

Méthodes variationnelles pour la Restauration d'Images

Arthur Leclaire



Semaine Images
5 Février 2024

Plan

Restauration d'Images

Descente de Gradient

- Interprétation du Gradient
- Descente de Gradient
- Différentiation automatique

Méthodes variationnelles pour l'imagerie

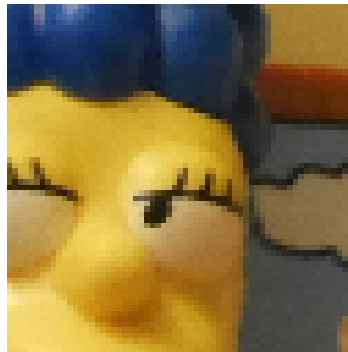
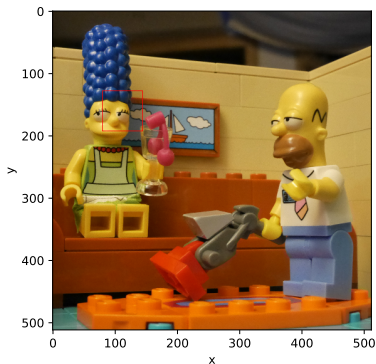
- Débruitage et Régularisations
- Problèmes inverses

Images numériques

Une image numérique est un tableau de nombres (souvent dans $\{0, \dots, 255\}$ ou dans $[0, 1]$)

$$u : \Omega \rightarrow \mathbf{R}^d$$

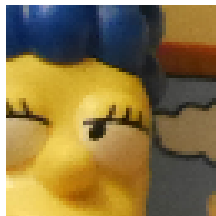
où $\Omega = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$ et où $d = 1$ (niveaux de gris) ou $d = 3$ (couleurs).



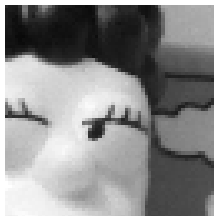
Gris v.s. Couleurs

Une image en niveau de gris s'écrit donc $u \in \mathbf{R}^{M \times N}$.

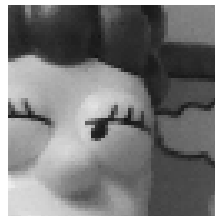
Une image en couleurs se décompose en $u = (u_R, u_G, u_B) \in \mathbf{R}^{M \times N \times 3}$.



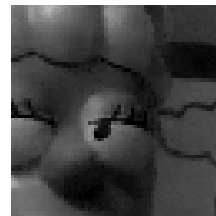
u



u_R



u_G



u_B

Dans la suite, on se concentrera sur les images de niveaux de gris.

Les méthodes expliquées aujourd'hui s'étendent à la couleur en travaillant sur les canaux séparément.

Acquisition et Dégradations

- Acquisition d'une image avec un procédé physique (objectif, capteur CCD)
- Plusieurs dégradations lors de l'acquisition :
 - Flou de défocalisation (lié à l'objectif)
 - Bruit photonique (lié au capteur)
 - Flou de bouger
 - Aberrations géométrique ou chromatiques
 - ...

Dans la suite, nous travaillerons avec un modèle simplifié

$$v = Au_0 + w$$

où

- u_0 est l'image propre (à retrouver)
- A est une application linéaire
- w un "bruit".

Bruit gaussien

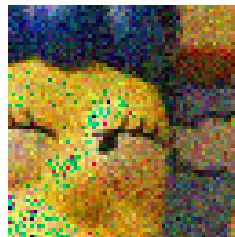
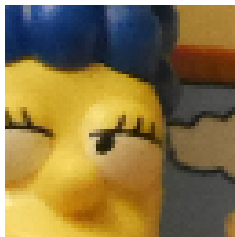
Considérons d'abord le cas $A = \text{Id}$ lié au problème de **débruitage**. On considère le modèle

$$v = u_0 + w$$

où w est un bruit blanc gaussien d'écart-type $\sigma > 0$. Cela signifie que

$$\forall (x, y) \in \Omega, \quad v(x, y) = u_0(x, y) + w(x, y)$$

où $w(x, y)$ est une réalisation de $\mathcal{N}(0, \sigma^2 I)$.



Question : Comment retrouver u_0 à partir de v ? (i.e. débruiter v)

Bruit gaussien

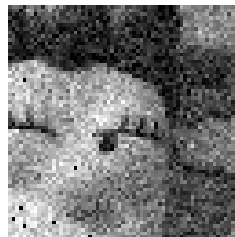
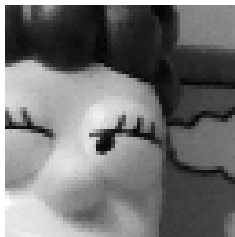
Considérons d'abord le cas $A = \text{Id}$ lié au problème de **débruitage**. On considère le modèle

$$v = u_0 + w$$

où w est un bruit blanc gaussien d'écart-type $\sigma > 0$. Cela signifie que

$$\forall (x, y) \in \Omega, \quad v(x, y) = u_0(x, y) + w(x, y)$$

où $w(x, y)$ est une réalisation de $\mathcal{N}(0, \sigma^2 I)$.



Question : Comment retrouver u_0 à partir de v ? (i.e. débruiter v)

Débruitage et Max de Vraisemblance

La vraisemblance du modèle de bruit gaussien s'écrit

$$p(v|u) = \frac{1}{(2\pi\sigma^2)^{mn/2}} \exp\left(-\frac{1}{2}\|v - u\|_2^2\right)$$

Maximiser la vraisemblance revient donc à $\underset{u}{\operatorname{Argmax}} p(v|u) = v$ ✗ Aucun effet !

Si l'on dispose de plusieurs réalisations indépendantes v_1, \dots, v_n ,

$$p(v_1, \dots, v_n|u) = \frac{1}{(2\pi\sigma^2)^{mn/2}} \exp\left(-\frac{1}{2} \sum_{k=1}^N \|v_k - u\|_2^2\right)$$

Maximiser la vraisemblance revient alors à

$$\underset{u}{\operatorname{Argmax}} p(v_1, \dots, v_n|u) = \underset{u}{\operatorname{Argmin}} \sum_{k=1}^N \|v_k - u\|_2^2 = \frac{1}{N} \sum_{k=1}^N v_k.$$

✗ Oui, mais en pratique, on ne dispose généralement que d'une réalisation !

Filtrage et Convolution

Soit k est un “noyau” défini sur un petit domaine $\omega = \{-s, \dots, s\}^2$

$$k * u(x, y) = \sum_{|x'| \leq s, |y'| \leq s} k(x', y') u(x - x', y - y')$$

i.e.

$$k * u(\mathbf{x}) = \sum_{\mathbf{x}' \in \omega} k(\mathbf{x}') u(\mathbf{x} - \mathbf{x}')$$

On dit souvent que l'on **filtre** u avec le noyau k . C'est un **filtrage linéaire**.

Si $k : \omega \rightarrow \mathbf{R}_+$ est tel que $\sum_{\mathbf{x}' \in \omega} k(\mathbf{x}') = 1$, $k * u(\mathbf{x})$ est une moyenne de u autour de \mathbf{x} .

Notez bien que si \mathbf{x} est au bord de Ω , il faut adopter une “condition de bord” pour calculer $u(\mathbf{x} - \mathbf{x}')$.

Exemple : Les dérivées discrétisées de u définies par
$$\begin{cases} \partial_1 u(x, y) = u(x + 1, y) - u(x, y) \\ \partial_2 u(x, y) = u(x, y + 1) - u(x, y) \end{cases}$$

sont des convolutions : $\partial_1 u = k_1 * u$ où k_1 qui vaut 0 partout sauf en deux points $\begin{cases} k_1(0, 0) = -1, \\ k_1(-1, 0) = 1. \end{cases}$

Flou et déflouage

- Un flou uniforme peut être modélisé par un opérateur de convolution $Au = k * u$
- Selon la forme du noyau k , on distingue plusieurs types de flou
- **Le problème du défloutage** (“non-aveugle”) consiste à retrouver u_0 à partir de

$$v = k * u_0 + w.$$

- On n'abordera pas aujourd'hui le problème de déflouage *aveugle* i.e. déflouage sans connaître k .



Flou isotrope



Flou de bouger



Image originale

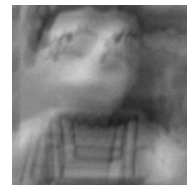


Image floutée

Exemple de flou de bouger

Transformée de Fourier discrète

Définition

La transformée de Fourier discrète (TFD) 2D est définie pour tout $u \in \mathbb{C}^\Omega$ par

$$\forall (\xi, \zeta) \in \mathbb{Z}^2, \quad \hat{u}(\xi, \zeta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} u(x, y) \exp \left(-2i\pi \left(\frac{\xi x}{M} + \frac{\zeta y}{N} \right) \right).$$

Remarque : La TFD étend implicitement les images à \mathbb{Z}^2 par (M, N) -périodicité.

Théorème

$$\forall (x, y) \in \Omega, \quad u(x, y) = \frac{1}{MN} \sum_{\xi=0}^{M-1} \sum_{\zeta=0}^{N-1} \hat{u}(\xi, \zeta) \exp \left(2i\pi \left(\frac{\xi x}{M} + \frac{\zeta y}{N} \right) \right).$$

De plus, pour toute $u \in \mathbb{C}^\Omega$, on a $\|\hat{u}\|^2 = MN\|u\|^2$.

Remarque : La TFD \mathcal{F} et son inverse \mathcal{F}^{-1} se calculent avec $\mathcal{O}(MN \log MN)$ opérations !
C'est l'algorithme *Fast Fourier Transform* (FFT).

Transformée de Fourier discrète

On rappelle que $\Omega = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$.

Définition

Soient $u, v \in \mathbb{C}^{\mathbb{Z}^2}$ qui sont (M, N) -périodiques. La convolution périodique de u et v est définie par

$$\forall \mathbf{x} \in \mathbb{Z}^2, \quad u * v(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega} u(\mathbf{x} - \mathbf{y})v(\mathbf{y}).$$

Théorème

Pour $u, v \in \mathbb{C}^{\mathbb{Z}^2}$ (M, N) -périodiques, on a $\widehat{u * v} = \hat{u}\hat{v}$.

Conséquence : La convolution $u * v = \mathcal{F}^{-1}(\hat{u}\hat{v})$ se calcule en $\mathcal{O}(MN \log MN)$ opérations.

Cela permet aussi de calculer des convolutions non-périodiques en prolongeant par zéro sur un domaine deux fois plus gros (*zero-padding*), puis en restreignant le résultat.

Débruitage par Filtrage linéaire

Definition

Soient $v \in \mathbb{C}^\Omega$ et $\lambda > 0$.

On définit un débruitage de v par filtrage linéaire en posant

$$\forall (\xi, \zeta) \in \Omega, \quad \hat{u}(\xi, \zeta) = \frac{\hat{v}(\xi, \zeta)}{1 + 2\lambda \hat{L}(\xi, \zeta)}$$

où $\hat{L}(\xi, \zeta) = 4 \left(\sin^2 \left(\frac{\pi \xi}{M} \right) + \sin^2 \left(\frac{\pi \zeta}{N} \right) \right)$.

L'image débruitée u s'écrit donc $h_\lambda * v$ où h_λ est défini par sa TFD :

$$\forall (\xi, \zeta) \in \Omega, \quad \hat{h}_\lambda(\xi, \zeta) = \frac{1}{1 + 2\lambda \hat{L}(\xi, \zeta)}$$

Remarque : On utilise parfois aussi $\hat{L}(\xi, \zeta) = 4\pi^2 \left(\left(\frac{\xi}{M} \right)^2 + \left(\frac{\zeta}{N} \right)^2 \right)$.

Critère de Qualité

Il existe plusieurs façons de mesurer la qualité de la restauration v en connaissant l'image propre u_0 .
Les deux plus simples sont le MSE pour *Mean Square Error* :

$$\text{MSE} = \sqrt{\frac{1}{MN} \|u_0 - v\|^2} = \left(\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (u_0(x, y) - v(x, y))^2 \right)^{1/2}$$

et le *Peak Signal to Noise Ratio* :

$$\text{PSNR} = 20 \log_{10} \left(\frac{\text{MAX}}{\text{MSE}} \right) .$$

La valeur MAX doit être ajustée suivant la valeur maximale atteignable par l'image (e.g. 1 ou 255).

Super-Résolution

La **super-résolution** consiste à chercher une autre version de v à plus haute résolution.

Ceci est un problème inverse où intervient un sous-échantillonnage de pas $s \in \mathbb{N}^*$:

$$u_{\downarrow s}(x, y) = u(sx, sy).$$

En pratique, on applique un filtre (*anti-aliasing*) avant de sous-échantillonner.

L'opérateur A s'écrit alors

$$Au = (k * u)_{\downarrow s}.$$

Le problème de **super-résolution** consiste à retrouver u_0 à partir de

$$v = (k * u)_{\downarrow s} + w.$$

Cette nouvelle image v est définie sur une grille s fois plus petite.

Inpainting

Un autre problème célèbre est l'*inpainting*, consistant à combler un trou dans une image



L'opérateur de dégradation s'écrit ici

$$Au = u1_{\omega}$$

où $\omega \subset \Omega$ est l'ensemble des pixels connus (et $\Omega \setminus \omega$ le masque).

Problème inverse

On parle de **problème inverse** lorsque l'on essaie de récupérer u_0 à partir de

$$v = Au_0 + w.$$

On dit que le problème est *mal posé* lorsque A n'est pas inversible, ou que son inverse est *instable*.

- **Exemple** : Pour le déflouage, $Au = k * u$, on peut inverser dans le domaine de Fourier :

$$u = \mathcal{F}^{-1} \left(\frac{\hat{v}}{\hat{k}} \right) = \mathcal{F}^{-1} \left(\hat{u}_0 + \frac{\hat{w}}{\hat{k}} \right) \longrightarrow \text{mais le bruit explose !}$$

Lorsque le problème est mal posé, il peut y avoir plusieurs solutions, ou des solutions aberrantes.

Il est utile de rajouter de l'information *a priori*, par exemple supposer que u est "régulière".

On utilisera donc des fonctions mesurant la régularité de u via les dérivées, par exemple

$$R(u) = \sum_{\mathbf{x} \in \Omega} |\partial_1 u(\mathbf{x})|^2 + |\partial_2 u(\mathbf{x})|^2.$$

Restauration par Optimisation

Ainsi, on cherchera à minimiser la fonction

$$F(u) = \frac{1}{2} \|Au - v\|_2^2 + \lambda R(u)$$

où $R(u)$ est une mesure de la régularité de u , et $\lambda \geq 0$ un paramètre.

On a donc un problème d'optimisation $\underset{u \in \mathbf{R}^\Omega}{\text{Argmin}} F(u)$ en très grande dimension.

(Pour une image 1024×1024 , plus d'un million de variables !)

Il nous faut donc des algorithmes efficaces pour résoudre ce type de problèmes.

La vitesse de convergence repose crucialement sur

la **régularité** et la **convexité** de F .

Plan

Restauration d'Images

Descente de Gradient

Interprétation du Gradient

Descente de Gradient

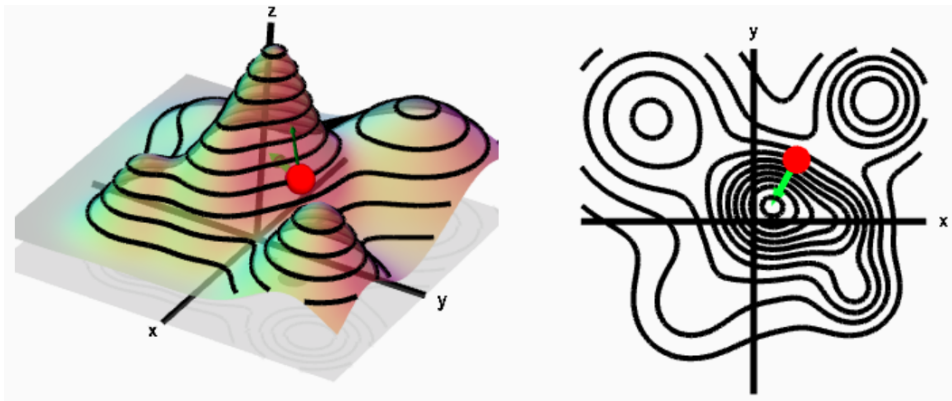
Différentiation automatique

Méthodes variationnelles pour l'imagerie

Débruitage et Régularisations

Problèmes inverses

D'abord, un dessin



https://mathinsight.org/directional_derivative_gradient_introduction

La plus grande pente

Soit $f : \mathbf{R}^d \rightarrow \mathbf{R}$ différentiable et soit $x \in \mathbf{R}^d$ fixé. On rappelle que

$$\nabla f(x) = (\partial_1 f(x), \partial_2 f(x), \dots, \partial_d f(x))^T.$$

Pour $v \in \mathbf{R}^d$ de norme 1 et $t \in \mathbf{R}$, on peut écrire

$$f(x + tv) = f(x) + t \nabla f(x) \cdot v + o(t).$$

L'inégalité de Cauchy-Schwarz donne

$$|\nabla f(x) \cdot v| \leq \|\nabla f(x)\|$$

avec égalité pour v colinéaire à $\nabla f(x)$.

Si $\nabla f(x) \neq 0$, cela donne les directions de plus grande pente

$$f(x + t \nabla f(x)) = f(x) + t \|\nabla f(x)\|^2 + o(t)$$

$$f(x - t \nabla f(x)) = f(x) - t \|\nabla f(x)\|^2 + o(t)$$

Lemme de Descente

Soit $f : \mathbf{R}^d \rightarrow \mathbf{R}$ différentiable à gradient L -Lipschitz (et donc f est \mathcal{C}^1), pour tous $x, y \in \mathbf{R}^d$,

$$\begin{aligned} f(y) &= f(x) + \int_0^1 \nabla f(x + t(y - x)) \cdot (y - x) dt \\ &= f(x) + \nabla f(x) \cdot (y - x) + \int_0^1 (\nabla f(x + t(y - x)) - \nabla f(x)) \cdot (y - x) dt \\ &\leq f(x) + \nabla f(x) \cdot (y - x) + \int_0^1 \|\nabla f(x + t(y - x)) - \nabla f(x)\| \|y - x\| dt \\ &\leq f(x) + \nabla f(x) \cdot (y - x) + \int_0^1 Lt \|y - x\|^2 dt \\ &\leq f(x) + \nabla f(x) \cdot (y - x) + \frac{L}{2} \|y - x\|^2. \end{aligned}$$

Conséquence : Si en plus on choisit $\tau \in [0, \frac{2}{L})$,

$$f(x - \tau \nabla f(x)) \leq f(x) - \tau \left(1 - \frac{\tau L}{2}\right) \|\nabla f(x)\|^2 \leq f(x).$$

Algorithme de Descente de Gradient

On considère ici la méthode de descente de gradient.

$$x_{n+1} = x_n - \tau_n \nabla f(x_n) ,$$

où $\tau_n > 0$ est une suite de pas.

- Lorsque $\tau_n = \tau$ est constante, on parle de descente de gradient à pas fixe.
- Lorsque l'on choisit

$$\tau_n \in \underset{t \in \mathbf{R}}{\operatorname{Argmin}} f(x_n - t \nabla F(x_n))$$

on parle de descente de gradient à pas optimal.

Le lemme précédent donne, pour f différentiable à gradient L -Lipschitz et $\tau < \frac{2}{L}$,

$$f(x_{n+1}) \leq f(x_n)$$

Ainsi, si f est minorée, on sait déjà que $f(x_n)$ converge.

Convexité et Minimum

On dit que $f : \mathbf{R}^d \rightarrow \mathbf{R}$ est convexe si pour tous $x, y \in \mathbf{R}^d$,

$$\forall t \in (0, 1), \quad f((1 - t)x + ty) \leq (1 - t)f(x) + tf(y).$$

On dit qu'elle est strictement convexe si l'on a l'inégalité stricte.

Si f est convexe et différentiable, on peut montrer que pour tous $x, y \in \mathbf{R}^d$,

$$f(y) \geq \nabla f(x) \cdot (y - x).$$

Conséquence : Si f est une fonction convexe différentiable, alors

$$x \in \text{Argmin } f \iff \nabla f(x) = 0.$$

Le minimum est unique dès que f est strictement convexe.

Forte Convexité

On dit que f est α -convexe (avec $\alpha > 0$) si $f - \frac{\alpha}{2} \|\cdot\|^2$ est convexe.

Lorsque $\alpha > 0$, on dit que f est **fortement convexe**.

Remarque : La convexité et la constante de Lipschitz du gradient se lisent sur la hessienne.

Si $A, B \in \mathbf{R}^{d \times d}$ sont symétriques, on note $A \succeq B$ si $B - A$ est une matrice positive, c'est-à-dire

$$\forall x \in \mathbf{R}^d, \quad Ax \cdot x \geq Bx \cdot x.$$

Ainsi, pour $f : \mathbf{R}^d \rightarrow \mathbf{R}$ de classe \mathcal{C}^2 ,

∇f est L -lipschitz si et seulement si $\forall x \in \mathbf{R}^d, \quad -L\text{Id} \preceq \nabla^2 f(x) \preceq L\text{Id}$.
i.e. les valeurs propres de $\nabla^2 f(x)$ sont toutes de valeur absolue $\leq L$.

f est α -convexe si et seulement si $\nabla^2 f \succeq \alpha\text{Id}$
i.e. les valeurs propres de $\nabla^2 f(x)$ sont toutes $\geq \alpha$.

Garanties de Convergence, Cas convexe

Théorème

Soit $f : \mathbf{R}^d \rightarrow \mathbf{R}$ convexe différentiable telle que ∇f est L -lipschitzienne. Supposons que f atteint son infimum en un point de \mathbf{R}^d . Soit $\tau \in (0, \frac{2}{L})$, $x_0 \in \mathbf{R}^d$ et (x_n) la suite définie par récurrence par

$$x_{n+1} = x_n - \tau \nabla f(x_n) .$$

Alors (x_n) converge vers un certain élément de $\text{Argmin } f$.

Théorème

Soit $f : \mathbf{R}^d \rightarrow \mathbf{R}$ différentiable α -fortement convexe et à gradient L -Lipschitz.

Alors f atteint son infimum en un unique $x_* \in \text{Argmin } f$, et si $\tau \leq \frac{1}{L}$, on a

$$\|x_n - x_*\|^2 \leq (1 - \tau\alpha)^n \|x_0 - x_*\|^2 .$$

Différentiation automatique

La librairie `Pytorch` permet de calculer des gradients de façon automatique.

```
1 import torch
2
3 def f(x):
4     return 1/3*x[0]**3 - x[1]**2
5
6 x = torch.tensor([2., 3.], requires_grad=True)
7 fx = f(x)
8 fx.backward()
9 print(x.grad)
10
```

tensor([4., -6.])

https://pytorch.org/tutorials/beginner/blitz/autograd_tutorial.html

Descente de gradient automatique

Mieux encore, elle permet aussi d'implémenter la descente de gradient de façon automatique.

```
1 def optim(f,niter=1000,lr=0.1):
2     u = torch.randn(M,N, requires_grad=True)
3     optimu = torch.optim.SGD([u], lr=lr)
4     losslist = []
5     for it in range(niter):
6         loss = f(u)
7         losslist.append(loss.detach())
8         optimu.zero_grad()
9         loss.backward()
10        optimu.step()
11    return np.array(u.detach()),losslist
```

Et maintenant, en route !

Les trois exercices du TP peuvent être téléchargés depuis ce dépôt git :

`https://github.com/leclairearthur/imageweek`

ou alors sur l'espace partage :

`https://partage.imt.fr/index.php/s/CiGPj2FBqKdEF8p`

Plan

Restauration d'Images

Descente de Gradient

Interprétation du Gradient

Descente de Gradient

Différentiation automatique

Méthodes variationnelles pour l'imagerie

Débruitage et Régularisations

Problèmes inverses

Restauration d'Images par Optimisation

Pour résoudre le problème inverse $v = Au_0 + w$, on cherche à minimiser

$$F(u) = f(u) + g(u)$$

où

- $f(u) = \frac{1}{2} \|Au - v\|_2^2$ est appelée *attache aux données*
- $g(u) = \lambda R(u)$ est une fonction de régularisation.

On notera

$$\inf F = \inf \{ F(u) , u \in \mathbf{R}^d \}$$

et

$$\text{Argmin } F = \{ u \in \mathbf{R}^d , F(u) = \inf F \}.$$

Lorsque f et g sont différentiables et convexes, on peut minimiser F par descente de gradient.

Pour régler le pas τ , il est utile de connaître la constante de Lipschitz de ∇F .

Un calcul fondamental

Mais il faut aussi savoir calculer le gradient de F ...

Pour $A \in \mathbf{R}^{m \times d}$, calculons le gradient de

$$f(u) = \frac{1}{2} \|Au - y\|_2^2.$$

Le gradient de f s'écrit...

Un calcul fondamental

Mais il faut aussi savoir calculer le gradient de F ...

Pour $A \in \mathbf{R}^{m \times d}$, calculons le gradient de

$$f(u) = \frac{1}{2} \|Au - y\|_2^2.$$

Le gradient de f s'écrit...

$$\nabla f(u) = A^T (Au - y).$$

Un calcul fondamental

Mais il faut aussi savoir calculer le gradient de F ...

Pour $A \in \mathbf{R}^{m \times d}$, calculons le gradient de

$$f(u) = \frac{1}{2} \|Au - y\|_2^2.$$

Le gradient de f s'écrit...

$$\nabla f(u) = A^T (Au - y).$$

La constante de Lipschitz de ∇f est donc...

Un calcul fondamental

Mais il faut aussi savoir calculer le gradient de F ...

Pour $A \in \mathbf{R}^{m \times d}$, calculons le gradient de

$$f(u) = \frac{1}{2} \|Au - y\|_2^2.$$

Le gradient de f s'écrit...

$$\nabla f(u) = A^T (Au - y).$$

La constante de Lipschitz de ∇f est donc... la norme d'opérateur $\|A^T A\|$.

Un calcul fondamental

Mais il faut aussi savoir calculer le gradient de $F...$

Pour $A \in \mathbf{R}^{m \times d}$, calculons le gradient de

$$f(u) = \frac{1}{2} \|Au - y\|_2^2.$$

Le gradient de f s'écrit...

$$\nabla f(u) = A^T (Au - y).$$

La constante de Lipschitz de ∇f est donc... la norme d'opérateur $\|A^T A\|$.

Selon la forme de A , on pourra calculer plus ou moins facilement cette norme d'opérateur.

Un calcul fondamental

Mais il faut aussi savoir calculer le gradient de $F...$

Pour $A \in \mathbf{R}^{m \times d}$, calculons le gradient de

$$f(u) = \frac{1}{2} \|Au - y\|_2^2.$$

Le gradient de f s'écrit...

$$\nabla f(u) = A^T (Au - y).$$

La constante de Lipschitz de ∇f est donc... la norme d'opérateur $\|A^T A\|$.

Selon la forme de A , on pourra calculer plus ou moins facilement cette norme d'opérateur.

Exemple : Calculons A^T lorsque A est un opérateur de convolution périodique $Au = k * u$.

Régularisations Explicites

On rappelle que les dérivées discrétisées de u sont définies par

$$\nabla u(x, y) = \begin{pmatrix} \partial_1 u(x, y) \\ \partial_2 u(x, y) \end{pmatrix} \quad \text{avec} \quad \begin{cases} \partial_1 u(x, y) = u(x+1, y) - u(x, y) \\ \partial_2 u(x, y) = u(x, y+1) - u(x, y) \end{cases}.$$

Une première régularisation possible est la **régularisation de Tychonov**

$$\|\nabla u\|_2^2 = \sum_{\mathbf{x} \in \Omega} \|\nabla u(\mathbf{x})\|^2 = \sum_{\mathbf{x} \in \Omega} |\partial_1 u(\mathbf{x})|^2 + |\partial_2 u(\mathbf{x})|^2.$$

Une deuxième est la régularisation par la **variation totale** :

$$\text{TV}(u) = \|\nabla u\|_1 = \sum_{\mathbf{x} \in \Omega} \|\nabla u(\mathbf{x})\| = \sum_{\mathbf{x} \in \Omega} \sqrt{|\partial_1 u(\mathbf{x})|^2 + |\partial_2 u(\mathbf{x})|^2}.$$

On verra que ces régularisations mènent à des résultats très différents.

Retour au problème de débruitage

On cherche à minimiser

$$F(u) = \frac{1}{2} \|u - v\|^2 + \lambda R(u)$$

où R est une régularisation.

Considérons d'abord la régularisation de Tychonov $R(u) = \|\nabla u\|_2^2$.

Retour au problème de débruitage

On cherche à minimiser

$$F(u) = \frac{1}{2} \|u - v\|^2 + \lambda R(u)$$

où R est une régularisation.

Considérons d'abord la régularisation de Tychonov $R(u) = \|\nabla u\|_2^2$.

Le calcul précédent donne $\nabla R(u) = 2\nabla^T \nabla u$.

Retour au problème de débruitage

On cherche à minimiser

$$F(u) = \frac{1}{2} \|u - v\|^2 + \lambda R(u)$$

où R est une régularisation.

Considérons d'abord la régularisation de Tychonov $R(u) = \|\nabla u\|_2^2$.

Le calcul précédent donne $\nabla R(u) = 2\nabla^T \nabla u$. Comme F est convexe,

$$u \in \text{Argmin } F \iff \nabla F(u) = 0 \iff u - v + 2\lambda \nabla^T \nabla u = 0 \iff u = (I + 2\lambda \nabla^T \nabla)^{-1} v$$

Retour au problème de débruitage

On cherche à minimiser

$$F(u) = \frac{1}{2} \|u - v\|^2 + \lambda R(u)$$

où R est une régularisation.

Considérons d'abord la régularisation de Tychonov $R(u) = \|\nabla u\|_2^2$.

Le calcul précédent donne $\nabla R(u) = 2\nabla^T \nabla u$. Comme F est convexe,

$$u \in \text{Argmin } F \iff \nabla F(u) = 0 \iff u - v + 2\lambda \nabla^T \nabla u = 0 \iff u = (I + 2\lambda \nabla^T \nabla)^{-1} v$$

Pour $p : \Omega \rightarrow \mathbf{R}^2$, $\nabla^T p$ est donné par

$$\nabla^T p(x, y) = p_1(x - 1, y) - p_1(x, y) + p_2(x, y - 1) - p_2(x, y).$$

En fait, $\text{div}(p) := -\nabla^T p$ est une divergence discrète et $-\nabla^T \nabla u =: \Delta u$ est un laplacien discret.

Solution explicite

Théorème

Soient $v \in \mathbb{C}^\Omega$ et $\lambda > 0$. La fonction $F : \mathbb{C}^\Omega \rightarrow \mathbf{R}_+$ définie par

$$\forall u \in \mathbb{C}^\Omega, \quad F(u) = \frac{1}{2} \|u - v\|_2^2 + \lambda \|\nabla u\|_2^2$$

atteint son minimum en un unique point u_* . De plus, ce u_* peut être calculé explicitement dans le domaine de Fourier par

$$\forall (\xi, \zeta) \in \Omega, \quad \hat{u}_*(\xi, \zeta) = \frac{\hat{v}(\xi, \zeta)}{1 + 2\lambda \hat{L}(\xi, \zeta)}$$

où $\hat{L}(\xi, \zeta) = |\hat{k}_1(\xi, \zeta)|^2 + |\hat{k}_2(\xi, \zeta)|^2 = 4 \left(\sin^2 \left(\frac{\pi \xi}{M} \right) + \sin^2 \left(\frac{\pi \zeta}{N} \right) \right)$.

Lien avec un modèle d'évolution

La descente de gradient sur

$$F(u) = \frac{1}{2} \|u - v\|_2^2 + \lambda \|\nabla u\|_2^2$$

s'écrit

$$u_{n+1} - u_n = -\tau(u_n - v) + 2\lambda\tau\Delta u_n .$$

La suite (u_n) converge vers u_* dès que $\tau < \frac{2}{L}$ où $L = \|I + 2\lambda\nabla^T\nabla\| = 1 + 16\lambda$ (voir juste après).

Lien avec un modèle d'évolution

La descente de gradient sur

$$F(u) = \frac{1}{2} \|u - v\|_2^2 + \lambda \|\nabla u\|_2^2$$

s'écrit

$$u_{n+1} - u_n = -\tau(u_n - v) + 2\lambda\tau\Delta u_n.$$

La suite (u_n) converge vers u_* dès que $\tau < \frac{2}{L}$ où $L = \|I + 2\lambda\nabla^T\nabla\| = 1 + 16\lambda$ (voir juste après).

Que se passe-t-il si l'on oublie l'attache aux données ?

La descente de gradient sur $u \mapsto \|\nabla u\|_2^2$ s'écrit

$$u_{n+1} - u_n = 2\tau\Delta u_n$$

C'est une discrétisation de l'équation de la chaleur $\partial_t u = c\Delta u$ avec condition initiale u_0 .

Le paramètre $\tau > 0$ joue le rôle d'un pas de temps.

Pourquoi méthodes "variationnelles" ?

L'analyse des équations aux dérivées partielles (EDP) vise à résoudre des problèmes comme

$$\begin{cases} -\Delta u + u = v & \text{dans } \Omega, \\ u = 0 & \text{sur } \partial\Omega. \end{cases}$$

La méthode variationnelle consiste à trouver une "énergie" à laquelle est associée l'EDP, ici :

$$F(u) = \int_{\Omega} |u - v|^2 + \int_{\Omega} |\nabla u|^2$$

et à minimiser F sur un ensemble $H_0(\Omega)$ de fonctions qui "s'annulent" sur le bord de Ω .

Pourquoi méthodes "variationnelles" ?

L'analyse des équations aux dérivées partielles (EDP) vise à résoudre des problèmes comme

$$\begin{cases} -\Delta u + u = v & \text{dans } \Omega, \\ u = 0 & \text{sur } \partial\Omega. \end{cases}$$

La méthode variationnelle consiste à trouver une "énergie" à laquelle est associée l'EDP, ici :

$$F(u) = \int_{\Omega} |u - v|^2 + \int_{\Omega} |\nabla u|^2$$

et à minimiser F sur un ensemble $H_0(\Omega)$ de fonctions qui "s'annulent" sur le bord de Ω .

La **méthode variationnelle** consiste à que pour une petite variation arbitraire δu

$$F(u + \delta u) \geq F(u).$$

En faisant le développement limité à l'ordre 1, on trouve, là encore, $\nabla F(u) = 0$.

Comme dans le cas discret, cela donne aussi $u - v - \Delta u = 0$ i.e. $-\Delta u + u = v$.

The Lipschitz constant...

Calculons la norme d'opérateur de $\nabla^T \nabla$:

$$\|\nabla^T \nabla\| = \sup_{\|u\|_2 \leq 1, \|v\|_2 \leq 1} \langle \nabla^T \nabla u, v \rangle = \sup_{\|u\|_2 \leq 1, \|v\|_2 \leq 1} \langle \nabla u, \nabla v \rangle = \sup_{\|u\|_2 \leq 1} \|\nabla u\|_2^2 = \sup_{\|u\|_2 \leq 1} \|\partial_1 u\|_2^2 + \|\partial_2 u\|_2^2.$$

De plus, la formule de Parseval donne que

$$\|\partial_1 u\|_2^2 + \|\partial_2 u\|_2^2 = \frac{1}{MN} \sum_{(\xi, \zeta) \in \Omega} 4 \left(\sin^2 \left(\frac{\pi \xi}{M} \right) + \sin^2 \left(\frac{\pi \zeta}{N} \right) \right) |\hat{u}(\xi, \zeta)|^2 \leq 8 \|u\|_2^2.$$

Ainsi, $\|\nabla^T \nabla\| \leq 8$.

Exemples de Débruitage



Image Bruitée
PSNR = 19.93



Régularisation Tychonov
PSNR = 25.89

TV lissée

Que faire maintenant si l'on veut minimiser

$$F(u) = \frac{1}{2} \|u - v\|_2^2 + \lambda \text{TV}(u).$$

Problème : La variation totale n'est pas différentiable !

Une Solution simple : Considérer une variante lissée ! Soit $\varepsilon > 0$. On pose

$$\text{TV}_\varepsilon(u) = \sum_{(x,y) \in \Omega} \sqrt{\varepsilon^2 + \partial_1 u(x,y)^2 + \partial_2 u(x,y)^2}.$$

On peut voir que

$$\nabla \text{TV}_\varepsilon(u) = \nabla^T \left(\frac{\nabla u}{\sqrt{\varepsilon^2 + \|\nabla u\|_2^2}} \right).$$

Et on peut même calculer que $\nabla \text{TV}_\varepsilon$ est $\frac{8}{\varepsilon}$ -Lipschitz.

Ceci permet de minimiser F par descente de gradient avec pas $\tau < \frac{2}{1 + \frac{8\lambda}{\varepsilon}}$.

Exemples de Débruitage



Image Bruitée
PSNR = 19.93



Régularisation Tychonov
PSNR = 25.89



Régularisation TV_{ϵ}
PSNR = 27.21

Optimisation pour les Problèmes inverses

Pour résoudre le problème inverse $v = Au_0 + w$, on peut donc minimiser

$$F(u) = f(u) + g(u)$$

avec $f(u) = \frac{1}{2}\|Au - v\|^2$ et $g(u) = \lambda R(u)$.

Pour les régularisations précédentes $R(u) = \|Bu\|_2^2$, F est bien convexe et différentiable.

Ceci permet de minimiser F par descente de gradient, en prenant $\tau < \frac{2}{L}$ où $L = \|A^T A + 2B^T B\|$.

- Pour $Au = k * u$, $A^T Au = \mathcal{F}^{-1}(|\hat{k}|^2 \hat{u})$.
- Pour $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ et $\|A\| = 1$.

Optimisation pour les Problèmes inverses

Pour résoudre le problème inverse $v = Au_0 + w$, on peut donc minimiser

$$F(u) = f(u) + g(u)$$

avec $f(u) = \frac{1}{2}\|Au - v\|^2$ et $g(u) = \lambda R(u)$.

Pour les régularisations précédentes $R(u) = \|Bu\|_2^2$, F est bien convexe et différentiable.

Ceci permet de minimiser F par descente de gradient, en prenant $\tau < \frac{2}{L}$ où $L = \|A^T A + 2B^T B\|$.

- Pour $Au = k * u$, $A^T A u = \mathcal{F}^{-1}(|\hat{k}|^2 \hat{u})$.
- Pour $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ et $\|A\| = 1$.

Good news : En utilisant la différentiation automatique, il vous suffit de programmer la $F(u)$.

Optimisation pour les Problèmes inverses

Pour résoudre le problème inverse $v = Au_0 + w$, on peut donc minimiser

$$F(u) = f(u) + g(u)$$

avec $f(u) = \frac{1}{2}\|Au - v\|^2$ et $g(u) = \lambda R(u)$.

Pour les régularisations précédentes $R(u) = \|Bu\|_2^2$, F est bien convexe et différentiable.

Ceci permet de minimiser F par descente de gradient, en prenant $\tau < \frac{2}{L}$ où $L = \|A^T A + 2B^T B\|$.

- Pour $Au = k * u$, $A^T A u = \mathcal{F}^{-1}(|\hat{k}|^2 \hat{u})$.
- Pour $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ et $\|A\| = 1$.

Good news : En utilisant la différentiation automatique, il vous suffit de programmer la $F(u)$.

Mais ! pour ne pas avoir de problème de stabilité, il faut comprendre ce qu'elle fait.
(Par exemple connaître la constante de Lipschitz du gradient L).

Descente de gradient projetée

Imaginons que l'on veuille contraindre la solution à être dans un convexe fermé $C \subset \mathbf{R}^d$.

$$\underset{u \in C}{\operatorname{Argmin}} F(u)$$

Pour cela, on peut utiliser la projection orthogonale sur le convexe fermé $p_C : \mathbf{R}^d \rightarrow C$.

Théorème

Soit $f : \mathbf{R}^d \rightarrow \mathbf{R}$ convexe différentiable telle que ∇f est L -lipschitzienne.

Soit $C \subset \mathbf{R}^d$ convexe fermé. Supposons qu'il existe $x \in C$ tel que $f(x) = \inf_C f$.

Soit $\tau \in (0, \frac{2}{L})$, $x_0 \in \mathbf{R}^d$ et (x_n) la suite définie par récurrence par

$$x_{n+1} = p_C(x_n - \tau \nabla f(x_n)) .$$

Alors (x_n) converge vers un élément de $\operatorname{Argmin}_C f$.

Exemple : pour l'inpainting, on s'intéresse au problème inverse sans bruit $v = Au$.

Dans ce cas, on minimise uniquement le terme de régularisation sous contrainte :

$$\min_{v=Au} R(u).$$

Conclusion

- De nombreux problèmes d'imagerie se formulent via des problèmes d'optimisation, potentiellement très compliqués.
- Si calculable, la descente de gradient donne une première méthode simple d'optimisation.
- Mais en pratique, c'est parfois plus compliqué.
Exemple : si l'on veut minimiser la variation totale (non-différentiable), utiliser la dualité convexe.
→ algorithme "primal-dual" [Chambolle and Pock, 2011].
- Les régularisations liées à des réseaux de neurones profonds sont des fonctions non-convexes.
Mais des méthodes spécifiques ont été conçues [Hurault et al., 2021].

MERCI POUR VOTRE ATTENTION !

References I



Chambolle, A. and Pock, T. (2011).

A first-order primal-dual algorithm for convex problems with applications to imaging.

Journal of Mathematical Imaging and Vision, 40(1) :120–145.



Hurault, S., Leclaire, A., and Papadakis, N. (2021).

Gradient step denoiser for convergent plug-and-play.

In *International Conference on Learning Representations*.