Total Variation (TV) Deblurring with Inexact Proximal ϵ-subgradient methods

Majela Pentón Machado

June, 2025

1 Overview

This repository contains Python implementations and experiments of optimization algorithms for solving the Total Variation (TV) deblurring problem:

$$\min_{x \in \mathbb{R}^{N \times N}} \frac{1}{2} \|Ax - b\|_F^2 + \tau \sum_{i,j=1}^{N} \|(\nabla x)_{i,j}\|_2 ,$$

where $A : \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times N}$ is a linear operator, $\tau > 0$, $\| \cdot \|_F$ is the Frobenius, $\|\cdot\|_2$ is the Euclidean norm in $\mathbb{R}^2$ and $\nabla : \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times N} \times \mathbb{R}^{N \times N}$ is the **discrete gradient** operator:

$$(\nabla_1 x)_{ij} = x_{i+1,j} - x_{i,j}, \quad (\nabla_2 x)_{ij} = x_{i,j+1} - x_{i,j}.$$

The method implemented is the

- PeSM: Inexact proximal ϵ-subgradient method

from the paper

- Millan, R.D., Machado, M.P.: "Inexact proximal ϵ-subgradient methods for composite convex optimization problems" Journal of Global Optimization (2019). (https://doi.org/10.1007/s10898-019-00808-8)

The method support different parameter values (sigma) for inexactness criteria, and we evaluate its performance using the metrics:

- PSNR: Peak Signal-to-Noise Ratio

- SSIM: Structural Similarity Index

- Time: Execution time in seconds
- Inner Iterations: The total number of inner iterations
- Outer Iterations: Number of outer iterations

2 Repository Structure

```
tv-deblurring/
├── algorithms.py ........................ Core optimization method
├── utilities_tv.py ...................... TV operators, projections, etc
├── main.py ................... Runs experiments and generates plots
├── results/
│   └── performance.csv ............................... Output metrics
├── plots/
│   ├── barplot_psnr.png ............................ PSNR comparison
│   ├── barplot_ssim.png ............................ SSIM comparison
│   ├── barplot_time.png ................... Execution time comparison
│   ├── barplot_inner.png ................. Inner iterations comparison
│   └── barplot_outer.png ................. Outer iterations comparison
├── requirements.txt
└── README.md
```

Install the required Python packages with:

```
pip install -r requirements.txt
```

3 Running Main

To run the experiments and generate performance metrics:

```
python main.py
```

This will:

- Run the method across different values of sigma
- Save results to results/performance.csv
- Generate bar plots in plots/

The numerical experiments use the *Cameraman* test image blurred by a uniform 3×3 averaging filter (box blur) followed by a Gaussian noise with zero mean and standard deviation 10^{-4}. The regularization parameter τ was set to 10^{-4} and we choose $x_0 = b$. The stopping criterion for the

PϵSM is that either it reach 1000 iterations or the relative difference $\|x_k - x_{k-1}\| \, / \, \|x_k\|$ is less than 10^{-4}. We test the PϵSM for various values of the inexactness parameter σ chosen between 0.1 and 0.9.

The TV deblurring problem is solved using separable PSFs with Kronecker structure. The implementation uses functions from

- A. Beck and Marc Teboulle, "Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems"

available in Matlab at

- https://sites.google.com/site/amirbeck314/software?authuser=0