

Chapter 5

Resampling methods: Bias, Variance, and their trade-off

We have defined various smoothers and nonparametric estimation techniques. In classical statistical theory we usually assume that the underlying model generating the data is in the family of models we are considering. For nonparametrics this assumption is relaxed and asymptotic and finite sample bias and variance estimates are not always easy to find in closed form. In this Chapter we discuss some resampling methods that are commonly used to get approximations of bias, variance, confidence intervals, etc...

In particular we will look at the problem of choosing smoothing parameters. Remember how most of the smoothers we have defined have some parameter that controls the smoothness of the final smooth or curve estimate. For kernel smoothers we defined the scale parameter, for local regression we defined the span or bandwidth, and for smoothing splines we had the penalty term. We will call all of these *the smoothing parameter* and denote it with λ . It should be clear from the context which of the specific smoothing parameters we are referring to.

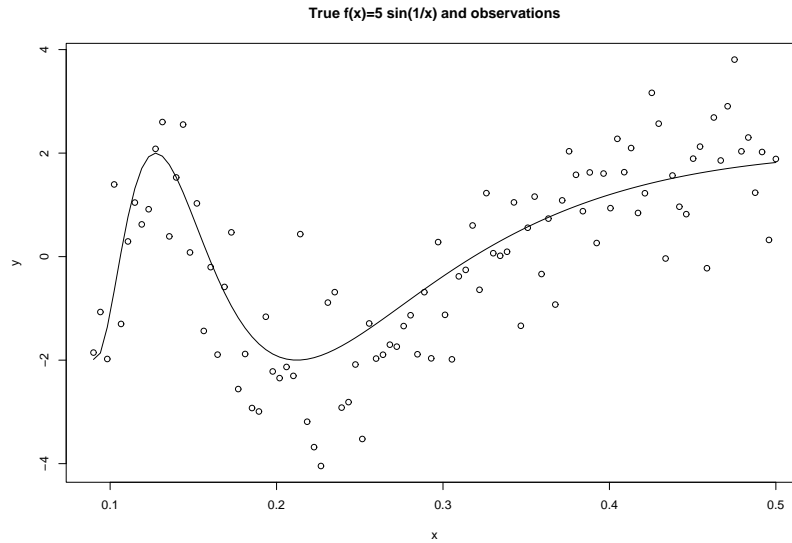


Figure 5.1: Outcomes of model with $f(x) = 5 \sin(1/x)$ and IID normal errors with $\sigma^2 = 1$

5.1 The bias-variance trade-off

In smoothing in general there is a fundamental trade-off between the bias and variance of the estimate, and this trade-off is governed by the smoothing parameter.

Through out this section we will be using an artificial example defined by

$$y_i = 5 \sin(1/x) + \epsilon_i, i = 1, \dots, n \quad (5.1)$$

with the ϵ_i IID $N(0, 1)$ or t_3 .

The trade-off is most easily seen in the case of the running mean smoother. The fitted running-mean smooth can be written as

$$\hat{f}_k(x_0) = \frac{1}{2k+1} \sum_{i \in N_k^S(x_0)} y_i$$

Under model (2.1). The variance is easy to compute. What is it?

The bias is

$$E[\hat{f}_k(x_0)] - f(x_0) = \frac{1}{2k+1} \sum_{i \in N_k^S(x_0)} \{f(x_i) - f(x_0)\}$$

Notice that as k , in this case the smoothing parameter, grows the variances decreases. However, the bigger the k the more $f(x_i)$'s get into the bias.

We have no idea of what $\sum_{i \in N_k^S(x_0)} f(x_i)$ is because we don't know f ! Let's see this in a more precise (not much more) way.

Say we think that f is smooth enough for us to assume that its second derivative $f''(x_0)$ is bounded. Taylor's theorem says we can write

$$f(x_i) = f(x_0) + f'(x_0)(x_i - x_0) + \frac{1}{2}f''(x_0)(x_i - x_0)^2 + o(|x_i - x_0|^2).$$

Because $\frac{1}{2}f''(x_0)(x_i - x_0)^2$ is $O(|x_i - x_0|^2)$ we stop being precise and write

$$f(x_i) \approx f(x_0) + f'(x_0)(x_i - x_0) + \frac{1}{2}f''(x_0)(x_i - x_0)^2.$$

Implicit here is the assumption that $|x_i - x_0|$ is small. This is the way these asymptotics work. We assume that the kernel size goes to 0 as n gets big.

Why did we only go up to the second derivative?

To makes things simple, let's assume that the covariates x are *equally spaced* and let $\Delta = x_{j+1} - x_j$ we can write

$$(2k+1)^{-1} \sum_{i \in N_k^S(x_0)} f(x_i) \approx f(x_0) + (2k+1)^{-1} \frac{k(k+1)}{6} f''(x_0) \Delta^2$$

So now we see that the bias increases with k^2 and the second derivative of the "true" function f . This agrees with our intuition.

Now that we have

$$E\{\hat{f}_k(x_0) - f(x_0)\}^2 \approx \frac{\sigma^2}{2k+1} + \frac{k(k+1)}{6} f''(x_0) \Delta^2$$

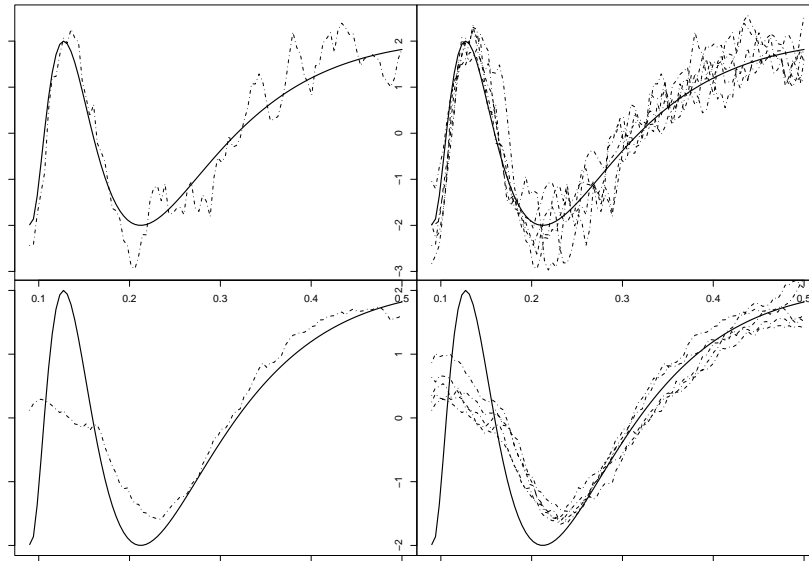


Figure 5.2: Smooths using running-mean smoother with bandwidths of .01 and 0.1. To the right are the smooths 25 replicates

we can actually find an optimal k

$$k_{opt} = \left\{ \frac{9\sigma^2}{2\Delta^4 \{f''(x_i)\}^2} \right\}$$

Usually this is not useful in practice because we have no idea of what $f''(x)$ is like. So how do we choose smoothing parameters?

In Figure 5.2 we show the smooths obtained with a running mean smoother with bandwidths of 0.01 and 0.1 on 25 replicates defined by (5.1). The bias-variance trade-off can be clearly seen.

5.1.1 Bias-variance trade-off for linear smoothers

Define S_λ as the hat matrix for a particular smoother when the smoothing parameter λ is used. The “smooth” will be written as $\hat{\mathbf{f}}_\lambda = S_\lambda \mathbf{y}$.

Define

$$\mathbf{v}_\lambda = \mathbf{f} - \mathbf{E}(\mathbf{S}_\lambda \mathbf{y})$$

as the *bias* vector.

Define $\text{ave}(\mathbf{x}^2) = n^{-1} \sum_{i=1}^n x_i^2$ for any vector \mathbf{x} . We can derive the following formulas:

$$\begin{aligned} \text{MSE}(\lambda) &= n^{-1} \sum_{i=1}^n \text{var}\{\hat{f}_\lambda(x_i)\} + \text{ave}(\mathbf{v}_\lambda^2) \\ &= n^{-1} \text{tr}(\mathbf{S}_\lambda \mathbf{S}'_\lambda) \sigma^2 + n^{-1} \mathbf{v}'_\lambda \mathbf{v}_\lambda \\ \text{PSE}(\lambda) &= \{1 + n^{-1} \text{tr}(\mathbf{S}_\lambda \mathbf{S}'_\lambda)\} \sigma^2 + n^{-1} \mathbf{v}'_\lambda \mathbf{v}_\lambda. \end{aligned}$$

Notice for least-squares regression \mathbf{S}_λ is idempotent so that $\text{tr}(\mathbf{S}_\lambda \mathbf{S}'_\lambda) = \text{tr}(\mathbf{S}_\lambda) = \text{rank}(\mathbf{S}_\lambda)$ which is usually the number of parameters in the model. This is why we will sometimes refer to $\text{tr}(\mathbf{S}_\lambda \mathbf{S}'_\lambda)$ as the *equivalent number of parameters* or degrees of freedom of our smoother.

5.2 Cross Validation: Choosing smoothness parameters

In the section, and the rest of the class, we will denote with \hat{f}_λ the estimate obtained using smoothing parameter λ . Notice that usually what we really have is the smooth $\hat{\mathbf{f}}_\lambda$.

We will use the model defined by (5.1). Figure 5.3 shows one outcome of this model with normal and t-distributed errors.

We are trying to find the λ that minimizes

$$\text{MSE}(\lambda) = n^{-1} \sum_{i=1}^n \mathbf{E}[\hat{f}_\lambda(x_i) - f(x_i)]^2$$

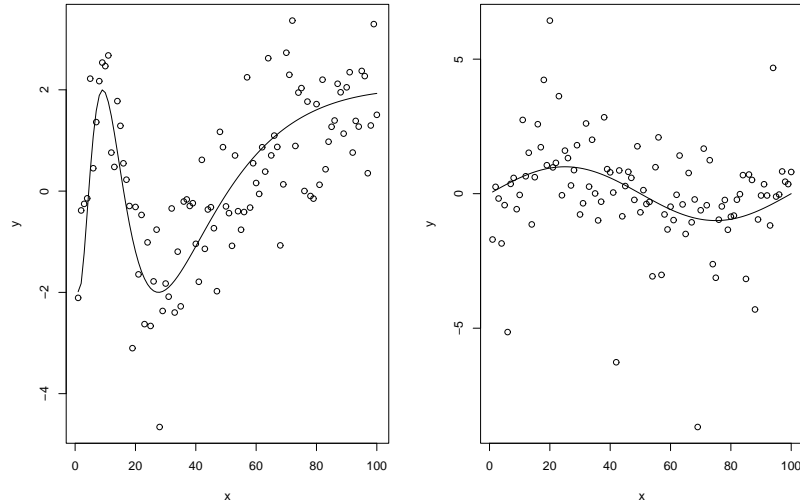


Figure 5.3: Outcomes of model (5.1)

Problem is we don't now f .

What if we could get a new set of data y_1^*, \dots, y_n^* from the same model producing the y_1, \dots, y_n ? This would be quite helpful because the *predictive squared error*

$$\text{PSE}(\lambda) = E[y_i^* - \hat{f}_\lambda(x_i)]^2 = E[\{y_i^* - f(x_i)\} - \{\hat{f}_\lambda(x_i) - f(x_i)\}] = \text{MSE}(\lambda) + \sigma^2.$$

says that $n^{-1} \sum_{i=1}^n [y_i^* - \hat{f}_\lambda(x_i)]^2$ is an average having expected value the MSE plus a constant. We could view this quantity as an estimate of $\text{MSE}(\lambda) + \sigma^2$. Since σ^2 doesn't depend on λ we could find the λ that minimizes it and think that we are close to the λ that minimizes the MSE.

Notice that the above calculation can be done because the y_i^* s are independent of the estimates $\hat{f}_\lambda(x_i)$ s, the same can't be said about the y_i s.

In practice it is not common to have a new set of data $y_i^*, i = 1, \dots, n$. Cross-validation tries to imitate this by leaving out points (x_i, y_i) one at a time and estimating the smooth at x_i based on the remaining $n - 1$ points. The cross-

validation sum of squares is

$$\text{CV}(\lambda) = n^{-1} \sum_{i=1}^n \{y_i - \hat{f}_\lambda^{-i}(x_i)\}^2$$

where $\hat{f}_\lambda^{-i}(x_i)$ indicates the fit at x_i computed by leaving out the i -th point.

We can now use CV to choose λ by considering a wide span of values of λ , computing $\text{CV}(\lambda)$ for each one, and choosing the λ that minimizes it. Plots of $\text{CV}(\lambda)$ vs. λ may be useful.

Why do we think this is good? First notice that

$$\begin{aligned} \text{E}\{y_i - \hat{f}_\lambda^{-i}(x_i)\}^2 &= \text{E}\{y_i - f(x_i) + f(x_i) - \hat{f}_\lambda^{-i}(x_i)\}^2 \\ &= \sigma^2 + \text{E}\{\hat{f}_\lambda^{-i}(x_i) - f(x_i)\}^2. \end{aligned}$$

Using the assumption that $\hat{f}_\lambda^{-i}(x_i) \approx \hat{f}_\lambda(x_i)$ we see that

$$\text{E}\{\text{CV}(\lambda)\} \approx \text{PSE}(\lambda)$$

However, what we really want is

$$\min_{\lambda} \text{E}\{\text{CV}(\lambda)\} \approx \min_{\lambda} \text{PSE}(\lambda)$$

but the law of large numbers says the above will do.

Why not simply use the averaged squared residuals

$$\text{ASR}(\lambda) = n^{-1} \sum_{i=1}^n \{y_i - \hat{f}_\lambda(x_i)\}^2?$$

It turns out this under-estimates the PSE. Notice in particular that the estimate $\hat{f}(x_i) = y_i$ always has ASR equal to 0! We will see how we can adjust the ASR to form “good” estimates of the MSE.

5.2.1 CV for linear smoothers

Now we will see some of the practical advantages of linear smoothers.

For linear smoothers in general it is not obvious what is meant by $\hat{f}_\lambda^{-i}(x_i)$. Let's give a definition...

Notice that any reasonable smoother will smooth constants into constants, i.e. $\mathbf{S}\mathbf{1} = \mathbf{1}$. If we think of the rows \mathbf{S}_i of \mathbf{S} as weights of a kernels, this condition is requiring that all the n weights in each of the n kernels add up to 1. We can define $\hat{f}_\lambda^{-i}(x_i)$ as the “weighted average”

$$\mathbf{S}_i \cdot \mathbf{y} = \sum_{j=1}^n S_{ij} y_j$$

but giving zero weight to the i th entry, i.e.

$$\hat{f}_\lambda^{-i}(x_i) = \frac{1}{1 - S_{ii}} \sum_{j \neq i} S_{ij} y_j.$$

From this definition we can find CV without actually making all the computations again. Lets see how:

Notice that

$$\hat{f}_\lambda^{-i}(x_i) = \sum_{j \neq i} S_{ij} y_j + S_{ii} \hat{f}_\lambda^{-i}(x_i).$$

The quantities we add up to obtain CV are the squares of

$$y_i - \hat{f}_\lambda^{-i}(x_i) = y_i - \sum_{j \neq i} S_{ij} y_j - S_{ii} \hat{f}_\lambda^{-i}(x_i).$$

Adding and subtracting $S_{ii} y_i$ we get

$$y_i - \hat{f}_\lambda^{-i}(x_i) = y_i - \hat{f}_\lambda(x_i) + S_{ii}(y_i - \hat{f}_\lambda^{-i}(x_i))$$

which implies

$$y_i - \hat{f}_\lambda^{-i}(x_i) = \frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_{ii}}$$

and we can write

$$\text{CV}(\lambda) = n^{-1} \sum_{i=1}^n \left\{ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_{ii}} \right\}^2$$

so we don't have to compute $\hat{f}_\lambda^{-i}(x_i)$!

Lets see how this definition of CV may be useful in finding the MSE.

Notice that the above defined CV is similar to the ASR except for the division by $1 - S_{ii}$. To see what this is doing we notice that in many situations $S_{ii} \approx [\mathbf{S}_\lambda \mathbf{S}_\lambda]_{ii}$ and $1/(1 - S_{ii})^2 \approx 1 + 2S_{ii}$ which implies

$$E[\text{CV}(\lambda)] \approx \text{PSE}(\lambda) + 2\text{ave}[\text{diag}(\mathbf{S}_\lambda) \mathbf{v}^2].$$

Thus CV adjusts ASR so that in expectation the variance term is correct but in doing so induces an error of $2S_{ii}$ into each of the bias components.

In Figure 5.4 we see the CV and MSE for $n = 100$ and $n = 500$ observatios

5.3 Bootstrap Standard Errors and Confidence Sets

Statistical science is the science of learning from experience. Efron and Tibshirani (1993) say “Most people are not natural-born statisticians. Left to our own devices we are not very good at picking out patterns from a sea of noisy data. To put it another way, we are all too good at picking out non existing patterns that happen to suit our purposes.”

Suppose we find ourselves in the following common data-analytic situation: a random sample $\mathbf{x} = (x_1, \dots, x_n)$ from an unknown probability distribution F has been observed and we wish to estimate a parameter of interest $\theta = t(F)$ on the basis of \mathbf{x} . For this purpose, we calculate an estimate $\hat{\theta} = s(\mathbf{x})$ from \mathbf{x} .

A common estimate is the *plug-in* estimate $t(\hat{F})$ where \hat{F} is the empirical distribution defined by

$$F(x) = \frac{\text{number of values in } \mathbf{x} \text{ equal to } x}{n}$$

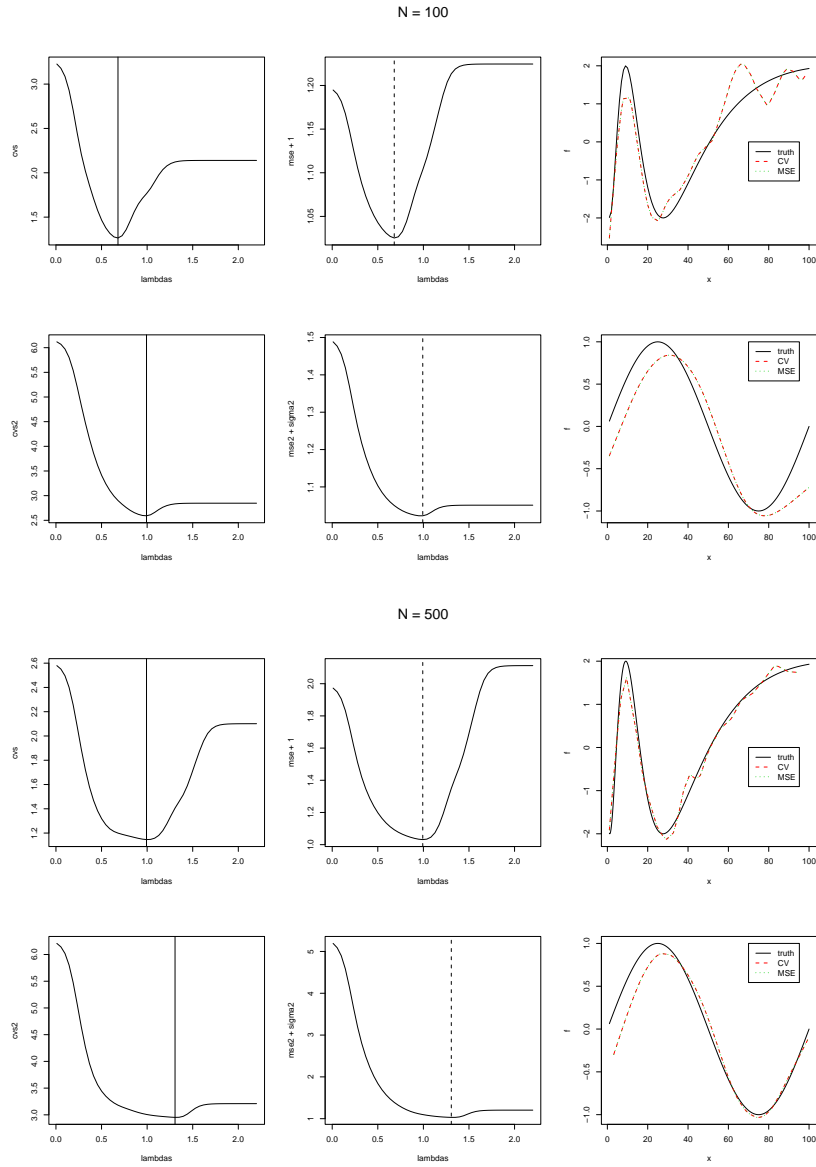


Figure 5.4: CV, MSE, and fits obtained for the normal and t models.

Can you think of a plug-in estimate that is commonly used?

The bootstrap was introduced by Efron (1979) as a computer based method to estimate the standard deviation of $\hat{\theta}$.

What are the advantages:

- It is completely automatic
- Requires no theoretical calculations
- Not based on asymptotic results
- Available no matter how complicated the estimator $\hat{\theta}$ is.

A bootstrap sample is defined to be a random sample of size n drawn from \hat{F} , say $\mathbf{x}^* = (x_1^*, \dots, x_n^*)$.

For each bootstrap sample \mathbf{x}^* there is a bootstrap replicate of $\hat{\theta}$,

$$\hat{\theta}^* = s(\mathbf{x}^*).$$

The bootstrap estimate of $\text{se}_F(\hat{\theta})$ is defined by

$$\text{se}_{\hat{F}}(\hat{\theta}^*). \quad (5.2)$$

This is called the *ideal bootstrap estimate* of the standard error of $s(\mathbf{x})$.

Notice that for the case where θ is the expected value or mean of x_1 we have

$$\text{se}_{\hat{F}}(\bar{x}^*) = \text{se}_{\hat{F}}(x_1^*)/\sqrt{n} = \sqrt{n^{-1} \sum_{i=1}^n (x_i - \hat{x})^2}/\sqrt{n}$$

and the ideal bootstrap estimate is the estimate we are used to. However, for any other estimator other than the mean obtaining (5.2) there is no neat formula that enables us to compute a numerical value in practice.

The bootstrap algorithm is a computational way of obtaining a good approximation to the numerical value of (5.2).

5.3.1 The bootstrap algorithm

The bootstrap algorithm works by drawing many independent bootstrap samples, evaluating the corresponding bootstrap replications, and estimating the standard error of $\hat{\theta}$ by the empirical standard error, denoted by $\hat{\text{se}}_B$, where B is the number of bootstrap samples used.

1. Select B independent bootstrap samples $\mathbf{x}_1^*, \dots, \mathbf{x}_B^*$, each consisting of n data values drawing with replacement from \mathbf{x} .
2. Evaluate the bootstrap replication corresponding to each bootstrap sample

$$\hat{\theta}^*(b) = s(\mathbf{x}_b^*), b = 1, \dots, B$$

3. Estimate the standard error $\text{se}_F(\hat{\theta})$ by the sample standard error of the B replicates

$$\hat{\text{se}}_B = \left[\frac{1}{B-1} \sum_{b=1}^B \{\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)\}^2 \right]$$

with

$$\hat{\theta}^*(\cdot) = B^{-1} \sum_{b=1}^B \hat{\theta}^*(b)$$

The limit of $\hat{\text{se}}_B$ as B goes to infinity is the ideal bootstrap estimate of (5.2). But how close is (5.2) to $\text{se}_F(\hat{\theta})$? See Efron and Tibshirani (1993) for more details.

5.3.2 Example: Curve fitting

In this example we will be estimating regression functions in two ways, by a standard least-squares line and by loess.

A total of 164 men took part in an experiment to see if the drug cholestyramine lowered blood cholesterol levels. The men were supposed to take six packets of

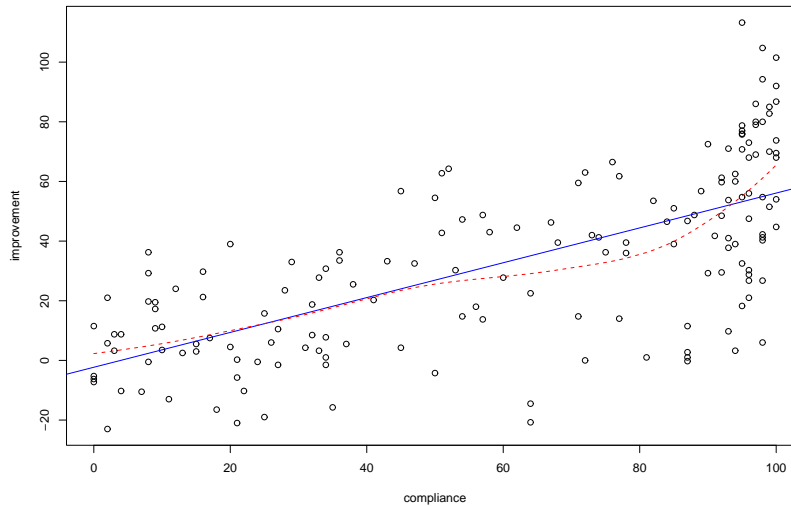


Figure 5.5: Estimated regression curves of Improvement on Compliance.

cholestyramine per day, but many of them actually took much less. Figure 5.5 shows compliance plotted against percentage of the intended dose actually taken. We also show a fitted line and a loess fit (using $\text{span}=2/3$). Notice the curves similar from 0 to 60, a little different from 60 to 80 and quite different from 80 to 100.

Assume the points a regression model

$$y_i = f(x_i) + \varepsilon_i, i = 1, \dots, n$$

with the ε_i IID.

Say we are interested in the difference in rate of change of $f(x)$ in the 60–80 and 80–100 sections. We could define as the parameter to describe this. How can we do this?

Notice that finding a standard error for this estimate is not straight-forward. We can use the bootstrap.

	$\hat{f}_{\text{line}}(60)$	$\hat{f}_{\text{line}}(80)$	$\hat{f}_{\text{line}}(100)$	$\hat{f}_{\text{loess}}(60)$	$\hat{f}_{\text{loess}}(80)$	$\hat{f}_{\text{loess}}(100)$
value:	33	44	56	28	35	66
\hat{se}_{50} :	2	2	3	5	4	4

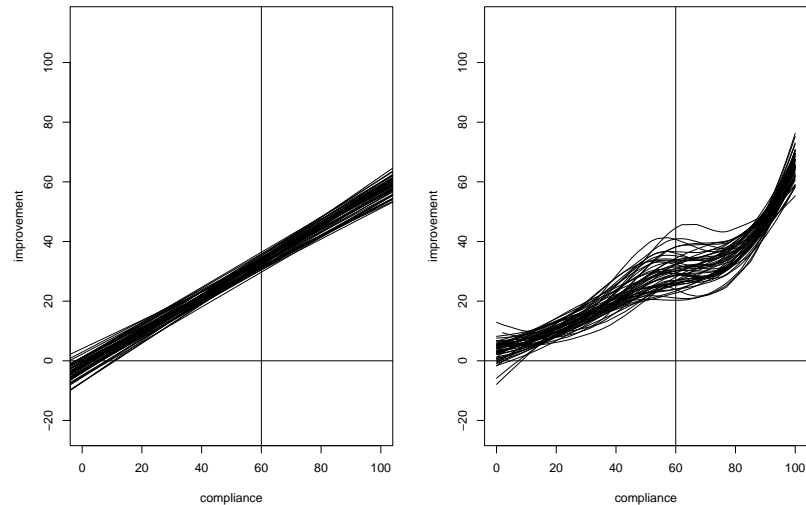


Figure 5.6: 50 bootstrap curves for each estimation technique.

Even when there is no parameter of interest, the bootstrap estimates of f give us an idea of what a confidence set is for the nonparametric estimates. We will see more of this in Chapter 7 and 8.

5.3.3 Confidence “intervals” for linear smoothers

It is easy to show that the variance-covariance matrix of the vector of fitted values $\hat{\mathbf{f}} = \mathbf{S}\mathbf{y}$ is

$$\text{cov}(\hat{\mathbf{f}}) = \mathbf{S}\mathbf{S}'\sigma^2$$

and given an estimate of σ^2 this can be used to give point-wise standard errors, mainly by looking at $\text{diag}(\mathbf{S}\mathbf{S}')\sigma^2$.

Can we construct confidence intervals? What do we need?

First of all we need to know the distribution (at least approximately) of $\hat{\mathbf{f}}$. If the errors are normal we know that \mathbf{f} is normally distributed. Why?

In the normal case, what are the confidence intervals for?

Remember that our estimates are usually biased, $E(\hat{\mathbf{f}}) = \mathbf{S}\mathbf{f} \neq \mathbf{f}$. If our null hypothesis is $\mathbf{S}\mathbf{f} = \mathbf{f}$ (in the case of splines this is equivalent to assuming $f \in \mathcal{G}$) then our confidence intervals are for \mathbf{f} otherwise they are for $\mathbf{S}\mathbf{f}$. We will start using the notation $\tilde{\mathbf{f}} = \mathbf{S}\mathbf{f}$. We can think of $\tilde{\mathbf{f}}$ as an approximation to “the truth” \mathbf{f} .

To see how point-wise estimates can be useful, notice that we can get an idea of how variable $\hat{\mathbf{f}}(x_0)$ is. However, it isn’t very helpful when we want to see how variable $\hat{\mathbf{f}}$ is as a whole.

However what if we want to know if a certain function, say a line, is in our “confidence interval”? Point-wise intervals don’t really help us with this.

5.3.4 Global confidence bands

Remember that $\hat{\mathbf{f}} \in \mathbb{R}^n$. This means that talking about confidence intervals doesn’t make much sense. We need to consider confidence sets.

For example if the errors are normal we know that

$$\chi(\tilde{\mathbf{f}}) = (\hat{\mathbf{f}} - \tilde{\mathbf{f}})'(\mathbf{S}\mathbf{S}'\sigma^2)^{-1}(\hat{\mathbf{f}} - \tilde{\mathbf{f}})$$

is χ_n^2 distributed. This permits us to construct confidence sets (which you can think of as random n -dimensional balls) for $\tilde{\mathbf{f}}$ of probability α

$$C_\alpha = \{\mathbf{g}; \chi(\mathbf{g}) \leq \chi_{1-\alpha}\} = \{\mathbf{g}; (\hat{\mathbf{f}} - \mathbf{g})'(\mathbf{S}\mathbf{S}'\sigma^2)^{-1}(\hat{\mathbf{f}} - \mathbf{g}) \leq \chi_{1-\alpha}\}.$$

Notice that the probability that the random ball doesn’t fall on the approximate truth $\tilde{\mathbf{f}}$ is α :

$$\Pr(\tilde{\mathbf{f}} \notin C_\alpha) = \Pr\left[(\hat{\mathbf{f}} - \tilde{\mathbf{f}})'(\mathbf{S}\mathbf{S}'\sigma^2)^{-1}(\hat{\mathbf{f}} - \tilde{\mathbf{f}}) > \chi_{1-\alpha}\right] = \alpha.$$

This is only the case if we know σ^2 .

Usually we construct an estimate

$$\hat{\sigma}^2 = (\mathbf{y} - \hat{\mathbf{f}})'(\mathbf{y} - \hat{\mathbf{f}}) / \{n - \text{tr}(2\mathbf{S} - \mathbf{S}\mathbf{S}')\}$$

and define confidence sets

$$C(\tilde{\mathbf{f}}) = \{\mathbf{g}; \nu(\mathbf{g}) \leq G_{1-\alpha}\}$$

based on

$$\nu(\tilde{\mathbf{f}}) = (\hat{\mathbf{f}} - \tilde{\mathbf{f}})'(\mathbf{S}\mathbf{S}'\hat{\sigma}^2)^{-1}(\hat{\mathbf{f}} - \tilde{\mathbf{f}}).$$

Here $G_{1-\alpha}$ is the $(1 - \alpha)$ th quantile of the distribution of $\nu(\tilde{\mathbf{f}})$.

Do we know it? Not necessarily.

In the case of linear regression, where the gaussian model is correct and \mathbf{S} is a p -dimensional projection, $\nu(\tilde{\mathbf{f}}) = \nu(\mathbf{f})$ has distribution $(n - p) + pF_{p, n-p}$.

When this is not the case we can argue that the distribution is approximately

$$\{n - \text{tr}(2\mathbf{S} - \mathbf{S}\mathbf{S}')\} + \text{tr}(\mathbf{S}\mathbf{S}')F_{\text{tr}(\mathbf{S}\mathbf{S}'), n - \text{tr}(2\mathbf{S} - \mathbf{S}\mathbf{S}')}$$

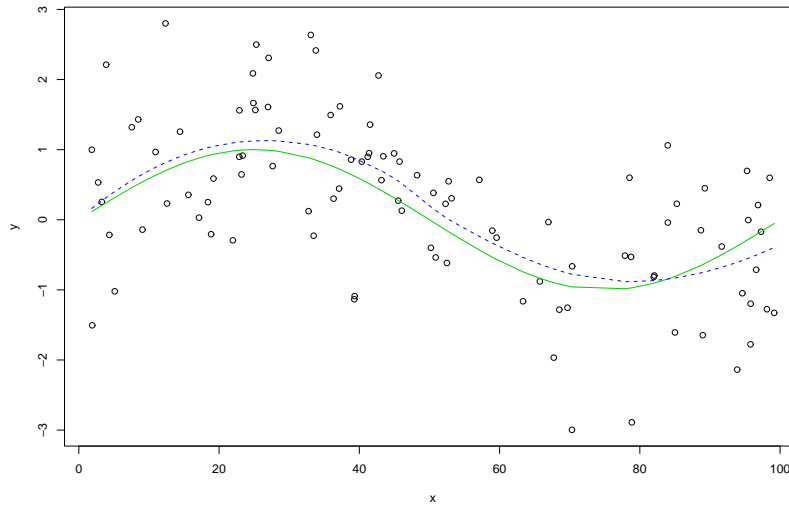
If we are not sure of the normality assumption or that $\tilde{\mathbf{f}} \approx \mathbf{f}$ we can use the bootstrap to construct an approximate distribution \hat{G} of G .

How do we do it?

5.3.5 Bootstrap estimate of $G_{1-\alpha}$

A bootstrap sample is generated in the following way

- For some data \mathbf{y} use some procedure (a linear smoother for example) to obtain an estimate $\hat{\mathbf{f}}$ of some estimand (in this case the regression function \mathbf{f}).

Figure 5.7: The regression curve and an outcome with $n = 100$ and $\sigma^2 = 1$.

- Obtain residuals $\hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \hat{\mathbf{f}}$.
- Take a simple random sample of size n from the residuals $\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n$. Notice that this makes them IID just like the ε s.

- Construct a “new” data set

$$\mathbf{y}^* = \hat{\mathbf{f}} + \hat{\boldsymbol{\varepsilon}}^*$$

with $\hat{\boldsymbol{\varepsilon}}^*$ the vector of resampled residuals.

- From the new data form a new estimate $\hat{\mathbf{f}}^*$.
- Finally we obtain the value of

$$\nu^* = (\hat{\mathbf{f}}^* - \hat{\mathbf{f}})'(\mathbf{S}\mathbf{S}'\hat{\sigma}^{*2})^{-1}(\hat{\mathbf{f}}^* - \hat{\mathbf{f}})$$

- We repeat this procedure many times and form an approximate distribution \hat{G} with the values of ν^* . We may use the $(1 - \alpha)$ th quantile of \hat{G} as an estimate of $G_{1-\alpha}$.

Let's consider the model $y_i = f(x_i) + \varepsilon_i, i = 1, \dots, n$ with ε_i IID normal. In Figure ?? we see qqplots of the true G , the bootstrap G and the F-distribution approximation.

5.3.6 Displaying the confidence sets

Displaying an $n - dimensional$ ball is not easy.

Global confidence bands usually show the projections of the confidence set onto each of the component sub-spaces. Notice that a function (now I'm using function and n -dimensional vector interchangeably) in this set would actually be in a confidence cube as opposed to a ball! So a vector within the confidence bands isn't necessarily in the confidence ball. However its true that being in the ball implies being within the band.

Another popular approach is selecting a few functions at random from $N(\hat{\mathbf{f}}, \mathbf{SS}'\hat{\sigma}^2)$ and checking to see if they are in the confidence set. If they are, we plot them. This enables us to see what kind of "shape" functions in the confidence set have. Maybe they all have a bump, maybe a large amount of them are close to being constant lines, etc....

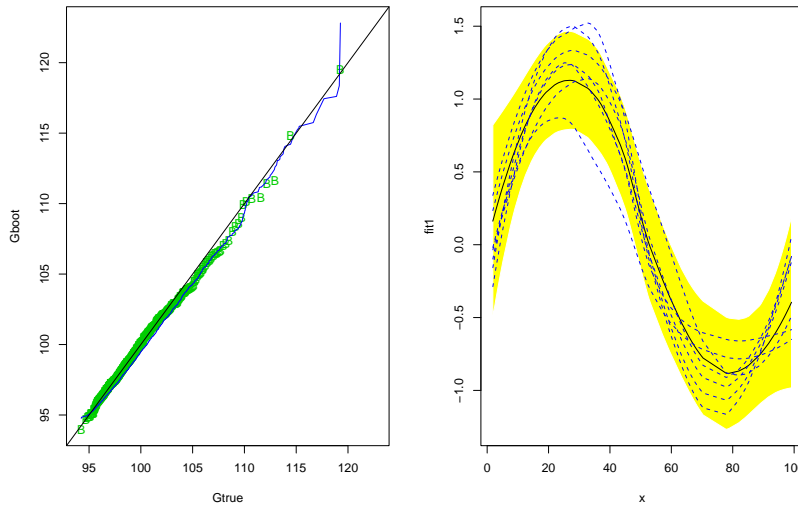
5.3.7 Approximate F-test

Using the F-distribution approximations we may construct F-tests for testing various hypotheses.

The p-value given by the S-Plus function `gam()` is usually testing for linearity and using an F-distribution approximation.

Suppose we wish to compare 2 smoothers $\hat{\mathbf{f}}_1 = \mathbf{S}_1\mathbf{y}$ and $\hat{\mathbf{f}}_2 = \mathbf{S}_2\mathbf{y}$. For example, $\hat{\mathbf{f}}_1$ may be linear regression and $\hat{\mathbf{f}}_2$ may be a "rougher" smoother.

Figure 5.8: QQ-plot of bootstrap vs. true G and the F-distribution approximation. We also see point-wise confidence intervals and curves in (blue) and out (green) of the bootstrap confidence set.



Let RSS_1 and RSS_2 be the residual sum of squares obtained for each smoother. Which one do you expect to be bigger?

and γ_1 and γ_2 be the degrees of freedom of each smoother, $\text{tr}(2\mathbf{S}_j - \mathbf{S}_j\mathbf{S}'_j)$, $j = 1, 2$. An approximation that may be useful for this comparison is

$$\frac{(RSS_1 - RSS_2)/(\gamma_2 - \gamma_1)}{RSS_2/(n - \gamma_2)} \sim F_{\gamma_2 - \gamma_1, n - \gamma_2}$$

There are moment corrections that can make this a better approximation (see H&T).

Figure 5.9: Same as previous figure but with t-distributed errors

