Computing high breakdown covariance matrices with missing data

???

???

???

July 2001
Abstract

Robust estimation of covariance matrices when some of the data at hand are missing is an important problem. It has been studied by Little and Smith (1987) and more recently by Author and Author (2001). The latter propose the use of high breakdown estimators and so-called hybrid algorithms (see e.g. Woodruff and Rocke 1994). In particular, the minimum volume ellipsoid of Rousseeuw (1984) is adapted to the case of missing data. To compute it, they use (a modified version of) the forward search algorithm (see e.g. Atkinson 1994).

In this paper, we propose to use instead a modification of the C-step algorithm proposed by Rousseeuw and Van Driessen (1999) which is actually a lot faster. Moreover, we conduct a simulation study to compare different robust estimators in terms of their efficiency and breakdown.

Keywords: C-step algorithm, minimum volume ellipsoid, outliers, robust statistics, S-estimators.
1 Introduction

Since the original works of Tukey (1960), Huber (1964) and Hampel (1971), robust statistics are nowadays widely used, and new or improved tools are continuously proposed. In this paper we focus on the robust estimation of location and scatter of a multivariate normal distribution with missing data.

The classical estimator of the covariance matrix, namely the maximum likelihood estimator (MLE) is very sensitive to model deviations. Indeed, ones shouldn’t forget that the common postulated models are only approximation of the reality. For example, there might be gross error in the data. Such errors appear as points lying very far from the core of the data and are extremely dangerous for classical statistical methods. It is therefore important to develop and use robust estimators for the mean and covariance of multivariate data since the latter can then be used in other analyses such as factor analysis. Yuan and Bentler (1998) showed that the influence of such data on covariance structure analysis is limited if the covariance matrix is robustly estimated.

The aim of robust statistics is thus to provide tools not only to asses the robustness properties of classical procedures but also to produce estimators and tests that are robust to model deviations. In the case of robust estimation of multivariate location and scatter, robust covariances have been first investigated by Maronna (1976). In particular, he shows that robust estimators based on a weighting scheme that is not redescending (no weight of zero), fails to be robust in high dimensions. This happens because for such estimators (including the classical MLE), their breakdown point, i.e. the maximal amount of model misspecification they can withstand before they “breakdown” or their bias becomes arbitrarily large, is at
most $1/(p + 1)$. When working in high dimension it is therefore crucial to consider high breakdown estimators.

The statistical literature contains several proposals for high breakdown estimators of the mean and covariance in multivariate data when it is suspected that the data contain outliers or extreme observations. A well known one is the minimum covariance determinant ($MCD$) of Rousseeuw (1984). When they are missing data only Little and Smith (1987) and Author and Author (2001) propose different solutions. In this paper we actually concentrate on the cases with missing data. We investigate different classical and robust estimators for the mean and covariance of multivariate normal data when some are missing. We propose the use of faster algorithms for their computation and compare them through extensive simulations in terms of their robustness properties when data are contaminated and also in terms of the speed of two different algorithms used to compute the robust estimators.

The paper is organised as follows. In section 2 we present the $MLE$ computed via the $EM$ algorithm as well as the robust modification proposed by Little and Rubin (1987). In section 3 we present the high breakdown estimators proposed by Author and Author (2001) with missing data. Section 4 presents a modification of a fast algorithm proposed by Rousseeuw and Van Driessen (1999), namely the $C$-step, to deal with missing data. In section 5, a simulation study is conducted to compare the speeds of the algorithms as well as the robustness properties of the different robust estimators.
2 Background

2.1 The EM algorithm

Classically one uses the EM algorithm (Dempster, Laird, and Rubin 1977) to estimate the MLE when there are missing data. The EM algorithm is an iterative procedure which switches from an E (expectation) step in which the expected values of the missing data are computed and an M (maximization) step in which the MLE estimating equations are solved.

The algorithm begins by defining the function

\[ Q(\theta \mid \theta^{(t)}) = E\left[ \log f(Y \mid \theta^{(t)}) Y_o, \theta \right] \]

where the observed data vector \( Y_o \) is a subset of the vector \( Y \) of size \( p \) with density \( f(Y, \theta) \). This is called the E-step. Next, we find \( \theta^{(t+1)} \) that maximize \( Q(\theta \mid \theta^{(t)}) \) with respect to \( \theta^{(t)} \). This is called the M-step. Replacing \( \theta^{(t)} \) by \( \theta^{(t+1)} \) and repeating the E and M-step produces a sequence of values of \( \theta \) that under appropriate conditions (Dempster et al. 1977) converges to the maximum likelihood estimates \( \hat{\theta} \).

If the multivariate normal model is postulated the E-step reduces to an expectation of the sufficient statistic (Dempster et al. 1977). Let \( \theta^{(t)} = (\mu^{(t)}, \Sigma^{(t)}) \) be the parameters estimates at iteration \( t \). Let in what follows, \( \Sigma \) denote the sum over \( i = 1, \ldots, n \). The E-step becomes

\[ E \left[ \sum y_{ij} \mid Y_o, \theta^{(t)} \right] = \sum y_{ij}^{(t)}, \quad j = 1, \ldots, p \]
\[
E \left[ \sum y_{ij} y_{ik} \mid Y_o, \theta^{(t)} \right] = \sum y_{ij}^{(t)} y_{ik}^{(t)} + C_{ijk}^{(t)}, \quad j, k = 1, \ldots, p
\]

where

\[
y_{ij}^{(t)} = \begin{cases} y_{ij}, & \text{if } y_{ij} \text{ is observed} \\ E \left[ y_{ij} \mid y_o, \theta^{(t)} \right], & \text{if } y_{ij} \text{ is missing} \end{cases}
\]

and

\[
C_{ijk}^{(t)} = \begin{cases} 0, & \text{if } y_{ij} \text{ and } y_{ik} \text{ are observed} \\ \text{Cov}(y_{ij} y_{ik} \mid y_o, \theta^{(t)}), & \text{if } y_{ij} \text{ or } y_{ik} \text{ are missing} \end{cases}
\]

The M-step becomes

\[
\mu_j^{(t+1)} = n^{-1} \sum y_{ij}^{(t)}, \\
\sigma_{jk}^{(t+1)} = n^{-1} \sum \left[ (y_{ij}^{(t)} - \mu_j^{(t+1)}) (y_{ik}^{(t)} - \mu_k^{(t+1)}) + C_{ijk}^{(t)} \right]
\]

With complete data it is well known that the MLE of mean and covariance is not robust. When there are missing data, the situation doesn’t change; see Author and Author (2001).

### 2.2 The ER algorithm

Little and Rubin (1987) propose to base the M-step on a robust estimator belonging to the general class of M-estimator (Huber 1981). They call the resulting procedure the ER
algorithm. They propose to downweight extreme observations and use squared Mahalanobis distances as a tool to detect them. They suggest to combine the usual E-step with the following modifications of the M-step (R-step):

\[
\mu_j^{(t+1)} = \frac{\sum w_i y_{ij}^{(t)}}{\sum w_i}
\]

and

\[
\sigma_{jk}^{(t+1)} = \frac{\sum w_i^2 \left[ \left( y_{ij}^{(t)} - \mu_j^{(t+1)} \right) \left( y_{ik}^{(t)} - \mu_k^{(t+1)} \right) \right]}{\sum w_i^2 - 1} + C_{ijk}^{(t)}
\]

with weights \( w_i = \omega(d_{io})/d_{io} \), where

\[
d_{io}^2 = (y_{io} - \mu_o)^T \Sigma_{oo}^{-1} (y_{io} - \mu_o)
\]

is the squared Mahalanobis distance computed using \( \mu_o \) and \( \Sigma_{oo} \), the partitions of respectively \( \mu \) and \( \Sigma \) corresponding to the observed part of \( y_i \) computed at step \( t \). Here \( \omega \) is a two-parameter weight defined by

\[
\omega(d_{io}) = \begin{cases} 
    d_{io} & \text{if } d_{io} \leq d_{io}^* \\
    d_{io} \exp\{-(d_i - d_{io})^2/2b_2^2 \} & \text{if } d_{io} > d_{io}^*
\end{cases}
\]

where \( d_{io}^* = \sqrt{p_i} + b_1/2 \), and \( p_i \) is the number of variables present for observation \( i \). The quantities \( b_1 \) and \( b_2 \) are to be specified by the analyst and Little and Smith (1987) proposed \( b_1 = 2 \) and \( b_2 = 1.25 \). If the case \( i \) is uncontaminated, the data are normal and missing values are missing at random, then (2) is asymptotically \( \chi^2_{p_i} \). The Wilson-Hilferty transformation of the chi-squared distribution yields \( (d_i^2/p_i)^{1/3} \sim N(1 - 2/(9p_i), 2/(9p_i)) \). Little and Smith
(1987) therefore also proposed a probability plot of

\[ Z_i = \frac{(d_i^2 / p_i)^{1/3} - 1 + 2/(9p_i)}{\sqrt{2/(9p_i)}} \]  

(3)

versus standard normal order statistics, that should reveal atypical observation.

The starting point of the \textit{ER} algorithm is the \textit{MLE} on the data where the missing ones have been replaced by the median of the corresponding observations. Although the \textit{ER} algorithm is relatively simple to implement, it suffer from an important drawback: its breakdown point is at most \(1/(p + 1)\) because it is based on a weighting scheme that is not redescending. This drawback will be highlighted by the simulation results. This means that if the proportion of outliers exceeds this value (or even is near it) the robust estimator is not robust anymore. Author and Author (2001) propose to modify the \textit{ER} algorithm by providing an high breakdown estimator as starting point instead of the usual starting point. Furthermore, since the weight in (1) are not directly applied to the correction matrix \(C\), Author and Author (2001) also propose to slightly modify (1) as

\[
\sigma_{jk}^{(t+1)} = \sum w_i^2 \left[ (y_{ij}^{(t)} - \mu_j^{(t+1)}) (y_{ik}^{(t)} - \mu_k^{(t+1)}) + C_{ijk}^{(t)} \right] / \sum w_i^2 - 1
\]

3 High breakdown estimators in incomplete data

To construct a high breakdown estimator of mean and covariance matrix in multivariate data when some are missing, Author and Author (2001) propose two strategies. The first one is to provide an high breakdown estimator such as the \textit{MCD} estimator as starting point
for the ER algorithm and the second is to adapt a high breakdown estimator such as an S-estimator (Rousseeuw and Yohai 1984) to incomplete data. For the latter we also need a high breakdown estimator as starting point.

3.1 The MCD estimator in incomplete data

Among the numerous high breakdown estimators a very well known is the minimum covariance determinant (MCD) (Rousseeuw and Leroy 1987). Its objective is to find $h$ observations (out of $n$) whose covariance matrix has the lowest determinant. The MCD mean estimator is then the sample mean of those $h$ points, and the MCD covariance estimator is their sample covariance matrix. One difficulty is that the MCD requires a decision on $h$. This choice will decide how many observations are to be trimmed. One way is to choose a value of $h$ that provides the maximum breakdown point. In this case, the maximal value of $h$ is given by (Rousseeuw and Leroy 1987):

$$h := \left\lfloor \frac{n + p + 1}{2} \right\rfloor$$

where $\lfloor x \rfloor$ denotes the integer part of $x$. But this is also the choice that gives the largest efficiency loss. So when we suspect that the sample is not heavily contaminated we can reasonably choose a larger value for the breakdown point of say 75% or 80% so we can take $h := \lfloor 0.75n \rfloor$ or $h := \lfloor 0.80n \rfloor$. It should also be stressed that generally (see Woodruff and Rocke 1993) the MCD estimator is used as a starting point for more efficient high breakdown estimators such as S-estimators (see below).

The time needed to run the MCD can be quite large. That’s why several authors
focus on the development of algorithms able to deal with this problem. Hawkins (1994) presents a feasible solution algorithm for the MCD which involve taking random “trial solutions” and refining each ones to a local optimum satisfying the condition for the MCD criterion. Atkinson (1993,1994) proposes the forward search algorithm which also permits the detection of multiple outliers. Author and Author (2001) use this algorithm in their work. More recently, Rousseeuw and Van Driessen (1999) present a new algorithm called C-step supposed to be even faster than the forward search algorithm and able to deal with very large data sets. In this section we briefly present the forward search algorithm and the modified version to deal with missing data of Author and Author (2001). In a separate section we present the C-step and its adaptation to missing data.

The forward search algorithm starts with a random subset of size $m_1 = p + 1$. The observation of the subset are then augmented by an amount $s$ (usually $s = 1$) in such a way that outliers are not likely to be included. Mahalanobis distances are both used to order observations for the forward search and to detect outliers.

Let $H_1$ be the random subset of size $m_1$ and compute the sample mean $\mu_1$, the covariance $\Sigma_1$ and the Mahalanobis distances. Another subset $H_2$ of size $m_2 = m_1 + s$ is chosen by means of the $m_2$ observations with the smallest Mahalanobis distances. Again $\mu_2$, $\Sigma_2$ and the Mahalanobis distances are computed but this time based on the new subset $H_2$. This procedure goes on until the size of the subset $H_k$ equals $n$. For each subset $H_k$ a criterion is computed. For the MCD estimator, this criterion is the determinant of the covariance matrix computed on the $h$ data with the smallest Mahalanobis distances. The forward search procedure is repeated for several randomly chosen initial subsets and the final MCD
estimator is then given by the MCD of the search with minimum determinant.

Adapting the procedure to the case of missing data is straightforward. Basically, the
EM algorithm is used to compute the sample mean and covariance when some observations
in the subset $H_k$ are missing. The only difference with the complete data case lies in the
calculation of the Mahalanobis distances. A straightforward adaptation is to restrict the
Mahalanobis distances to the observed variables as done in (2). To order the Mahalanobis
distances and take into account the non equal number of missing values for each observation,
the Mahalanobis distances are standardized by means of the Wilson-Hilferty transformation
given in (3).

3.2 The TBS estimator in incomplete data

The MCD estimator is known to be not very efficient and therefore is usually used as a
starting point for more efficient estimators. Among the latter, Author and Author (2001)
propose an adaptation of an $S$-estimator proposed by Rocke (1996). An $S$-estimator of
multivariate mean and covariance is defined in the complete case as the solution for $\mu$ and
$\Sigma$ which minimizes $|\Sigma|$ subject to (see Rousseeuw and Yohai 1984)

$$n^{-1} \sum \rho \left[ (y_i - \mu)^T \Sigma^{-1} (y_i - \mu) \right]^{1/2} = n^{-1} \sum \rho(d_i) = b_0$$

where $\rho$ is a rather general function and $b_0$ is a parameter to be fixed which determines the
desired breakdown point. A usual choice for the function $\rho$ is Tukey’s biweight (Beaton and
Tukey 1974). However, as Rocke (1996) argues, in high dimension it fails to downweight
outliers with large distances. This is measured by using the concept of asymptotic rejection
probability (ARP) which can be interpreted as the probability of an estimator, in large
samples under a reference distribution, to give a null (or nearly null) weight. Although the
ARP should be small for the sake of efficiency, it is useful to be able to downweight points
that are very improbable under the null model. Rocke (1996) shows that the ARP of the
S-estimator based on the biweight function tends to be 0 as the dimension \( p \) rise. This
means that points lying far away from the center of the data are not downweighted when \( p \) is
large. Therefore he proposes a modified biweight estimator, namely the TBS (for translated
biweight S-estimator), defined through,

\[
\rho(d; c, M) = \begin{cases} 
  d^2/2 & \quad 0 \leq d < M \\
  M^2/2 & \\
  -M^2(M^4 - 5M^2c^2 + 15c^4)/30c^4 & \\
  +d^2(0.5 + M^4/2c^4 - M^2/c^2) & \\
  +d^3(4M/3c^2 - 4M^3/3c^4) & \\
  +d^4(3M^2/2c^4 - 1/2c^2) & \\
  -4Md^5/5c^4 + d^6/6c^4 & \quad M \leq d \leq M + c \\
  M^2/2 + c(5c + 16M)/30 & \quad d > M + c 
\end{cases}
\]

\[
\frac{\partial}{\partial d} \rho(d; c, M) = \psi(d; c, M) = \begin{cases} 
  d & \quad 0 \leq d < M \\
  d \left(1 - \left(\frac{d-M}{c}\right)^2\right)^2 & \quad M \leq d \leq M + c \\
  0 & \quad d > M + c 
\end{cases}
\]

where the parameter \( c \) and \( M \) can be chosen to give the desired breakdown point and ARP.
Rocke (1996) discusses several choices for the function \( \rho \) defining the S-estimator. The striking feature is that whatever the choice, what remains very important is the starting point of the algorithm. Even the \( TBS \) estimator can lose its high breakdown properties if the starting point is not a high breakdown point estimator, as it would be the case if one choose the sample mean and covariance matrix on the whole data set. Author and Author (2001) therefore suggest to use the \( TBS \) estimator when working in high dimension with missing data use the \( MCD \) estimator as a starting point. They show that in the case of missing data, under the condition that the data are missing at random and under those given in Lopuhaä (1989), the resulting estimator is consistent. It is computed iteratively in the same way as the \( ER \) algorithm and it is therefore called the \( ERTBS \) estimator.

Given a good starting point \( \mu_0 \) and \( \Sigma_0 \), a chosen breakdown point \( \varepsilon^* \) and an \( ARP \) value \( \alpha \), the algorithm of the \( TBS \) is as follow. First the constants \( c \) and \( M \) are found by solving,

\[
\varepsilon^* \max_d \rho(d; c, M) = E_{\chi_p^2}[\rho(d; c, M)],
\]

\[
M + c = \sqrt{(\chi_p^2)^{-1}(1 - \alpha)}
\]

then, given these constants, the \( ERTBS \) estimates of \( \mu \) and \( \Sigma \) are found by using an iterative procedure like the \( ER \) algorithm where the \( R \)-step is:

\[\text{11}\]
\[
\begin{align*}
\mu^{(t+1)} &= \frac{\sum \nu_1 \left( \frac{d_{io}^{(t)}}{k} \right) y_i^{(t)}}{\sum \nu_1 \left( \frac{d_{io}^{(t)}}{k} \right)} \\
\Sigma^{(t+1)} &= \frac{p \sum \nu_1 \left( \frac{d_{io}^{(t)}}{k} \right) \left[ \left( y_i^{(t)} - \mu^{(t)} \right) \left( y_i^{(t)} - \mu^{(t)} \right)^T + C_{ji}^{(t)} \right]}{\sum \nu_2 \left( \frac{d_{io}^{(t)}}{k} \right)}
\end{align*}
\]

where

\[
\begin{align*}
\nu_1(d) &= \psi(d)/d, \\
\nu_2(d) &= d\psi(d)
\end{align*}
\]

and \( k \) is such that

\[
\frac{1}{n} \sum_{i=1}^{n} \rho \left( \frac{d_i}{k}; c, M \right) = b_0
\]

(4)

Rocke (1996) also proposes replacing the standardization step (4) with one that consists of equating the median of \( \rho(d_i; c, M) \) with the median under normality. In this case, the scaling factor \( k \) would be

\[
k = \frac{d_{(q)}}{\sqrt{\chi_p^2(q/(n + 1))}},
\]

where \( d_{(q)} \) denotes the \( q \)-th ordered distance, and \( q = \left\lfloor (n + p + 1)/2 \right\rfloor.\)
4 The C-step algorithm for the MCD estimator in incomplete data

A key idea of the C-step algorithm is the fact that starting from any approximation to the MCD, it is possible to find approximation with lower determinant. Indeed Rousseeuw and Van Driessen (1999) observed that from a subset $H_k$ of size $h$ in which $\mu$, $\Sigma$ and the Mahalanobis distances are computed, one can create a subset $H_{k+1}$ by taking among the $n$ observations the $h$ ones with the smallest Mahalanobis distances with the property that the determinant of $\Sigma$ based on $H_{k+1}$ is smaller. The underlying procedure is called the C-step by Rousseeuw and Van Driessen (1999). The initial subset is created by choosing randomly $p+1$ observations on which the Mahalanobis distances are computed to order the $n$ observations. The first $h$ ones define the initial subset $H_1$. If the determinant of $\Sigma$ based on the randomly chosen $p+1$ observations is nil, one adds one randomly chosen observation at the time until the determinant becomes positive.

The procedure to adapt the C-step algorithm to the case of missing data is basically the same as the one for the forward search algorithm. If for any subset $H_k$ there are missing values, we compute $\mu_k$ and $\Sigma_k$ with the EM algorithm. The Mahalanobis distances are also changed as in (2) and standardized using the Wilson-Hilferty transformation.

Each time one executes a C-step, one must compute a covariance matrix, a determinant and the Mahalanobis distances. This can be rather heavy if the data set is large. Therefore Rousseeuw and Van Driessen (1999) suggest a simplification: they show empirically that it is possible to make a distinction between good (robust) estimations and bad ones after
only two or three $C$-steps. This means that the $C$-step doesn’t need to be iterated until the covariance matrix with minimal determinant is found, the algorithm can switch to another initial subset. We found the same feature with our simulations.

Another particularity of the $C$-step algorithm is that it generates a nested system of subsets as shown in Figure 1. When $n > 1500$, the algorithm generates five subsets of size 300 which do not overlap. Together they form a merged set of size 1500 which in turn is a proper subset of the whole data. To construct this system, in the algorithm one draws 1500 observations without replacement. The first 300 observations are put in the first subset and so on. Because of this mechanism, each subset is roughly representative of the whole data set and the merged set is even more. When $600 < n < 1500$ the algorithm generate at most 4 subsets of 300 observations or more, so that each observation belongs to a subset and such that each subset have roughly the same size. Finally, when $n \leq 600$ no nested system is used and the data set is estimated as a whole.

In each case, the algorithm will execute two $C$-steps from several starting subsamples $H_1$.
for each created subset with a total of 500 starts for all the subsets together. In each subset, only the ten best results are kept (i.e. the subsets with lower determinant of their covariance matrix). Then the subsets are pooled to yield a set with at most 1500 observations. Each of these (at most 50) available solutions is then extended to the merged set. Again two $C$-steps are run on the merged set and the ten best results are kept. The $C$-steps are run on the latter until convergence for the final phase of the algorithm, i.e., extended to the whole data set. Through extensive simulations we now compare to forward search algorithm and the $C$-step algorithm for the computation of the MCD with missing data.

5 Simulation study

The aim of our simulation study is first to compare the behaviors of the different estimators proposed by Author and Author (2001) under different situations. Second, we also compare the speed of the two algorithms for the MCD with missing data in different settings, i.e. the (modified) forward search algorithm and our adaptation of the $C$-step. We will see that the $C$-step outperforms the forward search in all situations.

5.1 The design

For the problem of the generation of random samples, we used the proposition of Woodruff and Rocke (1993). The transformation of data that are $N(\mu, \Sigma)$ to $N(0, I)$ is affine, therefore, we assume without loss of generality that our uncontaminated data are $MN(0, I)$. To generate so-called shift-outliers (Woodruff and Rocke 1993), i.e. the ones which are the
hardest to find, we put the center of the contaminated data at a distances of $\sqrt{p} + \beta/\sqrt{2}$ where $\beta$ parametrizes the distances of the contamination from the main body of the data.

Table 1 shows the different value for $n$ and $p$ used in the simulations. For each case we set different value for the following parameters, (1) the percentage of observations $miss$ containing at least one missing value, (2) the percentage of contaminated data $\epsilon$ and (3) the value of $\beta$ in the formula of contamination. Table 2 summarizes the different combinations of the above parameters.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$p = 10$</th>
<th>$p = 20$</th>
<th>$p = 50$</th>
<th>$p = 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>100</td>
<td>200</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>200</td>
<td>400</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>500</td>
<td>600</td>
<td>1500</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: values for $n$ and $p$

Each algorithm for the robust estimators requires a decision on its initialization parameters. For the MCD estimator, both the forward search and the $C$-step require a decision on the proportion of data to be trimmed $h$ which was set at $h = [0.75n]$.

<table>
<thead>
<tr>
<th>$miss$</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>0</td>
<td>0.02</td>
<td>0.05</td>
</tr>
<tr>
<td>$\beta$</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: values for $miss$, $\epsilon$ and $\beta$

For the $ERTBS$ estimator different values for the breakdown point $bpd$ and the asymptotic rejection probability $arp$ were chosen and are given in Table 3.
Table 3: values for the $bdp$ and the $arp$

<table>
<thead>
<tr>
<th>$bdp$</th>
<th>$arp$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.001</td>
</tr>
<tr>
<td>0.4</td>
<td>0.005</td>
</tr>
<tr>
<td>0.5</td>
<td>0.01</td>
</tr>
</tbody>
</table>

All computational experiments were done on a SUN ultra5 with 128 MB of memory on a 400 Mhz processor. The core of the program was written in Fortran 77 and Splus was used as a front-end (to produce the various graphics). For all combinations of parameters, 1000 samples were generated.

5.2 Computational times

We describe now the time needed to run the two algorithms for the $MCD$. For each of the parameters given in Table 2 and for different sample sizes a time in second has been computed. Figure 2 show the results for the datasets of size $p = 10$ (for other sizes, the results are similar).

We notice the following features. First with both algorithms the time increases with the percentage of missing data for all combinations of the other parameters. Second, the percentage of contamination doesn’t seems to influence computational times for both algorithms. Third the sample size affects both algorithms but in a quite different manner. Indeed, when the sample size is doubled, the time for the $C$-step increases by a factor of about one third and the time for the forward search by a factor of four. This feature makes the $C$-step from 3 times to 22 times faster. One could however tunes the forward search algorithm through the choice of the increment factor $s$ but it is not known how much $s$ should be. Remind that each time an observation is added to the current sample, one needs to compute one
5.3 Comparing estimators

The aim of this subsection is to study the robustness properties of the different estimators proposed with incomplete data by means of simulations. It should be stressed that this exercise has not been done in Author and Author (2001). The estimators we consider here
are those presented in section 2 namely, the MLE computed via the EM algorithm (which is taken as a benchmark), the ER algorithm with the MLE as starting point (ER/MLE), the ER algorithm with the MCD as starting point computed using the forward search (ER/F), the ER algorithm with the MCD as starting point computed using the C-step (ER/C), the ERTBS algorithm with the MCD as starting point using the forward search algorithm (ERTBS/F) and the ERTBS algorithm with the MCD as starting point using the C-step algorithm (ERTBS/C).

![Figure 3: Estimators bias distribution for different amounts of contamination.](image_url)

The parameters are those defined in section 5.1. however we present here the results for only one of the combination since apart for the percentage of contamination, the results were similar for all other parameter settings. We chose (for the ERTBS) a bdp and arp of
respectively 0.3 and 0.01, a percentage of missing data of 10% and \( n \) and \( p \) at respectively 50 and 10. We also present the results for a value of \( n \) equal to 200.

We use boxplots to compare the estimators. Each boxplot includes estimates of (1) one of the element of the mean vector, (2) one of the diagonal elements of the covariance matrix and (3) one of the off-diagonal elements of the covariance matrix for the studied estimators. Normally one would expect a value close to 0 for (1) and (3) and a value close to 1 for (2). Only the results for \( \mu_1, \sigma_{11}, \) and \( \sigma_{12} \) are represented, since for other parameters, the same pattern is found. Figure 3 shows the distribution of the bias of the different estimators for each parameters (\( \mu_1, \sigma_{11}, \) and \( \sigma_{12} \)) and for different amounts of contamination. The MLE clearly fails even if the contamination is small (we also tried lower amounts, and the MLE was biased even with 2% contamination). However it is the most efficient with no
contamination but the efficiency loss for the robust estimators is quite small. The $ER/MLE$ is not able to take in account 10% of data contamination. The $ER$ and the $ERTBS$ with $MCD$ starts are very robust and can withstand at least 10% of data contamination. Finally, when computing the $ERTBS$ or the $ER$ there is no fundamental difference between the forward search algorithm and the $C$-step in term of resulting estimator. What the $C$-step gains in speed is not lost in accuracy.

If we want to see a difference between the $ERTBS$ and the $ER$ with $MCD$ start, we have to push the percentage of contamination up to 30%. We haven’t done a full coverage of such situation since its very unlikely that someone will want to study data sets with such a percentage of contamination. We show here an example based on one simulated dataset of size $n = 100$ and $p = 10$ with 30% of contamination and 10% of missing values. We plot the transformed Mahalanobis distances to see if the estimators do not breakdown and detect all the contaminated values. The results are displayed in Figures 4 and clearly the $ER/C$ breaks in such case but the $ERTBS/C$ does not since it is able to detect the 30 outliers (which were put in the first 30 observations).

6 Conclusion

In this paper we have considered high breakdown estimation of the mean and covariance of a multivariate normal distribution with missing data. We have proposed to use a modification of the $C$-step algorithm to compute the $MCD$ which is used as a starting point for the $ER$ of the $ERTBS$. We found through simulations that the computational speed is much more improved when on uses the $C$-step instead of the forward search. We have also conducted a
simulation study to compare the different high breakdown estimators computed in different ways. First we found that the results are independent of the chosen method to compute the MCD. As expected, the MLE breaks down at very low levels of contamination (2%). The \textit{ER} breaks down at at least 10% of contamination if the starting point is not the \textit{MCD} and breaks down at 30% anyway. The \textit{ERTBS} is the most robust overall and its variance is comparable to the one of the other estimators (including the \textit{MLE}) so that the efficiency loss in using this high breakdown estimator is very small. Finally, the program to compute the \textit{ERTBS} by means of the \textit{C-step} is available as an Splus library from the authors.
References


Author, . and . Author (2001). Robust correlation estimation with missing data. manuscript under revision, ???, ???


