Année universitaire
2020 - 2021

IMI – Optimisation et problèmes inverses — S8

TP1 – Descente de gradient

Marion Foare

Objectifs.

- Compréhension de l'**algorithme de descente de gradient**
- Compréhension de l'apport de la méthode numérique par rapport à une résolution directe
- Compréhension des limites de l'algorithme
- Compréhension du **modèle des moindres carrés**
- Compréhension du **modèle de Tikhonov**
- Compréhension des limites des modèles
- Implémentation de l'algorithme pour la minimisation des modèles des moindres carrés et de Tikhonov en 1D et en 2D

Déroulement. Ce TP se déroule sur **2 séances** et est à effectuer par binôme sous Matlab/Octave. Vous trouverez sur CPe-campus une archive contenant l'ensemble des fichiers nécessaires à la réalisation de ce TP.

Ce TP fait partie intégrante de la construction de votre cours sur les méthodes de descente, et les modèles différentiables. Il doit vous permettre à la fois de comprendre le fonctionnement de ces méthodes, d'illustrer votre cours, mais aussi de mettre en évidence l'importance de certaines hypothèses, l'apport de ces méthodes par rapport à des méthodes de résolution classiques, et les limites de ces méthodes. Il est dès lors indispensable que vous preniez le temps d'étudier l'influence des paramètres et, le cas échéant, de vos choix d'implémentation.

Bien qu'il soit relativement guidé, n'hésitez donc pas à sortir des sentier battus, à vous poser vos propres questions et à prendre du recul sur les pistes de réflexion proposées dans ce TP.

Configuration. Ce TP nécessite les librairies suivantes :

- Matlab : *Image Processing Toolbox*
- Octave : *Image toolbox* `pkg install -forge image`
  puis, en début de script : `pkg load image`

Compte-rendu. A la fin des 2 séances, vous devrez avoir rédigé un compte-rendu (max. 10 pages) sous forme de cours sur les méthodes de descente, et les modèles proposés. Le format est semi-libre : il peut s'agir d'un document pdf, ou d'un document partagé type Google doc ou Overleaf (ce dernier étant vivement encouragé).

Chaque semaine, un retour commenté pourra être fait sur votre travail, afin de corriger les erreurs, ou vous inciter à compléter certaines parties notamment.

L'évaluation de TP portera sur la compréhension des notions, évaluée en séances, l'évolution de votre compte-rendu et le compte-rendu final.

Ce TP a pour vocation d'être formateur, et peut nécessiter de travailler en groupe (fortement encouragé, qui plus est !). Toutefois, il vous est demandé **un compte-rendu par quadrinôme**, correspondant à **votre** restitution des notions. Il en va de même pour le code. Tout travail emprunté à un (ou plusieurs) autre(s) groupe(s) doit être **explicitement identifié**.

Nous rappelons que toute tentative de copie entrainera une sanction de l'ensemble des binômes concernés.

1 Contexte

On se concentrera dans ce TP sur des exemples en débruitage et déconvolution 1D (signal) et 2D (image). Pour autant, nous rappelons que l'optimisation est un domaine transversal, et permet, en fonction du choix de la fonction de coût, de traiter de nombreuses applications.
En particulier, on veut dans ce TP déruiter l'image ci-dessous :

On cherche pour cela à résoudre un problème d'optimisation, défini sous la forme :

$$\hat{x} \in \underset{x\in D}{\operatorname{argmin}}\ f(x)$$

Comme évoqué en cours, l'une des principales questions liées à l'optimisation est l'implémentation d'un algorithme qui nous permette d'approcher la solution même lorsqu'aucune forme explicite n'est envisageable, autrement dit de construire une suite $\{x_n\}_n$ qui converge vers $\hat{x}$.
Bien que chaque problème d'optimisation soit unique, et qu'il n'existe pas de méthode idéale permettant de résoudre efficacement tous les problèmes d'optimisation, les **méthodes dites "de descente"** font partie des algorithmes (itératifs) les plus classiques pour approcher numériquement le minimiseur $\hat{x}$. L'idée générale repose sur le fait de rechercher, à chaque itération, x_{k+1} tel que $f(x_{k+1}) < f(x_k)$.

2 Prise en main de la méthode

Cette section doit vous permettre de commencer à établir un plan de votre cours sur les méthodes de descente, et en particulier de la méthode de descente de gradient, que vous pourrez ensuite compléter au fur et à mesure.
Pour étudier le principe de la méthode de gradient, on se propose d'appliquer l'algorithme pour minimiser la fonction $f(x) = x^2$, i.e. résoudre le problème de minimisation 1D

$$\hat{x} = \underset{x\in\mathbb{R}}{\operatorname{argmin}}\ x^2 \tag{1}$$

Pistes de réflexion :

- résultat attendu ?
- solution analytique du problème (1)
- implémentation de la méthode de descente de gradient
- quel choix pour l'initialisation ?
- quel choix du pas de descente ?
- étude de la convergence de la méthode

Facultatif. Pour une transition plus douce entre cette partie et la suivante, il est possible de reprendre votre travail dans cette partie avec le problème intermédiaire suivant :

$$\hat{x} = \underset{x\in\mathbb{R}^N}{\operatorname{argmin}}\ \|x\|_2^2$$

3 Résolution du modèle des moindres carrés

Nous avons vu en cours que les fonctions de coût en image sont de la forme

$$f(x) = \mathcal{L}(Hx; z) + \lambda R(x)$$

où $\mathcal{L}$ est l'attache aux données (choisie en fonction du type de dégradation) et R est une régularisation (choisie en fonction du type de résultat souhaité).
Afin d'étudier l'influence de chaque terme, on se concentre dans un premier temps sur une fonction de coût sans régularisation (i.e. $f(x) = \mathcal{L}(Hx; z)$). Dans le cas d'une dégradation par un flou H et un bruit gaussien, le problème de minimisation est donc réduit au **modèle des moindres carrés**, défini par :

$$\hat{x} = \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \ \|Hx - z\|_2^2 \tag{2}$$

On pourra se limiter dans un premier temps au choix $H = Id$. Une fois la méthode implémentée et validée, on pourra introduire une dégradation de type flou et/ou décimation. Pour cela, vous pouvez utiliser la fonction fournies (cf section 6).

Attention. On veillera à choisir un signal de taille raisonnable ($N \leq 256$), afin que le temps de calcul reste raisonnable.

Pistes de réflexion :

- solution attendue ?
- solution analytique du problème (2)
- résolution numérique par descente de gradient du problème (2) pour un signal 1D z de votre choix (e.g. Heaviside, polynôme, etc.)
- comparaison des solutions obtenues par les deux méthodes (valeurs de la fonction de coût)
- influence du pas de descente de gradient
- influence du choix de l'initialisation
- limites de chacune des deux approches ?

4 Résolution du modèle de Tikhonov 1D

On souhaite étudier l'influence de la régularisation R. L'un des modèles les plus connus (utilisé par exemple pour la reconstruction tomographique) en traitement d'image est le modèle de Tikhonov, défini de la manière suivante :

$$\hat{x} = \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \ \|Hx - z\|_2^2 + \lambda \|\Gamma x\|_2^2 \tag{3}$$

où Γ modélise un opérateur de votre choix (identité, gradient, Laplacien...).

Pistes de réflexion :

- solution attendue ?
- solution analytique du problème (3)
- résolution numérique par descente de gradient du problème (3) pour le même signal 1D que dans la partie 3
- influence du choix de l'opérateur Γ
- comparaison des solutions obtenues par les deux méthodes (valeurs de la fonction de coût)
- apport de la régularisation ?
- influence du choix de λ

5 Résolution du modèle de Tikhonov 2D

L'extension aux images est presque immédiate si l'on considère x le vectorisé de l'image. Les seuls changements à effectuer concernent les matrices H et Γ (gradient 2D, Laplacien 2D). Attention cependant à ne pas choisir des images trop grandes : on rappelle que pour une image de taille $N \times N$, Γ est de taille $N^2 \times N^2$...
Pour pallier cela, on pourra utiliser les opérateurs D, L et H fournis, ainsi que leurs adjoints `Dadj`, `Ladj` et `Hadj`, plutôt que les matrices. Dans ce cas, les opérateurs s'appliquent à l'image **non vectorisée**.

Pistes de réflexion :

- résolution numérique par descente de gradient du problème (3) pour une image
- influence du choix de l'opérateur Γ
- influence du choix de λ
- commentaires

6 Index des fonctions

Matrices.

`H = matH(size(x),type,N)`	Crée la matrice H de flou de type `type` et de taille N
`D = matGamma(s,'gradient')`	Crée la matrice D de gradient (`s = size(x)`)
`L = matGamma(s,'laplacian')`	Crée la matrice L de Laplacien (`s = size(x)`)

Opérateurs (à privilégier en Octave).

`y = H(x,type)`	Calcule l'opération matricielle Hx (x **non** vectorisé)
`y = Hadj(x,type)`	Calcule l'opération matricielle $H^\top x$ (x **non** vectorisé)
`y = D(x)`	Calcule l'opération matricielle Dx (x **non** vectorisé)
`y = Dadj(x)`	Calcule l'opération matricielle $D^\top x$ (x **non** vectorisé)
`y = L(x)`	Calcule l'opération matricielle Lx (x **non** vectorisé)
`y = Ladj(x)`	Calcule l'opération matricielle $L^\top x$ (x **non** vectorisé)