient conjugué sur A et b.

- Comparer avec la solution x_* obtenue par MATLAB en tracant le graphe $k \mapsto \|x_k - x_*\|$ avec x_k le résultat de l'algorithme à l'itération k.
- Est-ce que l'algorithme converge avec la vitesse attendue ? Sinon, comment l'expliquer ?

Question 3 (Formulation matricielle)**.** On veut maintenant minimiser la fonctionnelle $\mathcal{J}_1$ vue en TD, dont on rappelle l'expression discrete :

$$\mathcal{J}_1(g) = \frac{1}{2\beta}\left(\sum_{i=1}^{n-1}\sum_{j=1}^{n}(g_{i+1,j} - g_{i,j})^2 + \sum_{i=1}^{n}\sum_{j=1}^{n-1}(g_{i,j+1} - g_{i,j})^2\right) + \frac{1}{2}\sum_{i,j}\frac{1}{n^2}\, m_{i,j}\,(f_{i,j} - g_{i,j})^2,$$

que l'on doit d'abord mettre sous la forme $\frac{1}{2}\langle g, A(g)\rangle + \langle b, g\rangle$, où $\langle M, N\rangle = \mathrm{Tr}(N^T M)$, et A est une application linéaire $M_n(\mathbb{R}) \to M_n(\mathbb{R})$. On note $\|\cdot\|_e$ la norme associée.

- Que vaut b dans ce cas ?
- Mettre le terme de gauche sous la forme $\|Dg\|_e^2 + \|gD\|_e^2$, avec D une matrice à déterminer.
- Déterminer A.

Question 4. À l'aide de la fonction `LireImage` qui vous est fournie, obtenir les matrices `mf` et `m`, qui correspondent aux notations du TD, et implémenter la fonction `A_J1` qui calcule $A(g)$ pour la fonctionnelle $\mathcal{J}_1$.

Question 5. Utiliser `GradientConj` pour minimiser $\mathcal{J}_1$. Vous pouvez afficher le résultat à l'aide la fonction `AfficherImage`. Pourquoi choisit-on la méthode du gradient conjugué ?

Question 6. Comparer la convergence de cette méthode avec une descente de gradient à pas fixe en essayant de jouer sur le pas. Faire un graphe de la convergence du gradient à pas fixe et du gradient conjugué.

Question 7. Qu'obtient-on en prenant des valeurs extrêmes pour β ? Comment l'expliquez-vous ?

3. Méthode de Newton

On veut maintenant utiliser la fonctionnelle $\mathcal{J}_\varepsilon$ pour faire l'*inpainting* de l'image.

Question 8. Pourquoi est-ce qu'on ne peut pas utiliser la méthode du gradient conjugué ici ?

Question 9. La méthode de Newton utilise la direction de descente définie par :

$$-[\nabla^2 F(g)]^{-1}(\nabla F(g))\,.$$

En utilisant `GradientConj` pour calculer cette direction de descente, compléter la fonction `Newton` qui implémente la méthode de Newton à pas fixe.

Question 10. Tracer la vitesse de convergence et la comparer à une descente de gradient à pas fixe. Comme le calcul de la direction de descente pour l'algorithme de Newton est relativement longue, il peut-être plus juste de simplement chronométrer l'exécution des deux méthodes. Faites-le à l'aide des commandes `tic`, qui démarre le chronométrage, et `toc`, qui l'arrête.

Faire un tableau des vitesses de convergence en fonction du nombre d'itérations du gradient conjugué dans le calcul de la direction de descente (prendre en particulier, 10, 40 et 100 itérations). Au maximum, de combien d'itérations a-t-on besoin pour la convergence du gradient conjugué ?

Question 11. Comparer les résultats qualitatifs de ce nouveau modèle avec le précédent. Proposer une explication.