Notes on automated parameter selection for regularization methods in image and signal reconstruction

Toby Sanders

sandertl20@gmail.com

August 16, 2019

Abstract

A terse introduction to parameter selection for regularization in image reconstruction is provided. The methods known as unbiased predictive risk estimator (UPRE) and Stein's unbiased risk estimator (SURE) are explained and derived. A simple matlab implementation of UPRE is also given.

Introduction

A common formulation of an inverse problem is stated as recovering a signal or image $u \in \mathbb{R}^n$ from acquired measurements of the form $b = Au + \xi$, where $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{R}^m$ and ξ is a noise term. A maximum likelihood method for recovering u from b is a method which maximizes the likelihood of the data given the recovered solution u, $p(b|u)$. This leads to methods such as least squares:

$$u^* = \arg\min_u \frac{1}{2}\|Au - b\|_2^2.$$

The solution to this problem is not unique whenever the linear system under-determined (the number of linear independent rows in A is less than n), and in general these problems are ill-posed. For these reasons, it is very common to introduce regularized methods (considered maximum a posteriori methods that maximize $p(u|b)$ instead), which may take the form

$$u_\lambda = \arg\min_u \frac{1}{2}\|Au - b\|_2^2 + \lambda R(u). \tag{1}$$

The regularization term $R(u)$ is a prior term that promotes regularized or *smooth* solutions that become less sensitive to noise. The data fitting term in the above equations can be modified to account for the noise model, though these ℓ_2 models are appropriate for Gaussian white noise models. Figure 1 highlights the effectiveness of a famous regularization method known as total variation (TV), where an example of an electron tomography image reconstruction is shown with (right) and without the regularization [2].

Now, if you find yourself presenting such methods at a conference or submitting your work to a journal, then you will inevitably be asked "how did you choose lambda?" This is a famous question, and the most common answer is of course, "based on experience." People love to ask this question, but rarely receive a good answer, because there were probably a million other problems to work on before the author got around to choosing this hyper-parameter. Selecting the parameter in a rigorous way is a full-blown research problem on its on. Some effective methods exist, and I have tried to outline two of these below in an understandable and meaningful way. Outlined are the unbiased predictive risk estimator (UPRE) and Stein's unbiased risk estimator (SURE) [4], which are closely related.

A broader overview of methods is given in the book of Vogel [5], and another good literature review is provided in the paper of Ramani, et al [1].

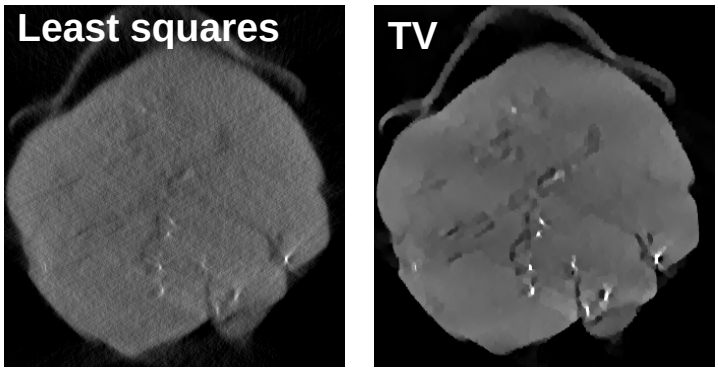

Figure 1: Comparison of a reconstructed tomography image with (right) and without (left) regularization.

The basic idea for UPRE and SURE

Let the true solution to the inverse problem simply be denoted by u, and consider solutions as functions of the regularization parameter λ given by $u_\lambda = u(\lambda)$ as written in (1). Then a reasonable criteria for choosing the right λ can be given by minimizing a loss function $F(\lambda) = L(u, u_\lambda)$, where L is a metric giving us some difference or loss between u and u_λ, e.g. $\|u - u_\lambda\|_2^2$. If we can estimate $F(\lambda)$ for any λ, then all we need to do is minimize F. At first it may not seem possible that we can do this since we obviously do not know the true solution u, however UPRE and SURE provide us with estimators of such quantities. In other words they provide *expected values* for the loss function that can be evaluated, though not the true loss.

One important assumption is necessary for the derivation of these estimators[1], that is the knowledge that the noise in the data vector is mean zero i.i.d. Gaussian with variance σ^2, i.e. $b \sim N(Au, \sigma^2 I)$, where the variance σ^2 is assumed to be known. However, if we do not know σ, then [3] provides an iterative method to estimate this variable, which empirically converges in very few iterations (e.g. 5 or less). Before deriving the methods, we first provide statements and explanations of the estimators below.

In the case of UPRE, we must assume the regularized solution is given by some linear transform of the data $u_\lambda = B_\lambda b$, e.g. for Tikhonov regularization with $R(u) = \frac{1}{2}\|Tu\|_2^2$, then $u_\lambda = (A^\mathsf{T}A + \lambda T^\mathsf{T}T)^{-1}A^\mathsf{T}b$. In this case, the error estimator is given by

$$F(\lambda) = \mathbb{E}\|A(u - u_\lambda)\|_2^2 = -m\sigma^2 + \|Au_\lambda - b\|_2^2 + 2\sigma^2 \text{trace } (AB_\lambda) \tag{2}$$

Hence, if we can minimize the above expression with respect to λ, then we have achieved an *optimal* solution for that regularization form. Figure 2 shows an example of the accuracy of the UPRE estimator on a small test problem where the true solution was known.

The SURE method is more general because it is not derived from inverse models whose solution is given by a linear transform. The estimate from SURE is given by

$$F(\lambda) = \mathbb{E}\|A(u - u_\lambda)\|_2^2 = -m\sigma^2 + \|Au_\lambda - b\|^2 + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial (Au_\lambda)_i}{\partial b_i}. \tag{3}$$

Notice the UPRE solution can then be derived directly from (3).

Though SURE is equivalent to UPRE for problems where the solution can be written with a linear transform, SURE provides an estimator for other types of models. Perhaps most notably it can be used to derive an estimator for ℓ_1 regularization problems (as opposed to Tikhonov ℓ_2 regularizations). See for example this case for the famous LASSO problem [6].

[1] It may still be possible to apply the approaches in case the assumption is not completely satisfied.

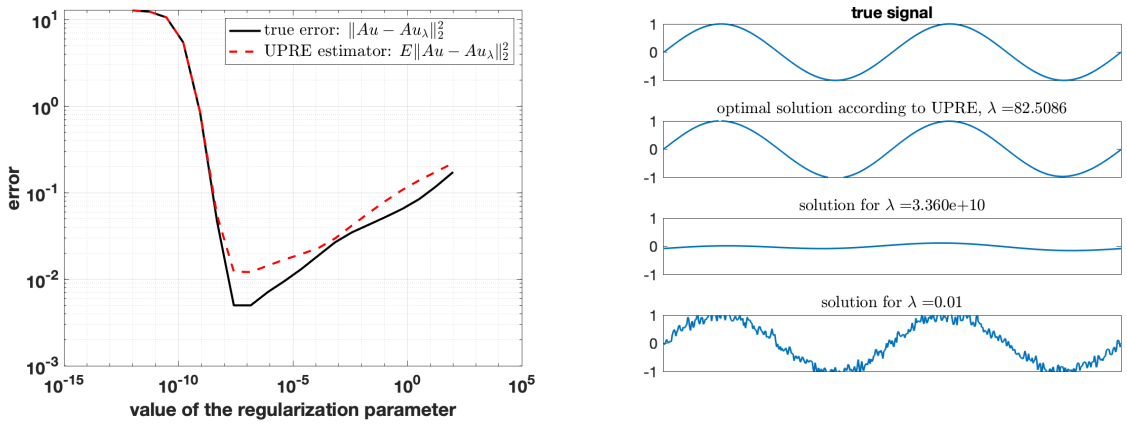

Figure 2: A test problem showing that the estimator given by UPRE is accurate to the true error (left plot), and the minimizing parameter values are nearly the same. On the right the true solution is shown with several of the reconstructions for different values of λ. The code to reproduce these images is provided after the references.

Alternatively, one can use the optimal ℓ_2 parameter found using a simpler UPRE approach to obtain a suitable ℓ_1 parameter through the Bayesian formulation [3]. This approach is attractive due to its simplicity.

Computational Hurdles

Evaluating (2) for the general case can be difficult due to the trace term for large scale imaging problems. In particular, B_λ is rarely ever computed and stored in memory. Moreover the product of this matrix with the A matrix would be a huge calculation, far more than computing a solution. This difficulty can be circumvented using randomized statistical methods.

For example, for any square matrix M if x is a random vector with independent entries mean zero and variance one, then $\mathbb{E}[x^\mathsf{T} Mx] = \text{trace}(M)$. Hence, if we'd like to find the trace of some product of several matrices, say A, B, C (also including inverses and so forth), though it is unreasonable to compute the product of the matrices to find the trace, in principle we should be able to compute $x^\mathsf{T} ABCx$, with only three matrix vector multiplies. If we do this several times for random vectors x and average the result, then we should obtain a good approximation for the trace. Yet still this calculation can be as costly as computing the solution u_λ, since say for Tikhonov regularization we need the trace of AB_λ, and $B_\lambda x = (A^\mathsf{T} A + \lambda T^\mathsf{T} T)^{-1} x$ is computed with an iterative method the same way the solution is computed. So it really depends on the application and available computing power to determine if this is a useful approach.

Perhaps even more useful, for many important problems, such as denoising, deconvolution, and Fourier reconstruction, the trace can even be computed analytically as shown in [3], which provides a massive reduction in computation for Tikhonov regularization. In my opinion, this really makes the approach most appealing for these types of applications, since it requires only very careful coding and hand calculations, but in turn very few flops.

A simple derivation of UPRE

Here equation (2) is proven. Let $u_\lambda = B_\lambda b$ and $b = Au + \xi$. Then

$$\begin{aligned}\mathbb{E}\|Au_\lambda - Au\|_2^2 &= \mathbb{E}\|Au_\lambda - (b-\xi)\|_2^2 \\ &= \|Au_\lambda - b\|_2^2 + \mathbb{E}\|\xi\|_2^2 + 2\mathbb{E}\langle \xi, Au_\lambda - b\rangle \\ &= \|Au_\lambda - b\|_2^2 + m\sigma^2 + 2\mathbb{E}\langle \xi, Au_\lambda - b\rangle\end{aligned}$$

Now we have to work a bit more to evaluate the last expectation: replace $Au_\lambda - b = (AB_\lambda - I)b$, and then $b = Au + \xi$ to obtain

$$\begin{aligned}\mathbb{E}\langle \xi, Au_\lambda - b\rangle &= \mathbb{E}\langle \xi, (AB_\lambda - I)(Au+\xi)\rangle \\ &= \mathbb{E}\langle \xi, (AB_\lambda - I)Au\rangle + \mathbb{E}\langle \xi, (AB_\lambda)\xi\rangle - \mathbb{E}\|\xi\|_2^2 \\ &= \sigma^2 \mathrm{trace}(AB_\lambda) - m\sigma^2\end{aligned}$$

In the last line, we have used the simple fact that for any matrix M and i.i.d normal vector ξ with variance σ^2, that $\mathbb{E}[\xi^\mathsf{T} M\xi] = \sigma^2 \mathrm{trace}(M)$. Combining this result with the first set of equations completes the derivation.

Estimation of the noise variance, σ^2

The methods of UPRE and SURE rely on knowledge of the noise variance σ^2, while assuming the distribution to be i.i.d. Gaussian. However, in [3] a method was proposed for estimating the variance and empirically shown to converge very quickly. For UPRE, it relies on the following fact: the value of σ which maximizes the probability of b in terms of expectations satisfies the inequality

$$\sigma^2 = \|Au_\lambda - b\|_2^2/(m - \mathrm{trace}(AB_\lambda)).$$

Based on this equality, it was suggested to use a fixed point method for σ along with the combination of the optimization of λ to be given by

$$\sigma_{k+1}^2 = \|Au(\lambda_k) - b\|_2^2/[m - \mathrm{trace}(AB(\lambda_k))].$$

It is also possible to derive a fixed point method for λ from UPRE as well. We plan to describe this in more detail in later work.

Derivation of SURE

Let $A \in \mathbb{R}^{m\times n}$, $u \in \mathbb{R}^n$, $b \sim N(Au, \sigma^2 I)$. Let us start by considering a denoising problem, so that $b = u + \xi$, $\xi \sim N(0, \sigma^2 I)$. Let $u_\lambda = u_\lambda(\xi)$ be an estimator of u, dependent upon the parameter λ, as a function of the random variable ξ. Then to obtain an estimator for $\|u - u_\lambda\|_2^2$ begin with the following:

$$\begin{aligned}\mathbb{E}\|u - u_\lambda\|^2 &= \mathbb{E}\|u - b + b - u_\lambda\|_2^2 \\ &= \mathbb{E}\|u-b\|_2^2 + \|b - u_\lambda\|^2 + 2\mathbb{E}(u-b)^\mathsf{T}(b-u_\lambda) \\ &= n\sigma^2 + \|b-u_\lambda\|^2 - 2\mathbb{E}\,\xi^\mathsf{T}(b-u_\lambda) \\ &= -n\sigma^2 + \|b-u_\lambda\|^2 + 2\mathbb{E}\,\xi^\mathsf{T} u_\lambda,\end{aligned} \tag{4}$$

where, among other things, we have used $\mathbb{E}\,\xi^\mathsf{T} b = n\sigma^2$. So we only need a statistical estimate for $\mathbb{E}[\xi^\mathsf{T} u_\lambda(\xi)]$ to complete the estimator. This is where we need the help of Stein [4], whose lemma says that for a random variable $X \sim N(\mu, \sigma^2)$ that $\mathbb{E}[(X-\mu)f(X)] = \sigma^2\mathbb{E}[f'(X)]$, which

for simple functions can be easily derived using integration by parts. This leads us to

$$\mathbb{E}[\xi^{\mathsf{T}} u_\lambda(\xi)] = \sum_{i=1}^{n} \mathbb{E}\, \xi_i u_\lambda(\xi)_i = \sigma^2 \sum_{i=1}^{n} \frac{\partial u_\lambda(\xi)_i}{\partial \xi_i} = \sigma^2 \mathrm{div} u_\lambda(\xi).$$

This leads to the SURE estimator as

$$F(\lambda) = \mathbb{E}\|u - u_\lambda\|^2 = -n\sigma^2 + \|b - u_\lambda\|^2 + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial u_\lambda(\xi)_i}{\partial \xi_i},$$

and an optimal λ satisfies

$$\lambda^* = \arg\min_\lambda \|b - u_\lambda\|^2 + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial u_\lambda(\xi)_i}{\partial \xi_i}$$

If we're not in the denoising case, and $b = Au + \xi$ for general A, then the same ideas lead us to an alternative SURE estimator as

$$\hat{R} = \mathbb{E}\|Au - Au_\lambda\|^2 = -n\sigma^2 + \|b - Au_\lambda\|^2 + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial Au_\lambda(\xi)_i}{\partial \xi_i}.$$

If $u_\lambda = (A^{\mathsf{T}}A + \lambda T^{\mathsf{T}}T)^{-1}A^{\mathsf{T}}b$, the Tikhonov solution, then

$$\hat{R} = -n\sigma^2 + \|b - Au_\lambda\|^2 + 2\sigma^2 \mathrm{trace}(AH^{-1}A^{\mathsf{T}}),$$

which is UPRE. Notice that everywhere that u_λ is written as a function of ξ and differentiated w.r.t. ξ, can be replaced with b and changes nothing. In other words

$$\hat{R} = \mathbb{E}\|Au - Au_\lambda\|^2 = -n\sigma^2 + \|b - Au_\lambda\|^2 + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial Au_\lambda(b)_i}{\partial b_i}$$

References

[1] S. Ramani, Z. Liu, J. Rosen, J. Nielsen, and J. A. Fessler. Regularization parameter selection for nonlinear iterative image restoration and MRI reconstruction using GCV and SURE-based methods. *IEEE Transactions on Image Processing*, 21(8):3659–3672, 2012.

[2] T. Sanders and I. Arslan. Improved three-dimensional (3D) resolution of electron tomograms using robust mathematical data-processing techniques. *Microscopy and Microanalysis*, 23(6):1121–1129, 2017.

[3] T. Sanders, R. B. Platte, and R. D. Skeel. Maximum evidence algorithms for automated parameter selection in regularized inverse problems. *arXiv preprint arXiv:1812.11449*, 2018.

[4] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, pages 1135–1151, 1981.

[5] C. R. Vogel. *Computational methods for inverse problems*, volume 23. SIAM, 2002.

[6] H. Zou, T. Hastie, R. Tibshirani, et al. On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.


```
% simple small example for UPRE
% Important note: this example is for demonstration purposes only. In
```

```
% practice one cannot compute matrix inverses, traces, etc. exactly, and
% must instead uses iterative methods
% A "**" comment in the code below incates a line of code that is only
% reasonable for small problems such as this example.

% written by Toby Sanders
% August 11, 2019

clear;
d = 500; % signal dimension
m = d; % number of samples
SNR = 5; % signal to noise ratio
k = 3; % order of the regularizer
mus = linspace(-12,2,20);  % log of test value for mu
mus = 10.^mus;
rng(705); % seed for reproducibility

% construct test signal and matrix
x = sin(4*pi*linspace(0,1,d)');   % signal
A = randn(m,d)/100; % sampling matrix with normal entries
b = A*x;  % data vector
sigma = mean(abs(b(:)))/SNR; % standard deviation
b = b+randn(numel(b),1)*sigma; % add noise to data

% set up reconstruction problem
Ax = A*x; % save Ax to compute true error
AtA = A'*A;  % save A transpose A
T = zeros(d);  % T is the kth order regularization operator
T(1:d+1:end) = -1; T(d+1:d+1:end) = 1; T(end,1) = 1;
T = T^k/(2^(k-1)); % scale the matrix
TtT = T'*T;

% loop over mus and obtain estimator
recs = zeros(d,numel(mus));
recs2 = recs;
terr = zeros(numel(mus),1);Eerr = terr;
for i = 1:numel(mus)
    tik.mu = mus(i);
    H = AtA+TtT/tik.mu;
    Hi = inv(H);  % **
    % compute solution for current parameter
    recs2(:,i) = H\(A'*b); % **
    terr(i) = norm(Ax-A*recs2(:,i),2)^2; % **
    % evaluate the estimator
    Eerr(i) = -m*sigma^2 ...
        + norm(A*recs2(:,i)-b,2)^2 ...
        + 2*sigma^2*trace(AtA*Hi);  % **
end
[mmval,mm] = min(Eerr); % best solution

%% plot results
figure(87);hold off;
loglog(mus,terr,'k','linewidth',2);hold on;
loglog(mus,Eerr,'r--','linewidth',2);
legend({'true error: $\| A u - Au_\lambda\|_2^2$',...
    'UPRE estimator: $E \| A u - Au_\lambda\|_2^2$'},...
    'interpreter','latex','fontsize',16,'fontweight','bold');
xlabel('value of the regularization parameter');
```

```
ylabel('error');
set(gca,'fontweight','bold','fontsize',16);
grid on;hold off;

figure(88);
subplot(4,1,1);plot(x,'linewidth',1.5);title('true signal');
set(gca,'fontsize',14,'Xtick',[]);axis([0 d -1 1]);
subplot(4,1,2);plot(recs2(:,mm),'linewidth',1.5);
title(['optimal solution according to UPRE, $\lambda = $',...
    num2str(1./mmval)],'interpreter','latex');
set(gca,'fontsize',14,'Xtick',[]);axis([0 d -1 1]);
subplot(4,1,3);plot(recs2(:,3),'linewidth',1.5);
title(['solution for $\lambda = $',num2str(1./mus(3),'%10.3e')],...
    'interpreter','latex');
set(gca,'fontsize',14,'Xtick',[]);axis([0 d -1 1]);
subplot(4,1,4);plot(recs2(:,end),'linewidth',1.5);
title(['solution for $\lambda = $',num2str(1./mus(end))],...
    'interpreter','latex');
set(gca,'fontsize',14,'Xtick',[]);axis([0 d -1 1]);
```