Stein's Unbiased Risk Estimator (SURE) Versus Maximum Likelihood (ML) in Inverse Problems

Toby Sanders

February 7, 2022

Abstract

A brief discussion on maximum likelihood and Stein's unbiased risk estimator in the context of inverse problems, and in particular image deconvolution.

A Perspective Based on Image Deconvolution

Assume blurry/noisy image data takes the form

$$b = h * v + \epsilon, \tag{1}$$

where $h \in \mathbb{R}^N$ is the PSF, $v \in \mathbb{R}^N$ is the image we want to estimate, and $\epsilon \sim N(0, \sigma^2 I)$. Any convolution $h * u$ we will also write as Hu, where H is a square circulant matrix. Since the noise is i.i.d. Gaussian, the ML estimator takes the form

$$\begin{aligned} u^*_{ML} &= \arg\max_u p(b|u) \\ &= \arg\max_u \exp\left(-\frac{1}{2\sigma^2}\|Hu - b\|_2^2\right) \\ &= \arg\min_u \|Hu - b\|_2^2. \end{aligned} \tag{2}$$

For any arbitrary estimate of v given by u^*, SURE provides the estimator for the blurred square error given by [1, 2]

$$\begin{aligned} SURE(u^*) &= \mathbb{E}\left[\|H(u^* - v)\|_2^2\right] \\ &= -N\sigma^2 + \|Hu - b\|_2^2 + 2\sigma^2 \sum_{j=1}^{N} \frac{\partial}{\partial b_j}(Hu)_j. \end{aligned} \tag{3}$$

Clearly, minimizing SURE given in (3) is not equivalent to the ML estimator in (2). This may be observed simply because SURE already contains the squared error term from the ML estimator, but it also includes the sum containing the partial derivatives. However, we want to understand their difference on a more philosophical level.

First note that ML simply works to match the data as closely as possible by minimizing the squared error. The solution (without any concern for dividing by zero) is given by multiplication with the pseudo inverse

$$u^*_{ML} = (H^\mathsf{T} H)^{-1} H^\mathsf{T} b. \tag{4}$$

This solution is that which effectively deconvolves by dividing by the PSF. It will return back the best match to the blurry data image, but will result in awful artifacts in the image due to the noise term. Effectively, $(H^\mathsf{T} H)$ contains eigenvalues very, very close to zero, so that $(H^\mathsf{T} H)^{-1}$ *blows-up* (see section below), making it very sensitive to changes in the data (a noise term).

SURE on the other hand works to minimize the statistical estimate of the (blurred) squared error of the recovered solution and the *true* underlying image, not the measured data. This inherently takes into account the nature of the inverse method applied and how sensitive this solution is to change in the data, i.e. random noise. This manifests in the last term of SURE, namely the term from (3) given by

$$\sum_{j=1}^{N} \frac{\partial}{\partial b_j}(Hu)_j. \tag{5}$$

Observe, this is precisely a measure of the sensitivity of the solution is to the data!!! Furthermore, the larger the noise is (σ), the larger this penalty is, which is seen by the $2\sigma^2$ coefficient in front of the sum in (3). So SURE may be seen in this light as including a penalty for inverse methods that are overly sensitive to perturbations in the data (indeed they should not be overly sensitive, right?). In conclusion, the SURE estimator manifests as a balancing act between fitting the data (the least squares data fit term as in the ML estimator), and reducing the sensitivity to the noise (the term in (5)).

Finally, for completeness, we mention the maximum a posteriori estimator (MAP), which is given by

$$\begin{aligned} u^*_{MAP} &= \arg\max_u p(u|b) \\ &= \arg\max_u p(b|u)p(u) \\ &= \arg\max_u \exp\left(-\frac{1}{2\sigma^2}\|Hu-b\|_2^2\right) p(u) \\ &= \arg\min_u \frac{1}{2\sigma^2}\|Hu-b\|_2^2 - \log p(u). \end{aligned} \tag{6}$$

This works in a similar fashion as SURE, by including a penalty term $-\log p(u)$. This will penalize *bad* or *undesirable* images. It can be a challenge to model $-\log p(u)$ (e.g. total variation) and scale it properly with the $p(b|u)$ term. The approach I often use is to evaluate MAP estimators for different parameters and choose the MAP estimator which minimizes SURE, thus ensuring we have optimized the MAP parameters. Indeed, this is precisely the method I used in my recent SURE-based PSF estimation paper.

Basic Analysis of the Sensitivity of Inverse Problems

Let's suppose more generally the data takes the form

$$b = Av + \epsilon, \tag{7}$$

where now A is an arbitrary matrix operator. Let the SVD of A be given by

$$A = USV^{\mathsf{T}} = \sum_{j=1}^{r} \sigma_j u_j v_j^{\mathsf{T}}, \tag{8}$$

where r is the rank of A, and u_j and v_j make up the columns of U and V respectively, and S is a diagonal containing the values σ_j. Let's assume for the sake of the discussion that A has fewer columns than rows, so that the pseudo inverse of A is given by

$$\begin{aligned} A^+ &= (A^{\mathsf{T}}A)^{-1}A^{\mathsf{T}} \\ &= (VS^{\mathsf{T}}SV^{\mathsf{T}})^{-1}VS^{\mathsf{T}}U^{\mathsf{T}} \\ &= V(S^{\mathsf{T}}S)^{-1}V^{\mathsf{T}}VS^{\mathsf{T}}U^{\mathsf{T}} \\ &= V(S^{\mathsf{T}}S)^{-1}S^{\mathsf{T}}U^{\mathsf{T}} \\ &= \sum_{j=1}^{r} \sigma_j^{-1} v_j u_j^{\mathsf{T}}. \end{aligned} \tag{9}$$

Combining this with (7) and (8), the naive ML solution to the inverse problem is

$$\begin{aligned} u^*_{ML} &= A^+ b = (A^+ A v + \epsilon) \\ &= \left(\sum_{j=1}^{r} \sigma_j^{-1} v_j u_j^{\mathsf{T}} \right) \left(\sum_{j=1}^{r} \sigma_j u_j v_j^{\mathsf{T}} \right) v + \left(\sum_{j=1}^{r} \sigma_j^{-1} v_j u_j^{\mathsf{T}} \right) \epsilon \\ &= \left(\sum_{j=1}^{r} v_j v_j^{\mathsf{T}} \right) v + \left(\sum_{j=1}^{r} \sigma_j^{-1} v_j u_j^{\mathsf{T}} \right) \epsilon \end{aligned} \tag{10}$$

Observe that the first term in the last expression returns the part of the solution u that is in the range of A, while the second term returns some additional noise term. The real issue is that small singular values σ_j will cause the term to *blow-up* in an awful way due to the appearance of their inverses.

References

[1] T. Sanders. Notes on automated parameter selection for regularization methods in image and signal reconstruction. 2019.

[2] F. Xue and T. Blu. A novel SURE-based criterion for parametric PSF estimation. *IEEE Transactions on Image Processing*, 24(2):595–607, 2014.