A Fast Algorithm for the Minimum Covariance Determinant Estimator

Peter J. Rousseeuw

Department of Mathematics
and Computer Science
Universitaire Instelling Antwerpen
Universiteiteitsplein 1
B-2610 Wilrijk
Belgium
(*peter.rousseeuw@uia.ua.ac.be*)

Katrien Van Driessen

Faculty of Applied Economics
Universitaire Faculteiten
Sint Ignatius
Prinsstraat 13
B-2000 Antwerp
Belgium
(*katrien.vandriessen@ufsia.ac.be*)

The minimum covariance determinant (MCD) method of Rousseeuw is a highly robust estimator of multivariate location and scatter. Its objective is to find h observations (out of n) whose covariance matrix has the lowest determinant. Until now, applications of the MCD were hampered by the computation time of existing algorithms, which were limited to a few hundred objects in a few dimensions. We discuss two important applications of larger size, one about a production process at Philips with $n = 677$ objects and $p = 9$ variables, and a dataset from astronomy with $n = 137{,}256$ objects and $p = 27$ variables. To deal with such problems we have developed a new algorithm for the MCD, called FAST-MCD. The basic ideas are an inequality involving order statistics and determinants, and techniques which we call "selective iteration" and "nested extensions." For small datasets, FAST-MCD typically finds the exact MCD, whereas for larger datasets it gives more accurate results than existing algorithms and is faster by orders of magnitude. Moreover, FAST-MCD is able to detect an exact fit—that is, a hyperplane containing h or more observations. The new algorithm makes the MCD method available as a routine tool for analyzing multivariate data. We also propose the distance-distance plot (D-D plot), which displays MCD-based robust distances versus Mahalanobis distances, and illustrate it with some examples.

KEY WORDS: Breakdown value; Multivariate location and scatter; Outlier detection; Regression; Robust estimation.

It is difficult to detect outliers in p-variate data when $p > 2$ because one can no longer rely on visual inspection. Although it is still quite easy to detect a single outlier by means of the Mahalanobis distances, this approach no longer suffices for multiple outliers because of the masking effect, by which multiple outliers do not necessarily have large Mahalanobis distances. It is better to use distances based on robust estimators of multivariate location and scatter (Rousseeuw and Leroy 1987, pp. 265–269). In regression analysis, robust distances computed from the explanatory variables allow us to detect leverage points. Moreover, robust estimation of multivariate location and scatter is the key tool to robustify other multivariate techniques such as principal-component analysis and discriminant analysis.

Many methods for estimating multivariate location and scatter break down in the presence of $n/(p+1)$ outliers, where n is the number of observations and p is the number of variables, as was pointed out by Donoho (1982). For the breakdown value of the multivariate M-estimators of Maronna (1976), see Hampel, Ronchetti, Rousseeuw, and Stahel (1986, p. 296). In the meantime, several positive-breakdown estimators of multivariate location and scatter have been proposed. One of these is the minimum volume ellipsoid (MVE) method of Rousseeuw (1984, p. 877; 1985). This approach looks for the ellipsoid with smallest volume that covers h data points, where $n/2 \le h < n$. Its breakdown value is essentially $(n-h)/n$.

Positive-breakdown methods such as the MVE and least trimmed squares regression (Rousseeuw 1984) are increasingly being used in practice—for example, in finance, chemistry, electrical engineering, process control, and computer vision (Meer, Mintz, Rosenfeld, and Kim 1991). For a survey of positive-breakdown methods and some substantive applications, see Rousseeuw (1997).

The basic resampling algorithm for approximating the MVE, called MINVOL, was proposed by Rousseeuw and Leroy (1987). This algorithm considers a trial subset of $p+1$ observations and calculates its mean and covariance matrix. The corresponding ellipsoid is then inflated or deflated to contain exactly h observations. This procedure is repeated many times, and the ellipsoid with the lowest volume is retained. For small datasets it is possible to consider all subsets of size $p+1$, whereas for larger datasets the trial subsets are drawn at random.

Several other algorithms have been proposed to approximate the MVE. Woodruff and Rocke (1993) constructed algorithms combining the resampling principle with three heuristic search techniques—simulated annealing, genetic algorithms, and tabu search. Other people developed algorithms to compute the MVE exactly. This work started with the algorithm of Cook, Hawkins, and Weisberg (1992), which carries out an ingenious but still exhaustive search of all possible subsets of size h. In practice, this can be done for n up to about 30. Recently, Agulló (1996) developed an exact algorithm for the MVE that is based on a branch and bound procedure that selects the optimal subset without requiring the inspection of all subsets of size h. This is substantially faster and can be applied up to (roughly) $n \leq 100$ and $p \leq 5$. Because for most datasets the exact algorithms would take too long, the MVE is typically computed by versions of MINVOL—for example, in S-PLUS (see the function *cov.mve*).

Presently there are several reasons for replacing the MVE by the minimum covariance determinant (MCD) estimator, which was also proposed by Rousseeuw (1984, p. 877; 1985). The MCD objective is to find h observations (out of n) whose classical covariance matrix has the lowest determinant. The MCD estimate of location is then the average of these h points, and the MCD estimate of scatter is their covariance matrix. The resulting breakdown value equals that of the MVE, but the MCD has several advantages over the MVE. Its statistical efficiency is better because the MCD is asymptotically normal (Butler, Davies, and Jhun 1993), whereas the MVE has a lower convergence rate (Davies 1992). As an example, the asymptotic efficiency of the MCD scatter matrix with the typical coverage $h = .75n$ is 44% in 10 dimensions, and the reweighted covariance matrix with weights obtained from the MCD attains 83% of efficiency (Croux and Haesbroeck in press), whereas the MVE attains 0%. The MCD's better accuracy makes it very useful as an initial estimate for one-step regression estimators (Simpson, Ruppert, and Carroll 1992; Coakley and Hettmansperger 1993). Robust distances based on the MCD are more precise than those based on the MVE and hence better suited to expose multivariate outliers—for example, in the diagnostic plot of Rousseeuw and van Zomeren (1990), which displays robust residuals versus robust distances. Moreover, the MCD is a key component of the hybrid estimators of Woodruff and Rocke (1994) and Rocke and Woodruff (1996) and of high-breakdown linear discriminant analysis (Hawkins and McLachlan 1997).

In spite of all these advantages, until now the MCD has rarely been applied because it was harder to compute. In this article, however, we construct a new MCD algorithm that is actually much *faster* than any existing MVE algorithm. The new MCD algorithm can deal with a sample size n in the tens of thousands. As far as we know, none of the existing MVE algorithms can cope with such large sample sizes. Because the MCD now greatly outperforms the MVE in terms of both statistical efficiency and computation speed, we recommend the MCD method.

1. MOTIVATING PROBLEMS

Two recent problems will be shown to illustrate the need for a fast, robust method that can deal with many objects (n) and/or many variables (p) while maintaining a reasonable statistical efficiency.

Problem 1 (Engineering). We are grateful to Gertjan Otten for providing the following problem. Philips Mecoma (The Netherlands), is producing diaphragm parts for TV sets. These are thin metal plates, molded by a press. Recently a new production line was started, and for each of $n = 677$ parts, nine characteristics were measured. The aim of the multivariate analysis is to gain insight into the production process and the interrelations between the nine measurements and to find out whether deformations or abnormalities have occurred and why. Afterward, the estimated location and scatter matrix can be used for multivariate statistical process control.

Due to the support of Herman Veraa and Frans Van Dommelen (at Philips PMF/Mecoma, Product Engineering, P.O. Box 218, 5600 MD Eindhoven, The Netherlands), we obtained permission to analyze these data and to publish the results.

Figure 1 shows the classical Mahalanobis distance

$$\mathrm{MD}(\mathbf{x}_i) = \sqrt{(\mathbf{x}_i - \mathbf{T}_0)'\mathbf{S}_0^{-1}(\mathbf{x}_i - \mathbf{T}_0)} \qquad (1.1)$$

versus the index i, which corresponds to the production sequence. Here $\mathbf{x}_i$ is nine-dimensional, $\mathbf{T}_0$ is the arithmetic mean, and $\mathbf{S}_0$ is the classical covariance matrix. The horizontal line is at the usual cutoff value $\sqrt{\chi^2_{9,.975}} = 4.36$.

In Figure 1 it seems that most observations are consistent with the classical multivariate normal model, except for a few isolated outliers. This should not surprise us, even in the first experimental run of a new production line because the Mahalanobis distances are known to suffer from masking. That is, even if there were a group of outliers (here, deformed diaphragm parts) they would affect $\mathbf{T}_0$ and $\mathbf{S}_0$ in such a way as to become invisible in Figure 1. To further investigate these data, we need robust estimators $\mathbf{T}$ and $\mathbf{S}$, preferably with a substantial statistical efficiency so that we can be confident that any effects that may become visible are real and not due to the estimator's inefficiency. After developing the FAST-MCD algorithm, we will return to these data in Section 7.

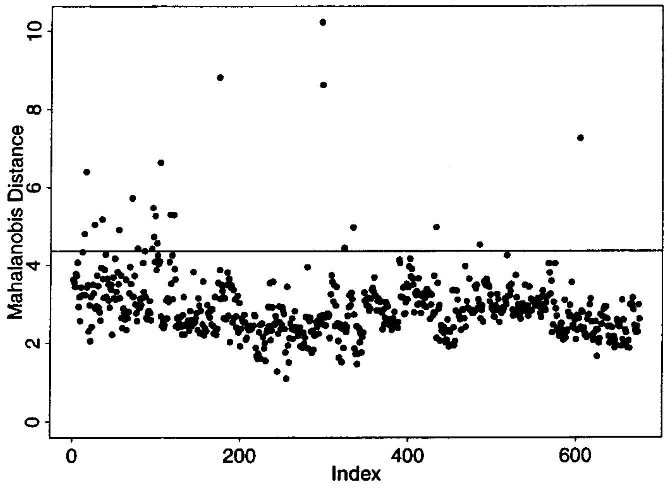

Figure 1. Plot of Mahalanobis Distances for the Philips Data.

Problem 2 (Physical Sciences). A group of astronomers at the California Institute of Technology are working on the Digitized Palomar Sky Survey (DPOSS); for a full description, see their report (Odewahn, Djorgovsky, Brunner, and Gal 1998). In essence, they make a survey of celestial objects (light sources) for which they record nine characteristics (such as magnitude, area, image moments) in each of three bands—blue, red, and near-infrared. They seek collaboration with statisticians to analyze their data, and gave us access to a part of their database, containing 137,256 celestial objects with all 27 variables.

We started by using quantile-quantile plots, Box–Cox transforms, selecting one variable out of three variables with near-perfect linear correlation, and other tools of data analysis. One of these avenues led us to study six variables (two from each band). Figure 2 plots the Mahalanobis distances (1.1) for these data (to avoid overplotting, Fig. 2 shows only 10,000 randomly drawn points from the entire plot). The cutoff is $\sqrt{\chi^2_{6,.975}} = 3.82$. In Figure 2 we see two groups of outliers with $\mathrm{MD}(\mathbf{x}_i) \approx 9$ and $\mathrm{MD}(\mathbf{x}_i) \approx 12$, plus some outliers still further away.

Returning to the data and their astronomical meaning, it turned out that these were all objects for which one or more variables fell outside the range of what is physically possible. So, the $\mathrm{MD}(\mathbf{x}_i)$ did help us to find outliers at this stage. We then cleaned the data by removing all objects with a physically impossible measurement, which reduced our sample size to 132,402. To these data we then again applied the classical mean $\mathbf{T}_0$ and covariance $\mathbf{S}_0$, yielding the plot of Mahalanobis distances in Figure 3.

Figure 3 looks innocent, like observations from the $\sqrt{\chi^2_6}$ distribution, as if the data would form a homogeneous population (which is doubtful because we know that the database contains stars as well as galaxies). To proceed further, we need high-breakdown estimates $\mathbf{T}$ and $\mathbf{S}$ and an algorithm that can compute them for $n = 132{,}402$. Such an algorithm will be constructed in the next sections.

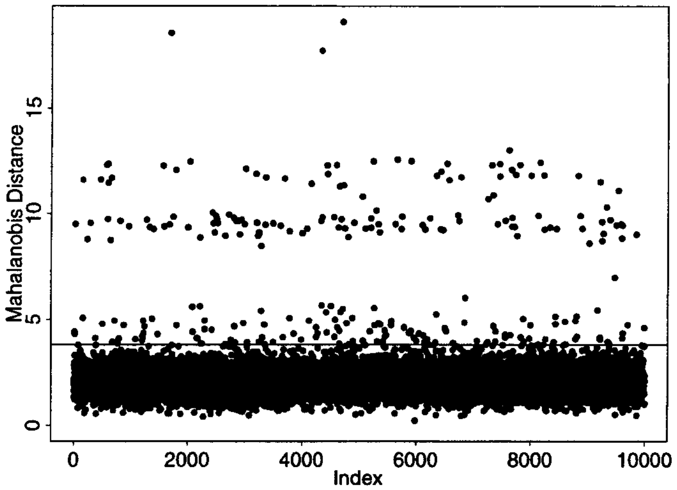

Figure 2. Digitized Palomar Data: Plot of Mahalanobis Distances of Celestial Objects, Based on Six Variables Concerning Magnitude and Image Moments.

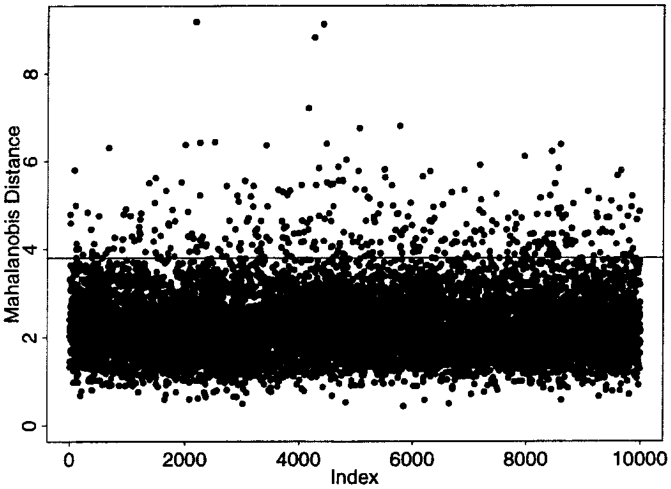

Figure 3. Digitized Palomar Data: Plot of Mahalanobis Distances of Celestial Objects as in Figure 2 After Removal of Physically Impossible Measurements.

2. BASIC THEOREM AND THE C-STEP

A key step of the new algorithm is the fact that, starting from any approximation to the MCD, it is possible to compute another approximation with an even lower determinant.

Theorem 1. Consider a dataset $X_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of p-variate observations. Let $H_1 \subset \{1, \ldots, n\}$ with $|H_1| = h$, and put $\mathbf{T}_1 := (1/h)\sum_{i \in H_1} \mathbf{x}_i$ and $\mathbf{S}_1 := (1/h)\sum_{i \in H_1}(\mathbf{x}_i - \mathbf{T}_1)(\mathbf{x}_i - \mathbf{T}_1)'$. If $\det(\mathbf{S}_1) \neq 0$, define the relative distances

$$d_1(i) := \sqrt{(\mathbf{x}_i - \mathbf{T}_1)'\mathbf{S}_1^{-1}(\mathbf{x}_i - \mathbf{T}_1)} \quad \text{for} \quad i = 1, \ldots, n.$$

Now take H_2 such that $\{d_1(i); i \in H_2\} := \{(d_1)_{1:n}, \ldots, (d_1)_{h:n}\}$, where $(d_1)_{1:n} \leq (d_1)_{2:n} \leq \cdots \leq (d_1)_{n:n}$ are the ordered distances, and compute $\mathbf{T}_2$ and $\mathbf{S}_2$ based on H_2. Then

$$\det(\mathbf{S}_2) \leq \det(\mathbf{S}_1)$$

with equality if and only if $\mathbf{T}_2 = \mathbf{T}_1$ and $\mathbf{S}_2 = \mathbf{S}_1$.

The proof is given in the Appendix. Although this theorem appears to be quite basic, we have been unable to find it in the literature.

The theorem requires that $\det(\mathbf{S}_1) \neq 0$, which is no real restriction because if $\det(\mathbf{S}_1) = 0$ we already have the minimal objective value. Section 5 will explain how to interpret the MCD in such a singular situation.

If $\det(\mathbf{S}_1) > 0$, applying the theorem yields $\mathbf{S}_2$ with $\det(\mathbf{S}_2) \leq \det(\mathbf{S}_1)$. In our algorithm we will refer to the construction in Theorem 1 as a *C-step,* where C stands for "concentration" because we concentrate on the h observations with smallest distances and $\mathbf{S}_2$ is more concentrated (has a lower determinant) than $\mathbf{S}_1$. In algorithmic terms, the C-step can be described as follows.

Given the h-subset H_{old} or the pair $(\mathbf{T}_{\text{old}}, \mathbf{S}_{\text{old}})$, perform the following:

1. Compute the distances $d_{\text{old}}(i)$ for $i = 1, \ldots, n$.
2. Sort these distances, which yields a permutation π for which $d_{\text{old}}(\pi(1)) \leq d_{\text{old}}(\pi(2)) \leq \cdots \leq d_{\text{old}}(\pi(n))$.

3. Put $H_{\text{new}} := \{\pi(1), \pi(2), \ldots, \pi(h)\}$.
4. Compute $\mathbf{T}_{\text{new}} := \text{ave}(H_{\text{new}})$ and $\mathbf{S}_{\text{new}} := \text{cov}(H_{\text{new}})$.

For a fixed number of dimensions p, the C-step takes only $O(n)$ time [because H_{new} can be determined in $O(n)$ operations without sorting all the $d_{\text{old}}(i)$ distances].

Repeating C-steps yields an iteration process. If $\det(\mathbf{S}_2) = 0$ or $\det(\mathbf{S}_2) = \det(\mathbf{S}_1)$, we stop; otherwise, we run another C-step yielding $\det(\mathbf{S}_3)$, and so on. The sequence $\det(\mathbf{S}_1) \geq \det(\mathbf{S}_2) \geq \det(\mathbf{S}_3) \geq \cdots$ is nonnegative and hence must converge. In fact, because there are only finitely many h-subsets, there must be an index m such that $\det(\mathbf{S}_m) = 0$ or $\det(\mathbf{S}_m) = \det(\mathbf{S}_{m-1})$, hence convergence is reached. (In practice, m is often below 10.) Afterward, running the C-step on $(\mathbf{T}_m, \mathbf{S}_m)$ no longer reduces the determinant. This is not sufficient for $\det(\mathbf{S}_m)$ to be the global minimum of the MCD objective function, but it is a necessary condition.

Theorem 1 thus provides a partial idea for an algorithm:

Take many initial choices of H_1 and apply C-steps
to each until convergence, and keep the
solution with lowest determinant. (2.1)

Of course, several questions must be answered to make (2.1) operational: How do we generate sets H_1 to begin with? How many H_1 are needed? How do we avoid duplication of work because several H_1 may yield the same solution? Can we do with fewer C-steps? What about large sample sizes? These matters will be discussed in the next sections.

Corollary 1. The MCD subset H of X_n is separated from $X_n \setminus H$ by an ellipsoid.

Proof. For the MCD subset H, and in fact any limit of a C-step sequence, applying the C-step to H yields H itself. This means that all $\mathbf{x}_i \in H$ satisfy $(\mathbf{x}_i - \mathbf{T})'\mathbf{S}^{-1}(\mathbf{x}_i - \mathbf{T}) \leq 9 = \{(\mathbf{x} - \mathbf{T})'\mathbf{S}^{-1}(\mathbf{x} - \mathbf{T})\}_{h:n}$, whereas all $\mathbf{x}_j \notin H$ satisfy $(\mathbf{x}_j - \mathbf{T})'\mathbf{S}^{-1}(\mathbf{x}_j - \mathbf{T}) \geq 9$. Take the ellipsoid $E = \{\mathbf{x}; (\mathbf{x} - \mathbf{T})'\mathbf{S}^{-1}(\mathbf{x} - \mathbf{T}) \leq 9\}$. Then $H \subset E$ and $X_n \setminus H \subset \text{closure}(E^c)$. Note that there is at least one point $\mathbf{x}_i \in H$ on the boundary of E, whereas there may or may not be a point $\mathbf{x}_j \notin H$ on the boundary of E.

The same result was proved by Butler et al. (1993) under the extra condition that a density exists. Note that the ellipsoid in Corollary 1 contains h observations but is not necessarily the smallest ellipsoid to do so, which would yield the MVE. We know of no technique like the C-step for the MVE estimator; hence, the latter estimator cannot be computed faster in this way.

Independently of our work, Hawkins and Olive (1999) discovered a version of Corollary 1 in the following form: "A necessary condition for the MCD optimum is that, if we calculate the distance of each case from the location vector using the scatter matrix, each covered case must have smaller distance than any uncovered case." This necessary condition could perhaps be called the "C-condition," as opposed to the C-step of Theorem 1, where we proved that a C-step always decreases $\det(\mathbf{S})$. In the absence of Theorem 1, Hawkins and Olive (1999) used the C-condition as a preliminary screen, followed by case swapping as a technique for decreasing $\det(\mathbf{S})$, as in the feasible solution approach (Hawkins 1994), which will be described in Section 6. The C-condition did not reduce the time complexity of this approach, but it did reduce the actual computation time in experiments with fixed n.

3. CONSTRUCTION OF THE NEW ALGORITHM

3.1 Creating Initial Subsets H_1

To apply the algorithmic concept (2.1), we first have to decide how to construct the initial subsets H_1. Let us consider the following two possibilities:

1. Draw a random h-subset H_1.
2. Draw a random $(p+1)$-subset J, and then compute $\mathbf{T}_0 := \text{ave}(J)$ and $\mathbf{S}_0 := \text{cov}(J)$. [If $\det(\mathbf{S}_0) = 0$, then extend J by adding another random observation, and continue adding observations until $\det(\mathbf{S}_0) > 0$.] Then compute the distances $d_0^2(i) := (\mathbf{x}_i - \mathbf{T}_0)'\mathbf{S}_0^{-1}(\mathbf{X}_i - \mathbf{T}_0)$ for $i = 1, \ldots, n$. Sort them into $d_0(\pi(1)) \leq \cdots \leq d_0(\pi(n))$ and put $H_1 := \{\pi(1), \ldots, \pi(h)\}$.

Option 1 is the simplest, whereas 2 starts like the MINVOL algorithm (Rousseeuw and Leroy 1987, pp. 259–260). It would be useless to draw fewer than $p + 1$ points, for then $\mathbf{S}_0$ is always singular.

When the dataset does not contain outliers or deviating groups of points, it makes little difference whether (2.1) is applied with 1 or 2. But because the MCD is a very robust estimator, we have to consider contaminated datasets in particular. For instance, we generated a dataset with $n = 400$ observations and $p = 2$ variables, in which 205 observations were drawn from the bivariate normal distribution

$$N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\right)$$

and the other 195 observations were drawn from

$$N_2\left(\begin{bmatrix} 10 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\right).$$

The MCD has its highest possible breakdown value when $h = [(n + p + 1)/2]$ (see Lopuhaä and Rousseeuw 1991), which becomes $h = 201$ here. We now apply (2.1) with 500 starting sets H_1. Using option 1 yields a resulting $(\mathbf{T}, \mathbf{S})$ whose 97.5% tolerance ellipse is shown in Figure 4(a). Clearly, this result has broken down due to the contaminated data. On the other hand, option 2 yields the result in Figure 4(b), which concentrates on the majority (51.25%) of the data.

The situation in Figure 4 is extreme, but it is useful for illustrative purposes. (The same effect also occurs for smaller amounts of contamination, especially in higher dimensions.) Approach 1 has failed because each random subset H_1 contains a sizable number of points from the majority group as well as from the minority group, which follows from the law of large numbers. When starting from a bad subset H_1, the iterations will not converge to the major solution. On the other hand, the probability of a $(p+1)$-subset without outliers is much higher, which explains why 2 yields many

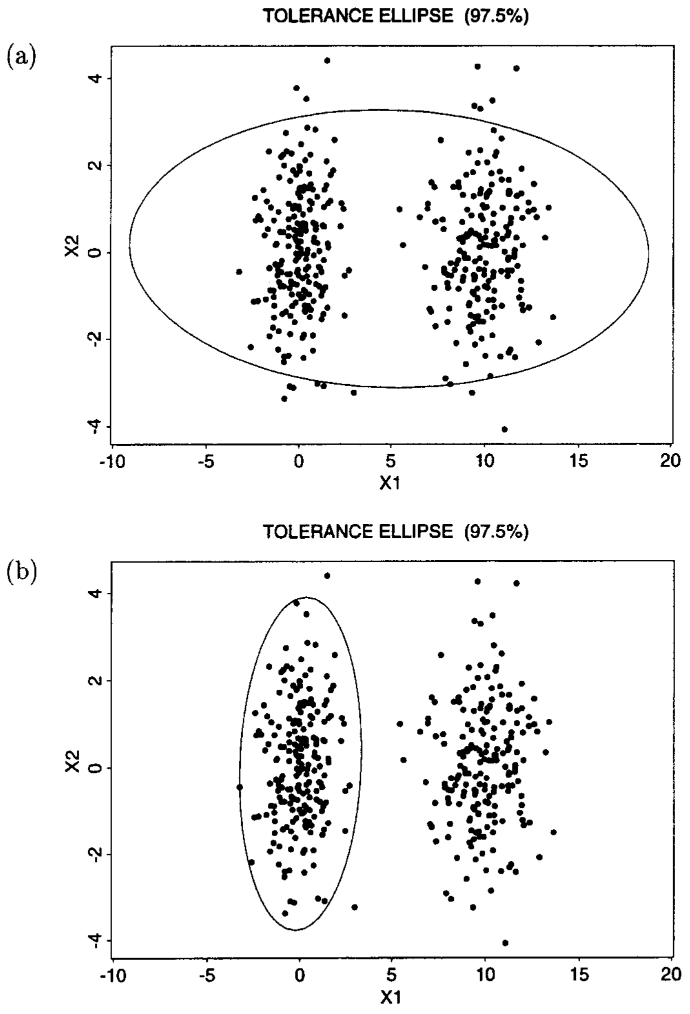

Figure 4. Results of Iterating C-Steps Starting From 500 Random Subsets H_1 of (a) Size h = 201 and (b) Size p+ 1 = 3.

subsets H_1 consisting of points from the majority and hence a robust result. From now on, we will always use 2.

Remark. For increasing n, the probability of having at least one "clean" $(p+1)$-subset among m random $(p+1)$-subsets tends to

$$1-(1-(1-\varepsilon)^{p+1})^m > 0, \tag{3.1}$$

where ε is the percentage of outliers. In contrast, the probability of having at least one clean h-subset among m random h-subsets tends to 0 because h increases with n.

3.2 Selective Iteration

Each C-step calculates a covariance matrix, its determinant, and all relative distances. Therefore, reducing the number of C-steps would improve the speed. But is this possible without losing the effectiveness of the algorithm? It turns out that often the distinction between robust solutions and nonrobust solutions already becomes visible after two or three C-steps. For instance, consider the data of Figure 4 again. The inner workings of the algorithm (2.1) are traced in Figure 5. For each starting subsample H_1, the determinant of the covariance matrix S_j based on $h = 201$ observations is plotted versus the step number j. The runs yielding a robust solution are shown as solid lines, whereas the dashed lines correspond to nonrobust results. To get a clear picture, Figure 5 only shows the first 100 starts. After two C-steps (i.e., for $j = 3$), many subsamples H_3 that will lead to the global optimum already have a rather small determinant. The global optimum is a solution that contains none of the 195 "bad" points. By contrast, the determinants of the subsets H_3 leading to a false classification are considerably larger. For that reason, we can save much computation time and still obtain the same result by taking just two C-steps and retaining only the (say, 10) best H_3 subsets to iterate further. Other datasets, also in more dimensions, confirm these conclusions. Therefore, from now on we will take only two C-steps from each initial subsample H_1, select the 10 different subsets H_3 with the lowest determinants, and only for these 10 we continue taking C-steps until convergence.

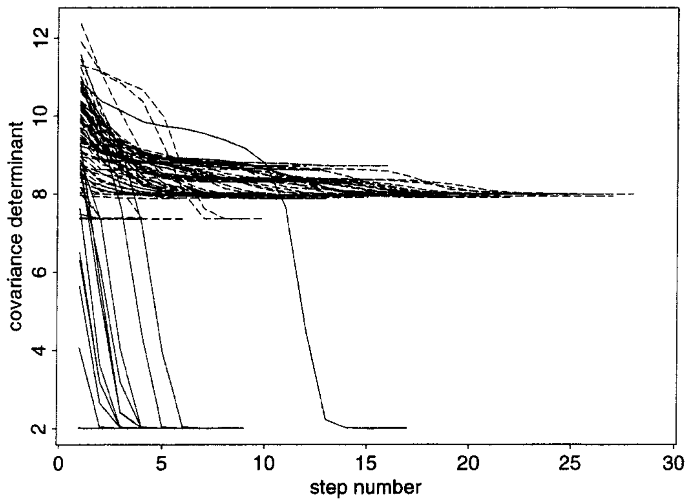

Figure 5. Covariance Determinant of Subsequent C-Steps in the Dataset of Figure 4. Each sequence stops when no further reduction is obtained.

3.3 Nested Extensions

For a small sample size n, the preceding algorithm does not take much time. But when n grows, the computation time increases, mainly due to the n distances that need to be calculated each time. To avoid doing all the computations in the entire dataset, we will consider a special structure. When $n > 1{,}500$, the algorithm generates a nested system of subsets that looks like Figure 6, where the arrows mean "is a subset of." The five subsets of size 300 do not overlap, and together they form the merged set of size 1,500,

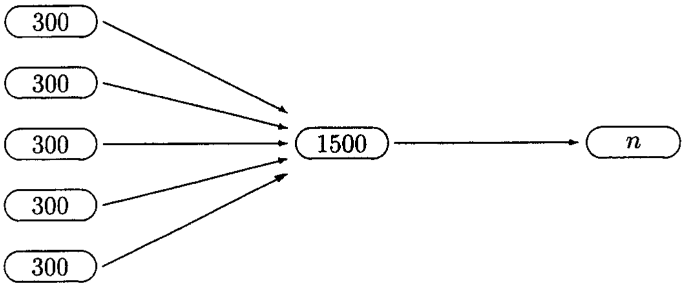

Figure 6. Nested System of Subsets Generated by the FAST-MCD Algorithm.

which in turn is a proper subset of the dataset of size n. [Already the algorithm of Woodruff and Rocke (1994) made use of partitioning for this purpose. The only difference with the nested extensions in Fig. 6 is that we work with two stages, hence our use of the word "nested," whereas Woodruff and Rocke partitioned the entire dataset, which yields more and/or larger subsets.] To construct Figure 6, the algorithm draws 1,500 observations, one by one, without replacement. The first 300 observations it encounters are put in the first subset, and so on. Because of this mechanism, each subset of size 300 is roughly representative for the dataset, and the merged set with 1,500 cases is even more representative.

When $n \leq 600$, we will keep the algorithm as in the previous section, while for $n \geq 1{,}500$ we will use Figure 6. When $600 < n < 1{,}500$, we will partition the data into at most four subsets of 300 or more observations so that each observation belongs to a subset and such that the subsets have roughly the same size. For instance, 601 will be split as $300 + 301$ and 900 as $450 + 450$. For $n = 901$, we use $300 + 300 + 301$, and we continue until $1{,}499 = 375 + 375 + 375 + 374$. By splitting 601 as $300 + 301$ we do not mean that the first subset contains the observations with case numbers $1, \ldots, 300$ but that its 300 case numbers were drawn randomly from $1, \ldots, 601$.

Whenever $n > 600$ (and whether $n < 1{,}500$ or not), our new algorithm for the MCD will take two C-steps from several starting subsamples H_1 within each subset, with a total of 500 starts for all subsets together. For every subset the best 10 solutions are stored. Then the subsets are pooled, yielding a merged set with at most 1,500 observations. Each of these (at most 50) available solutions $(\mathbf{T}_{\text{sub}}, \mathbf{S}_{\text{sub}})$ is then extended to the merged set. That is, starting from each $(\mathbf{T}_{\text{sub}}, \mathbf{S}_{\text{sub}})$, we continue taking C-steps, which now use all 1,500 observations in the merged set. Only the best 10 solutions $(\mathbf{T}_{\text{merged}}, \mathbf{S}_{\text{merged}})$ will be considered further. Finally, each of these 10 solutions is extended to the full dataset in the same way, and the best solution $(\mathbf{T}_{\text{full}}, \mathbf{S}_{\text{full}})$ is reported.

Because the final computations are carried out in the entire dataset, they take more time when n increases. In the interest of speed we can limit the number of initial solutions $(\mathbf{T}_{\text{merged}}, \mathbf{S}_{\text{merged}})$ and/or the number of C-steps in the full dataset as n becomes large.

The main idea of this subsection was to carry out C-steps in several nested random subsets, starting with small subsets of around 300 observations and ending with the entire dataset of n observations. Throughout this subsection, we have chosen several numbers such as five subsets of 300 observations, 500 starts, 10 best solutions, and so on. These choices were based on various empirical trials (not reported here). We implemented our choices as defaults so the user does not have to choose anything, but of course the user may change the defaults.

4. THE RESULTING ALGORITHM FAST-MCD

Combining all the components of the preceding sections yields the new algorithm, which we will call FAST-MCD. Its pseudocode looks as follows:

1. The default h is $[(n + p + 1)/2]$, but the user may choose any integer h with $[(n + p + 1)/2] \leq h \leq n$. The program then reports the MCD's breakdown value $(n - h + 1)/n$. If you are sure that the dataset contains less than 25% contamination, which is usually the case, a good compromise between breakdown value and statistical efficiency is obtained by putting $h = [.75n]$.

2. If $h = n$, then the MCD location estimate $\mathbf{T}$ is the average of the whole dataset, and the MCD scatter estimate $\mathbf{S}$ is its covariance matrix. Report these and stop.

3. If $p = 1$ (univariate data), compute the MCD estimate $(\mathbf{T}, \mathbf{S})$ by the exact algorithm of Rousseeuw and Leroy (1987, pp. 171–172) in $O(n \log n)$ time; then stop.

4. From here on, $h < n$ and $p \geq 2$. If n is small (say, $n \leq 600$), then

```
• repeat (say) 500 times:
    * construct an initial $h$-subset $H_1$ using method 2 in
      Subsection 3.1—that is, starting from a random
      $(p+1)$-subset;
    * carry out two C-steps (described in Sec. 2);
• for the 10 results with lowest $\det(\mathbf{S}_3)$:
    * carry out C-steps until convergence;
• report the solution $(\mathbf{T}, \mathbf{S})$ with lowest $\det(\mathbf{S})$.
```

5. If n is larger (say, $n > 600$), then

```
• construct up to five disjoint random subsets of size
  $n_{\text{sub}}$ according to Section 3.3 (say, five subsets of size
  $n_{\text{sub}} = 300$);
• inside each subset, repeat $500/5 = 100$ times:
    * construct an initial subset $H_1$ of size $h_{\text{sub}} =
      [n_{\text{sub}}(h/n)]$;
    * carry out two C-steps, using $n_{\text{sub}}$ and $h_{\text{sub}}$;
    * keep the 10 best results $(\mathbf{T}_{\text{sub}}, \mathbf{S}_{\text{sub}})$;
• pool the subsets, yielding the merged set (say, of size
  $n_{\text{merged}} = 1{,}500$);
• in the merged set, repeat for each of the 50 solutions
  $(\mathbf{T}_{\text{sub}}, \mathbf{S}_{\text{sub}})$:
    * carry out two C-steps, using $n_{\text{merged}}$ and $h_{\text{merged}} =
      [n_{\text{merged}}(h/n)]$;
    * keep the 10 best results $(\mathbf{T}_{\text{merged}}, \mathbf{S}_{\text{merged}})$;
• in the full dataset, repeat for the $m_{\text{full}}$ best results:
    * take several C-steps, using $n$ and $h$;
    * keep the best final result $(\mathbf{T}_{\text{full}}, \mathbf{S}_{\text{full}})$.
```

Here, m_{full} and the number of C-steps (preferably, until convergence) depend on how large the dataset is.

We will refer to the preceding as the FAST-MCD algorithm. Note that it is affine equivariant: When the data are translated or subjected to a linear transformation, the resulting $(\mathbf{T}_{\text{full}}, \mathbf{S}_{\text{full}})$ will transform accordingly. The computer program contains two more steps:

6. To obtain consistency when the data come from a multivariate normal distribution, we put

$$\mathbf{T}_{\mathrm{MCD}} = \mathbf{T}_{\mathrm{full}} \quad \text{and} \quad \mathbf{S}_{\mathrm{MCD}} = \frac{\mathrm{med}_i d^2_{(\mathbf{T}_{\mathrm{full}},\mathbf{S}_{\mathrm{full}})}(i)}{\chi^2_{p,.5}} \mathbf{S}_{\mathrm{full}}.$$

7. A one-step reweighted estimate is obtained by

$$\mathbf{T}_1 = \left(\sum_{i=1}^{n} w_i \mathbf{x}_i\right) \Big/ \left(\sum_{i=1}^{n} w_i\right)$$

$$\mathbf{S}_1 = \left(\sum_{i=1}^{n} w_i(\mathbf{x}_i - \mathbf{T}_1)(\mathbf{x}_i - \mathbf{T}_1)'\right) \Big/ \left(\sum_{i=1}^{n} w_i - 1\right),$$

where

$$w_i = 1 \quad \text{if} \quad d_{(\mathbf{T}_{\mathrm{MCD}},\mathbf{S}_{\mathrm{MCD}})}(i) \le \sqrt{\chi^2_{p,.975}}$$
$$= 0 \quad \text{otherwise.}$$

The program FAST-MCD has been thoroughly tested and can be obtained from our Web site *http://win-www.uia.ac.be/u/statis/index.html*. It has been incorporated into S-PLUS 4.5 (as the function "cov.mcd") and it is also in SAS/IML 7 (as the function "MCD").

5. EXACT FIT SITUATIONS

An important advantage of the FAST-MCD algorithm is that it allows for exact fit situations—that is, when h or more observations lie on a hyperplane. Then the algorithm still yields the MCD location $\mathbf{T}$ and scatter matrix $\mathbf{S}$, the latter being singular as it should be. From $(\mathbf{T}, \mathbf{S})$ the program then computes the equation of the hyperplane.

When n is larger than (say) 600, the algorithm performs many calculations on subsets of the data. To deal with the combination of large n and exact fits, we added a few steps to the algorithm. Suppose that, during the calculations in a subset, we encounter some $(\mathbf{T}_{\mathrm{sub}}, \mathbf{S}_{\mathrm{sub}})$ with $\det(\mathbf{S}_{\mathrm{sub}}) = 0$. Then we know that there are h_{sub} or more observations on the corresponding hyperplane. First we check whether h or more points of the full dataset lie on this hyperplane. If so, we compute $(\mathbf{T}_{\mathrm{full}}, \mathbf{S}_{\mathrm{full}})$ as the mean and covariance matrix of all points on the hyperplane, report this final result, and stop; if not, we continue. Because $\det(\mathbf{S}_{\mathrm{sub}}) = 0$ is the best solution for that subset, we know that $(\mathbf{T}_{\mathrm{sub}}, \mathbf{S}_{\mathrm{sub}})$ will be among the 10 best solutions that are passed on. In the merged set we take the set H_1' of the h_{merged} observations with smallest orthogonal distances to the hyperplane, and start the next C-step from H_1'. Again, it is possible that during the calculations in the merged set we encounter some $(\mathbf{T}_{\mathrm{merged}}, \mathbf{S}_{\mathrm{merged}})$ with $\det(\mathbf{S}_{\mathrm{merged}}) = 0$, in which case we repeat the preceding procedure.

As an illustration, the dataset in Figure 7 consists of 45 observations generated from a bivariate normal distribution, plus 55 observations that were generated on a straight line (using a univariate normal distribution). The FAST-MCD program (with default value $h = 51$) finds this line within .3 seconds. A part of the output follows:

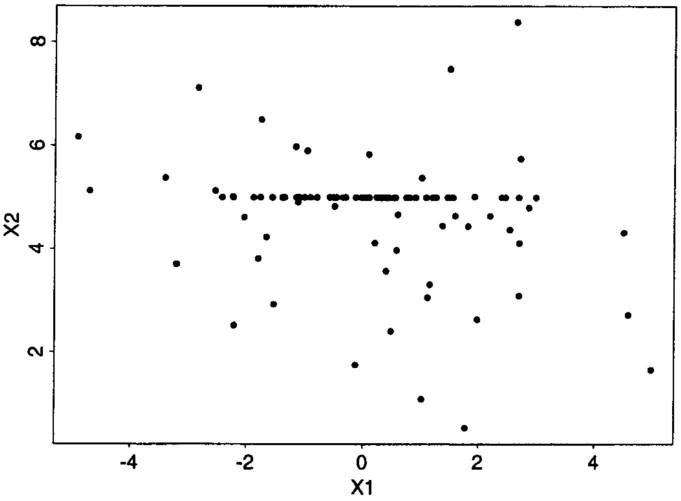

Figure 7. Exact Fit Situation (n = 100, p = 2).

There are 55 observations in the entire dataset of 100 observations that lie on the line with the equation

$$.000000(x_{i1} - m_1) + 1.000000(x_{i2} - m_2) = 0,$$

where the mean (m_1, m_2) of these observations is the MCD location:

$$\begin{matrix} .10817 \\ 5.00000 \end{matrix}$$

and their covariance matrix is the MCD scatter matrix

$$\begin{matrix} 1.40297 & .00000 \\ .00000 & .00000 \end{matrix}$$

Therefore, the data are in an "exact fit" position. In such a situation the MCD scatter matrix has determinant 0, and its tolerance ellipse becomes the line of exact fit.

If the original data were in p dimensions and it turns out that most of the data lie on a hyperplane, it is possible to apply FAST-MCD again to the data in this $(p-1)$-dimensional space.

6. PERFORMANCE OF FAST-MCD

To get an idea of the performance of the overall algorithm, we start by applying FAST-MCD to some small datasets taken from Rousseeuw and Leroy (1987). To be precise, these were all regression datasets, but we ran FAST-MCD only on the explanatory variables—that is, not using the response variable. The first column of Table 1 lists the name of each dataset, followed by n and p. We used the default value of $h = [(n + p + 1)/2]$. The next column shows the number of starting $(p + 1)$-subsets used in FAST-MCD, which is usually 500 except for two datasets in which the number of possible $(p + 1)$-subsets out of n was fairly small—namely, $\binom{12}{3} = 220$ and $\binom{18}{3} = 816$—so we used all of them.

The next entry in Table 1 is the result of FAST-MCD, given here as the final h-subset. Comparing these with the exact MCD algorithm of Agulló (personal communication, 1997), it turns out that these h-subsets do yield the exact global minimum of the objective function. The next column shows the running time of FAST-MCD in seconds on a Sun Ultra 2170. These times are much shorter than those of our MINVOL program for computing the MVE estimator.

Table 1. Performance of the FAST-MCD and FSA Algorithms on Some Small Datasets

Dataset	n	p	Starts	Best h-subset found	Time (seconds) FAST-MCD	Time (seconds) FSA
Heart	12	2	220	1 3 4 5 7 9 11	.6	.6
Phosphor	18	2	816	3 5 8 9 11 12 13 14 15 17	1.8	3.7
Stackloss	21	3	500	4 5 6 7 8 9 10 11 12 13 14 20	2.1	4.6
Coleman	20	5	500	2 3 4 5 7 8 12 13 14 16 17 19 20	4.2	8.9
Wood	20	5	500	1 2 3 5 9 10 12 13 14 15 17 18 20	4.3	8.2
Salinity	28	3	500	1 2 6 7 8 12 13 14 18 20 21 22 25 26 27 28	2.4	8.6
HBK	75	3	500	15 16 17 18 19 20 21 22 23 24 26 27 31 32 33 35 36 37 38 40 43 49 50 51 54 55 56 58 59 61 63 64 66 67 70 71 72 73 74	5.0	71.5

We may conclude that for these small datasets FAST-MCD gives very accurate results in little time.

Let us now try the algorithm on larger datasets, with $n \geq 100$. In each dataset, we generated over 50% of the points from the standard multivariate normal distribution $N_p(\mathbf{0}, \mathbf{I}_p)$, and the remaining points from $N_p(\boldsymbol{\mu}, \mathbf{I}_p)$, where $\boldsymbol{\mu} = (b, b, \ldots, b)'$ with $b = 10$. This is the model of "shift outliers." For each dataset, Table 2 lists n, p, the percentage of majority points, and the percentage of contamination. The algorithm always used 500 starts and the default value of $h = [(n + p + 1)/2]$.

The results of FAST-MCD are given in the next column, under "robust." Here "yes" means that the correct result is obtained—that is, corresponding to the first distribution [as in Fig. 4(b)]—whereas "no" stands for the nonrobust result, in which the estimates describe the entire dataset [as in Fig. 4(a)]. Table 2 lists data situations with the *highest* percentage of outlying observations still yielding the clean result with FAST-MCD, as was suggested by a referee. That is, the table says which percentage of outliers the algorithm can handle for given n and p. Increasing the number of starts only slightly improves this percentage. The computation times were quite low for the given values of n and p. Even for a sample size as high as 50,000, a few minutes suffice, whereas no previous algorithm we know of could handle such large datasets.

The currently most well-known algorithm for approximating the MCD estimator is the feasible subset algorithm (FSA) of Hawkins (1994). Instead of C-steps, it uses a different kind of steps, which for convenience we will baptize "I-steps," where the I stands for "interchanging points." An I-step proceeds as follows:

Given the h-subset H_{old} with its average $\mathbf{T}_{\text{old}}$ and its covariance matrix $\mathbf{S}_{\text{old}}$,

```
• repeat for each $i \in H_{\text{old}}$ and each $j \notin H_{\text{old}}$:
    * put $H_{i,j} = (H_{\text{old}} \setminus \{i\}) \cup \{j\}$
      (i.e., remove point $i$ and add point $j$);
    * compute $\Delta_{i,j} = \det(\mathbf{S}_{\text{old}}) - \det(\mathbf{S}(H_{i,j}))$;
• keep the $i'$ and $j'$ with largest $\Delta_{i',j'}$;
• if $\Delta_{i',j'} \leq 0$, put $H_{\text{new}} = H_{\text{old}}$ and stop;
• if $\Delta_{i',j'} > 0$, put $H_{\text{new}} = H_{i',j'}$.
```

An I-step takes $O(h(n-h)) = O(n^2)$ time because all pairs (i, j) are considered. If we would compute each $\mathbf{S}(H_{i,j})$ from scratch, the complexity would even become $O(n^3)$, but Hawkins (1994, p. 203) used an update formula for $\det(\mathbf{S}(H_{i,j}))$.

Table 2. Performance of the FAST-MCD and FSA Algorithms on Larger Datasets, With Time in Seconds

				FAST-MCD		FSA	
n	p	% $N_p(0, I_p)$	% $N_p(\mu, I_p)$	Robust	Time	Robust	Time
100	2	51	49	yes	2	yes	50
	5	53	47	yes	5	no	80
	10	63	37	yes	40	no	110
	20	77	23	yes	70	no	350
500	2	51	49	yes	7	no	2,800
	5	51	49	yes	25	no	3,800
	10	64	36	yes	84	no	4,100
	30	77	23	yes	695	no	8,300
1,000	2	51	49	yes	8	no	20,000
	5	51	49	yes	20	—	—
	10	60	40	yes	75	—	—
	30	76	24	yes	600	—	—
10,000	2	51	49	yes	9	—	—
	5	51	49	yes	25	—	—
	10	63	37	yes	85	—	—
	30	76	24	yes	700	—	—
50,000	2	51	49	yes	15	—	—
	5	51	49	yes	45	—	—
	10	58	42	yes	140	—	—
	30	75	25	yes	890	—	—

The I-step can be iterated: If $\det(\mathbf{S}_{\text{new}}) < \det(\mathbf{S}_{\text{old}})$, we can take another I-step with H_{new}; otherwise, we stop. The resulting sequence $\det(\mathbf{S}_1) \geq \det(\mathbf{S}_2) \geq \cdots$ must converge after a finite number of steps; that is, $\det(\mathbf{S}_m) = 0$ or $\det(\mathbf{S}_m) = \det(\mathbf{S}_{m-1})$, so $\det(\mathbf{S}_m)$ can no longer be reduced by an I-step. This is again a necessary (but not sufficient) condition for $(\mathbf{T}_m, \mathbf{S}_m)$ to be the global minimum of the MCD objective function. In our terminology, Hawkins's FSA algorithm can be written as follows:

- repeat many times:
  - * draw an initial h-subset H_1 at random;
  - * carry out I-steps until convergence, yielding H_m;
- keep the H_m with lowest $\det(\mathbf{S}_m)$;
- report this set H_m as well as $(\mathbf{T}_m, \mathbf{S}_m)$.

In Tables 1 and 2 we have applied the FSA algorithm to the same datasets as FAST-MCD, using the same number of starts. For the small datasets in Table 1, the FSA and FAST-MCD yielded identical results. This is no longer true in Table 2, where the FSA begins to find nonrobust solutions. This is because of the following:

1. The FSA starts from randomly drawn h-subsets H_1. Hence, for sufficiently large n all of the FSA starts are nonrobust, and subsequent iterations do not get away from the corresponding local minimum.

We saw the same effect in Section 3.1, which also explained why it is better to start from random $(p+1)$-subsets as in MINVOL and in FAST-MCD.

The tables also indicate that the FSA needs more time than FAST-MCD. In fact, time(FSA)/time(FAST-MCD) increases from 1 to 14 for n going from 12 to 75. In Table 2, the timing ratio goes from 25 (for $n = 100$) to 2,500 (for $n = 1{,}000$), after which we could no longer time the FSA algorithm. The FSA algorithm is more time-consuming than FAST-MCD because of the following:

2. An I-step takes $O(n^2)$ time, compared to $O(n)$ for the C-step of FAST-MCD.

3. Each I-step swaps only one point of H_{old} with one point outside H_{old}. In contrast, each C-step swaps $h - |H_{\text{old}} \cap H_{\text{new}}|$ points inside H_{old} with the same number outside of H_{old}. Therefore, more I-steps are needed, especially for increasing n.

4. The FSA iterates I-steps until convergence, starting from each H_1. On the other hand, FAST-MCD reduces the number of C-steps by the selective iteration technique of Section 3.2. The latter would not work for I-steps because of 3.

5. The FSA carries out all its I-steps in the full dataset of size n, even for large n. In the same situation, FAST-MCD applies the nested extensions method of Section 3.3, so most C-steps are carried out for $n_{\text{sub}} = 300$, some for $n_{\text{merged}} = 1{,}500$, and only a few for the actual n.

While this article was under review, Hawkins and Olive (1999) proposed an improved version of the FSA algorithm, as described at the end of our Section 2. To avoid confusion, we would like to clarify that the timings in Tables 1 and 2 were made with the original FSA algorithm described by Hawkins (1994), whereas the new version of FSA is substantially faster (although it retains the same computational complexity as the original FSA due to 2 in the preceding list).

In conclusion, we personally prefer the FAST-MCD algorithm because it is both robust and fast, even for large n.

7. APPLICATIONS

Let us now look at some applications to compare the FAST-MCD results with the classical mean and covariance matrix. At the same time we will illustrate a new tool, the *distance-distance plot.*

Example 1. We start with the coho salmon dataset (see Nickelson 1986) with $n = 22$ and $p = 2$, as shown in Figure 8(a). Each data point corresponds to one year. For 22 years the production of coho salmon in the wild was measured, in the Oregon Production Area. The x-coordinate is the logarithm of millions of smolts, and the y-coordinate is the logarithm of millions of adult coho salmon. We see that in most years the production of smolts lies between 2.2 and 2.4 on a logarithmic scale, whereas the production of adults lies between −1.0 and .0. The MCD tolerance ellipse excludes the years with a lower smolts production, thereby

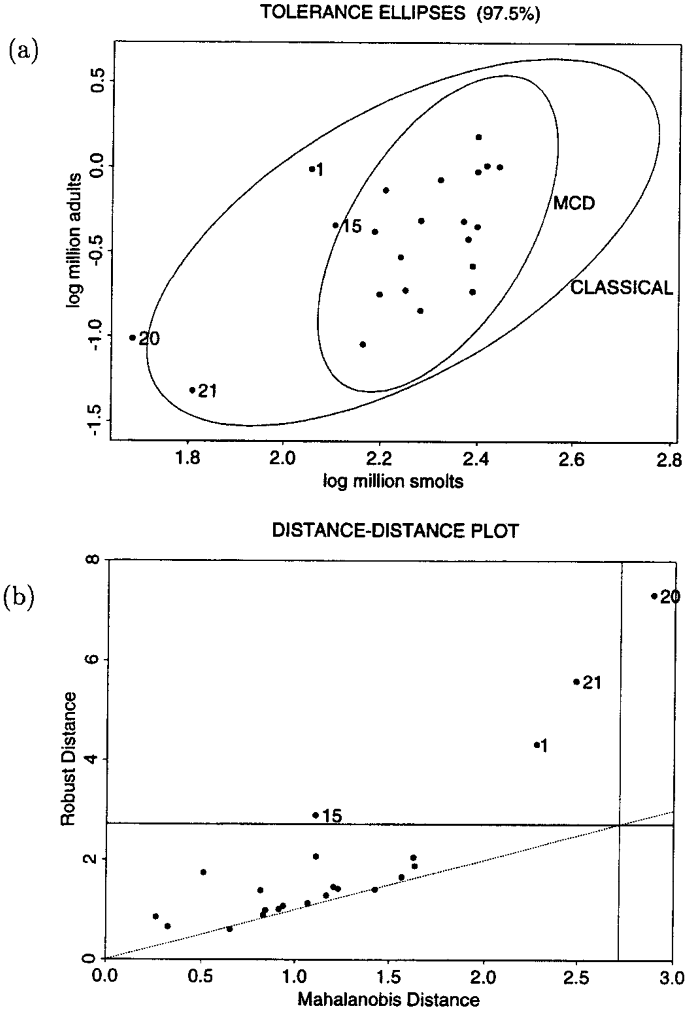

Figure 8. Coho Salmon Data: (a) Scatterplot With 97.5% Tolerance Ellipses Describing the MCD and the Classical Method; (b) Distance-Distance Plot.

marking them as outliers. In contrast, the classical tolerance ellipse contains nearly the whole dataset and thus does not detect the existence of far outliers.

Let us now introduce the distance-distance plot (D-D plot), which plots the robust distances (based on the MCD) versus the classical Mahalanobis distances. On both axes in Figure 8(b) we have indicated the cutoff value $\sqrt{\chi^2_{p,.975}}$ (here $p = 2$, yielding $\sqrt{\chi^2_{2,.975}} = 2.72$). If the data were not contaminated (say, if all the data would come from a single bivariate normal distribution), then all points in the D-D plot would lie near the dotted line. In this example many points lie in the rectangle where both distances are regular, whereas the outlying points lie higher. This happened because the MCD ellipse and the classical ellipse have a different orientation.

Naturally, the D-D plot becomes more useful in higher dimensions, where it is not so easy to visualize the dataset and the ellipsoids.

Problem 1 (continued). Next we consider Problem 1 in Section 1. The Philips data represent 677 measurements of metal sheets with nine components each, and the Mahalanobis distances in Figure 1 indicated no groups of outliers. The MCD-based robust distances RD($\mathbf{x}_i$) in Figure 9(a) tell a different story. We now see a strongly deviating group of outliers, ranging from index 491 to index 565. Something happened in the production process that was not visible from the classical distances shown in Figure 1. Figure 9(a) also shows a remarkable change after the first 100 measurements. These phenomena were investigated and interpreted by the engineers at Philips. Note that the D-D plot in Figure 9(b) again contrasts the classical and robust analysis. In Figure 9, (a) and (b), one can in fact see three groups—the first 100 points, those with index 491 to 565, and the majority.

Problem 2 (continued). We now apply FAST-MCD to the same n = 132,402 celestial objects with p = 6 variables as in Figure 3, which took only 2.5 minutes. (In fact, running the program on the same objects in all 27 dimensions took only 18 minutes!) Figure 10(a) plots the resulting MCD-based robust distances. In contrast to the homogeneous-looking Mahalanobis distances in Figure 3, the robust distances in Figure 10(a) clearly show that there is a majority with RD($\mathbf{x}_i$) $\leq \sqrt{\chi^2_{6,.975}}$ as well as a second group with RD($\mathbf{x}_i$) between 8 and 16. By exchanging our findings with the astronomers at the California Institute of Technology, we learned that the lower group consists mainly of stars and the upper group mainly of galaxies.

Our main point is that the robust distances separate the data in two parts and thus provide more information than

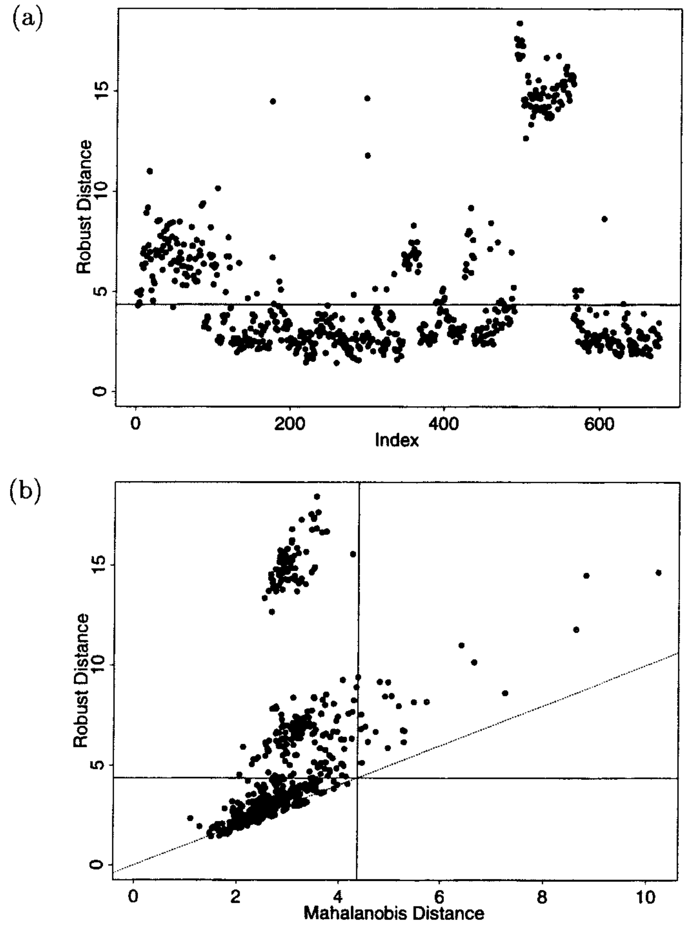

Figure 9. Philips Data: (a) Plot of Robust Distances; (b) Distance-Distance Plot.

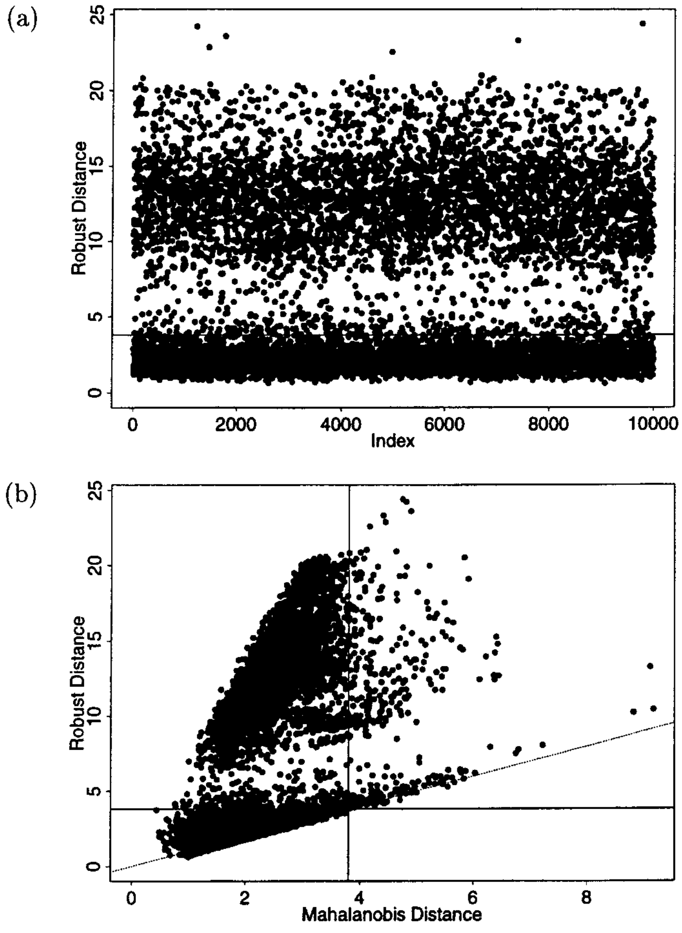

Figure 10. Digitized Palomar Data: (a) Plot of Robust Distances of Celestial Objects; (b) Their Distance-Distance Plot.

the Mahalanobis distances. That robust distances and Mahalanobis distances behave differently is illustrated in Figure 10(b), where we see the stars near the diagonal line and the galaxies above it.

Of course our analysis of these data was much more extensive and also used other data-analytic techniques not described here, but the ability to compute robust estimates of location and scatter for such large datasets was a key tool. Based on our work, these astronomers are thinking about modifying their classification of objects into stars and galaxies, especially for the faint light sources that are difficult to classify.

Example 2. We end this section by combining robust location/scatter with robust regression. The fire data (Andrews and Herzberg 1985) reported the incidences of fires in 47 residential areas of Chicago. One wants to explain the incidence of fire by the age of the houses, the income of the families living in them, and the incidence of theft. For this we apply the least trimmed squares (LTS) method of robust regression, with the usual value of $h = [(3/4)n] = 35$. In S-PLUS 4.5, the function "ltsreg" now automatically calls the function "cov.mcd," which runs the FAST-MCD algorithm, to obtain robust distances in x-space based on the MCD with the same h. Moreover, S-PLUS automatically provides the diagnostic plot of Rousseeuw and van Zomeren (1990), which plots the robust residuals versus the robust distances. For the fire data, this yields Figure 11, which shows the presence of one vertical outlier—that is, an observation with a small robust distance and a large LTS residual. We also see two bad leverage points—that is, observations (x, y) with outlying x and such that (x, y) does not follow the linear trend of the majority of the data. The other observations with robust distances to the right of the vertical cutoff line are good leverage points because they have small LTS residuals and hence follow the same linear pattern as the main group. In Figure 11 we see that most of these points are merely boundary cases, except for the two leverage points that are really far out in x-space.

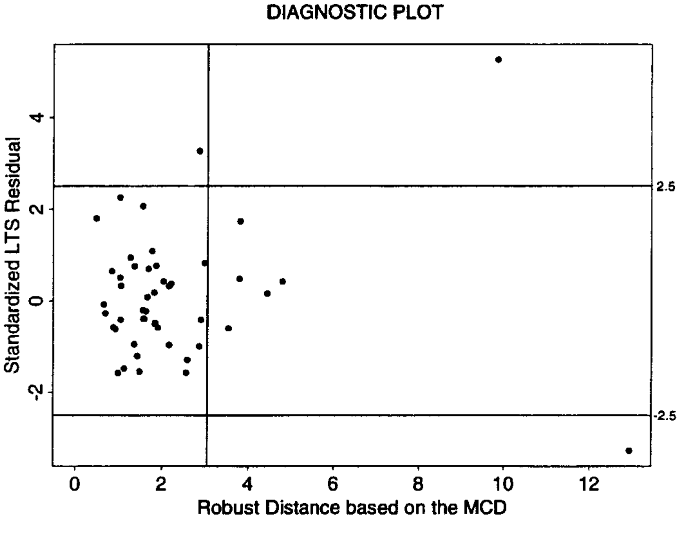

Figure 11. Diagnostic Plot of the Fire Dataset.

8. CONCLUSIONS

The algorithm FAST-MCD proposed in this article is specifically tailored to the properties of the MCD estimator. The basic ideas are the C-step (Theorem 1 in Sec. 2), the procedure for generating initial estimates (Sec. 3.1), selective iteration (Sec. 3.2), and nested extensions (Sec. 3.3). By exploiting the special structure of the problem, the new algorithm is faster and more effective than general-purpose techniques such as reducing the objective function by successively interchanging points. Simulations have shown that FAST-MCD is able to deal with large datasets while outrunning existing algorithms for MVE and MCD by orders of magnitude. Another advantage of FAST-MCD is its ability to detect exact fit situations.

Due to the FAST-MCD algorithm, the MCD becomes accessible as a routine tool for analyzing multivariate data. Without extra cost we also obtain the D-D plot, a new data display that plots the MCD-based robust distances versus the classical Mahalanobis distances. This is a useful tool to explore structure(s) in the data. Other possibilities include an MCD-based PCA and robustified versions of other multivariate analysis methods.

ACKNOWLEDGMENTS

We thank Doug Hawkins and José Agulló for making their programs available to us. We also dedicate special thanks to Gertjan Otten, Frans Van Dommelen, and Herman Veraa for giving us access to the Philips data and to S. C. Odewahn and his research group at the California Institute of Technology for allowing us to analyze their digitized Palomar data. We are grateful to the referees and *Technometrics* editors Max Morris and Karen Kafadar for helpful comments improving the presentation.

APPENDIX: PROOF OF THEOREM 1

Proof. Assume that $\det(\mathbf{S}_2) > 0$; otherwise the result is already satisfied. We can thus compute $d_2(i) = d_{(\mathbf{T}_2,\mathbf{S}_2)}(i)$ for all $i = 1, \ldots, n$. Using $|H_2| = h$ and the definition of $(\mathbf{T}_2, \mathbf{S}_2)$, we find

$$\begin{aligned}\frac{1}{hp}\sum_{i\in H_2} d_2^2(i) &= \frac{1}{hp}\operatorname{tr}\sum_{i\in H_2}(\mathbf{x}_i - \mathbf{T}_2)\mathbf{S}_2^{-1}(\mathbf{x}_i - \mathbf{T}_2)' \\ &= \frac{1}{hp}\operatorname{tr}\sum_{i\in H_2}\mathbf{S}_2^{-1}(\mathbf{x}_i - \mathbf{T}_2)(\mathbf{x}_i - \mathbf{T}_2)' \\ &= \frac{1}{p}\operatorname{tr}\mathbf{S}_2^{-1}\mathbf{S}_2 = \frac{1}{p}\operatorname{tr}(\mathbf{I}) = 1. \qquad \text{(A.1)}\end{aligned}$$

Moreover, put

$$\lambda := \frac{1}{hp}\sum_{i\in H_2} d_1^2(i) = \frac{1}{hp}\sum_{i=1}^{h}(d_1^2)_{i:n} \le \frac{1}{hp}\sum_{j\in H_1} d_1^2(j) = 1, \qquad \text{(A.2)}$$

where $\lambda > 0$ because otherwise $\det(\mathbf{S}_2) = 0$. Combining

(A.1) and (A.2) yields

$$\begin{aligned}\frac{1}{hp}\sum_{i\in H_2} d^2_{(\mathbf{T}_1,\lambda\mathbf{S}_1)}(i) &= \frac{1}{hp}\sum_{i\in H_2}(\mathbf{x}_i-\mathbf{T}_1)'\frac{1}{\lambda}S_1^{-1}(x_i-\mathbf{T}_1)\\ &= \frac{1}{\lambda hp}\sum_{i\in H_2} d_1^2(i) = \frac{\lambda}{\lambda} = 1.\end{aligned}$$

Grübel (1988) proved that $(\mathbf{T}_2,\mathbf{S}_2)$ is the unique minimizer of $\det(\mathbf{S})$ among all $(\mathbf{T},\mathbf{S})$ for which $(1/hp)\sum_{i\in H_2} d^2_{(\mathbf{T},\mathbf{S})}(i) = 1$. This implies that $\det(\mathbf{S}_2) \le \det(\lambda\mathbf{S}_1)$. On the other hand it follows from the inequality (A.2) that $\det(\lambda\mathbf{S}_1) \le \det(\mathbf{S}_1)$, hence

$$\det(\mathbf{S}_2) \le \det(\lambda\mathbf{S}_1) \le \det(\mathbf{S}_1). \qquad \text{(A.3)}$$

Moreover, note that $\det(\mathbf{S}_2) = \det(\mathbf{S}_1)$ if and only if both inequalities in (A.3) are equalities. For the first, we know from Grübel's result that $\det(\mathbf{S}_2) = \det(\lambda\mathbf{S}_1)$ if and only if $(\mathbf{T}_2,\mathbf{S}_1) = (\mathbf{T}_1,\lambda\mathbf{S}_1)$. For the second, $\det(\lambda\mathbf{S}_1) = \det(\mathbf{S}_1)$ if and only if $\lambda = 1$; that is, $\mathbf{S}_1 = \lambda\mathbf{S}_1$. Combining both yields $(\mathbf{T}_2,\mathbf{S}_2) = (\mathbf{T}_1,\mathbf{S}_1)$.

[Received December 1997. Revised March 1999.]

REFERENCES

Agulló, J. (1996), "Exact Iterative Computation of the Multivariate Minimum Volume Ellipsoid Estimator With a Branch and Bound Algorithm," in *Proceedings in Computational Statistics*, ed. A. Prat, Heidelberg: Physica-Verlag, pp. 175–180.

Andrews, D. F., and Herzberg, A. M. (1985), *Data*, Springer-Verlag, New York.

Butler, R. W., Davies, P. L., and Jhun, M. (1993), "Asymptotics for the Minimum Covariance Determinant Estimator," *The Annals of Statistics*, 21, 1385–1400.

Coakley, C. W., and Hettmansperger, T. P. (1993), "A Bounded Influence, High Breakdown, Efficient Regression Estimator," *Journal of the American Statistical Association*, 88, 872–880.

Cook, R. D., Hawkins, D. M., and Weisberg, S. (1992), "Exact Iterative Computation of the Robust Multivariate Minimum Volume Ellipsoid Estimator," *Statistics and Probability Letters*, 16, 213–218.

Croux, C., and Haesbroeck, G. (in press), "Influence Function and Efficiency of the Minimum Covariance Determinant Scatter Matrix Estimator," *Journal of Multivariate Analysis*.

Davies, L. (1992), "The Asymptotics of Rousseeuw's Minimum Volume Ellipsoid Estimator," *The Annals of Statistics*, 20, 1828–1843.

Donoho, D. L. (1982), "Breakdown Properties of Multivariate Location Estimators," unpublished Ph.D. qualifying paper, Harvard University, Dept. of Statistics.

Grübel, R. (1988), "A Minimal Characterization of the Covariance Matrix," *Metrika*, 35, 49–52.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986), *Robust Statistics, The Approach Based on Influence Functions*, New York: Wiley.

Hawkins, D. M. (1994), "The Feasible Solution Algorithm for the Minimum Covariance Determinant Estimator in Multivariate Data," *Computational Statistics and Data Analysis*, 17, 197–210.

Hawkins, D. M., and McLachlan, G. J. (1997), "High-Breakdown Linear Discriminant Analysis," *Journal of the American Statistical Association*, 92, 136–143.

Hawkins, D. M., and Olive, D. J. (1999), "Improved Feasible Solution Algorithms for High Breakdown Estimation," *Computational Statistics and Data Analysis*, 30, 1–11.

Lopuhaä, H. P., and Rousseeuw, P. J. (1991), "Breakdown Points of Affine Equivariant Estimators of Multivariate Location and Covariance Matrices," *The Annals of Statistics*, 19, 229–248.

Maronna, R. A. (1976), "Robust M-estimators of Multivariate Location and Scatter," *The Annals of Statistics*, 4, 51–56.

Meer, P., Mintz, D., Rosenfeld, A., and Kim, D. (1991), "Robust Regression Methods in Computer Vision: A Review," *International Journal of Computer Vision*, 6, 59–70.

Nickelson, T. E. (1986), "Influence of Upwelling, Ocean Temperature, and Smolt Abundance on Marine Survival of Coho Salmon (*Oncorhynchus Kisutch*) in the Oregon Production Area," *Canadian Journal of Fisheries and Aquatic Sciences*, 43, 527–535.

Odewahn, S. C., Djorgovski, S. G., Brunner, R. J., and Gal, R. (1998), "Data From the Digitized Palomar Sky Survey," technical report, California Institute of Technology.

Rocke, D. M., and Woodruff, D. L. (1996), "Identification of Outliers in Multivariate Data," *Journal of the American Statistical Association*, 91, 1047–1061.

Rousseeuw, P. J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871–880.

——— (1985), "Multivariate Estimation With High Breakdown Point," in *Mathematical Statistics and Applications, Vol B*, eds. W. Grossmann, G. Pflug, I. Vincze, and W. Wertz, Dordrecht: Reidel, pp. 283–297.

——— (1997), "Introduction to Positive-Breakdown Methods," in *Handbook of Statistics, Vol. 15: Robust Inference*, eds. G. S. Maddala and C. R. Rao, Amsterdam: Elsevier, pp. 101–121.

Rousseeuw, P. J., and Leroy, A. M. (1987), *Robust Regression and Outlier Detection*, New York: Wiley.

Rousseeuw, P. J., and van Zomeren, B. C. (1990), "Unmasking Multivariate Outliers and Leverage Points," *Journal of the American Statistical Association*, 85, 633–639.

Simpson, D. G., Ruppert, D., and Carroll, R. J. (1992), "On One-Step GM-estimates and Stability of Inferences in Linear Regression," *Journal of the American Statistical Association*, 87, 439–450.

Woodruff, D. L., and Rocke, D. M. (1993), "Heuristic Search Algorithms for the Minimum Volume Ellipsoid," *Journal of Computational and Graphical Statistics*, 2, 69–95.

——— (1994), "Computable Robust Estimation of Multivariate Location and Shape in High Dimension Using Compound Estimators," *Journal of the American Statistical Association*, 89, 888–896.