$$C_{jk} = \sum_{i=1}^{M} \frac{1}{w_i^2} V_{ji} V_{ki} \tag{15.6.10}$$

CITED REFERENCES AND FURTHER READING:

Efron, B. 1982, *The Jackknife, the Bootstrap, and Other Resampling Plans* (Philadelphia: S.I.A.M.). [1]

Efron, B., and Tibshirani, R. 1986, *Statistical Science* vol. 1, pp. 54–77. [2]

Avni, Y. 1976, *Astrophysical Journal*, vol. 210, pp. 642–646. [3]

Lampton, M., Margon, M., and Bowyer, S. 1976, *Astrophysical Journal*, vol. 208, pp. 177–190.

Brownlee, K.A. 1965, *Statistical Theory and Methodology*, 2nd ed. (New York: Wiley).

Martin, B.R. 1971, *Statistics for Physicists* (New York: Academic Press).

15.7 Robust Estimation

The concept of *robustness* has been mentioned in passing several times already. In §14.1 we noted that the median was a more robust estimator of central value than the mean; in §14.6 it was mentioned that rank correlation is more robust than linear correlation. The concept of outlier points as exceptions to a Gaussian model for experimental error was discussed in §15.1.

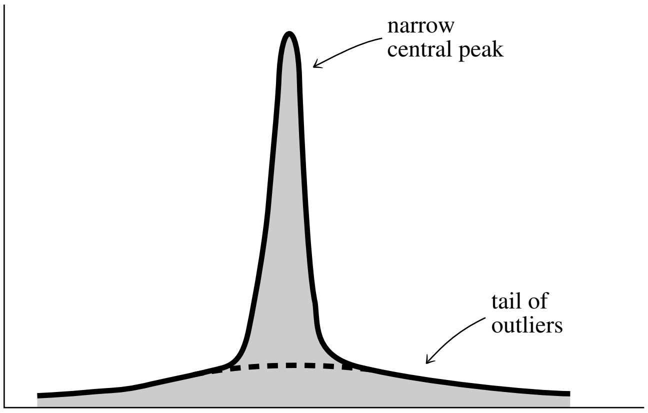
The term "robust" was coined in statistics by G.E.P. Box in 1953. Various definitions of greater or lesser mathematical rigor are possible for the term, but in general, referring to a statistical estimator, it means "insensitive to small departures from the idealized assumptions for which the estimator is optimized." [1,2] The word "small" can have two different interpretations, both important: either fractionally small departures for all data points, or else fractionally large departures for a small number of data points. It is the latter interpretation, leading to the notion of outlier points, that is generally the most stressful for statistical procedures.

Statisticians have developed various sorts of robust statistical estimators. Many, if not most, can be grouped in one of three categories.

M-estimates follow from maximum-likelihood arguments very much as equations (15.1.5) and (15.1.7) followed from equation (15.1.3). M-estimates are usually the most relevant class for model-fitting, that is, estimation of parameters. We therefore consider these estimates in some detail below.

L-estimates are "linear combinations of order statistics." These are most applicable to estimations of central value and central tendency, though they can occasionally be applied to some problems in estimation of parameters. Two "typical" L-estimates will give you the general idea. They are (i) the median, and (ii) *Tukey's trimean*, defined as the weighted average of the first, second, and third quartile points in a distribution, with weights 1/4, 1/2, and 1/4, respectively.

R-estimates are estimates based on rank tests. For example, the equality or inequality of two distributions can be estimated by the *Wilcoxon test* of computing the mean rank of one distribution in a combined sample of both distributions. The Kolmogorov-Smirnov statistic (equation 14.3.6) and the Spearman rank-order



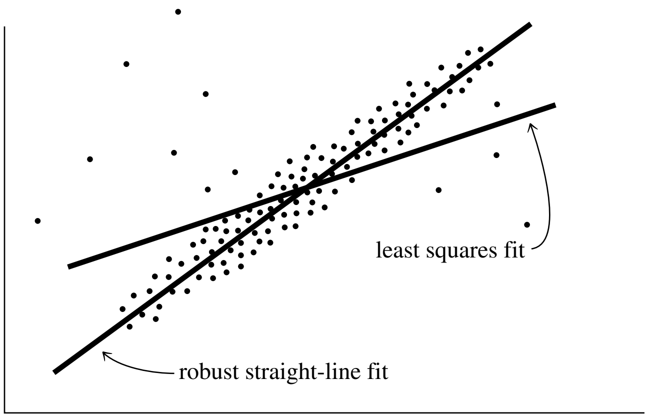


Figure 15.7.1. Examples where robust statistical methods are desirable: (a) A one-dimensional distribution with a tail of outliers; statistical fluctuations in these outliers can prevent accurate determination of the position of the central peak. (b) A distribution in two dimensions fitted to a straight line; non-robust techniques such as least-squares fitting can have undesired sensitivity to outlying points.

correlation coefficient (14.6.1) are R-estimates in essence, if not always by formal definition.

Some other kinds of robust techniques, coming from the fields of optimal control and filtering rather than from the field of mathematical statistics, are mentioned at the end of this section. Some examples where robust statistical methods are desirable are shown in Figure 15.7.1.

Estimation of Parameters by Local M-Estimates

Suppose we know that our measurement errors are not normally distributed. Then, in deriving a maximum-likelihood formula for the estimated parameters $\mathbf{a}$ in a model $y(x;\mathbf{a})$, we would write instead of equation (15.1.3)

$$P = \prod_{i=1}^{N} \{\exp\left[-\rho(y_i, y\,\{x_i;\mathbf{a}\})\right] \Delta y\} \tag{15.7.1}$$

where the function ρ is the negative logarithm of the probability density. Taking the logarithm of (15.7.1) analogously with (15.1.4), we find that we want to minimize the expression

$$\sum_{i=1}^{N} \rho(y_i, y\{x_i; \mathbf{a}\}) \tag{15.7.2}$$

Very often, it is the case that the function ρ depends not independently on its two arguments, measured y_i and predicted $y(x_i)$, but only on their difference, at least if scaled by some weight factors σ_i which we are able to assign to each point. In this case the M-estimate is said to be *local*, and we can replace (15.7.2) by the prescription

$$\text{minimize over } \mathbf{a} \qquad \sum_{i=1}^{N} \rho\left(\frac{y_i - y(x_i; \mathbf{a})}{\sigma_i}\right) \tag{15.7.3}$$

where the function $\rho(z)$ is a function of a single variable $z \equiv [y_i - y(x_i)]/\sigma_i$.

If we now define the derivative of $\rho(z)$ to be a function $\psi(z)$,

$$\psi(z) \equiv \frac{d\rho(z)}{dz} \tag{15.7.4}$$

then the generalization of (15.1.7) to the case of a general M-estimate is

$$0 = \sum_{i=1}^{N} \frac{1}{\sigma_i} \psi\left(\frac{y_i - y(x_i)}{\sigma_i}\right) \left(\frac{\partial y(x_i; \mathbf{a})}{\partial a_k}\right) \qquad k = 1, \dots, M \tag{15.7.5}$$

If you compare (15.7.3) to (15.1.3), and (15.7.5) to (15.1.7), you see at once that the specialization for normally distributed errors is

$$\rho(z) = \frac{1}{2} z^2 \qquad \psi(z) = z \qquad \text{(normal)} \tag{15.7.6}$$

If the errors are distributed as a *double* or *two-sided exponential*, namely

$$\text{Prob}\,\{y_i - y(x_i)\} \sim \exp\left(-\left|\frac{y_i - y(x_i)}{\sigma_i}\right|\right) \tag{15.7.7}$$

then, by contrast,

$$\rho(x) = |z| \qquad \psi(z) = \text{sgn}(z) \qquad \text{(double exponential)} \tag{15.7.8}$$

Comparing to equation (15.7.3), we see that in this case the maximum likelihood estimator is obtained by minimizing the *mean absolute deviation*, rather than the mean square deviation. Here the tails of the distribution, although exponentially decreasing, are asymptotically much larger than any corresponding Gaussian.

A distribution with even more extensive — therefore sometimes even more realistic — tails is the *Cauchy* or *Lorentzian* distribution,

$$\text{Prob}\,\{y_i - y(x_i)\} \sim \frac{1}{1 + \frac{1}{2}\left(\frac{y_i - y(x_i)}{\sigma_i}\right)^2} \tag{15.7.9}$$

This implies

$$\rho(z) = \log\left(1 + \frac{1}{2}z^2\right) \qquad \psi(z) = \frac{z}{1 + \frac{1}{2}z^2} \qquad \text{(Lorentzian)} \qquad (15.7.10)$$

Notice that the ψ function occurs as a weighting function in the generalized normal equations (15.7.5). For normally distributed errors, equation (15.7.6) says that the more deviant the points, the greater the weight. By contrast, when tails are somewhat more prominent, as in (15.7.7), then (15.7.8) says that all deviant points get the same relative weight, with only the sign information used. Finally, when the tails are even larger, (15.7.10) says the ψ increases with deviation, then starts *decreasing*, so that very deviant points — the true outliers — are not counted at all in the estimation of the parameters.

This general idea, that the weight given individual points should first increase with deviation, then decrease, motivates some additional prescriptions for ψ which do not especially correspond to standard, textbook probability distributions. Two examples are

Andrew's sine

$$\psi(z) = \begin{cases} \sin(z/c) & |z| < c\pi \\ 0 & |z| > c\pi \end{cases} \qquad (15.7.11)$$

If the measurement errors happen to be normal after all, with standard deviations σ_i, then it can be shown that the optimal value for the constant c is $c = 2.1$.

Tukey's biweight

$$\psi(z) = \begin{cases} z(1 - z^2/c^2)^2 & |z| < c \\ 0 & |z| > c \end{cases} \qquad (15.7.12)$$

where the optimal value of c for normal errors is $c = 6.0$.

Numerical Calculation of M-Estimates

To fit a model by means of an M-estimate, you first decide which M-estimate you want, that is, which matching pair ρ, ψ you want to use. We rather like (15.7.8) or (15.7.10).

You then have to make an unpleasant choice between two fairly difficult problems. Either find the solution of the nonlinear set of M equations (15.7.5), or else minimize the single function in M variables (15.7.3).

Notice that the function (15.7.8) has a discontinuous ψ, and a discontinuous derivative for ρ. Such discontinuities frequently wreak havoc on both general nonlinear equation solvers and general function minimizing routines. You might now think of rejecting (15.7.8) in favor of (15.7.10), which is smoother. However, you will find that the latter choice is also bad news for many general equation solving or minimization routines: small changes in the fitted parameters can drive $\psi(z)$ off its peak into one or the other of its asymptotically small regimes. Therefore, different terms in the equation spring into or out of action (almost as bad as analytic discontinuities).

Don't despair. If your computer budget (or, for personal computers, patience) is up to it, this is an excellent application for the downhill simplex minimization algorithm exemplified in amoeba §10.4 or amebsa in §10.9. Those algorithms make no assumptions about continuity; they just ooze downhill and will work for virtually any sane choice of the function ρ.

It is very much to your (financial) advantage to find good starting values, however. Often this is done by first fitting the model by the standard χ^2 (nonrobust) techniques, e.g., as described in §15.4 or §15.5. The fitted parameters thus obtained are then used as starting values in amoeba, now using the robust choice of ρ and minimizing the expression (15.7.3).

Fitting a Line by Minimizing Absolute Deviation

Occasionally there is a special case that happens to be much easier than is suggested by the general strategy outlined above. The case of equations (15.7.7)–(15.7.8), when the model is a simple straight line

$$y(x; a, b) = a + bx \tag{15.7.13}$$

and where the weights σ_i are all equal, happens to be such a case. The problem is precisely the robust version of the problem posed in equation (15.2.1) above, namely fit a straight line through a set of data points. The merit function to be minimized is

$$\sum_{i=1}^{N} |y_i - a - bx_i| \tag{15.7.14}$$

rather than the χ^2 given by equation (15.2.2).

The key simplification is based on the following fact: The median c_M of a set of numbers c_i is also that value which minimizes the sum of the absolute deviations

$$\sum_i |c_i - c_M|$$

(Proof: Differentiate the above expression with respect to c_M and set it to zero.)

It follows that, for fixed b, the value of a that minimizes (15.7.14) is

$$a = \text{median}\,\{y_i - bx_i\} \tag{15.7.15}$$

Equation (15.7.5) for the parameter b is

$$0 = \sum_{i=1}^{N} x_i \ \text{sgn}(y_i - a - bx_i) \tag{15.7.16}$$

(where $\text{sgn}(0)$ is to be interpreted as zero). If we replace a in this equation by the implied function $a(b)$ of (15.7.15), then we are left with an equation in a single variable which can be solved by bracketing and bisection, as described in §9.1. (In fact, it is dangerous to use any fancier method of root-finding, because of the discontinuities in equation 15.7.16.)

Here is a routine that does all this. It calls select (§8.5) to find the median. The bracketing and bisection are built in to the routine, as is the χ^2 solution that generates the initial guesses for a and b. Notice that the evaluation of the right-hand side of (15.7.16) occurs in the function rofunc, with communication via global (top-level) variables.

```
#include <math.h>
#include "nrutil.h"
int ndatat;
float *xt,*yt,aa,abdevt;

void medfit(float x[], float y[], int ndata, float *a, float *b, float *abdev)
```

Fits $y = a + bx$ by the criterion of least absolute deviations. The arrays x[1..ndata] and y[1..ndata] are the input experimental points. The fitted parameters a and b are output, along with abdev, which is the mean absolute deviation (in y) of the experimental points from the fitted line. This routine uses the routine rofunc, with communication via global variables.

```
{
    float rofunc(float b);
    int j;
    float bb,b1,b2,del,f,f1,f2,sigb,temp;
    float sx=0.0,sy=0.0,sxy=0.0,sxx=0.0,chisq=0.0;

    ndatat=ndata;
    xt=x;
    yt=y;
    for (j=1;j<=ndata;j++) {          As a first guess for a and b, we will find the least-
        sx += x[j];                       squares fitting line.
        sy += y[j];
        sxy += x[j]*y[j];
        sxx += x[j]*x[j];
    }
    del=ndata*sxx-sx*sx;
    aa=(sxx*sy-sx*sxy)/del;           Least-squares solutions.
    bb=(ndata*sxy-sx*sy)/del;
    for (j=1;j<=ndata;j++)
        chisq += (temp=y[j]-(aa+bb*x[j]),temp*temp);
    sigb=sqrt(chisq/del);             The standard deviation will give some idea of how
    b1=bb;                                big an iteration step to take.
    f1=rofunc(b1);
    b2=bb+SIGN(3.0*sigb,f1);
    Guess bracket as 3-σ away, in the downhill direction known from f1.
    f2=rofunc(b2);
    if (b2 == b1) {
        *a=aa;
        *b=bb;
        *abdev=abdevt/ndata;
        return;
    }
    while (f1*f2 > 0.0) {             Bracketing.
        bb=b2+1.6*(b2-b1);
        b1=b2;
        f1=f2;
        b2=bb;
        f2=rofunc(b2);
    }
    sigb=0.01*sigb;                   Refine until error a negligible number of standard
    while (fabs(b2-b1) > sigb) {          deviations.
        bb=b1+0.5*(b2-b1);            Bisection.
        if (bb == b1 || bb == b2) break;
        f=rofunc(bb);
        if (f*f1 >= 0.0) {
            f1=f;
            b1=bb;
        } else {
            f2=f;
            b2=bb;
        }
    }
    *a=aa;
    *b=bb;
```

```
    *abdev=abdevt/ndata;
}
```

```
#include <math.h>
#include "nrutil.h"
#define EPS 1.0e-7

extern int ndatat;                    Defined in medfit.
extern float *xt,*yt,aa,abdevt;

float rofunc(float b)
Evaluates the right-hand side of equation (15.7.16) for a given value of b. Communication with
the routine medfit is through global variables.
{
    float select(unsigned long k, unsigned long n, float arr[]);
    int j;
    float *arr,d,sum=0.0;

    arr=vector(1,ndatat);
    for (j=1;j<=ndatat;j++) arr[j]=yt[j]-b*xt[j];
    if (ndatat & 1) {
        aa=select((ndatat+1)>>1,ndatat,arr);
    }
    else {
        j=ndatat >> 1;
        aa=0.5*(select(j,ndatat,arr)+select(j+1,ndatat,arr));
    }
    abdevt=0.0;
    for (j=1;j<=ndatat;j++) {
        d=yt[j]-(b*xt[j]+aa);
        abdevt += fabs(d);
        if (yt[j] != 0.0) d /= fabs(yt[j]);
        if (fabs(d) > EPS) sum += (d >= 0.0 ? xt[j] : -xt[j]);
    }
    free_vector(arr,1,ndatat);
    return sum;
}
```

Other Robust Techniques

Sometimes you may have *a priori* knowledge about the probable values and probable uncertainties of some parameters that you are trying to estimate from a data set. In such cases you may want to perform a fit that takes this advance information properly into account, neither completely freezing a parameter at a predetermined value (as in `lfit` §15.4) nor completely leaving it to be determined by the data set. The formalism for doing this is called "use of *a priori* covariances."

A related problem occurs in signal processing and control theory, where it is sometimes desired to "track" (i.e., maintain an estimate of) a time-varying signal in the presence of noise. If the signal is known to be characterized by some number of parameters that vary only slowly, then the formalism of *Kalman filtering* tells how the incoming, raw measurements of the signal should be processed to produce best parameter estimates as a function of time. For example, if the signal is a frequency-modulated sine wave, then the slowly varying parameter might be the instantaneous frequency. The Kalman filter for this case is called a *phase-locked loop* and is implemented in the circuitry of good radio receivers [3,4].

CITED REFERENCES AND FURTHER READING:

Huber, P.J. 1981, *Robust Statistics* (New York: Wiley). [1]

Launer, R.L., and Wilkinson, G.N. (eds.) 1979, *Robustness in Statistics* (New York: Academic Press). [2]

Bryson, A. E., and Ho, Y.C. 1969, *Applied Optimal Control* (Waltham, MA: Ginn). [3]

Jazwinski, A. H. 1970, *Stochastic Processes and Filtering Theory* (New York: Academic Press). [4]