1

Introduction to Rank-based Regression

1.1 Introduction

The purpose of this book is to lay the groundwork for robust data science using rank-based methods. The field of machine learning has not yet fully embraced a class of robust estimators that addresses issues that limit the value of least squares estimation. For example, outliers in data sets may produce misleading results that are not suitable for inference. They can also affect results obtained from penalty estimators. We believe that robust estimators for regression problems are well-suited to data science. This book is intended to provide both practical and mathematical foundations in the study of rank-based methods. It will introduce a number of new ideas and approaches to the practice and theory of robust estimation and encourage readers to pursue further investigation in this field. While the main goal of this book is to provide a rigorous treatment of the subject matter, we begin with some introductory material to build insight and intuition about rank-based regression and penalty estimators, especially for those who are new to the topic and those looking to understand key concepts. To motivate the need for such methods, we will start with a discussion of the median as it is the key to rank-based methods and then build on that concept towards the notion of robust data science.

1.2 Robustness of the Median

1.2.1 Mean vs. Median

Our starting point is a brief review of two useful statistics: the mean and the median. We want to assess their suitability and usefulness in the context of simple linear regression. We state up front that the mean is the basis for the most

Rank-Based Methods for Shrinkage and Selection: With Application to Machine Learning. First Edition. A. K. Md. Ehsanes Saleh, Mohammad Arashi, Resve A. Saleh, and Mina Norouzirad.

popular regression methods but it is worth revisiting this approach relative to other methods.

Given a set of n values, $\boldsymbol{x} = \{x_1, \ldots, x_n\}$, the arithmetic mean (mean for short in our study) is the average of the values while the median is the middle value when placed in sorted order. The mean is given by

$$\text{mean}(\boldsymbol{x}) = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{1.2.1}$$

which is the sum of the values divided by n.

The median of the set depends on whether n is even or odd. First, we order the values from smallest to largest. After ordering, the new set is denoted by $\{x_{(1)}, \ldots, x_{(n)}\}$, where $x_{(1)} \leq \cdots \leq x_{(n)}$. These are referred to as the order statistics of $\boldsymbol{x}$ with $x_{(1)} = \min\{x_i,\ i = 1, \ldots, n\}$ and $x_{(n)} = \max\{x_i,\ i = 1, \ldots, n\}$, and $x_{(i)}$ is the ordered observations. If n is odd, we take the middle value in the ordered set, i.e. $\text{median}(x) = x_{(\frac{n+1}{2})}$. If n is even, we take the average of the two values surrounding the middle. This case can be expressed compactly as follows. Assuming the above ordering, then

$$\text{median}(\boldsymbol{x}) = \frac{1}{2}(x^{(l)} + x^{(u)}), \tag{1.2.2}$$

where $x^{(l)} = x_{\left(\frac{n}{2}\right)}$ and $x^{(u)} = x_{\left(\frac{n}{2}+1\right)}$. The lower and upper values are averaged to obtain the median. We can already see a problem in that the mean is easy to understand using one equation, but the median requires a number of steps and a somewhat complicated description. However, the median has significant value when it comes to finding the essence of a data set, as we will see shortly.

Table 1.1 compares and contrasts these two statistics. Three data sets are provided to illustrate the characteristics of each statistic. The first set $\{3, 1, 4, 2, 5, 3\}$ results in the same value of 3 for the mean and median. This is consistent with a visual examination of the values in the set. However, if we change one of the values from 3 to 100 to produce the second set, $\{100, 1, 4, 2, 5, 3\}$, the mean increases significantly to 19.2, while the median produces a value of 3.5 close to the previous value. In this case, the value of 100 is considered an *outlier* and we can observe its dramatic effect on the two statistics. The mean is greatly affected by one outlier which makes it unstable. On the other hand, the median seems rather stable in the presence of this outlier. Mean instability is more apparent with two outliers as in set 3.

If we now let $x_{(n)}$ represent the outlier value in set 2, hence $x_{(n)} = 100$ initially, we note that as $x_{(n)} \to \infty$, the mean tends to ∞ while the median remains

at 3.5. For example, $x_{(n)} = 1000$ changes the mean to 169.2 while the median stays at 3.5. This is problematic enough, but the standard deviation that uses the mean is also greatly affected. Let μ be the population mean, and let $\bar{x}$ of Eq. (1.2.1) be the estimate of μ. Further, let the population standard deviation, σ, be defined by

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2},$$

and the estimator of σ, which is the standard deviation (sd) of the sample set, be given by[1]:

$$\mathrm{sd}(\boldsymbol{x}) = \hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}. \tag{1.2.3}$$

Then, as the value of the outlier increases to infinity, the mean goes to infinity and, as a consequence, the standard deviation goes to infinity. Hence, one outlier can have a domino effect on various statistics based on the mean.

A robust statistic for a set containing outliers is the median absolute deviation (MAD) given by

$$\mathrm{MAD}(\boldsymbol{x}) = \underset{1\le i\le n}{\mathrm{median}}\big(|x_i - \mathrm{median}(\boldsymbol{x})|\big). \tag{1.2.4}$$

A consistent estimator of the population standard deviation based on MAD can be computed as follows[2]

$$\hat{\eta} = \mathrm{mad}(\boldsymbol{x}) = k \times \mathrm{MAD}(\boldsymbol{x}), \tag{1.2.5}$$

where $k = 1.4826$ (Johnson and Peng, 2008, cf.)[3]. If we take set 1 in Table 1.1, the original set without outliers, then $\mathrm{sd}(\boldsymbol{x}) = 1.41$ and $\mathrm{mad}(\boldsymbol{x}) = 1.48$. For the

Table 1.1 Comparison of mean and median on three data sets.

Measure	Set 1 {3,1,4,2,5,3}	Set 2 {100,1,4,2,5,3}	Set 3 {100,100,1,4,2,5,3}
Mean	3.0	19.2	30.7
Median	3.0	3.5	4.0

1 We follow the naming convention of R where $\mathrm{sd}(x)$ is the standard deviation.
2 We follow the naming convention of the R command where $\mathrm{mad}(x) = k \times \mathrm{MAD}(\boldsymbol{x})$.
3 In this book, $\hat{\eta}$ is used to denote the robust estimator of this quantity. We use a multiplying factor of 1.4826 to make it an unbiased summary statistic.

case of set 2 with $x_{(n)} = 100$, $\text{sd}(\boldsymbol{x}) = 39.6$ and $\text{mad}(\boldsymbol{x}) = 2.22$. If we now set $x_{(n)} = 1000$, then $\text{sd}(\boldsymbol{x}) = 407.0$ and $\text{mad}(\boldsymbol{x}) = 2.22$; that is, it remains the same. Clearly, a single large outlier can disrupt the integrity of the mean and standard deviation while the median and $\text{mad}(\boldsymbol{x})$ remain stable even as $x_{(n)} \to \infty$.

Next, if we examine set 3 listed as $\{100, 100, 1, 4, 2, 5, 3\}$ in the final column of Table 1.1, we now have two outliers and the mean shifts further beyond the range of the original elements of set 1. On the other hand, the median maintains relative stability. We refer to the median as having the *robustness* property for this reason.

1.2.2 Breakdown Point

To illustrate further, we could continue to add more outliers to the data set that lie between 100 and 120, up to the *breakdown point* when 50% of the data are outliers, and the median would still provide a good representation of the original data set. Of course, if we exceed the 50% point, the values in the original data become outliers and the outliers become the actual data. Hence, the breakdown point of an estimator lies between 0 and 0.5, with 0.5 being the best and 0 being the worst.

The breakdown point (Wilcox, 2012) is a measure of the tolerance level of a statistic or estimator to outliers. The breakdown point of an estimator is the proportion of incorrect (i.e. arbitrarily large) observations an estimator can handle before giving an incorrect (i.e. arbitrarily large) result. The higher the breakdown point of an estimator, the more robust it is. The mean statistic is the least tolerant (in fact, the breakdown point is $\frac{1}{n}$ for a sample of size n observations and nearly zero for large sample sizes) whereas the median may tolerate up to 50% of the data as outliers (the statistic remains in a compact set). This is rather remarkable and provides a strong basis to pursue rank-based methods that are based on the median.

Intuitively, we can understand that a breakdown point cannot exceed 50% because if more than half the observations are contaminated, it is not possible to distinguish between the underlying distribution and the contaminating distribution (Rousseeuw and Leroy, 1987). Therefore, the maximum breakdown point is 0.5 and there are estimators that achieve such a breakdown point. However, there is often a trade-off in such estimators where we may sacrifice certain regularity conditions, such as consistency and asymptotic normality, to be discussed later. It is better to retain these regularity properties even if the breakdown point is reduced below 0.5. In particular, the rank-based methods have a breakdown point around 0.3 but possess the desired regularity conditions and are very efficient estimators.

1.2.3 Order and Rank Statistics

It is important to clarify the difference between "ordered" and "rank" statistics. Let X_1, ..., X_n be a random sample of variables. Now, $X_{(i)}$ is the ith smallest random variable and $X_{(1)}$, ..., $X_{(n)}$ are referred to as *ordered* statistics of X_1, ..., X_n. By definition, $X_{(1)} \leq ... \leq X_{(n)}$. We say that X_i has *rank* R_i among X_1, ..., X_n if $X_i = X_{(R_i)}$ assuming the R_ith order statistic is uniquely defined. By "uniquely defined", we are assuming that ties do not occur. That is $X_{(i)} \neq X_{(j)}$ for all $i \neq j$. Table 1.2 provides examples to clearly illustrate these notions.

Based on the definition of rank, the median is the observation for which the rank equals $\frac{n+1}{2}$ in case of odd n, and the mean of two observations that are ranked $\frac{n}{2}$ and $\frac{n}{2}+1$, when n is even. In Table 1.2, for the first three sets, $n = 3$ (odd case), so the value of rank 2 is the median; hence, this is the second value in sets X and Y, and the first value in the third set, Z. For the fourth set, n is even, so we select the two values with ranks 2 and 3; thus, the median is average of these two values, i.e. $(4+5)/2 = 4.5$. Outliers, if any, will be at the extremes of order statistics. The fourth set is said to have a potential outlier at the extreme order statistic, $W_{(4)} = 99$.

The above series of examples provides some intuition as to why *rank-based* methods that use the median (and some other measures that will be identified later) may be of significant value in regression problems relative to least squares estimation (LSE) which is based on the mean. This not only applies to LSE, it also applies to any mean regression, such as generalized linear models (GLM). Many realistic data sets have a non-zero percentage of outliers and we may be interested in finding them and/or utilizing estimators that are robust when outliers are present. This is the value proposition of rank-based methods.

Table 1.2 Examples comparing order and rank statistics.

Sets	Ordered statistics	Rank statistics
$X = \{1, 2, 3\}$	$X_{(1)} = 1, X_{(2)} = 2, X_{(3)} = 3$	$R_1 = 1, R_2 = 2, R_3 = 3$
$Y = \{3, 2, 1\}$	$Y_{(1)} = 1, Y_{(2)} = 2, Y_{(3)} = 3$	$R_1 = 3, R_2 = 2, R_3 = 1$
$Z = \{5, 6, 4\}$	$Z_{(1)} = 4, Z_{(2)} = 5, Z_{(3)} = 6$	$R_1 = 2, R_2 = 3, R_3 = 1$
$W = \{5, 3, 99, 4\}$	$W_{(1)} = 3, W_{(2)} = 4,$	$R_1 = 3, R_2 = 1,$
	$W_{(3)} = 5, W_{(4)} = 99$	$R_3 = 4, R_4 = 2$

1.3 Simple Linear Regression

We have studied the characteristics of the mean and median for simple data sets to understand the differences between the two. We now explore the use of the mean and median for simple linear regression. A one-dimensional linear regression problem will be sufficient for our purposes here. We refer to the dimensionality of this problem as $p = 1$; for higher dimensions, $p \geq 2$. Suppose we are given a data set comprised of $\boldsymbol{x} = (x_1, \dots, x_n)^\top \in \mathbb{R}^n$ and $\boldsymbol{y} = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$. Our objective is to fit a line to the data such that the line has the smallest error, $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^\top \in \mathbb{R}^n$, and provides a good representation of the data. More concretely, we describe the simple linear model with the following equation

$$\boldsymbol{y} = \theta \mathbf{1}_n + \beta \boldsymbol{x} + \boldsymbol{\varepsilon}, \tag{1.3.1}$$

where the two parameters of interest are the slope of the line, β, and the y-intercept, θ, and $\mathbf{1}_n = (1, \dots, 1)^\top$ is an n-vector of 1. We seek to estimate θ and β using the least squares (LS) method. Note that $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{\varepsilon}$ are all vectors of length n, assuming we are given n data points from which to fashion a line.

1.3.1 Least Squares Estimator (LSE)

The classical LSE to determine β and θ is based on minimization of the residual sum of squares (RSS). We denote the estimated values from LSE as $\hat{\beta}_n^{\text{LS}}$ and $\hat{\theta}_n^{\text{LS}}$, respectively. For arbitrary values of $\hat{\theta}_n$ and $\hat{\beta}_n$ and a specific data point (x_i, y_i), the residual is defined as $\hat{\varepsilon}_i = e_i = y_i - (\hat{\theta}_n - \hat{\beta}_n x_i)$. Then the residual sum of the squares (RSS) is simply

$$\text{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2. \tag{1.3.2}$$

If we minimize the RSS for a given data set $\boldsymbol{x}$ and $\boldsymbol{y}$, then the estimates are

$$\hat{\beta}_n^{\text{LS}} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \tag{1.3.3}$$

and

$$\hat{\theta}_n^{\text{LS}} = \bar{y} - \hat{\beta}_n^{\text{LS}} \bar{x}. \tag{1.3.4}$$

Note the use of the mean values, $\bar{x}$ and $\bar{y}$, to estimate the two parameters. This is an important point to make since LSE provides the *optimal unbiased estimation*

of the slope and intercept for a given set of data points. More generally, it is the maximum likelihood estimate under the normality assumptions for the error.

1.3.2 Theil's Estimator

We can also estimate the parameters using the median by way of Theil's estimator (Theil, 1950). For a given set of points, $\{(x_i, y_i)\}_{i=1}^n$, Theil's estimators of β and θ can be obtained by finding the median of the pairwise slopes of all points in the data set, denoted by $\hat{\beta}_n^{\text{Theil}}$, and the median of partial-residuals, $(y_i - \hat{\beta}_n^{\text{Theil}} x_i)$, which we denote by $\hat{\theta}_n^{\text{Theil}}$. The actual residuals are $(y_i - (\hat{\theta}_n^{\text{Theil}} + \hat{\beta}_n^{\text{Theil}} x_i))$ which are obtained after estimation. See Hallander and Wolfe (1999) for more details. Hence, we have that

$$\hat{\beta}_n^{\text{Theil}} = \underset{1 \le i < j \le n,\, i \ne j}{\text{median}} \left(\frac{y_j - y_i}{x_j - x_i} \right), \tag{1.3.5}$$

and

$$\hat{\theta}_n^{\text{Theil}} = \underset{1 \le i \le n}{\text{median}} \left(y_i - \hat{\beta}_n^{\text{Theil}} x_i \right). \tag{1.3.6}$$

Theil's method is an unbiased estimator. Its breakdown point is known to be 29.3%. While this is not as high as 50%, it can tolerate outliers to a higher degree than LSE making it a more robust estimator.

1.3.3 Belgium Telephone Data Set

It is instructive to qualitatively compare the mean versus median approaches using a very simple data set. For this purpose, we will use the Belgium telephone data set (Kloke and McKean, 2012). This data set is provided in Table 1.3 and contains the number of calls (in units of tens of millions) made in Belgium in the years between 1950 and 1973. It has 24 data points, with 6 outliers. The outliers are due to a change in measurement technique without re-calibration for 6 years, as is often cited.

Table 1.3 Belgium telephone data set.

x	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961
y	0.44	0.47	0.47	0.59	0.66	9.73	0.81	0.88	1.06	1.2	1.35	1.49
x	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973
y	1.61	2.12	11.9	12.4	14.2	15.9	18.2	21.2	4.3	2.4	2.7	2.9

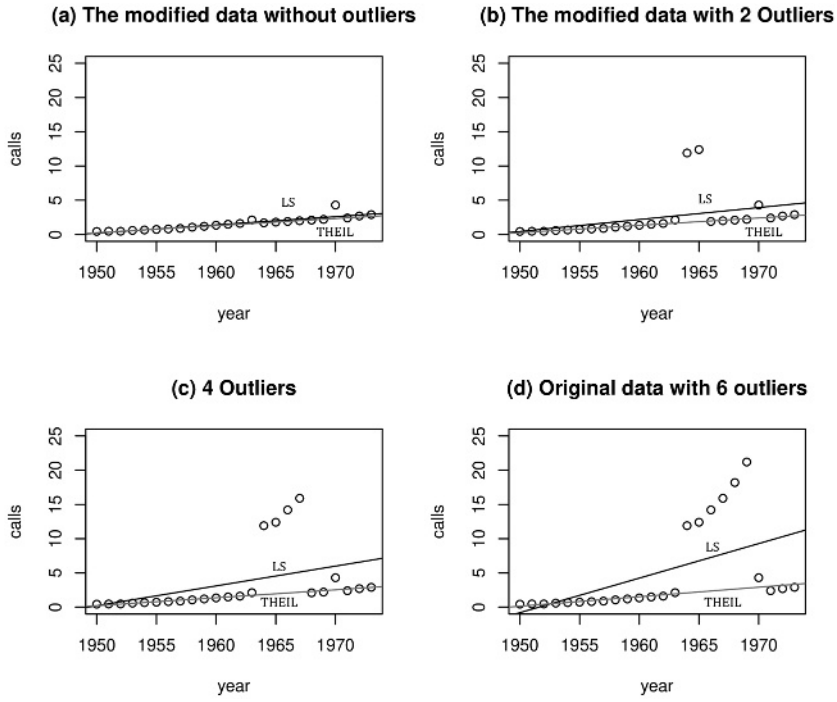

Figure 1.1 Four plots using different versions of the telephone data set with fitted lines. The median-based approach (Theil estimation) is robust and stable while the LS line varies depending on the number of outliers.

We will mainly use this data set throughout this chapter. However, we will also adjust the data values to control the number of outliers in order to accentuate the differences between the two competing approaches.

In Figure 1.1, we show four different cases. The first three data sets, Figures 1.1(a), (b) and (c) are modified to have zero, two, and four outliers, respectively. Figure 1.1(d) is the original data set with six outliers, as given in Table 1.3.

The results of linear regression using LSE and Theil are shown as fitted lines on the scatter plots of Figure 1.1. Note that, as we add more outliers in the data set, the lines associated with LS change and move toward the outliers, while the lines based on the median (Theil) remain relatively stable. This figure gives us an indication of how easily LSE values can shift due to a few outliers. The process of parameter estimation, the calculation of the standard error statistic and hypothesis testing for each case is discussed in the next section.

1.3.4 Estimation and Standard Error Comparison

We will now carry out a quantitative comparison. The equations for each estimator were provided earlier. We begin this section with a description of the standard error for each estimator and end with hypothesis testing. The standard error (s.e.) is a measure of the variability or accuracy of parameter estimation.

The LSE standard errors for the respective estimates are given by

$$\text{s.e.}(\hat{\theta}_n^{\text{LS}}) = \sqrt{\hat{\sigma}^2\left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]}, \tag{1.3.7}$$

and

$$\text{s.e}(\hat{\beta}_n^{\text{LS}}) = \sqrt{\frac{\hat{\sigma}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}, \tag{1.3.8}$$

where σ can be estimated using the residual standard error (RSE) as follows:

$$\hat{\sigma} \approx \text{RSE} = \sqrt{\text{RSS}/(n-2)}. \tag{1.3.9}$$

For Theil's estimator, we employ MAD to calculate the standard errors. Using following definition of partial-residuals, $r_i = (y_i - \hat{\beta}_n^{\text{Theil}} x_i)$, we obtain

$$\text{s.e.}(\hat{\theta}_n^{\text{Theil}}) = k \times \underset{1\le i\le n}{\text{median}}\left(|r_i - \hat{\theta}_n^{\text{Theil}}|\right), \tag{1.3.10}$$

and we use the slopes, $s_{ij} = (y_j - y_i)/(x_j - x_i)$, to obtain

$$\text{s.e.}(\hat{\beta}_n^{\text{Theil}}) = k \times \underset{1\le i<j\le n,\, i\ne j}{\text{median}}\left(|s_{ij} - \hat{\beta}_n^{\text{Theil}}|\right). \tag{1.3.11}$$

Our quantitative comparison involves only Figures 1.1(a) (no outliers) and 1.1(d) (six outliers). We use Eqs. (1.3.3) and (1.3.4) to obtain the LS estimates. First, we compute $\hat{\beta}_n^{\text{LS}}$ from Eq. (1.3.3) and use that result in Eq. (1.3.4) to compute $\hat{\theta}_n^{\text{LS}}$. For Figure 1.1(a), we obtain $\hat{\beta}_n^{\text{LS}} = 0.12$ and $\hat{\theta}_n^{\text{LS}} = -234.2$; however, for Figure 1.1(d) we obtain $\hat{\beta}_n^{\text{LS}} = 0.50$ and $\hat{\theta}_n^{\text{LS}} = -983.9$. Clearly, there is a significant change in the LS estimates when outliers are added.

To obtain the estimates using the median due to Theil, we apply Eqs. (1.3.5) and (1.3.6). First, we compute $\hat{\beta}_n^{\text{Theil}}$ from Eq. (1.3.5) and use that result in Eq. (1.3.6) to compute $\hat{\theta}_n^{\text{Theil}}$. For Figure 1.1(a), we obtain $\hat{\beta}_n^{\text{Theil}} = 0.10$ and $\hat{\theta}_n^{\text{Theil}} = -201.8$; and for Figure 1.1(d) we obtain $\hat{\beta}_n^{\text{Theil}} = 0.14$ and $\hat{\theta}_n^{\text{Theil}} = -270.4$. Both sets of results are summarized in Table 1.4. Also provided in the table are the standard error values for each case. Notably, the values do not change by much but the s.e. of the intercept improves greatly over LSE.

For hypothesis testing in least squares estimation, we use the t-statistic as follows:

$$t\text{-value}(\hat{\beta}_n^{\text{LS}}) = \frac{\hat{\beta}_n^{\text{LS}} - 0}{\text{s.e}(\hat{\beta}_n^{\text{LS}})}. \tag{1.3.12}$$

Once the t-statistic is computed, it is used to determine the p-value for hypothesis testing; in particular, $\mathcal{H}_o : \beta = 0$ vs. $\mathcal{H}_A : \beta \neq 0$. The LSE p-values provided in the table are very small indicating that the null hypothesis, $\mathcal{H}_o$, can be rejected in favor of the alternate hypothesis, $\mathcal{H}_A$.

For the Theil's estimator, we can not use the same method as LSE for hypothesis testing since we do not have any information about the distribution of the median. Theil's estimator falls into a special category known as nonparametric statistics (Corder and Foreman, 2014). This is a branch of statistics in which the underlying distribution is not known so alternative methods are used.

Instead of the approach taken above, the Wilcoxon test can be utilized here. The test essentially calculates the difference between pairs of data and analyzes their differences to establish if they are significantly different from one another in a statistical sense. The step-by-step procedure for comparing two related samples using *Wilcoxon signed-rank test statistic*, T, based on Corder and Foreman (2014) is as follows:

(1) For each item in a sample of n items, obtain a difference score D_i between two measurements (i.e. $D_i = y_i - (\hat{\theta}_n^{\text{Theil}} + \hat{\beta}_n^{\text{Theil}} x_i)$ $i = 1, ..., n$).
(2) Take the absolute value of the differences, $|D_i|$.
(3) Omit difference scores of zero, giving you a set of n' non-zero absolute difference scores, where $n' \leq n$. Thus, n' becomes the actual sample size.
(4) Then, assign ranks R_i from 1 to n' to each of the $|D_i|$ such that the smallest gets rank 1 and the largest gets rank n'. If two or more $|D_i|$ are equal, they are all assigned same average rank (i.e. midrank) of the ranks they would have been assigned individually had ties not occurred.
(5) Next, apply the sign $+$ or $-$ to each of the n' ranks, R_i, depending on whether D_i was originally positive or negative, producing $+R_i$ or $-R_i$, respectively.
(6) The Wilcoxon test T-statistic is taken as the absolute sum of either the positive ranks, $T = |\sum_i (+R_i)|$, or negative ranks, $T = |\sum_i (-R_i)|$, whichever is *smaller*. Hence, T is always positive.

Next, the T-statistic can be examined for significance. As n' increases, the sampling distribution converges to a normal distribution. Thus, for $n' \geq 20$, a z-score can be calculated. We compute the z-scores for large samples using

Table 1.4 Comparison of LS and Theil estimations of Figures 1.1(a) and (d).

Case (LSE)	Estimators	Coefficient	s.e	*t*-value	*p*-value
Figure 1.1(a)	$\hat{\theta}_n^{\text{LS}}$	−234.2	23.4	−10.0	<0.001
	$\hat{\beta}_n^{\text{LS}}$	0.120	0.01	10.1	<0.001
Figure 1.1(d)	$\hat{\theta}_n^{\text{LS}}$	−983.9	325.2	−3.0	0.006
	$\hat{\beta}_n^{\text{LS}}$	0.504	0.166	3.0	0.006
Case (Theil)	**Estimators**	**Coefficient**	**s.e**	***T*-statistic**	z^*
Figure 1.1(a)	$\hat{\theta}_n^{\text{Theil}}$	−201.8	0.06	–	–
	$\hat{\beta}_n^{\text{Theil}}$	0.104	0.02	107.0	−1.23
Figure 1.1(d)	$\hat{\theta}_n^{\text{Theil}}$	−270.4	0.32	–	–
	$\hat{\beta}_n^{\text{Theil}}$	0.139	0.18	116.5	−0.96

$$\bar{x}_T = \frac{n'(n'+1)}{4}, \quad s_T = \sqrt{\frac{n'(n'+1)(2n'+1)}{24}}, \text{ and } z^* = \frac{T - \bar{x}_T}{s_T},$$

where $\bar{x}_T$ is the mean and s_T is the standard deviation, and z^* is the z-score for an approximation of the data to the normal distribution.

To perform a two-sided test, we reject $\mathcal{H}_o$ if $z_{\text{critical}} < |z^*|$. Specifically, for $\alpha = 0.05$, $z_{\text{critical}} = 1.96$. In the case of Figure 1.1(d), we have $n' = 24$ and obtain $T = 116.5$ (the negative sum is smaller), so $z^* = (116.5 - 150)/35 = -0.96$. Therefore, we cannot reject $\mathcal{H}_o$. This suggests there is no difference in the two samples. Similarly, for Figure 1.1(a), we find that $T = 107.0$ so that $z^* = (107.0 - 150)/35 = -1.23$. Again, we cannot reject $\mathcal{H}_o$ suggesting no difference in the two samples. Table 1.4 shows the results of this approach[4] on Theil's estimator for the telephone data set in the two cases of Figures 1.1(a) and 1.1(d).

1.4 Outliers and their Detection

Typically, outliers are created by errors in transcribing the data, or noise in the data collection process, or simply anomalies in data values. It is relatively easy to observe an outlier on a graph in a one-variable setting, but much more difficult to identify in the case of multiple linear regression. Hence, a clear

4 The results are based on the Python library scipy.stats.wilcoxon() that provides the T-statistic.

definition of an outlier and reliable methods for their detection are difficult to provide. One approach is to examine the residuals after analysis for both LSE and median-based methods to determine if outliers can be separated from the main data. Order statistics can also be used to identify outliers.

1.4.1 Outlier Detection

The residuals for simple linear regression can be computed after estimation using a general equation as follows

$$\hat{e}_i = y_i - (\hat{\theta}_n + \hat{\beta}_n x_i), \ i = 1, \dots, n, \tag{1.4.1}$$

where $\hat{\theta}_n$ and $\hat{\beta}_n$ are arbitrary estimators of θ and β, respectively. We note that if there is an outlier present in the residuals, $\{\hat{e}_1, \dots, \hat{e}_n\}$, at least one of the extreme order statistics $\hat{e}_{(1)}$ or $\hat{e}_{(n)}$ must be an outlier. Therefore, it is possible to identify outliers by their position in the order statistics. Regardless of its size, the outlier will always be positioned at one extreme or the other. In reality, multiple outliers may exist in the data set and the residuals associated with them would form a set of order statistics at one extreme or the other, or both. This residual ordering can be done for LSE and the median-based method and then compared.

Consider Figure 1.2 showing the computed residuals for both LSE and median-based cases on the original telephone data set of Table 1.3 and depicted in Figure 1.1(d). In the upper half of the figure, we present the histograms of

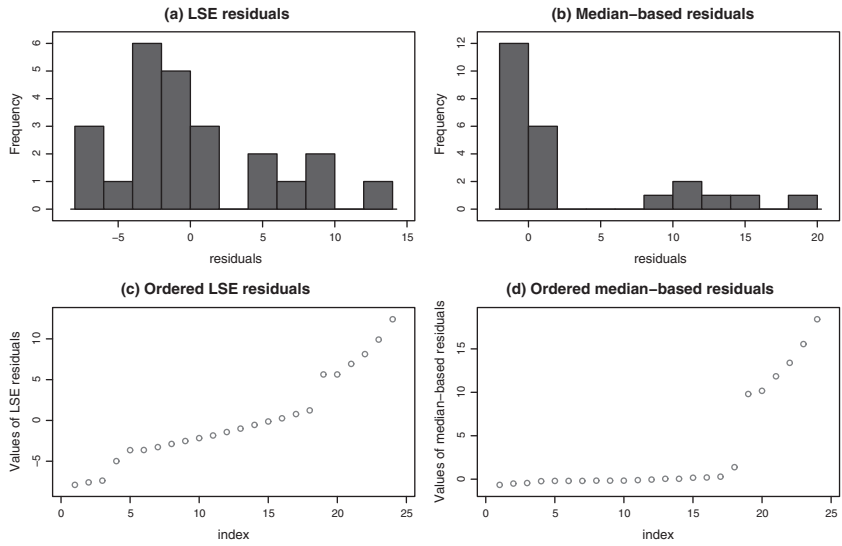

Figure 1.2 Histograms and ordered residual plots of LS and Theil estimators.

the computed residuals for the two cases. In the lower half, we plot the order statistics of each set of residuals from 1 to 24.

The LSE residuals form a distribution that makes it difficult to separate outliers from the rest of the data points. It is not easy to make a definitive statement about which bars on the histogram are associated with the six outliers. There appear to be outliers on both sides of 0 but one cannot be sure. On the other hand, the residuals for the median-based approach show a clear separation between the outliers and the rest of the data. A cutoff value of 5.0 provides a suitable dividing line to sift out the six outliers. Note that ordering does not help to separate the outliers from the data in the LSE case as shown in the ordered residuals below the histogram. In stark contrast, we see a clear distinction between the outliers and the rest of the data in the median case. Their respective order statistics are $\hat{e}_{(19)}, \ldots, \hat{e}_{(24)}$ since there are 24 points and the six outliers are large and positive.

These six data points may now be inspected to determine if they should be retained, modified or removed from the data set. Some points may be easily corrected if it was a manual entry mistake, or a measurement calibration error that can be fixed (as is the case here). The decision whether or not a data point is a true outlier is based on well-established rules or metrics for a given application, either based on, for example, 4 or 5 standard deviations away from the mean, or say, several quartiles away from the inner first and third quartiles (25% to 75% in terms of percentiles of the residuals). These approaches are prone to error and must be done carefully.

While this method of using order statistics is not foolproof, it shows promise and will be workable for higher dimension problems without modification. In general, it is difficult, error-prone and somewhat tedious to identify and remove outliers. A better approach is to use robust statistical procedures that are relatively insensitive to outliers so that they do not have to be detected or removed. A method using rank estimation is such an approach and is introduced in the next section.

1.5 Motivation for Rank-based Methods

1.5.1 Effect of a Single Outlier

As a preview of the robustness of rank-based methods, we show four cases in Figure 1.3 using the telephone data set to demonstrate that a single outlier can have deleterious effects on LSE but little or no effect on rank estimates, depending on its position relative to other observations in the data set. Outliers are generally viewed as nuisance observations in a data set and their effects are considered to be relatively benign so robust methods have not been widely embraced as yet. To provide a clear need for robust regression, we

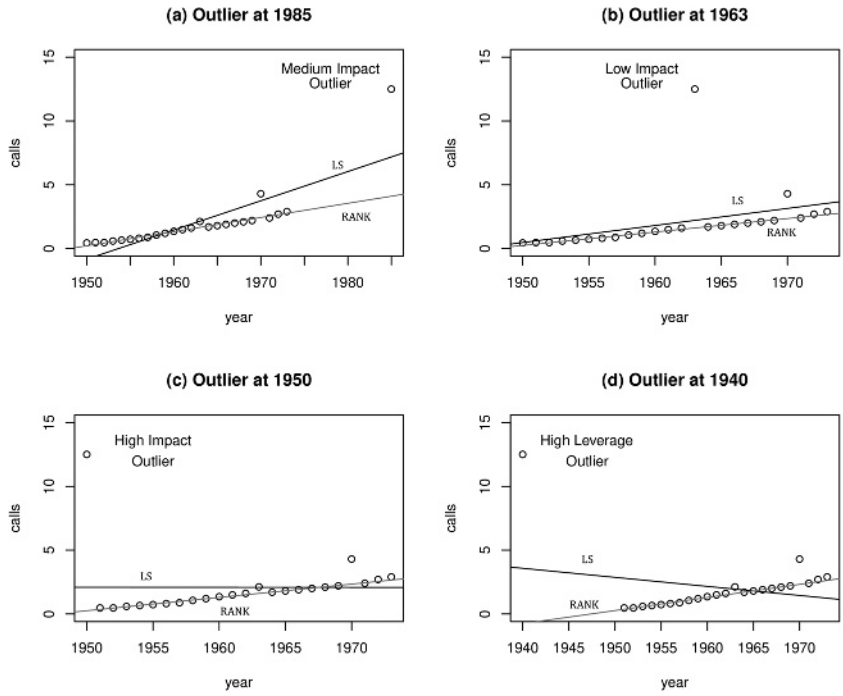

Figure 1.3 Effect of a single outlier on LS and rank estimators. The line due to rank is relatively unchanged compared to LS line.

provide 4 interesting cases but note that the results are specific to this data set. These examples should leave no doubt that robust methods should be used in mainstream machine learning tools, and that research in this area should be increased to resolve any issues that may limit their use.

Starting with Figure 1.3(a), we see that the insertion of one strategically placed outlier on the far right of the data set at (year = 1985, calls = 12.5), has a medium impact on the LSE result, relative to the other cases shown, while the rank result remains unaffected. However, when the outlier is moved to (year = 1963, calls = 12.5) as in Figure 1.3(b), the LSE and rank results are similar; hence, this is a low-impact outlier in this data set. It is only meant to show that in some cases, an obvious outlier may not influence the LSE result by much whereas the rank result remains the same as before.

Now consider the case of the outlier at (year = 1950, calls = 12.5) in Figure 1.3(c). This position constitutes a high-impact outlier for this data set. In fact, LSE tells us that there is no relationship between calls and year, which we know is incorrect. We will say more about this case shortly. In the last case, Figure 1.3(d), the added outlier at (year = 1940, calls = 12.5) is called a *high-leverage* point. This type of outlier can have serious consequences on the results of LSE.

In fact, it shows a negative correlation between year and calls which is clearly wrong. The point of all these examples is to illustrate the unpredictability of even a single outlier in determining the LSE, let alone a number of outliers and their effects. They show that not all outliers are created equal but the results of rank are quite robust in all four cases.

To emphasize the potential hazard of Figure 1.3(c), assume we had to select a small subset from 10 explanatory variables and year was just one of them. A result such as the high-impact case with a slope of 0 would suggest that we remove year as a variable of interest. Any method that relies on the LSE result to select a subset of explanatory variables would fail in this case. However, the rank estimate virtually ignores both the high-impact and high-leverage points and therefore would be a better basis for subset selection.

In essence, LSE makes an inherent assumption that if outliers exist, they are distributed uniformly in all directions. Of course, rank methods do not assume this but it would be a big limitation if rank methods were not accurate when outliers are balanced in all directions. Fortunately, the rank estimates are in line with LSE in the balanced case but provide robustness in the y-space for unbalanced cases. This is an important point to note if one were to abandon LSE in favor of rank methods.

Thus far, we have seen that using the median has certain barriers that have limited its use in the past, not least of which are more complicated expressions, the need for nonparametric analysis, and the cumbersome nature of the solution presented to this point. But given that Theil's method performed much better than LSE in the presence of outliers, other methods of obtaining the slope and intercept using the median deserves some attention. Before describing such methods, there are a number of drawbacks in Theil's method to review.

First, all pairwise slopes must be computed before the median is taken. Assuming n data points, this results in $n(n-1)/2$ slope calculations of the form $(y_j - y_i)/(x_j - x_i)$ before using Eq. (1.3.5). For our telephone example where $n = 24$, there are 276 slope calculations. Clearly this is a problem as n gets large since the computational complexity[5] grows according to $O(n^2)$. The number of partial-residuals of the form $(y_i - \hat{\beta}_n^{\text{Theil}} x_i)$ to be computed using Eq, (1.3.6) is $O(n)$, which is $n = 24$ in this case, so there is no issue here. Second, all of the points were spaced evenly on the x-axis which produced good accuracy using Theil; however, points are not generally evenly spaced. Therefore, accuracy could be reduced in uneven cases. And third, the process becomes unwieldy as the dimensionality of the problem increases. So far we have considered a simple

5 The computational complexity of an algorithm can be characterized using the $O(f(n))$ notation. For example, $O(n)$ refers to an algorithm that runs in linear time w.r.t. n, whereas $O(n^2)$ has a quadratic run time. Likewise, $O(2^n)$ has an exponential run time. A logarithmic run time would be represented as $O(\log(n))$, etc.

linear case ($p = 1$). However, if $p \gg 1$, this method is much more complicated to implement and the breakdown point decreases. All of these concerns must be addressed if LSE is to be replaced by a median-based method. Fortunately, rank-based methods will alleviate these issues.

1.5.2 Using Rank for the Location Model

To introduce rank-based methods, consider the location model

$$y_i = \theta + \varepsilon_i, \quad i = 1, \dots, n,$$

where ε_1, ..., ε_n are i.i.d. random variables. A location model attempts to estimate the center of a given sample, y_1, ..., y_n, of a random variable Y. This could be the sample mean or median or any other similar type of estimate depending on the underlying distribution. We will describe how to address this simple case to compare two similar estimators that seek the median.

In theory, the median is obtained using an L_1 loss function. To estimate the location parameter, we could use the absolute value loss function given by

$$B_n(\theta) = \sum_{i=1}^{n} |y_i - \theta|. \tag{1.5.1}$$

The median estimate, $\hat{\theta}_n^{\text{median}}$, is obtained as

$$\hat{\theta}_n^{\text{median}} = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \{B_n(\theta)\}. \tag{1.5.2}$$

The gradient function is

$$B_n'(\theta) = -\sum_{i=1}^{n} \operatorname{sgn}(y_i - \theta). \tag{1.5.3}$$

The absolute value loss function has a discontinuous first derivative that creates some theoretical problems since it does not satisfy certain regularity conditions.

Now consider multiplying the terms in the L_1 loss function by their respective ranks to form a rank-based loss function as follows

$$D_n(\theta) = \sum_{i=1}^{n} |y_i - \theta| \, R_{n_i}^{+}(\theta), \tag{1.5.4}$$

where $R_{n_i}^{+}(\theta)$ are the ranks of $|y_i - \theta|$ among $|y_1 - \theta|$, ... , $|y_n - \theta|$. The "+" sign indicates that we are taking the absolute value of the quantities before ranking them. Note that, in the new loss function, we are simply multiplying the absolute value of $y_i - \theta$ by its rank, $R_{n_i}^{+}(\theta)$. By doing so, we have a new

loss function that produces the median of the pairwise averages of y_i. The pairwise averages of a set of numbers, including the original numbers themselves, are referred to as the Walsh averages. In effect, we obtain the median of the Walsh averages using this rank-based loss function. The result is equivalent to the Hodges–Lehmann (HL) estimator (cf. Hettmansperger and McKean, 2011) given by:

$$\hat{\theta}_n^{\mathrm{HL}} = \underset{1\le i\le j\le n}{\text{median}} \left\{ \frac{y_i + y_j}{2} \right\}.$$

We will find it convenient to normalize the ranks by defining $u = R_{n_i}^+(\theta)/(n+1)$ such that $0 < u < 1$, and then use it to define a scoring function $\phi^+(u) = u$. This is a specific scoring function that we have chosen here. Since a variety of other scoring functions are possible, we use the notation, $a_n^+(R_{n_i}^+(\theta))$, for a generic score function and embed the actual function $\phi^+(u)$ inside this generic function. Combining the above, we define the rank dispersion function as follows,

$$D_n(\theta) = \sum_{i=1}^{n} |y_i - \theta|\, a_n^+(R_{n_i}^+(\theta)), \tag{1.5.5}$$

with the corresponding gradient given by

$$D_n'(\theta) = -\sum_{i=1}^{n} \mathrm{sgn}(y_i - \theta)\, a_n^+(R_{n_i}^+(\theta)). \tag{1.5.6}$$

The rank estimate is

$$\hat{\theta}_n^{\mathrm{R}} = \underset{\theta\in\mathbb{R}}{\text{argmin}} \{D_n(\theta)\}. \tag{1.5.7}$$

All we have done thus far is to create a new loss function by applying a weight consisting of a normalized rank to the L_1 loss function. Instead of obtaining the median, it produces the median of the Walsh averages. Although the median is a robust estimator of the location, we will find that the median of the Walsh averages is also a robust estimator but with a lower breakdown point. Note that the median is the 50th percentile of the data, but any estimator that provides a result that is in the vicinity of the median is also robust.

To illustrate the difference using an example, consider the set,

$$\boldsymbol{y} = (0.1, 1.2, 2.3, 3.4, 4.5, 5.0, 6.6, 7.7, 8.8, 9.9, 10.5)^\top.$$

We seek to estimate the location parameter, θ. In doing so, we can minimize Eq. (1.5.1) or Eq. (1.5.4). Which one should we choose? To answer this question, let us first examine their derivatives as a function of θ. As shown in Figure 1.4,

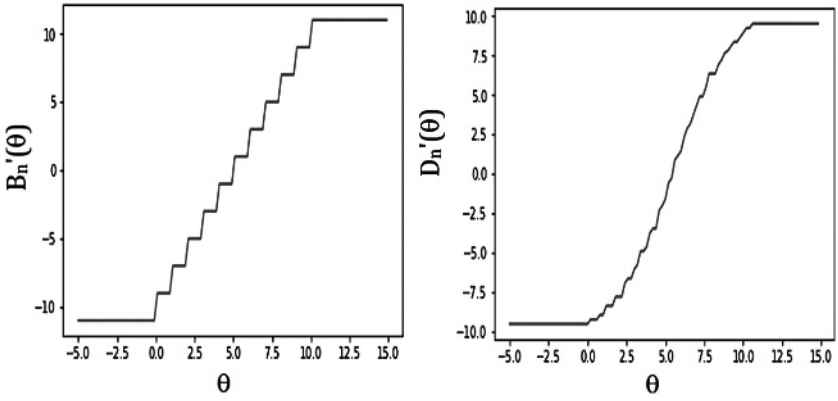

Figure 1.4 Gradients of absolute value ($B'_n(\theta)$) and dispersion ($D'_n(\theta)$) functions.

the gradient of the absolute value loss function, $B'_n(\theta)$, is a series of steps. Each step occurs at a different value of y_i. Hence, the number of steps is n which in this case is 11. However, the gradient of dispersion function, $D'_n(\theta)$, is similar in nature but smoothed out by the scoring function $a_n^+(\cdot)$ with more steps generated in the same interval. Each step occurs at values associated with each one of the Walsh averages. As a result, the number of steps is $\frac{n(n+1)}{2}$ which is 66 in this case, as shown in the graph on the right. Under the assumptions listed earlier, it is possible to show that the estimator of Eq. (1.5.7) is consistent and asymptotically normal (Hettmansperger and McKean, 2011) whereas the estimator of Eq. (1.5.2) does not have these desirable properties. This is why we should choose to minimize Eq. (1.5.4).

Of course, when we minimize $D_n(\theta)$ and $B_n(\theta)$, we do not obtain the same results. From inspection of the numbers, it is clear that the median is 5.0 which is obtained by minimizing $B_n(\theta)$. However, when we minimize $D_n(\theta)$ we obtain 5.55, which is not the median of the data set. As mentioned above, $D_n(\theta)$ produces the median of the Walsh averages, i.e. the pairwise averages of the given numbers. The Walsh averages for this data set are provided in Table 1.5 in sorted order. We can count a total of 66 averages, some of which are duplicated. As a result, there are a maximum of 66 steps in the gradient function, but if there are duplicates in the Walsh averages, the number of steps can be below 66. If we select the median of the Walsh averages from the table, we would select the average of the 33rd and 34th values which is 5.55 (shown in bold).

Although we do not obtain the actual median, the rank-based approach still provides both robustness and asymptotic normality. From Figure 1.4, we can intuitively understand why the dispersion function leads to the regularity

Table 1.5 Walsh averages for the set {0.1, 1.2, 2.3, 3.4, 4.5, 5.0, 6.6, 7.7, 8.8, 9.9, 10.5}.

0.10	0.65	1.20	1.20	1.75	1.75	2.30	2.30	2.30	2.55	2.85	2.85
3.10	3.35	3.40	3.40	3.65	3.90	3.90	3.95	4.20	4.45	4.45	4.45
4.50	4.75	5.00	5.00	5.00	5.00	5.00	5.30	**5.55**	**5.55**	5.55	5.55
5.80	5.85	6.10	6.10	6.10	6.35	6.40	6.60	6.65	6.65	6.90	6.95
7.15	7.20	7.45	7.50	7.70	7.70	7.75	8.25	8.25	8.55	8.80	8.80
9.10	9.35	9.65	9.90	10.20	10.50	—	—	—	—	—	—

conditions that we seek in an estimator and why it inherits the robustness property similar to the median, while not actually computing the true median. The asymptotic properties of the linear rank-based estimators and their relative efficiencies can be found in Puri and Sen (1985) for the interested reader.

1.5.3 Using Rank for the Slope

If we now wish to define a linear rank estimate of the slope for the simple linear model, we can use another formulation of the dispersion function:

$$D_n(\beta) = \sum_{i=1}^{n}(y_i - \beta x_i)a_n(R_{n_i}(\beta))$$

where $R_{n_i}(\beta) = R_{n_i}(y_i - \beta x_i)$ is the rank of $(y_i - \beta x_i)$ among $(y_1 - \beta x_1), \ldots, (y_n - \beta x_n)$, and $a_n(R_{n_i}(\beta))$ is a different scoring function which uses $\phi(u) = u - 1/2$, with $u = R_{n_i}(\beta)/(n+1)$ as before. Note that we do not use the "+" notation here since absolute values of the quantities are not taken. The minimization of this dispersion function does not typically produce the median of the slopes as was the case in Theil's estimator. Rather, it produces an estimate in the vicinity of the median. The actual quantile associated with the estimate depends on the data. The median is, of course, at the 50th percentile but we can retain some of the robustness properties (albeit at a lower breakdown point) by selecting a slope near the 50th percentile. This is best illustrated using another example.

Consider the following data set:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.2
y	3.2	4.0	4.2	4.7	6.5	5.5	6.7	20.2	22.0	8.0

If we use a simple linear model, then we seek an estimate of the slope, β. Since $n = 10$, there are a total of $n(n-1)/2 = 45$ slopes to compute. These ordered slopes are shown in the table below. The median is the 23rd entry which is 6.25.

However, the rank estimate, obtained by minimizing $D_n(\beta)$, is 5.40 which is the 21st entry (in bold). It is at the 46th percentile.

1	−46.67	−30.50	−10.00	1.00	2.00	2.14	2.60	3.50	3.75	4.00	4.00	4.13
13	4.17	4.22	4.33	4.36	4.60	5.00	5.00	5.00	**5.40**	5.83	6.25	6.67
25	8.00	8.25	8.33	11.50	12.00	18.00	18.00	23.5	24.29	25.71	27.00	29.67
37	32.00	34.60	38.75	38.75	45.67	55.00	73.50	76.50	135.00	—	—	—

Therefore, while not computing the median of the slopes, the rank-based method produces an estimate that is near the median. The benefits of using a median-based approach such as rank should be clear at this point. We will show how we can perform linear regression using rank that parallels the methodology of the least squares estimator but provides robustness. Although Theil's method is effective, we will see that rank-based methods provide a more efficient solution and they are the basis for the theory and methods to be described in the rest of this book.

1.6 The Rank Dispersion Function

In this section, we pursue a formal approach inspired by the robustness of the median in linear regression. We refer to this approach as a rank-based method in which values in the vicinity of the median are used for estimation. It would be unwieldy and cumbersome to use Theil's method to perform linear regression due to its somewhat ad hoc nature. Fortunately, a procedure was developed in the early 1970s in the work of Jurečková (1971) and Jaeckel (1972) that produces similar results, does not require the calculation of $O(n^2)$ slopes, provides for outlier detection, and extends easily to higher dimensions, i.e. multiple linear regression.

Recall that in LS estimation, we minimize a quantity we call the *quadratic* or L_2 loss function,

$$\text{SSE} = \sum_{i=1}^{n} (y_i - (\theta + \beta x_i))^2, \tag{1.6.1}$$

which is the sum of the squares of the error (SSE). Jaeckel (1972) developed a loss function which allows for rank-based linear regression. The rank estimator requires simply replacing the quadratic loss function with Jaeckel's dispersion function and then minimizing it for R-estimation (i.e. rank-based estimation). The dispersion function can be viewed either as a pseudo-norm, as noted in Kloke and McKean (2012), or a weighted sum of the residuals. The pseudo-norm designation is due to the fact that it behaves somewhat like the

well-known L_1-norm. As before, we will first compute the slope and use it to compute the intercept. For the slope, the dispersion function is given by

$$D_n(\beta) = \sum_{i=1}^{n}(y_i - (\theta + \beta x_i))\, a_n(R_{n_i}(y_i - (\theta + \beta x_i)). \tag{1.6.2}$$

This function requires some explanation. The first term is the residual itself. Note that it is a function of θ and β but we seek only β at this stage. The term $R_{n_i}(y_i - (\theta + \beta x_i))$ is the rank of the ith residual. However, the intercept θ is a constant and, as such, it will not change the rankings. That is, a constant added to a set of values will not change their relative ranks. Therefore, we can remove θ from the ranking function, i.e. we simply rank the partial-residuals

$$R_{n_i}(\beta) = R_{n_i}(y_i - \beta x_i).$$

Furthermore, we will see shortly that

$$\sum_{i=1}^{n} a_n(R_{n_i}(\beta)) = 0. \tag{1.6.3}$$

The term $a_n(\cdot)$ is a weighting function for the residuals, and contains a scoring function. Since this function sums to zero, the dispersion function is invariant w.r.t. θ. In particular, we can show that

$$\begin{aligned} D_n(\theta, \beta) &= \sum_{i=1}^{n}(y_i - (\theta + \beta x_i))a_n(R_{n_i}(y_i - \beta x_i)) \\ &= \sum_{i=1}^{n}(y_i - \beta x_i)a_n(R_{n_i}(y_i - \beta x_i)) - \theta \sum_{i=1}^{n} a_n(R_{n_i}(y_i - \beta x_i)) \\ &= \sum_{i=1}^{n}(y_i - \beta x_i)a_n(R_{n_i}(y_i - \beta x_i)) = D_n(\beta), \end{aligned} \tag{1.6.4}$$

which we can write compactly as

$$D_n(\beta) = \sum_{i=1}^{n}(y_i - \beta x_i)a_n(R_{n_i}(\beta)). \tag{1.6.5}$$

It can be shown that $D_n(\beta)$ is a non-negative, piece-wise linear, continuous, convex function of β and therefore is quite suitable to quickly obtain an estimate using numerical optimization routines. This was the essential breakthrough that made rank-based methods workable. Furthermore, the solution to Eq. (1.6.5) is consistent and satisfies certain asymptotic normality conditions

(Hettmansperger and McKean, 2011). In addition, the rank estimate is invariant to whether the data is centered or uncentered. Therefore, it is important to consider the intercept in rank methods.

The rank estimate $\hat{\beta}_n^{\mathrm{R}}$ can be found by minimizing the function as follows

$$\hat{\beta}_n^{\mathrm{R}} = \underset{\beta \in \mathbb{R}}{\operatorname{argmin}}\ D_n(\beta). \tag{1.6.6}$$

Another alternative to obtaining the optimal value for β is to take the derivative,

$$\frac{\partial D_n(\beta)}{\partial \beta} = -\sum_{i=1}^{n} x_i a_n(R_{n_i}(\beta)), \tag{1.6.7}$$

and set it to zero. Therefore, an estimate can be obtained for β from $\partial D_n(\beta)/\partial\beta = 0$ and produces the same $\hat{\beta}_n^{\mathrm{R}}$ as above. In this case, a root-finding method is needed to solve the problem.

Let us now define

$$L_n(\beta) = -\frac{1}{\sqrt{n}} \frac{\partial D_n(\beta)}{\partial \beta} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_i a_n(R_{n_i}(\beta)). \tag{1.6.8}$$

Then, a gradient test for $\mathcal{H}_o : \beta = 0$ vs. $\mathcal{H}_A : \beta \neq 0$ can be developed by defining

$$L_n(0) = -\frac{1}{\sqrt{n}} \frac{\partial D_n(\beta)}{\partial \beta}\bigg|_{\beta=0}.$$

The next step is to estimate the intercept θ given that we know $\hat{\beta}_n^{\mathrm{R}}$. This requires another dispersion function that uses the absolute values of the residuals as follows:

$$D_n(\theta) = \sum_{i=1}^{n} |y_i - (\theta + \hat{\beta}_n^{\mathrm{R}} x_i)|\, a_n^+(R_{n_i}(|y_i - (\theta + \hat{\beta}_n^{\mathrm{R}} x_i)|)), \tag{1.6.9}$$

which we can write compactly as

$$D_n(\theta) = \sum_{i=1}^{n} |y_i - (\theta + \hat{\beta}^{\mathrm{R}} x_i)|\, a_n^+(R_{n_i}^+(\theta)), \tag{1.6.10}$$

where

$$R_{n_i}^+(\theta) = R_{n_i}(|y_i - (\theta + \hat{\beta}_n^{\mathrm{R}} x_i)|).$$

Note that all terms are included here since the invariance property no longer holds. In addition, $R_{n_i}^+(\cdot)$ is the rank of the absolute values of the residuals, and $a_n^+(\cdot)$ uses a different scoring function. From this function, the intercept is obtained as follows

$$\hat{\theta}_n^{\mathrm{R}} = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} D_n(\theta). \tag{1.6.11}$$

As before, an alternative to obtaining $\hat{\theta}_n^{\mathrm{R}}$ is to take the derivative and set it to zero, i.e. $\partial D_n(\theta)/\partial\theta = 0$, where

$$\frac{\partial D_n(\theta)}{\partial \theta} = -\sum_{i=1}^{n} \operatorname{sgn}(y_i - (\theta + \hat{\beta}_n^{\mathrm{R}} x_i)) a_n^+(R_{n_i}^+(\theta)). \tag{1.6.12}$$

We can define the function $T_n(\theta)$ as

$$T_n(\theta) = -\frac{1}{\sqrt{n}} \frac{\partial D_n(\theta)}{\partial \theta} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \operatorname{sgn}(y_i - (\theta + \hat{\beta}_n^{\mathrm{R}} x_i)) a_n^+(R_{n_i}^+(\theta)). \tag{1.6.13}$$

A gradient test for $\mathcal{H}_o : \theta = 0$ vs. $\mathcal{H}_A : \theta \neq 0$ can be developed by defining

$$T_n(0) = -\frac{1}{\sqrt{n}} \frac{\partial D_n(\theta)}{\partial \theta}\bigg|_{\theta=0}.$$

1.6.1 Ranking and Scoring Details

The ranking and scoring functions will now be elaborated. We first want to solve the linear regression problem by minimizing the dispersion function, $D_n(\beta)$, w.r.t. β. The ranking function, $R_i = R(y_i - \beta x_i)$ ranks each of the partial-residuals $(y_i - \beta x_i)$ among $(y_1 - \beta x_1, \ldots, y_n - \beta x_n)$. For convenience, we use $R_{n_i}(\beta)$ to refer to the rank of the ith partial-residual. The rank itself will lie in the set $\{1, \cdots, n\}$ which means that its range is dependent on the number of observations, n. To avoid this, the rank is normalized by a factor of $(n+1)$ so that $0 < R_{n_i}(\beta)/(n+1) < 1$. The weighting component, $a_n(\cdot)$ in $D_n(\beta)$, involves a scoring function which is a function of these normalized ranks. The normalized rank values applied to a scoring function can be defined in either of the following ways

$$a_n(i) = \mathbb{E}\left[\phi\left(U_{n_i}\right)\right], \quad i = 1, \ldots, n \tag{1.6.14}$$

where $U_{n_1} \leq U_{n_2} \leq \cdots \leq U_{n_n}$ are the ordered statistics corresponding to the sample size of n from the $U(0,1)$ distribution; or

$$a_n(R_{n_i}(\beta)) = \phi\left(\frac{R_{n_i}(\beta)}{n+1}\right). \tag{1.6.15}$$

Here, $\phi(u) = 2u - 1$, $0 < u < 1$ is a non-decreasing, square-integrable scoring function. Without loss of generality, standardized scores can be defined such that

(1) $\int_0^1 \phi(u)\mathrm{d}u = 0$, i.e. the integral in [0,1] is 0

(2) $\int_0^1 \phi^2(u)\mathrm{d}u = 1$, i.e. the square integral in [0,1] is 1
(3) $\phi(1-u) = -\phi(u)$, i.e. it is skew-symmetric about $\frac{1}{2}$.

The Wilcoxon score function, given by

$$\phi(u) = \sqrt{12}\left(u - \frac{1}{2}\right), \tag{1.6.16}$$

satisfies the above conditions, since clearly

$$\int_0^1 \sqrt{12}\left(u - \frac{1}{2}\right)\mathrm{d}u = \sqrt{12}\left(\frac{u^2}{2} - \frac{u}{2}\right|_0^1 = 0,$$

and

$$\int_0^1 12\left(u - \frac{1}{2}\right)^2 \mathrm{d}u = \int_0^1 12\left(u^2 - u + \frac{1}{4}\right)\mathrm{d}u = 12\left(\frac{u^3}{3} - \frac{u^2}{2} + \frac{u}{4}\right|_0^1 = 1,$$

and

$$\phi(1-u) = \sqrt{12}\left(1 - u - \frac{1}{2}\right) = -\sqrt{12}\left(u - \frac{1}{2}\right) = -\phi(u).$$

In order to compute $\hat{\theta}_n^{\mathrm{R}}$, we minimize $D_n(\theta)$ of Eq. (1.6.10) which involves $R^+(\cdot)$ and $a_n^+(\cdot)$. The function $R_{n_i}^+(\cdot)$ ranks $|y_i - (\theta + \beta x_i)|$ among $|y_1 - (\theta + \beta x_1)|, \ldots, |y_n - (\theta + \beta x_n)|$. The corresponding signed-rank scores are generated using one of the following two ways

$$a_n^+(i) = \mathbb{E}\left[\phi^+\left(U_{n_i}\right)\right], \quad i = 1, \ldots, n \tag{1.6.17}$$

where the ordered $U_{n_i} \sim U(0,1)$, for $i = 1, \ldots, n$; or

$$a_n^+(R_{n_i}^+(\beta)) = \phi^+\left(\frac{R_{n_i}^+(\beta)}{n+1}\right), \tag{1.6.18}$$

where the function,

$$\phi^+(u) = \phi\left(\frac{u+1}{2}\right), \tag{1.6.19}$$

can be found from considering Wilcoxon score function of Eq. (1.6.16) to be

$$\phi^+(u) = \sqrt{12}\left(\frac{u+1}{2} - \frac{1}{2}\right) = \sqrt{3}u. \tag{1.6.20}$$

The plots for the standardized $\phi(u)$ and $\phi^+(u)$ are provided in Figure 1.5. We see that both functions are linear with respect to u, with the range $\frac{1}{n+1} < u < \frac{n}{n+1}$. From these plots, we can understand that $\sum_{i=1}^n a_n(\cdot) = 0$ but $\sum_{i=1}^n a_n^+(\cdot) \neq 0$. The residuals are weighted by these two scoring functions to form the loss functions.

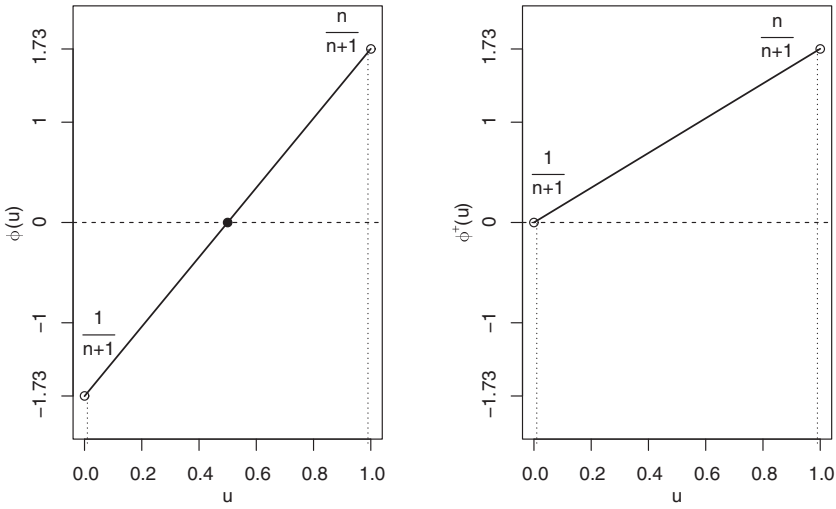

Figure 1.5 Scoring functions $\phi(u) = \sqrt{12}(u - 0.5)$ and $\phi^+(u) = \sqrt{3}u$.

The loss functions are then minimized to obtain the estimates. Both of these functions will be used for R-estimation in the next section.

1.6.2 Detailed Procedure for R-estimation

For demonstration purposes, we will now use the dispersion function to numerically obtain the R-estimations of θ and β for the original Belgium telephone data set (see Table 1.3 which is plotted in Figure 1.1(d)). We will detail the steps for numerical calculations and then show the results graphically to provide the complete picture. We first seek to estimate β using Eq. (1.6.5). The estimator $\hat{\beta}_n^{\mathrm{R}}$ is given by

$$\hat{\beta}_n^{\mathrm{R}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta x_i) \sqrt{12} \left(\frac{R_{n_i}(\beta)}{n+1} - \frac{1}{2} \right) \right\},$$

where $R_{n_i}(\beta) = R_{n_i}(y_i - \beta x_i)$. One can use a statistical software, e.g., the function `optim` of the package `CVXR` in R, to numerically solve this nonlinear optimization problem. Here, we will show how to do this by hand so that a deeper understanding can be obtained.

Clearly, we need to begin by picking a number of values for β to proceed with the solution to this problem until we find the minimum. For this purpose, we consider the three somewhat arbitrary values for β: $\beta^{(0)} = 0$, $\beta^{(1)} = 0.145$, and $\beta^{(2)} = 0.2$, knowing the minimum will likely be in this range.

The calculations to obtain $D_n(\beta)$ are reported in Table 1.6. We show column values for x, y, R, a_n and D_n as defined in the caption. There is one row in this table for each data point in the telephone data set. The D_n column is to be summed for each of the $\beta^{(0)}$, $\beta^{(1)}$, and $\beta^{(2)}$ cases, as required by Eq. (1.6.5). Thus, among these three values, the solution, i.e. $\hat{\beta}_n^{\mathrm{R}}$, is the one that gives the minimum value of "sum" in the last row of Table 1.6. In this example, $\hat{\beta}_n^{\mathrm{R}} = \beta^{(1)} = 0.145$ which will be confirmed later. But, more generally, determining the value of β by hand to find a global solution to the optimization problem is not an easy task and, as mentioned before, we need to use software to numerically find the minimum value. The detailed step-by-step procedure given here is meant to solidify our understanding of Eq. (1.6.5).

Equivalently, instead of the dispersion function, $D_n(\beta)$, one can use the root of its derivative, $L_n(\beta)$, Eq. (1.6.8), as

$$\hat{\beta}_n^{\mathrm{R}} = \left\{\beta : L_n(\beta) = -\frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_i \sqrt{12} \left(\frac{R_{n_i}(\beta)}{n+1} - \frac{1}{2} \right) = 0 \right\}.$$

The function `uniroot.all` from package *rootSolve* in R can be used here. We provide the required values in Table 1.6 in the column labeled L_n. Here we seek the "sum" that is closest to zero. Again, the zero-crossing occurs at $\hat{\beta}^{\mathrm{R}} = \beta^{(1)} = 0.145$.

We have shown two methods for R-estimation of β. Typically, we use $D_n(\beta)$ to compute the estimate, and $L_n(\beta)$ for hypothesis testing. It is instructive to examine a graphical equivalent of the procedure outlined above. This is shown in Figure 1.6. Consider the two panels on the left. The top panel shows the graph of $D_n(\beta)$ which is clearly a convex function of β. We note that the minimum is shown to be at 0.145. Similarly, the bottom panel is the function $L_n(\beta)$. We see the zero crossing of this function at 0.145 also, with 276 steps, one for each slope. The values of these two graphs can be matched with the data for $\beta^{(0)}$, $\beta^{(1)}$, and $\beta^{(2)}$ in Table 1.6.

Now we turn to the rank estimation of θ using Eq. (1.6.10). In this regard, based on the related Wilcoxon score in Eq. (1.6.20), it can be estimated as

$$\hat{\theta}_n^{\mathrm{R}} = \operatorname{argmin} \left\{ \sum_{i=1}^{n} \left| y_i - (\theta + \hat{\beta}_n^{\mathrm{R}} x_i) \right| \sqrt{3} \left(\frac{R_{n_i}^{+}(\theta)}{n+1} \right) \right\},$$

where

$$R_{n_i}^{+}(\theta) = R_{n_i}(|y_i - (\theta + \hat{\beta}_n^{\mathrm{R}} x_i)|).$$

Here, we are given that $\hat{\beta}_n^{\mathrm{R}} = 0.145$ for the telephone data set so the only unknown is θ.

Table 1.6 The individual terms that are summed in $D_n(\beta)$ and $L_n(\beta)$ for the telephone data set using initial values $\beta^{(0)} = 0$, $\beta^{(1)} = 0.145$, and $\beta^{(2)} = 0.2$. Here, $R = R_{n_i}(\beta)$, $a_n = a_{n_i}(R)$, $D_n = (y - \beta\, x)a_n$, $L_n = xa_n/\sqrt{n}$.

		$\beta^{(0)} = 0$				$\beta^{(1)} = 0.145$				$\beta^{(2)} = 0.2$			
x	y	R	a_n	D_n	L_n	R	a_n	D_n	L_n	R	a_n	D_n	L_n
1950	0.44	1	−1.59	−0.70	−634.27	17	0.62	−177.13	248.19	18	0.76	−296.88	303.35
1951	0.47	2.50	−1.39	−0.65	−551.83	16	0.48	−137.82	193.14	16	0.48	−189.01	193.14
1952	0.47	2.50	−1.39	−0.65	−552.11	14	0.21	−59.10	82.82	15	0.35	−135.08	138.03
1953	0.59	4	−1.18	−0.69	−469.53	13	0.07	−19.70	27.62	14	0.21	−81.06	82.86
1954	0.66	5	−1.04	−0.69	−414.51	12	−0.07	19.71	−27.63	13	0.07	−27.03	27.63
1955	0.73	6	−0.90	−0.66	−359.42	11	−0.21	59.13	−82.94	12	−0.07	27.04	−27.65
1956	0.81	7	−0.76	−0.62	−304.28	10	−0.35	98.58	−138.31	11	−0.21	81.14	−82.99
1957	0.88	8	−0.62	−0.55	−249.09	4	−1.18	335.25	−470.49	9	−0.48	189.39	−193.73
1958	1.06	9	−0.48	−0.51	−193.83	9	−0.48	138.03	−193.83	8	−0.62	243.52	−249.21
1959	1.20	10	−0.35	−0.42	−138.52	7	−0.76	216.91	−304.75	7	−0.76	297.68	−304.75
1960	1.35	11	−0.21	−0.28	−83.16	8	−0.62	177.47	−249.47	6	−0.90	351.85	−360.34
1961	1.49	12	−0.07	−0.10	−27.73	6	−0.90	256.35	−360.53	5	−1.04	406.04	−415.99
1962	1.61	13	0.07	0.11	27.75	5	−1.04	295.81	−416.20	4	−1.18	460.27	−471.70
1963	2.12	14	0.21	0.44	83.28	15	0.35	−98.48	138.81	10	−0.35	135.27	−138.81
1964	11.90	19	0.90	10.72	361.08	19	0.90	−247.37	361.08	19	0.90	−343.06	361.08
1965	12.40	20	1.04	12.89	416.84	20	1.04	−285.05	416.84	20	1.04	−395.53	416.84
1966	14.20	21	1.18	16.72	472.66	21	1.18	−321.11	472.66	21	1.18	−446.38	472.66
1967	15.90	22	1.32	20.93	528.53	22	1.32	−356.85	528.53	22	1.32	−496.93	528.53
1968	18.20	23	1.45	26.48	584.47	23	1.45	−391.27	584.47	23	1.45	−546.18	584.47
1969	21.20	24	1.59	33.78	640.45	24	1.59	−423.99	640.45	24	1.59	−593.73	640.45
1970	4.30	18	0.76	3.28	306.46	18	0.76	−215.77	306.46	17	0.62	−242.99	250.74
1971	2.40	15	0.35	0.83	139.37	1	−1.59	454.41	−641.11	1	−1.59	624.33	−641.11
1972	2.70	16	0.48	1.31	195.22	2	−1.45	414.67	−585.65	2.50	−1.39	542.76	−557.77
1973	2.90	17	0.62	1.81	251.12	3	−1.32	375.11	−530.15	2.50	−1.39	542.76	−558.05
Sum				122.78	28.95			107.78	0.0			108.19	−2.32

As before, consider three selected values for θ as $\theta^{(0)} = -285$, $\theta^{(1)} = -283$, $\theta^{(2)} = -280$. We already know that this is the proper range for the solution so these are suitable choices. Table 1.7 provides the calculations for these values. Again, we show column values for x, y, R^+, a_n^+ and D_n as defined in the caption. If we sum the D_n columns for each case, we arrive at the totals at the bottom of the table. We note that 124 is the smallest value and therefore we choose $\hat{\theta}^{\mathrm{R}} = \theta^{(1)} = -283.0$ as the solution based only on the three choices given.

It is possible to find the root of $T_n(\theta)$ in Eq. (1.6.13) rewritten as

$$\hat{\theta}_n^{\mathrm{R}} = \left\{\theta : T_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{sgn}(y_i - (\theta + \hat{\beta}_n^{\mathrm{R}} x_i))\sqrt{3}\left(\frac{R_{n_i}^+(\theta)}{n+1}\right) = 0\right\}.$$

In this regard, we provide the columns labeled T_n in Table 1.7 for each θ. This time we are interested in the column sum that is closest to zero. Of course, it is the same column as before and we find that $\hat{\theta}^{\mathrm{R}} = -283.0$. For a graphical

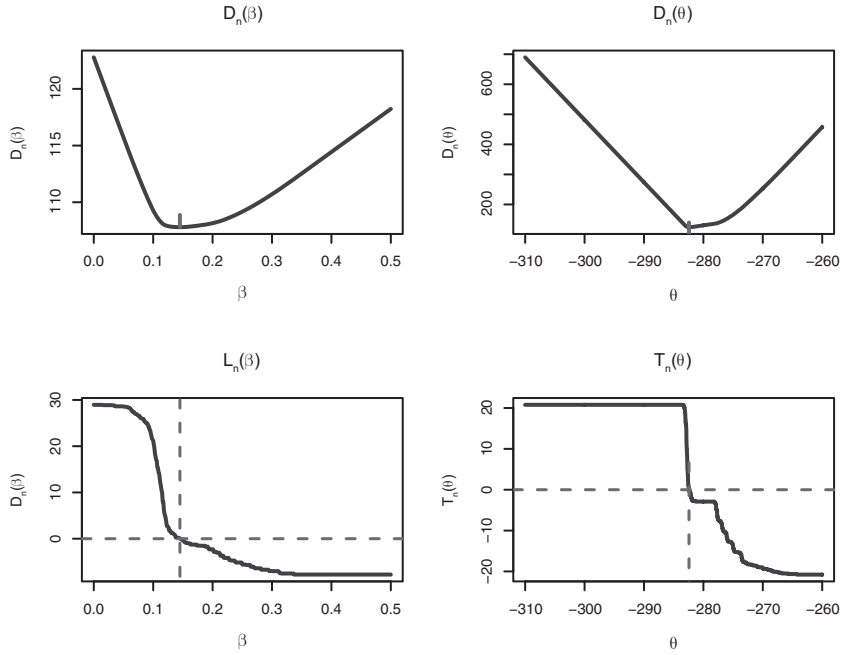

Figure 1.6 Dispersion functions and derivative plots for Figure 1.1(d).

equivalent of this process, we can return to Figure 1.6 and examine the panels on the right. We see the plot for $D_n(\theta)$ at the top and the minimum of the function is at -283.0 as expected. The "sum" values in Table 1.7 can be validated using this graph. Similarly, the bottom panel for $T_n(\theta)$ is the derivative plot with 24 steps and has a zero crossing at -283.0. This completes the detailed calculations for the original telephone data set.

Using the procedure outlined above, the R-estimates for all four cases shown in Figure 1.1 can be computed. This was done in Python using built-in optimization routines and is provided in Table 1.8 along with the LS results. We validated the results in R using Rfit (Kloke and McKean, 2012) which performs rank-based linear regression using the techniques described here. The slope and intercept R-estimates are only marginally affected by the outliers. On the other hand, large changes occur in the LS estimates in Table 1.8 for the corresponding cases. In addition, the rank estimates are similar to Table 1.4 which was based on Theil's method.

If we return to in Figure 1.6 which graphically illustrates the procedure to obtain estimates, we observe that $D_n(\beta)$ and $D_n(\theta)$ are both convex functions. This is a key feature of the dispersion function that allows it to be used to find the estimates using optimization routines. And we now have a formal

Table 1.7 The terms that are summed in $D_n(\theta)$ and $L_n(\theta)$ for the telephone data set using initial values $\theta^{(0)} = -285$, $\theta^{(1)} = -283$, and $\theta^{(2)} = -280$. Here, $R^+ = R^+_{n_i}(\theta)$, $a_n^+ = a^+_{n_i}(R^+)$, $D_n = |y - \theta - \hat{\beta}^{\mathrm{R}}\, x|\; a_n^+$, $L_n = \mathrm{sgn}(y - \theta - \hat{\beta}^{\mathrm{R}}\, x) a_n^+ / \sqrt{n}$.

		$\theta^{(0)} = -285$				$\theta^{(1)} = -283$				$\theta^{(2)} = -280$			
x	y	R^+	a_n^+	D_n	L_n	R^+	a_n^+	D_n	L_n	R^+	a_n^+	D_n	L_n
1950	0.44	17	1.18	7.46	0.24	3	0.21	0.02	0.04	2	0.14	0.51	−0.03
1951	0.47	16	1.11	6.89	0.23	1	0.07	0	−0.01	3	0.21	0.80	−0.04
1952	0.47	14	0.97	5.88	0.20	4	0.28	0.04	−0.06	5	0.35	1.38	−0.07
1953	0.59	13	0.90	5.43	0.18	5	0.35	0.06	−0.07	6	0.42	1.66	−0.09
1954	0.66	12	0.83	4.95	0.17	6	0.42	0.10	−0.09	7	0.48	1.94	−0.10
1955	0.73	11	0.76	4.47	0.16	7	0.48	0.16	−0.10	8	0.55	2.26	−0.11
1956	0.81	10	0.69	4.02	0.14	8	0.55	0.21	−0.11	9	0.62	2.59	−0.13
1957	0.88	4	0.28	1.61	0.06	14	0.97	0.45	−0.20	15	1.04	4.43	−0.21
1958	1.06	9	0.62	3.58	0.13	9	0.62	0.27	−0.13	10	0.69	2.91	−0.14
1959	1.20	7	0.48	2.77	0.10	11	0.76	0.33	−0.16	12	0.83	3.51	−0.17
1960	1.35	8	0.55	3.18	0.11	10	0.69	0.30	−0.14	11	0.76	3.21	−0.16
1961	1.49	6	0.42	2.42	0.09	12	0.83	0.36	−0.17	13	0.90	3.80	−0.18
1962	1.61	5	0.35	2.01	0.07	13	0.90	0.42	−0.18	14	0.97	4.13	−0.20
1963	2.12	15	1.04	6.36	0.21	2	0.14	0.01	−0.03	4	0.28	1.09	−0.06
1964	11.90	19	1.32	20.78	0.27	19	1.32	12.59	0.27	19	1.32	7.58	0.27
1965	12.40	20	1.39	22.38	0.28	20	1.39	13.75	0.28	20	1.39	8.48	0.28
1966	14.20	21	1.45	25.74	0.30	21	1.45	16.74	0.30	21	1.45	11.24	0.30
1967	15.90	22	1.52	29.35	0.31	22	1.52	19.91	0.31	22	1.52	14.15	0.31
1968	18.20	23	1.59	34.13	0.32	23	1.59	24.25	0.32	23	1.59	18.23	0.32
1969	21.20	24	1.66	40.37	0.34	24	1.66	30.06	0.34	24	1.66	23.77	0.34
1970	4.30	18	1.25	9.09	0.26	18	1.25	1.33	0.26	1	0.07	0.19	−0.01
1971	2.40	1	0.07	0.37	0.01	17	1.18	1.16	−0.24	18	1.25	5.97	−0.26
1972	2.70	2	0.14	0.75	0.03	16	1.11	0.92	−0.23	17	1.18	5.45	−0.24
1973	2.90	3	0.21	1.14	0.04	15	1.04	0.81	−0.21	16	1.11	5.07	−0.23
Sum				245	4.24			124	0.00			136	−0.60

Table 1.8 The LS and R estimations of slope and intercept for Figure 1.1 cases.

Estimators	Case (a)	Case (b)	Case (c)	Case (d)	Estimators	Case (a)	Case (b)	Case (c)	Case (d)
$\hat{\theta}_n^{\mathrm{LS}}$	−234.2	−340.1	−563.8	−983.9	$\hat{\theta}_n^{\mathrm{R}}$	−205.3	−210.7	−224.8	−283.7
$\hat{\boldsymbol{\beta}}_n^{\mathrm{LS}}$	0.120	0.175	0.289	0.504	$\hat{\beta}_n^{\mathrm{R}}$	0.105	0.108	0.115	0.145

procedure to perform rank-based linear regression through the dispersion functions, $D_n(\beta)$ and $D_n(\theta)$. The derivative functions, $L_n(0)$ and $T_n(0)$ also provide useful gradient test statistics, although this is not generally used in machine learning applications. The theoretical underpinnings of the approach may be found in Hettmansperger and McKean (2011).

1.7 Shrinkage Estimation and Subset Selection

We have discussed the use of rank-based methods for simple linear regression with one explanatory variable ($p = 1$) and shown that it offers enormous value compared to LSE. This section extends the previous study to multiple linear regression, i.e. when $p \geq 2$, where p is the number of explanatory variables. Real-world data sets typically have a large number of parameters associated with them. We will see that equations associated with $p \geq 2$ are natural extensions of the $p = 1$ case.

The goal of any regression method used in machine learning is to produce a compact model with high prediction accuracy using a minimum number of parameters. However, new problems arise in multiple linear regression that are not encountered in simple linear regression. In particular, problems may arise due to multicollinearity, over-fitting, under-fitting, identifying the subset of important variables, and shrinking the size of parameter values to improve model accuracy. A penalty function applied to the loss function allows us to mitigate the problems listed above while achieving the desired goals of regression. This subject will be detailed in Chapter 2.

The purpose of this section is to provide background and intuition surrounding the use of penalty functions for multiple linear regression. It gives an overview of penalty functions and offers a number of geometric interpretations of their effects. We will present the basic equations and why they are so useful for machine learning. In addition, the reader will become familiar with some of the terminology used in this field. This will also set the stage for the rest of this book as it focuses on penalty estimation as it pertains to rank-based methods in general.

1.7.1 Multiple Linear Regression using Rank

Most realistic problems fall into the category of multiple linear regression given by the matrix expression

$$\boldsymbol{y} = \theta \mathbf{1}_n + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.7.1}$$

where $\boldsymbol{y}$ is an $n \times 1$ vector of responses, θ is the intercept (which is also referred to as β_0), $\mathbf{1}_n = (1, \ldots, 1)^\top$ is an n-vector, $\boldsymbol{X}$ is an $n \times p$ design matrix, $\boldsymbol{\beta}$ is a $p \times 1$

vector of parameters, and $\boldsymbol{\varepsilon}$ is an $n \times 1$ error term. Each row of $\boldsymbol{X}$ represents one observation out of a total of n, while each column is a vector of values, one for each of the p variables. For the moment, and without loss of generality, if we assume that $\theta = 0$ for a centered problem (i.e. subtracting their respective means from all data columns), then we have

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}. \tag{1.7.2}$$

The LSE of this equation has an elegant closed-form solution as follows:

$$\hat{\boldsymbol{\beta}}_n^{\text{LS}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}. \tag{1.7.3}$$

Two potential problems may arise from this solution. First, if $p > n$, then the matrix $\boldsymbol{X}^\top \boldsymbol{X}$ is not invertible and the parameters are indeterminate. This is easy to check and can be resolved by either adding more data points or reducing the number of parameters (Tibshirani, 1996; Zou and Hastie, 2005). Second, if any of the variables are linearly related, then $\boldsymbol{X}^\top \boldsymbol{X}$ may not be invertible and the parameters are again indeterminate. The solution here is not so straightforward since it implies that there may exist an infinite number of possible solutions. Detecting this multicollinearity condition requires checking the eigenvalues of the correlation matrix, or its condition number, or the variance inflation factor (VIF) (Saleh et al., 2019).

As a simple illustration, let $p = 20$ such that we have variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{20}$. Further, let

$$\boldsymbol{x}_7 = 2\boldsymbol{x}_2,$$
$$\boldsymbol{x}_1 = 0.6\boldsymbol{x}_{11} + 0.4\boldsymbol{x}_4.$$

Then, we will encounter multicollinearity in the $\boldsymbol{X}^\top \boldsymbol{X}$ matrix since $\boldsymbol{x}_2$ is linearly related to $\boldsymbol{x}_7$ due to the first equation, and $\boldsymbol{x}_1$ is linearly related to $\boldsymbol{x}_4$ and $\boldsymbol{x}_{11}$ in the second equation. Therefore, the matrix will not be invertible for reasons cited above.

Outliers could also play a role in this context. We note that high-leverage observations may inadvertently contribute to multicollinearity when in fact there is none between certain variables. The use of rank-based methods would assist in producing robust estimates, or perhaps the identification, removal or correction of these influential observations. However, there may still exist inherent multicollinearity even in the clean data set if the design matrix is not of full rank[6]. Interestingly, the search for methods to address multicollinearity eventually led to the widespread use of penalty functions in machine learning, often for reasons unrelated to this problem.

6 Here we refer to the rank of a matrix.

Let us examine the solution to this problem in the context of multiple regression using penalty functions. This topic will be elaborated greatly in Chapter 2 but we provide the basic formulations here. For multiple linear regression, the unpenalized loss function for the LS method to obtain unbiased estimates is given by

$$J_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 \tag{1.7.4}$$

where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\top$. This is the sum of squares of the errors (SSE) for the multivariate case with an extra factor of $1/n$ to average the result. The term $\boldsymbol{x}_i$ is the ith row of the $\boldsymbol{X}$ matrix. In machine learning, this convex loss function is solved numerically as an optimization problem to obtain the unbiased estimates.

The rank-based loss function is the dispersion function for multiple linear regression as follows

$$D_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})a_n(R_{n_i}(\boldsymbol{\beta})). \tag{1.7.5}$$

This convex loss function is also minimized using an optimization loop to obtain the unbiased estimates.

Each of these loss functions can be modified to include a penalty function and solved iteratively in the same way to produce biased estimates. We note that the penalty is applied only to the p variables and not to the intercept. Therefore, it is best to center the data, that is, subtract the column means from each column of the design matrix and response, before applying a penalty function. In effect, we are setting the intercept to 0. This is not needed in rank-based methods since the explanatory variables are estimated separately from the intercept parameter. However, we will center the data for both to maintain consistency.

1.7.2 Penalty Functions

The importance of the penalty function cannot be overstated. The basic idea is to look into the parameter space to find an alternate set of estimates that provide a more compact, general and accurate model for prediction under the guidance of a penalty function. This process is commonly referred to in machine learning circles as *regularization*. Finding the right penalty function makes machine learning more of an art form rather than a science. In fact, the proper selection and tuning of a penalty function controls prediction accuracy and compactness of a model, as will be seen shortly.

To use a penalty in estimation, we start with a least squares loss function and simply add the penalty function to it. The penalized loss function forms a new objective function that is minimized to produce the estimates. A penalty estimator has the form

$$J_n^{\text{LS-Pen}}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 + p_\lambda(\boldsymbol{\beta}), \tag{1.7.6}$$

where $p_\lambda(\boldsymbol{\beta})$ is the penalty function and λ is the tuning parameter that controls the amount of shrinkage or model complexity (i.e. subset selection).

Likewise the general form of penalty estimators for rank-based methods using the dispersion function is as follows,

$$D_n^{\text{R-Pen}}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})a_n(R_{n_i}(\boldsymbol{\beta})) + p_\lambda(\boldsymbol{\beta}). \tag{1.7.7}$$

Penalty functions will not only mitigate the effects of multicollinearity, but they are effective at addressing other aforementioned issues that arise in multiple linear regression. In particular, if $p = 50$, for example, and we wanted to build a more compact model with say $p = 20$, using the most important explanatory variables, how should we go about doing this? We can choose a suitable penalty function for this purpose. The process is generally referred to as *subset selection*.

Another issue is that if $p = 50$ and we find that we are over-fitting the data, but with $p = 10$ we are under-fitting the data, how do we choose the proper value of p to get the "best" fit? Some penalty functions provide this facility. More generally, we use a penalty to shrink the magnitudes of the parameters to improve model prediction. Also, in the areas of logistic regression and neural networks, the iterations of an optimization loop may lead to large values which may cause overflow. In this case, we are interested in penalty estimators that tend to shrink parameter values.

There are two basic ways to shrink a parameter value: subtraction and division (or equivalently, multiply by a number less than 1). For example, assume we have estimate, $\hat{\beta}_n$, and we want to reduce its value. We can simply let $\hat{\beta}_n^{\text{shrinkage}} = \hat{\beta}_n - \lambda$ and increase λ until we achieve the desired reduction in value. If $\lambda = \hat{\beta}_n$, then $\hat{\beta}_n^{\text{shrinkage}} = 0$ and no further reduction is allowed. A second approach is to set $\hat{\beta}_n^{\text{shrinkage}} = \hat{\beta}_n/(1+\lambda)$ and increase λ until the required reduction is obtained. However, in this case, $\hat{\beta}_n^{\text{shrinkage}}$ does not reach 0, except at $\lambda = \infty$. As simple as these two cases may appear, they are the basis of the two most popular penalty estimators in use today: LASSO and ridge. The key characteristics of these two important penalty functions are illustrated in Figure 1.7.

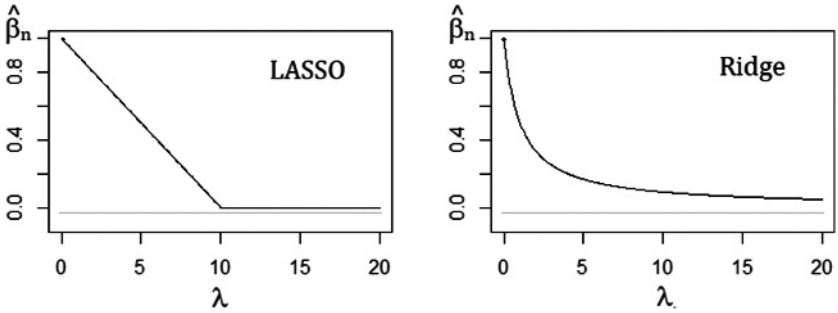

Figure 1.7 Key shrinkage characteristics of LASSO and ridge.

1.7.3 Shrinkage Estimation

The most widely used form of penalty estimation is referred to as a *ridge regression* (Hoerl and Kennard, 1970). In this case,

$$p_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} |\beta_j|^2, \tag{1.7.8}$$

where λ is a tuning parameter. It is often referred to as the L_2 penalty since it is based on the L_2-norm. Its primary purpose is to shrink the parameters thus moving the estimates away from the unbiased values and towards the null hypothesis, $\boldsymbol{\beta} = 0$. By properly tuning λ, it is possible to apply enough shrinkage to the parameters such that the resulting biased parameters provide an accurate model for prediction.

The objective function using the least squares (LS) loss and ridge penalty functions is given by

$$J_n^{\text{LS-Ridge}}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j|^2. \tag{1.7.9}$$

The objective function using the rank loss and ridge penalty functions is given by

$$D_n^{\text{R-Ridge}}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}) a_n(R_{n_i}(\boldsymbol{\beta})) + \lambda \sum_{j=1}^{p} |\beta_j|^2. \tag{1.7.10}$$

One can show that the penalty function effectively places a constraint on the sum of the squares of the parameters. If we had only two parameters, β_1 and β_2, the constraint would be as follows,

$$\beta_1^2 + \beta_2^2 \leq c, \tag{1.7.11}$$

where c is a constant. Note that this constraint would form a circular region about the origin, as shown in Figure 1.8. The figure represents the two-dimensional plane of possible β_1 and β_2 values. The circles represent the constraint $\beta_1^2 + \beta_2^2 = c$ due to the ridge penalty function for different values of c, as set by λ in the penalty function. The point marked "X" represents the unbiased rank or LS estimates. The contours around the estimates represent lines of equal value of the unpenalized loss function. The largest circle (labeled 0) is associated with $\lambda = 0$, while the point the origin (where $\beta_1 = \beta_2 = 0$) is due to $\lambda = \infty$.

Starting at the unbiased solution on the circle labeled 0, with $\lambda_0 = 0$, as we apply the penalty function with tuning factor λ, the estimates shrink and follow some hypothetical trajectory depicted in the figure until the edge of the circle 1 is reached at $\lambda = \lambda_1 > 0$. As λ increases to $\lambda = \lambda_2 > \lambda_1$ (circle 2), we shrink the estimates by a greater degree. At $\lambda = \lambda_3 > \lambda_2$, we reach the circle 3. This continues until all estimates reach 0 at $\lambda = \infty$ at the origin. Essentially, we are shrinking the estimates towards the null hypothesis, $\boldsymbol{\beta} = 0$. In practice, we seek

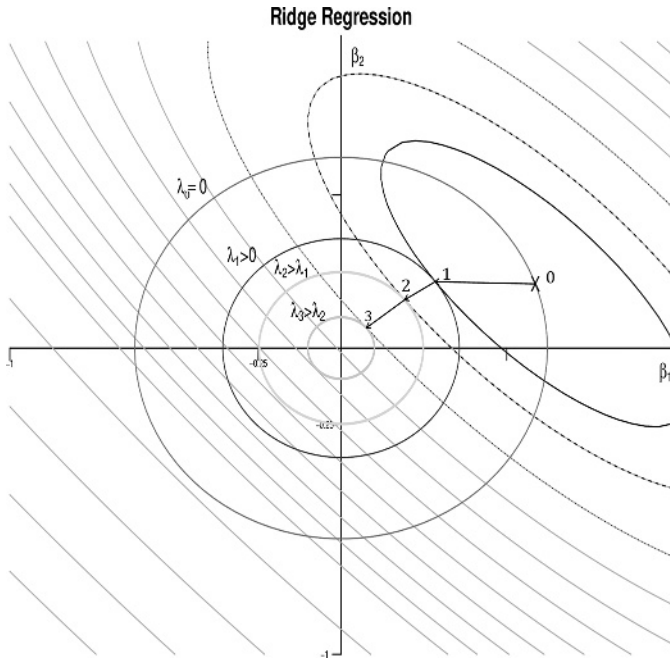

Figure 1.8 Geometric interpretation of ridge.

some intermediate point between the unbiased estimate and the origin. If we have over-fitting or under-fitting problems[7], we can try a number of different values of λ until a proper balance is struck between the two. The process of finding the optimal value, λ_{opt}, to balance this *bias-variance trade-off* is called regularization and is used to obtain high model prediction accuracy.

1.7.4 Subset Selection

The generic term "subset selection" refers to the process of identifying the most important explanatory variables from a large set of given variables. In particular, if there are p variables, the goal is to find a smaller subset, $p_0 < p$, with which to build an accurate predictive model. Unfortunately, this requires that 2^p models be built and tested to determine which one offers the best subset. This would be the "oracle" subset meaning the subset we would ideally want in our compact model. Of course, this process is computationally prohibitive so we seek other methods that have the potential to find this oracle subset with much less effort. Any approach that is able to find the oracle subset for any given p_0 is said to have the oracle property.

A first approach in this direction combining shrinkage and subset selection is referred to as *LASSO* (least absolute shrinkage and selection operator) (Tibshirani, 1996). In this case,

$$p_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} |\beta_j|, \tag{1.7.12}$$

where λ is the tuning parameter. This is also referred to as the L_1 penalty because it is reminiscent of the L_1-norm.

The objective function with least squares loss and LASSO penalty functions is given by

$$J_n^{\text{LS-LASSO}}(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{1.7.13}$$

The objective function with rank loss and LASSO penalty functions is given by

$$D_n^{\text{R-LASSO}}(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{y}_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}) a_n(R_{n_i}(\boldsymbol{\beta})) + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{1.7.14}$$

---

7 The definitions of over-fitting and under-fitting will be elaborated in the next chapter. Briefly, over-fitting refers to the tendency to try to fit every point in the data set as closely as possible, whereas under-fitting is inadvertently not fitting enough points in an attempt to create a generalized model.

Using this formulation, a constraint, given by c, is imposed by the penalty function as follows,

$$|\beta_1| + |\beta_2| \leq c. \tag{1.7.15}$$

The resulting constraint geometry is a diamond shape in two dimensions rather than the circle, as shown in Figure 1.9. The adjustment of $\lambda > 0$ has the effect of decreasing the size of the diamond. If $\lambda_0 = 0$, we obtain the unbiased estimates. As λ is increased, the diamond shrinks in size, as do the resulting parameters so it is a shrinkage estimator. Eventually, when $\lambda = \infty$, the origin is reached, as was the case in ridge regression.

The hypothetical trajectory for LASSO starts at the unbiased LSE or rank estimates ("X") on diamond 0 and terminates on diamond 1 with $\lambda = \lambda_1 > 0$. Both estimates shrink but neither one is 0 yet. However, when $\lambda = \lambda_2 > \lambda_1$ and the path lands on one corner of diamond 2, where $\beta_2 = 0$, and we are left with a model with only $\beta_1 \neq 0$. We can now remove β_2 and keep β_1. As such, LASSO is a subset selection estimator. Further increases in λ eventually forces $\beta_1 = 0$ at the origin.

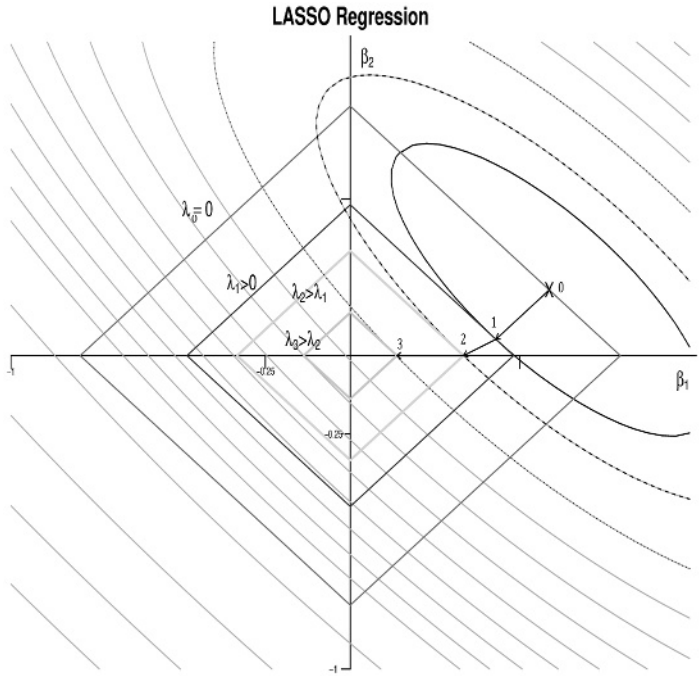

Figure 1.9 Geometric interpretation of LASSO.

Subset selection is very important when the number of parameters is very large. One can imagine a data set with $p = 50$ being reduced to $p_0 = 30$ by this method, and ideally only the important explanatory variables would be retained in the compact model. Unfortunately, LASSO does not have the oracle property; that is, it is possible that the "wrong corner" of the diamond may be reached during shrinkage.

For example, let $p = 7$ and assume we know a priori that $\beta_4 = 0$ and $\beta_6 = 0$ while all other parameters are non-zero. If a penalty estimator such as LASSO reports that initially we should set $\beta_3 = 0$ and $\beta_7 = 0$, then it does not have the oracle property. However, it may still produce a compact model but the parameter values may not be as meaningful. This is important because of model interpretability. We want to be able to eliminate parameters from the model that are not explanatory in terms of the response variable, leaving only those that are important. Whenever model interpretability is important, we require the oracle property be present in the penalty function during subset selection.

Improved methods have been developed that do indeed possess the oracle properties. One in particular is relevant to the discussion here: adaptive LASSO (aLASSO) (Zou, 2006). In the original LASSO, the λ value was applied uniformly to all parameters. In aLASSO, an additional weight is applied to each parameter as follows,

$$p_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} w_j |\beta_j|. \tag{1.7.16}$$

For example, the weight for each parameter could be set to $w_j = 1/|\hat{\beta}_j^*|$ to produce the new penalty function

$$p_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_j^*|}, \tag{1.7.17}$$

where $\hat{\beta}_j^*$, a known quantity, is the an unpenalized arbitrary estimate for β_j, i.e. the unbiased least squares or rank estimate. We usually consider a $\sqrt{n}$-consistent estimator as $\hat{\beta}_j^*$ to retain good properties for the penalized estimator.

We can understand the adaptive LASSO approach by realizing that small $\hat{\beta}_j^*$ values will lead to large weights, which is equivalent to a high λ with the diamond close to the origin (see Figure 1.9), while large $\hat{\beta}_j^*$ values would be equivalent to a small λ and accordingly a large diamond constraint. Hence, each parameter has its own effective λ based on the unpenalized estimates. This makes sense intuitively since the unpenalized solution has valuable

information about the magnitude of the estimates, which in turn can be used as a weighting factor to guide the aLASSO estimation in the right direction for the removal of unimportant variables.

1.7.5 Blended Approaches

Another possible approach is to combine the ridge and LASSO penalties into a single penalty referred to as the *elastic net* (Zou and Hastie, 2005) as follows

$$p_\lambda(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} |\beta_j|^2. \tag{1.7.18}$$

We now have two tuning parameters, λ_1 and λ_2. We note that if $\lambda_1 = 0$ we obtain ridge regression, whereas if $\lambda_2 = 0$ we obtain LASSO. A slight modification of elastic net involves replacing LASSO with aLASSO in order to include the oracle property, as follows

$$p_\lambda(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_j^*|} + \lambda_2 \sum_{j=1}^{p} |\beta_j|^2. \tag{1.7.19}$$

The two parameters, λ_1 and λ_2, are adjusted until the desired bias/variance trade-off is achieved (i.e. balancing over-fitting and under-fitting). If we let $\lambda_1 = \lambda\alpha$ and $\lambda_2 = \lambda(1-\alpha)/2$, then it is clear that if $\alpha = 0$ we are using ridge while if $\alpha = 1$ we are using aLASSO. In practice, we seek values of λ and α that provide the desired model accuracy. One can understand that the oracle property will be enforced only when $\alpha = 1$, but not necessarily as α approaches 0. However, it provides a better combination of the two penalty functions since it offers a solution to multicollinearity, shrinkage, and oracle subset selection.

1.8 Summary

To summarize this chapter, we have made the case for rank-based estimation as a robust alternative to least squares estimation (LSE) when dealing with outliers, which are contained in most realistic data sets. It was shown that LSE may produce misleading results in the presence of outliers. A number of different cases for the Belgium telephone data set were presented to compare and contrast rank and LSE. The rank estimates were demonstrably better in terms of robustness, especially in high-impact and high-leverage cases of a single outlier.

Further, to improve prediction accuracy, we have described the use of penalty functions which can be viewed as additional constraints on parameter values. Ridge regression provides parameter shrinkage and can be used to reduce the magnitude of parameter values. LASSO is another penalty function that features shrinkage and subset selection, although it does not hold the oracle property. aLASSO, a variant of LASSO, has the oracle property but typically requires unbiased estimates as weighting terms. We also presented the elastic net which is a combination of the two. Much more detail about these penalty functions will be presented in Chapter 2.

We described geometric interpretations of the effect of penalty functions on the estimates. They effectively introduce constraints on the parameter values to lie on a circle in the case of ridge, or on a diamond shape in the case of LASSO. Constraints on LSE and R-estimates provide effective solutions to the problems of multicollinearity, over-fitting/under-fitting, identifying important explanatory variables and shrinking parameter values.

A conceptual view of LSE, rank, ridge and LASSO is provided in Table 1.9. It offers another context of where rank-based methods fit into the overall picture. In particular, the quadratic loss function is based on the L_2-norm, as is the ridge penalty function. However, rank methods are (loosely) based on the L_1-norm, as is the LASSO penalty. Given that rank-based methods produce robust estimates, we say that it behaves like an L_1 loss function. We note the symmetry between the loss and penalty functions in this table. We also specify the key attribute of each method.

We have focused on rank-based methods for linear regression in this chapter. However, the rank-based approach can also be extended to logistic regression and neural networks as we shall demonstrate in the last two chapters of this book on machine learning. Furthermore, penalty functions can also be applied to logistic regression and neural networks in a similar manner. In the next few

Table 1.9 Interpretation of L_1/L_2 loss and penalty functions

Norm	Loss function	Penalty function	Interpretation of entries
L_2	Least squares (mean-based)	Ridge (shrinkage)	L_2 loss is LSE L_2 penalty is ridge
L_1	**Rank (median-based)**	LASSO (selection)	L_1 loss is R-estimation L_1 penalty is LASSO

chapters, we provide a more rigorous treatment of these and other subjects as they pertain to the theory and practice of rank-based methods.

1.9 Problems

1.1. We want to build a simple linear model using Theil's estimator for

$$y_i = \theta + \beta x_i, \quad i = 1, \ldots, 100.$$

The Theil estimator of a set of points (x_i, y_i) requires finding the median of pairwise slopes of all points,$(\frac{y_j - y_i}{x_j - x_i})$, and the median of the partial-residuals, $(y_i - \hat{\beta}_n^{\text{Theil}} x_i)$. The estimates are given by

$$\hat{\beta}_n^{\text{Theil}} = \underset{1 \le i < j \le n,\, i \ne j}{\text{median}} \left(\frac{y_j - y_i}{x_j - x_i} \right), \text{ and } \hat{\theta}_n^{\text{Theil}} = \underset{1 \le i \le n}{\text{median}} \left(y_i - \hat{\beta}_n^{\text{Theil}} x_i \right).$$

If $n = 100$, how many slope calculations do we need to compute $\hat{\beta}_n^{\text{Theil}}$? How many for $\hat{\theta}_n^{\text{Theil}}$? Which is $O(n^2)$ or $O(n)$ in computational complexity?

1.2. Consider the following small data set.

x_i	0.10	0.20	0.30	0.40	0.50	0.60
y_i	3.2	4.0	4.2	4.7	6.5	5.5

Let the data be modeled as

$$\boldsymbol{y} = \theta + \beta \boldsymbol{x} + \boldsymbol{\varepsilon}.$$

Perform hand calculations, or use R or Python, to do the following.

(a) Find the least squares estimates of θ and β using

$$\hat{\beta}_n^{\text{LS}} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \quad \text{and} \quad \hat{\theta}_n^{\text{LS}} = \bar{y} - \hat{\beta}_n^{\text{LS}} \bar{x}.$$

(b) Based on (a), find the outlier(s) using a residual histogram plot.
(c) Find median-based estimates for the intercept and slope parameters using Theil's method.
(d) Based on (c), find the outlier(s) from Theil's estimator using another residual histogram plot.

(e) Remove the outlier(s) from the data set found in part (d) and repeat the process. Comment on the results obtained with and without outliers.

1.3. As an exercise, show that $\phi(u) = -\sqrt{2}\cos(\pi u)$ satisfies the constraints of $\phi(u), 0 < u < 1$, such that $\int_0^1 \phi(u)\mathrm{d}u = 0$ and $\int_0^1 \phi^2(u)du = 1$. Show that it is also skew-symmetric about 1/2. What is $\phi^+(u)$ in this case? Plot the new scoring functions as in Figure 1.5 along with the Wilcoxon score functions from $0 < u < 1$.

1.4. For the data set in Problem 1.2, obtain the estimators for θ and β using a rank-based method as follows. You can calculate the solution by hand, or use R or Python.

(a) The estimator $\hat{\beta}_n^{\mathrm{R}}$ is given by

$$\hat{\beta}_n^{\mathrm{R}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - \beta x_i) \sqrt{12} \left(\frac{R_{n_i}(\beta)}{n+1} - \frac{1}{2} \right) \right\},$$

where $R_{n_i}(\beta) = R_{n_i}(y_i - \beta x_i)$. Carry out the minimization step by computing the sum

$$D_n(\beta) = \sum_{i=1}^{n} (y_i - \beta x_i) a_n(R_{n_i}(\beta)).$$

for $\beta = 4.5, 4.6, 4.7, 4.8, 4.9, 5.0$.

(b) The estimator $\hat{\theta}_n^{\mathrm{R}}$ is given by

$$\hat{\theta}_n^{\mathrm{R}} = \operatorname{argmin} \left\{ \sum_{i=1}^{n} \left| y_i - (\theta + \hat{\beta}^{\mathrm{R}} x_i) \right| \sqrt{3} \left(\frac{R_{n_i}^+(\theta)}{n+1} \right) \right\},$$

where

$$R_{n_i}^+(\theta) = R_{n_i}(|y_i - (\theta + \hat{\beta}^{\mathrm{R}} x_i)|).$$

Carry out the minimization step by computing the sum

$$D_n(\theta) = \sum_{i=1}^{n} |y_i - (\theta + \hat{\beta}^{\mathrm{R}} x_i)| \, a_n^+(R_{n_i}^+(\theta)),$$

for $\theta = 2.5, 2.6, 2.7, 2.8, 2.9, 3.0$.

(c) Plot the linear regression results from Problem 1.2 and the above results against one another superimposed on the original data set.

(d) Compare and contrast the three estimators used on this data set.

1.5. For the following data set, it is best to use R to solve the problems below. In R, the least square estimates can be obtained using a built-in function `lm()`. However, a package called Rfit (Kloke and McKean, 2012) should

be installed in order to quickly obtain the rank-based results. (If you are doing hand calculations only, use the last six of the eight (x, y) pairs of data in the second row.)

x	5551	794	1619	2079	918	1231	3641	4314
y	40	29	28	23	21	17	27	39
x	2786	3989	376	5428	2628	4497	3545	−2108
y	38	46	24	48	34	43	38	94

(a) Draw a scatter plot for this data set. What type of outlier(s) exist in the data, based on the cases given in Figure 1.3?
(b) Fit a line using a least squares method. Plot the associated line on the scatter plot.
(c) Fit a line using a rank-based method. Plot the associated line on the same scatter plot.
(d) Discuss the reasons why the two plots are so different. Which of the two estimates (rank or LS) is more representative of the data?

1.6. Given that the rank-based slope parameter, β, can be estimated using

$$D_n(\beta) = \sum_{i=1}^{n} (y_i - \beta x_i) a_n(R_{n_i}(\beta)),$$

verify this equation by writing a program in R or Python to reproduce the results of Table 1.6 using the telephone data set of Table 1.3. Identify the three points that were computed in Figure 1.6.

1.7. Given that the rank-based intercept, θ, of a simple linear regression problem can be estimated using

$$D_n(\theta) = \sum_{i=1}^{n} |y_i - (\theta + \hat{\beta}^{\mathrm{R}} x_i)| a_n^+(R_{n_i}^+(\theta)),$$

where

$$R_{n_i}^+(\theta) = R_{n_i}(|y_i - (\theta + \hat{\beta}^{\mathrm{R}} x_i)|).$$

verify this equation by writing a program in R or Python that reproduces the results of Table 1.7 using the telephone data set of Table 1.3. Identify the three points that were computed in Figure 1.6.

1.8. A bridge rank-based penalty estimator would take the form,

$$D_n^{\text{R-bridge}}(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i \boldsymbol{\beta}) a_n(R_{n_i}(\boldsymbol{\beta})) + \lambda \sum_{j=1}^{p} |\beta_j|^{\gamma},$$

where the exponent γ has been introduced in the last term.

(a) What type of penalty function is produced if $\gamma = 1$?
(b) What type of penalty function is produced if $\gamma = 2$?
(c) What if $\gamma = 1.5$ or any value between 1 and 2? How would you characterize this type of penalty function?

1.9. For a centered problem, the least squares estimator has the form:

$$\hat{\beta}_n^{\text{LS}} = (X_c^\top X_c)^{-1} X_c^\top \boldsymbol{y}.$$

One can show that the LS-ridge estimator has the form:

$$\hat{\beta}_n^{\text{LS-Ridge}} = (X_c^\top X_c + \lambda)^{-1} X_c^\top \boldsymbol{y}.$$

Using the following data from a previous problem, center the data and the solve the following problems.

x_i	0.10	0.20	0.30	0.40	0.50	0.60
y_i	3.2	4.0	4.2	4.7	5.0	5.5

(a) Compute $\hat{\beta}_n^{\text{LS}}$ and $\hat{\theta}_n^{\text{LS}}$ after centering.
(b) Compute $\hat{\beta}_n^{\text{R}}$ and $\hat{\theta}_n^{\text{R}}$.
(c) Find λ such that $\hat{\beta}_n^{\text{LS-Ridge}}$ has the same value as $\hat{\beta}_n^{\text{R}}$.
(d) Plot the results of (b) and (c) and comment on the results and suggest which is the better solution. In particular, compare the intercept of the two methods. Also, is it possible to use LS-ridge to produce the results obtained by rank-based methods?

1.10. (PROJECT) In this problem, you will use R to compare rank with LS. If you have not installed R on your computer, this would be a good time to do so. You should also use `install.packages("Rfit")` to install the Rfit package.

(a) Load the Rfit library and examine the telephone data used in this chapter.

```
library("Rfit")
data(telephone)
telephone
```

(b) Run rfit() on the telephone data and compare the rank results with Table 1.8, case (d).

```
fit <- rfit(calls ~ year, data=telephone)
summary(fit)
```

(c) Run lm() on the telephone data and compare the LS results with Table 1.8, case (d).

```
lmfit <- lm(calls ~ year, data=telephone)
summary(lmfit)
```

(d) Plot the results of the two regression methods. Which is the rank line and which is the LS line?

```
plot(calls ~ year,data=telephone,
        main="Rank vs. LSE",ylim=c(0,25))
abline(coef(lmfit),lwd=2,col="black")
abline(coef(fit),lwd=2,col="red")
```