

ISTA (Iterative Shrinkage-Thresholding Algorithm)

M D Sacchi *

GEOPH 531 W-2023

1 Preliminaries

The process of solving inverse problems and various signal processing problems, such as signal reconstruction, frequently involves addressing the following optimization problem:

$$\begin{aligned}\hat{\mathbf{x}} &= \underset{\mathbf{x}}{\operatorname{argmin}} \{f(\mathbf{x}) + \lambda g(\mathbf{x})\} \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} \{\|\mathbf{Ax} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1\}.\end{aligned}\tag{1}$$

Optimization problems expressed in this form are commonly referred to as l_2 - l_1 problems. In statistical literature, the problem described by Equation 1 is often termed Basis Pursuit Denoising (BPDN). The objective is to minimize J , which represents the sum of two convex functions: the quadratic l_2 loss (misfit) and the l_1 regularization term. The principal goal involves recovering a sparse signal \mathbf{x} from the observations \mathbf{y} . These algorithms play a significant role in Compressive Sensing by enabling the recovery of signals compressed through a randomized sampling process [Baraniuk, 2007].

The problem can also be formulated as:

$$\min \|\mathbf{x}\|_1 \quad \text{subject to } \|\mathbf{Ax} - \mathbf{y}\|_2^2 \leq \delta \tag{2}$$

Equation 1 represents the unconstrained form of BPDN [Chen et al., 1998], while Equation 2 corresponds to the constrained form. The unconstrained form simplifies the adjustment of a single tradeoff parameter λ to produce sparse solutions where $\|\mathbf{Ax} - \mathbf{y}\|_2^2 \leq \delta$. The unconstrained problem enables derivation of constrained solutions while maintaining an

*email: msacchi@ualberta.ca

intuitive connection to classical Tikhonov regularization. The unconstrained form replaces the l_2 -norm regularization with an l_1 -norm, offering a simple Bayesian interpretation that the constrained form lacks.

1.1 ISTA solution

The general problem involves minimizing the function $J = f(\mathbf{x}) + \lambda g(\mathbf{x})$ where f and g , where both f and g are convex functions

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \{f(\mathbf{x}) + \lambda g(\mathbf{x})\} \quad (3)$$

The solution progresses to minimize the l_2 misfit along with an l_1 regularization term [Daubechies et al., 2004]. The function $f(\mathbf{x})$ is approximated as:

$$f(\mathbf{x}) \approx f(\mathbf{x}_k) + \nabla f_k^T (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2\eta} \|\mathbf{x} - \mathbf{x}_k\|_2^2 \quad (4)$$

where ∇f_k^T is the gradient of $f(\mathbf{x})$ at \mathbf{x}_k . Notice that if you take the derivative of the last equation and equate it to zero; you will get the classical steepest descent step for updating the variable \mathbf{x} . Hence, we can propose an algorithm that updates \mathbf{x} in equation 1 via

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ f(\mathbf{x}_k) + \nabla f_k^T (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2\eta} \|\mathbf{x} - \mathbf{x}_k\|_2^2 + \lambda g(\mathbf{x}) \right\}. \quad (5)$$

I can complete squares in the last expression and obtain the following

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ f(\mathbf{x}_k) + \frac{1}{2\eta} \|(\mathbf{x} - \mathbf{x}_k) + \eta \nabla f_k\|_2^2 - \frac{\eta}{2} \nabla f_k^T \nabla f_k + \lambda g(\mathbf{x}) \right\}. \quad (6)$$

Now, I only keep terms that depend on the variable \mathbf{x} (the others are constants that become zero after differentiation)

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \frac{1}{2\eta} \|(\mathbf{x} - \mathbf{x}_k) + \eta \nabla f_k\|_2^2 + \lambda g(\mathbf{x}) \right\}. \quad (7)$$

The last expression can be written as a denoising problem

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|(\mathbf{x} - \mathbf{u})\|_2^2 + \eta \lambda g(\mathbf{x}) \right\}, \quad (8)$$

where $\mathbf{u} = \mathbf{x}_k - \eta \nabla f_k$. The above is a denoising problem where one tries to approximate the vector \mathbf{u} by \mathbf{x} with an additional regularization $g(\mathbf{x})$. The proximal operator gives the

solution to equation 8. We generally choose g such that the solution reduces to a univariate minimization problem with an analytical answer. Equation 8 is written as follows

$$\mathbf{x}_{k+1} = \text{Prox}_{g, \lambda\eta}[\mathbf{u}] \quad (9)$$

$$= \text{Prox}_{g, \lambda\eta}[\mathbf{x}_k - \eta \nabla f_k]. \quad (10)$$

1.2 Proximal operator for $g(\mathbf{x}) = \|\mathbf{x}\|_1$

The proximal operator for $g(\mathbf{x}) = \|\mathbf{x}\|_1$ is named the soft-thresholding operator $\mathcal{S}_{\lambda\eta}$ and is the solution that minimizes

$$\mathcal{L} = \frac{1}{2} \|\mathbf{x} - \mathbf{v}\|_2^2 + a \|\mathbf{x}\|_1 = \frac{1}{2} \sum_i |x_i - v_i|^2 + a \sum_i |x_i|, \quad (11)$$

where $a > 0$. Setting $\frac{\partial \mathcal{L}}{\partial x_k} = 0$ leads to

$$x_k - v_k + a \text{sign}(x_k) = 0. \quad (12)$$

The latter can be split into

$$v_k = x_k - a \quad \text{if} \quad x_k < 0 \quad (13)$$

$$v_k = x_k + a \quad \text{if} \quad x_k > 0 \quad (14)$$

The last expression needs to be inverted because we need x_k as a function of v_k , an operation that can be carried out graphically by first plotting the last expression $v_k = h_a(x_k)$ and then graphically (Figure 1) finding $x_k = h_a^{-1}(v_k) = \mathcal{S}_a(v_k)$ which is the soft-thresholding operator

$$\mathcal{S}_a(v_k) = \begin{cases} v_k - a & v_k > a \\ 0 & |v_k| \leq a \\ v_k + a & v_k < -a \end{cases} \quad (15)$$

The operator can be written in a more compact form as follows

$$\mathcal{S}_a(v_k) = \text{sign}(v_k) \max(|v_k| - a, 0). \quad (16)$$

1.3 Recap ISTA

Let us go back to our original problem $f(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{y}\|_2^2$ and $g(\mathbf{x}) = \|\mathbf{x}\|_1$. Then, according to equations 10 and 15

$$\mathbf{x}_{k+1} = \mathcal{S}_{\lambda\eta}[\mathbf{u}] \quad (17)$$

$$= \mathcal{S}_{\lambda\eta}[\mathbf{x}_k - \eta \mathbf{A}^T(\mathbf{Ax}_k - \mathbf{y})]. \quad (18)$$

where the proximal operator (Soft thresholding, equation 15) is applied element-wise. The step length η satisfies $\eta < 1/\lambda_{max}$, where λ_{max} is the largest eigenvalue of $\mathbf{A}^T \mathbf{A}$, determined using the Power Method [Golub and Van Loan, 1996].

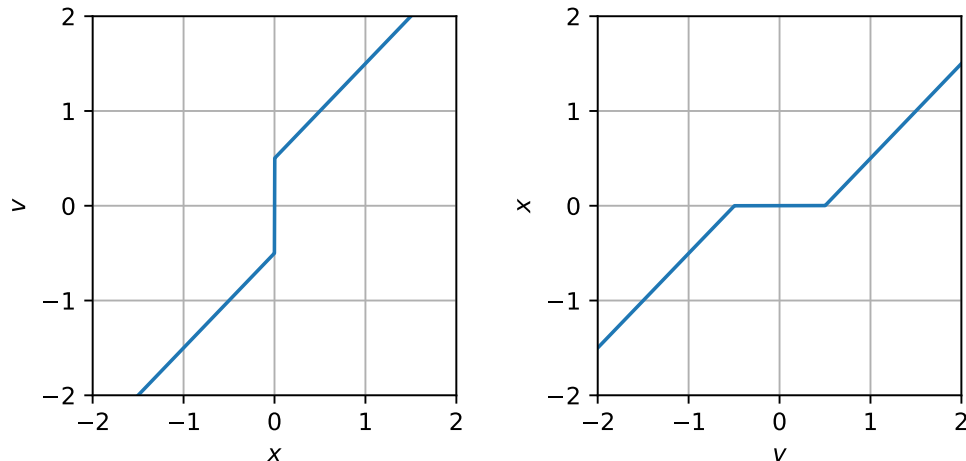


Figure 1: Left is equation 12, $v = h_a(x)$. Right is the soft thresholding operator $x = \mathcal{S}_a(v)$ (equation 15), $a = 0.5$.

2 Example

Figure 2 shows the inversion of a sparse sequence \mathbf{x} that has been compressed via a random matrix \mathbf{A} . The compressed data is given by $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ where \mathbf{y} has 40 points. The original signal \mathbf{x} has 150 points. This is an underdetermined problem, and we are exploiting the fact that \mathbf{x} is sparse to recover it from the measurement vector \mathbf{y} . I am comparing ISTA, FISTA (Fast-ISTA) [Beck and Teboulle, 2009] and IRLS [Sacchi et al., 1998]. Figure 2 provides the convergence curves of these three algorithms.

Notice that in the paper by Sacchi et al. [1998], IRLS is used to solve the Fourier sparse reconstruction problem via a Cauchy sparsity norm. A similar approach is used for multidimensional seismic signal reconstruction by Zwartjes and Gisolf [2007]. The last two references are a good starting point for understanding ND seismic data reconstruction as it is used today by seismic data processing contractors.

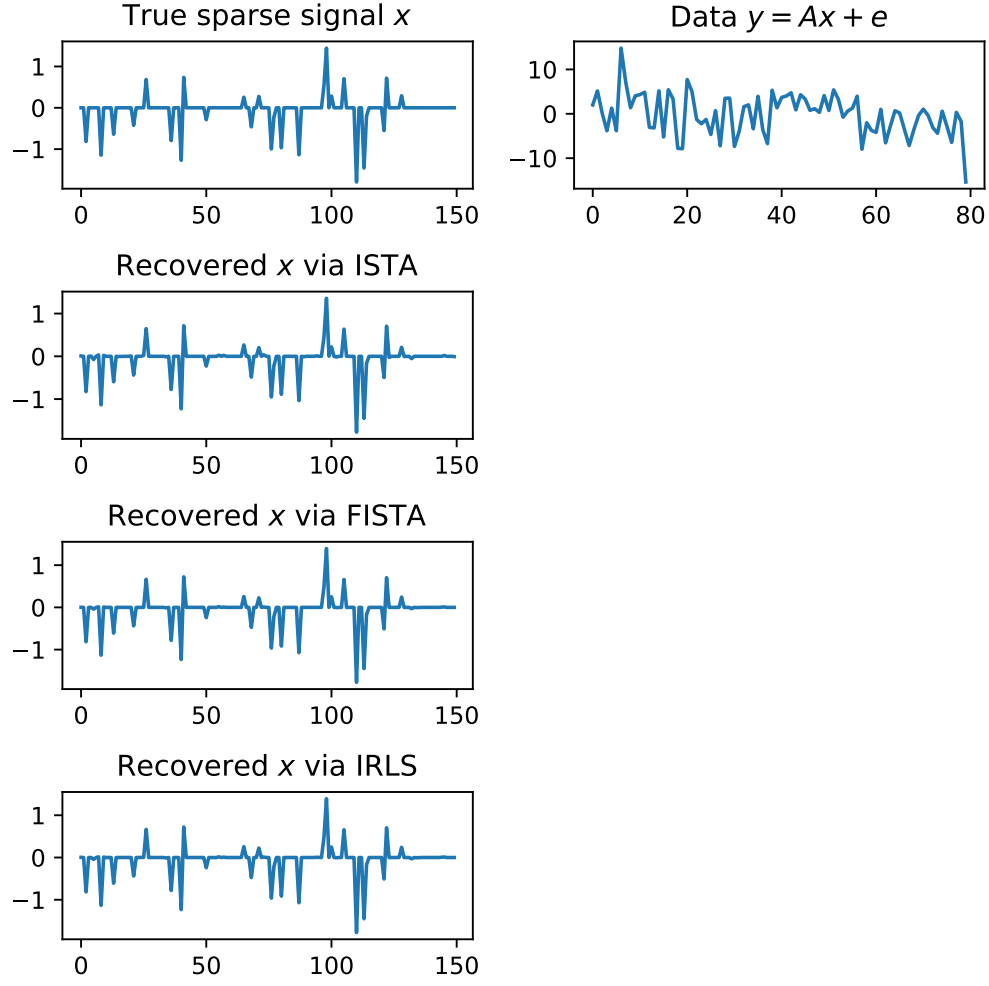


Figure 2: Inversions via ISTA, FISTA and IRLS.

	ISTA	FISTA	IRLS
$RMSE \times 100$	0.462	0.178	0.198

Table 1: Recovery error for the example in Figure 2 where $RMSE = \|\mathbf{x} - \mathbf{x}_{true}\|_2^2 / \|\mathbf{x}_{true}\|_2^2$.

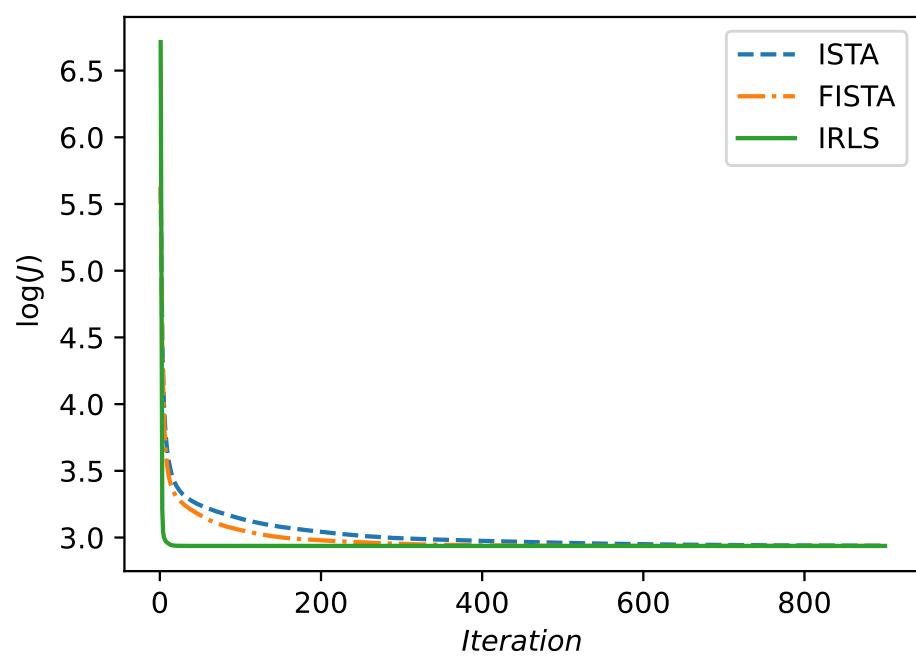


Figure 3: Convergence curves comparing ISTA, FISTA and IRLS.

3 ISTA Code

```
function ISTA(A,y,Niter,λ)
# ISTA solver. Finds x that minimizes
# J = 1/2||A x - y ||_2^2 + λ ||x||_1

    Soft(x,alpha) = sign(x)*max(abs(x)-alpha, 0)

    N,M = size(A)
    e = Power_Iteration(A) #
    η = 0.95/e

    x = zeros(Float64,M)

    J = zeros(Niter)
    for k = 1:Niter
        u = x .- η*A'*(A*x.-y)
        x = Soft.(u, η*λ)
        J[k] = 0.5*sum((A*x-y).^2) + λ*sum(abs.(x))
    end
    return x, J
end
```

References

- R. G. Baraniuk. Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007.
- A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.
- I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.
- G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

- M. Sacchi, T. Ulrych, and C. Walker. Interpolation and extrapolation using a high-resolution discrete fourier transform. *IEEE Transactions on Signal Processing*, 46(1):31–38, 1998.
- P. Zwartjes and A. Gisolf. Fourier reconstruction with sparse inversion. *Geophysical Prospecting*, 55(2):199–221, 2007.