Monte-Carlo Sure: A Black-Box Optimization of Regularization Parameters for General Denoising Algorithms

Raavi Gupta

IIT Bombay
Department of Electrical Engineering

Guide:
Prof. Satish Mulleti

EE779 Project
April 7, 2023

Table of Contents

Table of Contents

Introduction

The authors have proposed Stein's unbiased risk estimate (SURE) as a means of assessing the true mean-squared error (MSE) purely from the measured data without need for any knowledge about the noise-free signal. The method is a black-box approach which solely uses the response of the denoising operator to additional input noise

Notation Setup

We have the noisy data $y \in \mathbb{R}^n$ given by:

$$y = x + b \tag{1}$$

where $x \in \mathbb{R}^n$ represents the vector containing the samples of the unknown deterministic noise-free signal and $b \in \mathbb{R}^n$ denotes the vector containing zero-mean white Gaussian noise of σ^2 variance, respectively.

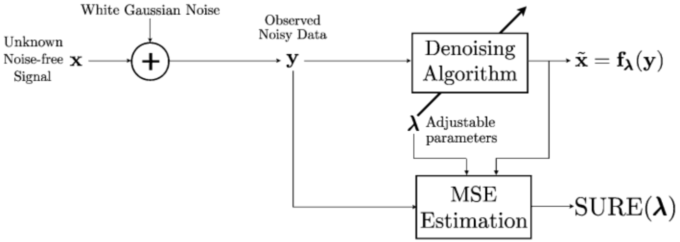
A denoising operator $f_\lambda : \mathbb{R}^n \to \mathbb{R}^n$ maps the input data y onto the signal estimate:

$$\tilde{x} = f_\lambda(y) \tag{2}$$

SURE corresponding to $f_\lambda(y)$ is a random variable $\eta : \mathbb{R}^n \to \mathbb{R}$ given by:

SURE

$$\eta(f_\lambda(y)) = \frac{||y - f_\lambda(y)||^2}{N} - \sigma^2 + \frac{2\sigma^2}{N}\text{div}_y\{f_\lambda(y)\}$$



Signal estimate $\tilde{x}$ is obtained by applying the λ-dependent denoising algorithm on the observed data y. The MSE box then computes the estimate SURE (λ) of the MSE between the noise-free x and the denoised as a function of λ, knowing only y and $f_\lambda(y)$.
Image source: [RBU08, Fig. 2]

Theorem (Source: [BL07])

The random variable $\eta(f_{\lambda(y)})$ *is an unbiased estimator of*

$$MSE(f_\lambda(y)) = \frac{||x - f_\lambda(y)||^2}{N} \tag{3}$$

that is

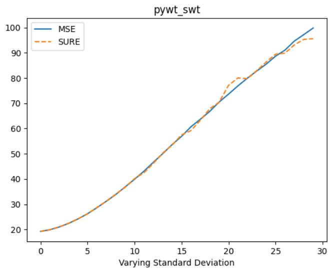
$$E_b\left(\frac{||x - f_\lambda(y)||^2}{N}\right) = E_b(\eta(f_{\lambda(y)})) \tag{4}$$

Algorithm for computing SURE(λ) for given $\lambda = \lambda_0$

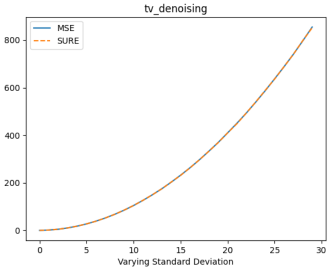
1: for $\lambda = \lambda_0$, evaluate $f_\lambda(y)$
2: Build $z = y + \epsilon b'$. Evaluate $f_\lambda(z)$ for $\lambda = \lambda_0$
3: Compute div $= \frac{b'^T(f_\lambda(z) - f_\lambda(y))}{N\epsilon}$ and SURE using the formula

Algorithm 1: b' is a zero-mean i.i.d. random vector of unit variance

Using the Denoising techniques provided in the paper



Variation of SURE and MSE using RSWST technique



Variation of SURE and MSE using Total-Variation Denoising (TVD) technique

Numerical data with different denoising algorithms

Average error between MSE and SURE		
Denoising Algorithm	Varying standard deviation	Varying epsilon
RSWST	0.88333	0.73407
TV Denoising	0.59852	0.43097
Bilateral Filter	0.23059	3.57725
Median Blurring	7856.20	8933.79125

Different ways of calculating divergence

1. Sobel's Operator
   It is a spatial filter that calculates the gradient of an image at each point, highlighting areas of rapid intensity change.

   Kernel used in x direction: $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

   Kernel used in y direction: $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

   The kernels are convolved with the image to calculate the gradient.
2. Employing central differences instead of forward difference for div calculations
3. Padding the image with a border in case of denoising operators like max/median blurring

Different ways of calculating divergence

1. Sobel's Operator
   It is a spatial filter that calculates the gradient of an image at each point, highlighting areas of rapid intensity change.

   Kernel used in x direction: $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

   Kernel used in y direction: $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

   The kernels are convolved with the image to calculate the gradient.
2. Employing central differences instead of forward difference for div calculations
3. Padding the image with a border in case of denoising operators like max/median blurring

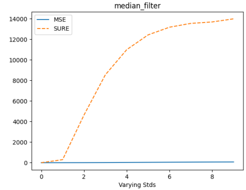
Different ways of calculating divergence

1. Sobel's Operator
   It is a spatial filter that calculates the gradient of an image at each point, highlighting areas of rapid intensity change.
   Kernel used in x direction: $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
   Kernel used in y direction: $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
   The kernels are convolved with the image to calculate the gradient.

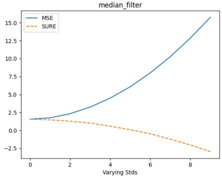
2. Employing central differences instead of forward difference for div calculations

3. Padding the image with a border in case of denoising operators like max/median blurring

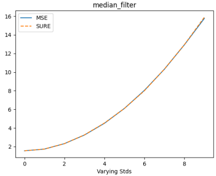
Comparison of results



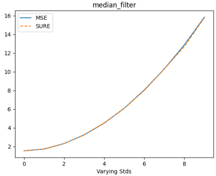
Using the algorithm only



Using Sobel operator



Using forward difference + padding



Using central difference + padding

References

[BL07] Thierry Blu and Florian Luisier. "The SURE-LET approach to image denoising". In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 16 (Dec. 2007), pp. 2778–86. DOI: 10.1109/TIP.2007.906002.

[RBU08] Sathish Ramani, Thierry Blu, and Michael Unser. "Monte-Carlo Sure: A Black-Box Optimization of Regularization Parameters for General Denoising Algorithms". In: *IEEE Transactions on Image Processing* 17.9 (2008), pp. 1540–1554. DOI: 10.1109/TIP.2008.2001404.

Thank You!