Topic 10. How to Choose Algorithmic Parameter in Splines: Generalized Cross Validation

Spring 2018

1 Introduction

Recall we have considered

$$\min_s J_\alpha(s),$$

where $s(x)$ is a cubic smoothing spline and

$$\begin{aligned} J_\alpha(s) &= \alpha \sum_{i=0}^{n} w_i[y_i - s(x_i)]^2 + (1-\alpha)\int_{x_0}^{x_n} s''(x)^2 dx \\ &= \alpha(\mathbf{y}-\mathbf{f})^T \mathbf{W}(\mathbf{y}-\mathbf{f}) + (1-\alpha)\mathbf{f}^T \mathbf{Q}^T \mathbf{M}^{-1}\mathbf{Q}\mathbf{f}. \end{aligned}$$

Recall we have

$$\mathbf{M}\cdot\sigma = \mathbf{Q}\cdot\mathbf{f},$$

where matrices $\mathbf{M}$ and $\mathbf{Q}$ are defined earlier. We gave the result:

$$\hat{\mathbf{f}} = [\alpha\mathbf{W} + (1-\alpha)\mathbf{Q}^T\mathbf{M}^{-1}\mathbf{Q}]^{-1}\alpha\mathbf{W}\mathbf{y}.$$

Remark: the above can be implemented numerically. One open question is how to choose the value of the algorithmic parameter α.

2 Cross Validation

For notational convenience, we reformulate the above optimization problem. We denote

$$h(\lambda) = \text{argmin}_s \sum_{i=0}^{n} w_i[y_i - s(x_i)]^2 + \lambda\int_{x_0}^{x_n} s''(x)^2 dx, \quad \lambda > 0,$$

where s is a splines with knots $x_0 < x_1 < \cdots < x_n$. In addition, we define that

$$h_{-k}(\lambda) = \text{argmin}_s \sum_{i=0, i\neq k}^{n} w_i[y_i - s(x_i)]^2 + \lambda\int_{x_0}^{x_n} s''(x)^2 dx, \quad 1 \le k \le n;$$

i.e., $h_{-k}(\lambda)$ is the minimum spline when the kth knot is ignored. The following function is called the *cross validation* criterion function:

$$\mathrm{CV}(\lambda) = \sum_{k=0}^{n} [y_k - h_{-k}(\lambda; x_k)]^2,$$

where $h_{-k}(\lambda; x_k)$ is the value of $h_{-k}(\lambda)$ at knot x_k. We choose λ such that $\mathrm{CV}(\lambda)$ is minimized.

If $\hat{f}(\lambda)$ denote the vector that contains the values of $h(\lambda)$ at $x_0, \ldots, x_n$, then we have

$$\hat{f}(\lambda) = \mathbf{S}(\lambda) \cdot \mathbf{y},$$

where

$$\mathbf{S}(\lambda) = [\mathbf{W} + \lambda \mathbf{Q}^T \mathbf{M}^{-1} \mathbf{Q}]^{-1} \mathbf{W}.$$

Similarly, the values of $h_{-k}(\lambda)$ at $x_0, \ldots, x_n$, form a vector $\hat{f}_{-k}(\lambda)$.

We can easily verify the following inequalities:

$$\begin{aligned} & \sum_{i=0, i\neq k}^{n} w_i [y_i - s(x_i)]^2 + w_k [h_{-k}(\lambda; x_k) - s(x_k)]^2 + \lambda \int_{x_0}^{x_n} s''(x)^2 dx \\ \geq \quad & \sum_{i=0, i\neq k}^{n} w_i [y_i - s(x_i)]^2 + \lambda \int_{x_0}^{x_n} s''(x)^2 dx \\ \geq \quad & \sum_{i=0, i\neq k}^{n} w_i [y_i - h_{-k}(x_i)]^2 + \lambda \int_{x_0}^{x_n} h''_{-k}(x)^2 dx. \end{aligned}$$

The above leads to the following fact: if you replace y_k by $h_{-k}(\lambda; x_k)$ and fit a smoothing spline with the same parameter λ, then the fitted spline is h_{-k}. Hence we have the following:

$$\hat{f}_{-k}(\lambda) = \mathbf{S}(\lambda) \cdot \begin{pmatrix} y_0 \\ \vdots \\ y_{k-1} \\ h_{-k}(\lambda; x_k) \\ y_{k+1} \\ \vdots \\ y_n \end{pmatrix}.$$

On the other hand, recall we have

$$\hat{f}(\lambda) = \mathbf{S}(\lambda) \cdot \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}.$$

Taking difference of the above two, we can have

$$\hat{f}(\lambda)_k - h_{-k}(\lambda; x_k) = \hat{f}(\lambda)_k - (\hat{f}_{-k}(\lambda))_k = \mathbf{S}(\lambda)_{kk} [y_k - h_{-k}(\lambda; x_k)].$$

The above leads to the following equivalent equations:

$$\begin{aligned} & \frac{\hat{f}(\lambda)_k - h_{-k}(\lambda; x_k)}{y_k - h_{-k}(\lambda; x_k)} = \mathbf{S}(\lambda)_{kk} \\ \Leftrightarrow \quad & \frac{y_k - \hat{f}(\lambda)_k}{y_k - h_{-k}(\lambda; x_k)} = 1 - \mathbf{S}(\lambda)_{kk} \\ \Rightarrow \quad & [y_k - h_{-k}(\lambda; x_k)]^2 = \frac{[y_k - \hat{f}(\lambda)_k]^2}{[1 - \mathbf{S}(\lambda)_{kk}]^2}. \end{aligned}$$

Hence we have

$$\mathrm{CV}(\lambda) = \sum_{k=0}^{n} [y_k - h_{-k}(\lambda; x_k)]^2 = \sum_{k=0}^{n} \frac{[y_k - \hat{f}(\lambda)_k]^2}{[1 - \mathbf{S}(\lambda)_{kk}]^2}. \tag{2.1}$$

3 Generalized Cross Validation

The advantage of having (2.1) is that as long as we have $\mathbf{S}(\lambda)$, we can easily compute for $\mathrm{CV}(\lambda)$. A disadvantage is that the denominator may fluctuate. In particular when $\mathbf{S}(\lambda)_{kk}$ is close to 1, the denominator approaches to zero. A remedy of the above is to replace $\mathbf{S}(\lambda)_{kk}$ by

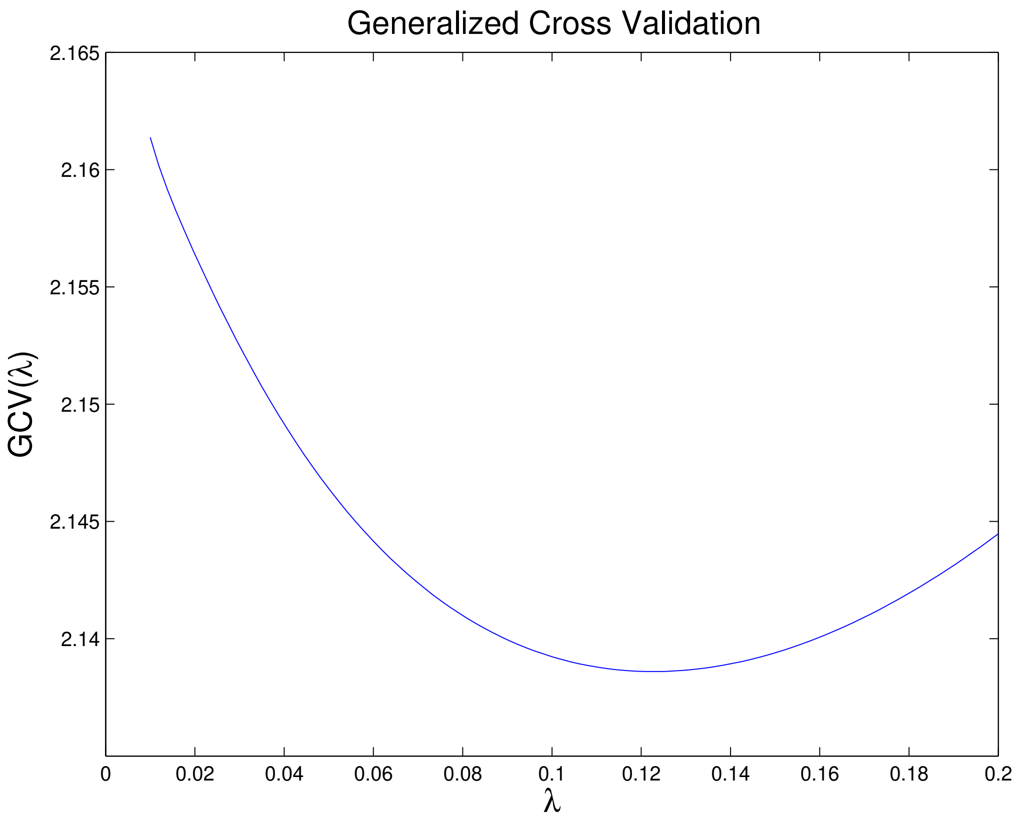
$$\frac{\sum_{k=0}^{n} \mathbf{S}(\lambda)_{kk}}{n+1} = \mathrm{Tr}(\mathbf{S}(\lambda))/(n+1),$$

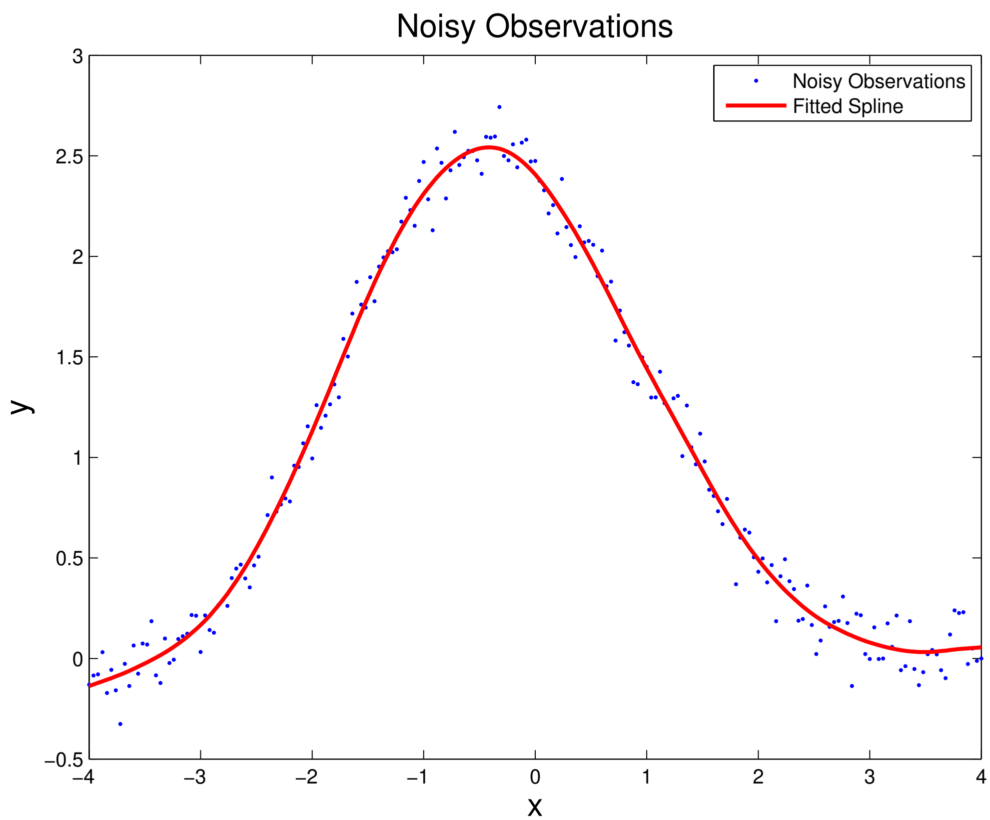
where Tr stands for the trace of a matrix. The *generalized cross validation* principle is to find a λ that minimizes the criterion function below:

$$\mathrm{GCV}(\lambda) = \sum_{k=0}^{n} \frac{[y_k - \hat{f}(\lambda)_k]^2}{[1 - \frac{\mathrm{Tr}(\mathbf{S}(\lambda))}{(n+1)}]^2}.$$

4 Numerical Example

Below is an example. We explain details in class.


Generalized Cross Validation
GCV(λ)
λ
2.165
2.16
2.155
2.15
2.145
2.14
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2



Matlab Script for the Numerical Example

```
%%% 1. create the noisy data points
x = -4:4; np1 = 201; y = [0 .15 1.12 2.36 2.36 1.46 .49 .06 0];
cs = spline(x,y);  xx = linspace(-4,4,np1);
yy = ppval(cs,xx) + randn(size(xx)).*0.1;

%%% 2. Compute for GCV(lambda)
W = speye(np1); h = 8/(np1-1);
bQ = zeros(np1-2,np1);
for i=1:np1-2, bQ(i,i+[0 1 2]) = [1 -2 1]./h; end;
bM = 2/3*h.*diag(ones(np1-2,1));
for i=1:np1-3, bM(i,i+1) = h/6; bM(i+1,i) = h/6; end;
lambda_lt = linspace(0.01, .2, 100);
for ii = 1:length(lambda_lt),
    lambda = lambda_lt(ii);
    Slambda = inv(W + lambda.*bQ'*inv(bM)*bQ)*W;
    GCV(ii) = norm(yy' - Slambda*(yy'))^2./(1-trace(Slambda)./np1)^2;
end;
```

```
clf; plot(lambda_lt, GCV)
title('Generalized Cross Validation','fontsize',16)
xlabel('\lambda','fontsize',16); ylabel('GCV(\lambda)','fontsize',16)
print -depsc2 gcv01

%%% 3. Identify lambda that minimizes GCV, then use it to determines a spline.
[temp, ilambda]=min(GCV); lambda_0 = lambda_lt(ilambda);
yyhat = inv(W + lambda_0.*bQ'*inv(bM)*bQ)*W*(yy');
figure(2); plot(xx,yy,'.'); hold on; plot(xx,yyhat,'r-','linewidth',2);
title('Noisy Observations','fontsize',16)
xlabel('x','fontsize',16); ylabel('y','fontsize',16)
legend('Noisy Observations','Fitted Spline')
print -depsc2 gcv02
```