FISTA and Image Denoising

Additional project for "Convex Optimization", WS 2021/22

Ferdinand Vanmaele
Vanmaele@stud.uni-heidelberg.de

November 7, 2022

Abstract

The presence of noise in images is unavoidable. It may be introduced by the image formation process, image recording, image transmission, etc. These random distortitions make it difficult to perform any required picture processing. [1] For example, in the image deblurring project [2] we showed that even a small amount of noise can lead to poor deblurring results. Consequently, traditional methods in image processing attempt to reduce/remove the noise component prior to further operations. In this project, we formulate image denoising as a 2-dimensional convex optimization problem, using a data term and *total variation* as regularizer. We solve this problem numerically using FISTA schemes, including recent modifications with fast convergence by the authors [7], and test both the convergence of the schemes and quality of the denoised dimages.

1 Motivation and Overview

The image denoising problem is formulated mathematically as follows. Let the observed intensity function $u_0(x, y) : [0, 1]^2 \to \mathbb{R}$ denote the pixel values of a noise image for $x, y \in [0, 1]$. Let $u(x, y)$ denote the desired clean image, so

$$u_0(x, y) = u(x, y) + \eta(x, y),$$

with additive noise η. Our goal is to reconstruct u from u_0. To this end, we consider minimization problems on a Hilbert space $\mathcal{H}$,

$$\min_{u\in\mathcal{H}} J(u) := \min_{u\in\mathcal{H}} F(u) + \alpha R(u), \qquad \alpha > 0, \tag{1}$$

with a data term F (reflecting the structure of the noise) and a regularization term R describing the structure of the desired image. Before defining these further, it is important to define the function space of the sought-for image u. The space should have enough regularity, to filter out noise, while still allowing "jumps" (edges) that are found in images. This rules out functions that are differentiable in the classical sense. On the other hand, Lebesgue spaces like L_2 contain noise , and thus do not allow separation of image and noise. Therefore we need a space "in between". A starting point is *weakly differentiable* functions and the corresponding Sobolev spaces $W^{1,p}(\Omega)$, $\Omega \subseteq \mathbb{R}^n$, known from functional analysis. However, even for $p \to 1$ the function space $W^{1,p}(\Omega)$ will not contain elements that exhibit true edges, i.e. discontinuities along lines. The lecture notes [3, 3.2] demonstrate this with an example. This leads to spaces of *bounded variation* (BV), with finite *total variation* (TV). The following overview is taken from [4, 1 TV Regularization]

1.1 Denoising problem

Rudin, Osher and Fatemi proposed to estimate the denoised image u as a solution to the minimization

problem

$$\text{argmin}_{u\in \text{BV}(\Omega)} \|u\|_{\text{TV}(\Omega)} + \frac{\lambda}{2}\int_\Omega (f(x)-u(x))^2\,\mathrm{d}x, \tag{2}$$

where λ is a positive parameter. This problem is refered to as the ROF problem. Denoising is performed as an infinite-dimensional minimization problem, where the search space is all bounded variation (BV) images.

A function u is in $BV(\Omega)$ if it is integrable and there exists a Radon measure Du such that

$$\int_\Omega u(x)\text{div}g(x)\,\mathrm{d}x = -\int_\Omega \langle g, Du(x)\rangle \qquad \forall g \in C_c^1(\Omega,\mathbb{R}^2)^2.$$

This measure Du is the distributional gradient of u. When u is smooth, $Du(x) = \nabla u(x)\,\mathrm{d}x$. The toal variation ($TV$) seminorm of u is defined as

$$\|u\|_{\text{TV}(\Omega)} := \int_\Omega |Du| := \sup\{\int_\Omega u\,\text{div}g\,\mathrm{d}x : g \in C_c^1(\Omega,\mathbb{R}^2)^2, \sqrt{g_1^2+g_2^2}\le 1\}.$$

When u is smooth, TV is equivalently the integral of its gradient magnitude,

$$\|u\|_{\text{TV}(\Omega)} = \int_\Omega |\nabla u|\,\mathrm{d}x.$$

The TV term in the minimization discourages the solution from having oscillations, yet is does allow the solution to have discontinuities. The second term encourages the solution to be close to the observed image f. By this combination, a minimization finds a denoised image. If $f \in L^2$, the minimizer of the ROF problem exists, is unique and is stable in L^2 with respect to perturbations in f.

TV-regularized denoising can be extended to other noise models. If the noise η is *impulsive* with only individual pixels selectively corrupted, i.e. pointwise given as

$$\eta(x,y) = \begin{cases} \xi & \text{with probability } r,\\ 0 & \text{with probability } 1-r,\end{cases}$$

then an L^1 data fidelity term is a better choice,

$$\text{argmin}_{u\in \text{BV}(\Omega)} \|u\|_{\text{TV}(\Omega)} + \lambda\int_\Omega |f(x)-u(x)|\,\mathrm{d}x.$$

For Poisson noise, we have

$$\text{argmin}_u \|u\|_{\text{TV}(\Omega)} + \lambda\int_\Omega (u(x)-f(x)\log(ux))\,\mathrm{d}x.$$

TV denoising has been similarly extended to multiplicative noise and Rician noise, and these models can be extended to use a spatially varying λ. This imposes a locally adapted regularization strenght at different points of space,

$$\text{argmin}_{u\in \text{BV}(\Omega)} \|u\|_{\text{TV}(\Omega)} + \frac{1}{2}\int_\Omega \lambda(x)(f(x)-u(x))^2\,\mathrm{d}x.$$

The choice of noise model can significantly affect the denoising results. For better results, the noise model should agree with the actual noise distribution in the image. In this project, we limit ourselves to additive Gaussian noise, since the used L^2 term is smooth and thus applicable to algorithms which require a smooth data term.

1.2 Discretization of TV

For numerical solution of the minimization problem (2), several approaches for implementing the TV seminorm have been proposed in the literature. [4, 2 Algorithms] Two popular choices for the discrete TV are the isotropic TV defined by

$$TV_I(u) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\sqrt{(u_{i,j}-u_{i+1,j})^2+(u_{i,j}-x_{i,j+1})^2} + \sum_{i=1}^{m-1}|u_{i,n}-u_{i+1,n}| + \sum_{j=1}^{n-1}|u_{m,j}-u_{m,j+1}|$$

and the ℓ_1-based, anisotropic TV defined by

$$TV_{\ell_1}(u) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\{|u_{i,j}-u_{i+1,j}|+|u_{i,j}-u_{i,j+1}|\} + \sum_{i=1}^{m-1}|u_{i,n}-u_{i+1,n}| + \sum_{j=1}^{n-1}|u_{m,j}-u_{m,j+1}|$$

where the above formulas assume the reflexive boundary conditions [5, II Discrete Total Variation Regularization Model]

$$u_{m+1,j} - u_{m,j} = 0, \quad \forall j \quad \text{and} \quad u_{i,n+1} - u_{i,n} = 0, \quad \forall i.$$

In this project we use the isotropic TV discretization and its implementation by the `pyproximal` Python library. For a further discussion on possible discretizations of TV, see [4, 2 Algorithms].

The discrete formulation of problem (2) with an $_2$ data term is then given by

$$\operatorname{argmin}_{u \in \mathbb{R}^{m \times n}} \frac{1}{2}\|u - f\|_F^2 + \lambda \operatorname{TV}_I(u), \quad \lambda > 0. \quad (3)$$

The first term $F := \frac{1}{2}\|u-f\|_F^2$ has Lipschitz-continus gradient $u-f$ with Lipschitz constant $\|I\|_F$. The second term $R := \lambda TV_I(u)$ is proper, closed and convex, and the sum (3) has a non-empty set of minimizers. We now look at concrete methods for solving this problem numerically, following [7, 1 Introduction].

2 Algorithms

A classical approach to solve problem (3) is Forward-Backward splitting, also known as *proximal gradient descent*. With an initial point $x_0 \in \mathbb{R}^{m \times n}$ chosen arbitrarily, the standard FBS iteration reads as

$$x_{k+1} := \operatorname{prox}_{\gamma_k R}(x_k - \gamma_k \nabla F(x_k)), \quad \gamma_k \in]0, 2/L]$$

with L the Lipschitz constant of ∇F, γ_k the step size, and

$$\operatorname{prox}_{\gamma R}(\cdot) := \operatorname*{argmin}_{x \in \mathbb{R}^{m \times n}} \gamma R(x) + \frac{1}{2}\|x - \cdot\|^2$$

the *proximity operator* of R.

Remark. A closed form of the proximity operator is in general not available. For the total variation operator, an approximation can be computed iteratively, for example with methods described in [5].

Similar to gradient descent, FBS is a descent method: the objective function value $J(x_k)$ is non-increasing under properly chosen step-sizes γ_k. The convergence properties of FBS are established in the literature, in terms of both sequence and objective function value:

- The convergence of the generated sequence $\{x_k\}_{k \in \mathbb{N}}$ and the objective function $J(x_k)$ are guaranteed as long as γ_k is chosen such that $0 < \gamma_k < \frac{2}{L}$.
- Convergence rate: we have $J(x_k) - \min_x J(x) = o(1/k)$ for the objective function value and $\|x_k - x_{k-1}\| = o(1/\sqrt{k})$ for the sequence $\{x_k\}$. Linear convergence rate can be obtained under certain conditions, such as strong convexity.

Numerous variants of FBS have been proposed under different purposes. The *inertial Forward-Backward* algorithm applied to problem (3) is the following iteration:

$$\begin{aligned} y_k &= x_k + a_k(x_k - x_{k-1}) \\ x_{k+1} &= \operatorname{prox}_{\gamma_k R}(y_k - \gamma_k \nabla F(x_k)), \quad \gamma_k \in (0, 2/L), \end{aligned}$$

where a_k is the *inertial parameter* which controls the momentum $x_k - x_{k-1}$. The convergence of this scheme can be guaranteed under proper choices of γ_k and a_k. Under the same step-size choice, the scheme can have better practical performance than FBS, but in general no convergence rate is established.

2.1 FISTA

FISTA is a particular example of the class of inertial FBS algorithms. What differentiates FISTA is the restriction on the step size γ_k and special rule for updating a_k. Moreover, by consequence of the updating rule for a_k, FISTA schemes have convergence rate guarantees on the objective function value $J(x_k)$. The original FISTA scheme from [6] is described in Algorithm 1.

Due to the choices of parameters, FISTA achieves the optimal $O(1/k^2)$ convergence rate for $J(x_k) - \min_{x \in \mathcal{H}} J(x)$. In practice, however, the FISTA scheme may show oscillatory behavior with regards to the error norm $\|x_k - x^*\|$. If the objective is strongly convex, there exists a primal a^* such that the iteration no longer interates. Under weaker conditions, modifications exist which aim to reduce this effect.[1]

[1] The authors [7] propose several restarting schemes to deal with this oscillatory behavior, for strongly convex and locally

Algorithm 1 FISTA

Initial: $t_0 = 1$, $\gamma = 1/L$ and $x_0 \in \mathcal{H}$, $x_{-1} = x_0$, $k = 1$

repeat

$$t_k = \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}, \quad a_k = \frac{t_{k-1} - 1}{t_k}$$
$$y_k = x_k + a_k(x_k - x_{k-1}),$$
$$x_{k+1} = \text{prox}_{\gamma R}(y_k - \gamma \nabla F(y_k)).$$
$$k = k + 1;$$

until *convergence;*

The convergence of the sequence $\{x_k\}_{k\in\mathbb{N}}$ was answered by Chambolle and Dossall [9], by considering a modified rule to update t_k. Let $d > 2$ and

$$t_k = \frac{k+d}{d}, \quad a_k = \frac{t_{k-1} - 1}{t_k} = \frac{k-1}{k+d}. \tag{4}$$

Such a rule maintains the $O(1/k^2)$ objective convergence rate, and also allows the authors to prove the convergence of $\{x_k\}_{k\in\mathbb{N}}$. For the rest of this report, we refer to (4) as FISTA-CD.

The practical performance of FISTA-CD is almost identical to FISTA if d is chosen close to 2. When relatively large values of d are chosen, significant practical acceleration can be obtained, even without proper theoretical justifications on how to choose the value of d. [7, 1.2 Problems] See Section 3 for a detailed comparison.

2.2 Modified FISTA

By studying the t_k updating rule of FISTA and its difference with the updating rule (4), the authors [7] propose a modified FISTA scheme which applies the following rule:

$$t_k = \frac{p + \sqrt{q + rt_{k-1}^2}}{2}, \quad a_k = \frac{t_{k-1} - 1}{t_k},$$

where $p, q \in (0, 1]$ and $r \in (0, 4]$, see Algorithm 2. Such a modification has two advantages when $r = 4$:

- It maintains the $O(1/k^2)$ convergence rate of the original FISTA-BT; [7, Theorem 3.3]
- It allows to show a convergence rate of $o(1/k)$ on the sequence $\{x_k\}_{k\in\mathbb{N}}$. [7, Theorem 3.5]

Algorithm 2 FISTA-Mod

Initial: $p, q > 0$ and $r \in (0, 4]$, $t_0 = 1$, $\gamma \leq 1/L$ and $x_0 \in \mathcal{H}$, $x_{-1} = x_0$.

repeat

$$t_k = \frac{p + \sqrt{q + rt_{k-1}^2}}{2}, \quad a_k = \frac{t_{k-1} - 1}{t_k},$$
$$y_k = x_k + a_k(x_k - x_{k-1}),$$
$$x_{k+1} = \text{prox}_{\gamma R}(y_k - \gamma \nabla F(y_k))$$

until *convergence*;

For the proposed scheme and FISTA-CD, owing to the free parameters in computing t_k, the authors [7] propose a so-called "lazy-start" strategy for practical acceleration. The idea of such a strategy is to slow down the speed of a_k approaching 1. In pratice, this amounts to the following choice of parameters: [7, Proposition 4.1]

- FISTA-Mod: $p \in \left[\frac{1}{80}, \frac{1}{10}\right]$, $q \in [0, 1]$ and $r = 4$;
- FISTA-CD: $d \in [10, 80]$.

The advantages of this strategy are explained in detail in [7, 4 Lazy-start strategy].

3 Numerical experiments

In this section, we apply FISTA and its modified versions FISTA-Mod and FISTA-CD to the discrete to-

strongly convex objectives. However, we were unable to get any practical benefits from this approach when applied to the TV denoising problem (3). This may be due to an implementation detail, or because the restarting schemes were initially formulated for 1D problems. As such, we do not consider them for the rest of the discussion.

tal variation denoising problem (3),

$$\operatorname*{argmin}_{u\in\mathbb{R}} J(u) := \operatorname*{argmin}_{u\in\mathbb{R}^{m\times n}} \frac{1}{2}\|u-f\|_F^2 + \lambda\,\mathrm{TV}_I(u), \quad \lambda > 0.$$

The following algorithms are compared:

- The original FISTA scheme;
- FISTA-Mod with $p = 1/20$ and $q = 1/2$, i.e. the lazy-start strategy;
- FISTA-CD with $d = 20$.

The following settings were chosen:

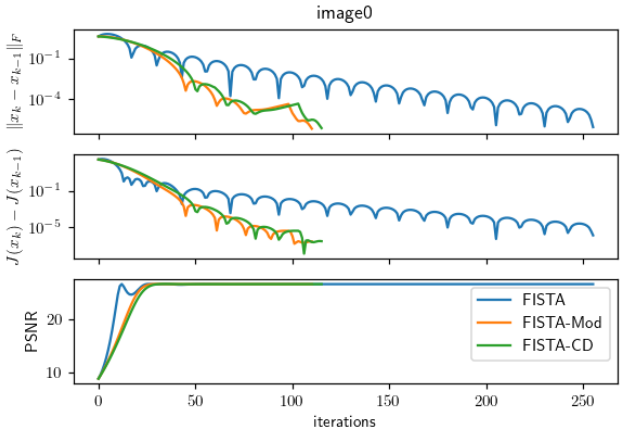
- All schemes have the same initial point $x_0 = 0 \in \mathbb{R}^{256\times 256}$ and $\lambda = 0.06$.
- Images of size (256, 256) were generated with StyleGan2, [8] manually selection to rule out images with artifacts. A total of 8 images was used in the experiments. Gaussian noise with mean 0 and variance 0.05 was added to each image u.
- Convergence was verified by checking the difference for iterates $\|x^{(k)} - x^{(k-1)}\|_F$ and $\|J(x^{(k)}) - J(x^{(k-1)})\|_F$. (Unlike the authors [7], we did not have a global minimizer x^* for problem (3) at hand.) All schemes were set to terminate upon $\|x^{(k)} - x^{(k-1)}\|_F < 10^{-6}$.
- The influence of iterations on the result image was computed through the PSNR,

$$PSNR = 20\log_{10}\left\{\frac{\max_{i,j}(u_{i,j})}{\frac{1}{mn}\sqrt{\sum_i\sum_j (u_{i,j} - x_{i,j}^{(k)})^2}}\right\}.$$

- The proximal operator for total variation was computed with the `pyproximal` Python library.

4 Conclusion

We formulated the image denoising problem for a set of images with Gaussian noise added, and solved it numerically using a set of FISTA schemes. FISTA-Mod and FISTA-CD had significantly better (empirical) convergence for both iterates $x^{(k)}$ and objective values $J(x^{(k)})$. The effect on the PSNR of the denoised images was – beyond a certain tolerance – neglectable, stabilizing after about 20 to 50 iterations for FISTA and its modifications. However, the modified algorithms do not result in an increased computational cost, and their use still seems preferable. Outside of an artificial setting, the original images are not available for computing the PSNR, but the iterates and objective values remain available metrics. In all cases, oscillation for all FISTA schemes was clearly noticeable.



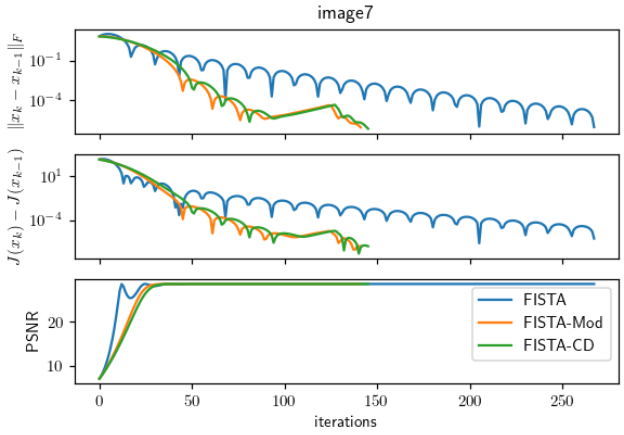
(a) Original (b) Noisy (c) Denoised

Figure 1: Results for a sample image [`image0.png`]

References

[1] L.I. Rudin, S. Osher, E. Fatemi. *Nonlinear total variation based noise removal algorithms*, Physica D 60 (1992), 259-268.

[2] F. Vanmaele. *Total Variation Restoration of Spatially Variant Blur*, Additional project for Mathe-



(a) Original (b) Noisy (c) Denoised

Figure 2: Results for a sample image [`image5.png`]

matical Image Processing, Heidelberg University, 2021.

[3] S. Petra. *Mathematical Image Processing*, Lecture Notes, Heidelberg University, 2021.

[4] P. Getreuer. *Rudin-Osher-Fatemi Total Variaiton Denoising using Split Bergman*, Image Processing On Line, 2012.

[5] A. Beck, M. Treboulle. *Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems*, IEEE Transactions on Image Processing, November 2009.

[6] A. Beck, M. Teboulle. *A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[7] J. Liang, C. B. Schoenlieb. *Improving "Fast Iterative Shrinkage-Thresholding Algorithm": Faster, Smarter and Greedier*, SIAM Journal on Scientific Computing, 2022

[8] T. Karras et al. *Analyzing and Improving the Image Quality of StyleGAN*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[9] A. Chambolle, C. H. Dossal. *On the convergence of the iterates of "FISTA"*, Journal of Optimization Theory and Applications, Springer Verlag, 2015.