Statistics 540, Smoothing

Scatterplot Smoothing

Overview

Problem ...

The usual setting for scatterplot smoothing is the idealized regression model

$$y_i = f(x_i) + \sigma\epsilon_i, \quad \epsilon_i \sim N(0,1), \quad i = 1, \ldots, n,$$

where the observations (x_i, y_i) are independent. Independence is crucial, whereas the assumption of normality is more of convenience than necessity as in least squares regression. The goal is to estimate the underlying expectation function f from the n observations (x_i, y_i). For the moment, we'll assume that x_i is a scalar.

Assumptions ...

We need to match the estimator $\hat{f}$ to the properties that we assume hold for f. For example, if we assume that f is periodic with period d, then we ought to have our smoother $\hat{f}$ share this property. A more reasonable assumption is that f is a continuous or perhaps differentiable function.

From the data alone, we *cannot* determine an upper bound on the roughness of f (see Donoho 1988), though we can obtain lower bounds. In the related context of density estimation, for example, we can find a one-sided interval for the number of modes (indicating that we need at least a certain number), but not a two-sided interval. How could the data indicate that the true density was not multimodal, with a mode at each observation?

Issues ...

One needs to keep a variety of issues in mind when smoothing, as trade-offs need to be made. Key attributes of all smoothers are

1. Smoothness properties of estimator, supporting rationale.
2. Local sensitivity to data.
3. Bias/variance trade-off.
4. Estimators are blend linear and nonlinear functions.

A common criterion that makes some of these issues more concrete is to define the estimator implicitly, as the solution of

$$\hat{f} = \arg\min_f \sum_i (y_i - f(x_i))^2 + \lambda \int_a^b f''(t)^2 dt \; . \tag{1}$$

Some further issues that are often forgotten until too late are

1. Behaviour at endpoints.
2. Effect of missing data and the assumption of equal spacing.
3. Robustness to outlying values.
4. Computational speed versus generality.

Approaches ...

The most common estimation methods are

Sliding regressions link a series of linear/polynomial fits computed from overlapping subsets of some width. The lowess smoother in LispStat is the best example of this group.

Kernel methods weight the data by a moving smoothing kernel K of some width (typical kernels resemble the Gaussian density). The estimator of $f(x)$ has the form of weighted average of the data,

$$\hat{f}(x_i) = \frac{\sum_i y_i K(\frac{x_i - x}{w})}{\sum_i K(\frac{x_i - x}{w})} \tag{2}$$

The choice of kernel function K is much less important in applications than the choice of the smoothing width w.

Smoothing splines join continuous low-order polynomials that satisfy some external smoothness assumption. These are the main topic for today's class.

Wavelets and thresholding which together comprise a localized orthogonal decomposition of the data with selected coefficients shrunken toward zero. We will study these separately later as time permits.

Each of these can be made more 'robust' (ie, tolerant of outliers) by adapting the estimation method appropriately. For example, lowess uses a robust regression rather than a least squares regression and one can replace the weighted average (2) of the kernel smoother by a robust estimate of location.

Cubic Splines

Knots ...

Let the points $a = x_1 < \cdots < x_n = b$ define a partition of the interval $[a, b]$, and assume that we have observations $(x_i, y_i),\ i = 1, \ldots, n$. The points x_i are known as the *knots*.

Splines ...

A spline is a piecewise polynomial function. The simplest spline is a piecewise constant function,

$$s_0(x) = y_j\ , \quad x_j \le x < x_{j+1}\ .$$

This spline has no continuity at the knot locations. The linear spline

$$s_1(x) = y_j + (y_{j+1} - y_j)\frac{x - x_j}{x_{j+1} - x_j} , \quad x_j \le x < x_{j+1} .$$

is continuous, but its first derivative (a zero order spline) is a step function. Note that we define s_0 from one knot, s_1 from two (an interval). The quadratic spline s_2 requires two intervals (3 knots) and the cubic spline $s(x) = s_3(x)$ requires three intervals (4 knots). Cubic splines occupy a special place in the theory of smoothers, and we'll focus on these.

Definition ...

The function $s(x)$ is a cubic spline on $[a, b]$ if it

1. Interpolates: $s(x_i) = y_i$, is
2. Smooth: $s(x), s'(x), s''(x)$ are continuous, and is a
3. Cubic polynomial on each interval $[x_i, x_{i+1}]$.

Extremal property ...

Cubic splines have an important extremal property. Among all interpolating, differentiable functions, the so-called *natural* cubic splines minimize the squared integrated second derivative:

$$\int_a^b s''(t)^2 dt \le \int_a^b g''(t)^2 dt, \text{ with } \quad s''(a) = s''(b) = 0, \tag{3}$$

Proof Begin by expanding the square, with the terms rearranged in a useful manner:

$$0 \le \int_a^b (s''(x) - g''(x))^2 dx = \int_a^b g''(x)^2 - s''(x)^2 - 2s''(x)(g''(x) - s''(x))dx$$

We are done if we can show that the last term is zero. Start by formulating it as an integration by parts, then use the fact that $s'''(x)$ is piecewise constant on the partition:

$$\begin{aligned}
\int_a^b s''(x)(g''(x) - s''(x))dx &= \int_a^b s''(x)d(g'(x) - s'(x)) \\
&= s''(x)(g'(x) - s'(x))|_a^b - \int_a^b s'''(x)(g'(x) - s'(x))dx \\
&= s''(b)(g'(b) - s'(b)) - s''(a)(g'(a) - s'(a)) \sum_i \int_{x_i}^{x_{i+1}} s'''(x)(g'(x) - s'(x) \\
&= s''(b)(g'(b) - s'(b)) - s''(a)(g'(a) - s'(a)) \sum_i k_i(g(x) - s(x))|_{x_i}^{x_{i+1}} \\
&= 0
\end{aligned}$$

if we also assume the standard condition that $s''(a) = s''(b) = 0$. With these additional boundary conditions (that produce a linear extrapolation), $s(x)$ is known as a *natural cubic spline.*

Unique calculation solution ...

Given n pairs (x_i, y_i) with distinct x's, there is but one cubic interpolating spline. The associated $n-1$ cubic polynomials have $4(n-1)$ coefficients that we must be able to determine uniquely. The interpolation condition implies $2(n-1)$ linear constraints on the coefficients. The smoothness of each derivative implies a further $n-2$ constraints (one for each interior knot. Combining all of these leads to a 'huge' linear system of equations with the $4(n-1)$ unknown coefficients and $4(n-1)-2$ linear equations. Adding the two additional 'natural' boundary conditions gives a complete system (which is nearly diagonal).

Regression splines ...

So far, nothing has been said about smoothing with splines. As defined, splines simply smoothly interpolate data with smoothly joined piecewise polynomials. Splines were designed as low-order *interpolating* polynomials, not as smoothers. They were needed to avoid the end-value problems that one runs into with high-order interpolating polynomials. There are two broad ways that one can smooth with piecewise polynomials. The first, loosely called regression splines, is quite simple and underlies Friedman's MARS method and the Turbo-Smoother of Friedman and Silverman.

Given observations $(x_1, y_1), \ldots, (x_n, y_n)$, use only a few observations as knots and compute a polynomial on each interval by least squares. Suppose $n = 100$ and we use $x_{50} = 0$ as the single knot. Fit a cubic on each interval. The initial polynomial is, say,

$$s(x) = A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \quad \text{for} \quad x \le 0.$$

Let $B(x) = \sum_{j=0}^{3} b_j x^j$ denote the polynomial for positive x.

When fitting from data, one does not simply fit two separate cubics via least squares — these two would not satisfy the smoothness conditions. In fact, adding the second polynomial adds one degree of freedom to the fit. Fix a_j and see what you can tell about $B(x)$. The interpolating condition $A(0) = B(0)$ implies $b_0 = a_0$. Continuity of $s'(x)$ and $s''(x)$ at zero implies that $b_1 = a_1$ and $b_2 = a_2$. All that's left is to find the new cubic term b_3. Non-zero knots make the algebra more complex, but each new cubic adds but one degree of freedom (a new coefficient) to the fit (four coefficients minus 3 linear constraints equals one new coefficient).

Smoothing splines ...

Smoothing splines traditionally mean something different. Here's the idea. Pick any smoother that you like, and let $\hat{y}_i$ denote its fitted values at the given x_i. No matter what smoother has been used to determine the $\hat{y}_i$, the extremal property (3) implies you can do better in terms of the criterion (1) by interpolating these $\hat{y}_i$ with a cubic spline. The trick to calculations, however, is to fit the minimizer of (1) rather than 'improving' another estimator.

Computing Smoothing Splines

Methods of calculation ...

You have several ways to approach the computation of smoothing splines. We'll consider two. The first is to return to the regression splines, and think of fitting these with a knot at every point. The second is more direct and relies upon a set of basis functions known as *B-splines* for the space of functions spanned by the cubic splines.

Via regression splines ...

To find a cubic interpolating spline via regression, consider the representation

$$s(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{j=2}^{n-1} \theta_j (x - x_j)_+^3 \tag{4}$$

where x_+ is the positive part of x. Superficially, this expression has $n+2$ unknowns, but the boundary conditions add two more constraints that nail things down. You can check that the resulting function satisfies our conditions for a cubic spline. (Numerically, one avoids this representation since it introduces very large values (the cubics) and leads to near singular design matrices even with fewer than $n-1$ intervals. It does help one see what is happening, however.)

The hard part of this representation is to decern how to incorporate the penalty term $\int f''(x)^2 dx$ into the estimation of the coefficients of the regression spline (4). If we substitute $s(x)$ from (4) into this integral for f, we obtain for the ith interval the sum

$$\int_{x_i}^{x_{i+1}} s''(x)^2 dx = \int_{x_i}^{x_{i+1}} (2\beta_2 + 6\beta_3 x + 6 \sum_{j \le i} \theta_j (x - x_j))^2 dx$$

The accumulating nature of the basis functions (they span all intervals to the right) make this expression pretty unwieldy. The remedy is to use a different, equivalent set of regressors.

B-splines ...

An alternative to the truncated power basis used implicitly in (4) is to use polynomials which are zero outside of a small range known as B-splines. For example, in the linear case, the regression spline formulation is (new θ's)

$$s_1(x) = \beta_0 + \beta_1 x + \sum_{j=2}^{n-1} \theta_j (x - x_j)_+ \ .$$

The associated regression design matrix has a triangular shape. Alternatively, we can parameterize s_1 using triangular functions that span just two intervals,

$$B_{j,2}(x) = \begin{cases} \frac{x - x_j}{x_{j+1} - x_j}, & x_j \le x < x_{j+1} \\ 1 - \frac{x - x_{j+1}}{x_{j+2} - x_{j+1}}, & x_{j+1} \le x < x_{j+1} \end{cases}$$

Then write the linear spline as

$$s_1(x) = \sum_j \gamma_j B_{j,2}(x) \ ,$$

so that the regressors are more nearly orthogonal. Consequently, the calculations are more stable, and since B-splines are nearly orthogonal making "$X'X$" is almost diagonal, the calculations are also quite fast. Cubic B-splines behave similarly and are computed in essence by integrating up from the linear B-splines. It is important to note that the B-splines are polynomials defined by the grid of x_i's and do *not* depend on the y_i.

B-splines are in general defined by the recurrence expression

$$B_{j,k}(x) = \frac{x - x_j}{x_{j+k-1} - x_j} B_{j,k-1}(x) + \frac{x_{j+k} - x}{x_{j+k} - x_j} B_{j+1,k-1}(x) \ ,$$

and one gets considerable simplifications when the grid of x's is equally spaced. The recursion is started with the indicator functions

$$B_{j,1} = 1 \quad \text{for } x_j \le x < x_{j+1}$$

and zero elsewhere (see deBoor 1978, eqns 4,5 of Chapter 10).

Smoothing, at last ...

Write the cubic spline in vector form as a linear combination of cubic B-splines,

$$s(x) = \sum_j \gamma_j B_j(x) \quad \Rightarrow (f(x_1), \ldots, f(x_n))' = B\gamma$$

where B is the matrix with element $B_{ij} = B_j(x_i)$. Then substitute this expression into the smoothing expression (1) to obtain

$$(Y - B\gamma)'(Y - B\gamma) + \lambda \gamma' M \gamma \tag{5}$$

where the matrix M has elements

$$M_{ij} = \int_a^b B_i''(x) B_j''(x) dx \ .$$

The expression (6) is now in the form more suited to be recognized as a penalized least squares estimator, with solution

$$(B'B + \lambda M)\hat{\gamma} = B'Y \ . \tag{6}$$

This type of estimator has a long history, including Marquart's method for nonlinear optimization (keeping the Hessian positive definite) and ridge regression (Hoerl and Kennard, where it provides a biased estimator to overcome collinearity).

Bayesian interpretation ...

There is also a Bayesian argument that leads to a penalized least squares solution. Assume that the data are conditionally normal following the usual regression model,

$$Y = X\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I_n).$$

However, add a prior distribution on the slopes, making them normal as well and centered on zero with variance v^2,

$$\beta \sim N(0, v^2 I_k) \ .$$

Thus, $Y|\beta \sim N(X\beta, \sigma^2 I_n)$ and $\text{Cov}(Y, \beta) = X \,\text{Var}(\beta) = v^2 X$, and the joint distribution of Y and β is

$$\begin{pmatrix} Y \\ \beta \end{pmatrix} = N\left(0, \begin{pmatrix} v^2 XX' + \sigma^2 I_n & v^2 X \\ v^2 X' & v^2 I_k \end{pmatrix}\right)$$

The posterior mean for β is then (using the usual regression expressions and letting $r = \sigma^2/v^2$)

$$\begin{aligned} E\ \beta|Y &= v^2 X'(v^2 XX' + \sigma^2 I_n)^{-1} Y \\ &= (1/r)(I_k - X'X(X'X + rI_k)^{-1})X'Y \\ &= (X'X + rI_k)^{-1} X'Y \ . \end{aligned}$$

The first step comes as a special case of the formula for the inverse of a partitioned matrix. In particular, for a square matrix partitioned as M_{ij} $(i, j = 1, 2)$, we have

$$(M_{11} - M_{12}M_{22}^{-1}M_{21})^{-1} = M_{11}^{-1} + M_{11}^{-1}M_{12}(M_{22} - M_{21}M_{11}^{-1}M_{12})^{-1}M_{21}M_{11}^{-1} \ . \qquad (7)$$

You can derive this expression by diagonalizing the matrix M in blocks using the regression expressions. You can reproduce the last step by rearranging the corresponding scalar expression as

$$1 - \frac{x^2}{x^2 + r} = \frac{r}{x^2 + r} = (x^2/r + 1)^{-1} \ .$$

Choosing the Smoothing Parameter

Picking λ ...

So how does one choose λ in the criterion (1)? A popular choice is based on cross-validation, a mechanism for assessing the out-of-sample performance of a statistical estimator.

Judging a regression model ...

So how ought one pick the variables in a regression model? One approach is to try to

pick the model that you believe will predict best when applied to a *new* set of data (from the same population as the one used to construct the model). Suppose that we observe n observations whose mean is some vector η which is unknown to us, though fixed:

$$Y = \eta + \sigma\epsilon, \quad \epsilon \sim N(0, I_n),$$

For model building, assume that we are going to approximate η by projecting it into a subspace associated with a collection of k predictors, collected into the $n \times k$ matrix X. Let $\hat{\eta} = X\hat{\beta}$ be the least squares estimate of η based on this projection of η into the span of X, $X\beta = H\eta$ for $H = X(X'X)^{-1}X'$. For convenience of notation, let $\|y\|^2 = E\ y'y$.

The expected prediction error sum of squares for predicting an independent vector $Y^* = \eta + \sigma\epsilon^*$ with the same mean η using the fit to the original n observations is

$$\begin{aligned} \|Y^* - \hat{\eta}\|^2 &= \|\eta - H\eta\|^2 + \|\sigma\epsilon^* + H\eta - \hat{\eta}\|^2 \\ &= \|\eta - H\eta\|^2 + \|\sigma\epsilon^*\|^2 + \|H\eta - HY\|^2 \\ &= \|\eta - H\eta\|^2 + n\sigma^2 + \|H\epsilon\|^2 \\ &= \underbrace{\|\eta - H\eta\|^2}_{\text{bias}^2} + \underbrace{(n+k)\sigma^2}_{\text{variance}} . \end{aligned} \tag{8}$$

The bias shrinks as we increase the size of the subspace for projection (ie, add more variables to the model, increasing k), whereas the variance term gets *larger* as the number of predictors k increases. We see the classic trade-off of bias versus variance. Notice that if you consider the MSE of $\hat{\eta}$, you will find

$$MSE(\hat{\eta}) = \|\eta - \hat{\eta}\|^2 = \|\eta - H\eta\|^2 + k\sigma^2 ,$$

dropping the term $n\sigma^2$ *which does not depend on the fitted model* and thus does not affect which model we would choose. That is, the model that minimizes $\|Y^* - \hat{\eta}\|^2$ also minimizes the MSE.

Now, if we hope to find the model that minimizes this sort of out-of-sample prediction error, we need some way of computing (8) from the available data. For example, the expected residual sum of squares (expected *in-sample* prediction error) is

$$\begin{aligned} \|Y - \hat{\eta}\|^2 &= \|\eta - H\eta\|^2 + \|\sigma\epsilon + H\eta - \hat{\eta}\|^2 \\ &= \|\eta - H\eta\|^2 + \|\sigma\epsilon + H\eta - HY\|^2 \\ &= \|\eta - H\eta\|^2 + \sigma^2\|\epsilon - H\epsilon\|^2 \\ &= \|\eta - H\eta\|^2 + (n-k)\sigma^2 . \end{aligned} \tag{9}$$

That is, the residual sum of squares is the sum of bias plus another term which now *also shrinks* as k increases. This is not the needed behaviour and leads to the problem of *overfitting* (using too many predictors). Among the patches for this problem are Mallow's (1973) C_p statistic as well as ...

Cross-validation ...

In the spirit of out-of-sample prediction, cross-validation seeks the regression model (really, the set of predictors that make up X) which minimizes

$$\sum_i (y_i - x_i' \hat{\beta}_{(-i)})^2 \tag{10}$$

where $\hat{\beta}_{(-i)}$ denotes the slope estimates based on all of the data *except* for the ith observation. Rather than setting apart some fraction of the data for validation, each observation is left out, in this case one at a time (better alternatives leave out 2 or more), and predicted from a model fit to the rest of the data. For smoothing, the corresponding expression is

$$\sum_i (y_i - \hat{f}_{(-i)})^2 \ , \tag{11}$$

with $\hat{f}_{(-i)}$ denoting the smooth fit without (x_i, y_i).

Expressions for regression calculations ...

For regression, there is a very useful special case of the partitioned inverse expression (7):

$$(M - ab')^{-1} = M^{-1} + \frac{M^{-1}ab'M^{-1}}{1 + a'M^{-1}b} \tag{12}$$

where M is a square matrix and a and b are conformable vectors. (Proof idea: Look at the geometric expansion $1/(1-x) = 1 + x + x^2 + \cdots$. Write $(M - ab)^{-1} = M^{-1}(I - ab'M^{-1})^{-1}$ and try to expand similarly.) In regression, the value of (12) is to notice that

$$(X'_{(-i)}X_{(-i)})^{-1} = (X'X - x_i x_i')^{-1} = (X'X)^{-1} + \frac{(X'X)^{-1}x_i x_i'(X'X)^{-1}}{1 - h_i} \ ,$$

where $h_i = x_i'(X'X)^{-1}x_i$ is the so-called *leverage* for the ith observation (the diagonal of the projection matrix $H = X(X'X)^{-1}X'$). This expression then gives

$$\begin{aligned} \hat{\beta}_{(-i)} &= (X'_{(-i)}X_{(-i)})^{-1}X'_{(-i)}Y_{(-i)} \\ &= \left((X'X)^{-1} + \frac{(X'X)^{-1}x_i x_i'(X'X)^{-1}}{1 - h_i} \right) (X'Y - x_i y_i) \\ &= \hat{\beta} - (X'X)x_i \frac{e_i}{1 - h_i} \ , \end{aligned}$$

where e_i is the usual residual $e_i = y_i - x_i'\hat{\beta}$. Thus its quite easy to compute the summands in (10):

$$\begin{aligned} y_i - x_i\hat{\beta}_{(-i)} &= y_i - x_i'(\hat{\beta} - (X'X)^{-1}x_i \frac{e_i}{1 - h_i}) \\ &= e_i + \frac{h_i e_i}{1 - h_i} \end{aligned}$$

$$= \frac{e_i}{1-h_i} ,$$

so that (10) becomes

$$\sum_i (y_i - x_i'\hat{\beta}_{(-i)})^2 = \sum_i \frac{e_i^2}{(1-h_i)^2} . \tag{13}$$

Thus, the cross-validation sum of squares (*CVSS*) is simply a weighted sum squared residuals, and very easy to compute.

Generalized cross-validation (*GCV*) ...

Generalized cross validation goes one step further. Notice first that

$$\sum_i h_i = \mathrm{tr}H = \mathrm{tr}X(X'X)^{-1}X' = k ,$$

the number of regressors (including the constant). Rather than compute *CVSS* in (13) directly, replace h_i in the denominator with its average, $\overline{h} = k/n$, obtaining the approximation

$$\begin{aligned} \sum_i (y_i - x_i'\hat{\beta}_{(-i)})^2 &\approx \sum_i \frac{e_i^2}{(1-\overline{h})^2} \\ &= \sum_i \frac{e_i^2}{(1-k/n)^2} \\ &= \left(\frac{n}{n-k}\right)^2 \sum_i e_i^2 \end{aligned}$$

which has from (9) expected value

$$\mathrm{bias}^2 + \frac{n^2}{n-k}\sigma^2 \approx \mathrm{bias}^2 + (n+k)\sigma^2 ,$$

as motivated by out-of-sample prediction. Another way to look at this last expression is to notice that the MSE of $\hat{\eta}$ is $\|\eta - H\eta\|^2 + k\sigma^2$, so that minimizing the CVSS is attempting to minimize the MSE of the fitted model.

Applications in smoothing ...

In general, things are not so simple when dealing with smoothing, but one gets a long way by treating most smoothers in a manner resembling regression. One can always resort to "brute force" to obtain $\hat{f}_{(-i)}$, but better methods are easily obtained and generalized cross validation simplifies things further.

Smoother matrix ...

The key step in leveraging the regression calculations is to express a smoother in linear form resembling the least squares projection $\hat{Y} = HY$. For example, the moving average smoother of length w can be written as

$$\hat{f}_{ma} = \frac{1}{w} W Y$$

where W is the $n \times n$ matrix whose i row contains the weights (mostly 1's) used to compute the ith smoothed value. Similarly, using the B-spine representation for the fitted model (6) we can write

$$\hat{f} = B\hat{\gamma} = B(B'B + \lambda M)^{-1}B'Y = S_\lambda Y \ . \tag{14}$$

We are not likely compute $\hat{f}$ this way, but it makes the problem more easy to think about. Note that S_λ depends *only* upon the x_i's and choice of λ. By comparison to the true projection matrix H from regression, S_λ for *smoothing splines*

1. Is *not* a projection matrix since $S_\lambda \neq S_\lambda^2$,
2. Is symmetric (look at (14)), and
3. Has trace which is often used as the degrees of freedom for the smoother. (Others, such as tr S_λ^2 are also possible since S_λ is not a projection matrix.)

MSE of smoother ...

Writing $\hat{f} = S_\lambda Y$, we get a nice expression for the MSE of the smoother,

$$\begin{aligned} \|f - S_\lambda Y\|^2 &= \|f - S_\lambda f - \sigma S_\lambda \epsilon\| \\ &= \|f - S_\lambda f\|^2 + \sigma^2 \|S_\lambda \epsilon\|^2 \\ &= f'(I - S_\lambda)^2 f + \sigma^2 \mathrm{tr} S_\lambda^2 \ . \end{aligned}$$

Compare this to the previous regression expressions and you'll see that the difference is really just the second term, which for regression would be $k\sigma^2$.

Cross-validation with smoothers ...

Note first that for smoothing splines that S_λ has two eigenvectors with eigenvalue 1,

$$S_\lambda \mathbf{1} = \mathbf{1}, \qquad S_\lambda x = x$$

where x denotes a linear vector with $x_j = jx_1$. A method that leads to simple expressions for CVSS is to define (suppressing λ from S_λ)

$$\hat{f}_{(-i)} = \frac{\sum_{j \neq i} S_{ij} y_j}{1 - S_{ii}} \ , \tag{15}$$

that is, set the weight on y_i to zero and renormalize the others so that they sum to one (How do you know they ought to sum to one?). From (15) we obtain

$$\begin{aligned} (1 - S_{ii})\hat{f}_{(-i)} &= \hat{f}_i - S_{ii} y_i \\ &= \hat{f}_i - S_{ii}(\hat{f}_i + e_i) \\ &= (1 - S_{ii})\hat{f}_i - S_{ii} e_i \end{aligned}$$

so that we can express the "leave-one-out" fit as

$$\hat{f}_{(-i)} = \hat{f}_i - \frac{S_{ii}}{1 - S_{ii}} e_i \ . \tag{16}$$

Thus the CVSS for smoothing becomes

$$\begin{aligned}\sum_i (y_i - \hat{f}_{(-i)})^2 &= \sum_i (y_i - \hat{f}_i + \frac{S_{ii}}{1-S_{ii}} e_i)^2 \\ &= \sum_i (e_i + \frac{S_{ii}}{1-S_{ii}} e_i)^2 \\ &= \sum_i \frac{e_i^2}{(1-S_{ii})^2} \ ,\end{aligned}$$

which corresponds to the expression (13) for regression. One then chooses the value of λ that minimizes this expression. Alternatively, using GCV, one can replace S_{ii} by the average $\mathrm{tr}S_\lambda/n$ as was done with regression.

Further reading... The book *Generalized Additive Models* by Hastie and Tibshirani discusses all this and more.