Variational Methods for Image Restoration

Arthur Leclaire

TELECOM Paris

IP PARIS

MVA Introduction à l'Imagerie Numérique
November, 5th, 2025

Today

- We will discuss imaging inverse problems.
- We will recall classical (simple) tools for solving inverse problems.
  In particular we will recall simple regularization techniques (Tychonov, smoothTV)
- We will discuss quantitative evaluation of image restoration.

Plan

Imaging Inverse Problems

Optimization for Inverse Problems

Metrics for Inverse Problems

Inverse problem with additive noise:

$$v = \mathcal{A}u_0 + w$$

where

- $u_0 \in \mathbf{R}^d$ is the clean image to recover
- $\mathcal{A} : \mathbf{R}^d \to \mathbf{R}^m$
- w is a noise

In many cases, the degradation operator $\mathcal{A}$ can be approximated with a linear operator A, and the noise model w is assumed to be Gaussian.

But, there are also inverse problems with non-linear $\mathcal{A}$ and non-Gaussian noise (e.g. Poisson noise).

Classical Inverse Problems

Application	Forward model	Notes
Denoising [58]	$A = I$	I is the identity matrix
Deconvolution [58, 59]	$\mathcal{A}(\boldsymbol{x}) = \boldsymbol{h} * \boldsymbol{x}$	$\boldsymbol{h}$ is a known blur kernel and $*$ denotes convolution. When $\boldsymbol{h}$ is unknown the reconstruction problem is known as blind deconvolution.
Superresolution [60, 61]	$A = SB$	S is a subsampling operator (identity matrix with missing rows) and B is a blurring operator cooresponding to convolution with a blur kernel
Inpainting [62]	$A = S$	S is a diagonal matrix where $S_{i,i} = 1$ for the pixels that are sampled and $S_{i,i} = 0$ for the pixels that are not.
Compressive Sensing [63, 64]	$A = SF$ or $A =$ Gaussian or Bernoulli ensemble	S is a subsampling operator (identity matrix with missing rows) and F discrete Fourier transform matrix.
MRI [3]	$A = SFD$	S is a subsampling operator (identity matrix with missing rows), F is the discrete Fourier transform matrix, and D is a diagonal matrix representing a spatial domain multiplication with the coil sensitivity map (assuming a single coil aquisition with Cartesian sampling in a SENSE framework [65]).
Computed tomography [58]	$A = R$	R is the discrete Radon transform [66].
Phase Retrieval [67–70]	$\mathcal{A}(\boldsymbol{x}) = \|A\boldsymbol{x}\|^2$	$\|\cdot\|$ denotes the absolute value, the square is taken elementwise, and A is a (potentially complex-valued) measurement matrix that depends on the application. The measurement matrix A is often a variation on a discrete Fourier transform matrix.

(source: [Ongie et al., 2020])

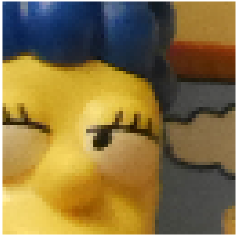
Gaussian denoising

Let's start with the case $A = \mathrm{Id}$, i.e. **image denoising**:

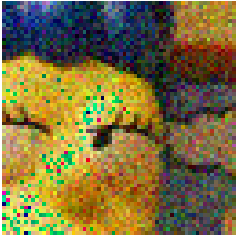
$$v = u_0 + w \quad \text{where} \quad w \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}).$$

We want to estimate u_0 from a single realization of v...

$\rightarrow$ We need to add some prior knowledge on the solution (e.g. regularity assumption).



u_0



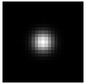
v

Deblurring

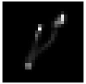
- A spatially invariant blur can be modeled by a convolution operator $Au = k * u$
- Several types of blur exist (motion, defocus)
- **Non-blind deblurring** consists in recovering u_0 from

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

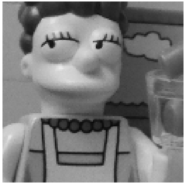
- We won't tackle blind deblurring here.



Isotropic blur



Motion blur



Original

Blurred

Example of motion blur

Super-Resolution

Super-résolution consists in finding another version of v at higher resolution.

This is an inverse problem corresponding to the subsampling operator with stride $s \in \mathbb{N}^*$:

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

In practice, we often apply an (*anti-aliasing*) filter before subsampling.

With prefiltering, we obtain the operator

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

Super-resolution consists in recovering u_0 from

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

The degraded image v is defined on a subgrid of stride s.

Inpainting

Inpainting consists in filling missing regions in images

The degradation operator then writes

$$Au = u\mathbf{1}_\omega$$

where $\omega \subset \Omega$ is the set of known pixels and $\Omega \setminus \omega$ the mask.

Inverse problem

We wish to recover u_0 from

$$v = Au_0 + w.$$

The problem is said ill-posed when A is not invertible or with unstable inverse.

Example : For deblurring, $Au = k * u$, we can invert A directly in Fourier domain:

Inverse problem

We wish to recover u_0 from

$$v = Au_0 + w.$$

The problem is said ill-posed when A is not invertible or with unstable inverse.

Example : For deblurring, $Au = k * u$, we can invert A directly in Fourier domain:

$$u = \mathcal{F}^{-1}\left(\frac{\hat{v}}{\hat{k}}\right) = \mathcal{F}^{-1}\left(\hat{u}_0 + \frac{\hat{w}}{\hat{k}}\right) \quad \longrightarrow \quad \text{but noise explodes !}$$

Inverse problem

We wish to recover u_0 from

$$v = Au_0 + w.$$

The problem is said ill-posed when A is not invertible or with unstable inverse.

Example : For deblurring, $Au = k * u$, we can invert A directly in Fourier domain:

$$u = \mathcal{F}^{-1}\left(\frac{\hat{v}}{\hat{k}}\right) = \mathcal{F}^{-1}\left(\hat{u}_0 + \frac{\hat{w}}{\hat{k}}\right) \quad \longrightarrow \quad \text{but noise explodes !}$$

When the problem is ill-posed, there may be multiple solutions or erroneous solutions.

It is thus useful to adopt an *a priori* on the solution, e.g. imposing some kind of regularity.

Image Restoration by Optimization

We will therefore try to solve

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

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\operatorname{Argmin}_{u \in \mathbf{R}^\Omega} F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \operatorname{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

Example: for denoising ($A = \mathrm{Id}$),

Image Restoration by Optimization

We will therefore try to solve

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

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\underset{u \in \mathbf{R}^\Omega}{\mathrm{Argmin}}\, F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \mathrm{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

Image Restoration by Optimization

We will therefore try to solve

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

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\underset{u \in \mathbf{R}^\Omega}{\mathrm{Argmin}}\, F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \mathrm{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

For differentiable F, we can always consider simple gradient descent.

Example: The gradient of $f(u) = \frac{1}{2}\|Au - v\|_2^2$ is

Image Restoration by Optimization

We will therefore try to solve

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

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\operatorname{Argmin}_{u \in \mathbf{R}^\Omega} F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \operatorname{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

For differentiable F, we can always consider simple gradient descent.

Example: The gradient of $f(u) = \frac{1}{2}\|Au - v\|_2^2$ is $\nabla f(u) = A^T(Au - v)$.

Image Restoration by Optimization

We will therefore try to solve

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

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\underset{u \in \mathbf{R}^\Omega}{\mathrm{Argmin}}\, F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \mathrm{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^T A + \lambda I)^{-1} A^T v$$

Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

For differentiable F, we can always consider simple gradient descent.

Example: The gradient of $f(u) = \frac{1}{2}\|Au - v\|_2^2$ is $\nabla f(u) = A^T(Au - v)$.

If $Au = k * u$ (periodic convolution), then $A^T u =$

Image Restoration by Optimization

We will therefore try to solve

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

where $R(u)$ imposes some kind of regularity of u, and $\lambda \geq 0$ is a parameter.

The problem $\underset{u \in \mathbf{R}^\Omega}{\mathrm{Argmin}}\, F(u)$ is very high-dimensional, and we need efficient algorithms.

Simple (nearly useless) regularization: Consider $R(u) = \frac{\lambda}{2}\|u\|_2^2$. Then $u_\lambda \in \mathrm{Argmin}_F$ is given by

$$A^T(Au_\lambda - v) + \lambda u_\lambda = 0 \quad \text{i.e.} \quad u_\lambda = (A^TA + \lambda I)^{-1}A^Tv$$

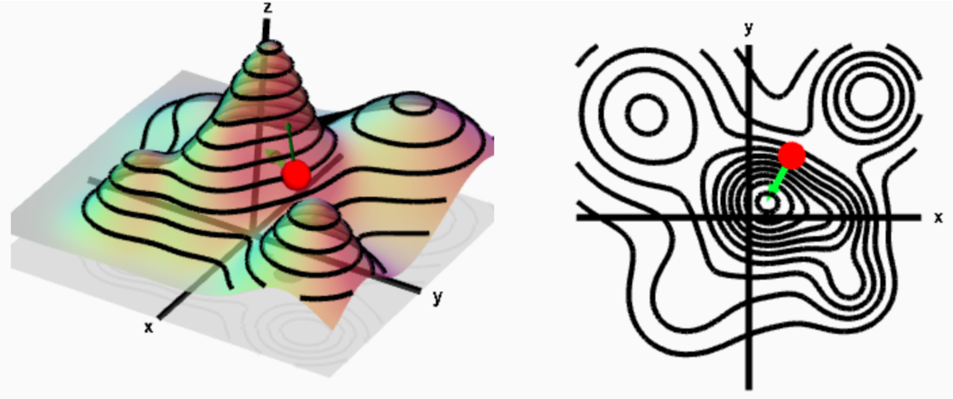
Example: for denoising ($A = \mathrm{Id}$), it just divides all values by $1 + \lambda$...

For differentiable F, we can always consider simple gradient descent.

Example: The gradient of $f(u) = \frac{1}{2}\|Au - v\|_2^2$ is $\nabla f(u) = A^T(Au - v)$.

If $Au = k * u$ (periodic convolution), then $A^Tu = \tilde{k} * u$ with $\tilde{k}(\mathbf{x}) = \overline{k(-\mathbf{x})}$.

The Steepest Descent



https://mathinsight.org/directional_derivative_gradient_introduction

Descent Lemma

Let $f : \mathbf{R}^d \to \mathbf{R}$ be differentiable with L-Lipschitz gradient. Then, for any $x, y \in \mathbf{R}^d$,

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

Consequence: If we choose $\tau \in [0, \frac{2}{L}]$, then

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

Gradient Descent

We consider here the gradient descent method:

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

where $\tau_n > 0$ is a sequence of step sizes.

- For $\tau_n = \tau$ constant, we speak of fixed step size.
- We speak of optimal step size if, at each iteration n, we choose

$$\tau_n \in \operatorname*{Argmin}_{t \in \mathbf{R}} f(x_n - t\nabla f(x_n)).$$

The descent lemma gives that for f differentiable with L-Lipschitz gradient and $\underline{\tau \leqslant \frac{2}{L}}$,

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

Thus, if f is lower bounded, $f(x_n)$ converges.

Convexity and Minimum

The function $f : \mathbf{R}^d \to \mathbf{R}$ is convex if for all $x, y \in \mathbf{R}^d$,

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

It is said strictly convex if the inequality is strict.

If f is convex and differentiable, one can show that for any $x, y \in \mathbf{R}^d$,

$$f(y) \geqslant f(x) + \nabla f(x) \cdot (y - x).$$

Consequence : If f is convex and differentiable, then

$$x \in \operatorname{Argmin} f \quad \Longleftrightarrow \nabla f(x) = 0.$$

The argmin is unique as soon as f is strictly convex.

Strong Convexity

We say that f is α-convex (with $\alpha \in \mathbf{R}$) if $f - \frac{\alpha}{2}\|\cdot\|^2$ is convex.

When $\alpha > 0$, we say that f is **strongly convex**.

Remark : The convexity and the gradient Lipschitz constant can be read on the Hessian:

If $A, B \in \mathbf{R}^{d\times d}$ are symmetric, we write $A \succeq B$ if $A - B$ if semi-definite positive, i.e.

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

For $f : \mathbf{R}^d \to \mathbf{R}$ of class $\mathscr{C}^2$,

$$\nabla f \text{ is } L\text{-Lipschitz iff} \quad \forall x \in \mathbf{R}^d,\ -L\mathrm{Id} \preceq \nabla^2 f(x) \preceq L\mathrm{Id}.$$
i.e. $\forall x$ the eigenvalues of $\nabla^2 f(x)$ have moduli $\leq L$.

$$f \text{ is } \alpha\text{-convex iff} \quad \forall x \in \mathbf{R}^d,\ \nabla^2 f(x) \succeq \alpha\mathrm{Id}$$
i.e. $\forall x$ the eigenvalues of $\nabla^2 f(x)$ are all $\geq \alpha$.

Convergence Guarantees, Convex Case

Theorem

Let $f : \mathbf{R}^d \to \mathbf{R}$ be convex differentiable with ∇f L-Lipschitz. Assume that $\operatorname{Argmin} f$ is non-empty. Let $\tau \in (0, \frac{2}{L})$, $x_0 \in \mathbf{R}^d$ and (x_n) the sequence defined by

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

Then (x_n) converges towards an element of $\operatorname{Argmin} f$.

Theorem

Let $f : \mathbf{R}^d \to \mathbf{R}$ be differentiable and α-strongly convex with L-Lipschitz gradient. Then there exists a unique $x_* \in \operatorname{Argmin} f$, and for $\tau < \frac{1}{L} \leqslant \frac{1}{\alpha}$, we have

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

Plan

Optimization for Inverse Problems

To solve the inverse problem $v = Au_0 + w$, we can thus minimize

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

with $f(u) = \frac{1}{2}\|Au - v\|^2$ and $g(u) = \lambda R(u)$, $\lambda > 0$.

Consider here $R(u) = \|Bu\|_2^2$ with $B \in \mathbf{R}^{p\times d}$, F is convex and differentiable.

Solutions are characterized by $\nabla F(u) = 0$ i.e. $A^T(Au - v) + 2\lambda B^T Bu = 0$.

Also, we can minimize F by gradient descent with $\tau < \frac{2}{L}$ where $L = \|A^T A + 2\lambda B^T B\|$.

- For $Au = k * u$, $A^T Au = \mathcal{F}^{-1}(|\hat{k}|^2\hat{u})$.
  If $|\hat{k}| \leq 1$, it follows that $\|A^T A\| \leq 1$.
- For $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ and $\|A\| = 1$.

Optimization for Inverse Problems

To solve the inverse problem $v = Au_0 + w$, we can thus minimize

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

with $f(u) = \frac{1}{2}\|Au - v\|^2$ and $g(u) = \lambda R(u)$, $\lambda > 0$.

Consider here $R(u) = \|Bu\|_2^2$ with $B \in \mathbf{R}^{p\times d}$, F is convex and differentiable.

Solutions are characterized by $\nabla F(u) = 0$ i.e. $A^T(Au - v) + 2\lambda B^T Bu = 0$.

Also, we can minimize F by gradient descent with $\tau < \frac{2}{L}$ where $L = \|A^T A + 2\lambda B^T B\|$.

- For $Au = k * u$, $A^T Au = \mathcal{F}^{-1}(|\hat{k}|^2\hat{u})$.
  If $|\hat{k}| \leq 1$, it follows that $\|A^T A\| \leq 1$.
- For $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ and $\|A\| = 1$.

Good news: By automatic differentiation you need only coding $F(u)$...

Optimization for Inverse Problems

To solve the inverse problem $v = Au_0 + w$, we can thus minimize

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

with $f(u) = \frac{1}{2}\|Au - v\|^2$ and $g(u) = \lambda R(u)$, $\lambda > 0$.

Consider here $R(u) = \|Bu\|_2^2$ with $B \in \mathbf{R}^{p\times d}$, F is convex and differentiable.

Solutions are characterized by $\nabla F(u) = 0$ i.e. $A^T(Au - v) + 2\lambda B^T Bu = 0$.

Also, we can minimize F by gradient descent with $\tau < \frac{2}{L}$ where $L = \|A^T A + 2\lambda B^T B\|$.

- For $Au = k * u$, $A^T Au = \mathcal{F}^{-1}(|\hat{k}|^2\hat{u})$.
  If $|\hat{k}| \leq 1$, it follows that $\|A^T A\| \leq 1$.
- For $Au = \mathbf{1}_\omega u$, $A^T A = A^2 = A$ and $\|A\| = 1$.

Good news: By automatic differentiation you need only coding $F(u)$...

But ! in order to avoid instability problems, you'd better know what F does...

Let us start with zero regularization!

Consider here

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

- We have an orthogonal decomposition $\mathbf{R}^d = K \oplus K^\perp$ with $K = \mathrm{Ker}[A]$ and $K^T = \mathrm{Im}[A^T]$
- Therefore $\mathrm{Argmin}_{\mathbf{R}^d} f$ is non-empty and we can define

$$A^+ v = \min_{u \in \mathrm{Argmin}\, f} \|u\|_2^2.$$

  It defines a linear operator A^+, called Moore-Penrose pseudo-inverse.
- The Moore-Penrose pseudo-inverse has a zero component in $\mathrm{Ker}[A]$.
- $A_{K^T} : K^T \to \mathrm{Im}(A)$ is invertible. Thus $A^+ = A_{K^T}^{-1} P$ (with P the orthogonal projection on $\mathrm{Im}(A^T)$).
- Actually, one can show that $A^+ v = \lim_{\lambda \to 0} (A^T A + \lambda I)^{-1} A^T v$.
- Gradient descent on $f(u)$ converges to $A^+ v$, as soon as initialization has null component on K.
- But $A^+ v$ is generally a bad solution for inverse problems because of bad conditioning.

Explicit Regularizations

We define the discrete derivatives of u by

$$\nabla u(x,y) = \begin{pmatrix} \partial_1 u(x,y) \\ \partial_2 u(x,y) \end{pmatrix} \quad \text{avec} \quad \begin{cases} \partial_1 u(x,y) = d_1 * u(x,y) = u(x+1,y) - u(x,y) \\ \partial_2 u(x,y) = d_2 * u(x,y) = u(x,y+1) - u(x,y) \end{cases} .$$

We define **Tychonov regularization** by

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

We define the **total variation** by

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

Back to denoising

Let us minimize

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

where R is a regularization and $\lambda > 0$.

Consider first Tychonov regularization $R(u) = \|\nabla u\|_2^2$.

Back to denoising

Let us minimize

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

where R is a regularization and $\lambda > 0$.

Consider first Tychonov regularization $R(u) = \|\nabla u\|_2^2$**.**

We have $\nabla R(u) = 2\nabla^T \nabla u$.

Back to denoising

Let us minimize

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

where R is a regularization and $\lambda > 0$.

Consider first Tychonov regularization $R(u) = \|\nabla u\|_2^2$**.**

We have $\nabla R(u) = 2\nabla^T \nabla u$. As F is convex,

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

Back to denoising

Let us minimize

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

where R is a regularization and $\lambda > 0$.

Consider first Tychonov regularization $R(u) = \|\nabla u\|_2^2$.

We have $\nabla R(u) = 2\nabla^T \nabla u$. As F is convex,

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

For $p : \Omega \to \mathbf{R}^2$, $\nabla^T p$ is given by

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

Actually, $\text{div}(p) := -\nabla^T p$ is a discrete divergence and $\Delta u := -\nabla^T \nabla u$ is a discrete Laplacian.

Explicit Solution: Wiener filtering

Theorem

Let $v \in \mathbb{C}^\Omega$ and $\lambda > 0$. The function $F : \mathbb{C}^\Omega \to \mathbf{R}_+$ defined by

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

has a minimum attained at a unique $u_* \in \mathbb{C}^\Omega$, which is given in Fourier domain:

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

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

Remarks:

- d_1, d_2 are the kernel derivatives, e.g. $d_1 = \delta_{(-1,0)} - \delta_{(0,0)}$. So $\hat{L}$ is the kernel of $-\Delta$ filter.
- The theorem adapts for deblurring with Tychonov regularization:

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

Link with an evolution model

The gradient descent on

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

writes as

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

The sequence (u_n) converges to u_* as soon as $\tau < \frac{2}{L}$ with $L = \|I + 2\lambda\nabla^T\nabla\| = 1 + 16\lambda$.

Link with an evolution model

The gradient descent on

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

writes as

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

The sequence (u_n) converges to u_* as soon as $\tau < \frac{2}{L}$ with $L = \|I + 2\lambda\nabla^T\nabla\| = 1 + 16\lambda$.

If we drop the data-fidelity... then gradient descent on $u \mapsto \|\nabla u\|_2^2$ gives

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

This is a discretization of the heat equation $\partial_t u = c\Delta u$ with initial condition u_0.

Smoothed Total Variation

What if we want to minimize

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

Problem: The total variation is not differentiable.

A simple solution: consider a smoothed variant: For $\varepsilon > 0$, let

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

One can see that

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

And one can show that $\nabla \mathrm{TV}_\varepsilon$ is $\frac{8}{\varepsilon}$-Lipschitz.
We can thus minimize F by gradient descent with $\tau < \frac{2}{1+\frac{8\lambda}{\varepsilon}}$.

Denoising Examples

Noisy
PSNR = 19.93

Tychonov denoising
PSNR = 25.89

$\mathrm{TV}_{\varepsilon}$ denoising
PSNR = 27.21

Projected Gradient Descent

Imagine that we want to constrain the solution into a convex closed set $C \subset \mathbf{R}^d$:

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

For that, we can use the orthogonal projection $p_C : \mathbf{R}^d \to C$.

Theorem

Let $f : \mathbf{R}^d \to \mathbf{R}$ be convex differentiable such that ∇f is L-Lipschitz.
Let $C \subset \mathbf{R}^d$ be a closed convex set. Assume that $\text{Argmin}_C f$ is non-empty.
For $\tau \in (0, \frac{2}{L})$, $x_0 \in \mathbf{R}^d$, let (x_n) be defined by

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

Then (x_n) converges to an element of $\text{Argmin}_C f$.

Example : For inpainting, we can deal with the noiseless problem $v = Au$.
In this case, we can perform constrained minimization of only the regularization term:

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

Proximal Operator

Definition (see e.g. the book [Bauschke, Combettes, 2011)

] Soit $g : \mathbf{R}^d \to \overline{\mathbf{R}}$. On définit l'opérateur proximal de g par

$$\mathrm{Prox}_g(u) = \underset{z \in \mathbf{R}^d}{\mathrm{Argmin}} \frac{1}{2}\|u - z\|_2^2 + g(z) \subset \mathbf{R}^d \ .$$

Proposition

Let $g : \mathbf{R}^d \to \mathbf{R} \cup \{+\infty\}$ a l.s.c. convex function such that $g \not\equiv +\infty$. Then

- $\mathrm{Prox}_g(u)$ *is single-valued, and thus defines a point* $\mathrm{Prox}_g(u) \in \mathbf{R}^d$*.*
- *If besides g is differentiable, then* $p = \mathrm{Prox}_g(u)$ *satisfies*

$$p = \mathrm{Prox}_g(u) \ \Leftrightarrow \ u = (\mathrm{Id} + \nabla g)(p) \ .$$

Example

- If $g = \imath_C$ the indicator function of $C \subset \mathbf{R}^d$ convex, then Prox_g is the orthogonal projection on C.
- If $f(u) = \frac{1}{2}\|Au - v\|_2^2$, then $\mathrm{Prox}_{\tau f}(u) = \tau(\mathrm{Id} + \tau A^T A)^{-1} A^T u$

Proximal Gradient Descent

In order to minimize $F = f + g$, one can use the proximal gradient descent (PGD) algorithm:

$$u_{n+1} = \mathrm{Prox}_{\tau g}(u_n - \tau \nabla f(u_n)) \ ,$$

where $\tau > 0$ is a fixed step size.

Theorem

Let $f : \mathbf{R}^d \to \mathbf{R}$ be a convex differentiable function with L_f-Lipschitz gradient.
Let $g : \mathbf{R}^d \to \mathbf{R} \setminus \{+\infty\}$ be convex l.s.c. such that $g \not\equiv +\infty$.
Assume that $F = f + g$ attains its infimum at some point in $\mathbf{R}^d$. Assume $\tau < \frac{2}{L_f}$.
Then the sequence (u_n) given by the PGD algorithm converges to some $u_* \in \mathrm{Argmin}\, F$.

Remarks:

- This theorem applies even for non-differentiable (or even non continuous) g.
- It generalizes the projected gradient descent algorithm.

Iterative Thresholding

Let $B \in \mathbf{R}^{d \times d}$ be an orthogonal matrix.
Let $q \in \{0, 1\}$ and define $R(u) = \lambda \|Bu\|_q$ for any $u \in \mathbf{R}^d$.
Let us define

$$\forall t \in \mathbf{R}, \quad h_{0,\lambda}(t) = t\mathbf{1}_{|t| > \sqrt{2\lambda}} \quad \text{and} \quad h_{1,\lambda}(t) = \operatorname{sgn}(t)(|t| - \lambda)_+ \ .$$

We then extend $h_{q,\lambda}$ to vector by applying it component-wise.

Theorem

For $u \in \mathbf{R}^d$, we define $T_q^\lambda u \in \mathbf{R}^d$ by

$$T_q^\lambda u = B^T h_{q,\lambda}(Bu) \ .$$

Then $T_q^\lambda u \in \operatorname{Prox}_{\lambda R}(u)$.

In order to minimize

$$F(u) = \frac{1}{2}\|Au - v\|_2^2 + \lambda \|Bu\|_1$$

we can thus use the following PGD algorithm (known as *Iterative Soft-Tresholding*):

$$u_{n+1} = T_1^\lambda(u_n - \tau \nabla f(u_n)) \ .$$

In order to solve image restoration problems, one can take B to be the Fourier/wavelet basis.

Plan

Euclidean metrics

- Given two images u and v of size $M \times N$ with graylevels between 0 and 255.
- Denote $\Omega = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$ the pixel domain
- Mean Square Error $\downarrow$:

$$\mathrm{MSE}(u, v) = \frac{1}{MN} \sum_{\mathbf{x} \in \Omega} (u(\mathbf{x}) - v(\mathbf{x}))^2$$

- Root Mean Square Error $\downarrow$:

$$\mathrm{RMSE}(u, v) = \left(\frac{1}{MN} \sum_{\mathbf{x} \in \Omega} (u(\mathbf{x}) - v(\mathbf{x}))^2 \right)^{\frac{1}{2}}$$

- Peak Signal to Noise Ratio $\uparrow$:

$$\mathrm{PSNR}(u, v) = 20 \log_{10} \left(\frac{\mathrm{MAX}}{\mathrm{RMSE}(u, v)} \right) \quad (\text{where } \mathrm{MAX} = 255)$$

- Useful for inverse problems such as denoising.
- Not ideal when one hopes to generate new content.

Structural similarity index measure (SSIM ↑) [Wang et al., 2004]

Between patches:

- Given two patches x, y (typically of size 8×8 or 11×11 with a Gaussian windowing)

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \in [-1, 1]$$

  with:
  - μ_x the pixel sample mean of x
  - μ_y the pixel sample mean of y
  - σ_x^2 the variance of x
  - σ_y^2 the variance of y
  - σ_{xy} the covariance of x and y
  - $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator, with the range $L = 255$ or 1 and $k_1 = 0.01$ and $k_2 = 0.03$ by default.

- $\mathrm{SSIM}(x,y)$ is the product of three terms:

Luminance	Contrast	Structure
$l(x,y) = \frac{2\mu_x\mu_y+c_1}{\mu_x^2+\mu_y^2+c_1}$	$c(x,y) = \frac{2\sigma_x\sigma_y+c_2}{\sigma_x^2+\sigma_y^2+c_2}$	$s(x,y) = \frac{\sigma_{xy}+c_2/2}{\sigma_x\sigma_y+c_2/2}$

Structural similarity index measure (SSIM $\uparrow$) [Wang et al., 2004]

Between images:

- Given two images u and v of size $M \times N$ with graylevels between 0 and 255, define the Mean-SSIM by averaging over all patches:

$$\mathrm{(M)SSIM}(u, v) = \mathrm{mean}(\{\mathrm{SSIM}(P_{\mathbf{x}}(u), P_{\mathbf{x}}(v)),\ \mathbf{x} + \omega \subset \Omega\})$$

  where $P_{\mathbf{x}}(u)$ is the restriction of u on the patch $\mathbf{x} + \omega$.
- There are also multiscale variants.
- SSIM is not a distance, its range is $[-1, 1]$.
- SSIM is closer to a perceptual distance, especially regarding local textures.

LPIPS $\downarrow$ [Zhang et al., 2018]

LPIPS: Learned Perceptual Image Patch Similarity

- Previous works on texture synthesis [Gatys et al., 2015] and style transfer [Gatys et al., 2016]
- [Johnson et al., 2016] have shown the importance of the VGG [Simonyan and Zisserman, 2015] features for perceptual similarity between images.
- This means that intermediate features of classification CNN are useful in their own: "**a good feature is a good feature**. Features that are good at semantic tasks are also good at self-supervised and unsupervised tasks, and also provide good models of both human perceptual behavior and macaque neural activity."

LPIPS model: Define a perceptual distance between 64×64 patches by computing a Euclidean norm between features:

$$\mathrm{LPIPS}(u, u_0)^2 = \sum_{\text{layers } \ell} \frac{1}{H_\ell W_\ell} \sum_{i,j} \| w_\ell \odot (F^\ell(u)_{i,j} - F^\ell(u_0)_{i,j}) \|_2^2$$

where for each layer ℓ, the neural response $F^\ell(u)$ is weighted by channel weights $w_\ell \in \mathbf{R}^{C_\ell}$ that are learned to reproduce human evaluation of distortion between patches.

References I

Gatys, L. A., Ecker, A. S., and Bethge, M. (2015).
Texture synthesis using convolutional neural networks.
In *Advances in Neural Information Processing Systems*, pages 262–270.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016).
Image style transfer using convolutional neural networks.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016).
Perceptual losses for real-time style transfer and super-resolution.
In *Computer Vision – ECCV 2016*, pages 694–711, Cham. Springer International Publishing.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. (2020).
Deep learning techniques for inverse problems in imaging.
IEEE Journal on Selected Areas in Information Theory, 1(1):39–56.

References II

Simonyan, K. and Zisserman, A. (2015).
Very deep convolutional networks for large-scale image recognition.
In Bengio, Y. and LeCun, Y., editors, *Proceedings of the International Conference on Learning Representations*.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004).
Image quality assessment: from error visibility to structural similarity.
IEEE transactions on image processing, 13(4):600–612.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018).
The unreasonable effectiveness of deep features as a perceptual metric.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.