Robust Smoothing: Smoothing Parameter Selection and Applications to Fluorescence Spectroscopy

Jong Soo Lee
Carnegie Mellon University, Department of Statistics
Pittsburgh, PA 15213
jslee@stat.cmu.edu
and
Dennis D. Cox
Rice University, Department of Statistics
6100 Main St. MS-138, Houston, TX 77005
dcox@stat.rice.edu

Fluorescence spectroscopy has emerged in recent years as an effective way to detect cervical cancer. Investigation of the data preprocessing stage uncovered a need for a robust smoothing to extract the signal from the noise. We compare various robust smoothing methods for estimating fluorescence emission spectra and data driven methods for the selection of smoothing parameter. The methods currently implemented in R for smoothing parameter selection proved to be unsatisfactory and we present a computationally efficient procedure that approximates robust leave-one-out cross validation.

Keywords: Robust smoothing; Smoothing parameter selection; Robust cross validation; Leave out schemes; Fluorescence spectroscopy

1 Introduction

In recent years, fluorescence spectroscopy has shown promise for early detection of cancer (Grossman *et al.*, 2002; Chang *et al.*, 2002). Such fluorescence measurements are obtained by illuminating the tissue at one or more excitation wavelengths and measuring the corresponding intensity at a number of emission wavelengths. Thus, for a single measurement we obtain discrete noisy data from several spectroscopic curves which are processed to produce estimated emission spectra. One important step in the processing is smoothing and registering the emission spectra to a common set of emission wavelengths.

However, the raw data typically contain gross outliers, so it is necessary to use a robust smoothing method. For real time applications we need a fast and fully automatic algorithm.

For this application, we consider several existing robust smoothing methods which are available in statistical packages, and each of these smoothers also has a default method for smoothing parameter selection. We found that the default methods of do not work well. To illustrate this point, we present Figure 1 where some robust fits, using the default smoothing parameter selection methods are shown. We have also included the plots of the "best" fit (as developed in this paper) for comparison purposes.

The best methods we found for smoothing parameter selection required some form of robust cross validation - leaving out a subset of the data, smoothing, then predicting the left out data, and evaluating the prediction error with a robust loss function. Full leave-one-out robust cross validation performed well for smoothing parameter selection but is computationally very time consuming. We present a new method, *systematic* K-fold robust cross validation, which reduces the computation time and still gives satisfactory results.

There has been previous work on the robust smoothing of spectroscopy data. In particular, Bussian and Härdle (1984) consider the robust kernel method for the estimation from Raman spectroscopy data with huge outliers. However, whereas they only consider one specific robust smoothing method, we make

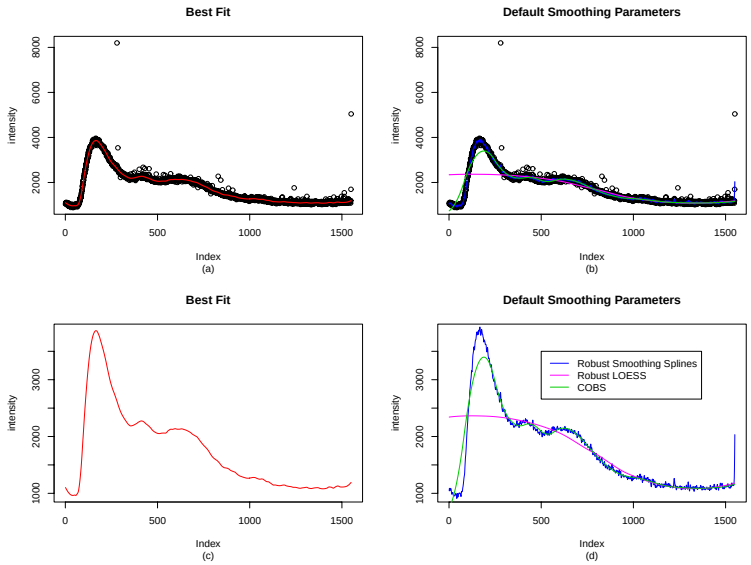

Figure 1: Parts (a) and (b) show raw data with the smoothed estimates superimposed. Part (c) is the fit corresponding to the best smoothing parameter value, and part (d) show the fit based on default methods for various robust smoothers. The default method from robust smoothing splines undersmooths, whereas the default methods of robust LOESS and COBS oversmooths.

some comparisons of available methods and make some recommendations. Furthermore, Bussian and Härdle (1984) do not treat the problem of automatic bandwidth selection.

The paper is organized as follows. We first review some robust smoothers in Section 2. Then in Section 3, we introduce the robust procedures for robust smoothing parameter selection along with some default methods from the functions in R. We also introduce a method which approximates the usual leave-one-out cross validation scheme to speed up the computation. In Section 4, we perform a preliminary study, determine which smoothers and smoothing parameter methods to include in a large scale simulation study, and then give results on this, along with the results on how the method works with real data. Finally, we conclude the paper with some recommendations and directions of possible

future research in Section 5.

2 Robust Smoothing Methodologies

In this section, we define the robust smoothing methods that will be considered in our study. Consider the observation model

$$y_i = m(x_i) + \epsilon_i, \quad i = 1, \ldots, n$$

where the ϵ_i are i.i.d. random errors, and we wish to obtain an estimate $\hat{m}$ of m. In order for this to be well defined, we need to assume something about the "center" of the error distribution. The traditional assumption that the errors have zero mean is not useful here as we will consider possibly heavy tailed distributions for the errors. In fact, each different robust smoother in effect defines a "center" for the error distribution, and the corresponding regression functions $m(x)$ could differ by a constant. If the error distribution is symmetric about 0, then all of the methods we consider will be estimating the conditional median of Y given $X = x$.

2.1 Robust Smoothing Splines

The robust smoothing spline is defined through an optimization problem: find $\hat{m}$ that minimizes

$$\frac{1}{n}\sum_{i=1}^{n} \rho(y_i - \hat{m}(x_i)) + \lambda \int (\hat{m}^{''})^2$$

for an appropriately chosen $\rho(\cdot)$ function and subject to $\hat{m}$ and $\hat{m}'$ absolutely continuous with $\int (\hat{m}'')^2 < \infty$. Of course, the choice of ρ is up to the investigator. We generally require that ρ be even symmetric, convex, and grows slower than $O(x^2)$ as $|x|$ gets large. Here, $\lambda > 0$ is the smoothing parameter, with larger values of λ corresponding to smoother estimates.

The implementation of the robust smoothing splines we have used is the R function `qsreg` from the package `fields`, version 3.04 (Oh, *et al.*, 2004), which

uses

$$\rho(x) = \begin{cases} \frac{x^2}{2C} & \text{if } |x| \leq C \\ |x| - \frac{C}{2} & \text{if } |x| > C \end{cases} \tag{1}$$

It is easily verified that the $\rho(x)$ function in (1) satisfies the requirements above. The break point C in (1) is a scale factor usually determined from the data. See Oh *et al.* (2004) for details. For the `qsreg` function, the default is $C = 10^{-5}\sqrt{var(x)}$.

2.2 Robust LOESS

We used the R function `loess` from the package `stats` (version 2.4.1) to perform robust locally weighted polynomial regression (Cleveland, 1979; Chambers and Hastie, 1992). At each x, we find $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ to minimize

$$\sum_{i=1}^{n} \rho(y_i - \beta_0 - \beta_1(x - x_i) - \cdots - \beta_p(x - x_i)^p) K\left(\frac{x - x_i}{h}\right),$$

where the resulting estimate is $\hat{m}(x) = \sum_{j=0}^{p} \hat{\beta}_j x^j$. We consider only the degree $p = 2$, which is the maximum allowed. With the `family = "symmetric"` option, the robust estimate is computed via the Iteratively Reweighted Least Squares algorithm using Tukey's biweight function for the reweighting. $K(\cdot)$ is a compactly supported kernel (local weight) function that downweights x_i that are far away from x. The kernel function used is the "tri-cubic"

$$K(x) = \begin{cases} (1 - |x|^3)^3 & \text{if } |x| < 1, \\ 0 & \text{if } |x| \geq 1. \end{cases}$$

The parameter h in the kernel is the bandwidth, which is specified by the fraction of the data λ ($0 < \lambda \leq 1$) within the support of $K(\cdot/h)$, with larger values of λ giving smoother estimates.

2.3 COBS

Constrained B-spline smoothing (COBS) is implemented in R as the function `cobs` in the `cobs` package, version 1.1-3.5 (He and Ng, 1999). The estimator

has the form

$$\hat{m}(x) = \sum_{j=1}^{N+q} \hat{a}_j B_j(x)$$

where N is the number of internal knots, q is the order (polynomial degree plus 1), $B_j(x)$ are B-spline basis functions (de Boor, 1978), and $\hat{a}_j$ are the estimated coefficients. There are two versions of COBS depending on the roughness penalty: an L_1 version and an L_∞ version. For the L_1 version the coefficients $\hat{a}_1, \ldots, \hat{a}_{N+q}$ are chosen to minimize

$$\sum_{i=1}^{n} |y_i - \hat{m}(x_i)| + \lambda \sum_{i=1}^{N} |\hat{m}'(t_{i+q}) - \hat{m}'(t_{i+q-1})|$$

where $t_1, \ldots, t_{N+2q}$ are the knots for the B-splines. The L_∞ version is obtained by minimizing

$$\sum_{i=1}^{n} |y_i - \hat{m}(x_i)| + \lambda \max_x |\hat{m}''(x)| .$$

The $\lambda > 0$ is a smoothing parameter similarly to the λ in robust smoothing splines. Here not only λ has to be determined as with other robust smoothers, but we also need to determine the number of internal knots, N, which acts as a smoothing parameter as well. In the `cobs` program, the `degree=` option determines the penalty: `degree=1` gives an L_1 constrained B-spline fit, and `degree=2` gives an L_∞ fit.

3 Smoothing Parameter Selection

In all robust smoothers, a critical problem is the selection of the smoothing parameter λ. One way we can do this is through subjective judgment or inspection. But for the application to automated real time diagnostic devices, this is not practical. Thus, we need to develop an accurate, rapid method to determine the smoothing parameter automatically.

For many nonparametric function estimation problems, the method of leave-one-out cross validation (CV) is often used for smoothing parameter selection

(Simonoff, 1996). The Least Squares CV function is defined as

$$LSCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{m}_\lambda^{(i)}(x_i)\right)^2$$

where $\hat{m}_\lambda^{(i)}$ is the estimator with smoothing parameter λ and the ith observation (x_i, y_i) deleted. One would choose λ by minimizing $LSCV(\lambda)$.

However, this method does not work well for the robust smoothers because the $LSCV$ function itself will be strongly influenced by outliers (Wang and Scott, 1994).

3.1 Default Methods

All of the packaged programs described above have a default method for smoothing parameter selection which is alleged to be robust.

The robust smoothing splines from `qsreg` function provides a default smoothing parameter selection method using pseudo-data, based on results in Cox (1983). The implementation is done via generalized cross validation (GCV) with empirical pseudo-data, and the reader is referred to Oh *et al.* (2004, 2007) for details.

In COBS, the default method for selecting λ is a robust version of the Bayes information criterion (BIC) due to Schwarz (1978). This is defined by He and Ng (1999) as

$$BIC(\lambda) = \log\left(\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{m}_\lambda(x_i)|\right) + \frac{1}{2}p_\lambda \log(n)/n$$

where p_λ is the number of interpolated data points.

The robust LOESS default method for smoothing parameter selection is to use the fixed value $\lambda = 3/4$. This is clearly arbitrary and one cannot expect it to work well in all cases.

Moreover, the pseudo-data method is known to produce an undersmoothed curve, and BIC method usually gives an oversmoothed curve. These claims can be verified by inspecting the plots in the corresponding papers referenced above (Oh *et al.* 2004, 2007; He and Ng 1999) and can also be seen in Figure 1 of

present paper. Such results led to an investigation of improved robust smoothing parameter selection methods.

3.2 Robust Cross Validation

To solve the problem of smoothing parameter selection in the presence of gross outliers, we propose a Robust CV (RCV) function

$$RCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\rho(y_i - \hat{m}_\lambda^{(i)}(x_i)) \tag{2}$$

where $\rho(\cdot)$ again is an appropriately chosen criterion function. Similarly to $LSCV$, the $\hat{m}_\lambda^{(i)}$ is the robust estimate with the ith data point left out. We consider various ρ functions in RCV. For each of the methods considered, there is an interpolation (`predict`) function to compute $\hat{m}_\lambda^{(i)}(x_i)$ at the left out x_i's.

We first considered the absolute cross validation (ACV) method proposed by Wang and Scott (1994). With this method, we find a λ value that minimizes

$$ACV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{m}_\lambda^{(i)}(x_i)|.$$

So this is a version of RCV with $\rho(\cdot) = |\cdot|$. Intuitively, the absolute value criteria is resistant to outliers, because the absolute error is much smaller than the squared error for large values. Wang and Scott (1994) found that the ACV criteria worked well for local L_1 smoothing.

Because the ACV function (as a function of λ) can be wiggly and can have multiple minima (Boente *et al.* 1997), we also considered Huber's ρ function to possibly alleviate this problem. Plugging in Huber's ρ into (2), we have a Huberized RCV ($HRCV$)

$$HRCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\rho(y_i - \hat{m}_\lambda^{(i)}(x_i)).$$

To determine the quantity C in $\rho(x)$, we used $C = 1.28 * MAD/.6745$, where MAD is the median absolute deviation of the residuals from an initial estimate (Hogg 1979) . Our initial estimate is obtained from the smooth using the ACV

estimate of λ. The constant 1.28/.6745 gives a conservative estimate since it corresponds to about 20% outliers from a normal distribution.

However, the $HRCV$ had a very little improvement over the ACV in simulation studies, and thus decided not to include the $HRCV$ in our investigation.

(For simulation results regarding the $HRCV$ please see the Appendix, pp.1-2).

3.3 Computationally Efficient Leave Out Schemes

We anticipate that the ordinary leave-one-out scheme for cross validation as described above will take a long time, especially for a dataset with large n. This is further exacerbated by the fact that the computation of the estimate for a fixed λ uses an iterative algorithm to solve the nonlinear optimization problem. Thus, we devise a scheme that leaves out many points at once and still gives a satisfactory result.

Our approach is motivated by idea of K-fold cross validation (Hastie *et al.* 2001) in which the data are divided into K randomly chosen blocks of (approximately) equal size. The data in each of the blocks is predicted by computing an estimate with that block left out. Our approach is to choose the blocks systematically so as to maximize the distance between x_i's within the blocks (which is easy to do with equispaced one dimensional x_i's as in our application). We call this method *systematic* K-fold cross validation. We expect the predictions at left out data points to be uniformly close to the original leave-one-out scheme. With an appropriate choice of K, for small bandwidths the influence of other left out points on prediction of a given value will be small because the effective span of the local smoother is small. For large bandwidths, the influence in prediction of left out points will be small because the effective span of the local smoother is large so that no point has much influence.

Define the sequence $(i : d : n) = \{i, i + d, \ldots, i + kd\}$ where k is the largest integer such that $i + kd \leq n$. Let $\hat{m}_\lambda^{(i:d:n)}$ denote the estimate with (x_i, y_i), (x_{i+d}, y_{i+d}), $\ldots$, (x_{i+kd}, y_{i+kd}) left out. Define the robust systematic K-fold

cross-validation with *phase* r (where $K = d/r$) by

$$RCV^{(d,r)}(\lambda) = \sum_{i\in(1:r:d)} \sum_{j\geq 0} \rho\left(y_{i+jd} - \hat{m}_\lambda^{(i:d:n)}\left(x_{i+jd}\right)\right).$$

If $r = 1$ so that $K = d$, we simply call this a systematic K-fold CV (without any reference to phase r). There are two parameters for this scheme: the d which determines how far apart are the left out x_i's, and the phase r which determines the sequence of starting values i. Note that we compute d/r curve estimates for each λ, which substantially reduces the time for computing the RCV function. We will have versions $RCV^{(d,r)}$ for each of the criterion functions discussed above.

The choice of d and r must be determined from data. Preliminary results show that $r > 1$ (which will not leave out all data points) does not give satisfactory results, and hence we will only consider the case $r = 1$.

(See Appendix pp.7-10 for tabulated results and pp.2-5 for discussions on the comparisons of $r = 1$ versus $r > 1$).

3.4 Methods to Evaluate the Smoothing Parameter Selection Schemes

For the simulation study, we use the following criteria to assess the performance of the smoothing parameter selection methods.

If we know the values of the true curve, we may compute the integrated squared error (ISE)

$$ISE(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(m(x_i) - \hat{m}_\lambda(x_i)\right)^2,$$

for each estimate and for various values of λ. (Note: many authors would refer to ISE in our formula as ASE or "Average Squared Error." However, the difference is so small that we will ignore it.) Similarly, we can compute integrated absolute error (IAE)

$$IAE(\lambda) = \frac{1}{n}\sum_{i=1}^{n}|m(x_i) - \hat{m}_\lambda(x_i)|.$$

We can determine which value of the λ will give us the "best" fit, based on L_1 (i.e., IAE) or L_2 (i.e., ISE) criteria. For comparison, we will take the squared root of ISE so that we show $\sqrt{ISE}$ in the results.

Thus, we may find $\lambda^*_{ISE} = \text{argmin}_\lambda ISE(\lambda)$, and $\lambda^*_{IAE} = \text{argmin}_\lambda IAE(\lambda)$ as our optimal λ values that gives us the best fit. We then make a comparison with the default λ values obtained by the methods in Section 3.1. In addition, we compute the $\hat{\lambda}$'s from the ACV (call this $\hat{\lambda}_{ACV}$), and compare them with the optimal values. To further evaluate the performances of ACV, we may compute the "loss functions" $ISE(\hat{\lambda})$ ($ISE(\lambda)$ with $\hat{\lambda}$ as its argument) and $IAE(\hat{\lambda})$, where $\hat{\lambda}$ is obtained from each of default or ACV. We then compare $ISE(\hat{\lambda})$ against $ISE(\lambda^*_{ISE})$ (and similarly for IAE).

It may be easier to employ what we call an "inefficiency measure,"

$$\text{Ineff}_{ISE}(\hat{\lambda}) = \frac{ISE(\hat{\lambda})}{\inf_\lambda ISE(\lambda)} \tag{3}$$

where in our case $\hat{\lambda} = \hat{\lambda}_{ACV}$, but keep in mind that $\hat{\lambda}$ may be obtained by any method. Note that this compares the loss for $\hat{\lambda}$ with the best possible loss for the given sample. We can easily construct $\text{Ineff}_{IAE}(\hat{\lambda})$, an inefficiency measure for IAE, as well. The $\text{Ineff}_{ISE}(\hat{\lambda})$ (or $\text{Ineff}_{IAE}(\hat{\lambda})$) has distributions on $[0, \infty)$, and if this distribution is concentrated near 1, then we have a confidence in the methodology behind $\hat{\lambda}$. Again, we will present $\sqrt{\text{Ineff}_{ISE}(\hat{\lambda})}$ in all the results, so that it is in the same unit as the $\text{Ineff}_{IAE}(\hat{\lambda})$.

4 Simulations and Application

4.1 Pilot Study

Before performing a large scale simulation study, we first did a pilot study for several reasons. First, we want to make sure that all the robust smoothers work well on our data. Second, we want to look at the performance of default smoothing parameter selection procedures from the R routines to see if they should be included in the larger scale simulation study. Lastly, in examining

the robust cross validation, we would like to see the effect of various leave out schemes in terms of speed and accuracy. We use the simulated data for this pilot study.

4.1.1 Simulation Models

There are two types of simulation models we use.

First, we construct a simulated model which resemble the data in our problem, because we are primarily interested in our application. To this end, we first pick a real data set (noisy data from one spectroscopic curve) that was shown in Figure 1 and fit a robust smoother, with the smoothing parameter chosen by inspection. We use the resulting smooth curve, $m(x)$, as the "true curve" in this simulation, and then we carefully study the residuals from this fit and get an approximate error distribution to generate simulated data points. We determined that most of the errors were well modeled by a normal distribution with mean 0 and variance 2025. We then selected 40 x values at random between $x = 420$ and $x = 1670$ and added in an independent gamma random variate with mean 315 and variance 28,350. Additionally, we put in two gross outliers similar to the ones encountered in raw data of Figure 1. We call this the Simulation 1. See Figure 2 (a), and note how similar our simulated data is to the real data. We have also chosen a fitted curve based on another sample data (data from a different spectroscopic curve), with an error distribution very similar to Simulation 1. See Figure 2 (b). Call this Simulation 2.

The second type of simulation models is based purely on mathematical models. We also have two curves for this type. First, we choose a beta curve, with the true curve as $m(x) = 700\text{beta}(3, 30)(x), 0 \leq x \leq 1$ (where beta(3,30) is the pdf of Beta(3,30) distribution), which looks very much like some of the fitted curves from real data. Second, we have included so-called a chirp function $(m(x) = 50(1-x)\sin(\exp(3(1-x))), 0 \leq x \leq 1)$, just to see how our methodology works in very general setting. Note the scale difference in the chirp function, so we adjusted the error distribution accordingly by changing the scale. We then

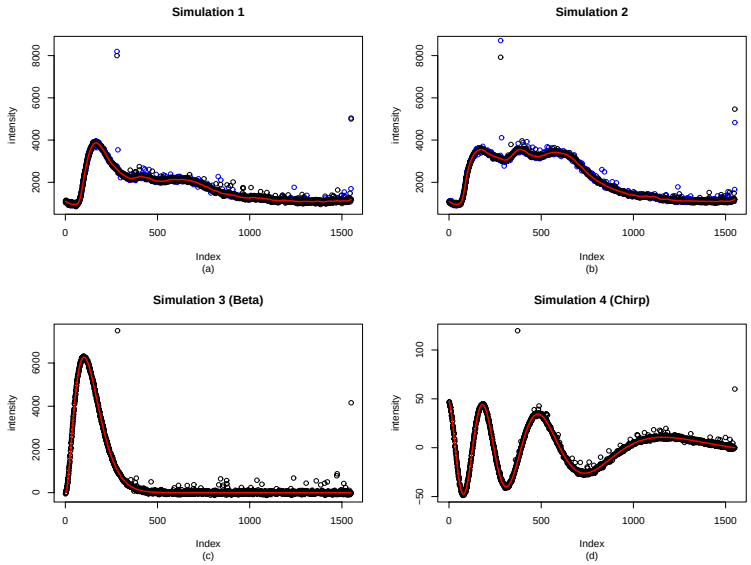

Figure 2: All simulation models considered here. The blue points in (a) and (b) are the real data points, whereas the black points are simulated data points.

create the error distributions closely resembling the errors in previous simulation models. We call the simulation based on beta function the Simulation 3, and we call the simulation with chirp function the Simulation 4. See Figure 2 (c) and (d).

In all simulation examples in this paper, we have $n = 1,550$ equispaced points.

4.1.2 Comparing Robust Smoothers and Smoothing Parameter Selection Methods

We compare the robust smoothers from Section 2 by applying them to the simulated data and comparing their computation time and performance.

Let us first step through some of the details regarding the smoothers. First, the robust smoothing splines, using the R function `qsreg`, has a default method as described in Section 3.1. It uses a range of 80 λ values as candidates for the default $\hat{\lambda}$. Since the range is wide enough to cover all the methods we try, we

use these same λ values for all investigation of `qsreg`.

Next, we consider the robust LOESS using `loess` function in R. In this case, the program does not supply us with the λ values, so we used a range from 0.01 to 0.1, with length of 80. This range is sufficiently wide to cover all the reasonable λ values and more.

When we use COBS with the R function `cobs`, we do need to make few adjustments. First, we tried both the L_1 (`degree=1`) and L_∞ (`degree=2`) versions of COBS, and the L_1 version was quite rough with discontinuous derivatives, so was dropped from further consideration. In addition, recall that for COBS, not only we have to determine λ but also the number of knots N as well. The default for the number of knots is 20 ($N = 20$), and we use this N for the default (BIC) method for the smoothing parameter selection of λ (using both the defaults by fixing $N = 20$ and λ selected by BIC). However, if we fix the number of knots at the default level, it gives unsatisfactory results, and some preliminary studies has shown that the increasing the number of knot to 50 ($N = 50$) will make the COBS fit better. But either way, the computing time of this smoother suffers dramatically. The computing times for obtaining the estimates (for one curve) were 0.03 seconds for `qsreg`, 0.06 seconds for `loess`, but 0.16 seconds for `cobs` (for both $N = 20$ and $N = 50$). For these reasons, we decided to increase the number of knots to 50 for the COBS experiments. Here, the candidate λ values range from 0.1 to 1,000,000.

Now, we present the L_1 and L_2 loss functions ($IAE(\lambda)$ and $\sqrt{ISE(\lambda)}$) for each of the $\hat{\lambda}$ values obtained by default and ACV, and compare them with optimal λ values. The results are shown in Table 1. This reveals that all default methods perform very poorly, as in Figure 1. This is especially apparent in default methods for robust smoothing splines and robust LOESS with fixed $\lambda = 3/4$. The default method for COBS (with $N = 20$) also does not work well. If we increase to number of knots to 50 ($N = 50$), then the default is not as bad as the default methods of the competing robust smoothers. However, the default method still performs more poorly than the ACV method. Furthermore, COBS's overall performance based on CV is poorer than those of robust

Table 1: A table of L_1 and L_2 loss function values for default, ACV, and optimum (IAE or ISE) criteria. For the COBS default, two sets of values are shown; Default with $N = 20$ or Default.50 where $N = 50$.

		Simulation 1		Simulation 2		Simulation 3		Simulation 4	
		IAE	$\sqrt{}$ISE	IAE	$\sqrt{}$ISE	IAE	$\sqrt{}$ISE	IAE	$\sqrt{}$ISE
qsreg	Default	24.45	31.80	16.30	21.05	23.46	29.99	0.53	0.69
	ACV	9.90	12.64	7.69	9.47	9.74	13.05	0.19	0.24
	Optimum	9.89	12.16	6.80	8.48	8.39	12.73	0.18	0.24
loess	Default	229.05	478.19	194.44	324.32	427.32	914.51	13.15	19.57
	ACV	8.61	10.76	6.19	7.50	8.79	11.30	0.19	0.24
	Optimum	8.12	10.33	5.73	6.86	7.92	11.27	0.18	0.24
cobs	Default	68.36	162.19	53.22	118.64	48.09	162.83	2.29	6.32
	Default.50	14.26	23.65	10.65	18.76	7.88	13.79	0.27	0.47
	ACV	10.11	12.72	9.52	14.72	6.98	9.83	0.19	0.25
	Optimum	10.11	12.72	9.45	14.71	6.98	9.80	0.19	0.25

smoothing splines or robust LOESS.

We remarked that COBS suffers from lack of speed compared to its competitors. This deficiency is not overcome by improvements in performance. Given all the complication with COBS, we dropped it from the rest of the pilot study and the simulations.

All in all, we conclude that the ACV method gives very satisfactory results. We see clearly that any method that does not incorporate the cross validation scheme does poorly. The only disadvantage of the leave-one-out cross validation methods is its computation time. Nevertheless, we demonstrate in the next section that there are ways to overcome this deficiency of ACV method.

4.1.3 Systematic K-fold Method

We investigate $ACV^{(d,r)}$ presented in Section 3.3 for faster computation and comparable performance.

Recall that we will only consider $r = 1$, the systematic K-fold scheme. (see

Table 2: A table of loss functions in Simulation 1, using robust smoothing splines (qsreg).

	ACV		Time
Scheme	IAE	$\sqrt{\text{ISE}}$	(seconds)
Default	24.45	31.80	2.96
Full LOO	9.90	12.64	4,477.47
$d = 50, r = 1$	9.90	12.64	145.50
$d = 25, r = 1$	9.89	12.16	70.53
$d = 5, r = 1$	9.89	12.16	12.26
Optimum	9.89	12.16	

Section 3.3). For our problem, with $d = 5$ and $r = 1$ (the systematic 5-fold CV), $\hat{\lambda}$ is almost optimal, but the computation is 365 times faster than the full leave-one-out (LOO) scheme! And the computation time is proportional (to a high accuracy) to how many times we compute a robust smoothing spline. Table 2 confirms that the performance of K-fold CV schemes ($d = K$, $r = 1$) are superior to default methods and comparable to the full LOO method.

Hence, based on all the results presented, we use the systematic K-fold CV as a main alternative to the full LOO CV.

(For more details and the results on other Simulations, see pp.2-10 of the Appendix).

4.1.4 Comparing Systematic K-fold CV with Random K-fold CV

We would like to see how the random K-fold schemes compare to our systematic schemes. In particular, since the systematic 5-fold CV worked very well in our example, we compare systematic 5-fold CV against the random 5-fold CV.

We take a $\hat{\lambda}$ from each of 100 draws (i.e., 100 different partitions) of random 5-fold ACVs. Then, we compute the inefficiency measure (3) introduced in the previous section, for both IAE and ISE. We do this for each of the 100 draws, and we compare them with the inefficiency measure of the systematic 5-fold CV.

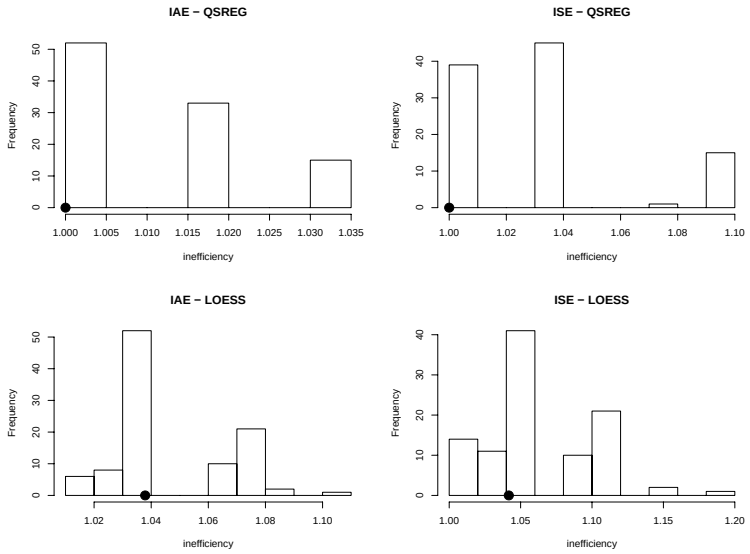

Figure 3: A histogram of inefficiencies obtained from the random 5-fold ACV. The systematic 5-fold inefficiency value is shown as a dot. The results are based on 100 draws of random 5-fold CVs, in Simulation 1.

We present histograms of the inefficiency values in Figure 3. The results in the figures suggest that the systematic 5-fold does well relative to random 5-fold and optimal value. Experience with other values of K and other simulations yielded similar results.

(For details, see Appendix, pp.11-17).

One problem with the random K-fold CV is that it introduces another source of variation that can lead to an undesirable consequence. For example, when the random K-fold CV randomly leaves out many points that are in the neighborhood of each other, it will do a poor job of predicting those left out points and hence produces a suboptimal result. It is known that a random K-fold CV result can be hugely biased for the true prediction error (Hastie *et al.* 2001). Therefore, we decided not to consider the random K-fold CV any further.

Table 3: A table comparing robust smoothers and loss functions and the mean and median inefficiency measure values.

	$\sqrt{E[ISE(\hat{\lambda})]}$	$E[IAE(\hat{\lambda})]$	$\sqrt{\text{Ineff}_{ISE}(\hat{\lambda})}$		$\text{Ineff}_{ISE}(\hat{\lambda})$	
			mean	median	mean	median
`qsreg`	12.61	9.89	1.11	1.05	1.05	1.03
`loess`	10.45	8.21	1.07	1.03	1.03	1.01

4.2 Large Simulation Study

We now report on the results of a large scale simulation study to further assess the performance of the robust estimators. Specifically, we compare the robust smoothing spline and robust LOESS.

The data is obtained just as in Section 4.1.1, where we take a vectorized "true" curve $m(x)$ and add a vector of random errors to it (with error distribution as described in that section), and repeat this M times with the same $m(x)$.

If we obtain $ISE(\lambda)$ and $IAE(\lambda)$ functions for each simulation, we can easily estimate mean integrated squared error ($E[ISE(\lambda)]$) by averaging over the number of replications, M. The $E[IAE(\lambda)]$ may likewise be obtained.

Our results are based on $M = 100$.

4.2.1 Detailed Evaluation with Full Leave-One-Out CV.

We begin by reporting our results based on the full LOO validation with ACV for the Simulation 1 model (we defer all other results until next subsection).

We want to assess the performance of the two robust smoothers of interest by comparing $E[ISE(\hat{\lambda})]$ values, with $\hat{\lambda} = \hat{\lambda}_{ACV}$, and similarly for $E[IAE(\hat{\lambda})]$. Also, we present the results of the inefficiency measures. Again, we will take squared roots of those quantities involving ISE so that we present $\sqrt{E[ISE(\hat{\lambda})]}$ and $\sqrt{\text{Ineff}_{ISE}(\hat{\lambda})}$ in the tables. The results are presented in Table 3. Clearly, all the integrated error measures of robust LOESS are lower than those of the robust smoothing splines, although these results by themselves do not indicate

Table 4: A table of median values of inefficiencies in all simulations. Each entry consists of either $\sqrt{\text{Ineff}}_{ISE}(\hat{\lambda})$ or $\text{Ineff}_{IAE}(\hat{\lambda})$ with $\hat{\lambda}$ value corresponding to different rows.

		Simulation 1		Simulation 2	
		IAE	$\sqrt{}$ ISE	IAE	$\sqrt{}$ ISE
qsreg	Default	2.59	2.68	2.34	2.41
	Full LOO	1.03	1.05	1.02	1.03
	$d = 5, r = 1$	1.02	1.02	1.00	1.00
loess	Default	28.67	47.29	32.95	43.48
	Full LOO	1.01	1.03	1.02	1.01
	$d = 5, r = 1$	1.01	1.03	1.00	1.00
		Simulation 3		Simulation 4	
		IAE	$\sqrt{}$ ISE	IAE	$\sqrt{}$ ISE
qsreg	Default	2.36	2.17	2.67	2.54
	Full LOO	1.09	1.06	1.05	1.04
	$d = 5, r = 1$	1.05	1.06	1.02	1.02
loess	Default	46.88	73.53	70.73	77.85
	Full LOO	1.04	1.02	1.05	1.01
	$d = 5, r = 1$	1.02	1.04	1.03	1.03

which smoothing method is more accurate.

(See Appendix, pp.18-20, for further details and results).

4.2.2 Results on All Simulations.

We now report on all simulation results, including those of the systematic K-fold CV as described in section 3.3. See Table 4 for the results, where we only report 5-fold systematic CV since 25- and 50-fold results are very similar. The result is that all of $\sqrt{\text{Ineff}}_{ISE}(\hat{\lambda})$ and $\text{Ineff}_{IAE}(\hat{\lambda})$ values are near 1 for all $\hat{\lambda}$ values obtained from full LOO schemes, for ACV. This is true for all four simulation models we considered. The numbers are very similar for all the K-fold CV

schemes (where we considered $K = 5, 25$, and 50).

(For the results on 25- and 50-fold CVs, see Appendix pp.21-22).

In contrast, the $\sqrt{\text{Ineff}}_{ISE}(\hat{\lambda})$ and $\text{Ineff}_{IAE}(\hat{\lambda})$ values for $\hat{\lambda}_{\text{default}}$ are at least 2 and can be as much as 77!

The results again demonstrate and reconfirm what we already claimed: our cross validation method is far superior to default methods and is basically as good as - if not better than - the full LOO method.

In conclusion, we have seen that robust LOESS gives the best accuracy for our application, although the computational efficiency of robust smoothing splines is better. At this point, we are not sure if the increase in accuracy is worth the increase in computational effort, since both procedures work very well. Either one will serve well in our application and in many situations to be encountered in practice. For our application, we recommend the systematic 5-fold CV scheme with the robust LOESS, based on our simulation results.

4.3 Application

Now, we discuss the results of applying our methods to real data. We have done most of the work in previous sections, and all we need for the application is to apply the smoother with appropriate smoothing parameter selection procedure to other curves.

Following the recommendations from the previous section, we used the robust LOESS with ACV based on systematic 5-fold CV ($d = 5$, $r = 1$) for smoothing parameter selection. We found that this worked well. See Figure 4 for a sample of results.

We have also performed some diagnostics from fitting to the real data, such as plotting residuals versus fitted values and Quantile-Quantile plot (Q-Q plot), as well as looking at the autocorrelation of the residuals to determine whether there is much correlation between adjacent grid points. The diagnostics did not reveal any problems, and therefore we are confident of the usefulness of our method. (The details are in Appendix p.20 and pp.23-24).

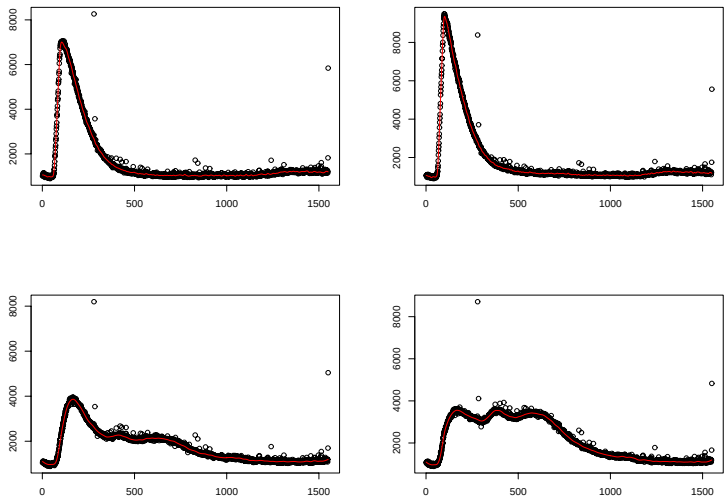

Figure 4: Plots of raw data with robust smoother superimposed in four of the emission spectra.

5 Conclusion and Discussion

We have seen how well our robust cross validation schemes perform in our application. Specifically, we are able to implement a smoothing parameter selection procedure in a fast, accurate way. We can confidently say that with either robust smoothing splines or robust LOESS using the ACV based on systematic K-fold cross validation works well in practice.

There may exists some questions raised regarding our method. One such issue may be the fact that we have only considered absolute value and Huber's rho functions in the RCV function. But we have shown that ACV with absolute value works very well, and most other criterion functions will undoubtedly bring more complications, so we do not foresee any marked improvements over ACV. In addition, there may be some concern about the choice of d and r in our leave out schemes. We obtained a ranges of candidate d and r values by trial and error, and this must be done in every problem. Nevertheless, this small preprocessing step will be beneficial in the long run. As demonstrated in the paper, we can

save an enormous amount of time once we figure out the appropriate values for d and r to be used in our scheme. Furthermore, if we need to smooth many different curves as was done here, then we need only to do this for some test cases. Some other issues such as use of default smoothing parameter selection methods and the use of random K-fold has been resolved in the course of the paper.

Now, we would like to discuss some possible future research directions. There exist other methods of estimating smoothing parameters in robust smoothers, such as Robust C_p (Cantoni and Ronchetti, 2001) and plug-in estimators (Boente *et al.*, 1997). We have not explored these in the present work, but they could make for an interesting future study.

Another possible problem to consider is the theoretical aspect of robust cross validation schemes. Leung *et al.* (1993) gives various conjectures, but as far as we know, there have been no results yet.

Also, extending the methods proposed here to the function estimation on a multivariate domain presents some challenges. In particular, the implementation of systematic K-fold cross validation was easy for us since our independent variable values were 1-dimensional and equally spaced.

In conclusion, we believe that our methods are flexible and easily implementable in a variety of situations. We have been concerned with applying robust smoothers only to the spectroscopic data from our application so far, but they should be applicable to other areas which involve similar data to ours.

Acknowledgements

The research is supported by the National Cancer Institute Grant No. PO1-CA82710 and by the National Science Foundation Grant No. DMS-0505584. We would like to thank Dan Serachitopol, Michele Follen, David Scott, and Rebecca Richards-Kortum for their assitance, suggestions, and advice.

References

[1] Boente, G., Fraiman, R., and Meloche, J. (1997) Robust plug-in bandwidth estimators in nonparametric regression, *Journal of Statistical Planning and Inference*, **57**, 109-142.

[2] Bussian, B., and Härdle, W. (1984), Robust smoothing applied to white noise and single outlier contaminated Raman spectra, *Applied Spectroscopy*, **38**, 309-313.

[3] Cantoni, E., and Ronchetti, E. (2001), Resistant selection of the smoothing parameter for smoothing splines, *Statistics and Computing*, **11**, 141-146.

[4] Chambers, J., and Hastie, T. (1992), *Statistical Models in S*, Wadsworth & Brooks/Cole: Pacific Grove, CA.

[5] Chang, S.K., Follen, M., Utzinger, U., Staerkel, G., Cox, D.D., Atkinson, E.N., MacAulay, C., and Richards-Kortum, R.R. (2002), Optimal excitation wavelengths for discrimination of cervical neoplasia, *IEEE Transaction on Biomedical Engineering*, **49**, 1102-1111.

[6] Cleveland, W.S. (1979), Robust locally weighted regression and smoothing scatterplots, *Journal of the American Statistical Association*, **74**, 829-836.

[7] Cox, D.D. (1983), Asymptotics for M-type smoothing splines, *Annals of Statistics*, **11**, 530-551.

[8] de Boor, C. (1978), *A Practical Guide to Splines*, Springer: New York.

[9] Grossman, N., Ilovitz, E., Chaims, O., Salman, A., Jagannathan, R., Mark, S., Cohen, B., Gopas, J., and Mordechai, S. (2001), Fluorescence spectroscopy for detection of malignancy: *H-ras* overexpressing fibroblasts as a model, *Journal of Biochemical and Biophysical Methods*, **50**, 53-63.

[10] Hastie, T., Tiibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning*, Springer: New York.

[11] He, X., and Ng, P. (1999), COBS: Qualitatively constrained smoothing via linear programming, *Computational Statistics*, **14**, 315-337.

[12] Hogg, R.V. (1979), Statistical robustness, *American Statistician*, **33**, 108-115.

[13] Leung, D., Marriot, F., and Wu, E. (1993), Bandwidth selection in robust smoothing, *Journal of Nonparametric Statistics*, **2**, 333-339.

[14] Oh, H., Nychka, D., Brown, T., and Charbonneau, P. (2004), Period analysis of variable stars by robust smoothing, *Journal of the Royal Statistical Society, Series C*, **53**, 15-30.

[15] Oh, H., Nychka, D., and Lee, T. (2007), The Role of pseudo data for robust smoothing with application to wavelet regression, *Biometrika*, **94**, 893-904.

[16] Schwarz, G., (1978), Estimating the dimension of a model, *Annals of Statistics*, **6**, 461-464.

[17] Simonoff, J., (1996), *Smoothing Methods in Statistics*, Springer: New York.

[18] Wang, F.T., and Scott, D.W. (1994), The L_1 method for robust nonparametric regression, *Journal of the American Statistical Association*, **89**, 65-76.

Supplemental Material for *Robust Smoothing: Smoothing Parameter Selection and Applications to Fluorescence Spectroscopy*

Jong Soo Lee and Dennis D. Cox

A Appendix

This Appendix contains some supplementary results not shown in the main text.

A.1 Huberized Robust Cross Validation

Recall that a Huberized *RCV* ($HRCV$) is defined as

$$HRCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \rho_H(y_i - \hat{m}_\lambda^{(i)}(x_i)).$$

In the main text, we have not included any analysis regarding $HRCV$

We first present the L_1 and L_2 loss functions ($IAE(\lambda)$ and $ISE(\lambda)$) for each of the λ values obtained by default, ACV, and $HRCV$, and compare them with optimal λ values. For comparison, we will take the squared root of ISE so that we show $\sqrt{ISE}$ in the results. The results are shown in Table 1.

We now investigate the performance of $HRCV$ to see how it compares with those of ACV. First, we perform various $HRCV^{(d,r)}$ schemes, using the same (d,r) values in $ACV^{(d,r)}$ above. The results of loss function values are also in Table 2 (at appropriate columns). Comparing with ACV results, we find that $HRCV$ method performs very similar to the ACV; in fact, the $HRCV$ losses seem to be lower.

Next, Figure 2 gives a comparison of the ACV and $HRCV$ curves. The $HRCV$ curve is slightly smoother than the ACV curve while preserving most of the features ACV curve has. This is true for comparing both (a) against (b) and (c) against (d) in Figure 2. We note that both $HRCV$ and the systematic K-fold has some smoothing effect, so that we see that (d) is the smoothest of all. Moreover, the $\hat{\lambda}_{ACV}$ and $\hat{\lambda}_{HRCV}$ values are equal. We will see more evidence of this similarity between ACV and $HRCV$ in the large simulation study.

The computational time of ACV and $HRCV$ are virtually identical (and hence presented only the times from ACV in Table 2). But with our proposal for getting a scale parameter for $HRCV$, it is necessary to have a fit from the ACV first and compute the residuals, a disadvantage for $HRCV$.

Table 1: A table of L_1 and L_2 loss function values for default, ACV, $HRCV$, and optimum (IAE or ISE) criteria. For the COBS default, two sets of default values are shown (default N or $N = 50$). The ISE values are square-rooted.

		Simulation 1		Simulation 2	
		IAE	$\sqrt{}$ ISE	IAE	$\sqrt{}$ ISE
qsreg	Default	24.45	31.80	16.30	21.05
	ACV	9.90	12.64	7.69	9.47
	HRCV	9.90	12.64	6.94	8.58
	Optimum	9.89	12.16	6.80	8.48
loess	Default	229.05	478.19	194.44	324.32
	ACV	8.61	10.76	6.19	7.50
	HRCV	8.61	10.76	5.85	6.99
	Optimum	8.12	10.33	5.73	6.86
cobs	Default ($N = 20$)	68.36	162.19	53.22	118.64
	Default ($N = 50$)	14.26	23.65	10.65	18.76
	ACV	10.11	12.72	9.52	14.72
	HRCV	10.11	12.72	9.52	14.72
	Optimum	10.11	12.72	9.45	14.71
		Simulation 3		Simulation 4	
		IAE	$\sqrt{}$ ISE	IAE	$\sqrt{}$ ISE
qsreg	Default	23.46	29.99	0.53	0.69
	ACV	9.74	13.05	0.19	0.24
	HRCV	9.74	13.05	0.20	0.25
	Optimum	8.39	12.73	0.18	0.24
loess	Default	427.32	914.51	13.15	19.57
	ACV	8.79	11.30	0.19	0.24
	HRCV	8.61	11.27	0.19	0.24
	Optimum	7.92	11.27	0.18	0.24
cobs	Default ($N = 20$)	48.09	162.83	2.29	6.32
	Default ($N = 50$)	7.88	13.79	0.27	0.47
	ACV	6.98	9.83	0.19	0.25
	HRCV	7.02	9.80	0.19	0.25
	Optimum	6.98	9.80	0.19	0.25

We determine that ACV is the best overall smoothing parameter selection scheme. Although $HRCV$ is the best method for finding the good estimate of the λ, the disadvantage exist with the $HRCV$ in that we first need to run a robust smoother with a good estimate of λ to get the residuals which are used to select the scale parameter in Huber's ρ function. This means that we need to run ACV first to get a good estimate of λ. Furthermore, the improvement by $HRCV$ is very slight over ACV (as was seen in Table 1).

A.2 Discussion of the Choices of the d and r Values

Recall our scheme: Define the sequence $(i : d : n) = \{i, i + d, \ldots, i + kd\}$ where k is the largest integer such that $i+kd \leq n$. Let $\hat{m}_\lambda^{(i:d:n)}$ denote the estimate with

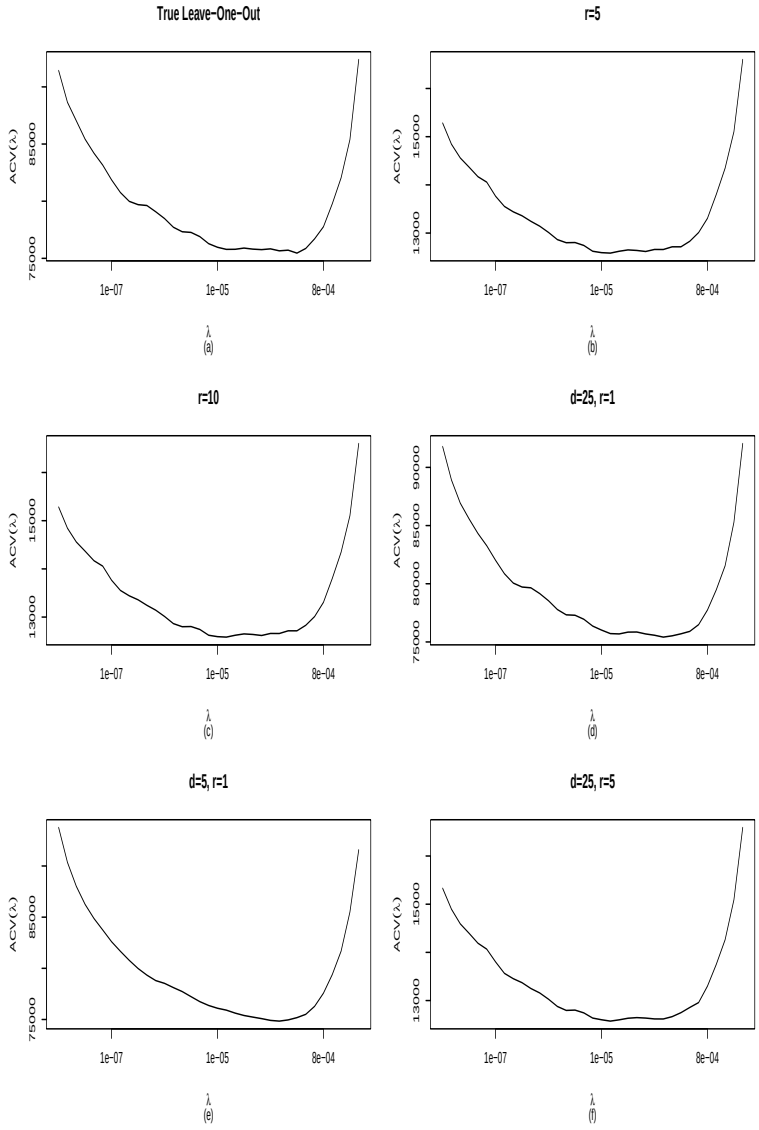

Figure 1: Plots of ACV curves with various leave out schemes. If a value of d is not given, then $d = n$, i.e. one data point at a time is left out.

(x_i, y_i), (x_{i+d}, y_{i+d}), ..., (x_{i+kd}, y_{i+kd}) left out. Define the robust systematic K-fold cross-validation with *phase* r (where $K = d/r$) by

$$RCV^{(d,r)}(\lambda) = \sum_{i \in (1:r:d)} \sum_{j \geq 0} \rho\left(y_{i+jd} - \hat{m}_{\lambda}^{(i:d:n)}\left(x_{i+jd}\right)\right).$$

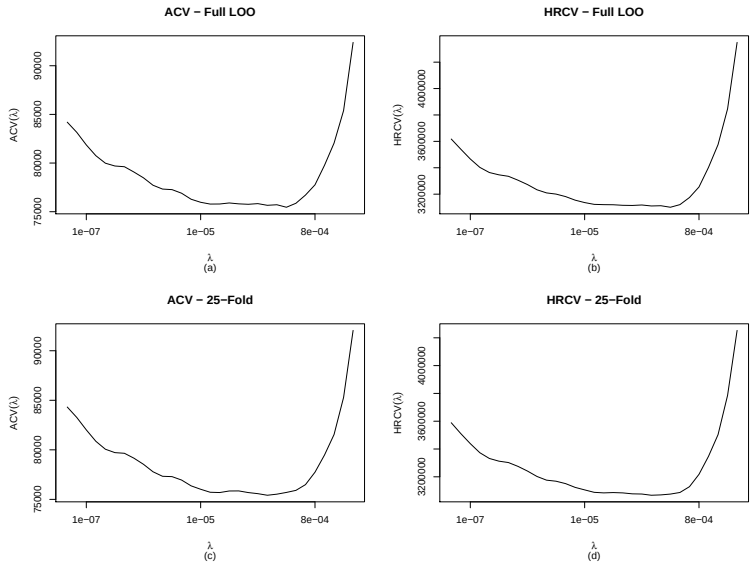

Figure 2: Comparing ACV with HRCV.

If $r = 1$ so that $K = d$, we simply call this a systematic K-fold CV (without any reference to phase r).

See Figure 3 for the graphical description of leave out schemes.

Let us discuss the choices of the d and r values for the $ACV^{(d,r)}$ and $HRCV^{(d,r)}$ scheme. We choose d and r based on "trial and error" from the data. For example, one way to determine the candidate (d,r) values is to fix one value and vary the other, and consider only those in a "reasonable" range. For our setting, we see that $d = 50$ and $r = 1$ resemble the full leave-one-out method reasonably well, and so we discard the scheme where $r = 1$ and d is over 50 (since the full LOO means $d = n$). Of course, different applications will dictate different values of (d,r), but we will see that this preprocessing step is well worth the time in practice.

In Table 2, we have loss functions from different $ACV^{(d,r)}$ along with those from default methods, the full leave-one-out ACV, and the optimal values, considering both IAE and ISE. We have also included their computation times. If the value of d is not given in the table, then $d = n$, i.e. we are leaving out only a single data point each time. We save some computational time by leaving out every rth point, and about n/r estimates are computed. We see that the smoothing parameter estimates do lose accuracy when $r \geq 5$. If we delete many points at once (e.g., $d = 50$ or $d = 25$), the computational time is cut dramatically, and with $r = 1$ we obtain good accuracy, even with small d values. In particular, with $d = 5$ and $r = 1$ (the systematic 5-fold CV), $\hat{\lambda}$ is the same as λ^*_{IAE} and λ^*_{ISE}, but the computation is 365 times faster than the true leave-one-out scheme! As expected, the computation time is proportional

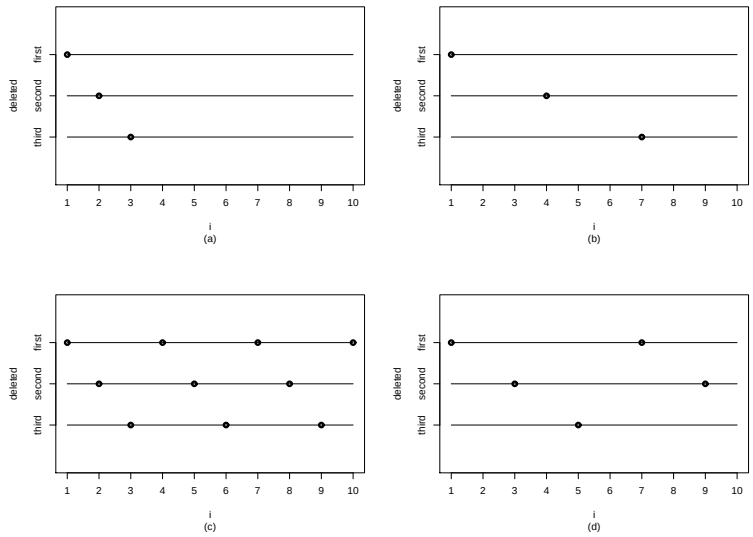

Figure 3: Plots of our cross validation schemes. Plot (a) describes the true leave-one-out case ($r = 1$, $d = n$). Plot (b) is the case $r = 3$, $d = n$. Plot (c) explains the case $r = 1$, $d = 3$. Plot (d) is the case $r = 2$, $d = 6$

(to a high accuracy) to how many times we compute a robust smoothing spline. Table 2 confirms that the performance of true leave-one out and systematic K-fold CV schemes ($d = K$, $r = 1$) are superior to default and other methods.

To supplement the numerical results, we look at the actual ACV functions for the various schemes. See Figure 1. Note that sometimes there seem to be multiple minima in the ACV function and can be wiggly. In this example, we see that the ACV curve with $d = 25$ and $r = 1$ (systematic 25-fold CV) very closely approximates the leave-one-out ACV curve. In fact, the results from Table 2 suggest that the systematic 5-fold and 25-fold CV actually performs better than true leave-one-out CV.

Now, if we look at the results of Simulation 1 for robust LOESS, we see that we get pretty much the same conclusion as we did when we used the robust smoothing splines, `qsreg`, except that robust loess is a little more computationally expensive (about twice as slow). For the results, see Table 3.

A.3 Results From Other Simulations

We obtain similar conclusions from all simulations (Simulations 1 to 4) as well. See the Tables 4 to 7.

Table 2: A table of loss functions in Simulation 1, using robust smoothing splines (qsreg). The times are based on ACV.

	ACV		HRCV		Time
Scheme	IAE	$\sqrt{\text{ISE}}$	IAE	$\sqrt{\text{ISE}}$	(seconds)
Default	24.45	31.80	24.45	31.80	2.96
Full LOO	9.90	12.64	9.90	12.64	4,477.47
$r = 5$	12.00	14.95	10.23	12.71	886.82
$r = 10$	12.89	15.85	12.00	14.95	443.46
$d = 50, r = 1$	9.90	12.64	9.90	12.64	145.50
$d = 25, r = 1$	9.89	12.16	9.89	12.16	70.53
$d = 5, r = 1$	9.89	12.16	9.89	12.16	12.26
$d = 50, r = 10$	12.40	15.34	12.40	15.34	14.26
$d = 25, r = 5$	12.00	14.95	10.59	13.16	13.87
Optimum	9.89	12.16			

Table 3: A table of loss functions in Simulation 1, using robust LOESS. The times are based on ACV. Compare with Table 2.

	ACV		HRCV		Time
Scheme	IAE	$\sqrt{\text{ISE}}$	IAE	$\sqrt{\text{ISE}}$	(seconds)
Default	229.05	478.19	229.05	478.19	0.08
Full LOO	8.61	10.76	8.61	10.76	8,349.17
$r = 5$	8.61	10.76	8.82	11.05	1,676.25
$r = 10$	8.71	10.90	9.39	11.83	841.63
$d = 50, r = 1$	8.61	10.76	8.61	10.76	268.55
$d = 25, r = 1$	8.43	10.54	8.43	10.54	133.20
$d = 5, r = 1$	8.43	10.54	8.29	10.37	23.22
$d = 50, r = 10$	8.71	10.90	9.22	11.60	26.90
$d = 25, r = 5$	8.71	10.90	8.71	10.90	26.63
Optimum	8.12	10.33	8.12	10.33	

Table 4: A table of loss functions for Simulation 1. This is a summary of results from Tables 2 and 3

		ACV		HRCV	
	Scheme	IAE	$\sqrt{\text{ISE}}$	IAE	$\sqrt{\text{ISE}}$
qsreg	Default	24.45	31.80	24.45	31.80
	Full LOO	9.90	12.64	9.90	12.64
	$r = 5$	12.00	14.95	10.23	12.71
	$r = 10$	12.89	15.85	12.00	14.95
	$d = 50, r = 1$	9.90	12.64	9.90	12.64
	$d = 25, r = 1$	9.89	12.16	9.89	12.16
	$d = 5, r = 1$	9.89	12.16	9.89	12.16
	$d = 50, r = 10$	12.40	15.34	12.00	14.95
	$d = 25, r = 5$	12.00	14.95	10.59	13.16
	Optimum	9.89	12.16	9.89	12.16
loess	Default	229.05	478.19	229.05	478.19
	Full LOO	8.61	10.76	8.61	10.76
	$r = 5$	8.61	10.76	8.82	11.05
	$r = 10$	8.71	10.90	9.39	11.83
	$d = 50, r = 1$	8.61	10.76	8.61	10.76
	$d = 25, r = 1$	8.43	10.54	8.43	10.54
	$d = 5, r = 1$	8.43	10.54	8.29	10.37
	$d = 50, r = 10$	8.71	10.90	9.22	11.60
	$d = 25, r = 5$	8.71	10.90	8.71	10.90
	Optimum	8.12	10.33	8.12	10.33

Table 5: A table of loss functions for Simulation 2.

		ACV		HRCV	
	Scheme	IAE	$\sqrt{}$ ISE	IAE	$\sqrt{}$ ISE
qsreg	Default	16.30	21.05	16.30	21.05
	Full LOO	7.69	9.47	6.94	8.58
	$r = 5$	7.69	9.47	7.35	9.11
	$r = 10$	7.35	9.11	7.35	9.11
	$d = 50, r = 1$	7.69	9.47	6.94	8.58
	$d = 25, r = 1$	7.35	9.11	6.94	8.58
	$d = 5, r = 1$	6.94	8.58	6.94	8.58
	$d = 50, r = 10$	7.35	9.11	7.35	9.11
	$d = 25, r = 5$	7.35	9.11	7.35	9.11
	Optimum	6.80	8.48	6.80	8.48
loess	Default	194.44	324.32	194.44	324.32
	Full LOO	6.19	7.50	5.85	6.99
	$r = 5$	6.31	7.69	6.08	7.34
	$r = 10$	7.27	9.01	6.19	7.50
	$d = 50, r = 1$	6.19	7.50	5.77	6.88
	$d = 25, r = 1$	6.19	7.50	5.77	6.88
	$d = 5, r = 1$	5.98	7.20	5.73	6.98
	$d = 50, r = 10$	6.46	7.91	6.08	7.34
	$d = 25, r = 5$	6.46	7.91	5.91	7.08
	Optimum	5.73	6.86	5.73	6.86

Table 6: A table of loss functions for Simulation 3.

		ACV		HRCV	
	Scheme	IAE	$\sqrt{}$ ISE	IAE	$\sqrt{}$ ISE
qsreg	Default	23.46	29.99	23.46	29.99
	Full LOO	10.66	13.59	9.74	13.05
	$r = 5$	10.66	13.59	9.74	13.05
	$r = 10$	10.66	13.59	9.74	13.05
	$d = 50, r = 1$	10.66	13.59	9.74	13.05
	$d = 25, r = 1$	10.66	13.59	9.74	13.05
	$d = 5, r = 1$	10.19	13.32	8.99	12.73
	$d = 50, r = 10$	10.66	13.59	9.74	13.05
	$d = 25, r = 5$	10.19	13.32	9.74	13.05
	Optimum	8.39	12.73	8.39	12.73
loess	Default	427.32	914.51	427.32	914.51
	Full LOO	8.79	11.30	8.61	11.27
	$r = 5$	8.12	12.67	8.46	11.32
	$r = 10$	8.72	24.85	8.09	11.81
	$d = 50, r = 1$	8.28	11.27	8.28	11.27
	$d = 25, r = 1$	8.28	11.27	8.28	11.27
	$d = 5, r = 1$	8.28	11.27	8.28	11.27
	$d = 50, r = 10$	8.72	24.85	8.09	11.81
	$d = 25, r = 5$	8.12	12.67	8.28	11.27
	Optimum	7.92	11.27	7.92	11.27

Table 7: A table of loss functions for Simulation 4.

		ACV		HRCV	
	Scheme	IAE	$\sqrt{\text{ISE}}$	IAE	$\sqrt{\text{ISE}}$
qsreg	Default	0.53	0.69	0.53	0.69
	Full LOO	0.19	0.24	0.20	0.25
	$r = 5$	0.19	0.24	0.20	0.25
	$r = 10$	0.19	0.24	0.20	0.25
	$d = 50, r = 1$	0.19	0.24	0.20	0.25
	$d = 25, r = 1$	0.19	0.24	0.19	0.24
	$d = 5, r = 1$	0.19	0.24	0.19	0.24
	$d = 50, r = 10$	0.19	0.24	0.20	0.25
	$d = 25, r = 5$	0.19	0.24	0.20	0.25
	Optimum	0.18	0.24	0.18	0.24
loess	Default	13.15	19.57	13.15	19.57
	Full LOO	0.19	0.24	0.19	0.24
	$r = 5$	0.18	0.24	0.18	0.24
	$r = 10$	0.18	0.24	0.18	0.24
	$d = 50, r = 1$	0.23	0.28	0.19	0.24
	$d = 25, r = 1$	0.19	0.24	0.19	0.24
	$d = 5, r = 1$	0.19	0.24	0.19	0.24
	$d = 50, r = 10$	0.18	0.24	0.18	0.24
	$d = 25, r = 5$	0.18	0.25	0.19	0.24
	Optimum	0.18	0.24	0.18	0.24

A.4 Random vs Systematic K-fold

See Figures 4 to 15. The figures contain 5-,25-, and 50-fold systematic CV compared with the corresponding random CVs. For the random K-fold CVs, we compute the inefficiency measure for each of the 100 draws, and we compare them with the inefficiency of the systematic K-fold CV. This is done by creating a histogram of inefficiency measures from the 100 draws, and indicating the inefficiency of the systematic K-fold CV by a dot. The results in the figures suggest that the systematic K-fold does well relative to random K-fold and optimal value (Ineff = 1). Even when the random K-fold can obtain results that are better than the systematic K-fold results, it can as well produce much worse results.

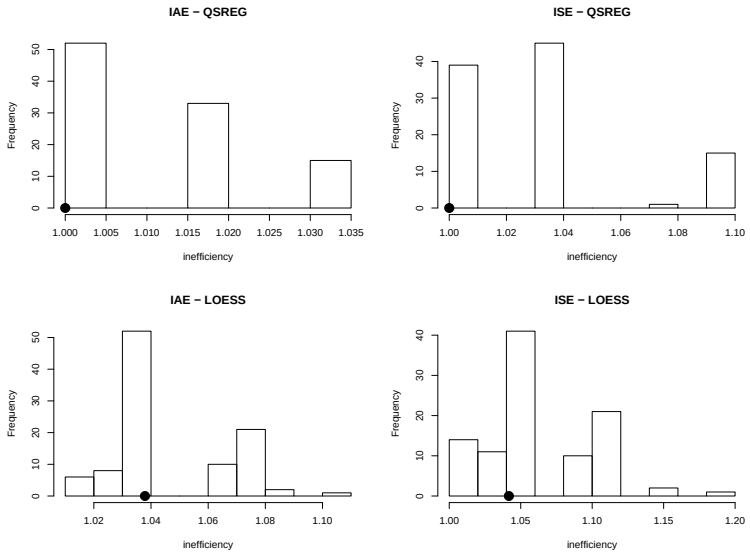

Figure 4: A histogram of inefficiencies obtained from the random 5-fold ACV, to be compared with systematic 5-fold value (dot). From Simulation 1. Based on 100 draws.

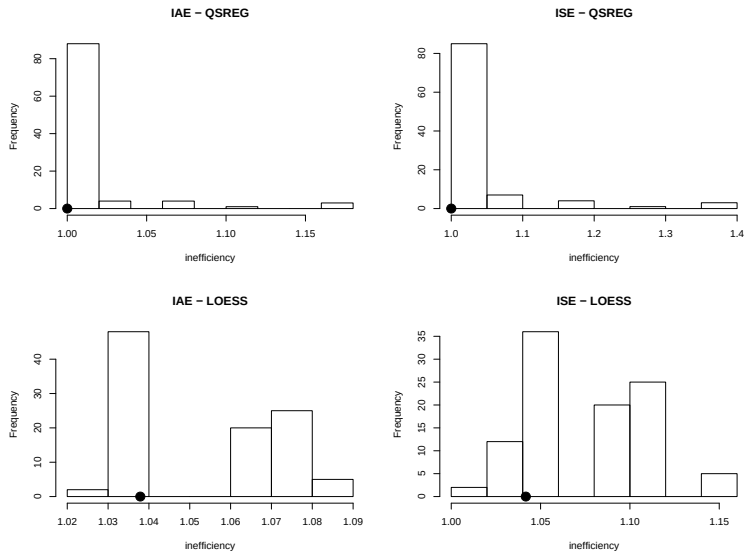

Figure 5: A histogram of inefficiencies obtained from the random 25-fold ACV, to be compared with systematic 25-fold value (dot). From Simulation 1. Based on 100 draws.

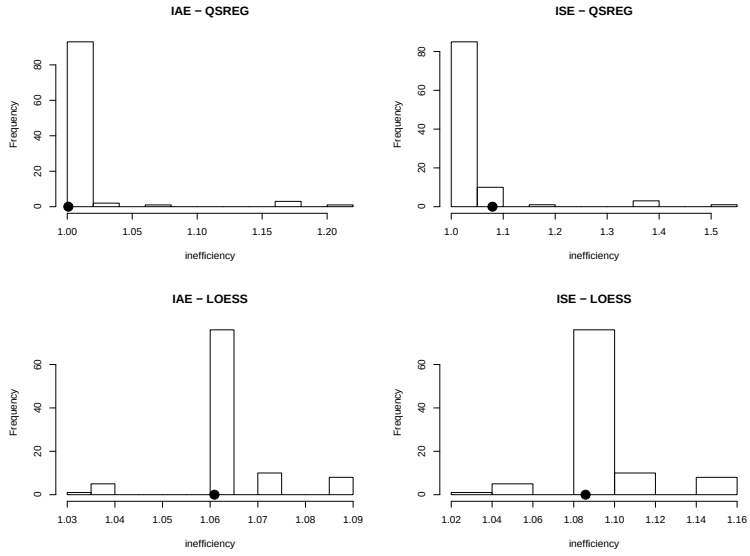

Figure 6: A histogram of inefficiencies obtained from the random 50-fold ACV, to be compared with systematic 50-fold value (dot). From Simulation 1. Based on 100 draws.

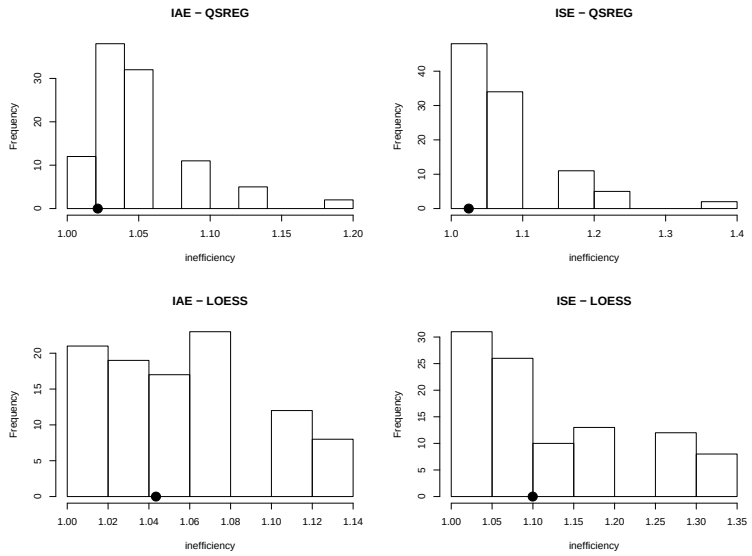

Figure 7: A histogram of inefficiencies obtained from the random 5-fold ACV, to be compared with systematic 5-fold value (dot). From Simulation 2. Based on 100 draws.

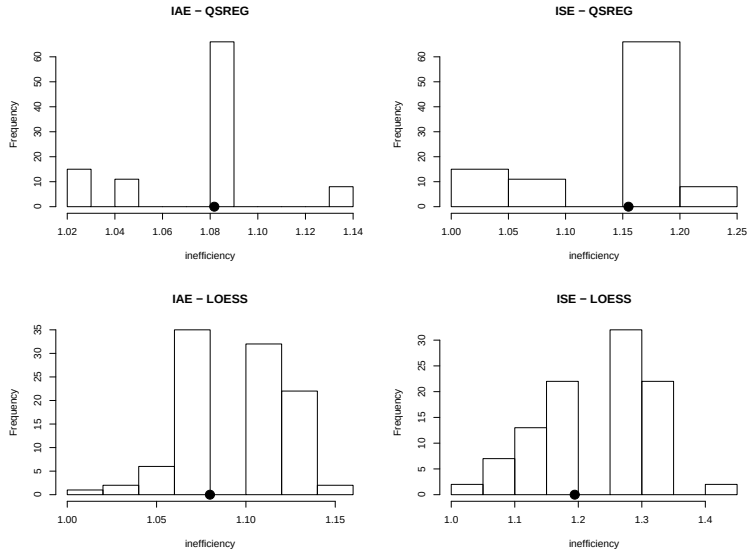

Figure 8: A histogram of inefficiencies obtained from the random 25-fold ACV, to be compared with systematic 25-fold value (dot). From Simulation 2. Based on 100 draws.

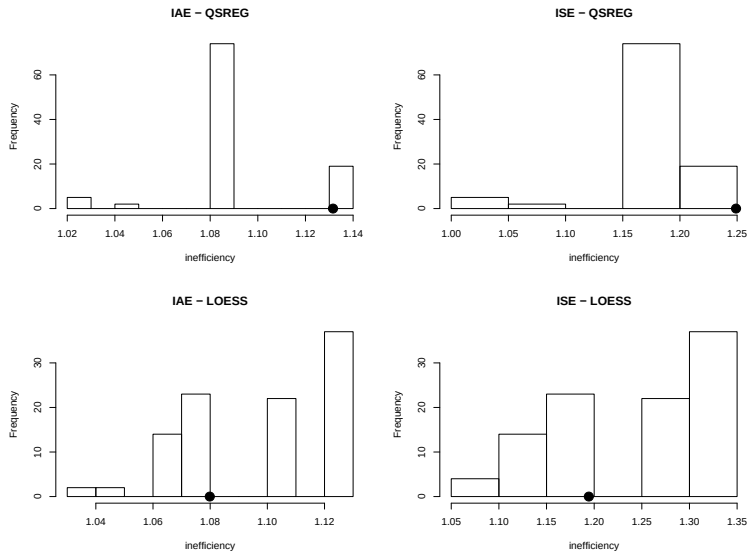

Figure 9: A histogram of inefficiencies obtained from the random 50-fold ACV, to be compared with systematic 50-fold value (dot). From Simulation 2. Based on 100 draws.

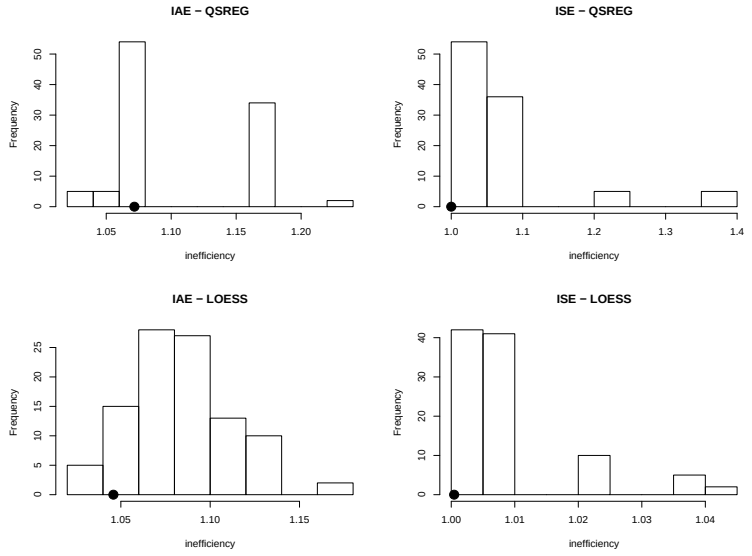

Figure 10: A histogram of inefficiencies obtained from the random 5-fold ACV, to be compared with systematic 5-fold value (dot). From Simulation 3. Based on 100 draws.

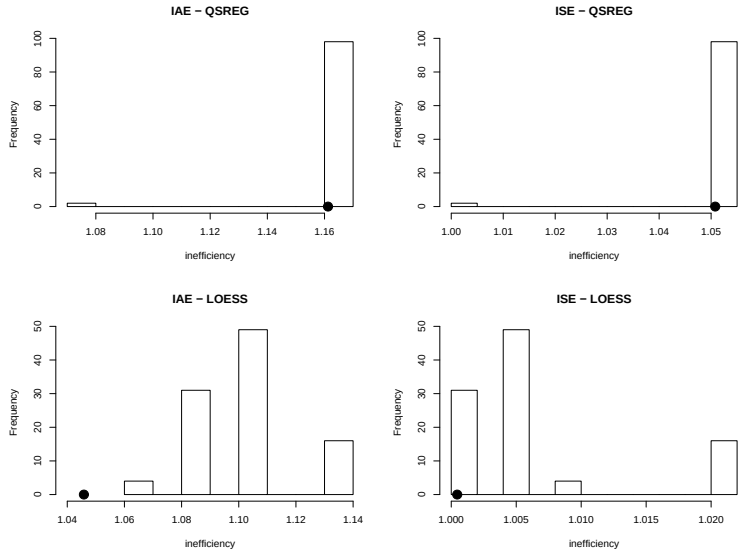

Figure 11: A histogram of inefficiencies obtained from the random 25-fold ACV, to be compared with systematic 25-fold value (dot). From Simulation 3. Based on 100 draws.

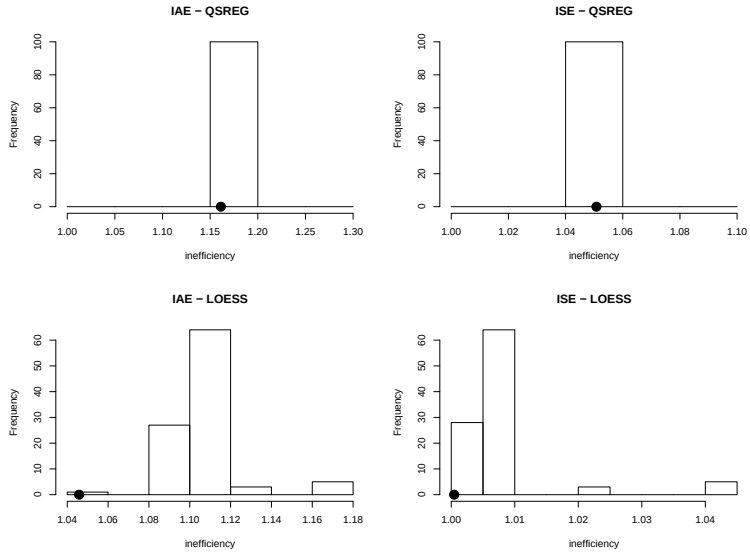

Figure 12: A histogram of inefficiencies obtained from the random 50-fold ACV, to be compared with systematic 50-fold value (dot). From Simulation 3. Based on 100 draws.

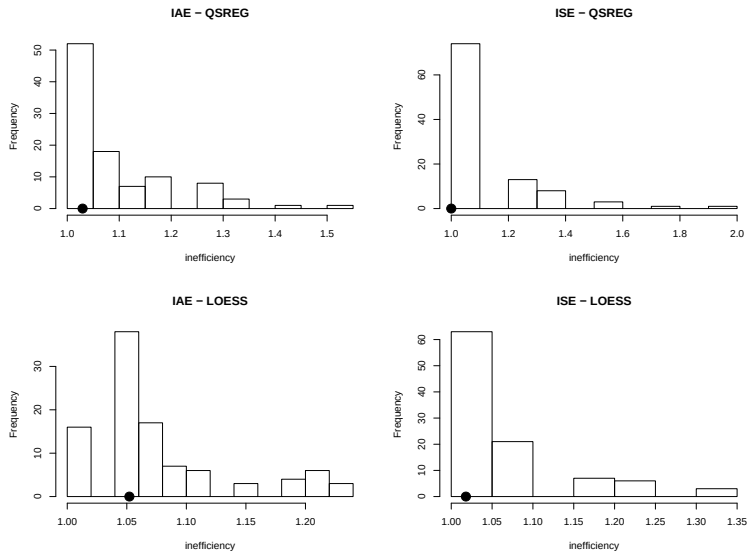

Figure 13: A histogram of inefficiencies obtained from the random 5-fold ACV, to be compared with systematic 5-fold value (dot). From Simulation 4. Based on 100 draws.

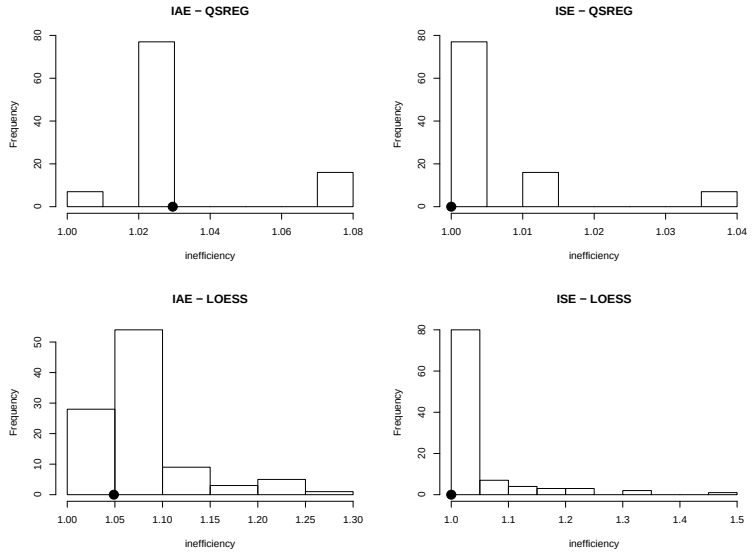

Figure 14: A histogram of inefficiencies obtained from the random 25-fold ACV, to be compared with systematic 25-fold value (dot). From Simulation 4. Based on 100 draws.

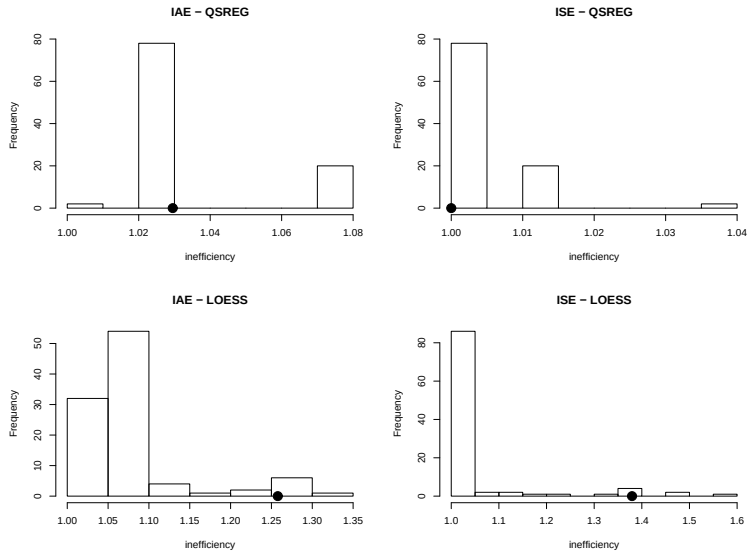

Figure 15: A histogram of inefficiencies obtained from the random 50-fold ACV, to be compared with systematic 50-fold value (dot). From Simulation 4. Based on 100 draws.

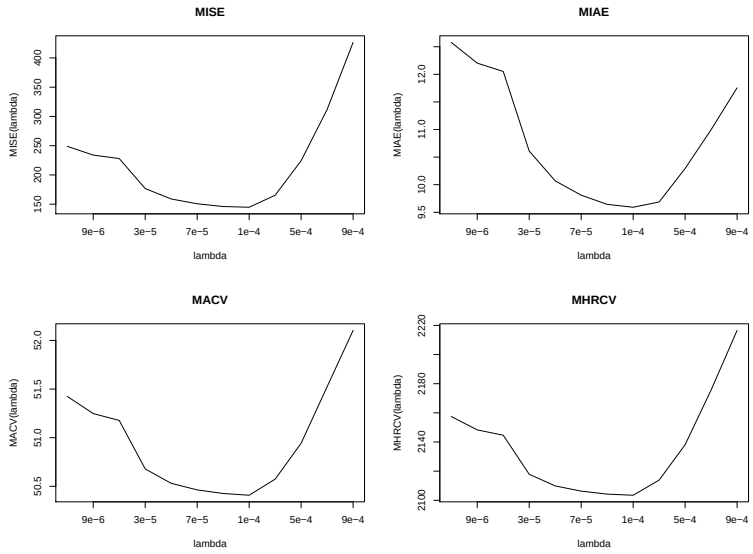

Figure 16: The means of ISE, IAE, ACV, and $HRCV$ for robust smoothing splines.

A.5 Large Simulation Study

The data is obtained just as in Section 4.1.1 of the main text, where we take a vector of "true" curve $m(x)$ and add a random error from the distribution specified in that section to each point of the vector, and repeat this M times with the same same $m(x)$.

If we obtain $ISE(\lambda)$ and $IAE(\lambda)$ functions for each simulation, we can easily estimate mean integrated squared error ($MISE = E[ISE(\lambda)]$) by averaging across the simulations (average over M), and $MIAE$ may likewise be obtained. We also obtain $ACV(\lambda)$ and $HRCV(\lambda)$ curves for each simulation, and we average across the simulations to get the mean curves $MACV(\lambda)$ and $MHRCV(\lambda)$. Our results are based on $M = 100$ simulated data sets.

We are interested in determining $\lambda^*_{ISE} = \text{argmin}_\lambda ISE(\lambda)$ and $\lambda^*_{IAE} = \text{argmin}_\lambda IAE(\lambda)$, and we are also interested in comparing the two theoretical curves $ISE(\lambda)$ and $IAE(\lambda)$ with $ACV(\lambda)$ and $HRCV(\lambda)$. For the comparison plots, we selected a range of λ's that included the minimizers and and roughly an order of magnitude on each side of the minimizer.

Figure 16 shows plots of the means of these four curves for robust smoothing splines, and Figure 17 shows the results for robust LOESS. These plots suggest that both robust cross validation functions do a better job of tracking $MISE$ than $MIAE$. We were somewhat surprised by this as we expected ACV would be more consistent with $MIAE$.

We see in Figure 16 that the minimizing λ are virtually the same in all four functions, and the shapes of the four functions are very similar. In Figure 17,

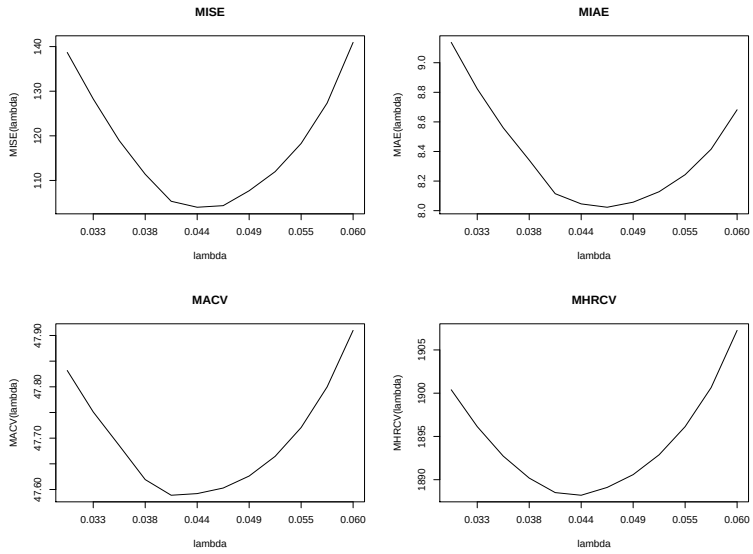

Figure 17: The means of ISE, IAE, ACV, and $HRCV$ for robust LOESS.

Table 8: A table comparing robust smoothers and loss functions. The values in rows $E[ISE(\hat{\lambda})]$ are square rooted to be on the same unit as the $E[IAE(\hat{\lambda})]$.

		`qsreg`	`loess`
ACV	$\sqrt{E[ISE(\hat{\lambda})]}$	12.61	10.45
	$E[IAE(\hat{\lambda})]$	9.89	8.21
$HRCV$	$\sqrt{E[ISE(\hat{\lambda})]}$	12.39	10.30
	$E[IAE(\hat{\lambda})]$	9.77	8.13

the minimizing λ are slightly different, although they are close to each other. However, looking at the ordinate values, we see that minimum values of both theoretical curves ($MISE$ and $MIAE$) in Figure 17 are smaller than in the corresponding plots in Figure 16. This leads us to suspect that the robust LOESS is better suited for our problem.

We want to assess the performance of the two robust smoothers of interest by comparing $E[ISE(\hat{\lambda})]$ values, with $\hat{\lambda}$ a robust cross validation estimate, and similarly for $E[IAE(\hat{\lambda})]$. The result are presented in Table 8. Clearly, all the integrated error measures of robust LOESS are lower than those of the robust smoothing splines. In addition, we see that the values for $HRCV$ are uniformly slightly better than those for ACV.

Next, we present the results of the inefficiency measures in Table 9. Again, this gives evidence that $HRCV$ is slightly better than ACV, as the mean and median inefficiency measures are smaller in all cases. Interestingly, the robust

Table 9: The mean and median inefficiency measure values. The ISE values are square rooted.

		ISE		IAE	
		`qsreg`	`loess`	`qsreg`	`loess`
ACV	**mean**	**1.107**	**1.067**	**1.047**	**1.029**
	min	1.000	1.000	1.000	1.000
	1Q	1.003	1.007	1.006	1.005
	median	1.048	1.026	1.029	1.012
	3Q	1.156	1.074	1.059	1.036
	max	2.066	2.504	1.406	1.200
$HRCV$	**mean**	**1.070**	**1.033**	**1.034**	**1.019**
	min	1.000	1.000	1.000	1.000
	1Q	1.000	1.000	1.006	1.001
	median	1.025	1.011	1.020	1.009
	3Q	1.108	1.031	1.048	1.023
	max	1.502	1.200	1.224	1.145

LOESS has in most cases smaller inefficiencies, indicating that one can do a better job of estimating the optimal smoothing parameter for robust LOESS than for robust smoothing splines (although these results by themselves do not indicate which smoothing method is more accurate).

FInally, we look at all 4 simulation results by means of inefficiencies. These are presented in Tables 10-13.

A.6 Diagnostics

Here, we discuss the diagnostics of fitting the real data with a robust smoother. We picked the same excitation wavelength (310 nm) that we have been using throughout.

First, we did the usual checks on residuals, such as plotting residuals versus fitted values (residual plot) and Quantile-Quantile plot (Q-Q plot). Since our data contain outliers, some of the residuals are very large, which needs to be taken into account.

For the residual plot, we used the original residuals with the limits on the y-axis chosen so that very large residuals are not shown. We only lose 28 observations out of 1550 by this limitation on the y values. Looking at Figure 18 (a), we see no discernible patterns in the plot of residuals versus fitted values.

We have also produced a Q-Q plot, but with the trimmed residuals obtained as follows. All the residuals that are smaller than the 2.5th percentile are set equal to the 2.5th percentile, and the residuals larger than the 97.5th percentile are set equal to the value at the 97.5th percentile. If we glance at Figure 18 (b), most points fall near the line, except the upper half of positive sample quantiles. However, this is not a big cause for concern, as we are not trying to test for the normality of residuals.

Table 10: A table of median values of inefficiencies in Simulation 1. The ISE values are square rooted.

		ACV		HRCV	
		IAE	$\sqrt{\text{ISE}}$	IAE	$\sqrt{\text{ISE}}$
qsreg	Default	2.59	2.68	2.59	2.68
	Full LOO	1.03	1.05	1.02	1.03
	$d = 50, r = 1$	1.03	1.05	1.02	1.02
	$d = 25, r = 1$	1.03	1.04	1.02	1.02
	$d = 5, r = 1$	1.02	1.02	1.01	1.01
loess	Default	28.67	47.29	28.67	47.29
	Full LOO	1.01	1.03	1.01	1.01
	$d = 50, r = 1$	1.01	1.03	1.01	1.02
	$d = 25, r = 1$	1.01	1.03	1.01	1.03
	$d = 5, r = 1$	1.01	1.03	1.01	1.02

Table 11: A table of median values of inefficiencies in Simulation 2. The ISE values are square rooted.

		ACV		HRCV	
		IAE	$\sqrt{\text{ISE}}$	IAE	$\sqrt{\text{ISE}}$
qsreg	Default	2.34	2.41	2.34	2.41
	Full LOO	1.02	1.03	1.01	1.02
	$d = 50, r = 1$	1.02	1.02	1.01	1.02
	$d = 25, r = 1$	1.02	1.03	1.01	1.02
	$d = 5, r = 1$	1.00	1.00	1.00	1.00
loess	Default	32.95	43.48	32.95	43.48
	Full LOO	1.02	1.01	1.01	1.01
	$d = 50, r = 1$	1.01	1.01	1.00	1.01
	$d = 25, r = 1$	1.01	1.01	1.01	1.01
	$d = 5, r = 1$	1.00	1.00	1.00	1.00

Table 12: A table of median values of inefficiencies in Simulation 3. The ISE values are square rooted.

		ACV		HRCV	
		IAE	$\sqrt{\text{ISE}}$	IAE	$\sqrt{\text{ISE}}$
qsreg	Default	2.36	2.17	2.36	2.17
	Full LOO	1.09	1.06	1.10	1.04
	$d = 50, r = 1$	1.08	1.05	1.09	1.04
	$d = 25, r = 1$	1.08	1.05	1.09	1.04
	$d = 5, r = 1$	1.05	1.06	1.07	1.03
loess	Default	46.88	73.53	46.88	73.53
	Full LOO	1.04	1.02	1.04	1.01
	$d = 50, r = 1$	1.04	1.01	1.05	1.00
	$d = 25, r = 1$	1.03	1.02	1.03	1.02
	$d = 5, r = 1$	1.02	1.04	1.03	1.03

Table 13: A table of median values of inefficiencies in Simulation 4. The ISE values are square rooted.

		ACV		HRCV	
		IAE	$\sqrt{\text{ISE}}$	IAE	$\sqrt{\text{ISE}}$
qsreg	Default	2.67	2.54	2.67	2.54
	Full LOO	1.05	1.04	1.04	1.02
	$d = 50, r = 1$	1.04	1.06	1.04	1.02
	$d = 25, r = 1$	1.04	1.04	1.04	1.02
	$d = 5, r = 1$	1.02	1.02	1.03	1.01
loess	Default	70.73	77.85	70.73	77.85
	Full LOO	1.05	1.01	1.05	1.00
	$d = 50, r = 1$	1.03	1.02	1.04	1.01
	$d = 25, r = 1$	1.03	1.02	1.03	1.01
	$d = 5, r = 1$	1.03	1.03	1.03	1.02

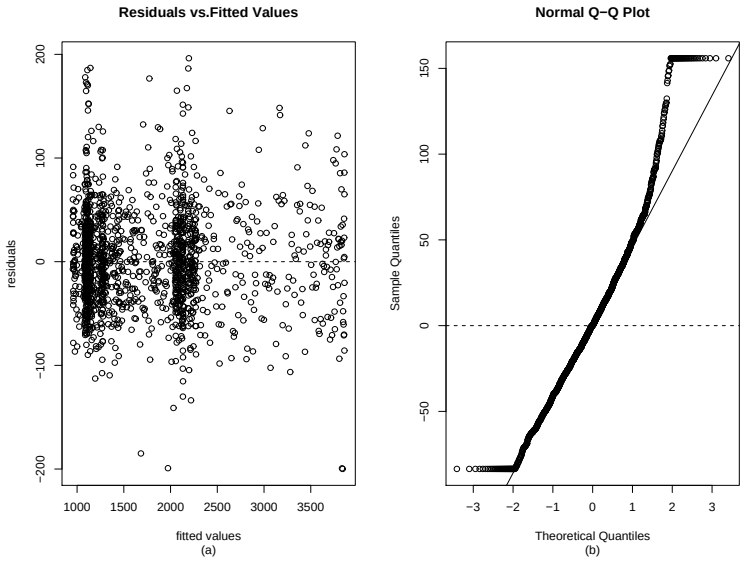

Figure 18: A plot of residuals vs. fitted values and the Q-Q plot.

In addition, we want to look at the autocorrelation of the residuals to determine whether there is much correlation between adjacent emission wavelengths (grid points x_i). When we computed the autocorrelations of the trimmed residuals as described above for the Q-Q plot, we found that there are only small autocorrelations, suggesting that the assumption of independent errors is valid. See Figure 19.

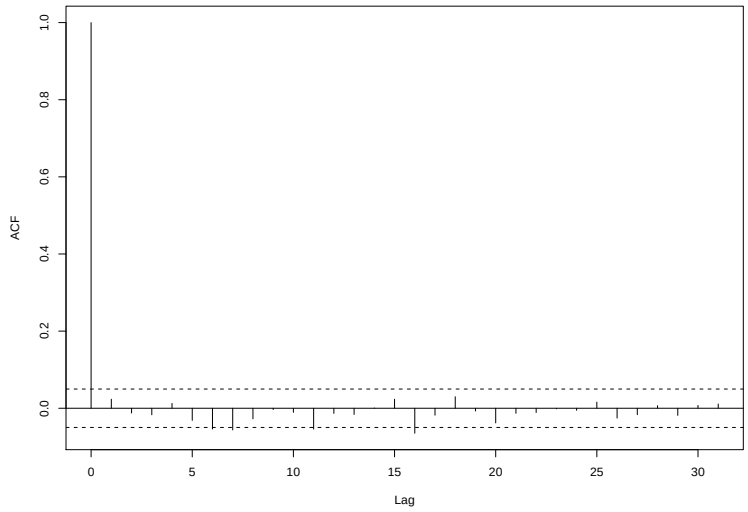

Figure 19: The autocorrelation plots for the trimmed residuals.