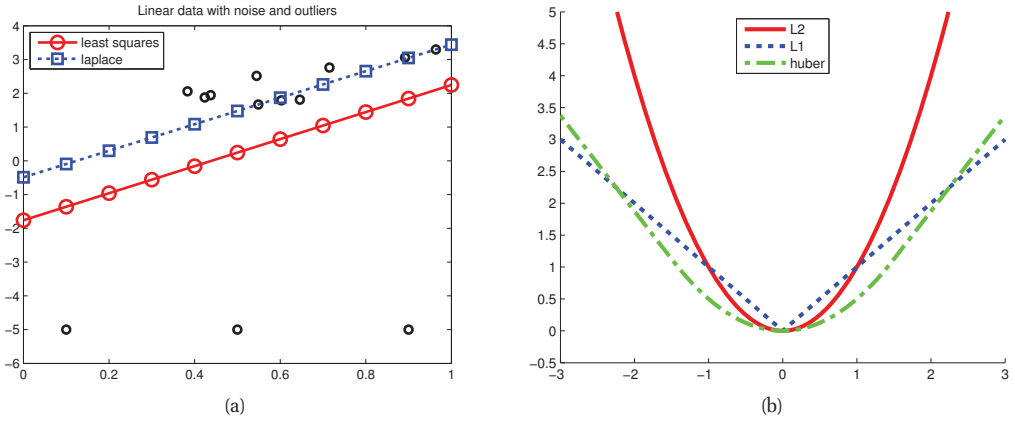

Figure 7.6 (a) Illustration of robust linear regression. Figure generated by `linregRobustDemoCombined`. (b) Illustration of ℓ_2, ℓ_1, and Huber loss functions. Figure generated by `huberLossDemo`.

Models where the NLL is convex are desirable, since this means we can always find the globally optimal MLE. We will see many examples of this later in the book. However, many models of interest will not have concave likelihoods. In such cases, we will discuss ways to derive locally optimal parameter estimates.

7.4 Robust linear regression *

It is very common to model the noise in regression models using a Gaussian distribution with zero mean and constant variance, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, where $\epsilon_i = y_i - \mathbf{w}^T\mathbf{x}_i$. In this case, maximizing likelihood is equivalent to minimizing the sum of squared residuals, as we have seen. However, if we have **outliers** in our data, this can result in a poor fit, as illustrated in Figure 7.6(a). (The outliers are the points on the bottom of the figure.) This is because squared error penalizes deviations quadratically, so points far from the line have more affect on the fit than points near to the line.

One way to achieve **robustness** to outliers is to replace the Gaussian distribution for the response variable with a distribution that has **heavy tails**. Such a distribution will assign higher likelihood to outliers, without having to perturb the straight line to "explain" them.

One possibility is to use the Laplace distribution, introduced in Section 2.4.3. If we use this as our observation model for regression, we get the following likelihood:

$$p(y|\mathbf{x}, \mathbf{w}, b) \quad = \quad \text{Lap}(y|\mathbf{w}^T\mathbf{x}, b) \propto \exp(-\frac{1}{b}|y - \mathbf{w}^T\mathbf{x}|) \tag{7.24}$$

The robustness arises from the use of $|y - \mathbf{w}^T\mathbf{x}|$ instead of $(y - \mathbf{w}^T\mathbf{x})^2$. For simplicity, we will assume b is fixed. Let $r_i \triangleq y_i - \mathbf{w}^T\mathbf{x}_i$ be the i'th residual. The NLL has the form

$$\ell(\mathbf{w}) = \sum_i |r_i(\mathbf{w})| \tag{7.25}$$

Likelihood	Prior	Name	Section
Gaussian	Uniform	Least squares	7.3
Gaussian	Gaussian	Ridge	7.5
Gaussian	Laplace	Lasso	13.3
Laplace	Uniform	Robust regression	7.4
Student	Uniform	Robust regression	Exercise 11.12

Table 7.1 Summary of various likelihoods and priors used for linear regression. The likelihood refers to the distributional form of $p(y|\mathbf{x}, \mathbf{w}, \sigma^2)$, and the prior refers to the distributional form of $p(\mathbf{w})$. MAP estimation with a uniform distribution corresponds to MLE.

Unfortunately, this is a non-linear objective function, which is hard to optimize. Fortunately, we can convert the NLL to a linear objective, subject to linear constraints, using the following **split variable** trick. First we define

$$r_i \triangleq r_i^+ - r_i^- \tag{7.26}$$

and then we impose the linear inequality constraints that $r_i^+ \geq 0$ and $r_i^- \geq 0$. Now the constrained objective becomes

$$\min_{\mathbf{w},\mathbf{r}^+,\mathbf{r}^-} \sum_i (r_i^+ - r_i^-) \qquad \text{s.t.} \quad r_i^+ \geq 0, r_i^- \geq 0, \mathbf{w}^T\mathbf{x}_i + r_i^+ + r_i^- = y_i \tag{7.27}$$

This is an example of a **linear program** with $D + 2N$ unknowns and $3N$ constraints.

Since this is a convex optimization problem, it has a unique solution. To solve an LP, we must first write it in standard form, which as follows:

$$\min_{\boldsymbol{\theta}} \mathbf{f}^T\boldsymbol{\theta} \;\; \text{s.t.} \;\; \mathbf{A}\boldsymbol{\theta} \leq \mathbf{b}, \;\; \mathbf{A}_{eq}\boldsymbol{\theta} = \mathbf{b}_{eq}, \;\; \mathbf{l} \leq \boldsymbol{\theta} \leq \mathbf{u} \tag{7.28}$$

In our current example, $\boldsymbol{\theta} = (\mathbf{w}, \mathbf{r}^+, \mathbf{r}^-)$, $\mathbf{f} = [\mathbf{0}, \mathbf{1}, \mathbf{1}]$, $\mathbf{A} = []$, $\mathbf{b} = []$, $\mathbf{A}_{eq} = [\mathbf{X}, \mathbf{I}, -\mathbf{I}]$, $\mathbf{b}_{eq} = \mathbf{y}$, $\mathbf{l} = [-\infty\mathbf{1}, \mathbf{0}, \mathbf{0}]$, $\mathbf{u} = []$. This can be solved by any LP solver (see e.g., (Boyd and Vandenberghe 2004)). See Figure 7.6(a) for an example of the method in action.

An alternative to using NLL under a Laplace likelihood is to minimize the **Huber loss** function (Huber 1964), defined as follows:

$$L_H(r, \delta) = \begin{cases} r^2/2 & \text{if } |r| \leq \delta \\ \delta|r| - \delta^2/2 & \text{if } |r| > \delta \end{cases} \tag{7.29}$$

This is equivalent to ℓ_2 for errors that are smaller than δ, and is equivalent to ℓ_1 for larger errors. See Figure 7.6(b). The advantage of this loss function is that it is everywhere differentiable, using the fact that $\frac{d}{dr}|r| = \text{sign}(r)$ if $r \neq 0$. We can also check that the function is C_1 continuous, since the gradients of the two parts of the function match at $r = \pm\delta$, namely $\frac{d}{dr}L_H(r, \delta)|_{r=\delta} = \delta$. Consequently optimizing the Huber loss is much faster than using the Laplace likelihood, since we can use standard smooth optimization methods (such as quasi-Newton) instead of linear programming.

Figure 7.6(a) gives an illustration of the Huber loss function. The results are qualitatively similiar to the probabilistic methods. (In fact, it turns out that the Huber method also has a probabilistic interpretation, although it is rather unnatural (Pontil et al. 1998).)