Chapter 8

Total Least Squares and Robust Methods

In discussing least squares estimation we have been rather conservative in the assumptions we made about the errors in our observations, preferring to deal with the standard statistical model rather than introducing additional complications. Last time we did deal with the data covariance problem for the case of normally distributed data errors by supposing that the problem can be transformed into one in which $C_{yy} = \sigma^2 I$. Now we look at a few issues related to violations of the assumptions used so far.

8.1. Total Least Squares and the Bootstrap

We began LSE with the assumption that the variables X are independent, and $\vec{Y}$ is dependent, with the form

$$\vec{Y} = X\vec{\theta} + \vec{e} \tag{1}$$

The X's were supposed fixed and all the randomness or statistical aspects are introduced through $\vec{e}$. But it might be quite unrealistic to assume that X can be pre-ordained in this way. How would we go about this if both X and Y are random?

First we have to replace the fixed design matrix X with a random matrix Ξ. We retain the notation X as a particular realization of this random matrix. Thus we write

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \\ \vdots \\ \vec{x}_n \end{bmatrix} \qquad \Xi = \begin{bmatrix} \vec{\xi}_1 \\ \vec{\xi}_2 \\ \vec{\xi}_3 \\ \vdots \\ \vec{\xi}_n \end{bmatrix} \tag{2}$$

Our original model $y_i = \vec{x}_i\theta + e_i$ with $\vec{x}_i$ fixed and e_i random with zero mean and variance σ^2 now becomes

$$E\left[\vec{Y}|\xi = X\right] = X\vec{\theta} \qquad \text{and} \qquad Var[Y|\xi = X] = E\left[(\vec{Y} - X\vec{\theta})^2|\xi = X\right] = \sigma^2 \tag{3}$$

With both Ξ and Y having random attributes we have a new model of the data as n independent random vectors $(Y_1, \vec{\xi}_1), (Y_2, \vec{\xi}_2), \ldots, (Y_n, \vec{\xi}_n)$ drawn from a multidimensional joint distribution. The previous model is a conditional version of the new model, and our old analysis technique is conditional on the observed values $\vec{x}_1, \vec{x}_2, ..., \vec{x}_n$.

What are the consequences for this new model on determining the statistical properties of our estimates for θ? Under the old conditional model we had that θ was an unbiased estimate, now expressed in the following way

$$E(\hat{\theta}|\Xi = X) = \theta \tag{4}$$

We proceed in the same way for the new model writing another expectation (the outer bracket in the following equation) with respect to the distribution of Ξ

$$\begin{aligned} E(\hat{\theta}) &= E(E(\hat{\theta}|\Xi)) \\ &= E(\theta) \\ &= \theta, \end{aligned} \tag{5}$$

and discover that the least squares estimate is unbiased under the new model too. The variance is less obvious. Recall that for the conditional variance and LS estimate of the last chapter we have

$$Var[\hat{\theta}_i|\Xi = X] = \sigma^2(X^T X)^{-1}_{ii}$$

In the unconditional case we find

$$\begin{aligned} Var[\hat{\theta}_i] &= Var[E(\hat{\theta}_i|\Xi)] + E[Var(\hat{\theta}_i|\Xi)] \\ &= Var[\theta_i] + E[\sigma^2(\Xi^T\Xi)^{-1}_{ii}] \\ &= \sigma^2 E[(\Xi^T\Xi)^{-1}_{ii}] \end{aligned} \tag{6}$$

This is a highly non-linear function of the random vectors $\vec{\xi}_1,\ \vec{\xi}_2, \ldots, \vec{\xi}_n$, and in general quite difficult to evaluate analytically.

The above tells us that the new model is still unbiased, but has different covariances. We can use the bootstrap to evaluate the variability of one of the $\hat{\theta}_j$ under the new model or to evaluate some other property of the model, such as $E[Y|\xi = x_0]$ or the correlation coefficient in linear regression. If we know the probability distribution of the random vector (Y, ξ) we just use a Monte Carlo simulation to draw a large number (say M) from this distribution. For each (Y_m, ξ_m) with $m = 1, \ldots, M$ we calculate an estimate for θ and use this collection as an approximation to the sampling distribution of $\hat{\theta}$. From this we can get an approximate value for the standard error in θ, etc. This is sometimes called a parametric bootstrap technique. Of course if the distribution of (Y, ξ) is unknown we can use the empirical distribution in the same way as described in section 5.2.3.

1:1 Total Least Squares - A Simple Example

In *Numerical Recipes* by Press *et al.* there is a nice discussion (and code supplied) for fitting a straight line by total least squares. This is well worth a read if you have this kind of problem. Here we treat our simplest

LS problem as an illustration of the ideas. Recall that at the beginning of chapter 7 we considered pairs of observations (x_i, y_i) where $i = 1, 2, \dots, n$. Once again we suppose that a reasonable model is of the form

$$y = \beta_0 + \beta_1 x, \tag{7}$$

but now we allow for the fact that there is iid zero mean Gaussian noise of variance σ^2 in both x and y (see Figure 8.1)

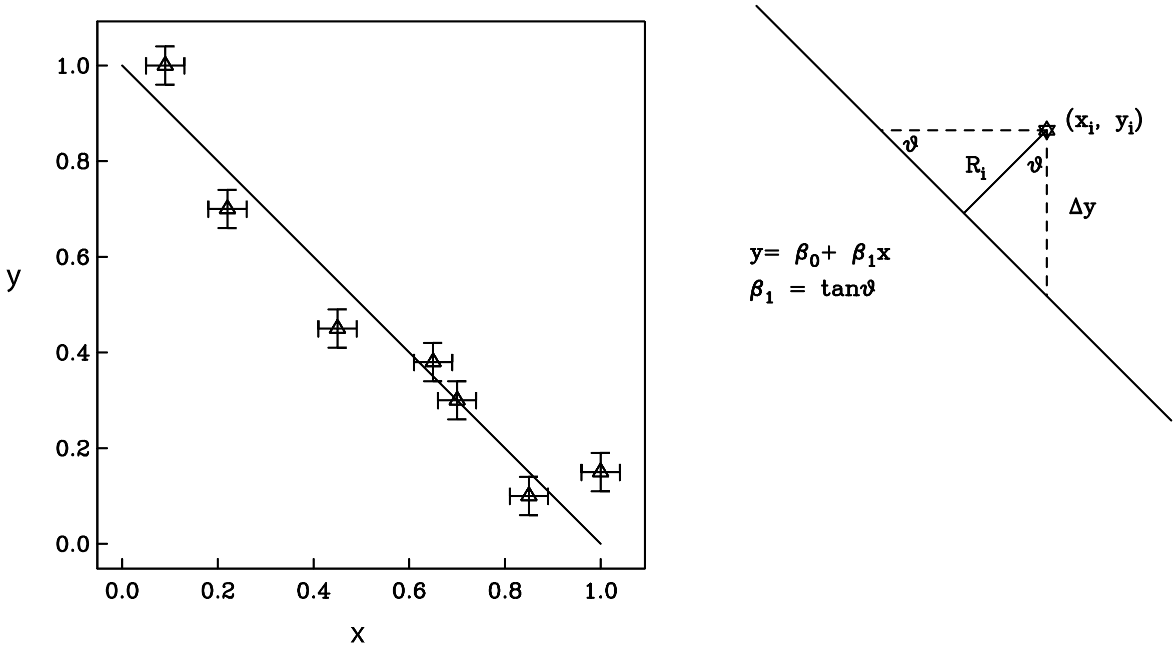

Figure 8.1: The total least squares problem for a straight line. Note that in the illustration the uncertainties in x and y are equal.

In ordinary LS estimation we would find the β_i that minimize the sum of the squares of the vertical distance between the line and the data. The analogous estimator for total least squares (TLS) puts the cloud of measured (x_i, y_i) as close as possible to the line $y = \beta_0 + \beta_1 x$, using a different measure of distance, in this case the perpendicular distance R_i, which we can write as

$$R_i = \Delta y_i \cos\theta = \frac{\Delta y_i}{\sqrt{1 + \tan^2\theta}} = \frac{\Delta y_i}{\sqrt{1 + \beta_1^2}} \tag{8}$$

where we have identified the slope of the desired line as $\beta_1 = \tan\theta$, and Δy_i as the usual LS vertical residual. With this notation we find that the TLS estimator of β_0 and β_1 is found by minimizing

$$\begin{aligned} R(\beta_0, \beta_1) &= \sum_i R_i^2 \\ &= \frac{1}{1 + \beta_1^2} \sum_i (y_i - \beta_0 - \beta_1 x_i)^2 \end{aligned} \tag{9}$$

which differs from ordinary LS only in the premultiplier.

Now we proceed as usual and look for the stationary points of R. For β_0 we get

$$0 = \frac{\partial R}{\partial \beta_0} = \frac{1}{1+\beta_1^2} \sum_i -2(y_i - \beta_0 - \beta_1 x_i). \tag{10}$$

So for the TLS solution

$$\beta_0 = \frac{1}{n}\left[\sum_i y_i - \beta_1 \sum x_i\right] = \bar{y} - \beta_1 \bar{x} \tag{11}$$

As in ordinary LS the TLS solution must pass through the center of mass $(\bar{x}, \bar{y})$ of the data cloud. For β_1 we get

$$0 = \frac{\partial R}{\partial \beta_1} = \frac{1}{1+\beta_1^2} \sum_i -2(y_i - \beta_0 - \beta_1 x_i)x_i - \frac{2\beta_1}{(1+\beta_1^2)^2} \sum_i (y_i - \beta_0 - \beta_1 x_i)^2. \tag{12}$$

Now we can eliminate β_0 by substituting the result from (11) into (12). Cleaning up yields a quadratic in β_1

$$0 = c_2 \beta_1^2 + c_1 \beta_1 + c_0$$

with coefficients

$$\begin{aligned} c_2 &= \sum_i (x_i - \bar{x})(y_i - \bar{y}) \\ c_1 &= \sum_i [(x_i - \bar{x})^2 - (y_i - \bar{y})^2] \\ c_0 &= -\sum_i (x_i - \bar{x})(y_i - \bar{y}) \end{aligned}$$

Generalizing to the case with unequal weights for different data pairs is straightforward, but if the variances in x and y are different one needs to choose a different distance measure, so that the direction with larger variance contributes less to R_i.

2. Robustness

In Chapter 5, we got our first flavor of the differences between parametric and non-parametric methods when we discussed Monte Carlo methods and the bootstrap. In **parametric** methods we assume a functional form for the pdf of the observations. It is often the case that we want to find the parameters in that function, and we have spent a fair amount of time discussing optimal methods for doing this. In **non-parametric** methods the goal is to make minimal assumptions about the pdf for the observations. We might for example be prepared to assume that the observations are iid or that the underlying distribution is continuous and has a density function. One might think of non-parametric problems in the same light as problems with an infinite number of unknown parameters: in other geophysical contexts, these are often known as inverse problems.

In **robust** data analysis one usually assumes a functional form for the probability distribution, but worries about whether the procedure is sensitive to small departures from that basic form. In a rather loose mathematical sense, when we talk of a robust method we mean one that works well over a collection of distributions that lie in the neighborhood of, and also include, the assumed family.

There are a number of reasons why we need robust methods, ranging from gross blunders in measurement or recording of data, to intrinsic limitations in precision, or perhaps just implicit reliance on the central limit theorem to justify assumptions about underlying distributions. An example of a robust method that we have already seen in section 5.1.3 involves using the 10% trimmed mean to minimize the influence of erroneous data on estimates for the mean. Robust data processing methods have been in use for a long time: some of the first geophysical examples involved assuming mixtures of normal distributions with different variances, usually low level contamination by a small fraction of outliers from a distribution with large variance. The L1 norm minimization (as opposed to least squares) that is associated with maximum likelihood estimation from a Laplacian distribution is another early example.

There are several important questions to address in developing robust methods: Is the procedure sensitive to small departures from the model? A more quantitative question would be the following: To 1st order what is the sensitivity to departures from assumptions? And perhaps the most important issue is: How far can one depart from the model before real trouble arises?

The last of these questions is concerned with the breakdown point for the method, and may be addressed by considering what fraction of the data we need to change before the method breaks down. This gives rise to the idea of developing methods that are resistant to bad data. For example if we consider the mean of a collection of observations, $x_i,\ i = 1, \ldots, n$ we find that a single bad datum can destroy the estimate. Since we have

$$\bar{x} = \frac{1}{n} \sum_{i}^{n} x_i \tag{13}$$

the breakdown point is $\frac{1}{n}$. But the situation is much better if one uses the α- trimmed mean

$$\bar{x}_\alpha = \frac{1}{k_h - k_l + 1} \sum_{i=k_l}^{k_h} x_{(i)} \tag{14}$$

with $k_l + n - k_h = \alpha n$, and $x_{(i)}$ the ith order statistic.

2:1 Loss and Influence Functions

Many robust methods rely on the idea of the influence function which is essentially the derivative of a loss function like the $||\vec{r}||^2$ minimized in least squares estimation (LSE). Let us suppose that we can write our parametrized model of interest as

$$y_i = \sum_{j=1}^{p} \theta_j c_j(x_i) + \epsilon_i = g(x_i) + \epsilon_i \qquad i = 1, \ldots, n \tag{15}$$

or in matrix form

$$\vec{y} = C\vec{\theta} + \vec{\epsilon} \tag{16}$$

The LSE minimizes a quadratic loss function and if the ϵ_i are Gaussian, the MLE and LSE are equivalent. That is

$$\max_{\hat{\theta}} l(\hat{\theta}) = \min_{\hat{\theta}} \sum_{i=1}^{n} r_i^2 \tag{17}$$

The residuals r_i depend on $\hat{\theta}$ and the distribution of ϵ_i, so let us replace the quadratic form in (17) by a more general loss function ρ. Following a maximum likelihood kind of approach we can write

$$\rho(\epsilon) = -\ln[\mathcal{L}(\epsilon, \hat{\theta})]$$

with an implicit dependence on θ and find the values for $\hat{\theta}$ that minimize this new loss function. $\mathcal{L}(\epsilon, \hat{\theta})$ is the likelihood function for θ as introduced in Section 5.5 in the context of estimating parameters for a specific statistical distribution. The necessary condition for the minimum is

$$\sum_{i=1}^{n} c_j(x_i)\psi(\epsilon_i) = 0 \qquad j = 1, 2, \ldots, p \tag{18}$$

where $\psi(\epsilon) = \rho'(\epsilon)$ is proportional to something known in the statistical literature as the **influence function**, and is implicitly dependent on the choice of the parameters $\hat{\theta}$. In (18) the $c_j(x_i), j = 1, \ldots, p$ form the rows of the design matrix C.

For a normal distribution with both ML and LS estimates we have $\rho(\epsilon) = \epsilon^2$ and $\psi(\epsilon) = \epsilon$, while with Laplacian or exponentially distributed uncertainties and maximum likelihood estimation we would have $\rho(\epsilon) = |\epsilon|$ and $\psi(\epsilon) = sgn(\epsilon)$. These loss and influence functions are illustrated in Figure 8.2, from which you can see that in the Gaussian case the influence of a data point rises linearly with its deviation or residual from the model prediction $g(x_i)$. In contrast the exponential error gives equal influence to all data regardless of how far they lie from the model. It's easy to see why this is the case when you consider the median as an estimate of scale. One chooses the mid point of the ordered data, and it is irrelevant how far the rest of the observations lie from the median - each data point has an equal influence on the estimate.

Robust estimation makes substantial use of the idea of the influence function. If we think again of the trimmed mean, the influence function is linear in the central $1 - \alpha$ fraction of the observations, then drops to

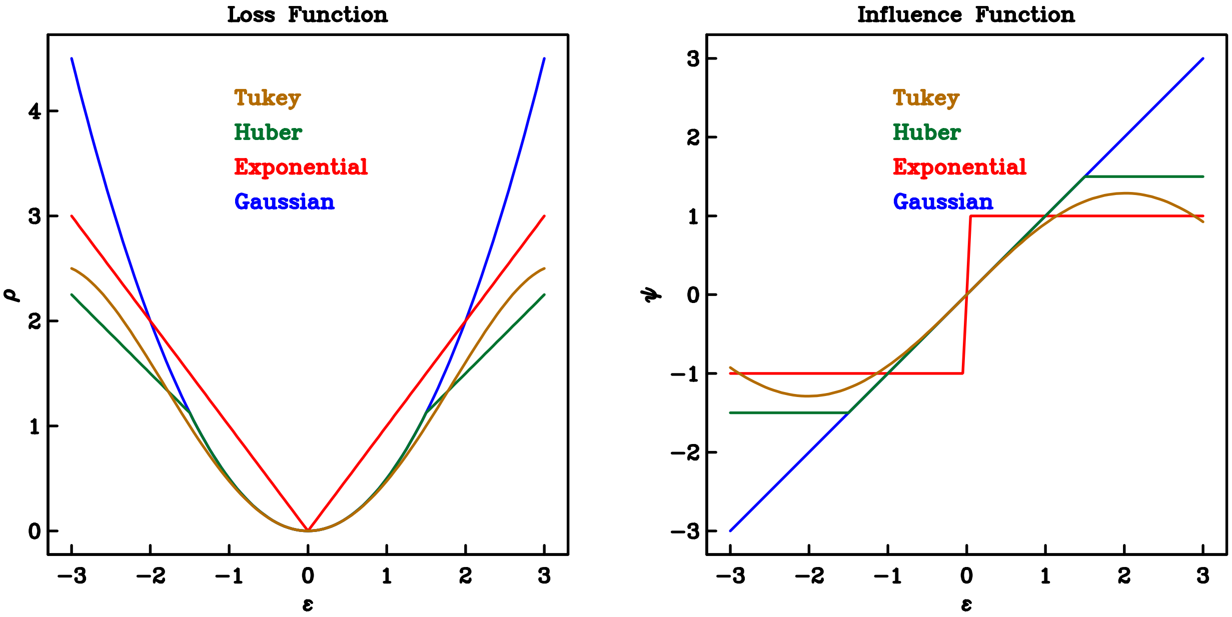

Figure 8.2: Loss and influence function for ML estimation with Gaussian and exponentially distributed noise. Robust M-type loss and influence functions for Huber's t-function with $t = 1.5$, and Tukey's biweight with $t = 4.5$.

zero. There is a whole class of robust estimates known as M-type (for maximum likelihood type) estimates that make use of the loss function minimization formulation. The idea is to make $\rho(\epsilon)$ increase less rapidly than $\sum r_i^2$ so that the parameter value we estimate can be resistant to a few gross outliers in an otherwise well-behaved distribution. Some examples of robust influence functions are also shown in Figure 8.2.

Two widely used examples are Huber's t-function with

$$\psi(\epsilon) = \begin{cases} \epsilon, & |\epsilon| < t \\ t, & |\epsilon| \geq t \end{cases} \tag{19}$$

And Tukey's biweight function

$$\psi(\epsilon) = \begin{cases} \epsilon(1 - \epsilon^2/t^2)^2, & |\epsilon| < t \\ 0, & |\epsilon| \geq t \end{cases} \tag{20}$$

It is readily seen that while Huber's t limits the influence of outlying data, it still allows them to contribute in the fitting process, while Tukey's biweight is a redescending function, that rejects any influence from data lying further than t from the model. The choice of t is in both cases somewhat *ad hoc* and must be made cautiously.

An outstanding piece of business is how we should solve equation (18) for the θ_j. (18) is in general non-linear, so we resort to iterative methods. We can use Newton's method: we start with an initial guess for $\hat{\theta}$ (the LS estimate would do fine), call this $\hat{\theta}_0$ and then iterate from there using

$$\frac{\partial(l(\hat{\theta}))}{\partial\hat{\theta}} = \left.\frac{\partial l}{\partial\hat{\theta}}\right|_{\hat{\theta}_0} + \left.\frac{\partial^2 l}{\partial\hat{\theta}^2}\right|_{\hat{\theta}_0}(\hat{\theta} - \hat{\theta}_0) = 0. \tag{21}$$

But it's not always obvious how to evaluate $\frac{\partial^2 l}{\partial \hat{\theta}^2}$, so a commonly used alternative is to resort to iteratively reweighted least squares. To do this we go back to (18), and rewrite it as

$$\sum_{i=1}^{n} c_j(x_i) \frac{\psi(\epsilon_i)}{\epsilon_i} \epsilon_i = 0 \qquad j = 1, 2, \ldots, p \tag{22}$$

Now we let $g_0(x_i)$ be the model prediction for g based on $\hat{\theta}_0$, an initial guess at the solution and

$$w_{i_0} = \begin{cases} \frac{\psi(\epsilon_i)}{\epsilon_i} = \frac{\psi(y_i - g_0(x_i))}{(y_i - g_0(x_i))} & y_i \neq g_0(x_i) \\ \lim y_i \rightarrow g_0(x_i) & \quad \text{otherwise.} \end{cases} \tag{23}$$

This transforms (18) to a weighted LS problem with

$$C^T W_k C \hat{\theta}_k = C^T W_k \vec{y} \qquad k = 0, 1, \ldots \tag{24}$$

with W_k the diagonal matrix of weights, $(w_{1k}, w_{2k}, \ldots, w_{nk})$, at the kth iteration. The problem is re-solved until it converges to a solution. Note that convergence is only guaranteed in the regime where the loss function is convex, that is if $\psi(\epsilon)$ is non-decreasing.

For much more on the topic of robust estimation see the book by Hampel *et al.* (1986), or alternatively the brief discussion in Press *et al.*. An application of some of these ideas can be found in Constable (1988, Parameter estimation in non-Gaussian noise, Geophys. J. Int. Vol 94,131–142). In that paper the form for the loss function is inferred by an iterative procedure using the distribution of residuals and M-type estimation.