Robust estimation of multivariate scatter in non-affine equivariant scenarios

by

Mikhail Danilov

Specialist, St.Petersburg State University, Russia, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Statistics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

January 2010

Abstract

We consider the problem of robust estimation of the scatter matrix of an elliptical distribution when observed data are corrupted in a cell-wise manner. The first half of the thesis develops a framework for dealing with data subjected to independent cell-wise contamination. Each data cell (as opposed to data case in traditional robustness) can be contaminated independently of the rest of the case. Instead of downweighting the whole case just because one or two components of it are contaminated we attempt to identify those affected cells, remove the offending values and treat them as missing at random for subsequent likelihood-based processing. We explore several variations of the detection procedure that takes into account the multivariate structure of the data and end up with a heuristic algorithm that can identify and remove a large proportion of dangerous independent contamination. Although there are not many existing methods to measure against, the proposed covariance estimate compares very favourably to naïve alternatives such as pairwise estimates or simple univariate Winsorising.

The cell-wise data corruption mechanism that we deal with in the second half of this thesis is missing data. Missing data on their own have been well studied and likelihood methods are well developed. The new setting that we are interested in is when missing data come together with the traditional case-wise contamination. Both issues have been studied extensively over that last few decades but little attention has been paid to how to address them both at the same time. We propose a modification of the S-estimate that allows robust estimation of multivariate location and scatter matrix in the presence of missing completely at random (MCAR) data. The method is based on the idea of the maximum likelihood of the observed data and extends it into the world of S-estimates. The estimate comes complete with the computation algorithm which is an adjusted version of the widely used Fast-S procedure. Some simulation results and applications to real datasets confirm the superiority of our method over available alternatives.

A quick investigation in the concluding chapter also suggests that combining the two main ideas presented in this thesis can yield an estimate that is robust against both types of contamination (case-wise and cell-wise) simultaneously.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my research adviser Dr Ruben Zamar for his continuing support and guidance during these years. Ruben introduced me to the world of Robust Statistics and has helped me achieve this important milestone I am at today.

I am also thankful to my supervisory committee members Dr Matías Salibián-Barrera and Dr Lang Wu for the insightful suggestions they have made during the preparation of this thesis. I very much appreciate the time that Dr Victoria Savalei and Dr Will Welch, my university examiners, and Dr Xuming He, my external examiner from the University of Illinois at Urbana-Champaign, took to read this document and provide thoughtful feedback that has appreciably improved it.

The Department of Statistics at UBC certainly could not function without all the hard work done by our most helpful office staff; neither could I. Thank you Christine, Peggy, Viena and Elaine for making the University appear to be such a well-oiled machine. And thank you for all the conversations and personal advice we have had over the years.

To my grandparents

Chapter 1

Introduction

1.1 Robust estimates

Classical parametric estimates rely heavily on the distributional assumptions of the model that data are assumed to be coming from. In the simpler case that we consider in this thesis, they assume that the data are independent and identically distributed (i.i.d.) drawn from a distribution parametrized by a vector of parameters. There are a number of ways in which these assumptions can be violated. For example, it might happen that (a) data are dependent; (b) the main distribution is not what it is assumed to be; or (c) data are not identically distributed but instead comes from a mixture of distributions. Most of the robust statistical theory focuses on the assumptions violation of the latter kind. We assume that the observed data are a mixture of *good data* (or *clean data*) and *contamination*. The distribution function of the observed data can be described as

$$F = (1 - \varepsilon)F_0(\boldsymbol{\theta}) + \varepsilon G, \tag{1.1}$$

where $F_0(\boldsymbol{\theta})$ is the distribution of the clean data parametrized by the parameter of interest $\boldsymbol{\theta}$, G is an arbitrary contaminating distribution that we do not have much interest in, and $\varepsilon \geq 0$ is the proportion of data cases affected by the contamination. Both G and ε are unknown but we have little interest in estimating them, at least at first. We will call the model in (1.1) the *classical contamination model*. This way of thinking about robustness was introduced by Tukey (1960).

The crucial assumption above is that there is a clear distinction between clean data cases and contaminated ones. We can model this mixture by introducing a Bernoulli random variable B with $\mathbf{P}\{B = 1\} = \varepsilon$. The observed random vector $\boldsymbol{X} \sim F$ is then given by

$$\boldsymbol{X} = (1 - B)\boldsymbol{X}_0 + B\boldsymbol{X}_G, \tag{1.2}$$

where $\boldsymbol{X}_0 \sim F_0$, $\boldsymbol{X}_G \sim G$ and B is independent from both of them. The indicator B can be seen as a latent variable that, albeit unobserved, has realized into 0 or 1 for each data case. Most robust estimates try, directly or otherwise, to guess what the value of B is for each case and include or exclude the observation into the analysis accordingly. It is important that there are enough good data to let the estimate decide which cases are clean and which are contaminated. The maximum fraction ε under which the estimate

can still estimate parameters of $F_0(\boldsymbol{\theta})$ instead of being misled by G is called its *breakdown point*. The best possible breakdown point is equal to 0.5 but not all estimates achieve it.

Robust data analysis does not necessarily try to completely eliminate cases coming from G and forget about them. On the contrary, by acknowledging that the observed data come from the mixture of two distributions, we can better estimate parameters of the core of the data, which will give us more power in identifying outlying cases coming from G. As a byproduct of robust estimation one can get more reliable information about the contamination indicator B. A standard example to illustrate this point proceeds by considering a univariate sample with a couple of large outliers. The usual technique for identifying outliers is to consider their z- or t-scores $z_i = (x_i - \hat{m})/\hat{s}$, where $\hat{m}$ and $\hat{s}$ are the estimated center and scale of the data. With suitably large outlying values the sample mean $\hat{m}$ and variance estimate $\hat{s}^2$ will be so much inflated that z-scores will remain relatively small even for the contaminated values. But if one were to use robust estimates for the mean and variance of the underlying population then the estimates $\hat{m}_{\text{rob}}$ and $\hat{s}^2_{\text{rob}}$ would be in the vicinity of their true values and the z-scores for the contaminated value would be large due to the large numerator and regular size denominator. Numerically it has been illustrated numerous times elsewhere (e.g. Maronna et al. (2006), p. 6) so we will not go into details.

An important property of the classical contamination model is that if a random variable $\boldsymbol{X}$ follows it then any transformation $g(\boldsymbol{X})$ will follow it as well (with different F_0 and G if the transformation g is non-trivial). A particularly important special case of such transformations are affine transformations such as $\boldsymbol{AX} + \boldsymbol{b}$ so we can say that the contamination model is affine invariant. If $B = 1$ for some cases of $\boldsymbol{X}$ then the same cases of $g(\boldsymbol{X})$ will be contaminated.

1.2 Independent contamination

A step away from the traditional robust thinking that we are taking in this thesis is to relax the assumption that each observed data case is either clean or contaminated without any intermediate possibilities. Multivariate datasets are often collected from different sources and later joined together to form a data table with one row per subject. Data values within one case can be of very different nature, e.g. chemical measurements for some variables and questionnaire instruments for other, and they can be erroneously measured or entered completely independently from each other. It might very well be the case that 99 values out of a hundred measurements for a patient are good and just one has been corrupted during data entry process. The classical contamination model will treat such a case as contaminated and will attempt to eliminate its influence on the estimate. Even if it succeeds in doing so, this obviously is a waste of data. When dimension is large and variables are contaminated independently with positive probability, the chances of having at least one bad data cell per case may easily exceed the breakdown point of

traditional robust estimates rendering them next to useless in such a situation. These types of contamination models have been defined and studied by Alqallaf et al. (2009) and we provide a formal definition in section 2.1.1. The main idea, however, is that we do not classify *cases* as clean or contaminated anymore — instead we classify data *cells*. Each cell will have its own unobserved indicator variable B. We will talk about the proportion ε of contaminated cells.

One substantial distinction of the *independent contamination model* from its classical counterpart is that it is not invariant to multivariate transformations of data anymore. Seemingly invertible transformations, such as affine transformation with non-singular $\boldsymbol{A}$ can spread contamination from one bad value to the whole case. Affine equivariance, which is a property of estimates to behave in a predictable fashion when data are subjected to an affine transformation, is generally considered to be a desirable and important characteristic for multivariate location-scatter and regression estimates (both robust and classical). Unfortunately, this property does not mix well with the lack of affine invariance of the independent contamination model. An affine-equivariant location-scatter estimate performs just as poorly on a dataset with only fraction ε of contaminated cells as it would on the same dataset subjected to an affine transformation with the contamination spread across the majority of the cells. This is obviously undesirable and prompts us to give up affine equivariance of estimates in hopes of improving their performance under the independent contamination model. This point is numerically illustrated in section 2.1.4.2. Chapter 2 of this thesis discusses various approaches to dealing with independent contamination which are all based on the common idea: identify the cells that are likely to be contaminated and remove them from the analysis.

Another aspect of poor data quality that can affect individual cells in the data matrix is *missing data*. Missing data can be seen as a special kind of contamination for which the information about the indicator B is known. If B is independent of the rest of the data, which is a typical assumption in robust theory, then we say that the data are *missing completely at random (MCAR)*. See Little and Rubin (2002) for more details on other types of missing data. When missing data is the only problem with a dataset, and especially if the MCAR assumption holds, Maximum Likelihood methods are well developed allowing us to estimate parameters of interest. This scenario is too easy and does not warrant a robust approach because the information on what cells are "contaminated" is given to us upfront. In Chapter 3 we propose an estimate for the location-scatter parameters of multivariate data subjected to two of the data corruption mechanisms at the same time: classical contamination (by case) and missing data (by cell). The addition of the cell-specific missing data to the classical contamination puts us the realm of non-affine equivariant data models and estimates.

Finally, in Chapter 4 we will consider how well our estimates can cope with the situation when all three of the data corrupting mechanisms are present: structurally contaminated cases, independently contaminated cells and missing data cells. There we will also outline

the future developments of this work that we hope to pursue in the future.

1.3 Mahalanobis distances

In this thesis we will focus on the problem of estimating scatter matrices of *elliptical distributions*. These distributions can be parametrized by their location vector $\boldsymbol{m}$ and scatter matrix $\boldsymbol{\Sigma}$ so that their probability density function is expressed as

$$f(\boldsymbol{x}) = \det(\boldsymbol{\Sigma})^{-\frac{1}{2}}\, h\big((\boldsymbol{x}-\boldsymbol{m})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m})\big)\,, \tag{1.3}$$

where $h(\cdot)$ is an appropriately scaled non-increasing integrable function on $[0, +\infty]$. The density of such a variable only depends on $\boldsymbol{x}$ through its so called *Mahalanobis distance*:

$$\mathrm{MD}^2(\boldsymbol{x}; \boldsymbol{m}, \boldsymbol{\Sigma}) = (\boldsymbol{x}-\boldsymbol{m})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m}). \tag{1.4}$$

We call it the squared Mahalanobis distance from $\boldsymbol{x}$ to $\boldsymbol{m}$ using covariance matrix $\boldsymbol{\Sigma}$. This can be seen as a generalization of the Euclidean distance adjusted for the fact that data variations in some directions are deemed to be less important than in others. Isolines of Mahalanobis distances and therefore of the density $f(\boldsymbol{x})$ look like concentric ellipsoids with axes parallel to the eigenvectors of $\boldsymbol{\Sigma}$ and radii proportional to the corresponding eigenvalues. Squared Mahalanobis distance can be seen as a measure of outlierness of the data case. Small distances correspond to more likely outcomes and vice versa.

The most common family among elliptical distributions is the multivariate normal with

$$h_{\mathcal{N}}(d) = (2\pi)^{-p/2} \exp(-d/2), \text{ for any } d \geq 0. \tag{1.5}$$

The lack of robustness of the Gaussian Maximum Likelihood Estimate (MLE) comes from the fact that $-\log(h_{\mathcal{N}}(d))$ is unbounded and increases too fast when $d \to \infty$ and thus gives too much leverage to outliers with large Mahalanobis distances. Most robust methods for elliptical data rely on Mahalanobis distances in one form or another and many use them directly to downweight potentially contaminated data points.

The main character of this thesis will be the concept of *partial Mahalanobis distance*. This is a Mahalanobis distance of a sub-vector of $\boldsymbol{x}$ with one or more components removed. If $\boldsymbol{X}$ follows an elliptical distribution centered at $\boldsymbol{m}$ with scatter matrix $\boldsymbol{\Sigma}$ then a sub-vector $\boldsymbol{X}_{-\{j_1,\ldots,j_m\}}$ with m components removed will follow the elliptical distribution from the same family centered around $\boldsymbol{m}_{-\{j_1,\ldots,j_m\}}$ with scatter matrix $\boldsymbol{\Sigma}_{-[j_1,\ldots,j_m]}$. The latter matrix is a sub-matrix of $\boldsymbol{\Sigma}$ from which columns and rows indexed $j_1, \ldots, j_m$ are removed. The partial Mahalanobis distance is then defined as

$$\mathrm{MD}^2_{j_1,\ldots,j_m}(\boldsymbol{x}) = \mathrm{MD}^2\big(\boldsymbol{x}_{-\{j_1,\ldots,j_m\}}; \boldsymbol{m}_{-\{j_1,\ldots,j_m\}}, \boldsymbol{\Sigma}_{-[j_1,\ldots,j_m]}\big)\,. \tag{1.6}$$

The idea of calculating Mahalanobis distances of a subvector of a data case for the purpose of computing robust estimates has been used before at least by Little and Smith (1987) and Cheng and Victoria-Feser (2002). To the best of our knowledge, the term "partial Mahalanobis distance" has not been used in statistical literature and we have not been able to find any results regarding its properties (such as our Lemma A.2).

In Chapter 3, partial Mahalanobis distances will be used as a proxy for the full distances which cannot be computed due to the presence of the missing data. In Chapter 2, partial Mahalanobis distances corresponding to a set of variables are used to assess whether there is a contaminated value among them.

When $\boldsymbol{X} \sim \mathcal{N}_p(\boldsymbol{m}, \boldsymbol{\Sigma})$ the squared Mahalanobis distances $\mathrm{MD}^2(\boldsymbol{X}; \boldsymbol{m}, \boldsymbol{\Sigma})$ to the true center with the true covariance matrix are distributed as χ^2_p. Partial Mahalanobis distances are thus distributed as χ^2_q, where q is the number of variables included in the sub-vector. Clean-data distributions of partial Mahalanobis distances in different dimensions will obviously have different magnitudes ($\mathbb{E}[\chi^2_q] = q$) but also slightly different shapes (converging to normal for large q) and coefficients of variation ($\mathbf{sd}[\chi^2_q] \,/\mathbb{E}[\chi^2_q] = \sqrt{2/q} \to 0$, as $q \to \infty$). All these issues will affect the construction of the estimates of $\boldsymbol{\Sigma}$ in both Chapters 2 and 3.

Chapter 2

Filtering approach to independent contamination

2.1 Introduction

2.1.1 General contamination model

For multivariate data, the classical contamination model assumes that each *case* (which we will also call *data point*) either comes from the population of interest or is completely erroneous and contains no useful information at all. For most standard robust procedures it is necessary that at least half of the points are clean because otherwise there is a possibility that the estimate will focus on estimating the contaminating distribution instead of the distribution of interest. In reality, however, it is often the case that different variables in the dataset are measured by different devices, recorded by different people and therefore may be contaminated by themselves independently of other measurements in the same row/case/subject. Typically there will be groups of variables that are either clean or contaminated together but a variety of configurations is possible. These contamination models have been formalized by Alqallaf et al. (2009) and formally we can express the observed random $p \times 1$ vector $\boldsymbol{X}$ as

$$\boldsymbol{X} = (1 - \boldsymbol{B})\boldsymbol{X}_0 + \boldsymbol{B}\boldsymbol{X}^*, \tag{2.1}$$

where $\boldsymbol{X}_0$ is a random vector following the distribution of interest, $\boldsymbol{X}^*$ is the contaminating distribution and $\boldsymbol{B}$ is a random diagonal matrix consisting of Bernoulli random variables with probabilities ε_j, for $j = 1, \ldots, p$. The dependence structure of $\text{diag}(\boldsymbol{B})$, in the general case, can be arbitrary. Model (2.1) encompasses a broad range of contaminating scenarios and includes the classical contamination model when all elements of $\text{diag}(\boldsymbol{B})$ are replicates of the same random indicator, e.i. $\boldsymbol{B} = B\boldsymbol{I}_p$, with B following a Bernoulli distribution.

2.1.2 Independent contamination model

Although certain aspects of the general model (2.1) can and have been studied by Alqallaf et al. (2009) it is undeniably complex and little can be done to solve it in its entirety. In this paper we will restrict attention to one special case that lies on the opposite end from the classical contamination model: the *independent contamination* (IC) model. It

assumes that all elements of diag($\boldsymbol{B}$) are mutually independent and so are the elements of $\boldsymbol{X}^*$. It means that every component of every realization of $\boldsymbol{X}$ comes either from $\boldsymbol{X}_0$ or from $\boldsymbol{X}^*$ independently from the remaining $p-1$ components. Under this contamination model, instead of contaminated data cases we will deal with contaminated *data values* (or *data cells*). Alternatively, this can be called *univariate contamination* to contrast it with the multivariate structural contamination of the classical robustness. It is possible that different variables have different probabilities of being contaminated but for simplicity we will often assume that they are all equal to a common $\varepsilon > 0$.

One of the most important distinctions between the classical and the independent contamination models is the interpretation of ε. An important quantity that is most directly affected by it is the number of clean data points available for analysis. In the former, a contaminated dataset will have proportion $(1-\varepsilon)$ of completely clean cases that can be used for estimation after discarding/downweighting the erroneous ε fraction. In the latter, however, on average only $(1-\varepsilon)^p$ of cases are entirely from $\boldsymbol{X}_0$. For larger p this number maybe be very small or even result in not having clean cases at all which rules out the possibility of simply discarding/downweighting all contaminated cases altogether. This means that classical robust methods are not well suited for estimation on independently contaminated data and new specialized methods are required. Alqallaf et al. (2009) have quantified this statement and we will return to it later in this section.

2.1.3 Effect of independent contamination on classical estimates

The negative effect of independent contamination on non-robust scatter estimates can be seen from two sides: (a) asymptotic or distributional point of view; and (b) its effect on finite samples.

The former is relatively easy to describe. The Gaussian MLE of the scatter matrix is the sample covariance matrix which treats multivariate data in a pairwise fashion. To understand the effect of contamination on such an estimate we only need to consider two independently contaminated variables X_1 and X_2 and investigate how their variances and covariances compare to the corresponding moments of clean X_{01} and X_{02}. As can be seen from the algebraic manipulations shown in Appendix A.1 the effect of independent contamination is the following:

1. Variances $\mathbf{Var}\, X_j$ overestimate true variances $\mathbf{Var}\, X_{0j}$ depending on the magnitude (second non-central moment) of $\boldsymbol{X}^*$. For typical contaminating distributions they overestimate the clean variances but can also slightly underestimate them if the contamination is, in fact, inlying instead of outlying. The bias in the variance estimates is unbounded.

2. Covariances $\mathbf{Cov}(X_1, X_2)$ underestimate true covariances $\mathbf{Cov}(X_{01}, X_{02})$ in absolute terms:

$$\mathbf{Cov}(X_1, X_2) = (1-\varepsilon)^2 \mathbf{Cov}(X_{01}, X_{02})$$

and the size of the effect does not depend on the distribution of $\boldsymbol{X}^*$ but only on the proportion of contaminated cells. It is the destroyed relationship between the two variables that matters but not the exact form of the disturbance.

3. Correlations are also guaranteed to get closer to zero but since they are affected by the estimated variances the effect depends on the magnitude of the contamination: large values of $\boldsymbol{X}^*$ will push estimated correlations closer to zero.

Although a little more algebraically involved, expected values of covariance estimates on *finite samples* under IC are similar to the asymptotic effects described above. The bias (to the covariance estimates) does not appear to be gross regardless of the value of contamination and one may even think that classical estimates of covariances are relatively robust under the independent contamination model. It seems obvious, however, that even only a pair of large erroneous values can completely destroy the estimate on a finite sample. The easiest way to describe this lack of robustness is by looking at the sample variance of the estimate when data are independently contaminated by large values.

The closed form algebra of dealing with fourth order moments (variances of variances) is not enlightening, so we show, in Table 2.1, some numerical results illustrating the dangers that untreated independent contamination poses to finite sample estimates. We have generated bivariate normal datasets of $n = 100$ centered at 0 with unit variances and correlation of 0.5. Then we contaminated them with a variety of point-mass independent contamination located at the values ranging from 1 (inlying) to 40 (grossly outlying) and the proportion of contamination ranging from 0 (clean data) to 20%. As contamination gets further away from the center, the sampling standard deviation of the covariance estimate increases quadratically and has no bound even when the fraction of contamination is small. Estimating a parameter with true value of 0.5 using an estimate with sampling standard deviation of 25 (or more), is useless for all practical purposes. This illustrates the importance of developing robust methods capable of dealing with this kind of contamination.

	Value of contamination					
ε	1.00	2.00	5.00	10.00	20.00	40.00
0	0.11	0.11	0.11	0.11	0.11	0.11
0.05	0.11	0.12	0.22	0.56	1.94	7.44
0.1	0.11	0.14	0.32	0.99	3.67	14.39
0.2	0.11	0.16	0.50	1.70	6.53	25.82

Table 2.1. Sampling standard deviations of pairwise covariance estimates based on independently contaminated sample of size 100. The true covariance is equal to 0.5.

Although we will only study multivariate scatter estimates, similar observations can be made for the linear regression problem. On average, independent contamination will result in OLS regression coefficients being pulled closer to zero or, in other words, weaker

estimated relationship between the response and predictor variables. On finite samples, however, behaviour can be erratic and very unstable if the contamination is far from the center of the data.

2.1.4 Affine-equivariant robust estimates

2.1.4.1 Lack of affine-equivariance

Affine equivariance is an important and often desirable property of statistical estimates. It guarantees that the estimate will behave in an easily predictable deterministic way if the data were subjected to an affine transformation. More specifically, a location estimate $\boldsymbol{T}(X)$ and a scatter estimate $\boldsymbol{\Sigma}(X)$ are called *affine equivariant* if, for any vector $\boldsymbol{a}$ and a non-singular square matrix $\boldsymbol{B}$,

$$\boldsymbol{T}(\boldsymbol{a} + \boldsymbol{B}X) = \boldsymbol{a} + \boldsymbol{B}\boldsymbol{T}(X) \tag{2.2}$$

$$\boldsymbol{\Sigma}(\boldsymbol{a} + \boldsymbol{B}X) = \boldsymbol{B}'\boldsymbol{\Sigma}\boldsymbol{B}, \tag{2.3}$$

where X is a $p \times n$ data matrix.

The concept of affine equivariance is, however, quite alien to the independent contamination model we are considering in this work. Non-trivial linear transformations will mix clean data cells with contaminated ones yielding a case with every single value having some contamination in it. If the transformation is well mixing and the dimension of the data is relatively high then even a relatively small fraction of independent contamination in the original data may yield a transformed data set with no clean cases at all.

This phenomenon of propagation of outliers has been well studied by Alqallaf et al. (2009) who showed that affine equivariant estimates (which is to say almost all standard robust procedures) of location and scatter do not perform well under independent contamination. They have proven that all reasonably behaved (i.e. δ-consistent, see the original paper for definition) location estimates that are affine equivariant will have a low breakdown point of, at maximum, $1 - 0.5^{1/p}$, which approaches zero as the dimension p increases.

The conceptual problem with such estimates is that they require a certain proportion (at least 50%) of data, that is cases, to be sampled from the uncontaminated distribution of interest. Under the independent contamination model the majority of cases will have at least some components contaminated and therefore won't satisfy this requirement.

2.1.4.2 Breakdown of scatter matrices

The theorem and the concept of δ-consistency in Alqallaf et al. (2009) is only formulated for location estimates but scatter matrices can be affected by independent contamination just as severely. We have conducted a simple simulation experiment to demonstrate

ε	log Condition Number				log Determinant			
	MCD	S-est	MLE	wins	MCD	S-est	MLE	wins
0	0.80	0.75	0.73	0.71	−0.64	−0.14	−0.13	−0.34
0.01	0.81	0.79	1.36	0.72	−0.50	1.35	**229.03**	0.22
0.02	0.82	0.81	1.08	0.71	−0.41	3.11	**236.44**	0.74
0.03	0.85	0.84	0.98	0.72	−0.30	5.36	**240.59**	1.28
0.04	0.87	0.87	0.88	0.72	−0.25	8.43	**243.44**	1.79
0.05	0.92	**24.71**	0.88	0.72	−0.22	**34.13**	**245.60**	2.31
0.06	0.94	**25.16**	0.83	0.73	−0.21	**59.44**	**247.34**	2.80
0.07	**25.39**	**25.52**	0.81	0.71	**24.72**	**69.05**	**248.81**	3.30
0.08	**25.72**	**25.79**	0.80	0.73	**49.97**	**91.80**	**250.05**	3.80
0.09	**25.91**	**25.94**	0.79	0.71	**75.33**	**114.85**	**251.13**	4.29
0.1	**26.07**	**26.09**	0.78	0.72	**100.65**	**126.96**	**252.08**	4.78

Table 2.2. Breaking down (boldface) of affine equivariant estimates of multivariate scatter. Fraction ε of contamination (with value 10^6) was placed independently on each of the 10 variables in the dataset.

this phenomenon. We generated 200 datasets of size 500 drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{10})$ and contaminated them with various amounts of point-mass independent contamination ranging from $\varepsilon = 0.01$ to $\varepsilon = 0.10$. The point mass was put at the large value of 10^6 to make the breakdown more obvious.

To assess the performance of the estimates we used the logarithm of the condition number of the estimated covariance matrix. This is a widely used quantity to evaluate deviations in the *shape* of the scatter estimate. It is recommended by Maronna et al. (2006) and we will use it extensively throughout this thesis. For these simulations we also report the logarithm of the determinant of the estimated matrix which naturally measures how well the *scale* of the matrix is estimated. It can be seen as a complementary measure to the condition number which effectively factors the scale out. Note that both logarithmic measures would be equal to 0 if applied to the true scatter matrix $\boldsymbol{I}_{10}$. For each value of ε the data generation and estimation was repeated 200 times and medians are reported.

The first thing one might wonder when looking at Table 2.2 is why the first line corresponding to $\varepsilon = 0$ does not consist of all zeros. The reason is that the numbers shown represent the expected values of the statistics computed on a finite ($n = 500$) data set. Although these statistics are *asymptotically* unbiased, they are not unbiased when the sample size is *finite*. Even when data are completely clean and the most efficient estimate (MLE) is used, the estimated covariance matrix is expected to have slightly smaller determinant and somewhat larger Condition Number. The expected value of the former is $n!/(n^p(n-p)!) < 1$, as shown, for example, in Goodman (1963). And the latter is very natural because the Condition Number is restricted to be larger than one by definition and therefore it can only have expected value equal to one if it has no variability at all, which, obviously, is not the case on finite data. In light of this, we will simply consider

the line $\varepsilon = 0$ as the benchmark to compare against.

As shown in Table 2.2, the Minimum Covariance Determinant (MCD) scatter estimate (computed with Fast-MCD algorithm by Rousseeuw and Van Driessen (1999)) has no difficulties dealing with contamination up to $\varepsilon = 0.06$, but at $\varepsilon = 0.07$ it breaks down and grossly overestimates both the condition number and the determinant. Note that with $\varepsilon = 0.07$ the average proportion of completely clean cases in a 10-dimensional data set is $(1-\varepsilon)^p = (1-0.07)^{10} = 0.48$, which is just under 50%. With MCD, the reason for this breakdown is really simple: the estimate is equal to the sample covariance matrix of the subset (of size $n/2$) of the data that minimizes its determinant. But now *all* subsets of size more than $n(1-\varepsilon)^p$ contain at least some contaminated cases that cause the covariance matrix to explode. The breaking down of the condition number is less intuitive and is due to the optimization part of the MCD — it chooses a subset that has as many variables uncontaminated as possible to minimize the determinant, which is approximately equal to the product of the diagonal elements in this uncorrelated case.

Similar things happen to the S-estimate except that it breaks down even earlier at $\varepsilon = 0.05$ which yields $n(1-0.05)^{10} = 0.6n$ completely clean cases on average. Since the breakdown point of S-estimates is known to be 50% this early failure is likely due to the insufficient search in the numerical algorithm (Fast-S).

For comparison we also show the results for the sample covariance, which explodes in size as soon as any contamination is introduced but seemingly preserves its perfectly round shape. The apparently good performance of the MLE in terms of condition numbers is solely due to the fact that the true covariance matrix in this experiment was identity and that the contamination was placed equally on all variables. Had any of these two conditions been violated, the shape of the MLE covariance would have been affected dramatically by the contamination.

On the other hand, this gross contamination can easily be dealt with by employing one of the simplest non-affine equivariant estimates — the Winsorised covariance. For each variable $j = 1, \ldots, p$, estimate its center $\hat{m}_j$ with median, and its standard deviation $\hat{s}_j$ with Median Absolute Deviation (MAD). Then replace each univariate value x in the jth variable by $x_{\text{wins}} = \hat{m}_j + \text{sign}(x - \hat{m}_j) \min\left(2.5, \frac{|x-\hat{m}_j|}{\hat{s}_j}\right) \hat{s}_j$. After all variables have been independently Winsorised, compute the sample covariance matrix of the modified dataset. The last column in Table 2.2 shows that the determinant of the Winsorised estimate is affected by the contamination but no abrupt breaking down takes place. Such a crude estimate performs reasonably well when the true covariance matrix is diagonal, i.e. the data have no structure, but in most practical situations it is far from being the best. In this paper we will review, propose and compare several approaches for dealing with independent contamination.

2.1.4.3 Practical difficulties

Dealing with non-affine equivariant models and estimates has one major inconvenience: there is a continuum of different data configurations (covariance structures) that cannot be reduced to one standardized scenario (e.g. identity covariance) without loss of generality. If we can fully study and understand how an affine-equivariant location-scatter estimate behaves under spherical data then we can easily deduce its properties under any elliptical distribution from the same family. This, unfortunately, is not the case with independent contamination and non-affine equivariant estimates. For example, as we showed before, the IC can only make estimated correlations closer to zero if an OLS estimate is used and therefore poses little threat to the shape estimates when true correlations are already equal to zero. Strong relationships in a highly correlated data can be completely destroyed by IC and the estimated shape may be very different from the true one.

One thing that will make our studies easier is the fact that independent contamination is invariant to non-mixing transformations of data which include translations and scaling. Whether a value is contaminated or clean, the addition or multiplication of it by anything (but 0) will preserve its status. Having this in mind, we will make sure that all our estimates are translation and scale equivariant, which, in turn, allows us to focus on data centered around zero with unit variances.

Understanding that the "degree of dependence" in the data is the key factor affecting the amount of damage that can be done by IC, in all our exploratory simulation studies we will focus on three typical scenarios that we will call "low", "medium" and "high" correlation. We have chosen and fixed three matrices for each dimension p in the following manner. First, generate an i.i.d. sample of size n from $\mathcal{N}_p(\mathbf{0}, \boldsymbol{I})$ with values of n depending on the desired degree of dependence. Then let the covariance matrix of future Monte Carlo data to be the sample covariance of this small sample. We used $n = p + 1$ to obtain "highly" correlated data, $n = p + 5$ for "medium" correlation and $n = 10p$ for "low". Of course, these matrices would be different every time a new sample is generated so we produced them once and saved for future use throughout the paper. Table 2.3 shows some characteristics of the pre-chosen simulation matrices for several typical dimensions.

Another compromise that needs to be made is the way in which we evaluate the performance of matrix estimates. We are faced by two problems: (a) the lack of affine-equivariance forces us to use various data scenarios but we would like to have performance measures comparable across them; (b) matrices are complicated and multidimensional but we would like to have a simple *scalar* measure evaluating their performance.

The first issue is addressed by standardizing covariance estimates $\hat{\boldsymbol{\Sigma}}$ using the known target value of $\boldsymbol{\Sigma}$. When estimates are fully affine equivariant this step guarantees that our performance measure is affine invariant. In our case, however, it is impossible to have a fully affine invariant measure and this step will only make the numbers *comparable* across different covariance scenarios. The natural standardization implied by affine invariance

		max EV	min EV	$\det^{1/p}$	avg cor	max cor
	low	1.3	0.72	0.98	0.099	0.19
$p = 4$	med	2.4	0.15	0.63	0.41	0.76
	high	2.3	0.036	0.47	0.42	0.77
	low	1.5	0.57	0.96	0.069	0.19
$p = 10$	med	2.5	0.045	0.61	0.23	0.56
	high	3.2	0.00068	0.22	0.3	0.76
	low	1.6	0.50	0.95	0.055	0.19
$p = 20$	med	2.8	0.0042	0.53	0.143	0.59
	high	3.6	0.0013	0.37	0.167	0.66

Table 2.3. Summary of the selected covariance matrices used in all further simulations. Maximum and minimum eigenvalues, a power of the determinant and average and maximum absolute correlations are shown.

considerations is given by the sandwich formula:

$$\text{std}^*(\hat{\boldsymbol{\Sigma}}) = (\boldsymbol{A}^{-1})'\hat{\boldsymbol{\Sigma}}\boldsymbol{A}^{-1}, \tag{2.4}$$

where $\boldsymbol{A}$ is the matrix used to linearly transform spherical data into elliptical. In particular, $\boldsymbol{A}'\boldsymbol{A} = \boldsymbol{\Sigma}$. One problem is that when only $\boldsymbol{\Sigma}$ is known then its square root, that is matrix $\boldsymbol{A}$, is not fully defined. For any $\boldsymbol{A}$ such that $\boldsymbol{A}'\boldsymbol{A} = \boldsymbol{\Sigma}$, and any orthogonal matrix $\boldsymbol{R}$, their superposition $\boldsymbol{R}\boldsymbol{A}$ is also a square root of $\boldsymbol{\Sigma}$ because $(\boldsymbol{R}\boldsymbol{A})'(\boldsymbol{R}\boldsymbol{A}) = \boldsymbol{A}'(\boldsymbol{R}'\boldsymbol{R})\boldsymbol{A} = \boldsymbol{A}'\boldsymbol{A} = \boldsymbol{\Sigma}$. This means that standardizing operation is not fully defined in (2.4) as a function of $\boldsymbol{\Sigma}$. Requiring the square root matrix $\boldsymbol{A}$ to be symmetric is a common remedy to this problem. Although it makes the procedure well defined, it remains somewhat arbitrary in the sense that the symmetric square root matrix is not necessarily the same as the matrix that was originally used to transform the spherical data into elliptical. Another problem with using the obvious standardization in (2.4) is that it puts a little bit of extra computational burden on us. Instead, we will use the simplified standardization formula:

$$\text{std}_{\boldsymbol{\Sigma}}(\hat{\boldsymbol{\Sigma}}) = \boldsymbol{\Sigma}^{-1}\hat{\boldsymbol{\Sigma}}. \tag{2.5}$$

It can easily be seen that eigenvalues of matrices in expressions (2.4) and (2.5) are the same. As will be explained in the next paragraph, our performance scalar measure is solely based on eigenvalues of the estimated matrix and therefore both standardizations are equivalent for our purposes. The latter, however, does not involve picking an arbitrary square root of $\boldsymbol{\Sigma}$ and takes one matrix multiplication less than the former.

Once we have the standardized estimate $\hat{\boldsymbol{\Sigma}}_0 = \text{std}_{\boldsymbol{\Sigma}}(\hat{\boldsymbol{\Sigma}})$ we want to describe how much it deviates from its target value, the identity matrix, by a single scalar value. There are many ways to that. For example: (a) $\log\det\left(\hat{\boldsymbol{\Sigma}}_0\right)$, the logarithm of the generalized variance, which is equal to zero when $\hat{\boldsymbol{\Sigma}}_0 = \boldsymbol{I}$ and the sign indicates if the variance is under- or over-estimated; (b) $\log(\text{cond}(\hat{\boldsymbol{\Sigma}}_0))$, the logarithm of the condition number,

which ignores the scale and focuses on the shape of the covariance matrix; (c) $\|\hat{\boldsymbol{\Sigma}}_0 - \boldsymbol{I}\|_2$, Frobenius norm of the difference between matrices; and there probably exist many others. For this paper we choose just one measure and use it repeatedly. The measure we employ has a reasonable statistical interpretation and therefore seems to be a decent candidate. It is defined as

$$d_{\text{LRT}}(\hat{\boldsymbol{\Sigma}}_0) = \operatorname{tr} \hat{\boldsymbol{\Sigma}}_0 - \log\det\left(\hat{\boldsymbol{\Sigma}}_0\right) - p. \tag{2.6}$$

In order to understand it, consider testing $H_0 : \boldsymbol{\Sigma} = \boldsymbol{I}$ vs $H_A : \boldsymbol{\Sigma} \neq \boldsymbol{I}$ based on a sample of size n from $\mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$ such that its sample covariance matrix is equal to $\hat{\boldsymbol{\Sigma}}_0$. Then it is easy to show (Seber, 2004, p.93) that the Likelihood Ratio Test (LRT) statistic is equal to $n \times d_{\text{LRT}}(\hat{\boldsymbol{\Sigma}}_0)$. Obviously, $d_{\text{LRT}}(\hat{\boldsymbol{\Sigma}}_0) = 0$ when $\hat{\boldsymbol{\Sigma}}_0 = \boldsymbol{I}$ and since the LRT statistic achieves its minimum value when $\hat{\boldsymbol{\Sigma}}_0 = \boldsymbol{I}$, we can conclude that $d_{\text{LRT}}(\hat{\boldsymbol{\Sigma}}_0) \geq 0$ for any $\hat{\boldsymbol{\Sigma}}_0 \in \text{SPD}(p)$. Additionally, we can see that $d_{\text{LRT}}(\hat{\boldsymbol{\Sigma}}_0)$ can be expressed as a function of $\{\lambda_i\}$, the eigenvalues of $\hat{\boldsymbol{\Sigma}}_0$, as

$$d_{\text{LRT}}(\hat{\boldsymbol{\Sigma}}_0) = \sum_{i=1}^{p} (\lambda_i - \log \lambda_i) - p, \tag{2.7}$$

justifying the simplified standardization in (2.5).

We will usually consider the average of $d_{\text{LRT}}(\hat{\boldsymbol{\Sigma}}_0)$ over a number of replications as the reciprocal of the measure of performance. Since the quantity is always positive, it is an alternative to Mean Squared Error (MSE) rather than bias. Like any MSE-type measure, it is also non-descriptive: it tells us how different the matrices are but does not explain in which way.

2.1.5 Existing robust methods

When the proportion of contamination is not large, the most popular is case-wise down-weighting/deletion, that is removal of every data line that has at least one value suspected to be an outlier. It works fine when there is enough completely clean cases but is obviously inefficient if one believes that the contamination model is truly independent.

In a recent paper Van Aelst et al. (2009) propose a method to assign weights to individual cells in a dataset. They consider all cases that receive small weights from the Stahel–Donoho projection pursuit procedure (Donoho (1982) and Stahel (1981)) and attempt to restore weights for those cells that "do not contribute" much to the Stahel–Donoho outlierness. They propose that the location and scatter estimates $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}_{\text{w}}$ can then be computed as

$$\hat{\mu}_a = \frac{\sum_{i=1}^{n} w_{ia} x_{ia}}{\sum_{i=1}^{n} w_{ia}}, \text{ for } a = 1, \ldots, p, \text{ and}$$
$$(\hat{\boldsymbol{\Sigma}}_{\text{w}})_{ab} = \frac{\sum_{i=1}^{n} \sqrt{w_{ia}}\sqrt{w_{ib}}(x_{ia} - \hat{\mu}_a)(x_{ib} - \hat{\mu}_b)}{\sum_{i=1}^{n} \sqrt{w_{ia}}\sqrt{w_{ib}}}, \text{ for } a, b = 1, \ldots, p,$$

where w_{ij} is the weight for the jth variable in the ith case. The paper focuses on the weight assignment procedure and the performance of the location estimate $\hat{\boldsymbol{\mu}}$ but pays little attention to the scatter estimate $\hat{\boldsymbol{\Sigma}}_{\mathrm{w}}$. In our work the scatter matrix estimate is of primary interest.

2.1.5.1 Winsorising and Huberising

Winsorising that we already mentioned above is arguably the next easiest way to proceed. It acknowledges the nature of the independent contamination and deals with data on a cell-by-cell — as opposed to case-by-case — basis. This simple method is not without merit and the biggest conceptual problem with it is that it effectively makes up data that are not observed. One immediate implication is that, even when there is no contamination at all, the estimate will have *intrinsic bias* — an asymptotic bias caused by such systematic but arbitrary (from the point of view of covariance matrix) modifications. The smaller the value of the Winsorising constant is, the larger the intrinsic bias it will produce. Large Winsorising constants, on the other hand, will offer less protection against the influence of outliers.

Huberised correlation, which is defined as a regular Pearson correlation after applying $x_{\mathrm{hub}} = \psi((x - \hat{m}_j)/\hat{s}_j)$ transformation, is a generalization of Winsorised estimates. The function $\psi(\cdot)$ can be any odd, bounded and non-decreasing. Winsorising is a special case when $\psi(x) = \min(x, k)$ for $x > 0$, with k being the Winsorising constant. Huberising gives more flexibility but does not address the fundamental problem of the intrinsic bias. Huberised estimates of scatter matrices and some approaches to eliminate intrinsic bias have been studied by Alqallaf (2003).

A step further in this direction is the approach suggested by Little and Smith (1987) that completely removes suspicious data cells from the analysis by marking them as missing values and computing the Maximum Likelihood (ML) covariance estimate on the incomplete data. It does not eliminate intrinsic bias either. We will study these estimates in a more systematic fashion in Section 2.2.

2.1.5.2 Pairwise estimates

To approach the problem of independent contamination from a completely different angle it has been suggested that pairwise estimates can be of use (e.g. Rousseeuw and Molenberghs (1993) discussed this in other contexts). Recall that the main reason why standard (affine-equivariant) robust estimates cannot be used is that the proportion of completely clean cases $(1 - \varepsilon)^p$ may drop below 50% when the dimension p is large. Therefore it seems reasonable that we can compute each element of the scatter matrix by considering only two variables at a time. If the bivariate covariance estimate is robust with breakdown point of 50% then we can allow ε as large as $1 - \sqrt{0.5} = 0.29$ and still have a reasonable estimate of the scatter matrix. After all, the usual sample covariance matrix is just a

collection of bivariate covariances. If each bivariate estimate is unbiased or consistent then the resulting matrix estimate is naturally unbiased or consistent by construction as well.

Difficulties arise when such estimates are computed on *finite samples*. The immediate problem that one runs into when attempting to implement this idea is that, unlike its non-robust counterpart, the resulting matrix is not guaranteed to be semi positive definite (SPD), which is a necessary property if it is intended to be used in any kind of spectral analysis. Methods have been proposed to deal with this problem and to force positive definiteness onto the estimated scatter matrix (see the previous reference or Maronna and Zamar (2002), for example). What this correction fails to address, however, is that the finite sample pairwise estimate, even if positive definite, does not capture the multivariate spectral structure of the scatter matrix well. This, of course, is more pronounced when there is some non-trivial structure in the true scatter to begin with. Matrices with weak correlations, especially in low dimensions, can be estimated reasonably well by pairwise estimates.

Looking back at the Winsorised/Huberised estimates one may think that the intrinsic bias can be corrected by adjusting the estimated pairwise covariances using the knowledge of the ψ-function and assuming multivariate normal distribution of the data. Such corrections, indeed, can be made, producing an estimate which is unbiased under no-contamination model but they will disturb the spectral structure and may even result in non-SPD results. Such a method would be pairwise in nature and share all the deficiencies that we have described above.

See Alqallaf (2003) for more elaborate comparison of Huberised correlation estimates with pairwise quadrant correlations. We will largely ignore pairwise estimates and focus on those that attempt to capture the fine structure of the data.

2.2 General approach to independent contamination

2.2.1 Basic idea

In this paper we will focus on methods of dealing with independent contamination that can be described as a three-stage process: (a) detect contamination; (b) process it in some way; (c) apply a standard (e.g. maximum likelihood) estimate of location-scatter to the processed data. The last step is largely determined by the first two which vary depending on the specific method being used. In this section we will discuss what options are available and elaborate on some of them. Section 2.3 will describe outlier detection in more detail.

2.2.2 Detection of contamination

Identifying which data cells might not be coming from the distribution of interest is the first and arguably the most important step of analyzing contaminated data. Note that we will be focusing on data *cells* rather than data *cases* which were the primary unit of interest for affine equivariant robust estimates. We can crudely classify all detection techniques into *univariate* and *multivariate*. The former will treat each variable independently from all others and will only use information within this one variable to decide which cases are likely to be contaminated. The latter will try to use assumed relationships in the clean data to flag data values that do not fit them very well. We will discuss both strategies later in this subsection.

What is common between all detection strategies is that they try to maximize *recall*, i.e. the proportion of contaminated values that are identified as such, and *precision*, i.e. the proportion of tagged values that are indeed contaminated, and to find the optimal balance between them. We want to detect as much real contamination as possible while minimizing the number of false positives, i.e. clean data cells mistakenly labelled as contaminated. The exact price of mistakes of both kinds depends on the subsequent processing but whatever is being done there are some commonalities worth mentioning. Letting contamination go undetected will cause larger *contamination induced* bias. If the detection method allows large contamination to slip through then the damage can be quite dramatic. Filtering out too much of the good data has three negative effects: (a) when data are clean the resulting estimate may acquire *intrinsic bias*; (b) reduced amount of good data may lead to the loss of efficiency of the resulting estimate; (c) less good data to counterweight contamination may cause higher contamination-induced bias. Most detection methods have a cutoff constant that allows fine tuning of this balance.

2.2.2.1 Univariate detection

This is the simplest mechanism widely used in practice even when users are unaware that they are dealing with independent contamination. It proceeds by computing robust estimates of location $\hat{m}_j$ and scatter $\hat{s}_j$ for each variable X_j, computing z-scores $\left|\frac{x_{ij}-\hat{m}_j}{\hat{s}_j}\right|$ for each data value x_{ij} and comparing it to a fixed threshold. Any value above the cutoff is flagged as potential outlier. When the clean data are assumed to follow a normal distribution the cutoffs are usually chosen somewhere between 1.5 and 3.5. Anything larger than 3.5 guarantees that the number of false positives is very close to zero and therefore there is little need to chose it much larger than 3.5. Going below the value of 1.5, on the other hand, would filter out too much good data (13%) and is typically deemed as too much sacrifice. Outliers normally cause $\hat{s}_j$ to overestimate the true standard deviation and the detection won't be as harsh as it may seem assuming that the scale is known.

Instead of comparing z-values to a fixed cutoff it is also possible to do *trimming*, that is to label the $[\alpha n]$ largest as well as $[\alpha n]$ smallest values for each variable as outliers. Other

ad-hoc approaches to univariate contamination detection can be imagined but we do not attempt to create a comprehensive list here. We also do not compare these approaches to each other and will always use the z-score method with an appropriate cutoff when referring to univariate detection.

These methods work great for detecting truly gross contamination and even by themselves are capable of preventing an estimate from premature breakdown. The problem is that they allow moderate contamination (with z-values around and under the cutoff) go unnoticed, which, depending on the true covariance structure, can still cause serious bias.

2.2.2.2 Multivariate detection

When the true covariance matrix is not diagonal and the variables are correlated then each data value in the data set carries some information about the values of other variables in the same case. Likewise, knowing $(p-1)$ components of a random p-vector may give us a good idea what the remaining value should be, provided that we know the true scatter matrix and the distribution family which the data come from. One can imagine a variety of methods that exploit this relationship to detect contaminated values. One such method was proposed by Little and Smith (1987) and compares the Mahalanobis distance (to the estimated center) of a full case to the Mahalanobis distance of a leave-one-out $(p-1)$-tuple. Generally speaking this is a step forward from the univariate approach as it uses more information available in the data. The problem, however, is that the true scatter matrix is usually not known because it is the very thing that we are trying to estimate.

There are a number of open questions with this approach and we have some ideas and modifications for it that we will discuss in Section 2.3 together with some numerical simulation results comparing the performance of various detection methods.

2.2.3 Processing of contamination

2.2.3.1 Overview of various methods

Once the contaminated data cells have been detected, an action needs to be taken to eliminate or, at least, reduce their negative influence on the estimate. We will describe three general approaches in this section.

Winsorising is the simplest procedure. When a data value is too large to be plausible under a given model it can simply be pulled closer to the assumed center until the value is believable. If a cutoff for z-score has been used then Winsorised values are taken according to the following rule:

$$x_{\text{wins}} = \begin{cases} x & \text{if } |x - \hat{m}_j| \le c_{\text{wins}}\hat{s}_j \\ \hat{m}_j + c_{\text{wins}}\hat{s}_j & \text{if } x > \hat{m}_j + c_{\text{wins}}\hat{s}_j \\ \hat{m}_j - c_{\text{wins}}\hat{s}_j & \text{if } x < \hat{m}_j - c_{\text{wins}}\hat{s}_j, \end{cases}$$

where c_{wins} is the cutoff from the univariate detection process. Winsorising can also be used with multivariate detection procedures but we will discuss it in Section 2.3.

This contamination processing method is very simple and does not require any specialized maximum likelihood procedure to compute the estimate afterwards as the sample covariance can be computed on the Winsorised data set. It is relatively good at preserving the information in clean data that was mistakenly classified as contamination. Even after Winsorising, the information that a relatively large value on the given side of the center has been observed is still available in the modified data set. The price for this is the fact that outliers will still have their direct say in the final estimate. Their influence will be reduced to an acceptable level but not completely eliminated. A side effect is that most univariate outliers become equally dangerous as all of them get pulled down to the same value.

As we mentioned in the introduction, despite being relatively harmless to good data, Winsorising still has intrinsic bias when some of the good data are labelled as contamination. Combined with incomplete elimination of outliers' influence it makes this approach suboptimal if performance is valued higher than simplicity.

If the whole estimation problem was univariate then Winsorising could be seen as assigning a weight to large values that is inversely proportional to the value itself. But when the observed value is very large and it is certain that it does not come from the distribution of interest, we would often like to completely eliminate such a value from the analysis. In traditional robust estimates it is done by assigning a zero (or at least $o(x)$) weight to such observations. At the moment we have not yet figured out a way to assign an arbitrary weight to a data cell[1] but a zero weight can be achieved by removing the offending value from the data set and marking its spot as *missing at random (MAR)*. This has been proposed by Little and Smith (1987) in the context of survey questionnaires where they believed that unusual responses to one question can be removed without discarding all of the respondent's data.

After missing values have been introduced the covariance matrix can be estimated using an appropriate EM-algorithm, which is particularly efficient and straightforward to implement if the data are assumed to be multivariate normal.

The major advantage of this approach is that gross contamination gets completely wiped out. If the contaminating mechanism is independent of the clean data then the removed truly contaminated values are genuinely missing at random. With regards to good data, however, it still causes some intrinsic bias. If a genuine good data value gets classified as an outlier and subsequently removed from the data set then it is obviously not missing at random. The fact that it is missing is directly related to the fact that the observed value was far from where it was expected to be. One may think that completely removing a data value is less harmful than changing it to an arbitrary smaller number but

[1] Of course, we could *assign* an arbitrary weight to any data cell but we do not have an appropriate estimation procedure to *make use* of such an assignment.

this is not necessarily the case. Winsorising preserves the information that a large value has been observed while marking it as MAR completely eliminates it, which contributes to the bias.

To attempt to combine the best of both worlds we propose a method that will not create arbitrary data values but at the same time will preserve some information about large values being observed. Instead of completely removing a suspected outlying value we can artificially *censor* it. We will record that a large value has been observed — together with the censoring cutoff and on which side of the center the value appeared — but will discard the actual value. This censoring information can be incorporated into the maximum likelihood analysis to be performed on the cleaned data set. The major advantage is that this method will not have intrinsic bias when data are uncontaminated and follow the distribution assumed in the MLE because the possibility of false positives is already incorporated into the analysis. This allows the user to take smaller cutoff values without the risk of introducing too much intrinsic bias. On the negative side we can mention that (a) it is harder to implement the censored ML procedure; (b) the true contamination will still have its influence and affect the estimate in a similar way to what it does to Winsorised estimates. We will continue the discussion of this method and its implementation in section 2.2.3.2.

2.2.3.2 Implementation of the censored estimate

The EM-algorithm popularized by Dempster et al. (1977) can be used to obtain the MLE in this situation. The usual multivariate Gaussian EM-algorithm for MAR data is easy to implement and is well described in the literature (see, for example, Little and Rubin (2002)). The modifications to accommodate censored data are relatively straightforward and we describe them below along with some computational challenges encountered while implementing them.

The standard EM-algorithm for the multivariate Gaussian data is an iterative procedure that starts with some $(\boldsymbol{m}^{(0)}, \boldsymbol{\Sigma}^{(0)})$ and then, at the iteration step $t = 0, 1, 2, \ldots$, updates the parameters $(\boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)})$ according to the following rules. First, for each partly observed case $\boldsymbol{x}_i$, $i = 1, \ldots, n$ we define a random variable $\boldsymbol{X}_i^{\text{cond}}$ by conditioning the multivariate $\mathcal{N}(\boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)})$ on the observed part of $\boldsymbol{x}_i$ such that $\boldsymbol{X}_{[i]}^{\text{cond}} = \boldsymbol{x}_{[i]}$. Then the estimates are updated as follows:

$$\boldsymbol{m}^{(t+1)} = \sum_{i=1}^{n} \mathbb{E}\boldsymbol{X}_i^{\text{cond}}, \text{ and } \boldsymbol{\Sigma}^{(t+1)} = \sum_{i=1}^{n} \left\{ (\mathbb{E}\boldsymbol{X}_i^{\text{cond}})(\mathbb{E}\boldsymbol{X}_i^{\text{cond}})' + \mathbf{Cov}(\boldsymbol{X}_i^{\text{cond}}) \right\}. \quad (2.8)$$

The distribution for the components of $\boldsymbol{X}_i^{\text{cond}}$ corresponding to the missing parts of $\boldsymbol{x}_i$ is

well known and is multivariate normal itself with parameters given by

$$(\mathbb{E}\boldsymbol{X}_i^{\text{cond}})_{[-i]} = \boldsymbol{m}_{[-i]}^{(t)} + \boldsymbol{\Sigma}_{[-i,i]}^{(t)} \left(\boldsymbol{\Sigma}_{[i,i]}^{(t)}\right)^{-1} (\boldsymbol{x}_{[i]} - \boldsymbol{m}_{[i]}^{(t)}) \tag{2.9}$$

$$(\mathbf{Cov}(\boldsymbol{X}_i^{\text{cond}}))_{[-i,-i]} = \boldsymbol{\Sigma}_{[-i,-i]}^{(t)} - \boldsymbol{\Sigma}_{[-i,i]}^{(t)} \left(\boldsymbol{\Sigma}_{[i]}^{(t)}\right)^{-1} \boldsymbol{\Sigma}_{[i,-i]}^{(t)}. \tag{2.10}$$

We use subscript $[i]$ to denote the sub-vector corresponding to the components observed in the ith case. Similarly $[-i]$ corresponds to all missing components of the ith case. Submatrices are also indexed in this fashion. The rows and columns of $\mathbf{Cov}(\boldsymbol{X}_i^{\text{cond}})$ corresponding to the observed values in $\boldsymbol{x}_i$ are all equal to zero because of the conditioning.

With censored data there are three types of values in the dataset: (a) observed; (b) left censored; (c) right censored. It is also possible to have completely missing cases (assumed to be MAR) but they can be considered as a special case of left-censoring at negative infinity. For each case $\boldsymbol{x}_i$, we will denote its observed values as $\boldsymbol{x}_{[i]}$, its left censored values as $\boldsymbol{x}_{[-l(i)]}$, and right censored as $\boldsymbol{x}_{[-r(i)]}$. In the most general situation each case can have its own set of censoring values. Let us assume that for each $\boldsymbol{x}_i$ we have a corresponding vector $\boldsymbol{\gamma}_i$ of such values. Using censoring information means conditioning our likelihood inference on the fact that $\boldsymbol{x}_{[-l(i)]} > \boldsymbol{\gamma}_{[-l(i)]}$ and $\boldsymbol{x}_{[-r(i)]} < \boldsymbol{\gamma}_{[-r(i)]}$, where vector inequalities are to be understood as "all components must satisfy". Components of $\boldsymbol{\gamma}_i$ corresponding to $\boldsymbol{x}_{[i]}$ are ignored.

With censored data, the updating expressions in (2.8) change to

$$\boldsymbol{m}^{(t+1)} = \sum_{i=1}^{n} \hat{\boldsymbol{x}}_i, \text{ and } \boldsymbol{\Sigma}^{(t+1)} = \sum_{i=1}^{n} \hat{\boldsymbol{x}}_i \hat{\boldsymbol{x}}_i' + \hat{\boldsymbol{C}}_i, \tag{2.11}$$

where

$$\hat{\boldsymbol{x}}_i = \mathbb{E}\Big[\boldsymbol{X} \sim \mathcal{N}_p(\boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)}) \,\Big|\, \boldsymbol{X}_{[i]} = \boldsymbol{x}_{[i]}, \boldsymbol{X}_{[-l(i)]} > \boldsymbol{\gamma}_{[-l(i)]}, \boldsymbol{X}_{[-r(i)]} < \boldsymbol{\gamma}_{[-r(i)]}\Big], \text{ and} \tag{2.12}$$

$$\hat{\boldsymbol{C}}_i = \mathbf{Cov}(\boldsymbol{X} | \text{the same conditions as above}). \tag{2.13}$$

Quantities (2.12) and (2.13) are conceptually simple but unfortunately there are no closed form expressions similar to (2.9) and (2.10) available in this case. We will evaluate them using Monte Carlo methods by taking samples from the conditional distribution. To sample from the conditional distribution in (2.12) and (2.13) we will first condition $\mathcal{N}(\boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)})$ on fully observed values $\boldsymbol{x}_{[i]}$ using expressions (2.9) and (2.10) and then condition again on the censoring information using Monte Carlo methods. So in the end the problem reduces to sampling from a non-central octant (defined by $\boldsymbol{\gamma}_{[-l(i)]}$ and $\boldsymbol{\gamma}_{[-r(i)]}$) of another multivariate normal (which is a conditional itself). The whole estimation algorithm is outlined below.

Overview of the algorithm:
(variables in square brackets on the left indicate the depth of nested looping)

- $[\emptyset]$ The outermost is the EM-loop over t: on each step we go from $(\boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)})$ to $(\boldsymbol{m}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)})$ by using expression (2.11). Iterate until some convergence criterion is reached.

- $[t]$ For each case i, we need to compute its $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{C}}_i$ which are the first and second moments of a conditional distribution specified in (2.12). The rest of the algorithm explains how to do this.

- $[t, i]$ Use Monte Carlo to compute the moments. Need to sample from the conditional distribution $\mathcal{N}_p(\boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)})$ given $\boldsymbol{X}_{[i]} = \boldsymbol{x}_{[i]}, \boldsymbol{X}_{[-l(i)]} > \boldsymbol{\gamma}_{[-l(i)]}, \boldsymbol{X}_{[-r(i)]} < \boldsymbol{\gamma}_{[-r(i)]}$. The rest of the algorithm explains how to sample from it.

- $[t, i]$ Condition $\mathcal{N}_p(\boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)})$ on fully observed values $\boldsymbol{x}_{[i]}$ first: the distribution is the multivariate normal with parameters given by (2.9) and (2.10). Let us call these parameters $\boldsymbol{m}_c$ and $\boldsymbol{\Sigma}_c$, where "c" stands for conditional and subscripts t and i are omitted for brevity. Dimension of this conditional distribution is q_i, the number of censored values in the ith case.

- $[t, i]$ Sample from $\boldsymbol{Y} \sim \mathcal{N}_{q_i}(\boldsymbol{m}_c, \boldsymbol{\Sigma}_c)$ *conditional* on the censoring information which we can re-index as $Y_j > \gamma_j$ for $j = 1, \ldots, q_i$. Without loss of generality we can assume that all censoring is done on the left — otherwise just multiply both Y_j and γ_j by negative one. The rest of the algorithm explains how to sample from the censored distribution.

- $[t, i]$ Use Gibbs sampler (e.g. Casella and George (1992)) to sample one variable at a time, conditioning on the rest of the case. Iterate through variables $j = 1, \ldots, q_i$ in a loop for $N_B + N_G$ full circles to obtain N_G samples and discard the first N_B as burn-in. The rest of the algorithm gives details of the Gibbs sampling.

- $[t, i]$ Initiate $\boldsymbol{y}$ with $y_j = \max(\gamma_j, (m_c)_j)$, for $j = 1, \ldots, q_i$.

- $[t, i, j]$ Conditional distribution of the jth variable given $\boldsymbol{Y}_{-j} = \boldsymbol{y}_{-j}$ is a truncated ($Y_j > \gamma_j$) univariate normal with parameters $\tilde{m}_j$ and $\tilde{\sigma}_j^2$ which can be computed using the same formulae as in (2.9) and (2.10):

$$\tilde{m}_j = (m_c)_j + (\boldsymbol{y}_{-j} - (\boldsymbol{m}_c)_{-j})((\boldsymbol{\Sigma}_c)_{-j,-j})^{-1}(\boldsymbol{\Sigma}_c)_{-j,j}$$
$$\tilde{\sigma}_j^2 = (\boldsymbol{\Sigma}_c)_{j,j} - (\boldsymbol{\Sigma}_c)_{j,-j}((\boldsymbol{\Sigma}_c)_{-j,-j})^{-1}(\boldsymbol{\Sigma}_c)_{-j,j}.$$

- $[t, i, j]$ To get one sample from the truncated multivariate normal, take

$$y_j = \tilde{m}_j + \tilde{\sigma}_j^2 \Phi^{-1}(U), \text{ where } U \sim \text{Uniform}\left(\Phi\left(\frac{\gamma_j - \tilde{m}_j}{\tilde{\sigma}_j}\right), 1\right). \tag{2.14}$$

This idea was discussed in Robert (1995). The paper also offers improved methods of sampling from truncated distributions but we decided to use this simple method for our estimate.

The number of samples N_G that we obtain using the Gibbs sampler to compute expectations $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{C}}_i$ is a performance tuning parameter. In the ideal world without time limitations, we would take $N_G = \infty$ and that would give us exact values for the integrals. In practice, however, we use the Gibbs sampling as an intermediate step of the EM-algorithm, and it may not be worth our time to compute the integrals precisely when $(\boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)})$ are still far from their limiting value. Therefore we propose to increase N_G as EM-iterations progress. In our implementation we started with $N_G = 100$ and doubled it every time the following condition hold true:

$$\delta(\boldsymbol{\Sigma}^{(t+1)}, \boldsymbol{\Sigma}^{(t)}) > \delta(\boldsymbol{\Sigma}^{(t)}, \boldsymbol{\Sigma}^{(t-1)}), \text{ with } \delta(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \frac{\sum_{a=1}^{p}\sum_{b=1}^{p} |\boldsymbol{\Sigma}_{1ab} - \boldsymbol{\Sigma}_{2ab}|}{\operatorname{tr} \boldsymbol{\Sigma}_1}. \quad (2.15)$$

This is based on the understanding that if $\{\boldsymbol{\Sigma}^{(t)}\}$ is on its way to convergence then the consecutive improvements will be smaller and smaller. But once the error in $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{C}}_i$ due to insufficient N_G is comparable to the variation from one step of EM to another, the pattern will not have to hold anymore and the condition (2.15) be satisfied eventually.

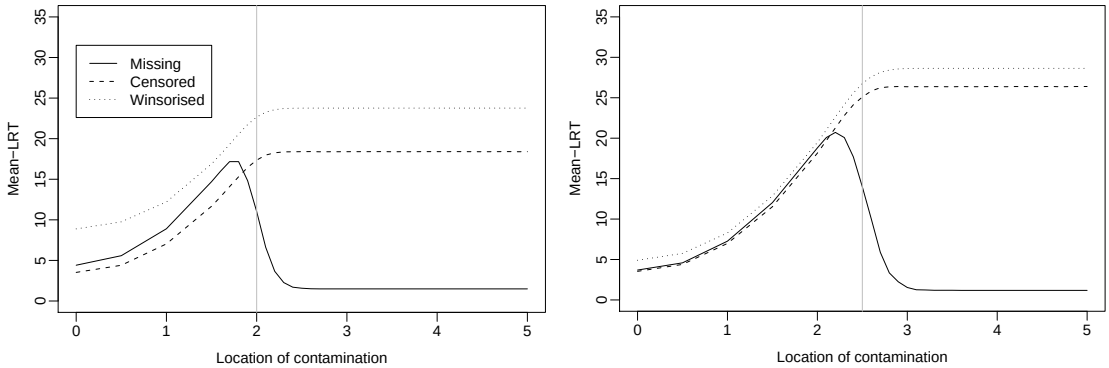
2.2.3.3 Numerical comparison of the three methods

To shed some light on the relative performance of the methods described above (MAR, Winsorising and censoring) we conduct a simple simulation study.

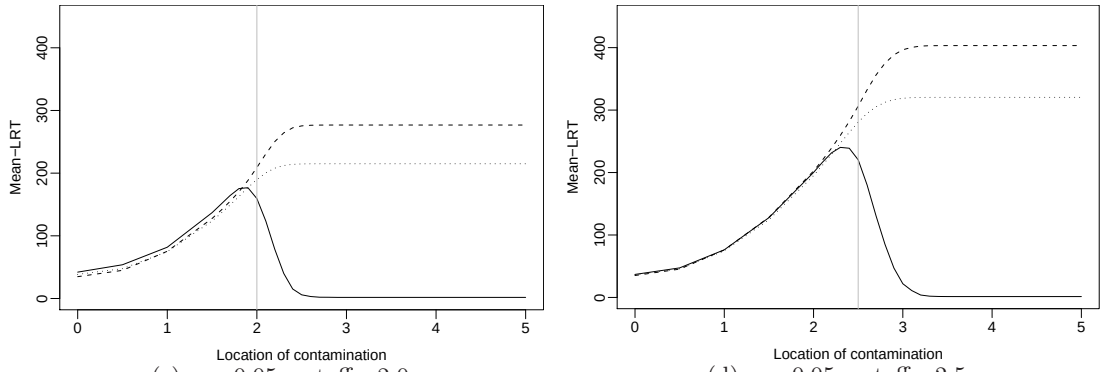
Clean data are generated from the 20-variate normal distribution with *high* correlations (see section 2.1.4.3); sample size $n = 200$. Once clean data are generated we inject a certain fraction ε of univariate point mass contamination in each of the 20 variables independently. The contamination is located at value k that varies from 0 to 5; once $k > 5$ (and probably even earlier) the contamination will be detected with 100% certainty and thus the performance of the methods will not depend on k anymore. We consider two different scenarios: (a) $\varepsilon = 0.005$, which is a very small proportion of contamination; (b) and $\varepsilon = 0.05$ which represent a significant amount of contamination. Remember that, on average, proportion $1 - (1 - \varepsilon)^p \approx p \times \varepsilon$ of all *cases* will be at least partially affected by such contamination: that is 9.5% and 64% respectively.

Univariate detection is then applied to each dataset. We used two different cutoffs: 2.0 and 2.5 to demonstrate what effect this choice has on the final estimates. After that, the three estimates, are computed on the datasets: (a) one will mark all suspect values as *MAR* and then compute the MLE; (b) *Winsorising* will bring larger (in absolute value) values down to the cutoff and compute the sample covariance matrix of modified data; (c) and *censoring* estimate will also bring larger values down to the cutoff but treat them as censored while computing the MLE (see section 2.2.3.2 for the details of implementation).

Once the estimate $\hat{\boldsymbol{\Sigma}}$ is computed we standardize it using the true known covariance matrix: $\text{std}_{\boldsymbol{\Sigma}_h}(\hat{\boldsymbol{\Sigma}}) = \boldsymbol{\Sigma}_h^{-1}\hat{\boldsymbol{\Sigma}}$, and assess its deviation from the identity matrix $\boldsymbol{I}$ by using the LRT-type statistic $d_{\text{LRT}}(\text{std}_{\boldsymbol{\Sigma}_h}(\hat{\boldsymbol{\Sigma}}))$, described in section 2.1.4.3. We focus on the performance of the covariance estimates.



(a) $\varepsilon = 0.005$, cutoff= 2.0 (b) $\varepsilon = 0.005$, cutoff= 2.5



(c) $\varepsilon = 0.05$, cutoff= 2.0 (d) $\varepsilon = 0.05$, cutoff= 2.5

Figure 2.1. Comparison of the three ways of processing contaminated values. See entire section 2.2.3.3 for discussion.

The whole process is replicated 500 times and the average of the $d_{\text{LRT}}(\text{std}_{\boldsymbol{\Sigma}_h}(\hat{\boldsymbol{\Sigma}}))$ scores is computed for all 2(values of ε) $\times$ 2(cutoffs) $\times$ 3(estimates) $\times$ (various values of k) scenarios. To simplify our language we will call this MSE-LRT-type quantity *bias* throughout this section but, in fact, it is closer to the MSE as it combines actual bias with the finite sample variability. The results, as functions of k, are shown on the four panels in Figure 2.1.

The general trend in all four scenarios is what one would expect: the bias increases as contamination moves away from the center until it is detectable by the given cutoff. Once contamination is detected, it does not matter anymore how gross it was. The censored and Winsorised estimates will remain constant and the bias of the MAR estimate will drop down to almost zero. This makes a strong case for using MAR approach when contamination is expected to be large.

Comparing panels 2.1(a) and 2.1(b), which differ in the value of the cutoff used for

detection, we can see two distinct phenomena happening. First, with larger cutoff, the worst possible bias, achieved near the cutoff value, is higher. This is no surprise because we allow larger contamination to go unnoticed and affect the estimate. The same can be said when $\varepsilon = 0.05$ in figures 2.1(c) vs 2.1(d). Second, with smaller cutoff, the discrepancy between Winsorised and censored estimates is more pronounced. As the fraction of contamination is very small here (only 0.5%), this is almost the ideal scenario for which censored estimate is designed. It makes the best use of the clean data and would be completely unbiased if there was no contamination at all. The other two estimates modify clean data which results in some distortion of estimates. With larger cutoff, such as 2.5, only 1.2% of clean data are affected and is not of much importance. When cutoff is equal to 2.0, a whole 5% of clean data are being modified by the MAR and Winsorised estimates and we can see that the censored estimate has the lowest bias.

Another interesting comparison is across different levels of contamination, or rows in Figure 2.1. Obviously, the overall level of bias is much higher when more contamination is present. Other than that, we can also see that for gross, and hence fully detected, contamination, censored estimate appears to be better than Winsorised when $\varepsilon = 0.005$, but the reverse holds when ε goes up to 0.05. As mentioned before, the former is due to the superior handling of clean data by the censored estimate. For larger ε, however, contamination plays the more important role and overrules the small improvement achieved by handling clean data with care. The censored estimate appears to have larger worst bias than Winsorised because of the way the detected contamination is handled. Simplifying things a little bit, we can say that censored ML estimate treats censored values as if they were equal to $\mathbb{E}_{\mathcal{N}}[X|X > \text{cutoff}]$ which is always larger than the cutoff value itself. This produces larger bias than that of Winsorised estimate. This is not to be considered a drawback of censored estimates as their cutoff can be taken smaller without inducing too much intrinsic bias due to modifying clean data.

To summarize our understanding of the three estimates we do a quick bullet-point summary of pairwise differences between them. We start with the comparison of *Censored* vs *Winsorised* estimate:

- The two estimates are very *similar* from the point of view of performance.
- Censoring offers *better* treatment of clean data. It is more conceptually sound because no data are being modified. But the numerical difference when estimating covariance matrices is fairly small and may be lost or obscured if non-negligible fraction of contamination is present.
- Censoring also treats outliers somewhat *better*. Censoring does not fix suspected data at a particular value but rather specifies an interval where it is allowed to be placed in an attempt to maximize the likelihood function. This slightly reduces the influence the contaminated value has on the estimate. It must be noted that both

censoring and Winsorising treat truly contaminated data in a suboptimal way. The difference between this bad and that bad is very subtle and can hardly be observed in simulations or actual data analysis.

- Censoring is *slower* to compute and conceptually more complicated. This seems to be the only drawback of the method compared to Winsorisation and if one were not under any time performance constraints then censoring should always be preferred. Note that cutoffs should be lowered accordingly to avoid extra bias as shown in Figures 2.1(c) or 2.1(d).

Comparing *Censoring* vs *Marking-as-MAR* approaches we can say the following about censoring:

- Censoring provides *better* treatment of clean data. Systematic elimination of large data values as done by MAR-treatment causes intrinsic bias. This improvement is noticeable when very little contamination is present but otherwise is of little importance.

- Censoring is much *worse* with respect to how it deals with true contamination. Outliers still do have an impact on the estimate instead of being completely removed from the analysis. This can become very important if many gross (i.e. larger than the cutoff) outliers are observed.

- Implementation of the censored MLE is considerably *slower* than the MLE assuming MAR. The EM-algorithm implementation of the latter is very simple and computationally efficient. Adding censored observations into the process makes it slower because closed form E-step is not possible anymore (see section 2.2.3.2).

- It is *harder* to combine censoring with multivariate detection. Although it appears to be possible to use censored estimates based on results of multivariate detection, we have not implemented it in this thesis. Because of this limitation we have not been able to compute the combined estimate that seems to be the natural conclusion from all the above: high-cutoff multivariate detection to mark definite outliers and low-cutoff multivariate detection to censor values that are only suspicious, followed by an MLE using all the detection results.

And finally *Winsorising* vs *Marking-as-MAR*:

- Everything that was just said above about censoring and Marking-as-MAR applies to the Winsorising except the comment about computational efficiency. MAR-approach, even with Gaussian EM-algorithm, is computationally more complex than the sample covariance used in Winsorising.

Based on the results of this section we have chosen MAR-approach as our preferred estimate and will consequently use it in conjunction with various multivariate detection techniques described in section 2.3.

2.3 Multivariate detection of independent contamination

2.3.1 Basic principle: sharing information between variables

We may say that the univariate detection method judges individual data values based on how well they fit with the rest of the data from the same *column*. If a value falls too far from its expected value as measured by its estimated standard deviation then we have reasonable ground to believe that it does not come from the same distribution as the rest of this variable and therefore can be labelled as outlier.

When data are multivariate and have non-trivial dependence structure this thinking can be extended to utilize more information to judge individual data values. If variables are dependent then they carry some information about each other and therefore can be used to help and cross-judge each other. The plausibility of each data value can now be assessed based on how well it fits with the rest of the *case* (as well as the column).

There are two additional pieces of information needed to take advantage of this dependence. First, stronger distributional assumptions are required: at the very least the distribution has to be elliptical so that we can assume that outlierness is a monotone function of Mahalanobis distances. Additionally, in order to use standardized cutoffs to judge outlierness, we will assume that the distribution of the clean data is multivariate normal. Second, and most unfortunate requirement, is that we need to know the covariance matrix of the true distribution we are trying to estimate. When working with real data we will get a rough estimate of covariance matrix to do a rough detection and filtering first and then will try to improve it iteratively; see Section 2.3.7 for more details. To highlight the main idea, for the time being, we assume that the true center $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$ are already known and we only need to detect contaminated values.

Univariate detection uses the language of z-values that can be generalized to Mahalanobis distances in case of multivariate elliptical data. Mahalanobis distances to the center are a great and widely used tool for the detection of outlying *cases* of data. Under independent contamination model many cases will be partially contaminated and therefore the *full Mahalanobis distances*, computed using the whole p-dimensional data points, can only be used for initial screening to help us focus our attention on the cases that are likely to contain contaminated cells. Cases with reasonably small Mahalanobis distances will be assumed to be clean and not considered further in this section.

2.3.2 Partial Mahalanobis distances (P-approach)

Let $\boldsymbol{x}$ be a contaminated data case. It corresponds to a certain $\boldsymbol{x}_i = \boldsymbol{x}$ in the dataset but we omit the index i for clarity because we only consider one case at a time. Being contaminated means that the full squared Mahalanobis distance of $\boldsymbol{x}$ to the assumed center

of the data is larger than a pre-specified cutoff value:

$$\mathrm{MD}^2(\text{full}) = \mathrm{MD}^2(\boldsymbol{x}; \boldsymbol{m}, \boldsymbol{\Sigma}) > C_p = (\chi^2_p)^{-1}(1 - p_{\text{cutoff}}), \tag{2.16}$$

where p_{cutoff} is a small number controlling the rate of false positives.

Under the independent contamination model and reasonably small ε only a relatively small number $\varepsilon \times p$ out of p cells are expected to be contaminated. We will try to find out which ones. First, let us assume a working hypothesis that only one cell out of p is contaminated. If we can identify and remove this value then the rest of the case will be clean and coming from the assumed multivariate normal distribution projected on the corresponding $(p-1)$ dimensions. Therefore the squared *partial Mahalanobis distance* of the case with the contaminated cell removed will follow a χ^2_{p-1} distribution:

$$\mathrm{MD}^2_j = \mathrm{MD}^2(\boldsymbol{x}_{-j}; \boldsymbol{m}_{-j}, \boldsymbol{\Sigma}_{[-j]}) \sim \chi^2_{p-1}, \tag{2.17}$$

when x_j is the only contaminated cell in $\boldsymbol{x}$.

Going through all $j = 1, \ldots, p$ and looking for the one that brings MD^2_j down to the range of χ^2_{p-1} can be used to detect the contaminated value. The cutoff $C_{p-1} = (\chi^2_{p-1})^{-1}(1 - p_{\text{cutoff}})$ used to judge whether MD^2_j is trustworthy is a little bit smaller than the C_p that was used for complete cases. Three distinct outcomes are possible depending on the number of MD^2_j that pass the test (see Figure 2.2 for an illustration when $p = 2$):

1. $\#\{j \mid \mathrm{MD}^2_j \le C_{p-1}\} = 1$ and let us denote it j_*. This is the perfect result. It means that the presence of the j_*th variable makes the case outlying while the absence of it brings it back within the acceptable range. This is a clear indication that variable j_* (and only j_* as far as we know) is contaminated and should be removed. Detection process may stop here for this case once this one variable has been identified and acted upon.

2. $\#\{j \mid \mathrm{MD}^2_j \le C_{p-1}\} = s \ge 2$. This situation is conceptually confusing but arises in practice fairly often. Removal of any one of the s variables can make the overall outlying case look normal. Therefore all s variables are suspected to be contaminated but there is no evidence suggesting that more than one actually is. This is the end of the detection process for this case but the decision on which variable is the actual contaminator is still to be made. See the next subsection 2.3.3 for our approach to answering this question.

3. $\#\{j \mid \mathrm{MD}^2_j \le C_{p-1}\} = 0$. This means that the removal of any single variable is not enough to make this case clean. It is an indication that more than one variable is contaminated and we need to go deeper into the detection process. At this stage no decisions, or even guesses, about which variable is contaminated are made. At the next level, all pairs of variables will be considered and partial Mahalanobis

distances $\mathrm{MD}^2_{j_1,j_2}$ computed by removing one pair at a time. The whole process is repeated exactly the same except that pairs of variables play the role of single variables. Again three possible outcomes are possible and the whole process can go deeper and deeper until at least one of the partial Mahalanobis distances is sufficiently small (i.e. less than C_{p-k}). Computational complexity naturally increases as we go deeper because $\binom{p}{k}$ combinations of variables need to be considered at level k.

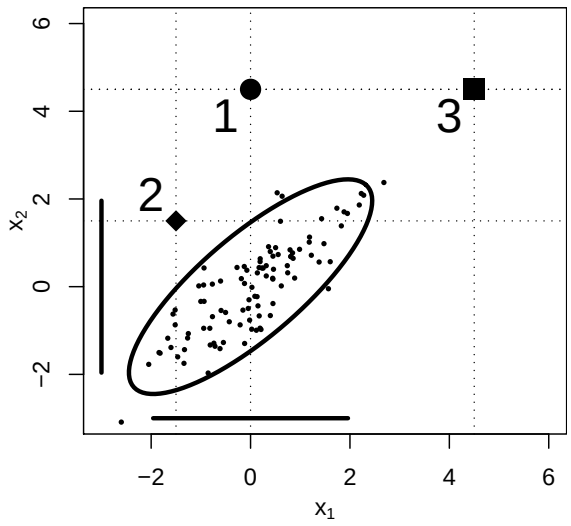


Figure 2.2. Illustration of the three possibilities in P-approach for 2-dimensional data. 95% ellipsoid and 95% univariate ranges are shown in bold. See description on page 28.

It should be noted that our belief that clean partial Mahalanobis distances follow a χ^2 distribution (with appropriate degrees of freedom) is based on the assumption that the removed values were indeed contaminated and therefore chosen at random from the point of view of the clean-data generating mechanism. Otherwise, removing the largest component of a clean case would bias the partial Mahalanobis distances downward. But even if this happens and we remove a clean component from a clean case, the process is likely to stop at the next step because the remaining partial Mahalanobis distance is even more likely to satisfy the constraint of being smaller than C_{p-k}. Therefore there is no need to adjust the assumed distribution of the partial Mahalanobis distances to accommodate the possibility of false positives.

2.3.3 Resolving "ties"

Suppose that after considering all possible $\mathrm{MD}^2_j, j = 1, \ldots, p$ we have found that both $\mathrm{MD}^2_{j_1}$ and $\mathrm{MD}^2_{j_2}$, for some $j_1 \neq j_2$, fall within the acceptable range of χ^2_{p-1}. What it tells us is that both $(p-1)$-tuples obtained by removing either j_1st or j_2nd variable from the case under consideration have passed the plausibility test and could be coming from the

assumed multivariate normal distribution.

Both variables j_1 and j_2 are suspected to be contaminated and the rest are believed to be clean based on available evidence so far. We envision two distinct ways to proceed in such a situation:

- Conservative. Remove both variables j_1 and j_2 from this case as both of them are suspected. This will reduce the probability of contaminated values passing through the detection process unnoticed (better recall). On the other hand, it is likely that many clean values will be removed mistakenly producing a higher rate of false positives and associated with it loss of efficiency and intrinsic bias. This might be an acceptable solution if only few (e.g. two) variables are suspected out of many. But if this happens at level $k \gg 1$ and $s \gg 2$ then it might lead to eliminating a large proportion of the data case.

- Greedy. Identify the variable (or set of variables if $k \geq 2$) which is the most likely to be contaminated and remove it from the case. This allows us to preserve as much clean data as possible while still having a cleaned data case as the result. Realistically we do not have any strong evidence against either one of the suspected values in particular and any decision we make is going to be a "best bet", assuming we do not find any other more conclusive evidence. To make the bet more likely to succeed we will compare what we have against both variables and pick the one that appears more suspicious. Here again we see two possible ways to decide which values are the most likely to be contaminated based on (a) implausibility of the suspected values; or (b) plausibility of the rest of the case after the suspected values are removed. We discuss both alternatives below.

Implausibility of the suspected values. Consider the easiest scenario: we have two variables j_1 and j_2 such that both $\mathrm{MD}^2_{j_1} < C_{p-1}$ and $\mathrm{MD}^2_{j_2} < C_{p-1}$, while the full Mahalanobis distance $\mathrm{MD}(\text{full}) > C_p$. We need to decide which of the two variables is the most improbable to have come from the assumed distribution and therefore is the most likely to be the contaminated value. In order to do this we attempt to use all available information about the assumed distribution that might have generated x_{j_1} and x_{j_2}. This includes the parameters of the p-variate normal distribution but also the information contained in this particular data case itself. When two values are suspected we want to exclude them from influencing our decision about each other. To do so we suggest to consider the conditional distribution of each of the two variables given all the values in this case that are deemed trustworthy at the moment — that is all other values except x_{j_1} and x_{j_2}. Assuming multivariate normality, these distributions would be normal and the parameters are easy to compute.

We can consider conditional z-values given the trustworthy data as a measure of implausibility of the suspected values — the larger their absolute value the less likely it is to

be coming from the assumed distribution. We will consider squared conditional z-values because they generalize easily to conditional Mahalanobis distances when $k \geq 2$. Compare

$$\left(\frac{x_{j_1} - \mathbb{E}\big[X_{j_1}|X_{-\{j_1,j_2\}}\big]}{\mathbf{sd}\big[X_{j_1}|X_{-\{j_1,j_2\}}\big]}\right)^2 \text{ vs } \left(\frac{x_{j_2} - \mathbb{E}\big[X_{j_2}|X_{-\{j_1,j_2\}}\big]}{\mathbf{sd}\big[X_{j_2}|X_{-\{j_1,j_2\}}\big]}\right)^2, \tag{2.18}$$

and tag the x_j with the largest result. In principle, both conditional squared z-scores might be well within the range of χ^2_1 but we still need to remove at least one variable to make the partial Mahalanobis distance acceptable. The cost of mistake in this somewhat arbitrary betting is not likely to be very high. Even if we let the contaminated variable slip through the detection process we have already made sure that it won't have significant influence on the estimate because the Mahalanobis distance of the remaining tuple is well under control. In other words, what remains of the case won't be an outlier anymore although it might still be contaminated.

When three or more variables are tied during the $k = 1$ stage, that is $s \geq 3$, then we simply need to consider s squared conditional z-scores instead of two, and each of them should be conditioned on $p - s$ trustworthy values.

When $k \geq 2$, ties can happen between subsets of variables instead of single variables. The strategy to find the subset that has the highest chance of being actually contaminated is exactly the same as in the one-variable-at-a-time situation. We can consider the plausibility of the subset of values given all trustworthy values in the case, which would be all values not involved in any of the suspected subsets. Suppose that s subsets of variables $S_1, \ldots, S_s$ result in reasonably small Mahalanobis distances $\mathrm{MD}^2_{S_1}, \ldots, \mathrm{MD}^2_{S_s}$ when removed from the case. What we want to look at are the following conditional Mahalanobis distances:

$$(\boldsymbol{x}_{S_l} - \hat{\boldsymbol{X}}_{S_l})' \left[\mathbf{Cov}(X_{S_l}|X_{-S_{\text{all}}})\right]^{-1} (\boldsymbol{x}_{S_l} - \hat{\boldsymbol{X}}_{S_l}), \text{ for } l = 1, \ldots, s, \tag{2.19}$$

where $S_{\text{all}} = \bigcup_{l=1}^{s} S_l$ and $\hat{\boldsymbol{X}}_{S_l} = \mathbb{E}[X_{S_l}|X_{-S_{\text{all}}}]$ and choose the largest of them. We show in Appendix A.2 that this conditional Mahalanobis distances can be represented as differences of two partial Mahalanobis distances with nested sets of variables. The expression in (2.19) is then equal to

$$\mathrm{MD}^2_{S_{\text{all}} \setminus S_l} - \mathrm{MD}^2_{S_{\text{all}}}, \text{ for } l = 1, \ldots, s, \tag{2.20}$$

where the first of the two terms above is the Mahalanobis distance of the case with all suspicious variables removed except those involved in S_l. Expression (2.20) looks cleaner than (2.19) but they have the same computational complexity. Depending on the situation one of the two might be easier to interpret than the other. In general we find that (2.19) is preferable when thinking about concepts, and (2.20) when dealing with computations.

Plausibility of the remaining values. Instead of evaluating how unlikely the potentially contaminated values are, we can instead consider how likely the *remaining values*

in the case appear under the assumed distribution. When dealing with outliers, smaller Mahalanobis distances correspond to more likely outcomes. Thus the smallest of the partial Mahalanobis distances MD_j will correspond to the most plausible combination of variables. The same is true when $k > 1$ — the smallest MD_{S_l} corresponds to the most likely $\boldsymbol{x}_{-S_l}$. Because partial Mahalanobis distances are already computed, this requires no additional computations to resolve the "tie" between the suspected sets which makes this approach computationally attractive.

When $k = 1$ and $s = 2$, this way of thinking actually leads us to the same conclusions as the implausibility approach described above. The largest squared conditional z-score corresponds to the largest of the two differences $\mathrm{MD}^2_{j_2} - \mathrm{MD}_{j_1,j_2}$ and $\mathrm{MD}^2_{j_1} - \mathrm{MD}_{j_1,j_2}$. Suppose that the first difference is larger and thus variable j_1 is tagged based on the implausibility thinking. Then $\mathrm{MD}^2_{j_2} > \mathrm{MD}^2_{j_1}$, which means that the partial Mahalanobis distance by removing j_1st variable is smaller and therefore it should be chosen based on the plausibility as well. So the two approaches are equivalent in this simple case. It is also equivalent to the implicit selection done by the D-approach described in section 2.3.5.

We show some numerical simulation results comparing the three methods (one conservative and two greedy) along with another approach in section 2.3.6.

2.3.4 Computational considerations

2.3.4.1 Inverses of submatrices

In terms of CPU time, the most demanding part of the procedure described in section 2.3.2 is the loop through all variables and combinations of variables looking for the one that produces the smallest partial Mahalanobis distance. If we are focusing on one case at a time then for each candidate variable two operations need to be done: (a) invert the corresponding submatrix of $\boldsymbol{\Sigma}$; (b) compute the Mahalanobis distance with it.

When dimension p is large, inverting a large number of large matrices can be a computationally intensive task. A matrix can be inverted by Gaussian elimination in $O(p^3)$ steps or by Strassen's algorithm in $O(p^{2.8})$ and it needs to be done for $\binom{p}{k}$ matrices at level k. Fortunately, however, inverses of large submatrices can be more efficiently deduced from the inverse of the whole matrix. Suppose that the inverse of a block matrix $\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}$, where $\boldsymbol{\Sigma}_{11}$ is $p_1 \times p_1$ and $\boldsymbol{\Sigma}_{22}$ is $p_2 \times p_2$, is known and we are interested in the inverse of its submatrix $\boldsymbol{\Sigma}_2$. Recall the following expression for the inverse of a block matrix from Petersen and Pedersen (2008):

$$\boldsymbol{A}^{-1} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{C}_1^{-1} & -\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}\boldsymbol{C}_2^{-1} \\ -\boldsymbol{C}_2^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1} & \boldsymbol{C}_2^{-1} \end{pmatrix}, \tag{2.21}$$

where $\boldsymbol{C}_1 = \boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}$ and $\boldsymbol{C}_2 = \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}$ are the Schur complements

of $\boldsymbol{A}$. Apply equation (2.21) to $\boldsymbol{A} = \boldsymbol{\Sigma}^{-1}$ and look at the bottom right corner submatrix. Then $\boldsymbol{\Sigma}_{22} = \boldsymbol{C}_2^{-1}$ and thus

$$\boldsymbol{\Sigma}_{22}^{-1} = \boldsymbol{C}_2 = (\boldsymbol{\Sigma}^{-1})_{22} - (\boldsymbol{\Sigma}^{-1})_{21} \left((\boldsymbol{\Sigma}^{-1})_{11}\right)^{-1} (\boldsymbol{\Sigma}^{-1})_{12}. \tag{2.22}$$

It is easy to see that computing this inverse only requires $O(pp_2p_1)$ operations plus those needed to invert the whole $\boldsymbol{\Sigma}$ and $(\boldsymbol{\Sigma}^{-1})_{11}$. The overall inverse only needs to be computed once per dataset and can easily be done in $O(p^3)$. Inverting $(\boldsymbol{\Sigma}^{-1})_{11}$ takes $O(p_1^3)$ and is negligible if p_1 does not grow with p. In fact, if storage space is not an issue and we can save all $\binom{p}{k}$ inverses from step k, then we only need to apply this inversion rule sequentially with $p_1 = 1$ when going to the next level. So the computing time necessary for one inversion procedure is of order $O(p^2)$ which is a good improvement over $O(p^{2.8})$.

Computing Mahalanobis distance of a p dimensional case once the inverse of the covariance is known also takes $O(p^2)$ steps which is the same complexity as inverting the matrix. In particular it means that we cannot save much computational time (at least not an order of magnitude) by reusing the covariance inverses from case to case because computing Mahalanobis distances is just as slow as inverting submatrices of $\boldsymbol{\Sigma}$. In practice, we will, of course, reuse the inverted submatrices when convenient in order to save finite computational time.

2.3.4.2 Limiting the depth of search

The price to pay for the ability of P-approach to detect obscured combinations of univariate contamination is its computational complexity. At level k, the number of inversions is $\binom{p}{k}$ and the number of Mahalanobis distances needed to be computed is $n_k\binom{p}{k}$, where n_k is the number of cases that we have not been able to clean by removing $k-1$ variables at a time. Therefore the whole screening procedure takes an order $O(n_k p^{2+k})$ basic operations to complete the kth level.

Under independent contamination model, for small fractions ε of contaminated data that we expect to see in a typical dataset, majority of cases will not have many variables contaminated. The distribution of the number of contaminated values per case is $\text{Binom}(p, \varepsilon)$. Each new level requires looking through more and more combinations of variables, but once k becomes larger than $\varepsilon \times p$, the number of cases that have that many variables contaminated is getting smaller and smaller. It means that more work is required to clean up fewer cases and may become very "inefficient" at some point.

To make the detection algorithm computationally feasible we suggest to limit the number of levels that it is allowed to dig into the data. All cases that are not cleaned (all partial MDs are larger than C_{p-k}) can be declared fully contaminated and removed from further analysis. Elimination of complete cases is a relatively safe process from the intrinsic bias point of view because, assuming IC model, the number of contaminated variables is independent of the clean case before contamination. The worst consequence of this early

stopping is the reduced efficiency due to the loss of potentially useful data. To control this effect the stopping rule can be based on the proportion of uncleaned cases n_k/n rather than k itself.

If computational time is of no concern or if the dimension p is low then, of course, the full detection until $n_k = 0$ can be performed. In our simulations, because of the repetitive nature of the process, we are concerned with computational time and will routinely use $n_k/n < 0.25$ as the stopping rule for detection.

2.3.5 Differences of Mahalanobis distances (D-approach)

2.3.5.1 Contributions to Mahalanobis distances

As explained in section 2.3.4.2, looking through all subsets of size k out p variables can be a time consuming operation as k grows large. One possible way to reduce computational demands is to use stepwise detection by choosing one most contaminated variable at a time and keeping it out of consideration for all consequent levels.

The basic premise of this approach is that individual variables in each case *contribute* to the full Mahalanobis distance. If the full distance is large and we can single out the variable that contributes the most to it then we have reasonable evidence to believe that this variable is contaminated. The contribution of the jth variable can be measured by the difference between the full Mahalanobis distance for the case and the partial distance with that variable removed:

$$D_j = \mathrm{MD}^2(\text{full}) - \mathrm{MD}_j^2 = \mathrm{MD}^2(\boldsymbol{x}; \boldsymbol{m}, \boldsymbol{\Sigma}) - \mathrm{MD}^2(\boldsymbol{x}_{-j}; \boldsymbol{m}_{-j}, \boldsymbol{\Sigma}_{[-j]}).$$

After computing D_j for all $j = 1, \ldots, p$, variable $j_1 = \arg\max_j D_j$ is declared contaminated and is to be removed from further analysis. If $\mathrm{MD}_{j_1}^2$ is in accordance with χ^2_{p-1} distribution, i.e. smaller than C_{p-1}, then we can declare this case decontaminated and proceed to the next case. If the partial Mahalanobis distance is still large then more values must be contaminated within this case and the procedure can be repeated by considering all differences

$$D_{j_1,j} = \mathrm{MD}_{j_1}^2 - \mathrm{MD}_{j_1,j}^2 = \mathrm{MD}_{j_1}^2 - \mathrm{MD}^2(\boldsymbol{x}_{-\{j_1,j\}}; \boldsymbol{m}_{-\{j_1,j\}}, \boldsymbol{\Sigma}_{-[j_1,j]}), \text{ for } j \neq j_1.$$

and declaring variable $j_2 = \arg\max_j D_{j_1,j}$ as contaminated. This process should be repeated until the remaining partial Mahalanobis distance with some k variables removed is small enough to be believed to have come from χ^2_{p-k}.

This method of detecting independent contamination was proposed by Little and Smith (1987) in the context of analyzing survey data. They however neither defined the contamination model nor studied the properties of the suggested method. However, they gave a thorough hands-on illustration of how it can be applied in practice and a motivation for why such a method might be useful.

The major advantage of this approach over the P-approach of section 2.3.2 is that it only takes $(p-k)$ inverses and $n_k^*(p-k)$ computations of Mahalanobis at level k; which do not increase with k.

On the other hand, we can see two potential problems with this approach that we will discuss in more details in the next two subsections. First, to be discussed in section 2.3.5.2, is that the conceptual foundation for deciding which variable is contaminated is flawed. Second, more practical and illustrated in section 2.3.5.3, is the usual drawback of stepwise procedures compared to full searches: it can fail to correctly identify contamination if more than one variable is involved.

2.3.5.2 Distribution of D_j

One major problem with D-approach is that we deal with quantities whose distribution we know very little about. When the case is contaminated then every D_j, for all $i, j = 1, \ldots, p$, contains at least one contaminated value and therefore their distribution is completely unknown and depends heavily on the contamination. We choose to label the variable that produces the largest D_j as contaminated but this approach — formulated this way — is completely ad hoc and has no distributional background to back it up.

When $k = 1$ this method is equivalent to the first level of D-approach when the simplified rule of resolving ties is applied. As $\mathrm{MD}^2(\text{full})$ is fixed for a given case, choosing the largest difference D_j is equivalent to choosing the smallest partial Mahalanobis distance MD_j^2. The D-approach looks for a variable j that *maximizes its contribution* to the Mahalanobis distance and thus might seem more intuitive than the method that *minimizes* Mahalanobis distance of *the rest of the case*. This intuitiveness, however, is misleading because the former method chooses the *worst* value out of a collection of *bad* values and therefore we know little about what to expect from it. When doing the latter, on the other hand, we have some solid ground to stand on: if the contaminated variable is removed then the remaining $(p-1)$-Mahalanobis distance is distributed as χ^2_{p-1}. The distinction becomes more apparent as we proceed to detect more than one contaminated variable per case.

The lemma in Appendix A.2, applied with $p_1 = 1$, provides a good interpretation for the differences D_j. Under multivariate normal uncontaminated data they can be seen as squared conditional z-scores for each value given the remaining of the case and are distributed as χ^2_1. This is parallel to the univariate detection except that now the information from the same data case is used to enhance the expected value and the standard deviation for each cell.

When at least one value in $\boldsymbol{x}$ is contaminated then for every variable j necessarily either x_j or $\boldsymbol{x}_{-j}$ (or both) is contaminated and therefore the distribution of the squared conditional z-score is not χ^2_1 anymore and is affected by the contamination. Therefore none of the differences D_j will be following χ^2_1 for any data case that we are interested in

cleaning.

If the process of removing one variable at a time is repeated until all p variables are used then the full Mahalanobis distance can be decomposed into p incremental terms. When the case is completely clean and comes from $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ these p terms are independent and distributed as χ^2_1 each (if we assume that the ordering of the variables was arbitrary and forget that the largest difference is chosen at each step). This is a meaningful construct and this is what makes the D-approach appealing: each variable independently contributes something to the full Mahalanobis distance of the case and we want to detect and disarm the largest contributor. The moment at least one component of the case gets contaminated, the independence and the χ^2_1 distributional claims do not hold anymore and all differences get tied together and interconnected in an unpredictable way through the non-gaussian contaminated data.

2.3.5.3 Example: failure of stepwise D-approach

As we have mentioned above, besides the conceptual difference between P- and D-approaches, the largest computational distinction is that the D-approach attacks the problem of finding contaminated values in a stepwise manner instead of the full search through all combinations of variables. While being obviously faster at levels $k \geq 2$, this poses a significant risk of missing contamination that would be very obvious if an appropriate method were used. The D-approach looks (intentionally or not) at the difference between the observed value of each variable and its expected value given all other variables in the same case. When more than one variable is contaminated it is possible that *masking* will occur: a contaminated value may look plausible given another contaminated value while good values will have unusually large z-scores when conditioned on all the contaminated variables.

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	MD(rem)
$\boldsymbol{x}_{\text{clean}}$	-1.8	0.2	0.0	-1.4	-1.8	5.8
$\boldsymbol{x}_{\text{cont}}$	-1.8	**6.0**	**6.0**	-1.4	-1.8	176.3
D_j	0.35	122.0	94.6	**132.0**	82.7	176.3
$D_{4,j}$	**1.85**	0.54	1.48	—	0.62	44.2
$D_{4,1,j}$	—	2.70	0.35	—	**5.13**	42.4
$D_{4,1,5,j}$	—	1.26	1.26	—	—	37.3

Table 2.4. Failure of the D-approach to identify contamination that is masking one another.

Consider the following example summarized in Table 2.4. A 5-dimensional high-correlation covariance matrix (see Section 2.1.4.3) is used. We start with a typical random value $\boldsymbol{x}_{\text{clean}}$ drawn from the multivariate normal distribution with mean zero and said covariance matrix. Variables 2 and 3 are severely contaminated by values of 6.0 to produce a contaminated vector $\boldsymbol{x}_{\text{cont}}$ that all further analysis will be based on. Mahalanobis distance goes up from a typical 5.8 to a very outstanding 176.3 that calls for a cleaning procedure to be applied. If we were to use the D-approach and decide which values are

contaminated based on the corresponding values of $D_j = \mathrm{MD}^2(\text{full}) - \mathrm{MD}_j^2$, by removing one variable at a time, then the innocent variable number 4 would be singled out based on the $D_4 = 132.0$. Mahalanobis distance of the remaining tuple is still 44.2 that justifies further cleaning. The next variable to go at the $k = 2$ step is number 1 with the not-so-large value of $D_{4,1} = 1.85$ which does not improve the overall Mahalanobis distance much either. Finally, going even deeper, variable number 5 is removed by the "cleaning" process, leaving us with a completely contaminated 2-tuple. This is obviously a total failure on the part of the cleaning mechanism because it removes all the good values until only contamination is left intact. Fortunately, the D-approach would not stop at $k = 3$ as the Mahalanobis distance is still large and would proceed to remove both contaminated values one after the other, leaving us with an empty case. Nevertheless, this example clearly illustrates the deficiencies of the stepwise approach to cleaning.

On the other hand, if P-approach is used it will yield much more satisfactory results. At level 1, the minimal partial Mahalanobis distance is $\mathrm{MD}_4^2 = 176.3 - 132.0 = 44.2 \gg C_4$ indicating that removing one value is not enough to make this case clean. Then P-approach proceeds to compute $\binom{5}{2} = 10$ partial distances by removing 2 variables at a time. They can be easily divided into two unequal groups: 9 very large (with minimal value 42.4) and one value $\mathrm{MD}_{2,3}^2 = 4.98$. This is the perfect outcome number 1 as discussed in section 2.3.2 which means that variables 2 and 3 are to be called contaminated and the rest of the case can be assumed to be clean.

2.3.6 Numerical comparison of detection methods using known covariance

To compare the several approaches to multivariate contamination detection described earlier in this section we conduct a Monte Carlo simulation study. We assume that the true location and scatter is known. Section 2.3.9 at the end of this chapter will have some results for when the true parameters are unknown and have to be estimated.

These simulations are similar to those in section 2.2.3.3. We generate independent samples from $\mathcal{N}_{20}(\mathbf{0}, \mathbf{\Sigma}_{\mathrm{m}})$, where $\mathbf{\Sigma}_{\mathrm{m}}$ is a covariance matrix with moderate correlation (as per section 2.1.4.3). The samples are subjected to the point mass independent contamination at value k with cell-probability $\varepsilon = 0.10$. We vary the value of k in the interval $[0, 4]$. To each contaminated sample we apply four detection algorithms:

1. P-approach with Implausibility tie-breaking (based on conditional Mahalanobis distances given all unsuspected values). We will refer to it as "P.full".

2. P-approach with Plausibility tie-breaking (based on the smallest partial Mahalanobis distance). We will refer to it as "P.fast".

3. P-approach with Conservative tie-handling (remove all suspected values). We will refer to it as "P.all".

4. D-approach (stepwise). We will refer to it simply as "D".

Cutoff p-value of $p_{\text{cutoff}} = 0.05$ was used for all four detection methods. For each case, the fraction of contaminated cells that has been correctly identified (recall) and the proportion of tagged cells that were truly contaminated (precision) is recorded. The whole procedure is repeated $N_{\text{rep}} = 20{,}000$ times and the average recall and precision are reported in Figure 2.3.

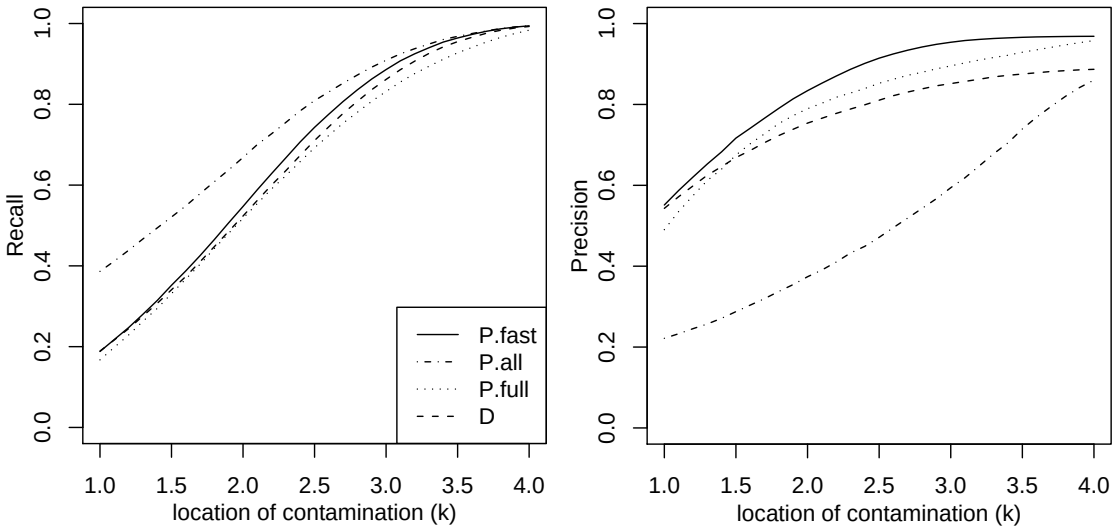


Figure 2.3. Recall/precision performance of the four multivariate detection methods when the true covariance structure is known.

Looking at the graph a number of observations can be made:

1. Not surprisingly, contamination which is farther from the center is easier to detect for all methods.

2. Precision of the conservative tie resolution (P.all, dash-dotted line) is fairly low and is unlikely to be justified even by the somewhat better recall rate.

3. Plausibility-based tie resolution in P-approach (P.fast, solid line) actually works better than the more complicated implausibility approach (P.full, dotted line). Our understanding is that some potentially useful information is lost when we condition the suspected values only on those that do not appear in any of the suspected sets, which makes the decision process more arbitrary.

4. P-approach with fast tie resolution (solid line) does indeed perform better than D-approach (dashed line). The difference is less obvious in terms of recall because both approaches proceed removing values from a data case until it appears to have reasonably small Mahalanobis distance. The problem is that D-approach takes more

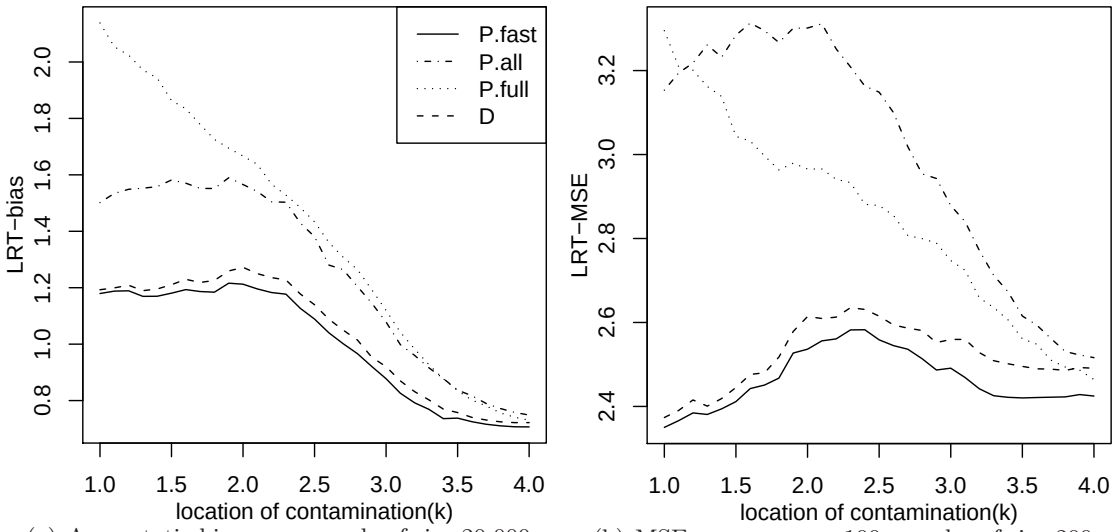
unnecessary steps by removing clean values to get there. This is reflected in its lower precision rate.

We have also run these simulations on data with high correlation (instead of moderate) and, although the overall detection rate is non-surprisingly higher, the trends are the same as discussed above. This makes us believe that the results should be generalizable to a variety of data configurations.

To investigate the importance of the precision loss of D-approach and P-approach with conservative tie resolution we go one step further and compute the MLE of the covariance matrix using the detection information from the four methods. We have 20,000 individual samples at our disposal and we use them in two different ways to learn more about the performance of the estimates:

1. Compute MLE based on one large sample of size 20,000 for each of the four methods and each value of k. This will describe the asymptotic bias of the estimates based on the four detection methods. Results are shown in Figure 2.4(a).

2. Divide available cases into 100 samples of size $10p = 200$; then compute the MLE of covariance based on each smaller sample independently. This will describe the finite sample MSE performance of the estimates. Results are shown in Figure 2.4(b).



(a) Asymptotic bias: one sample of size 20,000

(b) MSE: average over 100 samples of size 200

Figure 2.4. Performance of the estimates of covariance based on the four multivariate detection methods when the true covariance structure is assumed to be known.

By analogy to section 2.2.3.3, after computing each estimate we standardize it using the true value $\mathbf{\Sigma}_{\text{m}}$ and compute the LRT-statistic based on it. In Figure 2.4(a), the LRT statistic itself is reported. In Figure 2.4(b), the average of 100 replicates of the LRT

statistic is shown. We can clearly see that P.full (dotted line) and P.all (dash-dotted line) perform significantly worse than the other two estimates. In particular, we can see that the slightly better recall for the price of significantly reduced precision of the the conservative tie resolution in P-approach (dash-dotted line) did not pay for itself even in terms of asymptotic bias and even more so in terms of finite sample MSE. D-approach and P-approach with plausibility-based tie resolution perform much better, with P-approach being the best of the four. We will use this detection method as default in the next section where we consider how the covariance matrix can be estimated when it is unknown.

2.3.7 Unknown true covariance matrix

2.3.7.1 Iterative estimation and detection

The covariance matrix needs to be *estimated* in order to detect contaminated values using multivariate methods discussed in this section. The situation is intrinsically circular: a covariance estimate is required in order to compute a covariance estimate. This is not unlike re-weighted robust estimates when a set of weights is required to compute a weighted estimate that can, in turn, be used to compute a new set of weights. We propose a similar procedure to be used regarding our covariance estimates. First, get a rough covariance estimate and use it to detect the first batch of contaminated values. Then use the filtered data to obtain a covariance estimate with one of the methods mentioned in section 2.2.3. This new estimate is expected to be better than the initial one and therefore we can use it to improve the contamination detection. This process can be repeated several times for best results.

The iterative procedure described in this subsection is only a general framework rather than a complete algorithm. In order to make use of it one will need to choose three core components:

1. Initial estimate of covariance. We suggest using univariate detection (followed by an MLE) as step 0 for reasons described in section 2.3.8. Moreover, values detected as contaminated using a conservative univariate method are to be remembered and treated as missing throughout the estimation process.

2. The detection process for identifying outliers given a covariance matrix. We will use the P-approach as we consider it to be the most conceptually sound.

3. The processing step: Winsorised, censored, ML assuming MAR, or a combination of them to estimate covariance matrix given the results of the detection process. We will use the Marking-as-MAR approach because of its strong ability to eliminate all negative influence of outliers.

One other question that needs to be answered before one can have a working algorithm is how to update the information regarding which values are contaminated from

one iteration step to another. Two basic options can be thought of: (a) retain the list of contaminated values from previous steps, use it during the detection process (i.e. not use previously suspected values in conditional quantities), and only *expand the list* at each iteration; (b) forget about old suspected values as soon as a new covariance estimate is computed and create a new list from scratch with the help of the new estimate. The former approach may lead to an increasingly large list of suspected values that will result in increasingly ill conditioned estimates. We advocate the use of the second approach with one exception that the information from the univariate step 0 is preserved throughout the process (because of its conservatism).

Finally, a stopping rule for the iteration process has to be agreed on. There are two major entities that change from one step to another and they are closely related: the set of detected contaminated values and the estimated covariance matrix. Ideally, the process will converge and the set of detected values will remain constant from one iteration to the next. In practice, however, it is common that the process enters a cycle with a period of two or more iterations and keeps going trough the same set of states over and over again. The iteration procedure is a Markov process by nature: all future iterations depend solely on the current configuration. The set of all possible configurations is finite and only few of them have reasonable probabilities of being visited. Therefore we can keep a record of all visited states and as soon as any state is revisited the iterations can be stopped. A maximum number of iteration will be imposed to make sure the process does not run out of control.

2.3.7.2 Adjustable cutoff values

One inherent drawback of the described iterative procedure is that undetected contamination at step i will affect the subsequent covariance estimate which, in turn, can make the contamination undetectable at step $i+1$. Bias induced to the covariance matrix by independent contamination, as discussed in section 2.1.3, generally makes the detection process less sensitive (overestimated variances, underestimated off-diagonal elements). This can go on forever and the contamination never becomes detected even though it could be easily identified had the true covariance been known.

It is unlikely that this problem can be solved completely but we propose an adjustment that will help detect borderline contamination at least in some situations. After conducting a series of experiments, and relying on algebraic findings of section 2.1.3, we have concluded that the quantities which are affected the most by unfiltered independent contamination are the estimates of individual variances, i.e. the diagonal of the covariance matrix. We propose to compare these estimated variances to something solid and reliable in order to evaluate whether filtering was sufficiently thorough or not.

There exist several reliable and commonly used robust estimates of univariate variance but, in order to fix ideas, we will consider squared Median Absolute Deviation (sMAD)

as *the* one-dimensional variance estimate and denote it $\hat{s}_j^2$ for the jth variable. The main advantage of these estimates is that they are truly robust and do not depend on any tuning parameters. On the other hand they are very simple and do not share potentially useful information between variables. It is possible that a sophisticated method will perform better than this simple alternative but the reverse is extremely undesirable. Most dangerous contamination comes in the form of outliers and therefore it will cause the variance to be overestimated. Therefore, *better* can be interpreted as *smaller* (within limits). In particular, we may want to require that the diagonal elements of our multivariate covariance estimate be at least as small as the sMADs.

If multivariately estimated variances are larger than sMADs it means that not all contamination has been detected and filtered, which calls for increased sensitivity of the detection mechanism. Aside from the covariance matrix itself, it is the cutoff value (expressed as p-value because dimensionality varies between levels) that is responsible for the sensitivity of the process. If estimated variances are too large (compared to sMADs) then the cutoff p-value needs to be increased (smaller cutoffs).

A decision rule needs to be set up describing when estimated variances are considered to be too large. The simplest method, that we employed in our simulations, is to compare the two average variances and if the "multivariate" average is larger than sMAD's then the cutoff p-value is to be increased. As the process is already iterative we suggest that adjustments to the cutoff are made in small increments. For example, p-value can be multiplied by a fixed factor slightly larger than one. In summary: on each iteration, after obtaining a new covariance estimate the following check should be done:

$$\text{if } \operatorname*{ave}_j(\operatorname{diag}(\hat{\mathbf{\Sigma}})) > \operatorname*{ave}_j(\hat{s}_j^2) \text{ then } p_{\text{cutoff}} = \min\{1.01 \times p_{\text{cutoff}}, p_{\max}\},$$

where p_{cutoff} is the p-value used to obtain cutoff values $C_k = (\chi_k^2)^{-1}(1 - p_{\text{cutoff}})$. This rule should not be invoked until after several initial iterations allowing the system to settle to a near-stable state. The upper bound $p_{\max}$ for p_{cutoff} is set as a precaution to safeguard against filtering out all data if something goes unexpectedly wrong. If contamination cannot be detected with these relatively small cutoffs (large p-value) then it could be declared undetectable and perhaps a flag should be raised warning that the estimate is unreliable. A reasonable value could be $p_{\max} = 0.1$.

Stopping rule for the iterative process has to be modified to accommodate the adjustments of cutoffs. Now we only want to stop if, in addition to running into a previously visited state, one of the following two conditions are true: (a) current estimate already agrees with sMADs, i.e. $\text{ave}_j(\operatorname{diag}(\hat{\mathbf{\Sigma}})) \leq \text{ave}_j(\hat{s}_j^2)$; or (b) contamination seems undetectable and we have given up, i.e. current $p_{\text{cutoff}} \geq p_{\max}$.

Figure 2.5 shows a typical run through iterations for a 10-dimensional dataset of size 200 with medium covariance structure and 10% contamination placed at 2.5 standard deviations from the center. The first plot shows *recall* of the detection process,

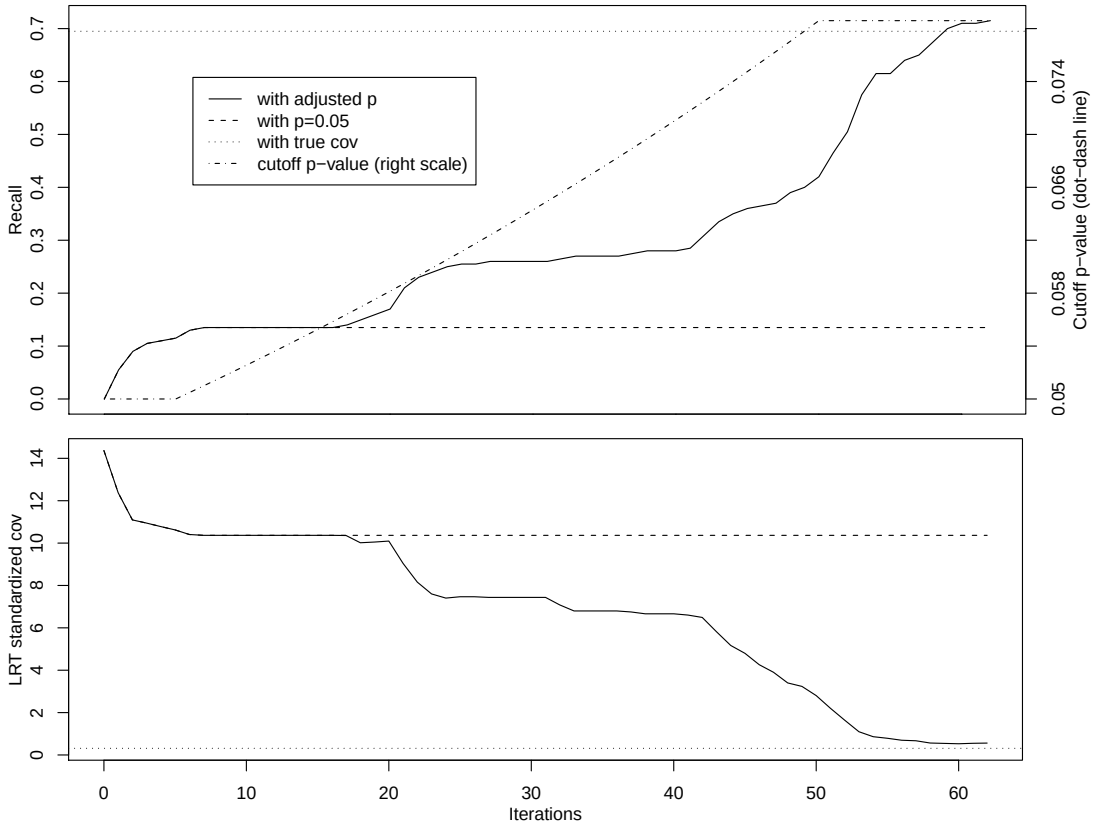


Figure 2.5. Positive effect of adjusting cutoff p-value during iterations (a typical run).

i.e. the proportion of contamination that has been correctly identified: large values correspond to better performance. The second plot shows the LRT-measure (described in section 2.1.4.3) of the standardized covariance estimate ($\boldsymbol{\Sigma}^{-1}\hat{\boldsymbol{\Sigma}}$) after removing (marking as MAR) the detected contamination: the closer it is to zero, the better the estimate is. Solid lines correspond to iterations with cutoff values adjusted as described above, dashed lines — to unadjusted cutoff p-value of 0.05, and dotted lines indicate what could be achieved if the true covariance matrix was used for detection instead of the estimates. Dash-dotted line in the first plot shows (on a different scale given on the right) how the cutoff p-value was increasing during the iterations; it is not a performance measure but simply a monitoring tool. It can be seen that after a dozen of iterations with fixed cutoffs (dashed lines), the system fully stabilizes (the set of detected values does not change from one iteration to another) but the performance is still relatively poor. Allowing the cutoff p-value to gradually increase to 0.087 helps detect as much contamination as can be detected if the true covariance was known (with $p_{\text{cutoff}} = 0.05$) without jeopardizing precision too much. Precision is not shown on the graph but it floats within $(0.8, 0.9)$ interval and even improves slightly due to the adjustments of the cutoff p-values.

Note that this is a typical scenario for moderate correlation structure and borderline

contamination. If the correlation structure is stronger or the contamination is more outlying then no adjustments of cutoff values might be necessary in order to detect it. If the correlation structure is weaker then it might very well be possible that such borderline contamination cannot even be detected even with the true covariance matrix, making it futile trying to detect it with estimated matrices.

2.3.8 Combination with univariate filtering

2.3.8.1 Univariate vs multivariate detection

In Section 2.2.2 we described univariate and multivariate methods as two distinct approaches to independent contamination detection. In reality, however, the two approaches are related but the relationship is not entirely straightforward.

The strongest parallels can be drawn when the assumed covariance structure is diagonal. As we have shown in section 2.3.5.2 the differences D_j of Mahalanobis distances can be seen as squares of conditional z-values given the rest of the values in the same case. Assuming independence of variables, conditional z-scores are equal to the corresponding marginal z-scores, which are the primary tool of univariate detection. This algebraic equality is the similarity between the two approaches.

The differences arise from the way we analyze the Mahalanobis distances. To begin with, univariate detection looks at *all* $n \times p$ marginal z-scores and highlights those that exceed a certain threshold. Multivariate methods only look at those *cases* that have unusually large *full* Mahalanobis distances. And once a case is suspected, the multivariate methods will end up declaring at least one value contaminated. It means that the decision of whether contamination is present or not is based on the full Mahalanobis distances and not on the partial distances that only affect decisions about *which* variables are contaminated.

First, consider a rather typical scenario when multivariate detection is more sensitive. This happens when contaminating value is moderate (i.e. not large) and the covariance matrix has enough structure so that the conditional variance of each value given the rest is much smaller than the marginal. Then the marginal z-value is not large enough to raise an alarm but the conditional z-value might be much larger because it involves reciprocal of a small conditional standard deviation. Large conditional z-value inevitably contributes to the full Mahalanobis distance and the case goes on the suspect list.

Example 1. Consider a five dimensional high-correlation covariance matrix $\mathbf{\Sigma}$ as described in section 2.1.4.3. Take a typical observation $\boldsymbol{x}_{\text{clean}} = (-0.3, -1.3, -1.7, -0.4, 0.8)'$ from $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Contaminate the second variable with a moderate value of 1.8, resulting in the observed case $\boldsymbol{x}_{\text{cont}} = (-0.3, \mathbf{1.8}, -1.7, -0.4, 0.8)'$. Just by looking at this case one value at a time nothing is suspected to be contaminated as all five values are typical of the standard normal distribution. However, the full Mahalanobis distance $\text{MD}^2(\boldsymbol{x}_{\text{cont}}) = 885.8$ positively indicates contamination. P-approach then considers five partial Mahalanobis

distances by removing one variable at a time: 797.1, 4.1, 13.3, 116.1, 330.0 and easily identifies that the second value is contaminated. Cases like this are the reason why multivariate detection methods are preferred to univariate alternatives. Of course, not every case is as extreme as this one but the general tendency is illustrated.

Unfortunately, however, there are situations when multivariate methods fail to detect obvious univariate contamination. The primary reason are the thresholds used to identify contaminated cases which increase with dimension (degrees of freedom of χ^2). It is possible for the full Mahalanobis distance to be less than C_p while some individual value yields a squared z-value in excess of C_1. This is typical for covariance matrices with low correlation structure.

Example 2. Consider a no-correlation identity covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{I}_{20}$. Let $\boldsymbol{x}_{\text{clean}}$ be a sample from near the center of $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_{20})$ such that $(x_{\text{clean}})_1 = -1.3$ and the sum of squares of the remaining 19 coordinates is 6.29 (we are not showing all 20 dimensions purely for the sake of conciseness). Suppose that the first coordinate of $\boldsymbol{x}_{\text{clean}}$ is contaminated by a large value of 5, yielding an obviously contaminated case $\boldsymbol{x}_{\text{cont}}$. A univariate detection method will have no problem identifying the first value as contaminated because the p-value corresponding to the z-score of the first variable is a minuscule 5.7×10^{-7}. Even with Bonferoni correction for the multiple testing of 20 variables it will still be detected as outlier at any reasonable level of significance. A multivariate method starts by computing the full Mahalanobis distance $\text{MD}^2(\boldsymbol{x}_{\text{cont}}) = 5^2 + 6.29 = 31.29$ which yields a p-value of 0.051 (based on χ^2_{20}) that would not raise an alarm at any significance level smaller than 0.05. Although $(x_{\text{cont}})_1^2 = 25$ is a very unusual value for χ^2_1, when combined with relatively small values of the rest of the variables which contribute almost nothing to the full Mahalanobis distance, it is still quite typical for χ^2_{20}.

The root of the problem is that the range of χ^2 distribution increases with the degrees of freedom. Increase of only 6.18 square units is required to bring a typical (median) χ^2_1 observation to the top 1% tail. For χ^2_{20}, for example, an equivalent perturbation (from median to 99th percentile) requires an increase of 18.23 square units. When covariance structure is weak and variables contribute to the full Mahalanobis distance more or less independently, it requires a larger contamination to set off a 20 dimensional filter than it does for univariate detection.

p	Range	Rel Range	Bonf RR
1	6.18	1.00	1.00
5	10.73	1.74	1.18
10	13.87	2.24	1.34
20	18.23	2.95	1.56
50	26.82	4.34	2.00
100	36.47	5.90	2.48
200	50.11	8.11	3.13

Table 2.5. Range $(\chi^2_p)^{-1}_{0.99} - (\chi^2_p)^{-1}_{0.5}$ as function of dimension.

Table 2.5 summarizes the ranges of χ^2 distribution for a variety of dimensions. The first column is the range itself showing how much a median observation needs to be increased in order to become a 99th percentile outlier. Second column is the ratio of this range for the given dimension to the same range of χ^2_1. It illustrates how much larger (in square terms) a univariate outlier should be in order to be detected by a p-dimensional methods when covariance matrix is diagonal. To show that this problem is deeper than just multiple testing issues, the third column shows the same range as the second but assumes that the significance level in $p = 1$ has been Bonferoni corrected:

$$(\text{Bonf RR})_p = \frac{(\chi^2_p)^{-1}_{0.99} - (\chi^2_p)^{-1}_{0.5}}{(\chi^2_1)^{-1}_{1-0.01/p} - (\chi^2_1)^{-1}_{0.5}}$$

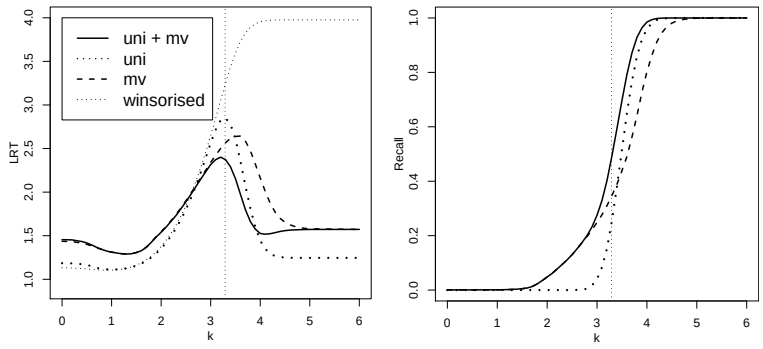
This *obscuration of contamination* is an inherent property of Mahalanobis distances and is not necessarily a weakness of our approach. It is just another manifestation of the curse of dimensionality and should be recognized as such. With strong correlation structure multivariate methods are likely to be more sensitive but when covariance matrix is nearly diagonal they may (or may not) merely obscure individual contaminated values by mixing them with other uncontaminated values in the same case. The conclusion is that both univariate and multivariate detection methods must have their place in the estimation process.

We suggest that every multivariate detection process is preceded by a round of univariate detection with a large conservative cutoff. It will ensure that contamination described in Example 2 does not go undetected. Keeping the cutoff value large will ensure that the number of false positives is negligible and no unnecessary intrinsic bias is introduced. Simulations in section 2.3.9 illustrate this danger numerically. The computational cost of univariate detection is also negligible compared to the multivariate methods. It is best seen as a precautionary safeguard and may not yield any significant improvements in performance in most situations. One example when the addition of univariate detection does result in a better estimate is shown in the next sub-section.

2.3.8.2 Numerical illustration

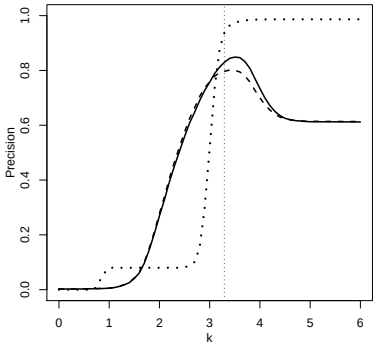
To illustrate the potential benefits of using univariate detection to complement multivariate procedures we conduct a simple simulation study. Samples of size $n = 200$ from multivariate normal distribution with $p = 20$ and *low correlation* structure (as described in section 2.1.4.3) are generated. Univariate contamination at 5% level is introduced randomly and independently into all 20 variables. All contaminated values are set to be equal to k, which we vary from 0 to 6. We compute three scatter estimates (assuming the true covariance is unknown) using different detection procedures: (a) univariate detection at 0.999 level (cutoff at 3.29); (b) multivariate detection at 0.95 level using iterative procedure outlined in section 2.3.7; (c) both of them combined. Estimates are

computed using EM-algorithm on the datasets with induced missingness. As an additional reference, we also computed Winsorised sample covariance for each contaminated sample. Estimates are standardized, d_{LRT} is computed (see section 2.1.4.3) and the whole process is repeated 600 times to get the average measure of how much the estimates deviate from their target value.



(a) LRT measure

(b) Recall



(c) Precision

Figure 2.6. Advantages of univariate detection for high-dimensional data ($p = 20$) with weak correlation structure.

Simulation results are shown in Figure 2.6. In addition to the average LRT score we show the average recall and precision rates. The main observation that can be made from these plots is that the pure multivariate estimate (bold dashed line) can be improved by incorporating (bold solid line) univariate detection information into the estimate. When k is between approximately 3 and 5 all three graphs show an improvement: lower LRT in the first and higher rates in the other two. The region $k \in (3, 5)$ is when the conservative univariate detection starts identifying outliers as such but they are still not large enough to put the whole case on the suspect list because of the wider spread of χ^2_{20}. This is a typical scenario when true correlation structure is close to identity: sharing information between variables does not significantly help improve the detection of outliers. Even apparently higher recall and precision rates of multivariate detection compared to univariate are likely due to the lower multivariate cutoff. An illustration of the benefits of multivariate

detection when true correlation is *moderate* is given in section 2.3.9.

2.3.9 Simulation study with unknown covariance

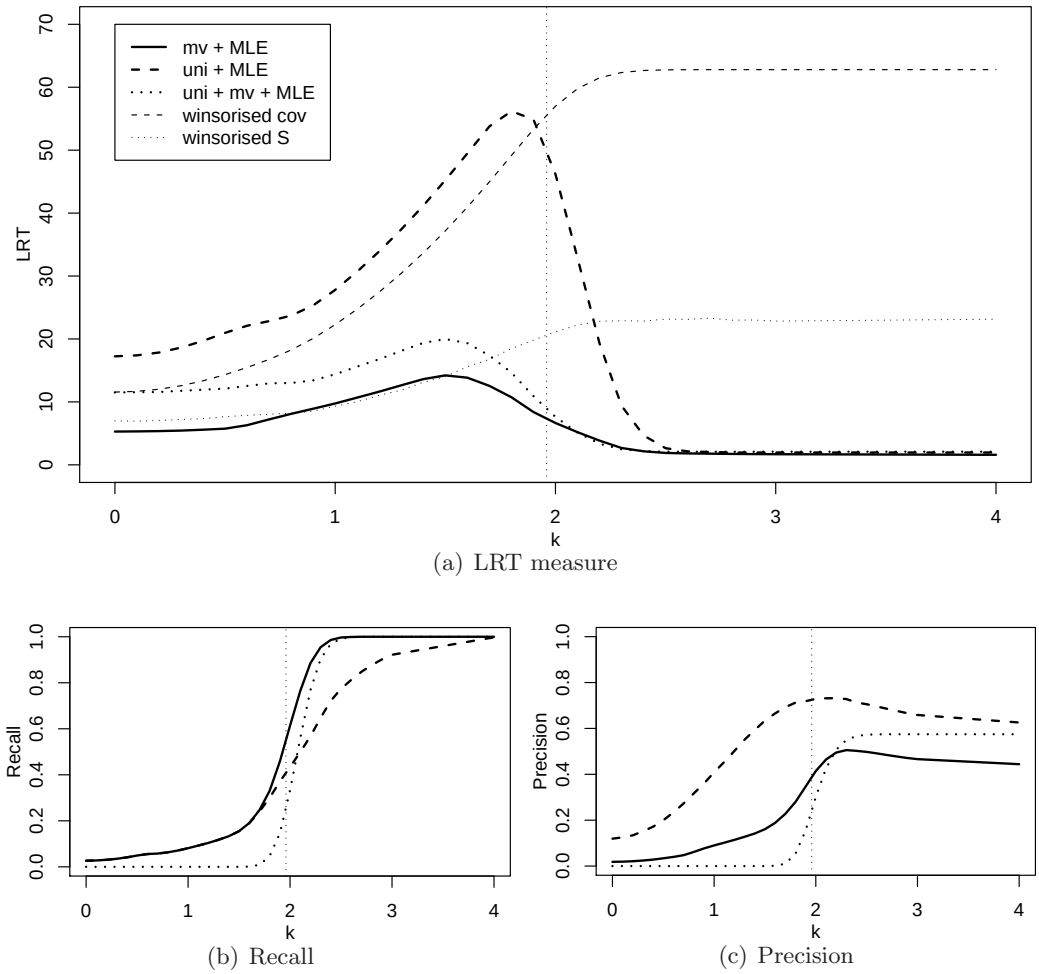
To demonstrate the advantages of multivariate outlier detection (P-approach) over univariate detection as well as over other more traditional approaches, we conduct a simulation study in the same manner as described in section 2.3.8.2. Now we will generate clean data with "moderate" correlation structure (as per section 2.1.4.3) because some degree of dependency is required to make multivariate approach beneficial. The results we are showing have been obtained with $p = 20$, moderate correlation, sample size $n = 200$, and proportion of contamination $\varepsilon = 0.05$. We have also tried running the simulations with several different dimensions, sample sizes as well as "high" correlation structure. Different numbers but similar trends can be observed with minor common-sense variations but we do not show them for the sake of brevity. This makes us believe that the behaviour reported in this section is typical for any elliptical data with non-independent correlation structure.

In this set of simulations we have used a lower univariate cutoff value (based on the 95th percentile as opposed to the 99.9th before) than in section 2.3.8.2 for the following reason. When variables are sufficiently correlated, as in our setup, univariate detection does not produce any significant improvement over multivariate detection. When used with a conservative large cutoff it affects only a negligible proportion of clean data and thus does not harm the performance of the multivariate-only estimate. Therefore a proper combination of multivariate detection with a large-cutoff univariate (the estimate we recommend) produces the same results as multivariate-only detection in this scenario. On the other hand, using a lower univariate cutoff (a) gives more chance to traditional Winsorised approaches; (b) allows us to demonstrate dangers of using a univariate cutoff which is too low.

Simulation results, namely the LRT-measure, precision and recall as functions of the contamination location k, are shown in Figure 2.7. In addition to the estimates showed in section 2.3.8.2 we also computed an S-estimate of the covariance matrix based on univariately Winsorised data (light dotted line). Most importantly, it can be seen that the multivariate detection (bold solid) yields the best results: uniformly (over k) lowest LRT-measure and highest recall and precision.

Both Winsorised estimates, based on sample covariance and S-estimate, suffer from one problem — the bias remains flat and high even after the contamination is easily detectable. For the Winsorised covariance the problem can be easily fixed by the univariate detection followed by the MLE estimate (bold dashed line). We will comment on how a similar approach can be used with S-estimate in Chapter 4.

An interesting phenomenon to observe is the difference between the multivariate estimate and its combination with univariate (bold dotted). Relatively high rate of false



(a) LRT measure

(b) Recall

(c) Precision

Figure 2.7. Performance of multivariate detection compared to alternative methods under "moderately" correlated data with $p = 20$. Vertical dotted line shows an aggressive (95th percentile) univariate cutoff at 1.94 used in these simulations. Multivariate detection was also done with the initial $p_{\text{cutoff}} = 0.05$.

positives in univariate detection (5% of good data) causes higher bias. This can also be seen by comparing Winsorised (regular dashed) and univariately filtered MLE (bold dashed) estimates. Winsorising preserves some information from the tails of good data and thus is a less harmful to the estimate than completely eliminating them. The negative effect of harsh univariate filtering is most pronounced for smaller, still undetected, values of k. Our interpretation is that reducing the amount of good data, is more damaging in the presence of outliers.

Additionally, the following estimates were computed but not graphed in order not to overload the figure.

- Sample covariance graph proceeds just under the Winsorised covariance for up to $k = 2$, at which point it continues to grow quadratically to infinity.

- Likewise, S-estimate of covariance follows the curve of the Winsorised S-estimate but continues to grow to infinity as k increases. In smaller dimensions, when overall fraction of contaminated cases does not exceed 50%, it continues growing beyond the univariate cutoff value but will eventually stabilize and not breakdown.

- Pairwise S-estimate corrected for positive-definiteness discussed in section 2.1.5.2 was also considered and we have observed that, as expected, it behaves quite poorly. It starts off at the average LRT-score of 52 when k is small and becomes increasingly bad as the contamination moves away from the centre (96 at $k = 4$). As the correction procedure uses linear combinations of variables, it is only natural that it will not perform well under independent contamination. To separate the poor performance of the estimate itself from the problems of the correction procedure, we have tried using the clean dataset (before contamination is introduced) to do the correction. The assisted "estimate" starts at the average LRT-score of 51 for small k and stabilizes at 40 for more obvious contamination (or when no contamination is present at all). This is still a very poor result compared to those shown in Figure 2.7.

We have also computed MLE estimates after detecting contamination using the *known covariance* structure of the data. The performance of univariate-detection estimate is comparable to that of its counterpart with estimated scales. The multivariate detection, however, can be very considerably improved if the true covariance structure is known. The average-LRT was between values 1 and 2 which would result in a horizontal line in the scale of Figure 2.7(a).

It must be noted that the apparent uniform superiority of the combined (multivariate with high-cutoff univariate) is only uniform over k but will vary depending on data configuration. As we have seen in Figure 2.6, when data does not have enough correlation structure to support multivariate detection, it results in higher false-positive rate than univariate and the overall estimate performance is slightly deteriorated. It nevertheless appears to be a reasonable compromise because the loss of performance in no-correlation case is relatively small but the benefits that can be gained by removing contaminated values to preserve existing structure are substantial.

Chapter 3

S-estimate of multivariate location and scatter in the presence of missing data

3.1 Introduction

Data of poor quality are very common in statistical applications. Robust statistical methods approach the problem of estimating distribution parameters when available data are not clean enough to satisfy strict distributional assumptions of classical methods. There are many reasons why data can be regarded as having poor quality, but the two most obvious and common are contamination and incompleteness.

Data contamination, often manifested by outliers, occurs when instead of observing the data from the population (or distribution) of interest, we observe a mixture of good data with data from some arbitrary erroneous distribution. In terms of distribution functions, the contamination model can be written as

$$F = (1 - \varepsilon)F_0 + \varepsilon G, \tag{3.1}$$

where F is the cdf of the observed data, F_0 is the cdf of the good part of the data that we are interested in estimating, G is an arbitrary contaminating distribution and finally $\varepsilon \in (0, 1)$ is the proportion of contaminated cases. In this chapter we consider *traditional contamination model* where the mixing happens on the level of data *cases* as opposed to data *cells* as considered in Chapter 2.

Missing data occur when some values in the data set cannot be observed for some reason. For instance, they may have been impossible to measure or have been lost during data manipulation. Unlike contamination, which affects whole observations, missingness affects individual components of data cases. Observations that are fully missing can be discarded from the beginning as they contain no useful information for estimating the distribution parameters. There are many mechanisms that can generate missing data but in this paper we will only deal with data which are Missing Completely at Random (MCAR). This assumption means that the probability of a particular value being missing does not depend on either observed or missing data.

There is a variety of robust methods to estimate the parameters of F_0 when the ob-

served data follow model (3.1). For instance, Minimum Covariance Determinants (MCD) (Rousseeuw, 1985) and more generally S-estimates (Davies, 1987) are common choices for the estimation of multivariate location and scatter parameters when F_0 is an elliptical distribution.

The problem of missing data has also been well studied during the past three decades. The two most widely accepted and used approaches are the Maximum Likelihood (MLE) of the observed data, typically computed with the EM-algorithm (Dempster et al., 1977), and the multiple imputation of the missing values (Rubin, 1987). Both methods, however, rely heavily on distributional assumptions about the data and therefore are not generally well equipped to handle contamination such as in (3.1).

Although the two problems with data seem to be fundamental, extremely common and well studied by themselves, little work has been done on how to address both of them at once. To estimate multivariate location and scatter robustly in the presence of missing data, Little and Smith (1987) proposed an ER-algorithm, a modified version of the EM-algorithm with the maximization step replaced by its weighted analog. The weights are based on the Mahalanobis distances — to the current center using the current covariance — of the observed part of each observation. They did a simulation study to show that the ER-algorithm is superior to the EM-algorithm in the presence of outliers but did not attempt to build any theory or investigate any properties of it. Further, Cheng and Victoria-Feser (2002) picked up the idea of Little and Smith (1987), modified the way the weights are applied to the data and the way they are computed (using Translated Biweight S-estimate (TBS) of Rocke (1996)). They notice that the estimates computed using the ER-algorithm can lose their robustness if the fraction of contamination exceeds $1/(p+1)$ and remedied it by using an MCD estimate on missing data — which they developed for the purpose — as a high-breakdown starting value. Finally, they brought the idea that the limit of the iterative procedure in the ER-algorithm can be seen as an S-estimate of multivariate location and scatter but did not define such estimator on missing data.

Multiple Imputation (Rubin (2004)) is a paradigm which is often used as an alternative to model-based approaches for dealing with challenges imposed by missing data. However we have not been able to locate any literature on the application of these ideas to robust estimators. For the reasons discussed below it appears to be a challenging problem that cannot be solved by simply combining existing procedures and thus requires more thorough development which is beyond the scope of this paper. In order to do *proper* multiple imputation (Wu, 2010, p.119) by averaging the predictive distribution for missing data over the posterior distribution of parameter (given the observed), one needs to have a robust way of computing the posterior, that is a robust Bayesian estimate, which is not available in the literature at the moment. In order to do *improper* (or *naive*) imputation one still needs a good estimate of the parameter to base the predictive distribution on. The estimate has to be robust and computable on partially missing data which leads to a circular dependence. A common approach to breaking such self-dependencies is to start with a simple (but robust) initial estimate and iteratively do (naive) imputation

and estimation until some convergence criterion is satisfied. In highly structured data, however, imputing missing values using imprecise estimates of the parameter will often lead to the generation of outliers. If the proportion of partially missing cases is high then even a robust estimator (computed on the imputed dataset) may not be able to deal with the many newborn outliers and obtain the next value of the parameter which is good enough to make progress.[2] In addition to conceptual challenges, multiple imputations is also computationally challenging because it requires computing slow robust estimates a large number of times.

In practice, some simple ad-hoc approaches are most commonly used to deal with missing data. One such approach, the *complete cases* analysis, is to remove all cases containing missing values and apply a traditional robust estimate to the remaining data. Naturally, it leads to loss of information and consequently to reduced statistical efficiency or even, when the proportion of missing data or the dimension is large, to the inability to compute the estimate at all.

In this paper we propose a robust method to estimate location and scatter parameters of an elliptical distribution when part of the data is missing. It is an extension of the S-estimate and employs ideas similar to the ML-of-the-observed-data approach. Unlike Little and Smith (1987) and Cheng and Victoria-Feser (2002), our approach derives from a proper definition of S-estimate on missing data. We can also show that our estimate is asymptotically unbiased while the previously defined estimates have intrinsic bias which does not go away as the sample size goes to infinity. In Section 3.3 we describe the theoretical generalization of the S-estimate to handle missing values. In Section 3.4 we explain how the estimate can be computed using a modification of the Fast-S algorithm (Salibian-Barrera and Yohai, 2006) and what important adjustments need to be made. The comparison of our estimate with the ER-algorithm and the ERTBS estimate is presented in Section 3.4.3. We run several simulations studies and describe some of the properties that the extended S-estimate possesses in Section 3.5. In particular, in Section 3.5.3 the estimate is applied to real non-normal data with artificially introduced missingness. We conclude that the new method compares favourably with all known alternatives.

3.2 Example

In order to illustrate the importance of having an estimation method that is both robust and can handle missing data, we consider a longitudinal dataset called *nursing* analysed

[2] We have tried this numerically on datasets with $p = 10$, $n = 100$, 10% of MCAR data, "high" correlation of the clean part of the data and 10% of strategically placed contamination. We have used two initial robust estimates: (a) diagonal matrix with univariate squared MAD estimates of variances; and (b) pairwise matrix of bivariate S-estimates. For both initial estimates, in a large proportion of the Monte Carlo sampled datasets, the procedure is unable to overcome the problem caused by the imputed outliers and yields estimates that are two orders of magnitude worse than can be achieved by the extended S-estimate proposed in this paper.

in Murphy et al. (1999). The dataset deals with the distress that parents experience after a violent death of their child. It contains measurements for 21 psychological characteristics (e.g. *BSI:Depression, Active Coping subscale of COPE*, etc.) taken at 5 time points after the death of the child. The data are available for 271 individuals. As it is common in longitudinal studies, observations at some time points for some people are not available, resulting in 23% of all data being missing.

To take advantage of these data, we assume that we are interested in estimating the covariance matrix of the whole dataset. It can be of interest on its own, for example, to explore the correlations between variables, or as a building block in other multivariate analysis such as PCA or multiple regression. For simplicity, we restrict our attention to 17 out of 21 characteristics[3] and only consider three measurements (first, middle and the last) for each of them. This gives us a sub-dataset with 51 variables, 267 cases and 18.6% missing data.

When presented with such a dataset, one may take several approaches to analyze it. One is to compute **Maximum Likelihood** estimates (MLE) of the location and covariance matrix using, for example, the EM algorithm. When suspecting that outliers might be present in the data, another possible approach would be to remove all cases containing missing values and compute a robust estimate for the covariance matrix based on **complete cases** only. Alternatively, one may try use the **pairwise** approach: robustly estimate each element of the covariance matrix based on all observations that do not have missing values in the corresponding pair of coordinates. Optionally, if it is important to have a positive-definite scatter estimate, it can be corrected using a version[4] of the data-dependent method described in Maronna and Zamar (2002). In this section, we compute all the above and also our new extended S-estimate on the *nursing* dataset, and show that some new features can be found in the data that could not be seen with the three conventional methods. We will also speculate up to why other estimates perform poorly in this case.

As we mentioned before, the dimensionality of these estimates is fairly high so we will not show the actual estimates here but will rather explore what conclusions can be made based on them. In order to identify outliers, we want to analyze Mahalanobis distances for all 267 observations. In order to do that, we need to extend the definition of Mahalanobis distance to accommodate the possibility of having missing values in the data. We do this in the spirit of the method that we are proposing: for each case, compute Mahalanobis distance for the observed components and scale it up according to the dimension. The scaling is necessary to make distances from different dimensions comparable to each other. If data were multivariate normal the squared partial Mahalanobis distances would be

[3]We focused on 51 variables in total partly due to the limitations of the computer code for ERTBS estimate that was available to us.

[4]The procedure of Maronna and Zamar (2002) relies on data without missing values as it makes use of robust scales of linear combinations of variables. To overcome this limitation we take the easy route and simply impute variable-wise medians when a missing value is involved in such a linear combination.

distributed as χ^2 with as many degrees of freedom as there are observed components. We use this property to convert observed distances to the Uniform$[0,1]$ distribution and then transform the scores back to some standardized distribution that we choose to be χ^2_p, the distribution of distances for the complete cases. Corrected Mahalanobis distances are

$$\mathrm{MD}^*(\boldsymbol{x};\boldsymbol{m},\boldsymbol{\Sigma}) = q\left(\mathrm{MD}\left(\boldsymbol{x}^{\mathrm{obs}};\boldsymbol{m}_{\mathrm{obs}(\boldsymbol{x})},\boldsymbol{\Sigma}_{\mathrm{obs}(\boldsymbol{x}),\mathrm{obs}(\boldsymbol{x})}\right);|\mathrm{obs}(\boldsymbol{x})|,p\right),$$

where

$$\mathrm{MD}(\boldsymbol{v};\boldsymbol{m},\boldsymbol{\Sigma}) = \sqrt{(\boldsymbol{v}-\boldsymbol{m})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{v}-\boldsymbol{m})}$$

denotes the usual Mahalanobis distance of vector $\boldsymbol{v}$ to center $\boldsymbol{m}$ using covariance matrix $\boldsymbol{\Sigma}$,

$$q(y;p_1,p_2) = \sqrt{F^{-1}_{\chi^2_{p_2}}\left(F_{\chi^2_{p_1}}(y^2)\right)} \tag{3.2}$$

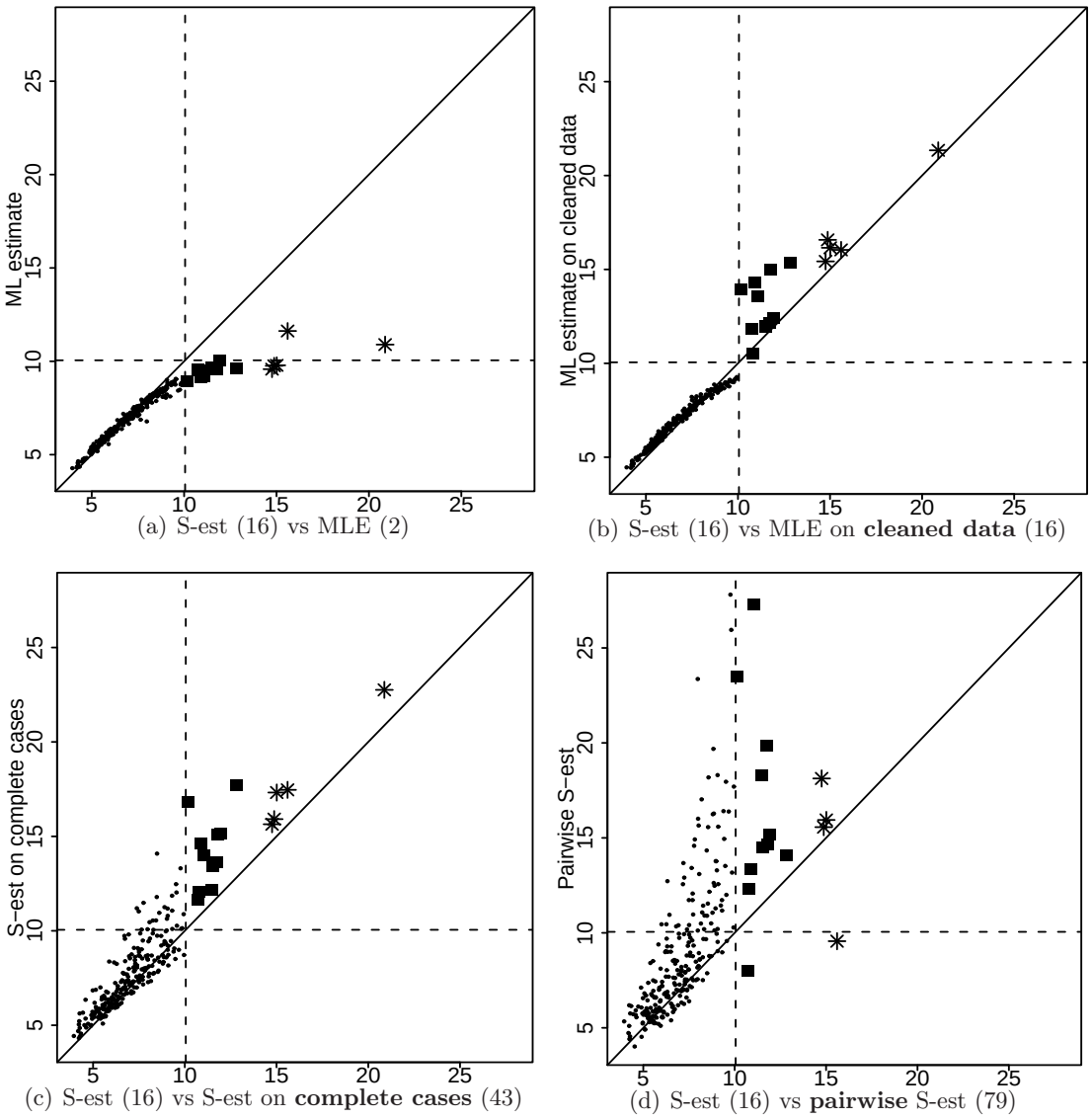
is the scaling function based on the quantiles of χ^2 distribution, and $\mathrm{obs}(\boldsymbol{x})$ denotes the set of indices of $\boldsymbol{x}$ corresponding to the non-missing components. This way the corrected Mahalanobis distances will follow the χ^2_p distribution assuming that the original data are drawn from the p-variate normal distribution with mean $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$.

In Figure 3.1 we display (corrected) Mahalanobis distances computed using several location and covariance estimates. They are shown as scatter plots against the Mahalanobis distances computed using the new proposed S-estimate in order to illustrate the differences between the estimates. The dashed lines show our outlier cutoff points. They are chosen such that, under the multivariate normal assumption, there is only 1% chance of the maximum Mahalanobis distance (out of $n = 267$) exceeding it. More specifically, c^* is such that

$$\mathbf{P}\{\max_{k=1,\ldots,n} X_k > (c^*)^2\} = 0.01, \text{ where } X_k \overset{\text{iid}}{\sim} \chi^2_p, \text{ with } n = 267 \text{ and } p = 51. \tag{3.3}$$

Looking at Figure 3.1(a), we can see that the extended S-estimate (horizontal axis) allows us to identify 5 obvious outliers (marked with asterisks) and another 11 points that exceed the formal cutoff value (marked with solid squares). The **ML estimate** computed using the EM-algorithm (vertical axis), however, identifies only 2 points that are above the cutoff and separated from the rest of the data. This suggests that masking of outliers is taking place in this dataset and that a robust method, such as the extended S-estimate, is necessary to unveil them. In Figure 3.1(b) we removed the 16 outliers (identified with the S-estimate) from the dataset and recomputed the ML estimate using the remaining 252 cases. After that we used the **"cleaned" ML estimate** to compute Mahalanobis distances for all 267 cases. Now we can see that they agree well with the robust Mahalanobis distances, which confirms our fears about masking of outliers and reinforces our belief of the importance of robust estimation in the case of this dataset.

The S-estimate based on **complete cases** (132 in this dataset), shown in Figure 3.1(c),



(a) S-est (16) vs MLE (2)

(b) S-est (16) vs MLE on **cleaned data** (16)

(c) S-est (16) vs S-est on **complete cases** (43)

(d) S-est (16) vs **pairwise** S-est (79)

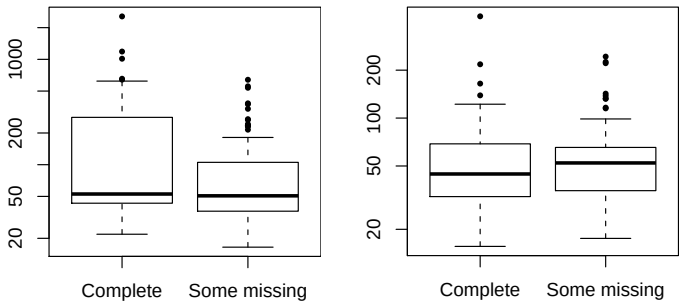
Figure 3.1. Mahalanobis distances for the nursing dataset. The dashed lines are the outlier cutoff points at $c^* \approx 10.05$. Numbers in parentheses indicate the number of outliers identified using the corresponding estimate.

exhibit a different problem: it identifies too many (43 in this case) outliers. It is well known that when the sample size is relatively small and the dimension of the data is large, estimates of the covariance matrix tend to become more singular (see, for example, Goodman (1963)). This is true for regular sample covariance matrices and even more so for robust estimates. When this happens, the covariance matrix describes the data as being mostly on a hyperplane in the original $\mathbb{R}^p$ space. All points that do not belong to that hyperplane will thus have large Mahalanobis distances w.r.t. that covariance matrix and will be labeled as outliers. Based on only half of the available data, the *complete-cases* estimate lacks efficiency which contributes to misclassifying some points as outliers. In this

example, we have $\det(\mathbf{\Sigma}_{CC})^{1/p} = 0.45$ which is noticeably smaller than $\det(\mathbf{\Sigma}_S)^{1/p} = 0.50$, where $\mathbf{\Sigma}_{CC}$ is the complete cases estimate and $\mathbf{\Sigma}_S$ is the S-estimate. It should also be noted that if the proportion of missing values was even higher, then it would be likely that the number of complete cases would become smaller than the dimension of the data and then no estimate could be computed at all.

Finally, the plot in Figure 3.1(d) demonstrates the poor performance of the **pairwise S-estimate**. It highlights even more (79 to be exact) outliers than the complete-cases estimate. Also, the cloud of points in the scatter plot becomes very blurred which means that the order (or degree of outlierness) is barely preserved when compared with the one suggested by the extended S-estimate. This is happening because the pairwise estimate fails to capture any data structures of dimension more than two. Therefore, Mahalanobis distances — which rely on the fine structure of multivariate data — become very erratic.

To complete the list of possible methods to analyze this dataset we have also computed the ERTBS estimate by Copt and Victoria-Feser (2003). We have had to modify the computer code provided by the authors as the dimension of the data exceeds their upper limit of 50 variables. The method highlights 79 possible outliers. A closer look at the "outliers", however, reveals that most of them have none or only few components missing. To quantify this statement we have computed the average squared Mahalanobis distance for the 135 complete cases and compared it with the average dimension-corrected Mahalanobis distance for the 132 partly missing observations. See boxplots in Figure 3.2(a) comparing



(a) ERTBS: average 184.4 vs 89.6 (b) Ext S-est: average 56.0 vs 57.9

Figure 3.2. Estimated squared Mahalanobis distances for complete vs incomplete cases based on ERTBS and extended-S estimates. The boxplots are on the log scale.

squared Mahalanobis distances (using ERTBS estimate) between the two groups of observations. The distributions are obviously not identical. While realizing that one can only speculate on the nature of this phenomenon we attribute it to the fact that the ERTBS method more severely penalizes fully observed cases. On the other hand, our extended S-estimate does not suffer from this phenomenon and its Mahalanobis distances are independent from the number of observed variables as evidenced by nearly identical boxplots

in Figure 3.2(b).

Overall, we can say that the proposed extended S-estimate allows us to identify outliers in the data that would otherwise be hard or impossible to identify using currently available methods. Those 16 data points may be of interest on their own or can just be of help in getting a better estimate for the covariance structure of the rest of the data.

In the next section we show how to modify the definition of S-estimates so that they can handle missing data.

3.3 Adapting robust estimates to missing values

3.3.1 Motivation

Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be an i.i.d. sample of a $p \times 1$ random vector $\boldsymbol{X}$ following the contamination model (3.1), where F_0 is an elliptical distribution with location and scale parameters $(\boldsymbol{m}, \boldsymbol{\Sigma})$. Suppose that the sample we observe is incomplete and assume here and in the rest of the paper that the data are missing completely at random (MCAR). Our goal is to estimate the parameters $(\boldsymbol{m}, \boldsymbol{\Sigma})$ of the "core" distribution F_0.

If the data were fully observed then the S-estimate of multivariate location and scatter would be a possible approach. On the other hand, if the data come from the distribution F_0, instead of the mixture F, then the MLE on the observed data would be appropriate. Later in this section we show how to combine these two approaches to solve the problem at hand.

Let $L(\boldsymbol{m}, \boldsymbol{\Sigma}; \boldsymbol{x})$ be the likelihood function for a single point. For an i.i.d. sample X the likelihood function $L^{\text{cmpl}}(\boldsymbol{m}, \boldsymbol{\Sigma}; X)$ is simply a product of the likelihoods for the individual cases:

$$L^{\text{cmpl}}(\boldsymbol{m}, \boldsymbol{\Sigma}; X) = \prod_{i=1}^{n} L(\boldsymbol{m}, \boldsymbol{\Sigma}; \boldsymbol{x}_i). \tag{3.4}$$

In order to get the likelihood function for the observed data when some values are missing, one can integrate (3.4) over the unobserved values

$$L^{\text{obs}}(\boldsymbol{m}, \boldsymbol{\Sigma}; X) = \int L^{\text{cmpl}}(\boldsymbol{m}, \boldsymbol{\Sigma}; X^{\text{obs}}, X^{\text{mis}}) \mathrm{d}X^{\text{mis}} = \prod_{i=1}^{n} \left[\int L(\boldsymbol{m}, \boldsymbol{\Sigma}; \boldsymbol{x}_i^{\text{obs}}, \boldsymbol{x}_i^{\text{mis}}) \mathrm{d}\boldsymbol{x}_i^{\text{mis}} \right] = \prod_{i=1}^{n} \tilde{L}_i(\boldsymbol{m}, \boldsymbol{\Sigma}; \boldsymbol{x}_i^{\text{obs}}), \tag{3.5}$$

where $\tilde{L}(\cdot; \boldsymbol{x}_i^{\text{obs}})$ is the part of the likelihood function that only depends on the observed components of $\boldsymbol{x}_i$. In case of elliptical distributions, when the p.d.f. of $\boldsymbol{X}$ can be expressed as

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} h((\boldsymbol{x} - \boldsymbol{m})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{m})),$$

it can be shown(Lemma 2 in the Appendix B.1) that the likelihood of the observed part

of $\boldsymbol{X}$ takes the following form

$$\tilde{L}_i(\boldsymbol{m}, \boldsymbol{\Sigma}; \boldsymbol{x}_i^{\text{obs}}) = \det\big(\boldsymbol{\Sigma}_{[i]}\big)^{-\frac{1}{2}} \tilde{h}_{p_i}\Big((\boldsymbol{x}_i^{\text{obs}} - \boldsymbol{m}_{[i]})'\boldsymbol{\Sigma}_{[i]}^{-1}(\boldsymbol{x}_i^{\text{obs}} - \boldsymbol{m}_{[i]})\Big), \tag{3.6}$$

where $\boldsymbol{m}_{[i]}$ is a $p_i \times 1$ sub-vector of $\boldsymbol{m}$ with components corresponding to the observed components of $\boldsymbol{x}_i$ and $\boldsymbol{\Sigma}_{[i]}$ is a $p_i \times p_i$ sub-matrix of $\boldsymbol{\Sigma}$ with rows and columns corresponding to the observed components of $\boldsymbol{x}_i$. The functions $\tilde{h}_{p_i}(\cdot)$ depend on the function $h(\cdot)$ and on the number of observed components in $\boldsymbol{x}_i$ but they do not depend on the parameters $\boldsymbol{m}$ or $\boldsymbol{\Sigma}$. For each case, the argument of $\tilde{h}_{p_i}(\cdot)$ is the squared Mahalanobis distance of the observed components of $\boldsymbol{x}_i$. As we usually maximize the log-likelihood function instead of the likelihood itself, let us take note of the logarithm of (3.5) taking (3.6) into account:

$$\tilde{\ell}^{\text{obs}}(\boldsymbol{m}, \boldsymbol{\Sigma}; X) = -\frac{1}{2}\sum_{i=1}^{n} \log\det\big(\boldsymbol{\Sigma}_{[i]}\big) + \sum_{i=1}^{n} \log \tilde{h}_{p_i}\Big((\boldsymbol{x}_i^{\text{obs}} - \boldsymbol{m}_{[i]})'\boldsymbol{\Sigma}_{[i]}^{-1}(\boldsymbol{x}_i^{\text{obs}} - \boldsymbol{m}_{[i]})\Big). \tag{3.7}$$

In the classical analysis of missing data this expression is usually optimized using the EM-algorithm of Dempster et al. (1977).

Note that the ML approach makes no allegations regarding the missing values other than assuming that they are missing completely at random. In particular, it is not trying to impute or predict them. It simply takes into account the observed data as if we were not even planning on collecting the missing part. Moreover, the observed data is summarized with only their Mahalanobis distances w.r.t. the appropriate sub-vectors and the sub-matrices of the estimation parameters. We borrow this idea from the ML paradigm and carry it forward to the S-estimates.

S-estimates of multivariate location and scatter on complete data were defined by Davies (1987) as $(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\Sigma}})$ that minimize $\det(\boldsymbol{\Sigma})$ subject to

$$\frac{1}{n}\sum_{i=1}^{n} \rho_c\big(\text{MD}^2(\boldsymbol{x}_i; \boldsymbol{m}, \boldsymbol{\Sigma})\big) = b, \tag{3.8}$$

where $\rho_c(d) = \rho(d/c)$ and $\rho : \mathbb{R}_+ \to \mathbb{R}_+$ is a non-decreasing continuous function such that $\rho(0) = 0$ and $\rho(\infty) = 1$. The constant b needs to be in the interval $(0, 1)$ in order for (3.8) to have a solution. To simplify the numerical solution of this optimization problem, we can equivalently reformulate it as follows. First, note that any scatter matrix $\boldsymbol{\Sigma}$ can be represented as $\boldsymbol{\Sigma} = \sigma(\boldsymbol{\Sigma})\boldsymbol{\Sigma}_0$, where $\det(\boldsymbol{\Sigma}_0) = 1$ and $\sigma(\boldsymbol{\Sigma}) = \det(\boldsymbol{\Sigma})^{1/p}$. Then the Mahalanobis distance in (3.8) becomes

$$\text{MD}^2(\boldsymbol{x}_i; \boldsymbol{m}, \boldsymbol{\Sigma}) = (\boldsymbol{x}_i - \boldsymbol{m})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}) = (\boldsymbol{x}_i - \boldsymbol{m})'\frac{\boldsymbol{\Sigma}_0^{-1}}{\sigma(\boldsymbol{\Sigma})}(\boldsymbol{x}_i - \boldsymbol{m}) = \frac{\text{MD}^2(\boldsymbol{x}_i; \boldsymbol{m}, \boldsymbol{\Sigma}_0)}{\sigma(\boldsymbol{\Sigma})}, \tag{3.9}$$

and $\det(\boldsymbol{\Sigma}) = \sigma(\boldsymbol{\Sigma})^p$, which is a monotone increasing function of $\sigma(\boldsymbol{\Sigma})$. For any vector $\boldsymbol{m}$ and matrix $\boldsymbol{\Sigma}_0$ with $\det(\boldsymbol{\Sigma}_0) = 1$, it is possible to find a scalar $s = s(\boldsymbol{m}, \boldsymbol{\Sigma}_0)$, such that

$(\boldsymbol{m}, s\boldsymbol{\Sigma}_0)$ satisfies (3.8), or, in other words,

$$\frac{1}{n}\sum_{i=1}^{n}\rho_c\left(\frac{\mathrm{MD}^2(\boldsymbol{x}_i;\boldsymbol{m},\boldsymbol{\Sigma}_0)}{s}\right)=b. \tag{3.10}$$

The unique solution exists because the left-hand-side is a continuous monotonely non-increasing function of s that is equal to $\rho_c(\infty)$ at $s=0$, and 0 at $s=+\infty$. Note that $\sigma(\boldsymbol{\Sigma})$ and $s(\boldsymbol{m},\boldsymbol{\Sigma}_0)$ are essentially the same thing in a sense that $\sigma(\boldsymbol{\Sigma})=s(\boldsymbol{m},\det(\boldsymbol{\Sigma})^{-1/p}\,\boldsymbol{\Sigma})$, for any $(\boldsymbol{m},\boldsymbol{\Sigma})$ satisfying (3.8). Therefore, minimizing the $\det(\boldsymbol{\Sigma})$ over all $(\boldsymbol{m},\boldsymbol{\Sigma})$ satisfying (3.8) is equivalent to minimizing $s(\boldsymbol{m},\boldsymbol{\Sigma}_0)$, that solves (3.10), over all positive-definite matrices with

$$\det(\boldsymbol{\Sigma}_0)=1. \tag{3.11}$$

Throughout this paper we will use Tukey's bisquare[5] ρ-function of the form

$$\rho_c(d)=\rho(d/c)=\min\{1-(1-(d/c))^3,1\},\ \text{for any } d,c\geq 0, \tag{3.12}$$

but any other ρ-function can be used with minimal modifications. Rocke's biflat ρ-function appears to be a promising alternative for high-dimensional data but we have chosen to restrict our attention to Tukey's bisquare function in this thesis. Refer to Appendix B.2 for further discussion concerning this choice.

The constant b in (3.10) controls the breakdown properties of the estimator, such that the breakdown point is equal to $\min\{b,1-b\}$. The constants c and b are usually chosen jointly so that the estimator is Fisher-consistent when the data follow uncontaminated normal distribution. This can be achieved by choosing c and b such that

$$\mathbb{E}\big[\rho_c\big(\mathrm{MD}^2(\boldsymbol{X};\boldsymbol{0}_p,\boldsymbol{I}_{p\times p})\big)\big]=b,\ \text{when } \boldsymbol{X}\sim\mathcal{N}_p(\boldsymbol{0},\boldsymbol{I}). \tag{3.13}$$

As $\rho_c(d)$ is a non-increasing function of c, it is easy to see that larger values of b correspond to smaller values of c and vice versa. In the limit, if $b\to 0$ then $c\to\infty$ and the S-estimate will become equivalent to the Gaussian MLE.

3.3.2 Definition

For the case when observations $\boldsymbol{x}_i$ are partly missing, we propose the following extended S-estimate. Later we will discuss the required changes and our motivation for making them.

Definition 1. *The extended S-estimate of multivariate location and scatter is defined as* $(\hat{\boldsymbol{m}},\hat{s}\hat{\boldsymbol{\Sigma}})$ *where* $\hat{s}=s(\hat{\boldsymbol{m}},\hat{\boldsymbol{\Sigma}})$, *and* $(\hat{\boldsymbol{m}},\hat{\boldsymbol{\Sigma}})$ *minimizes the scale* $s(\boldsymbol{m},\boldsymbol{\Sigma})$, *which is the solution* s *of*

$$\frac{1}{n}\sum_{i=1}^{n}\tilde{\rho}_i\left(\frac{\mathrm{MD}_i^2(\boldsymbol{x}_i^{\mathrm{obs}};\boldsymbol{m},\boldsymbol{\Sigma})}{s}\right)=b, \tag{3.14}$$

[5]The name is due to the corresponding ψ-function which is the *derivative* of $\rho(\cdot)$.

subject to

$$\sum_{i=1}^{n} \log\big(\det\big(\boldsymbol{\Sigma}_{[i]}\big)\big) = 0, \tag{3.15}$$

where

$$\mathrm{MD}_i^2(\boldsymbol{x}_i; \boldsymbol{m}, \boldsymbol{\Sigma}) = \mathrm{MD}^2(\boldsymbol{x}_i^{\mathrm{obs}}; \boldsymbol{m}_{[i]}, \boldsymbol{\Sigma}_{[i]}) = (\boldsymbol{x}_i^{\mathrm{obs}} - \boldsymbol{m}_{[i]})' \boldsymbol{\Sigma}_{[i]}^{-1} (\boldsymbol{x}_i^{\mathrm{obs}} - \boldsymbol{m}_{[i]}), \tag{3.16}$$

are the Mahalanobis distances of the observed data, and

$$\tilde{\rho}_i(d) = \frac{c_i k_i}{n^{-1} \sum c_j k_j} \rho_{c_i}(d), \tag{3.17}$$

with constants c_i and k_i such that

$$\mathbb{E}\big[\rho_{c_i}\big(X_1^2 + \cdots + X_{p_i}^2\big)\big] = b, \text{ and} \tag{3.18}$$

$$k_i = \big\{\mathbb{E}\big[c_i \rho'_{c_i}\big(X_1^2 + \cdots + X_{p_i}^2\big) X_1^2\big]\big\}^{-1}. \tag{3.19}$$

Here p_i is the number of observed components of $\boldsymbol{x}_i$, and $X_j \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$, for $j = 1, \ldots, p$. Note that c_i and k_i only depend on i through p_i.

As it is easy to see, the modifications are threefold: the argument of the ρ-function, the scaling of it and the optimization constraint on $\boldsymbol{\Sigma}$.

Inside the ρ-function we now have the Mahalanobis distance of the observed part of each data point. The distance is computed to the corresponding subvector of the location vector $\boldsymbol{m}$ and is normalized with the submatrix $\boldsymbol{\Sigma}_{[i]}$ of the overall covariance matrix $\boldsymbol{\Sigma}$. This is analogous to what we saw in the likelihood expression for the elliptical distributions in equations (3.6) and (3.7).

The role of c in $\rho_c(\cdot)$ is to decide which points are outliers based on their squared Mahalanobis distance to the center. If the distance (relative to c) is small then the point is given a large weight, but if $\mathrm{MD}^2 \geq c$ then the point is treated as outlier. To recognize the fact that Mahalanobis distances in larger dimensions tend to be larger, we use $\rho_{c_i}(d) = \rho(d/c_i)$ in (3.17). By (3.18), the constants c_i tend to be bigger in higher dimensions and so the Mahalanobis distances are scaled appropriately.

The three scaling factors in front of the $\rho_{c_i}(d)$ in (3.17) all have their respective purposes.

1. c_i ensures that the extended S-estimate reduces to the Gaussian MLE of the observed data when the constant b, responsible for the breakdown point, is chosen to be small. As $b \to 0$, all $c_i \to \infty$ and therefore $\rho'_{c_i}(d) = 3/c_i + o(d/c_i^2)$, so that all $k_i \to \big\{\mathbb{E}\big[3 \times X_1^2\big]\big\}^{-1} = 1/3$. Then the ρ-functions become

$$\tilde{\rho}_i(d) \approx \frac{c_i}{n^{-1} \sum c_j} \rho_{c_i}(d) \approx \frac{c_i}{n^{-1} \sum c_j} \frac{d}{c_i} = \frac{n}{\sum c_j} d,$$

because $\rho_{c_i}(t) = t/c_i + o(c_i^{-1})$, when $c_i \to \infty$. So the optimization problem (3.14) turns into minimizing

$$s(\boldsymbol{m}, \boldsymbol{\Sigma}) = \frac{n}{b \sum c_j} \sum_{i=1}^{n} \mathrm{MD}_i^2(\boldsymbol{x}_i; \boldsymbol{m}, \boldsymbol{\Sigma}),$$

under the constraint $\sum_{i=1}^{n} \log\left(\det\left(\boldsymbol{\Sigma}_{[i]}\right)\right) = 0$. Which, in turn, is equivalent to maximizing the Gaussian likelihood of the observed data (see Lemma 3 in Appendix B.3).

2. k_i is necessary for the estimate to be Fisher-consistent. It guarantees that data of varying dimensions are combined properly. See Section 3.3.3 and the proof of Theorem 1 in Appendix B.4 for the discussion of the Fisher-consistency of the estimate and the role of k_i.

3. Finally, the normalizing average of $c_j k_j$ over $j = 1, \ldots, n$ in the denominator of (3.17) guarantees that the expectation of the left hand side of (3.14) under independent Gaussian data is equal to its right hand side (i.e. to b). Namely

$$\mathbb{E}\left[n^{-1} \sum_{i=1}^{n} \tilde{\rho}_i \left(\mathrm{MD}^2(\boldsymbol{X}_i; \boldsymbol{0}_{p_i}, \boldsymbol{I}_{p_i \times p_i}) \right) \right] = n^{-1} \sum_{i=1}^{n} \frac{c_i k_j}{n^{-1} \sum c_j k_j} b = b. \tag{3.20}$$

This is also a necessary condition for the estimate to be Fisher-consistent.

The modified constraint in (3.15) reflects the changes in the determinant part of the expression (3.7) as compared with the usual log-likelihood function of complete data. Instead of fixing the determinant of the scatter matrix, we constraint the average of the log of the determinants of the submatrices of $\boldsymbol{\Sigma}$. As we have seen above, this modified constraint is necessary for the extended S-estimate to be a generalization of the Gaussian MLE when $b \approx 0$.

Note that this new constraint is still about the scale of $\boldsymbol{\Sigma}$. Namely, for any positive-definite matrix $\boldsymbol{\Sigma}$ there exists a matrix $a\boldsymbol{\Sigma}$ satisfying (3.15). It is easy to find such $a > 0$ once we notice that

$$\begin{aligned} \sum_{i=1}^{n} \log\left(\det\left(a\boldsymbol{\Sigma}_{[i]}\right)\right) &= \sum_{i=1}^{n} \log\left(a^{p_i} \det\left(\boldsymbol{\Sigma}_{[i]}\right)\right) \\ &= \sum_{i=1}^{n} \left[p_i \log a + \log\left(\det\left(\boldsymbol{\Sigma}_{[i]}\right)\right) \right] = \log a \sum_{i=1}^{n} p_i + \sum_{i=1}^{n} \log\left(\det\left(\boldsymbol{\Sigma}_{[i]}\right)\right), \end{aligned} \tag{3.21}$$

and therefore the desired

$$a = \exp\left(- \frac{\sum_{i=1}^{n} \log\left(\det\left(\boldsymbol{\Sigma}_{[i]}\right)\right)}{\sum_{i=1}^{n} p_i} \right). \tag{3.22}$$

One more important property of the extended S-estimate is that it is a generaliza-

tion of the usual S-estimate and reduces to the latter when no missing data are present. Equation (3.14), having all Mahalanobis distances computed using the full covariance, reduces to (3.8). In constraint (3.15) every term of the summation is equal to $\log\det(\boldsymbol{\Sigma})$ and therefore the constraint reduces to $\log\det(\boldsymbol{\Sigma}) = 0$ which is equivalent to $\det(\boldsymbol{\Sigma}) = 1$ required by (3.11). Constants $k_{i_1} = k_{i_2}$ and $c_{i_1} = c_{i_2} = c$, for all $1 \leq i_1, i_2 \leq n$, because they only depend on the number of observed components which is now equal to p for all cases. Therefore function $\tilde{\rho}_i$ defined in (3.17) is equal to $\rho_{c_i} = \rho_c$.

3.3.3 Fisher-consistency

As mentioned above, an effort has been made to ensure that the estimate is asymptotically unbiased. The agreement between the constants c_i and b ensures that the scale is estimated correctly. The constants k_i are chosen so that the covariance submatrices of different dimensions are all on the same scale and can be combined to form an asymptotically unbiased estimate of the covariance structure. We have the following result:

Theorem 1. *The extended S-estimate is locally Fisher-consistent under multivariate normal data even in the presence of missing data.*

Local Fisher-consistency here means that the true values of $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$ locally minimize the functional value of $s(\boldsymbol{m}, \boldsymbol{\Sigma})$. See Appendix B.4 for the definition of the asymptotic functional $s(\boldsymbol{m}, \boldsymbol{\Sigma})$ and for the proof of this theorem. Simulation study in section 3.5.2 indicates that our estimate is in fact consistent and asymptotically normal at the usual $\sqrt{n}$-rate.

Multivariate normality is not a restrictive assumption here. If data are believed to follow another family of elliptical distributions then the Definition 1 can be modified so that the estimate is still Fisher consistent. To achieve that, the expectations in (3.18) and (3.19) should be computed with respect to the distribution of choice instead of the standard normal. See proof of Theorem 1 in Appendix B.4 for more details.

3.3.4 Scale and location equivariance

Affine equivariance is often viewed as a desirable property for statistical estimates. When no missing data are present, an affine equivariant location-scatter estimate is such that

$$\hat{\boldsymbol{\Sigma}}(Y) = \boldsymbol{A}\hat{\boldsymbol{\Sigma}}(X)\boldsymbol{A}', \text{ and } \hat{\boldsymbol{m}}(Y) = \boldsymbol{A}\hat{\boldsymbol{m}}(X) + \boldsymbol{b}, \tag{3.23}$$

where $Y = X\boldsymbol{A} + \boldsymbol{b}$ is an affinely transformed version of the data matrix X, with a non-singular square matrix $\boldsymbol{A}$. Note, that it is crucial for the equivariance property that the matrix A is invertible.

In order to define affine transformations on data with missing values, the arithmetic needs to be expanded a bit. We will assume that any arithmetic operation applied to a

missing value results in a missing value, unless it was a multiplication by zero, in which case the result is also a zero. Under these rules, only a limited number of very specific affine transformations can be inverted so that the notion of affine equivariance can even be applied. In particular, any transformation that attempts to "mix" missing values with observed data is not invertible and therefore the concept of affine equivariance is not applicable. A small class of invertible affine transformations that can always be applied to partly missing data consists of location shifts and changes of scale of individual variables. In terms of equation (3.23), it means that $\boldsymbol{A}$ is a diagonal matrix. We will call them *location and scale* transformations. For some datasets where not every variable has missing values, some other matrices $\boldsymbol{A}$ may result in an invertible transformation, but we will not study them because they depend heavily on the missingness pattern of a particular dataset.

Theorem 2. *The extended S-estimate is equivariant under location and scale transformations.*

Proof. See Appendix B.5. □

Definition 1 describes the extended S-estimate $(\hat{\boldsymbol{m}}, \hat{s}\hat{\boldsymbol{\Sigma}})$ but provides little, if any, insight into how to compute it. In Section 3.4 we propose a set of modifications to the Fast-S algorithm, originally described by Salibian-Barrera and Yohai (2006), that allows us to compute the extended S-estimate on partly missing data.

3.4 Computational aspects of S-estimate with missing values

3.4.1 Modified Fast-S algorithm

In Section 3.3 we described how the S-estimate can be defined in the presence of missing data. But the optimization problem in (3.14), even for the usual S-estimate as in (3.10), is complex. The estimates $\hat{\boldsymbol{m}}$ and $\hat{\boldsymbol{\Sigma}}$ optimize the scale $s(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\Sigma}})$ which is the solution to a non-linear data-dependent equation, $\hat{\boldsymbol{\Sigma}}$ is generally highly multi-dimensional and, above all, it is constrained to being positive-definite and to satisfy (3.15).

One possible solution, in the case of the usual S-estimate, is to employ the approximate subsampling- and reweighting-based Fast-S algorithm of Salibian-Barrera and Yohai (2006). They describe the algorithm for regression S-estimates but the adaptation to multivariate location and scatter estimates is straightforward. In this section, we briefly mention the basic algorithm but the focus is on the changes that need to be made in order to accommodate missing values and compute the estimate as defined by (3.14) and (3.15).

Algorithms 1 and 2 — the latter being a subroutine called by the former — describe the extended Fast-S algorithm for partly missing data. The general structure is similar to that of the usual Fast-S algorithm. The lines that require modification are marked

with asterisks on the right margin and further explained in this section. To implement the algorithm in `R`, we modified the code for Fast-S estimate of multivariate location and scatter published by Joossens and Roelant (2007).

Algorithm 1 Fast-S algorithm

1: initialize $s_t^* = \infty$, for $t = 1, \ldots, N_b$(small) — the list of best (smallest) scales
2: **for** $j = 1$ to N_s(large) **do**
3: take a random subsample $X_{(j)}$ of size n_s(small) *
4: compute its average $\boldsymbol{m}_0 = m_0(X_{(j)})$ and covariance $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_0(X_{(j)})$ *
5: rescale $\boldsymbol{\Sigma}_0 := a\boldsymbol{\Sigma}_0$, such that constraint (3.15) holds *
6: iterate *small number r of times*: $(\boldsymbol{m}_{\mathrm{w}}, \boldsymbol{\Sigma}_{\mathrm{w}}) = \texttt{iter.rew}\,(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0; r)$ (see Algorithm 2)
7: compute Mahalanobis distances $d_i = d_i(\boldsymbol{x}_i; \boldsymbol{m}_{\mathrm{w}}, \boldsymbol{\Sigma}_{\mathrm{w}})$, for all i (same as line 6 in Alg 2)
8: find scale $s_{(j)}$ solving equation (3.14) (same as line 7 in Alg 2)
9: **if** $s_{(j)} < s^*_{N_b}$ **then**
10: update $\{s_t^*\}_{t=1}^{N_b}$: save $s_{(j)}$ as s_t^*, for some t, keep the list ordered
11: save $\boldsymbol{m}_{\mathrm{w}}$ as $\boldsymbol{m}_t^*$ and $\boldsymbol{\Sigma}_{\mathrm{w}}$ as $\boldsymbol{\Sigma}_t^*$
12: **end if**
13: **end for**
14: initialize $s_{\mathrm{best}} = \infty$
15: **for** $t = 1$ to N_b **do**
16: iterate *utill convergence*: $(\boldsymbol{m}_t^{**}, \boldsymbol{\Sigma}_t^{**}) = \texttt{iter.rew}\,(\boldsymbol{m}_t^*, \boldsymbol{\Sigma}_t^*; \text{converge})$ (see Algorithm 2)
17: Mahalanobis distances $d_i = d_i(\boldsymbol{x}_i; \boldsymbol{m}_t^{**}, \boldsymbol{\Sigma}_t^{**})$, for $i = 1, \ldots, n$ (same as line 6 in Alg 2)
18: find scale s_t^{**} solving equation (3.14) (same as line 7 in Alg 2)
19: **if** $s_t^{**} < s_{\mathrm{best}}$ **then**
20: $s_{\mathrm{best}} = s_t^{**}$
21: $\boldsymbol{m}_{\mathrm{best}} = \boldsymbol{m}_t^{**}$ and $\boldsymbol{\Sigma}_{\mathrm{best}} = \boldsymbol{\Sigma}_t^{**}$
22: **end if**
23: **end for**
24: (optional) refine $(\boldsymbol{m}_{\mathrm{best}}, \boldsymbol{\Sigma}_{\mathrm{best}}) = \texttt{iter.rew}\,(\boldsymbol{m}_{\mathrm{best}}, \boldsymbol{\Sigma}_{\mathrm{best}}; \text{converge})$ with the *exact* reweighting as in eqns. (3.27)–(3.28) instead of approximate
25: **return** $(\boldsymbol{m}_{\mathrm{best}}, s_{\mathrm{best}}\boldsymbol{\Sigma}_{\mathrm{best}})$

The algorithm proceeds by taking a large number N_s of small subsamples of size n_s (line 3), computing their means and covariance matrices and reiterating from there. The **subsample size** n_s needs to be small enough to ensure the diversity of subsamples, but large enough to produce non-degenerate covariance estimates. For complete data, it is usually taken to be $n_s = p + 1$, which would be insufficient when some of the values are missing. We take

$$n_s = \lceil p/(1 - f_{\mathrm{mis}}) \rceil + 1,$$

where f_{mis} is the overall fraction of missing values in the dataset and $\lceil x \rceil$ denotes the smallest integer number larger than x.

Once a properly sized subsample is taken, its **location and covariance estimates** need to be computed (line 4). With complete data, the sample mean and the sample covariance can be used. With incomplete data, for the location estimate, we have chosen

Algorithm 2 `iter.rew` $(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0, r)$ — Iterative reweighting (called from Algorithm 1)

1: **if** $r =$ converge **then**
2: set $r :=$ some big number
3: **end if**
4: initialize `converged` := FALSE and $k := 1$
5: **while** $(k \leq r)$ & (NOT `converged`) **do**
6: compute Mahalanobis distances $d_i := d_i(\boldsymbol{x}_i; \boldsymbol{m}_{k-1}, \boldsymbol{\Sigma}_{k-1})$, for all $i = 1$ $*$
7: find scale s_k solving equation (3.14) $*$
8: compute weights $w_i := w_i(d_i/s_k)$, for $i = 1, \ldots, n$ $*$
9: $\boldsymbol{m}_k := m_{\mathrm{w}}(X, \{w_i\})$, *approximate* weighted estimate of location $*$
10: $\boldsymbol{\Sigma}_k := \boldsymbol{\Sigma}_{\mathrm{w}}(X, \{w_i\})$, *approximate* weighted estimate of scatter $*$
11: rescale $\boldsymbol{\Sigma}_k := c\boldsymbol{\Sigma}_k$, such that (3.15) holds (same as line 5 in Alg 1)
12: `converged` := $(||\boldsymbol{m}_k - \boldsymbol{m}_{k-1}||_2/||\boldsymbol{m}_k||_2 <$ some small number)
13: **end while**
14: **return** $(\boldsymbol{m}_r, \boldsymbol{\Sigma}_r)$

to use coordinate-wise means using only observed data. And the subsample covariance is estimated by first imputing coordinate-wise medians of the whole dataset in place of missing values in the subsample, and then computing the usual sample covariance matrix.

An alternative way to get initial estimates could be to compute MLE estimates under the normal model using the EM algorithm. However, the EM algorithm does not converge well when sample sizes are just slightly greater than the dimension and thus it was discarded. A modification of the EM algorithm is used, however, later on to compute weighted estimates of multivariate location and scatter.

Rescaling of $\boldsymbol{\Sigma}_0$ in line 5 is straightforward and can be done using the expression in (3.22). Note, however, that in actual computations we do not compute the sub-matrix determinants n times, but rather group the data according to their missingness patterns and compute the determinant for each pattern.

The next step is to do a small number r of **iterations of reweighting** using $(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0)$ as the starting point. Algorithm 2 describes how this is done. In line 6 we compute Mahalanobis distances for all the data points using the current values of $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$. Index i on function $d_i(\cdot)$ indicates that the distance should be computed in the subspace of $\mathbb{R}^p$ corresponding to the observed components of $\boldsymbol{x}_i$, as described in (3.16).

The **scale** s_k computed in line 7 is necessary for proper scaling of Mahalanobis distances d_i when using them to compute weights in line 8. The process of solving equation (3.14) is essentially the same as solving equation (3.10) of the complete data. The only difference is that the functions $\tilde{\rho}_i(\cdot)$ now depend on the dimension of the observed part of $\boldsymbol{x}_i$. The left-hand side of both equations is a non-increasing function of s and therefore they can easily be solved by iteratively computing

$$s_{(j+1)} = s_{(j)} \frac{1}{b} \frac{1}{n} \sum_{i=1}^{n} \tilde{\rho}_i \left(\frac{d_i(\boldsymbol{x}_i^{\mathrm{obs}}; \boldsymbol{m}, \boldsymbol{\Sigma})}{s_{(j)}} \right)$$

until some convergence criterion is reached.

The **weight function** $w_i(\cdot)$ should agree with the ρ-function $\tilde{\rho}_i(\cdot)$ to ensure the best convergence while iteratively reweighting. It can be shown (Maronna et al., 2006, p.220) that a traditional S-estimate can be represented as a weighted average and a weighted covariance matrix of the observations with weights equal to the derivative of the ρ-function applied to the Mahalanobis distances computed using the estimate itself. Therefore using the derivative of the ρ-function as the weight function for the iterative process is the natural choice. In our case of ρ-function varying from case to case we use

$$w_i(d) = \tilde{\rho}_i'(d) = \frac{c_i k_i}{n^{-1}\sum c_j k_j}\rho_{c_i}'(d) = \frac{3k_i}{n^{-1}\sum c_j k_j}\left(1 - \frac{d}{c_i}\right)^2 \text{ if } d < c_i, \text{ and 0 otherwise.} \tag{3.24}$$

We consider the **approximate weighted estimate** of location and scatter employed in lines 9 and 10 of Algorithm 2 to be an important contribution and will discuss it more detail in the next section.

3.4.2 Weighted estimate of multivariate location and scatter

Once weights are calculated, a weighted estimate of location and scatter for the whole dataset needs to be computed (lines 9 and 10 in Algorithm 2).

When data are complete, these weighted estimates are equal to

$$\hat{\boldsymbol{m}}_{\mathrm{w}} = m_{\mathrm{w}}(X, \{w_i\}) = \frac{\sum_{i=1}^n w_i \boldsymbol{x}_i}{\sum_{i=1}^n w_i}, \text{ and} \tag{3.25}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathrm{w}}(X, \{w_i\}) = \frac{\sum_{i=1}^n w_i (\boldsymbol{x}_i - \hat{\boldsymbol{m}}_{\mathrm{w}})(\boldsymbol{x}_i - \hat{\boldsymbol{m}}_{\mathrm{w}})'}{\sum_{i=1}^n w_i}, \tag{3.26}$$

and it is justified by the proof from (Maronna et al., 2006, p.220) which considers the derivatives $\frac{\partial s}{\partial \boldsymbol{m}}$ and $\frac{\partial s}{\partial \boldsymbol{\Sigma}}$ in the constrained optimization problem with $\det(\boldsymbol{\Sigma}) = 1$.

When some data are missing, a similar derivation can be done considering the Lagrangian for the constrained optimization problem in Definition 1. There are two major differences compared with the complete data situation. First, all Mahalanobis distances are computed based on the observed components only and therefore use different submatrices of $\boldsymbol{\Sigma}$. Second, the constraint (3.15) now involves the determinants of various submatrices of $\boldsymbol{\Sigma}$. This yields us the following two equations defining the reweighted estimates $\hat{\boldsymbol{m}}_{\mathrm{w}}$ and $\hat{\boldsymbol{\Sigma}}_{\mathrm{w}}$

$$\sum_{i=1}^n w_i \langle \hat{\boldsymbol{\Sigma}}_{\mathrm{w}[i]}^{-1} (\boldsymbol{x}_i^{\mathrm{obs}} - \hat{\boldsymbol{m}}_{\mathrm{w}[i]}) \rangle = 0 \tag{3.27}$$

$$\sum_{i=1}^n w_i \langle \hat{\boldsymbol{\Sigma}}_{\mathrm{w}[i]}^{-1} (\boldsymbol{x}_i^{\mathrm{obs}} - \hat{\boldsymbol{m}}_{\mathrm{w}[i]})(\boldsymbol{x}_i^{\mathrm{obs}} - \hat{\boldsymbol{m}}_{\mathrm{w}[i]})' \hat{\boldsymbol{\Sigma}}_{\mathrm{w}[i]}^{-1} \rangle = \text{const} \times \sum_{i=1}^n \langle \hat{\boldsymbol{\Sigma}}_{\mathrm{w}[i]}^{-1} \rangle, \tag{3.28}$$

with the exact value of const in (3.28) being not important because it only affects $\hat{\boldsymbol{\Sigma}}_{\mathrm{w}}$ as a multiplicative factor that goes away when we rescale $\boldsymbol{\Sigma}$ using (3.22). Angle brackets $\langle\cdot\rangle$ indicate that the vectors and matrices in dimensions less than p need to be expanded to size p with the gaps filled with zeros. For example, if $p = 4$ and some $\boldsymbol{x} = (x_1, \mathrm{NA}, x_3, \mathrm{NA})'$ then $\boldsymbol{x}^{\mathrm{obs}} = (x_1, x_3)' \in \mathbb{R}^2$ but $\langle \boldsymbol{x}^{\mathrm{obs}} \rangle = (x_1, 0, x_3, 0)' \in \mathbb{R}^4$.

Unfortunately, the solutions $\hat{\boldsymbol{m}}_{\mathrm{w}}$ and $\hat{\boldsymbol{\Sigma}}_{\mathrm{w}}$ to (3.27)–(3.28) are not easy to compute. A numerical algorithm such as Newton-Raphson can be used to solve them but, since the whole procedure needs to be repeated numerous times, it is unacceptably slow. The problem is also complicated by the fact that the solution is required to satisfy (3.15) and to be positive-definite, so that an alternative parametrization of $\hat{\boldsymbol{\Sigma}}_{\mathrm{w}}$ such as those described in Pinheiro and Bates (1996) needs to be used. It makes closed form expressions for the gradient and Hessian matrix of (3.27)–(3.28) unfeasible and the brute force optimization slow. We will refer to the solutions of (3.27)–(3.28) as the *exact weighted* mean and scatter and will use them only as a final tuning step once we get close to the S-estimate using an approximate reweighting described below.

Equations (3.27)–(3.28) resemble two other objects that we are familiar with. First, if there were no weights on the left hand side then they would be the ML equations for the multivariate normal data. On the other hand, if the weights were present on the both sides of (3.28) then the equations would define the MLE for the weighted sample (i.e. where the weight of each observation is taken to be w_i instead of $1/n$). In both cases, such equations could be efficiently solved by the EM algorithm: the regular EM in the first case, and the EM where every summation term corresponding to x_i is multiplied by w_i in the second. Our equations, however, do not correspond to any likelihood function and therefore cannot be solved by the un-modified EM-algorithm.

To compute an *approximate weighted* estimate we will use the ER-algorithm of Little and Smith (1987) with fixed weights. Our goal is to compute a reweighted estimate given the weights and therefore we are not going to recompute them on each iteration as it is done in the original paper. We compute the next iteration as

$$\boldsymbol{m}^{(t+1)} = \left(\sum_{i=1}^{n} w_i \hat{\boldsymbol{x}}_i^{(t)}\right) / \left(\sum_{i=1}^{n} w_i\right) \tag{3.29}$$

$$\boldsymbol{\Sigma}^{(t+1)} = \sum_{i=1}^{n} \left\{ w_i (\hat{\boldsymbol{x}}_i^{(t)} - \boldsymbol{m}^{(t)})(\hat{\boldsymbol{x}}_i^{(t)} - \boldsymbol{m}^{(t)})' + C_i^{(t)} \right\}, \tag{3.30}$$

where

$$\hat{\boldsymbol{x}}_i^{(t)} = \mathbb{E}\Big[\boldsymbol{X} \,\Big|\, \boldsymbol{X}_{[i]} = \boldsymbol{x}_i^{\mathrm{obs}}, \boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)} \Big] \text{ and} \tag{3.31}$$

$$C_i^{(t)} = \mathrm{Cov}\Big[\boldsymbol{X} \,\Big|\, \boldsymbol{X}_{[i]} = \boldsymbol{x}_i^{\mathrm{obs}}, \boldsymbol{m}^{(t)}, \boldsymbol{\Sigma}^{(t)} \Big]. \tag{3.32}$$

Closed form expressions for $\hat{\boldsymbol{x}}_i^{(t)}$ and $C_i^{(t)}$ (conditional distribution of a sub-vector of a

gaussian random vector given the remaining components) are well known and given below:

$$(\hat{\boldsymbol{x}}_i^{(t)})_{[i]} = \boldsymbol{x}_i^{\text{obs}}, \quad (\hat{\boldsymbol{x}}_i^{(t)})_{[-i]} = \boldsymbol{m}_{[-i]}^{(t)} + \boldsymbol{\Sigma}_{[-i,i]}^{(t)} \left(\boldsymbol{\Sigma}_{[i]}^{(t)}\right)^{-1} (\boldsymbol{x}_i^{\text{obs}} - \boldsymbol{m}_{[i]}^{(t)}) \tag{3.33}$$

$$(C_i^{(t)})_{[-i,-i]} = \boldsymbol{\Sigma}_{[-i,-i]}^{(t)} - \boldsymbol{\Sigma}_{[-i,i]}^{(t)} \left(\boldsymbol{\Sigma}_{[i]}^{(t)}\right)^{-1} \boldsymbol{\Sigma}_{[i,-i]}^{(t)}, \text{ and zeros elsewhere.} \tag{3.34}$$

They can be efficiently computed using Gauss–Jordan sweep (SWP) operator. Subscripts $[-i]$ indicate that only the variables corresponding to the missing components of the ith case need to be considered. Note that the weight in (3.30) only multiplies the part that depends on the data but not the $C_i^{(t)}$. This directly corresponds to what we saw in (3.28) where the right hand side does not depend on $\boldsymbol{x}_i$ and does not carry weights.

For a given set of weights the iterative procedure in (3.29)–(3.32) is repeated until some convergence criterion is reached yielding us the final $\hat{\boldsymbol{m}}_{\text{aw}}$ and $\hat{\boldsymbol{\Sigma}}_{\text{aw}}$ that we consider to be good enough approximations of $\hat{\boldsymbol{m}}_{\text{w}}$ and $\hat{\boldsymbol{\Sigma}}_w$ given by (3.27)–(3.28).

After preliminary subsampling and more thorough iterations through the N_b best subsamples are done, the iterative reweighting can be performed on $(\boldsymbol{m}_{\text{best}}, \boldsymbol{\Sigma}_{\text{best}})$ with the slow *exact weighting* step (at lines 9 and 10 in Algorithm 2) to refine the final estimate and to ensure that it satisfies Definition 1 exactly. In practice, however, we have noticed that the faster approximate ER-based weighting performs just as well as the slow exact procedure.

3.4.3 Comparison with the ER-algorithm and the ERTBS estimate

In this section we describe the differences between the extended S-estimate, ER-algorithm and ERTBS.

The main conceptual difference is that in this paper we provide a *definition* of the estimate together with an *algorithm* that attempts to compute it. The extended S-estimate is defined as a pair $(\boldsymbol{m}, \boldsymbol{\Sigma})$ that minimizes a robust scale $s(\boldsymbol{m}, \boldsymbol{\Sigma})$ defined in (3.14). On the other hand, both the ER-algorithm and ERTBS describe just an algorithm rather than an estimate per se. Having a proper definition allows us to study the properties of the estimate and to fine tune the corresponding computational algorithm.

This fundamental distinction also affects the actual algorithm that is used to compute the estimate. Little and Smith (1987) propose that the ER-algorithm starts iterating from the MLE of the location-scatter that can be found using the EM-algorithm. As duly noted by Cheng and Victoria-Feser (2002) this approach has a low breakdown point. The starting value can be so badly affected by outliers that the ER-algorithm will have no chance to downweight the contamination properly and will converge to a value far away from the truth. Copt and Victoria-Feser (2003) suggest to remedy this problem by starting the iterations from a location-scatter estimate that is already robust and has high breakdown point. For this purpose they develop modifications to the Fast-MCD of Rousseeuw and Van Driessen (1999) and OGK pairwise covariance estimate of Maronna and Zamar (2002)

that make them capable of analyzing incomplete data. Once the starting value is chosen the ERTBS algorithm iterates to convergence and the limiting value is returned as the estimate. The extended S-estimate, however, has an objective function to optimize and the algorithm is designed with that in mind. Iterative reweighting is done because it improves the scale: we observed in simulations that the exact weighted estimate consistently reduces the scale on each step. The approximate weighted estimate, unfortunately, does not have this exact property but it reduces the scale most of the time. Multiple initial subsamples are taken and several best of them are chosen based on the scale that can be achieved starting from them.

Another difference arises from the way we see the reweighting step. For the extended S-estimate, the approximate weighted covariance, which closely resembles the ER-algorithm, is only a computationally cheaper version of the exact weighted covariance. Therefore updating the weights while iterating between E- and R-steps would be inappropriate. Both ER-algorithm by Little and Smith (1987) and ERTBS do update weights on every step of their iterations.

Going into more technical but nevertheless important details, let us consider how an individual step of the ER-algorithm is performed. When updating $\mathbf{\Sigma}$ in (3.30), only the part depending on the data is multiplied by the weight but not the C_i. This is also the case in Little and Smith (1987). If both terms were downweighted, as done by Cheng and Victoria-Feser (2002), then the iterative algorithm would become equivalent to the EM-algorithm performed on the weighted sample which will find the MLE for the weighted sample. When weights are based on the data, and especially on the incomplete data, this is undesirable as it will lead to a biased estimate even when there is no contamination in the data. Such a procedure does not estimate the parameters of the distribution of interest but rather the parameters of the weighted distribution where points with large Mahalanobis distances have smaller weights than in the original distribution. When data are complete and the weights are based on the full Mahalanobis distances this only leads to the underestimation of variances but the correlation structure remains unbiased — this is why the weighted covariance matrix (which can also be seen as the MLE of the weighted sample) is adequate in the complete-data scenario. When some of the data are missing and the weights are based on partial Mahalanobis distances the bias becomes more dangerous: not only the variances are underestimated but also the relationships between variables are distorted.

Although we use the slightly modified ER-algorithm to compute the approximate weighted mean and covariance, the most importance difference is the weights involved. The ones used in the original ER-algorithm put the x-part and the C_i part of (3.30) on two different scales. If we consider the unconditional expected value of one term in (3.30) assuming that $\boldsymbol{m}^{(t)}$ and $\mathbf{\Sigma}^{(t)}$ are already equal to the true values of the parameters, we would like the expectation to be equal to the true $\mathbf{\Sigma}$ as well. In other words, once we have come close to the true $\boldsymbol{m}$ and $\mathbf{\Sigma}$ we do not want to go away. With the weights

used by Little and Smith (1987), however, the part of the matrix corresponding to the observed components is going to be downweighted and the part corresponding to the missing variables will enter with weight one which will cause the combination of them to be biased. Our weights are scaled up (by means of the constants k_i in front of the ρ-function) to ensure that everything is on the same scale regardless of how many components of a given observation are missing. It is confirmed further in simulations that the extended S-estimate is unbiased when sample size is medium to large and the data are coming from the multivariate normal distribution with missing values.

The last little difference between (3.29) and the updating of the location vector in the ER-algorithm is that we use the same weights for the mean vector as for the covariance matrix. Little and Smith (1987) chose to use the square roots of the covariance weights following what was done by Campbell (1980).

This concludes the definition of the extended S-estimator and we move on to the empirical assessment of its performance in the next section.

3.5 Simulations results and numerical evaluation

3.5.1 Monte Carlo study

In order to evaluate the performance of the new method we conduct a simulation study. Our main goal is to assess how well the new extended S-estimate can handle missing values and compare it with the S-estimate on the full data without missing values and with the conventional alternatives discussed in Section 3.2.

To make the simulations setup representative of the problems encountered in real scientific research, we used the correlation structure of the data from the Palomar Digital Sky Survey (DPOSS) (Djorgovski et al., 1998). As noted in Theorem 2, our estimate is location and scale equivariant and therefore we can consider data that are centered at zero and have unit variances without loss of generality. We estimated the correlation matrix of the full DPOSS dataset (27 variables, 132,402 cases) using the S-estimate and saved it as $\widetilde{\mathbf{\Sigma}}$. "Clean data" in our simulations are drawn from the multivariate normal distribution with mean zero and covariance matrix $\widetilde{\mathbf{\Sigma}}$. The data in the DPOSS dataset are highly correlated (the condition number of $\widetilde{\mathbf{\Sigma}}$ is approximately 5,000) presenting us with a challenging estimation problem. To further emulate the real world, we also draw "outliers" from the DPOSS dataset to add to the randomly generated clean data. We consider three different levels of contamination in this simulation study: 0%(clean data), 5% and 10%.

The procedure described below is repeated $N = 2{,}000$ times so that we are able to estimate the expected value of the estimates as well as their sampling variances.

1. Data generation:
   (a) generate a sample X_{clean} of size $n = 1{,}000$ from $\mathcal{N}(\mathbf{0}_{27}, \widetilde{\mathbf{\Sigma}})$;

(b) obtain X_{cont5} (and X_{cont10}) by replacing a fraction $\varepsilon_{\text{cont}} = 0.05$(or 0.10) of cases of X_{clean} with cases randomly drawn from the pool of outliers in the original DPOSS dataset (top 20% of Mahalanobis distances); since our simulated data have zero mean and unit variances, the outliers are also standardized using robust univariate estimates of location and scale based on the whole dataset

(c) randomly choose which cells are going to be missing; they are uniformly distributed over all cases and variables, with the total fraction fixed at $\varepsilon_{\text{mis}} = 0.1$

- obtain X_{mis} by introducing missing values to X_{clean}
- obtain $X_{\text{cont5,mis}}$ (and $X_{\text{cont10,mis}}$) by introducing missing values to X_{cont5} (or X_{cont10})

2. Compute estimates on *incomplete* data. Calculate the following five estimates on each of X_{mis}, $X_{\text{cont5,mis}}$ and $X_{\text{cont10,mis}}$

   (a) extended S-estimate; this is our main interest

   (b) ERTBS[6] covariance estimate of Copt and Victoria-Feser (2003); this is our main competitor

   (c) S-estimate[7] on only complete cases; this is reasonable alternative when ε_{mis} and dimension are not very large

   (d) matrix of pairwise S-estimates; this estimate is likely to fail due to its inability to recognize multidimensional structures

   (e) Gaussian MLE by means of the EM-algorithm; on clean data it will serve as a gold standard, and on contaminated data it will give an indication of the severity of contamination.

3. Compute estimates on *complete* data which we will use as baseline. We only compute

   (a) S-estimate; and

   (b) sample covariance (MLE),

   because ERTBS reduces to a form of S-estimate when no data are missing and so does the complete cases S-estimate.

Note that all S-estimates here and in the following sections, as well as the example in Section 3.2, have been computed with the value $b = 0.5$ that ensures the maximum possible breakdown point. Additionally, the following values of the tuning parameters were used for the Fast-S algorithm (both original and extended): $N_s = 100$, $N_b = 5$ and $r = 5$ (see Algorithm 1 for their meanings).

To make our evaluation criteria more invariant to the correlation structure of the data, we transform all covariance estimates using the true covariance matrix $\widetilde{\mathbf{\Sigma}}$, so that our target

[6] The R and Fortran code was kindly provided by the authors.

[7] In addition to the S-estimate we also computed the MCD covariance estimate based on complete cases, but its performance was even worse than that of $\hat{\mathbf{\Sigma}}_{\text{cc}}$ and therefore the numerical results are not reported.

	complete data		with 10% missing data				
$\varepsilon_{\text{cont}} \downarrow$	S	MLE	**ext S**	ERTBS	S cc only	S pw	MLE via EM
clean	0.78	0.76	**0.97**	1.70	14.23	1,292	0.94
5%	0.98	7.85	**1.19**	1.71	17.00	1,563	8.24
10%	1.48	24.82	**1.69**	1.75	21.23	1,912	25.52

Table 3.1. Simulation results showing **mean squared errors** (i.e. average L_2-distances to the identity matrix) for the standardized covariance estimates. "ext S" is the extended S-estimate; "S cc only" is the S-estimate on complete cases; and "S pw" is the pairwise S-estimate.

is now the identity matrix. We call this *standardization* of the covariance estimates:

$$\text{std}_{\widetilde{\boldsymbol{\Sigma}}}(\hat{\boldsymbol{\Sigma}}) = (\widetilde{\boldsymbol{\Sigma}}^{-\frac{1}{2}})'\hat{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{\Sigma}}^{-\frac{1}{2}}, \tag{3.35}$$

where $\hat{\boldsymbol{\Sigma}}$ is any of the covariance estimates described above. Note that this standardization is invariant to scaling transformations of the data and therefore our choice of data with unit variances maintains its generality.

The estimates that we compute are highly multidimensional and therefore, in order to be able to evaluate their performance, we need to choose some appropriate distances in the space of 27-dimensional positive definite matrices. The first measure that we use is simply the L_2-norm (Frobenius norm) between the standardized covariance estimate and the identity matrix

$$d_1(\hat{\boldsymbol{\Sigma}}) = ||\text{std}_{\widetilde{\boldsymbol{\Sigma}}}(\hat{\boldsymbol{\Sigma}}) - \boldsymbol{I}||_2^2. \tag{3.36}$$

For the N replicates, we compute the average $N^{-1}\sum_{i=1}^{N} d_1(\hat{\boldsymbol{\Sigma}}_i)$, and it is, in effect, the mean squared error (MSE) for the chosen estimator. Table 3.1 summarizes them for the estimators mentioned above. A number of observations can be made based on the results of these simulations:

1. The extended S-estimate compares favourably with the competitors both in the presence of outliers and without them. The runner up (ERTBS estimate) has MSE at least 40% larger.
2. Although not shown in the table, by decomposing the MSE into the squared bias and the variance, we can see that the extended S estimate has no bias with the clean data and under 5% contamination the bias is responsible for the 8% of the MSE.
3. On the other hand, the ERTBS estimate has some intrinsic bias constituting about 25% of the MSE even when computed on the data with no contamination (but with 10% missing values). Its variance is also larger than that of our estimate (1.3 squared units vs 1.1).
4. When 10% of missing data is introduced, the MSE of the extended S-estimate goes up by approximately 20%. This may seem extensive at first, considering that the amount of data was only reduced by ten per cent, but we can see that it is on par

with the efficiency loss of the ML estimate under the same missingness scenario. More on this in Section 3.5.2.

5. When data are clean, the S-estimate is almost as efficient as the Gaussian ML estimate of covariance. This is well known for the regular S-estimates in higher dimensions and we confirm that this is also true for the extended S-estimate.
6. Outliers (at least those used in these simulations) have their moderate negative influence on the S-estimates but the effect is about the same in the presence of missing data as it is under the complete data scenario. Therefore we conjecture that the extended S-estimate inherits the good robustness properties of the traditional S-estimate.
7. Both pairwise and the complete-cases approaches are hardly an alternative to a specialized method such as the extended S-estimate or ERTBS. The average number of complete cases in this scenario is only $n(1-\varepsilon_{\text{mis}})^p = 1{,}000 \times 0.9^{27} \approx 58$ which is hardly enough to obtain a reasonable estimate of covariance matrix in dimension $p = 27$. The problem of the pairwise estimate is that it completely misses the multivariate structure of the data. If the Frobenius norm was computed directly on the estimated matrix (without standardization) then it would likely perform on par with other estimates, but the standardization step exposes the lack of multivariate coherency in it.

We also employ another measure of quality to investigate how well the multivariate structure of the covariance matrix is estimated. This time we look at the condition numbers of the standardized covariance estimates:

$$d_2(\hat{\boldsymbol{\Sigma}}) = \frac{\boldsymbol{v}_1(\text{std}_{\widetilde{\boldsymbol{\Sigma}}}(\hat{\boldsymbol{\Sigma}}))}{\boldsymbol{v}_p(\text{std}_{\widetilde{\boldsymbol{\Sigma}}}(\hat{\boldsymbol{\Sigma}}))},$$

where $\boldsymbol{v}_1$ and $\boldsymbol{v}_p$ are the largest and smallest eigenvalues of the given matrix respectively. Note that $d_2(\boldsymbol{\Sigma}) \geq 1$ for any $\boldsymbol{\Sigma}$, and $d_2(\boldsymbol{\Sigma}) = 1$ whenever $\boldsymbol{\Sigma} = \widetilde{\boldsymbol{\Sigma}}$. This measure is by no means an absolute description of the shape of the estimated matrix but it is rather just one of many possible characteristics and many researches find it useful; in particular, this measure is used in Maronna et al. (2006).

Table 3.2 shows the averages and standard deviations of $d_2(\hat{\boldsymbol{\Sigma}}_i)$ over $i = 1, \ldots, N$. The closer the expected value is to 1 or, in other words, the smaller it is, the better the estimate. The ML estimate in the "complete and clean data" corner of the table gives us an idea of what is the best performance that we may hope to achieve for the given data setup.

Because pairwise estimates $\hat{\boldsymbol{\Sigma}}_{\text{pw}}$ may or may not be positive-definite, rendering the computation of condition numbers meaningless, we will apply the correcting procedure outlined by Maronna and Zamar (2002) to enforce positive-definiteness. We have also tried using the corrected pairwise estimates to compute the values in Table 3.1 but it

	complete data		with 10% missing data				
$\varepsilon_{\text{cont}} \downarrow$	S	MLE	**ext S**	ERTBS	S cc only	S pw	MLE via EM
clean	1.86	1.85	**2.02**	3.10	25.9	261.7	2.00
	(0.08)	(0.06)	(**0.08**)	(0.4)	(11.9)	(675.3)	(0.08)
5%	1.89	4.08	**2.04**	3.12	35.7	764.7	4.31
	(0.06)	(1.10)	(**0.08**)	(0.4)	(33.9)	(3,445)	(1.18)
10%	1.96	6.22	**2.11**	3.14	44.3	720.9	6.55
	(0.08)	(1.54)	(**0.09**)	(0.4)	(159.1)	(2,357)	(1.65)

Table 3.2. Simulation results showing expected values and sampling standard deviations of the **condition numbers** of standardized covariance estimates.

turns out that the un-corrected estimates perform an order of magnitude better from the L_2 point view. This is natural considering that the L_2-norm is pairwise in nature and the cost of the forced positive-definiteness is the disturbance to individual components of the covariance matrix resulting in increased variance and even bias. One may argue that a matrix which is not semi-positive definite cannot be called an estimate of the covariance matrix, to which we say that then the pairwise numbers in Table 3.1 should simply be seen as a lower bound for the MSE of the corrected pairwise estimate (and they are large enough to not consider the pairwise estimate as a competitor). When comparing condition numbers, however, the correcting procedure must be applied to ensure that they are positive.

Table 3.2 shows about the same tendencies as Table 3.1. Outliers and the presence of missing data have some negative effect on the S-estimates but it is well under control. ERTBS performs relatively well but not as well as the extended S-estimate. The alternatives, however, perform even worse in terms of estimating the condition number than they did in the L_2-norm. For the complete-cases S-estimate, this can probably be attributed to the fact that the reduced sample size (due to removing all cases with at least one value missing) causes the estimates to become more singular and therefore have much larger condition number. The problem with the pairwise estimate is that it is incapable of capturing any multivariate structure of the data and thus the eigen-structure estimated by the pairwise estimates is more or less arbitrary.

3.5.2 Performance on clean data

Objective. In this subsection we conduct another simulation study to explore the behaviour of the extended S-estimate under clean but partially missing multivariate normal data. The following properties will be discussed further in this section: bias (finite sample and asymptotic), consistency and the rate of convergence, asymptotic normality, and the asymptotic loss of efficiency due to missing values. Covariance estimates $\hat{\mathbf{\Sigma}}$ are multi-dimensional structures consisting of $p(p+1)/2$ univariate parameters. A proper theory should handle them as one unit and consider multi-dimensional deviations from

the true value (bias as vector, sampling variance-covariance as matrix). To simplify things in our Monte Carlo exploratory study we will treat the $p(p+1)/2$ parameter estimates $\hat{\mathbf{\Sigma}}_{ab}$, $1 \leq a \leq b \leq p$, completely separately from each other. We will focus on the *univariate* bias $\mathbb{E}\hat{\mathbf{\Sigma}}_{ab} - \mathbf{\Sigma}_{ab}$, sampling variance $\mathbf{Var}\hat{\mathbf{\Sigma}}_{ab}$ and Mean Squared Error $\mathrm{MSE}(\hat{\mathbf{\Sigma}}_{ab}) = \mathbb{E}(\hat{\mathbf{\Sigma}}_{ab} - \mathbf{\Sigma}_{ab})^2$, for all pairs of a and b. Due to the limited space we will report numerical results for one or two representative combinations of (a, b) in some detail and briefly verify that the findings hold true for all other elements of $\hat{\mathbf{\Sigma}}$.

Simulation setup. This time we use dimensions $p_1 = 5$ and $p_2 = 10$ and the true covariance matrix $\mathbf{\Sigma}$ with moderate correlations. Half of the diagonal of $\mathbf{\Sigma}$ is equal to 1, the other half is equal to 2 and all off-diagonal elements are equal to 0.5 (so that $\mathrm{cond}(\mathbf{\Sigma}_{(5)}) = 7.4$, $\mathrm{cond}(\mathbf{\Sigma}_{(10)}) = 12.1$). We generate random samples from a multivariate normal distribution with location zero and covariance $\mathbf{\Sigma}$ and introduce random missingness (uniformly over all variables and cases) such that the total fraction of missing data is 0%(fully observed), 5%, 10%, 20% or 40%. To explore the trends as the sample size increases, we run $N = 1{,}000$ replicates for each of the various sample sizes n ranging from 20 to 10,000. We consider the fully observed case as a reference as we know that the traditional S-estimate is asymptotically normal on complete data (Davies, 1987).

3.5.2.1 Bias

Recall that our choice of scaling constants k_i in front of the ρ-function (and therefore weights) was motivated by the desire to make the extended S-estimate at least asymptotically unbiased. It was supported by Theorem 1 which guarantees that the extended S-estimate is at least "locally" Fisher-consistent. In this set of simulations we attempt to investigate whether the asymptotic result is transferable to finite sample sizes.

To investigate the bias we focus on the estimated expected value for each of the $p(p+1)/2$ elements of $\hat{\mathbf{\Sigma}}$ based on the $N = 1{,}000$ replicates:

$$\hat{\mathbb{E}}\hat{\mathbf{\Sigma}}_{ab} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{\Sigma}_{ab}^{(i)}.$$

An illustration of how these values typically vary depending on n and ε_{mis} is given in Table 3.3 using two elements of $\hat{\mathbf{\Sigma}}$ as examples. In the left half of the table corresponding to $\mathbf{\Sigma}_{11}$ the target value is the variance of the first variable and is equal to 1; on the right, it is a covariance element $\mathbf{\Sigma}_{12} = 0.5$. Being Monte Carlo quantities, these expected values have a certain amount of random error in them which can be characterized by their estimated standard errors

$$\widehat{\mathrm{se}}(\hat{\mathbf{\Sigma}}_{ab}) = \frac{1}{N}\frac{1}{N-1}\sum_{i=1}^{N}\left(\hat{\mathbf{\Sigma}}_{ab}^{(i)} - \hat{\mathbb{E}}\hat{\mathbf{\Sigma}}_{ab}\right)^2.$$

	$\boldsymbol{\Sigma}_{11} = 1$					$\boldsymbol{\Sigma}_{12} = 0.5$				
$\varepsilon_{\text{mis}} \rightarrow$	0	0.05	0.1	0.2	0.4	0	0.05	0.1	0.2	0.4
$n = 20$	**1.09**	**1.13**	**1.22**	**1.31**	**1.31**	0.53	**0.55**	**0.57**	**0.58**	0.49
$n = 50$	0.98	0.99	0.99	1.02	**1.15**	0.49	0.49	0.49	0.50	0.54
$n = 100$	*1.00*	*1.00*	*1.00*	*1.00*	1.04	*0.50*	*0.50*	*0.50*	*0.50*	0.51
$n = 200$	*1.00*	*1.00*	*1.00*	*0.99*	*1.00*	*0.49*	*0.49*	*0.50*	*0.50*	*0.50*
$n \geq 500$	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*	*0.50*	*0.50*	*0.50*	*0.50*	*0.50*

Table 3.3. Estimated expected values of $\hat{\boldsymbol{\Sigma}}_{11}$ and $\hat{\boldsymbol{\Sigma}}_{12}$ in $p = 5$ for a variety of sample sizes and proportions of missingness. Compare to the true values $\boldsymbol{\Sigma}_{11} = 1$ and $\boldsymbol{\Sigma}_{12} = 0.5$ for the assessment of bias. Elements shown in boldface have estimated bias in excess of 3 estimated standard errors and therefore are deemed to be biased. Italicized area is where no bias has been detected for any elements of $\hat{\boldsymbol{\Sigma}}$ (not only these two). Results for $p = 10$ are very similar (not shown).

Values in Table 3.3 that differ from their corresponding true values by more than three times their estimated standard errors are shown in boldface because it is likely that the estimate is biased under such configurations of n and ε_{mis}. For these two elements of $\hat{\boldsymbol{\Sigma}}$ we can say with enough certainty that the estimates become unbiased (at least for practical purposes) as soon as n is larger than 200 or even 100.

We have computed the quantities described above for all elements of $\hat{\boldsymbol{\Sigma}}$ and performed formal t-tests on all of them to check whether the averages differ significantly from the corresponding true values of $\boldsymbol{\Sigma}$. When sample size is 200 or more we have been unable to detect any bias for any elements of $\boldsymbol{\Sigma}$. The same is true for the sample size of 100 and the proportion of missingness of 20% or less. This region is shown in *italic* in Table 3.3. The results for the smaller sample sizes (20 and 50) or higher proportions of missingness (40% on sample size 100) are relatively unstable and suggest that there might be some small bias. We have not been able to characterize it or eliminate its source.

$\varepsilon_{\text{mis}} \rightarrow$	0	0.05	0.1	0.2
$\boldsymbol{\Sigma}_{11} = 1.0$	1	0.97	0.94	0.90
$\boldsymbol{\Sigma}_{44} = 2.0$	1	0.97	0.93	0.87
$\boldsymbol{\Sigma}_{12} = 0.5$	1	1.01	1.02	1.07

Table 3.4. Asymptotic multiplicative bias of ERTBS estimate: $\hat{\text{E}}(\hat{\boldsymbol{\Sigma}}_{ab})/\boldsymbol{\Sigma}_{ab}$ for three representative elements of $\hat{\boldsymbol{\Sigma}}$. Results shown are for $p = 5$. Bias for $p = 10$ is similar.

A similar set of simulations for ERTBS reveals that it has a persistent bias even with a larger sample size of 1,000 and smaller proportion of missing data such as 5%. For our covariance structure the ERTBS underestimates diagonal elements (variances) and overestimates the rest (covariances) as shown in Table 3.4. We could not study the performance of ERTBS for $n >$ 1,000 or proportion of missingness exceeding 20% because the computer code became unstable for larger values of these parameters. Note that a corresponding table for the extended S-estimate would consist of all 1's because the

estimate has no detectable bias for larger sample sizes.

3.5.2.2 Consistency and rate of convergence

In this subsection the size n of the sample on which the estimate $\hat{\mathbf{\Sigma}}$ is computed becomes of more importance so we introduce new notation to address this dependence: $\hat{\mathbf{\Sigma}}_{ab|n}$ will stand for an estimate for $\mathbf{\Sigma}_{ab}$ based on a sample of size n. To investigate whether the estimates $\hat{\mathbf{\Sigma}}_{ab|n}$ are consistent as $n \to \infty$ we consider their estimated mean squared error for all values of n and will observe that it converges to 0, which implies the convergence in probability to the true parameter because, for any $\varepsilon > 0$, using Markov's inequality it we get

$$\mathbb{P}\Big[\Big|\hat{\mathbf{\Sigma}}_{ab|n} - \mathbf{\Sigma}_{ab}\Big| < \varepsilon\Big] = \mathbb{P}\Big[\Big(\hat{\mathbf{\Sigma}}_{ab|n} - \mathbf{\Sigma}_{ab}\Big)^2 < \varepsilon^2\Big] < \frac{\mathbb{E}\Big[\Big(\hat{\mathbf{\Sigma}}_{ab|n} - \mathbf{\Sigma}_{ab}\Big)^2\Big]}{\varepsilon^2} \to 0, \text{ as } n \to \infty.$$

We conjecture that $\mathrm{MSE}(\hat{\mathbf{\Sigma}}_{ab|n})$ not only converges to zero but does so at the usual rate of $1/n$, i.e. that $\mathrm{MSE}(\hat{\mathbf{\Sigma}}_{ab|n}) = O(1/n)$. To check this (and the fact that they converge at all) we consider values of $n \times \widehat{\mathrm{MSE}}(\hat{\mathbf{\Sigma}}_{ab|n})$ that are expected to be of order $O(1)$ if the rate of convergence is indeed $1/n$. These quantities for one diagonal and one off-diagonal element of $\hat{\mathbf{\Sigma}}$ (in dimension $p = 5$) are shown in Table 3.5 for all five levels of missingness. Looking at this table one line at a time, we can see that the variation between numbers

	$n \to$	20	50	100	200	500	1,000	2,000	5,000	10,000
$\mathbf{\Sigma}_{11}$	$\varepsilon_{\text{mis}} = 0$	4.07	2.53	2.70	2.35	2.66	2.41	2.60	2.55	2.57
	$\varepsilon_{\text{mis}} = 0.05$	4.97	2.69	2.91	2.54	2.90	2.62	2.78	2.71	2.72
	$\varepsilon_{\text{mis}} = 0.10$	9.57	3.03	3.03	2.69	3.09	2.75	3.01	2.94	3.01
	$\varepsilon_{\text{mis}} = 0.20$	16.78	3.80	3.79	3.35	3.62	3.51	3.44	3.63	3.56
	$\varepsilon_{\text{mis}} = 0.40$	30.08	12.49	8.18	5.69	6.20	5.34	5.78	5.25	5.98
$\mathbf{\Sigma}_{12}$	$\varepsilon_{\text{mis}} = 0$	2.40	1.69	1.79	1.53	1.56	1.60	1.59	1.65	1.63
	$\varepsilon_{\text{mis}} = 0.05$	2.91	1.85	1.93	1.63	1.70	1.71	1.72	1.79	1.76
	$\varepsilon_{\text{mis}} = 0.10$	4.46	2.04	2.10	1.91	1.84	1.88	1.93	1.94	1.96
	$\varepsilon_{\text{mis}} = 0.20$	5.55	2.87	2.59	2.25	2.31	2.21	2.23	2.41	2.17
	$\varepsilon_{\text{mis}} = 0.40$	11.21	8.63	5.51	4.14	3.91	3.87	3.76	3.79	4.02

Table 3.5. Values of $n \times \widehat{\mathrm{MSE}}(\hat{\mathbf{\Sigma}}_{ab})$ to show that the rate of convergence of individual elements of $\mathbf{\Sigma}$ is $1/n$. Numbers within each line are stable, or at least stabilize as n becomes larger, suggesting that $\mathrm{MSE}(\hat{\mathbf{\Sigma}}_{ab}) = O(1/n)$ for all values of ε_{mis}. Other elements of $\hat{\mathbf{\Sigma}}$ exhibit similar behaviour. Results shown are for $p = 5$.

becomes smaller as n gets bigger and can be seen as converging to a constant or, in other words, behaving as $O(1)$. In particular, the numbers do not increase, which implies that the MSE's are indeed converging to zero, validating our consistency claim. Additionally, if we consider one column at a time, this table gives us an idea of how the variance of

individual estimates increases when the proportion of missing data becomes larger; we will summarize this in a more condensed form in section 3.5.2.4.

As usual, we have looked at other 13 elements of $\hat{\mathbf{\Sigma}}$ as well as all 55 elements of $\hat{\mathbf{\Sigma}}$ when $p = 10$, and believe that consistency holds in all situations. We can also consider a combined MSE for the whole matrix estimate by adding up individual MSE's for all elements of $\mathbf{\Sigma}$. The resulting overall estimated MSE also converges to zero at the $1/n$ rate.

3.5.2.3 Asymptotic normality

To evaluate whether estimates $\hat{\mathbf{\Sigma}}_{ab|n}$ are (univariately) asymptotically normal we perform a test of normality on each collection (of size $N = 1{,}000$) of estimates $\{\hat{\mathbf{\Sigma}}^{(i)}_{ab|n}\}_{i=1}^{N}$ for all combinations of (a, b) and various values of n. We will only consider the situation $\varepsilon_{\text{mis}} = 0.4$ as it is the most extreme and thus the estimate is the most likely to deviate from good behaviour. If it appears to be normal under these conditions, it is most likely to be normal when the proportion of missingness is lower.

For each collection of Monte Carlo replicates of estimates $\{\hat{\mathbf{\Sigma}}^{(i)}_{ab|n}\}_{i=1}^{N}$ we conducted six tests of normality available in `nortest` package in `R`: Anderson–Darling, Cramer–von Mises, Kolmogorov–Smirnov, Pearson χ^2, Shapiro–Francia and Shapiro–Wilk's (e.g. Thode (2002)), and obtained p-values for them. Then we chose the median of the six p-values to be our aggregated p-value for the normality test of this sample. For each sample size n (and ε_{mis} which is fixed) we have $p(p-1)/2$ of these p-values which are subject to multiple comparison problems and therefore the smallest of them is likely to be fairly small even when data (i.e. estimate values) are indeed normal. To assess whether it is plausible that the observed p-values come from the Uniform$[0, 1]$ distribution, i.e. that the estimates follow normal distribution, we look at several smallest p-values from each batch of $p(p+1)/2$ aggregated p-values. Denote them $\pi_{(1)}, \pi_{(2)}, \ldots, \pi_{(7)}$ (seven is an arbitrary number) and compare to their expected values based on the first seven order statistics of a sample of size $p(p+1)/2$ from the Uniform$[0, 1]$ distribution. The results are shown in Table 3.6. The last line is the benchmark we compare all other lines to. If observed values are much smaller than their expected values than we can reject H_0 that the estimate values are normally distributed. The numbers suggest that the estimates are definitely not normal for $n \leq 200$, probably not normal for $n = 500$ (for the large proportion of missingness $\varepsilon_{\text{mis}} = 0.4$) and are likely to be approximately normal for $n \geq 1{,}000$. Therefore we can say that our simulation study suggests that the extended S-estimates are asymptotically (univariately) normal.

3.5.2.4 Efficiency

Once it is known that the variance of the estimate converges to zero at the usual $1/n$ rate, an interesting question to ask is what is the efficiency loss associated with the missingness

$n \downarrow$	$\pi_{(1)}$	$\pi_{(2)}$	$\pi_{(3)}$	$\pi_{(4)}$	$\pi_{(5)}$	$\pi_{(6)}$	$\pi_{(7)}$
$\leq$ **100**	0.000	0.000	0.000	0.000	0.000	0.000	0.000
200	0.000	0.001	0.001	0.003	0.004	0.009	0.013
500	0.001	0.002	0.048	0.051	0.053	0.079	0.095
1,000	0.041	0.047	0.065	0.135	0.158	0.198	0.207
2,000	0.048	0.065	0.072	0.087	0.103	0.137	0.158
5,000	0.051	0.084	0.114	0.139	0.186	0.199	0.249
10,000	0.020	0.059	0.060	0.064	0.079	0.086	0.113
expected H_0	0.018	0.036	0.054	0.072	0.090	0.108	0.126

Table 3.6. Several (seven) most extreme (among all $p(p+1)/2$ elements of $\hat{\mathbf{\Sigma}}$) p-values from univariate normality tests. "Expected" is computed under H_0 that data are normal and considering the $p(p+1)/2$ tests as independent. When $n \geq 1{,}000$, the observed p-values are in agreement with their expected values so we can conclude that all their distributions are approximately normal. Results shown are for $p = 10$.

of the data. We already mentioned this issue briefly in sections 3.5.1 and 3.5.2.2 but now we can provide more complete answers. It turns out that the efficiency loss only mildly depends on the dimension of the data. We have also run an identical set of simulations for the Maximum Likelihood estimator (using EM algorithm) which we can use as a gold standard. Table 3.7 shows the average efficiency loss relative to the fully observed data:

$$\frac{1}{p(p+1)/2}\sum_{a=1}^{p}\sum_{b=a}^{p}\frac{\widehat{\mathrm{Var}}\left[\hat{\mathbf{\Sigma}}_{ab|n}(\text{partly missing data})\right]}{\widehat{\mathrm{Var}}\left[\hat{\mathbf{\Sigma}}_{ab|n}(\text{complete data})\right]}.$$

Note that the sample size in the numerator and denominator are taken to be the same

$\varepsilon_{\text{mis}} \rightarrow$	0.05	0.1	0.2	0.4
ext S, $p = 5$	1.08	1.19	1.45	2.54
ext S, $p = 5$	1.07	1.15	1.35	2.16
MLE, $p = 5$	1.07	1.14	1.34	2.02
MLE, $p = 10$	1.06	1.13	1.31	1.95
imaginary best	1.05	1.11	1.25	1.67

Table 3.7. Efficiency loss due to missing values. Simulation results for moderate to weak correlation structure.

but the ratio itself does not depend on n as long as n is sufficiently large ($n \geq 500$ is this example); for smaller n the efficiency deteriorates a little faster. The last line is the imaginary best case scenario where we assume that all missing values were concentrated on as few cases as possible instead of being spread all over the dataset. It would completely eliminate $n \times \varepsilon_{\text{mis}}$ cases and the efficiency loss (of a fully efficient estimate) due to the reduced sample size would be $(1 - \varepsilon_{\text{mis}})^{-1}$.

In these simulations, for clean data with $\varepsilon_{\text{mis}} = 0$ (i.e. the baseline for Table 3.7),

the asymptotic relative efficiency of the S-estimate w.r.t. the Maximum Likelihood estimate (i.e. sample covariance) was 0.79 for $p = 5$ and 0.91 for $p = 10$. These numbers are provided to give an idea of the relative efficiencies *between* the lines in Table 3.7.

Note that all the above results are only valid for the parameter values used in these simulations. The lack of affine equivariance makes it difficult or impossible to generalize results and the best we can do is to consider different typical scenarios. We have made an attempt to diversify over different types of data in the several examples/simulations that we present in this paper.

3.5.3 Performance on real data

In order to assess how the extended S-estimate performs on real data, we consider the *ionosphere* dataset from the UCI repository (Asuncion and Newman, 2007). The dataset contains 225 "informative" complex-valued measurements from 16 radar antennas (total $p = 32$) located in Goose Bay, Labrador measuring free electrons in the ionosphere.

The dataset does not have missing data, so we introduce some artificial random missingness and see how the new S-estimate performs compared with other alternatives. To establish the "gold standard" we compute the traditional S-estimate $(\hat{\boldsymbol{m}}_{\mathrm{c}}, \hat{\boldsymbol{\Sigma}}_{\mathrm{c}})$ of location and scatter on the full complete dataset. Corresponding Mahalanobis distances, shown in Figure 3.3, reveal that 75 out of 225 data points can be considered outliers w.r.t. the cutoff value $c^* = 8.56$ chosen according to (3.3). This justifies the need for a robust estimate when estimating location and scatter parameters for this dataset.

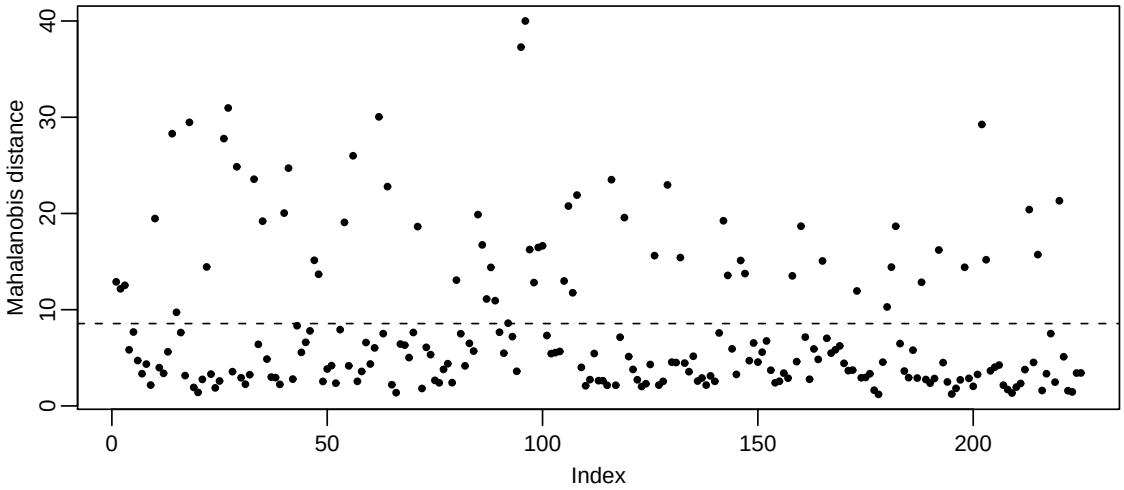


Figure 3.3. Robust Mahalanobis distances for ionosphere data.

To account for the effect of randomness when generating missingness structure, we conduct a bootstrap-like simulation where we generate 200 different random missingness patterns for each level of $\varepsilon_{\mathrm{mis}} = 0.03; 0.05; 0.1; 0.2$. We control the proportion of missing data for each variable in the dataset to be exactly $\varepsilon_{\mathrm{mis}}$. For each of the resulting

datasets (that only differ in what values are considered missing) we compute (1) the extended S-estimate; (2) ERTBS estimate; (3) the S-estimate based on complete cases; and (4) the pairwise S-estimate (corrected for positive-definiteness when necessary).

The expected number of complete cases for the four chosen levels of missingness are 84.9, 43.6, 7.7 and 0.18. Therefore the *complete-cases* analysis cannot be conducted for $\varepsilon_{\text{mis}} > 0.05$ because the dimension of the data would exceed the number of available data points. Additionally, although we can see no theoretical reason for it, *ERTBS* also fails to compute for proportions of missingness of 10% or more. For these two types of estimates we will only report the results for 3% and 5% of missing data.

As in Section 3.5.1, the covariance estimates are 32×32 matrices with non-trivial correlation structure and as such they require some adequate distance measure to summarize their performance. We use four different criteria (distances) and show that the new S-estimate compares favourably with the alternatives in all four of them. Let $(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\Sigma}})$ be some estimate of the multivariate location-scatter computed on the data with a fraction of observations regarded as missing.

The first two measures are the same distances used in Section 3.5.1:

$$d_1(\hat{\boldsymbol{\Sigma}}) = ||\text{std}_{\hat{\boldsymbol{\Sigma}}_c}(\hat{\boldsymbol{\Sigma}}) - \boldsymbol{I}||_2 \quad \text{and} \quad d_2(\hat{\boldsymbol{\Sigma}}) = \frac{\boldsymbol{v}_1(\text{std}_{\hat{\boldsymbol{\Sigma}}_c}(\hat{\boldsymbol{\Sigma}}))}{\boldsymbol{v}_p(\text{std}_{\hat{\boldsymbol{\Sigma}}_c}(\hat{\boldsymbol{\Sigma}}))}.$$

To add another dimension to the evaluation of the eigen-structure of the covariance estimates, we also computed determinants of the standardized matrices. Note that the target value for the determinants is 1 and the estimates can be both larger or smaller than that. To make the assessment easier we consider absolute values of the logarithms of the determinants:

$$d_3(\hat{\boldsymbol{\Sigma}}) = \left|\log\left(\det\left(\text{std}_{\hat{\boldsymbol{\Sigma}}_c}(\hat{\boldsymbol{\Sigma}})\right)\right)\right|,$$

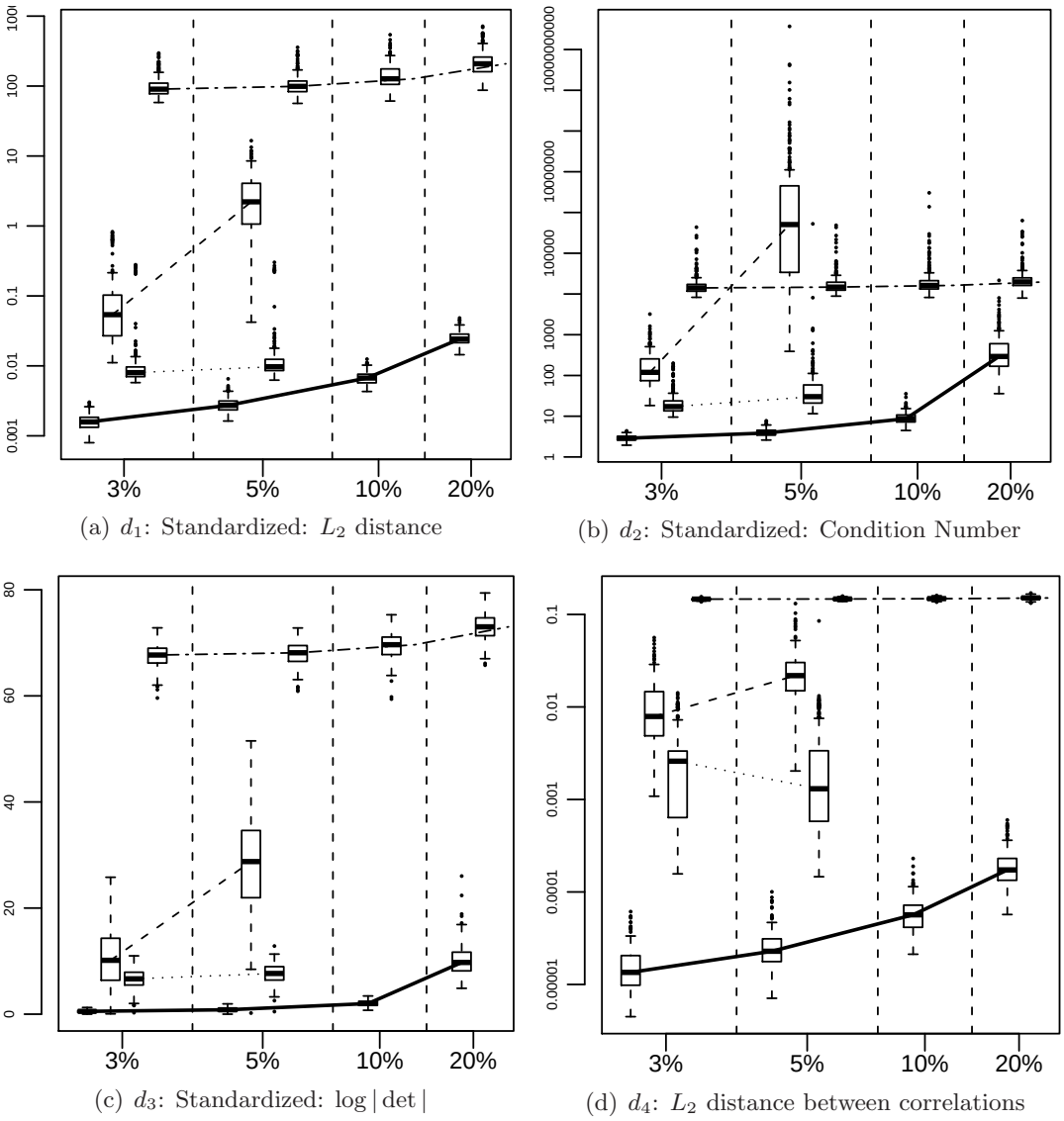
so that smaller values of $d_3(\cdot)$ correspond to better estimates.

The last measure that we use is simply the L_2-distance between the estimated correlation matrix and the true one:

$$d_4(\hat{\boldsymbol{\Sigma}}) = ||\text{Corr}(\hat{\boldsymbol{\Sigma}}) - \text{Corr}(\hat{\boldsymbol{\Sigma}}_c)||_2, \tag{3.37}$$

where $\text{Corr}(\cdot)$ denotes the correlation matrix corresponding to the given covariance matrix. We chose to work with correlation matrices instead of the covariance in order to equalize the effects that variables with different variances will have on this criterion.

As in Section 3.5.1 we selectively correct pairwise estimates for positive-definiteness using the procedure from Maronna and Zamar (2002). For the d_1-criterion the estimates were left un-corrected because this produced slightly better results. For d_4, however, the correction improved the performance of the pairwise estimate so we report it that way. Criteria d_2 and d_3 require correction because they lose their meaning if the matrix is not



(a) d_1: Standardized: L_2 distance

(b) d_2: Standardized: Condition Number

(c) d_3: Standardized: $\log|\det|$

(d) d_4: L_2 distance between correlations

Figure 3.4. Boxplots for the four estimates with different levels of missingness and four criteria to evaluate their quality. Bold solid lines are our S-estimates, dotted lines are ERTBS estimates, dashed lines are complete-cases S-estimates, and dash-dot lines are the pairwise S-estimates. All graphs are shown on the log-scale except the log-determinant which is a logarithm itself.

positive-definite.

Figure 3.4 summarizes the results of the simulation study on the ionosphere data. Four panels correspond to the four criteria described above and all of them are shown on the logarithmic scale (except d_3 which is already a logarithm itself). Each panel shows the behaviour of the four estimates over the four missingness levels. Note that for $\varepsilon_{\text{mis}} = 0.1$ and $\varepsilon_{\text{mis}} = 0.2$ we do not have results for ERTBS and the complete-cases S-estimate because they could not be computed due excessive amount of missing values or insufficient amounts of data respectively.

Each graph consists of a series of four boxplots connected with lines to ease identification. The horizontal axis is ordinal and is not drawn to scale. The results for the extended S-estimate (connected with solid bold lines) are almost uniformly better than those for the competitors. If we look at the worst (i.e. largest) result out of the 200 runs for the S-estimate (for each criterion and each missingness level) they are still better (i.e. lower) than the best result for the competitors. This demonstrates that we have achieved a noticeable improvement over available methods and, in some cases, we can do estimation when no reasonable alternative (other than pairwise methods) was available.

In principle, the poor performance of ERTBS might be attributed to the unfair choice of gold standard. To address this issue we have tried to compute the deviations of ERTBS from the same estimate computed on the complete data. Unfortunately, the computer code for ERTBS doesn't run on complete data. To remedy this problem we introduced one single missing value at a random location in the data. It turned out that the estimate varies a lot depending on the location of the missing cell (standard deviations are 30 times bigger than the corresponding quantities for the extended S-estimate). So we decided to average the covariance estimate over all possible locations of the missing cell and used that average as "gold standard" for ERTBS. Only d_3 improved slightly but was still significantly worse than that of the extended S-estimate. The other three distances deteriorated even more.

3.6 Conclusions

We have defined an estimate that can robustly estimate multivariate location and scatter in the presence of missing data. Our method applies the idea of the likelihood of the observed data to the S-estimate proposed by Davies (1987). The main innovation is the definition of the extended S-estimate on incomplete data that involves Mahalanobis distances of the observed part of each observation and the modified ρ-function that varies from one observation to another depending on the number of observed variables in each case. A special scaling is applied to the ρ-function to ensure that the estimate has no asymptotic bias (which is a non-trivial achievement in the presence of missing data).

We have also modified the Fast-S algorithm of Salibian-Barrera and Yohai (2006) to be able to compute the newly defined extended S-estimates. A number of changes had to be made but the most important is the extension of the weighted average and weighted sample covariance matrix to datasets with missing values.

We have applied the algorithm to real as well as randomly generated data and showed that it performs noticeably better than ERTBS of Cheng and Victoria-Feser (2002) and an order of magnitude better than impromptu methods such as complete-cases analysis or pairwise robust covariance matrix estimates with levels of missingness as high as 20% (and potentially higher). Simulations also suggest that when applied to clean multivariate normal data with missing values, the estimate is $\sqrt{n}$-consistent, asymptotically normal and is comparable with MLE in its response to missingness.

Chapter 4

Final discussion

4.1 Combined robust estimate

In Chapter 2 we discussed scatter estimates that are robust under independent contamination: individually contaminated cells spread all around the dataset. An important assumption, however, was that it was the only type of contamination and no case-wise outliers were considered. A natural extension of Chapter 2 would be to allow both types of contamination to be present at the same time: most cases have one or two contaminated cells but on top of that there is a certain proportion of completely contaminated cases.

Assuming that we only want to focus on point-mass contamination, the task of looking for the worst possible contamination was relatively simple in the independent contamination situation — the only parameter to vary was the magnitude (distance from origin) of contamination that we denoted k. With case-wise contamination, we also need to worry about the direction in which the contamination is located in addition to its magnitude.

In this section we will explain a possible approach to treating both types of contamination simultaneously and try to give motivation for why we think it might work. The discussion will be illustrated by simple numerical examples which are not meant to be all-encompassing. We do, however, believe that the general ideas hold in most scenarios and are not limited to the specific examples shown here.

A Conjecture. Detection methods for independent contamination can be used to disarm case-wise outliers if the covariance structure is known well enough to identify them. If we remove several cells from a case-wise outlier[8] so that the remaining sub-vector is not outlying anymore, its influence on the covariance estimate is significantly reduced and can be tolerated.

This can be illustrated with an example. Consider a large, $n = 500$, dataset from multivariate normal distribution of $p = 10$ with highly correlated covariance matrix $\mathbf{\Sigma}_h$ (as per section 2.1.4.3). We independently contaminate it with probability $\varepsilon_{\text{ind}} = 0.05$ at value $k = 2.5$. On top of that we add $\varepsilon_{\text{str}} = 0.1$ fraction of case-wise contamination at location $\boldsymbol{x}_c$. Then we apply P-approach detection (with $p_{\text{cutoff}} = 0.05$) using the true covariance matrix $\mathbf{\Sigma}_h$ and mark suspected values as missing to compute the MLE of covariance using the EM-algorithm. This pseudo-estimate is then standardized and the LRT-statistic (see section 2.1.4.3) is computed. It will be used as the main assessment tool

[8] By *outlier* here we mean a case with a large Mahalanobis distance with respect to the true covariance structure

in this section. We have tried a variety of locations for $\boldsymbol{x}_c$ (by randomly sampling directions on a sphere and trying different magnitudes for each of them) and not-surprisingly have found that the worst LRT-bias is produced by contamination located along the eigenvector of $\boldsymbol{\Sigma}_h$ corresponding to its smallest eigenvalue, $\boldsymbol{x}_c = m\boldsymbol{v}_{10}$, with some $m \in \mathbb{R}$. We will use case-wise contamination of this form throughout the section. If the value of m is too large than the contamination is detectable and the bias is small. If the value of m is too small than the contamination does not do as much damage as it possibly could. The trend is shown in Figure 4.1 and we can see that the most dangerous value of m is such that $\mathrm{MD}(\boldsymbol{x}_c; \mathbf{0}, \boldsymbol{\Sigma}_h) = m/(\sqrt{\lambda_{10}}||\boldsymbol{v}_{10}||_2) = 4.2$ and worst value of the LRT-bias is approximately 1.2. To get an idea of whether 1.2 is a small number or large, we have

Figure 4.1. LRT-bias of the ML pseudo-estimate of covariance after performing P-approach outlier detection with known covariance matrix. For a sense of scale compare to the numbers in Table 4.1.

computed the amount of LRT-bias than can be observed if no filtering is performed at all. To see the really dangerous effect of contamination, $\boldsymbol{x}_c$ has to be placed further away from the origin. The curve in Figure 4.1 remains flat at 0.14 for such magnitudes of $\boldsymbol{x}_c$. Sample covariance matrix (with no filtering) was computed and the results are shown in Table 4.1. To illustrate the relative bias coming from the two types of contamination we provide the numbers for two scenarios: (a) only case-wise contamination at $\boldsymbol{x}_c$; (b) independent contamination plus case-wise (the same scenario as in Fugure 4.1 above). We can see

$\mathrm{MD}(\boldsymbol{x}_c)$	4.2	5	10	50	100	500
Case + Cell-wise	466.9	466.8	469.0	648.6	1278.2	22534.2
Case-wise only	0.685	1.115	6.745	220.0	894.4	22514.2

Table 4.1. LRT-bias of sample covariance without outlier detection against the magnitude of case-wise contamination. Compare these numbers to those in Figure 4.1.

that for some values of $\boldsymbol{x}_c$ the filtering (with known covariance) reduces the amount of bias dramatically and conclude that the problem of dual contamination can be solved if only we can obtain a reasonably good estimate to perform the detection with. Thus the conjecture is plausible.

The Problem. The true covariance matrix is not known and needs to be estimated. The iterative estimate in Section 2.3.7 is not robust against structural[9] contamination. The lack of robustness is due to the MLE estimate which is used on every step of the iterative procedure to update the covariance structure after new filtering information becomes available. This is the usual chicken-and-egg problem of using non-robust estimates to perform outlier detection: the outliers affect the estimates and thus mask themselves from being detected. No matter how many times the procedure is repeated the outliers may never get discovered.

To illustrate this problem we continue the example described above and attempt to filter out the structural contamination with the iterative detection procedure without using the knowledge of the true covariance structure. We compute the MLE covariance estimates, which are now true estimates because no information about the data generating distribution has been used to produce them, and their standardized LRT-scores as before. Results for several values of $\mathrm{MD}(\boldsymbol{x}_c)$ are shown in Table 4.2 which has the same structure as Table 4.1 above. The bias increases up to very large numbers as contamination moves away

$\mathrm{MD}(\boldsymbol{x}_c)$	4.2	5	10	50	100	500
Case+Cell + Detection	1.32	2.08	9.99	256.5	957.6	0.82

Table 4.2. LRT-bias with MLE-based iterative detection against the magnitude of structural contamination.

from the center. The apparent drop at $\mathrm{MD}(\boldsymbol{x}_c) = 500$ is due to the univariate filtering finally being able to detect some components of the contamination. But multivariate detection methods with the MLE-based updating alone are incapable of detecting the contamination and therefore cannot be considered robust under this scenario.

The Solution. If the conjecture above is true then all that is needed to make the estimate of section 2.3.7 robust to case-wise contamination is to replace the non-robust MLE step by a robust method that can estimate covariance matrix in the presence of missing data and structural outliers. Such an estimate has been constructed in Chapter 3. The new iterative algorithm can be schematically described by the following steps (omitting location estimate for clarity):

1. compute some initial estimate $\boldsymbol{\Sigma}^{(0)}$ on dataset X; set $i = 0$
2. detect contaminated values in X with P-approach using $\boldsymbol{\Sigma}^{(i)}$ as the assumed covariance structure; denote filtered dataset as $X^{(i+1)}$

[9] By *structural* here we understand a case-wise contamination that cannot be detected using simple univariate detection methods.

3. compute $\mathbf{\Sigma}^{(i+1)}$, an extended S-estimate of covariance based on $X^{(i+1)}$; set $i = i+1$; go to step 2.

To demonstrate that such an estimate has desired robustness properties we continue the example studied above. We follow the same strategy: add fraction $\varepsilon_{\text{str}} = 0.1$ of contamination of the form $\boldsymbol{x}_c = m\boldsymbol{v}_{10}$ with different values of m to a dataset which already has $\varepsilon_{\text{ind}} = 0.05$ of independent contamination at $k = 2.5$ in it. The LRT-score of standardized estimates is shown in Figure 4.2 and it can be seen that the worst value is only 3.0 and is achieved at $\text{MD}(\boldsymbol{x}_c) = 5.7$. Any contamination further away from the origin is detected, partially eliminated and does not inflict much damage on the estimate. The worst case bias is larger but still on the same scale as what could be achieved had the true covariance matrix been known (reproduced as the dashed line in the same graph). On the other hand, it is many orders of magnitude smaller than that of the pure MLE-based procedure of Section 2.3.7 shown in Table 4.2.

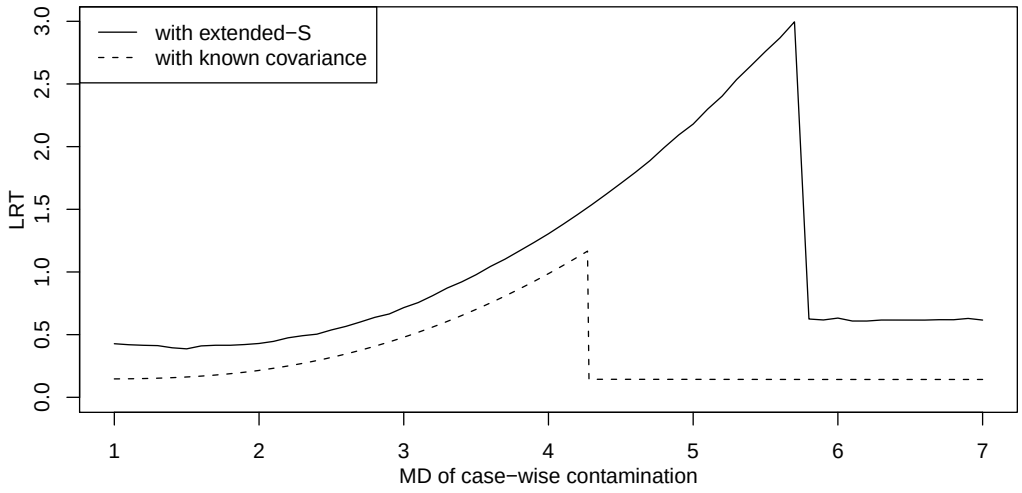


Figure 4.2. LRT-bias of iterative detection procedure based on the extended S-estimate of covariance. Dashed line for the pseudo-estimate with known covariance is the same as the solid line in Figure 4.1 and is reproduced here for the ease of comparison.

While not being fully general, this example suggests that combining the results of Chapters 2 and 3 has a potential for an estimate robust against both types of contamination. The ability to process data with missing values is also automatically built into the estimates.

Difficulties. Although the combined procedure described above is promising in terms of robustness, there are several concerns that need to be taken into account and resolved in the future.

- Computational efficiency. The extended S-estimate is an iterative procedure (reweighting) relying on an iterative procedure (EM-algorithm). Incorporating this structure into another iterative procedure (detection-estimation) yields an estimate that is

fairly slow even in dimension $p = 10$ and currently is prohibitively slow for $p = 20$. Mixing the two iteration loops into one by updating weights and the set of contaminated values at the same time might be an option. Computing "quick and dirty" version of S-estimate instead of completing full iterations until convergence is another. Tradeoffs need to be studied to evaluate where time can be saved for the smallest loss of precision.

- Efficiency of outlier detection. When no case-wise contamination is present the statistical efficiency of the extended S-estimate is worse than that of the MLE. This leads to the covariance estimates that are less effective in identifying independent contamination. The extended S-estimate has nothing to offer dealing with independent contamination because the proportion of contaminated cases is too high. Thus using the extended S-estimate on data that only has independent contamination is pure loss compared to using the MLE. This is not just a loss of efficiency as is typical for robust estimates — here the loss of precision in covariance estimates leads to more contaminated cells being undetected and therefore higher bias. Again, the tradeoffs need to be studied before any recommendations are made.

4.2 Other directions of future work

In addition to merging the results of Chapter 2 and 3, we see a number of other directions in which the work done in this thesis can be continued. With respect to Chapter 2, below are a few topics that we consider important and intriguing.

- *Definition of the estimate as an estimating equation or optimization problem.* Currently we have only defined the estimate in section 2.3.7 as a heuristic iterative algorithm. In order to be able to study its statistical properties we will likely need a more solid definition similar to how the extended S-estimate is defined in (3.14). The iterative procedure could then we viewed as one possible way of computing such an estimate. One possible way to tackle this problem can be to change the contamination model (2.1) into fully parametric model by assuming that the contaminating distribution G is known (e.g. a heavy-tailed family such as t-distributions) and then find what the maximum likelihood estimate would be in such scenario.

- *Non-zero weights for cells.* As a result of multivariate detection in Chapter 2 we declare each cell as either clean or contaminated. This is parallel to completely removing a data case from analysis in traditional robustness. Most optimal robust methods however do not completely discard suspicious cases but rather downweight them with a weight somewhere between zero and one. We imagine that a similar strategy would also be preferable for dealing with contaminated cells (not only cases) but a method for incorporating cell-weight information into the covariance estimates

is still to be devised. Censoring suspicious values instead of marking them as MAR is already a step in that direction, but more control over the weights is still desirable.

- *Affine-equivariance and covariance estimation.* We would like to extend the δ-consistency theory of Alqallaf et al. (2009) from location estimates to scatter matrices. In other words, we aim to prove mathematically what has been shown numerically in section 2.1.4.2.

- *Conditional censoring and Winsorising.* Univariate censoring is done by estimating the expected value $\hat{m}$ and standard deviation $\hat{s}$ of a data cell and if the observed value exceeds $\hat{m} + a\hat{s}$ for some $a > 0$ then the value is removed and the MLE is computed conditional on the fact that the value was larger than $\hat{m} + a\hat{s}$. Similarly when the observed value is less $\hat{m} - a\hat{s}$. Armed with multivariate detection methods, once suspicious values within a data case have been identified, we can compute the *conditional* expected value $\hat{m}_c$ and standard deviation $\hat{s}_c$ given all trustworthy values in the case. Then we can use $\hat{m}_c \pm a\hat{s}_c$ as the censoring constant for the particular data cell. Same reasoning can be applied to Winsorising: replace a large value by $\hat{m}_c \pm a\hat{s}_c$ instead of its unconditional counterpart.

- *Outlier detection in different dimensions.* In section 2.3.8 we saw that univariate detection may be in contradiction with multivariate screening: what appears to be an outlier in one dimension (because it exceeds a cutoff value based on χ^2_1 distribution) might not be detected by considering its full Mahalanobis distance and comparing it to the corresponding quantile of χ^2_p. We feel that this problem of what is and what is not an outlier should be studied in more detail. Initial questions that we can focus on are: (a) How common is it that a univariate outlier does not push the full Mahalanobis distance above the cutoff? (b) How dangerous are they? If such an outlier slips through the detection process, how much bias will it produce in the covariance estimate? The difference between dimensions does not have to be as extreme as 1 against p. For example, we may want to explore the differences in outlier detection based on full Mahalanobis distances of dimension p and partial distances of dimension $p - 1$.

Regarding the extended S-estimate defined and studied in Chapter 3, the following theoretical questions appear to be of relatively high importance.

- *Uniqueness of the solution to the optimization problem defining the estimate.* In particular, what are the assumptions about the configuration of missing values that will ensure that the solution exists and is unique.

- *Asymptotic normality* has been conjectured but not proven yet. Influence Function and Asymptotic variance are likely possible to be computed by differentiating estimating equations for the extended S-estimate.

- *Breakdown point.* We expect that the presence of missing data, and therefore the reduced amount of information about the clean distribution, will have a negative effect on the breakdown point of S-estimates. This should be studied and quantified.

Bibliography

Alqallaf, F. (2003), "A new contamination model for robust estimation with large high-dimensional data sets," Ph.D. thesis, University of British Columbia.

Alqallaf, F., Van Aelst, S., Yohai, V., and Zamar, R. (2009), "Propagation of outliers in multivariate data," *Ann. Statist.*, 37, 311–331.

Asuncion, A. and Newman, D. (2007), "UCI Machine Learning Repository," .

Campbell, N. A. (1980), "Robust Procedures in Multivariate Analysis I: Robust Covariance Estimation," *Applied Statistics*, 29, 231–237.

Casella, G. and George, E. (1992), "Explaining the Gibbs sampler," *American Statistician*, 46, 167–174.

Cheng, T.-C. and Victoria-Feser, M.-P. (2002), "High-breakdown estimation of multivariate mean and covariance with missing observations," *British Journal of Mathematical and Statistical Psychology*, 55, 317–335.

Copt, S. and Victoria-Feser, M.-P. (2003), "Fast algorithms for computing high breakdown covariance matrices with missing data," Tech. Rep. 2003.04, Universite de Geneve.

Davies, P. (1987), "Asymptotic Behaviour of S-Estimates of Multivariate Location Parameters and Dispersion Matrices," *Ann. Statist.*, 15, 1269–1292.

Dempster, A., Laird, N., and Rubin, D. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, 1–38.

Djorgovski, S. G., Gal, R. R., Odewahn, S. C., de Carvalho, R. R., Brunner, R., Longo, G., and Scaramella, R. (1998), "The Palomar Digital Sky Survey (DPOSS)," .

Donoho, D. L. (1982), "Breakdown properties of Multivariate Location Estimators," Ph.D. thesis, Harvard University, Cambridge, MA.

Goodman, N. R. (1963), "The Distribution of the Determinant of a Complex Wishart Distributed Matrix," *The Annals of Mathematical Statistics*, 34, 178–180.

Joossens, K. and Roelant, E. (2007), "Fast algorithm to compute the S-estimation of location and covariance (in R)," `http://www.econ.kuleuven.be/public/NDBAE06/programs/locest/fastsloc.r.txt`.

Little, R. J. and Rubin, D. B. (2002), *Statistical Analysis with Missing Data*, Wiley-Interscience.

Little, R. J. and Smith, P. J. (1987), "Editing and imputing for quantitative survey data," *J. Amer. Statist. Assoc.*, 82, 58–68.

Maronna, R., Martin, R., and Yohai, V. (2006), *Robust Statistics: Theory and Methods*, J. Wiley.

Maronna, R. A. and Zamar, R. H. (2002), "Robust Estimates of Location and Dispersion for High-Dimensional Datasets," *Technometrics*, 44, 307–317.

Murphy, S. A., Gupta, A. D., Cain, K. C., Johnson, L. C., Lohan, J., Wu, L., and Mekwa, J. (1999), "Changes in Parents' Mental Distress After the Violent Death of An Adolescent or Young Adult Child: A Longitudinal Prospective Analysis," *Death Studies*, 23, 129–159.

Petersen, K. B. and Pedersen, M. S. (2008), "The Matrix Cookbook," Version 20080216.

Pinheiro, J. and Bates, D. (1996), "Unconstrained parametrizations for variance-covariance matrices," *Statistics and Computing*, 6, 289–296.

Robert, C. (1995), "Simulation of truncated normal variables," *Statistics and computing*, 5, 121–125.

Rocke, D. M. (1996), "Robustness properties of S-estimators of multivariate location and shape in high dimension," *Ann. Statist.*, 24, 1327–1345.

Rousseeuw, P. (1985), "Multivariate estimation with high breakdown point," *Mathematical Statistics and Applications*, 8, 283–297.

Rousseeuw, P. and Molenberghs, G. (1993), "Transformation of non positive semidefinite correlation matrices," *Communications in Statistics-Theory and Methods*, 22, 965–984.

Rousseeuw, P. J. and Van Driessen, K. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, 41.

Rubin, D. (1987), "Multiple Imputation for Nonresponse in Surveys. New York: J," .

— (2004), *Multiple imputation for nonresponse in surveys*, Wiley-IEEE.

Salibian-Barrera, M. and Yohai, V. (2006), "A fast algorithm for S-regression estimates," *Journal of Computational and Graphical Statistics*, 15, 414–427.

Seber, G. A. (2004), *Multivariate observations*, John Wiley & Sons Inc.

Stahel, W. A. (1981), "Breakdown of covariance estimators," Tech. rep., Fachgruppe für Statistik, ETH, Zürich.

Thode, H. (2002), *Testing for normality*, CRC.

Tukey, J. (1960), "A survey of sampling from contaminated distributions," *Contributions to probability and statistics*, 448–485.

Van Aelst, S., Vandervieren, E., and Willems, G. (2009), "Stahel-Donoho Estimators with Cellwise Weights," *Journal of Statistical Computation and Simulation*, (to appear).

Wu, L. (2010), *Mixed effects models for Complex Data*, CRC Press.

Appendix A

Supplementary material for chapter 2

A.1 Asymptotic effect of independent contamination

Consider two random variables X and Y and their independently contaminated versions $\tilde{X} = (1-b_X)X + b_X X^*$ and $\tilde{Y} = (1-b_Y)Y + b_Y Y^*$, where b_X and b_X are Bernoulli random variables with probabilities of success ε_X and ε_Y respectively. Random variables X^* and Y^* represent an arbitrary contamination. Following the independent contamination model we assume that all random variables except the pair (X,Y) are mutually independent.

Without loss of generality, we assume that $\mathbb{E}[X] = \mathbb{E}[Y] = 0$.

Expectations are the easiest: $\mathbb{E}\left[\tilde{X}\right] = \varepsilon_X \mathbb{E}[X^*]$ and $\mathbb{E}\left[\tilde{Y}\right] = \varepsilon_Y \mathbb{E}[Y^*]$. The effect of the contamination will be linearly proportional to the proportion and the magnitude of the contamination.

Then proceed with the variances: $\mathbb{E}\left[\tilde{X}^2\right] = (1-\varepsilon_X)\mathbb{E}\left[X^2\right] + \varepsilon_X \mathbb{E}\left[(X^*)^2\right]$ and therefore

$$\mathbf{Var}\tilde{X} = \mathbb{E}\left[\tilde{X}^2\right] - (\mathbb{E}\tilde{X})^2 = (1-\varepsilon_X)\mathbb{E}\left[X^2\right] + \varepsilon_X \mathbb{E}\left[(X^*)^2\right] - \varepsilon_X^2 (\mathbb{E}[X^*])^2 = (1-\varepsilon_X)\mathbf{Var}X + \varepsilon_x(1-\varepsilon_X)\mathbb{E}\left[(X^*)^2\right] + \varepsilon_X^2 \mathbf{Var}(X^*), \quad \text{(A.1)}$$

which is guaranteed to be larger than $\mathbf{Var}X$ provided that $\mathbb{E}\left[(X^*)^2\right] > (1-\varepsilon_X)^{-1}\mathbf{Var}X$. The corresponding is, of course, true for $\mathbf{Var}\tilde{Y}$.

Covariance is follows:

$$\mathbf{Cov}(\tilde{X},\tilde{Y}) = \mathbb{E}\left[\tilde{X}\tilde{Y}\right] - \mathbb{E}\left[\tilde{X}\right]\mathbb{E}\left[\tilde{Y}\right] = (1-\varepsilon_X)(1-\varepsilon_Y)\mathbb{E}[XY] + \varepsilon_X\varepsilon_Y\mathbb{E}[X^*Y^*] - \varepsilon_X\mathbb{E}[X^*]\,\varepsilon_Y\mathbb{E}[Y^*] = (1-\varepsilon_X)(1-\varepsilon_Y)\mathbb{E}[XY] + \varepsilon_X\varepsilon_Y(\mathbb{E}X^*Y^* - \mathbb{E}[X^*]\,\mathbb{E}[Y^*]) = (1-\varepsilon_X)(1-\varepsilon_Y)\mathbf{Cov}(X,Y) + \varepsilon_X\varepsilon_Y\mathbf{Cov}(X^*,Y^*) = (1-\varepsilon_X)(1-\varepsilon_Y)\mathbf{Cov}(X,Y), \quad \text{(A.2)}$$

which is clearly smaller than $\mathbf{Cov}(X,Y)$ provided that the contamination is indeed independent. Another interesting observation is that the bias in the covariance does not depend on the distribution of the contamination but only on the amount of it.

Despite the fact that variances can be reduced by special types of independent con-

tamination (inliers), the correlations can be shown to be always reduced in absolute value:

$$\left|\mathbf{Cor}(\tilde{X},\tilde{Y})\right| = \left|\frac{\mathbf{Cov}(\tilde{X},\tilde{Y})}{\sqrt{\mathbf{Var}\tilde{X}\mathbf{Var}\tilde{Y}}}\right| \leq \left|\frac{(1-\varepsilon_X)(1-\varepsilon_Y)\mathbf{Cov}(X,Y)}{\sqrt{(1-\varepsilon_X)\mathbf{Var}X(1-\varepsilon_Y)\mathbf{Var}Y}}\right| = \sqrt{(1-\varepsilon_X)(1-\varepsilon_Y)}\left|\mathbf{Cor}(X,Y)\right|. \quad \text{(A.3)}$$

Note, however, that this is a very rough upper bound because typically the variances will be dominated by the term corresponding to the magnitude of the contamination such as $\varepsilon_x(1-\varepsilon_X)\mathbb{E}\big[(X^*)^2\big]$.

Conclusions from this basic math:

- Covariances and correlations are reduced in absolute value by the independent contamination
- Covariances only depend on the proportions of contamination
- Variances also depend on the magnitude of the contamination
- Variances are usually increased but can be slightly reduced by inliers.

A.2 Proof: differences of Mahalanobis distances

Lemma 1. *Let* $\boldsymbol{x} = \begin{pmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{pmatrix} \in \mathbb{R}^p$, $\boldsymbol{m} = \begin{pmatrix}\boldsymbol{m}_1\\ \boldsymbol{m}_2\end{pmatrix} \in \mathbb{R}^p$ *and* $\boldsymbol{\Sigma} = \begin{pmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{pmatrix} \in \mathrm{SPD}(p)$, *with* $\boldsymbol{x}_1 \in \mathbb{R}^{p_1}$, $\boldsymbol{m}_1 \in \mathbb{R}^{p_1}$ *and* $\boldsymbol{\Sigma}_{11} \in \mathrm{SPD}(p_1)$. *Then the difference of squared Mahalanobis distances can be seen as a conditional squared Mahalanobis distance assuming* $\boldsymbol{x}$ *is a realization of* $\boldsymbol{X} \sim \mathcal{N}_p(\boldsymbol{m},\boldsymbol{\Sigma})$ *given* $\boldsymbol{X}_2 = \boldsymbol{x}_2$

$$\mathrm{MD}^2(\boldsymbol{x};\boldsymbol{m},\boldsymbol{\Sigma}) - \mathrm{MD}^2(\boldsymbol{x}_2;\boldsymbol{m}_2,\boldsymbol{\Sigma}_{22}) = \mathrm{MD}^2(\boldsymbol{x}_1;\hat{\boldsymbol{x}}_1,\hat{\boldsymbol{\Sigma}}_{11}), \quad \text{(A.4)}$$

where $\hat{\boldsymbol{\Sigma}}_{11} = (\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1}$ *and* $\hat{\boldsymbol{x}}_1 = \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{x}_2$. *Additionally, if we consider random vector* $\boldsymbol{X} \sim \mathcal{N}_p(\boldsymbol{m},\boldsymbol{\Sigma})$ *instead of* $\boldsymbol{x}$ *then* $\mathrm{MD}^2(\boldsymbol{X};\boldsymbol{m},\boldsymbol{\Sigma}) - \mathrm{MD}^2(\boldsymbol{X}_2;\boldsymbol{m}_2,\boldsymbol{\Sigma}_{22}) \sim \chi^2_{p_1}$ *and is independent of* $\mathrm{MD}^2(\boldsymbol{X}_2;\boldsymbol{m}_2,\boldsymbol{\Sigma}_{22})$.

Proof. Without loss of generality assume $\boldsymbol{m} = \boldsymbol{0}$. The inverse of $\boldsymbol{\Sigma}$ can be written using an expression for the inverse of a block matrix from Petersen and Pedersen (2008):

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{A}\boldsymbol{B}\boldsymbol{A}', \text{ with}$$
$$\boldsymbol{A} = \begin{pmatrix}\boldsymbol{I} & \boldsymbol{0}\\ -\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} & \boldsymbol{I}\end{pmatrix}, \text{ and } \boldsymbol{B} = \begin{pmatrix}(\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Sigma}_{22}^{-1}\end{pmatrix} = \begin{pmatrix}\hat{\boldsymbol{\Sigma}}_{11} & 0\\ 0 & \boldsymbol{\Sigma}_{22}\end{pmatrix}. \quad \text{(A.5)}$$

The full Mahalanobis distance can then be written as

$$\mathrm{MD}^2(\boldsymbol{x};\mathbf{0},\boldsymbol{\Sigma}) = \boldsymbol{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{x} = (\boldsymbol{x}'\boldsymbol{A})\boldsymbol{B}(\boldsymbol{x}'\boldsymbol{A})'. \tag{A.6}$$

Consider that

$$\boldsymbol{x}'\boldsymbol{A} = (\boldsymbol{x}_1', \boldsymbol{x}_2')\begin{pmatrix} \boldsymbol{I} & \mathbf{0} \\ -\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} & \boldsymbol{I} \end{pmatrix} = (\boldsymbol{x}_1' - \boldsymbol{x}_2'\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}, \boldsymbol{x}_2') = \big((\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1)', \boldsymbol{x}_2'\big),$$

Expression in (A.6) can then be continued as

$$\mathrm{MD}^2(\boldsymbol{x};\mathbf{0},\boldsymbol{\Sigma}) = (\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1)'\hat{\boldsymbol{\Sigma}}_{11}^{-1}(\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1) + \boldsymbol{x}_2'\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{x}_2 = \\ \mathrm{MD}^2(\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1;\mathbf{0},\hat{\boldsymbol{\Sigma}}_{11}) + \mathrm{MD}^2(\boldsymbol{x}_2;\mathbf{0},\boldsymbol{\Sigma}_{22}). \tag{A.7}$$

Therefore the difference between the full and partial Mahalanobis distances is a conditional Mahalanobis distance itself:

$$\mathrm{MD}^2(\boldsymbol{x};\boldsymbol{m},\boldsymbol{\Sigma}) - \mathrm{MD}^2(\boldsymbol{x}_2;\boldsymbol{m},\boldsymbol{\Sigma}_{22}) = \mathrm{MD}^2(\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1;\mathbf{0},\hat{\boldsymbol{\Sigma}}_{11}) = \mathrm{MD}^2(\boldsymbol{x}_1;\hat{\boldsymbol{x}}_1,\hat{\boldsymbol{\Sigma}}_{11}). \tag{A.8}$$

This is the difference between the observed value of $\boldsymbol{x}_1$ and its conditional expectation given the rest of $\boldsymbol{x}$ using the corresponding conditional variance-covariance matrix. Note that the algebraic derivations are valid regardless of the actual distribution of $\boldsymbol{x}$ and the multivariate normality is only required for the conditional interpretation.

Being a Mahalanobis distance itself, the difference in (A.8) is distributed as $\chi^2_{p_1}$ when $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{m},\boldsymbol{\Sigma})$ is put in place of $\boldsymbol{x}$. It can most easily be shown by considering a random variable $\boldsymbol{X}_1 - \hat{\boldsymbol{X}}_1$ which (a) is multivariate normal as it is a linear combination of such; (b) has marginal expectation $\mathbf{0}$; and (c) has marginal variance-covariance matrix $\hat{\boldsymbol{\Sigma}}_{11}$. Therefore the middle term in (A.8) is indeed distributed as $\chi^2_{p_1}$.

Independence of $\mathrm{MD}^2(\boldsymbol{X}_1 - \hat{\boldsymbol{X}}_1;\mathbf{0},\hat{\boldsymbol{\Sigma}}_{11})$ and $\mathrm{MD}^2(\boldsymbol{X}_2;\boldsymbol{m},\boldsymbol{\Sigma}_{22})$ follows from the independence of $\boldsymbol{X}_1 - \hat{\boldsymbol{X}}_1$ and $\boldsymbol{X}_2$ which can be easily verified by considering their covariance matrix:

$$\begin{aligned}\mathbf{Cov}(\boldsymbol{X}_1 - \hat{\boldsymbol{X}}_1, \boldsymbol{X}_2) &= \mathbf{Cov}(\boldsymbol{X}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{X}_2, \boldsymbol{X}_2) \\ &= \mathbf{Cov}(\boldsymbol{X}_1, \boldsymbol{X}_2) - \mathbf{Cov}(\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{X}_2, \boldsymbol{X}_2) \\ &= \boldsymbol{\Sigma}_{12} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\mathbf{Cov}(\boldsymbol{X}_2, \boldsymbol{X}_2) = \boldsymbol{\Sigma}_{12} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{22} = \mathbf{0}.\end{aligned} \tag{A.9}$$

□

Appendix B

Supplementary material for chapter 3

B.1 Proof: marginal distribution of an elliptical r.v.

The following result is quite basic but we were unable to find it in the literature so we decided to include the proof.

Lemma 2. *Let $\boldsymbol{X}$ follow an elliptical p-dimensional distribution with location vector $\boldsymbol{m}$ and scatter matrix $\boldsymbol{\Sigma}$:*

$$f(\boldsymbol{x}) = \det(\boldsymbol{\Sigma})^{-\frac{1}{2}}\, h((\boldsymbol{x}-\boldsymbol{m})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m})). \tag{B.1}$$

Consider a q-dimensional subvector $\tilde{\boldsymbol{X}} = \boldsymbol{X}_{i_1,\ldots,i_q}$, for some $1 \le i_1 < \cdots < i_q \le p$. Then the marginal distribution of $\tilde{\boldsymbol{X}}$ is also elliptical with parameters that are corresponding subvector and submatrix of $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$ respectively:

$$f_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{x}}) = \det\left(\widetilde{\boldsymbol{\Sigma}}\right)^{-\frac{1}{2}} \tilde{h}((\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{m}})'\widetilde{\boldsymbol{\Sigma}}^{-1}(\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{m}})), \tag{B.2}$$

with $\tilde{\boldsymbol{m}} = \boldsymbol{m}_{i_1,\ldots,i_q}$ and $\widetilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{(i_1,\ldots,i_q):(i_1,\ldots,i_q)}$. The function $\tilde{h}(\cdot)$ only depends on $h(\cdot)$ and the dimension q, but not on $\boldsymbol{m}$ or $\boldsymbol{\Sigma}$.

Proof. Without loss of generality, we assume that $i_j = j$ for all $j = 1, \ldots, q$. Also, to simplify notation, lets assume that the location vector $\boldsymbol{m} = \boldsymbol{0}$.

Then the matrix $\boldsymbol{\Sigma}$ can be partitioned as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \widetilde{\boldsymbol{\Sigma}} & \boldsymbol{B} \\ \boldsymbol{B}' & \boldsymbol{C} \end{pmatrix}, \tag{B.3}$$

and thus its inverse is

$$\boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \widetilde{\boldsymbol{\Sigma}}^{-1} + \widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{B}\boldsymbol{M}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1} & -\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{B}\boldsymbol{M} \\ -\boldsymbol{M}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1} & \boldsymbol{M} \end{pmatrix}, \tag{B.4}$$

where $\boldsymbol{M} = (\boldsymbol{C} - \boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{B})^{-1}$.

Partition vector $\boldsymbol{x}'$ as $(\tilde{\boldsymbol{x}}', \boldsymbol{x}_m')$ and then the density of $\boldsymbol{X}$ is

$$\begin{aligned} f(\boldsymbol{x}) &= \det(\boldsymbol{\Sigma})^{-\frac{1}{2}}\, h\left((\tilde{\boldsymbol{x}}', \boldsymbol{x}_m')\begin{pmatrix} \widetilde{\boldsymbol{\Sigma}}^{-1} + \widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{B}\boldsymbol{M}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1} & -\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{B}\boldsymbol{M} \\ -\boldsymbol{M}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1} & \boldsymbol{M} \end{pmatrix}\begin{pmatrix} \tilde{\boldsymbol{x}} \\ \boldsymbol{x}_m \end{pmatrix}\right) \\ &= \det(\boldsymbol{\Sigma})^{-\frac{1}{2}}\, h\left(\tilde{\boldsymbol{x}}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}'\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{B}\boldsymbol{M}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}} - 2\boldsymbol{x}_m'\boldsymbol{M}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}} + \boldsymbol{x}_m'\boldsymbol{M}\boldsymbol{x}_m\right) \\ &= \det(\boldsymbol{\Sigma})^{-\frac{1}{2}}\, h\left(\tilde{\boldsymbol{x}}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}} + \left(\boldsymbol{M}^{\frac{1}{2}}\boldsymbol{x}_m - \boldsymbol{M}^{\frac{1}{2}}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}}\right)'\left(\boldsymbol{M}^{\frac{1}{2}}\boldsymbol{x}_m - \boldsymbol{M}^{\frac{1}{2}}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}}\right)\right). \end{aligned} \tag{B.5}$$

The marginal density for $\tilde{\boldsymbol{X}}$ is the integral of $f((\tilde{\boldsymbol{x}}', \boldsymbol{x}_m')')$ over $\boldsymbol{x}_m$, and, by changing the integration variable to $\boldsymbol{z} = \boldsymbol{M}^{\frac{1}{2}}\boldsymbol{x}_m - \boldsymbol{M}^{\frac{1}{2}}\boldsymbol{B}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}}$, we get

$$\begin{aligned} f_{\tilde{\boldsymbol{X}}}(\tilde{\boldsymbol{x}}) = \int f\left((\tilde{\boldsymbol{x}}', \boldsymbol{x}_m')'\right)\mathrm{d}\boldsymbol{x}_m &= \det(\boldsymbol{\Sigma})^{-\frac{1}{2}}\int h\left(\tilde{\boldsymbol{x}}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}} + \boldsymbol{z}'\boldsymbol{z}\right)\det(\boldsymbol{M})^{-\frac{1}{2}}\,\mathrm{d}\boldsymbol{z} \\ &= (\det(\boldsymbol{\Sigma})\times\det(\boldsymbol{M}))^{-\frac{1}{2}}\int h\left(\tilde{\boldsymbol{x}}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}} + \boldsymbol{z}'\boldsymbol{z}\right)\mathrm{d}\boldsymbol{z} \\ &= \left(\det\left(\widetilde{\boldsymbol{\Sigma}}\right)\times\det(\boldsymbol{M})^{-1}\mid\times\det(\boldsymbol{M})\right)^{-\frac{1}{2}}\int h\left(\tilde{\boldsymbol{x}}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}} + \boldsymbol{z}'\boldsymbol{z}\right)\mathrm{d}\boldsymbol{z} \\ &= \det\left(\widetilde{\boldsymbol{\Sigma}}\right)^{-1/2}\tilde{h}(\tilde{\boldsymbol{x}}'\widetilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{x}}), \end{aligned} \tag{B.6}$$

and the Lemma is proven. □

B.2 Our choice of ρ-function: Tukey's bisquare

Tukey's bisquare ρ-function, shown in equation (3.12), has been widely used in robust estimates for many years and has been an accepted standard for univariate and low-dimensional estimates. Rocke (1996) argued, however, that S-estimates of multivariate location and scatter based on Tukey's bisquare ρ-function fail to maintain practical robustness when dimension p of the data is large (e.g. $p > 10$). The reason is that the cutoff value for mahalanobis distances, above which data points are treated as outliers, dictated by the bisquare function is too large and only tremendously gross outliers are filtered out. This results in (1) larger-than-necessary bias from large but not gross outliers; (2) asymptotic efficiency as good as sample covariance, which is associated with the lack of robustness.

Dotted line in Figure B.1 shows the weights assigned to observations ($p = 20$) by Tukey's ρ-function as function of their Mahalanobis distances. Solid line on the same plot is the p.d.f. of χ^2_{20} distribution which those Mahalanobis distances follow when there is no contamination in the data. We can see that the weights remain positive even when Mahalanobis distances are so large that they are absolutely improbable under χ^2_{20} distribution. This means that outliers located $(80-20)/\sqrt{20} \approx 13$ standard deviations (in terms of

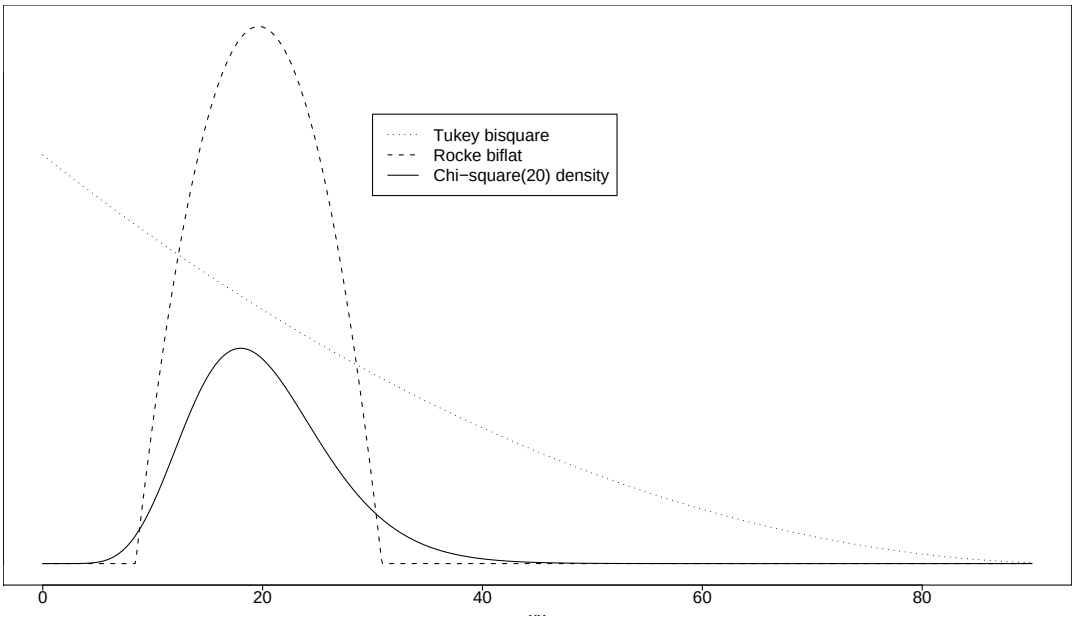


Figure B.1. Tukey's and Rocke's weight functions. Rocke's function is positive approximately where Mahalanobis distances of clean data are concentrated and zero outside.

Mahalanobis distances) from the center will still have positive weights and their chance to disturb the estimate. This outlier-blindness of Tukey's ρ-function becomes even more pronounced as dimension of the data increases.

The problem stems from the simplistic procedure used to coordinate Tukey's ρ-function between different dimensions:

$$\rho_p(d) = \rho(d/c_p), \text{ where } c_p \text{ is such that } \mathbb{E}_{\chi^2_p}[\rho(D/c_p)] = 0.5.$$

The assumption being used is that in larger dimensions Mahalanobis distances become proportionally larger and therefore all that is required to do to keep ρ-functions "consistent" across dimensions is to scale the distances accordingly. In reality, however, the distribution growth is not nearly proportional: the mean of χ^2_p is equal to p, but its standard deviation is only $\sqrt{2p}$. The distribution becomes more concentrated than what is assumed by the scaling mechanism, which results in very outlying Mahalanobis distances (in terms of actual distribution) being accepted as clean.

The solution proposed by Rocke (1996) is to recognize the fact that the shape of the distribution is changing and choose the ρ- and the weight functions accordingly. He suggests centering the weight function around the expected value of Mahalanobis distances (assuming clean data) and fixing the fraction of clean data that we are willing to downweight in order to combat outliers. The latter is being referred to as *Asymptotic Rejection Probability* (ARP) and we will denote it 2α throughout this section. Following Maronna et al.

(2006), the ρ-function can then be written as

$$\rho(d) = \begin{cases} 0 & \text{for } 0 \le d \le 1 - \gamma_p \\ \left(\frac{d-1}{4\gamma_p}\right)\left[3 - \left(\frac{d-1}{\gamma_p}\right)^2\right] + \frac{1}{2} & \text{for } 1 - \gamma_p < d < 1 + \gamma_p \\ 1 & \text{for } d \ge 1 + \gamma_p, \end{cases}$$

where $\gamma_p = \min\left(\chi^2_p(1-\alpha)/p - 1, 1\right)$. A properly scaled version of Rocke's weight function (with $\alpha = 0.05$) is shown as dashed line in Figure B.1. It is easy to see how it focuses on the same area as the χ^2_{20} distribution and aggressively rejects everything on the tails.

If the true value of the scatter matrix were known and the majority of the data were following the multivariate normal distribution then this weighting method would be unquestionably superior to the "flawed" Tukey's ρ-function.

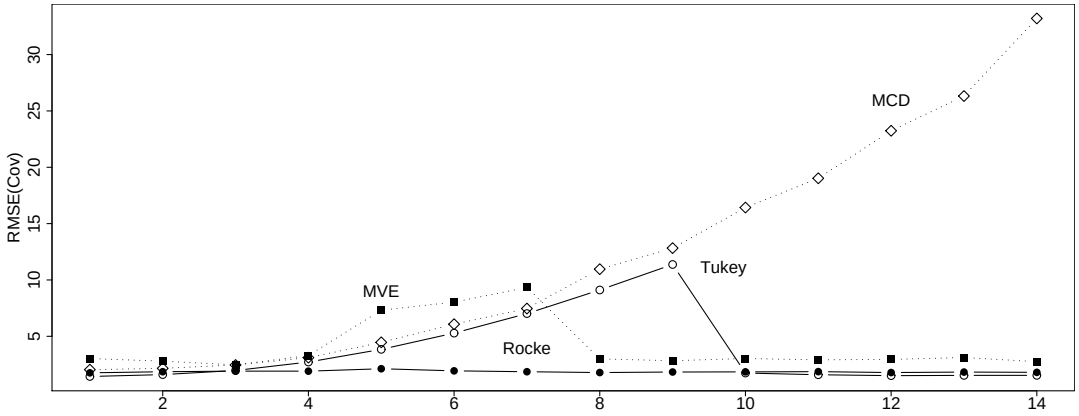
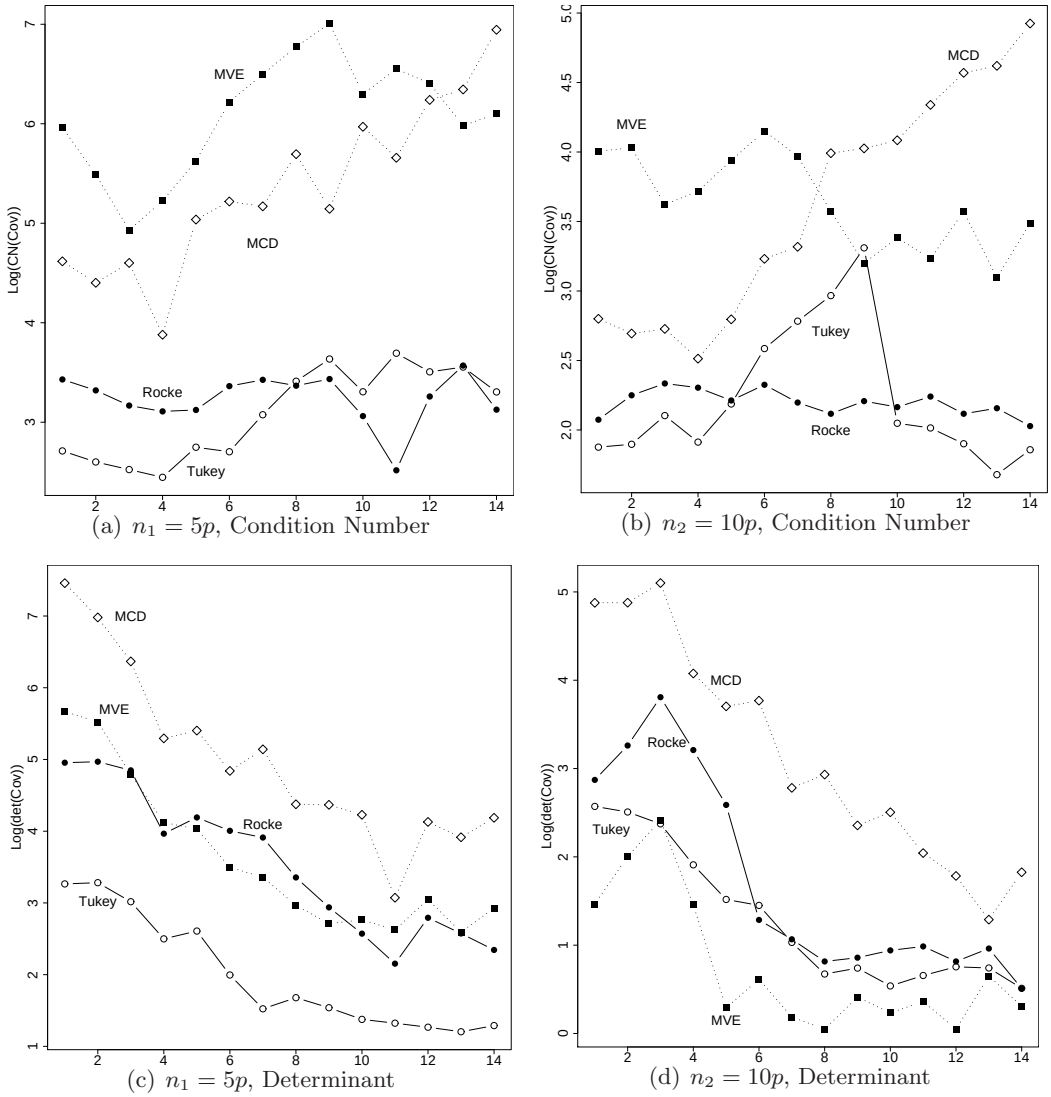


Figure B.2. RMSE of the *covariance* estimate against the location of 10% contamination. The plots closely resemble Figure 6.10 in Maronna et al. (2006) for the RMSE of the *location* estimate.

Maronna et al. (2006), in Section 6.8, conduct a small simulation study and conclude that Rocke's function should be recommended when dimension of data is greater than 10. In the study, multivariate normal data with identity covariance matrix are generated and 10% of them are replaced by point mass contamination located at $(k, 0, \ldots, 0)$, with k varying from 1 to 14. Affine-equivariance of all methods allows the authors to only consider this one covariance structure and this one specific location of contamination without loss of generality. Figure 6.10 (in the book) clearly shows that Rocke's estimate with $\alpha = 0.05$ is the obvious winner (for $p = 20$) when Root Mean Squared Error (RMSE) of the location estimate used as the performance gauge.

In this thesis we are more concerned with estimates of the scatter rather than the location, and so, in an attempt to compare performance of Rocke's and Tukey's estimates, we replicated the simulations from Maronna et al. (2006) and had a closer look at the performance of scatter matrix estimates when $p = 20$.

Figure B.2 shows the RMSE of the *scatter matrix* estimate and it is remarkably similar to the plot for RMSE of *location* shown in Maronna et al. (2006). Although it can conceivably be of interest to some researchers, RMSE is not generally considered an appropriate measure of performance for scatter matrix estimates. Its component-wise nature precludes it from evaluating how well the *shape* of the data (defined by eigenvectors and eigenvalues) has been estimated.



(a) $n_1 = 5p$, Condition Number (b) $n_2 = 10p$, Condition Number

(c) $n_1 = 5p$, Determinant (d) $n_2 = 10p$, Determinant

Figure B.3. Comparing Rocke's and Tukey's estimates of multivariate scatter. With finite sample sizes no clear winner can be named and Tukey's estimate is clearly better in terms of determinant. As sample sizes increases, and bias takes over variability, Rocke's estimate starts to appear more advantageous.

Two primary measures that we use to evaluate scatter matrix estimates in this thesis are (1) logarithm of Condition Number; (2) absolute logarithm of Determinant. If the target scatter matrix is not identity (as might be the case when dealing with non-affine

equivariant estimates, such as estimates on incomplete data) then the estimates also need to be standardized before eigenvalues are computed, but we do not need to worry about this in these simulations. Figure B.3 shows the averages (over 2,000 replications) of these two measures against k in the same way as it was done for RMSE before. We consider two sample sizes $n_1 = 5p = 100$ and $n_2 = 10p = 200$ to illustrate what happens when more (or less) data are available.

Start with the performance of the *determinant* as it is easier to characterize. For small sample size, $n_1 = 5p$, Tukey's estimate is clearly better than Rocke's for all values of k. For larger sample size, the two lines get closer and Tukey's determinant is only marginally better than Rocke's. Two immediate observations can be made: (1) determinant performance for finite sample sizes is opposite to what is suggested by RMSE; (2) as sample size increases, Rocke's estimate is getting closer to Tukey's and it is entirely possible that it will out perform it asymptotically as all sample variability disappears.

With respect to the *condition number*, it is hard to name a winner for $n_1 = 5p$, but Tukey's estimate is probably marginally better than Rocke's. For larger sample size $n_2 = 10p$, for most values of k, Tukey's still performs slightly better but has a large peak near $k = 8$ which might be considered undesirable by many. Whatever side one chooses in this competition, it can be seen from these plots that the decision to use Tukey's or Rocke's estimate on finite dataset is not as clear cut as RMSE performance would make them believe.

Our understanding is that most of the performance loss by Rocke's estimate is due to increased variability. The weight function is very selective which makes the estimate unstable. The window of positive weights is narrow and a good initial estimate is required in order to make it cover the exactly correct subset of clean data. Once the good window has been found the filtering of outliers will be very precise; but it is hard to find it in the first place.

In fact, by looking deeper into the variability of Rocke's estimate we have noticed that for the most part it is not even *sampling variability* but rather the *variability of the algorithm* itself. When repeatedly applied to exactly same dataset, the estimate yields different answers depending on the value of the random seed. If one if interested in the estimated shape of the distribution (e.g. condition number), this variability is rather significant. The implementation of Rocke's S-estimate that we used in this investigation is different from the Fast-S algorithm we use for the extended S-estimate throughout the thesis: it uses subsampling (the source of randomness) to find MVE location-scatter estimate and then proceeds with iterative reweighting from this "good" starting point. MVE is known to be robust but non-efficient and it appears that part of this inefficiency is passed to the Rocke's estimate due to its strong dependence on the starting value.

In this thesis we are faced with additional variability caused by incomplete data. To make our investigation more consistent and to be able to focus on important aspects of extending S-estimates (rather than their behaviour on complete data) we have chosen the

more computationally stable Tukey's ρ-function throughout the exposition and simulations in this thesis. The extended S-estimate can be modified to work with Rocke's ρ-function but one must be prepared to encounter computational instability when dealing with finite data; at least until a more stable method of computing S-estimates with very selective weighting is found.

B.3 Proof: MLE and constraint optimization

Lemma 3. *Maximizing the Gaussian log-likelihood of the observed data* $L(X; \boldsymbol{m}, \boldsymbol{\Sigma}) = -\frac{1}{2}L_1(X; \boldsymbol{\Sigma}) - \frac{1}{2}L_2(X; \boldsymbol{m}, \boldsymbol{\Sigma})$*, where* $L_1(X; \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \log\det\left(\boldsymbol{\Sigma}_{[i]}\right)$ *and* $L_2 = \sum_{i=1}^{n} \mathrm{MD}_i^2(X; \boldsymbol{m}, \boldsymbol{\Sigma})$*, is equivalent to minimizing* $L_2(X; \boldsymbol{m}, \boldsymbol{\Sigma})$ *under the constraint* $L_1(X; \boldsymbol{\Sigma}) = 0$ *and then scaling the covariance matrix appropriately.*

Proof. Suppose that $(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0)$ is the solution to the second problem. Let us consider any arbitrary $(\boldsymbol{m}_1, \boldsymbol{\Sigma}_1)$ and will show that $L(X; \boldsymbol{m}_1, \boldsymbol{\Sigma}_1) \leq L(X; \boldsymbol{m}_0, s_0\boldsymbol{\Sigma}_0)$ for some $s_0 > 0$.

In addition to (3.21) that states that for any $\boldsymbol{m} \in \mathbb{R}^p$, $\boldsymbol{\Sigma} \in \mathrm{SPD}(p)$ and $a > 0$,

$$L_1(X; a\boldsymbol{\Sigma}) = \log(a) \sum_{i=1}^{n} p_i + L_1(X; \boldsymbol{\Sigma}), \tag{B.7}$$

let us note that L_2 is also scale equivariant in a sense that

$$L_2(X; \boldsymbol{m}, a\boldsymbol{\Sigma}) = a^{-1} L_2(X; \boldsymbol{m}, \boldsymbol{\Sigma}). \tag{B.8}$$

Let a_1 be such that $L_1(X; a_1\boldsymbol{\Sigma}_1) = 0$ according to (3.22). Then $L_2(X; \boldsymbol{m}_1, a_1\boldsymbol{\Sigma}_1) \geq L_2(X; \boldsymbol{m}_0, \boldsymbol{\Sigma}_0)$ because $a_1\boldsymbol{\Sigma}_1$ satisfies the constraint $L_1(a_1\boldsymbol{\Sigma}_1) = 0$ and $\boldsymbol{\Sigma}_0$ is known to minimize $L_2(\cdot)$ among such matrices. Therefore, using (B.8) twice, we have

$$L_2(X; \boldsymbol{m}_1, \boldsymbol{\Sigma}_1) = a_1 L_2(X; \boldsymbol{m}_1, a_1\boldsymbol{\Sigma}_1) \geq a_1 L_2(X; \boldsymbol{m}_0, \boldsymbol{\Sigma}_0) = L_2(X; \boldsymbol{m}_0, a_1^{-1}\boldsymbol{\Sigma}_0). \tag{B.9}$$

Additionally, we can see that $L_1(X; \boldsymbol{\Sigma}_1) = L_1(X; a_1^{-1}\boldsymbol{\Sigma}_0)$ because both of them are equal to $-\log(a_1)\sum_{i=1}^{n} p_i$ according to (B.7) and because $L_1(X; a_1\boldsymbol{\Sigma}_1) = 0$.

Putting these two facts together and recalling that $L = -(L_1 + L_2)/2$ we get

$$L(X; \boldsymbol{m}_1, \boldsymbol{\Sigma}_1) \leq L(X; \boldsymbol{m}_0, a_1^{-1}\boldsymbol{\Sigma}_0),$$

which proves that $(\boldsymbol{m}_1, \boldsymbol{\Sigma}_1)$ does not maximize $L(\cdot)$. □

B.4 Fisher-consistency and the choice of k_i

To discuss the Fisher-consistency of our estimate we must first define its asymptotic functional version. At this point we need to pay closer attention to the mechanism generating

missing data. We will assume that missingness is completely independent from the actual data (both observed and missing) and that various missingness patterns have well-defined probabilities of occurring. The distribution of our data can be seen as a mixture of distributions on different coordinate subspaces of $\mathbb{R}^p$. It can be fully described by F, the distribution of full data on $\mathbb{R}^p$, and a collection of probabilities $\pi_{\varkappa}$ of various missingness patterns enumerated by $\varkappa = \{\text{observed}, \text{missing}\}^p$. We will denote this set $\{\text{observed}, \text{missing}\}^p$ of all possible missingness patterns as $\mathcal{P}$.

For each $\varkappa \in \mathcal{P}$, we define $|\varkappa|$ to be the number of observed variables in $\varkappa$, and for any vector $\boldsymbol{m}$ and matrix $\boldsymbol{\Sigma}$, let $\boldsymbol{m}_{[\varkappa]}$ and $\boldsymbol{\Sigma}_{[\varkappa]}$ be the subvector of $\boldsymbol{m}$ and submatrix of $\boldsymbol{\Sigma}$ corresponding to the observed components of $\varkappa$.

We define the population version of the extended S-estimate analogously to its finite sample version presented in Definition 1.

Definition 2. *The* asymptotic *extended S-estimate of multivariate location and scatter is defined as* $(\hat{\boldsymbol{m}}, \hat{s}\hat{\boldsymbol{\Sigma}})$ *where* $(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\Sigma}})$ *minimizes the scale* $s(\boldsymbol{m}, \boldsymbol{\Sigma})$*, which is the solution of*

$$\sum_{\varkappa \in \mathcal{P}} \pi_{\varkappa} \mathbb{E}_F \left[\tilde{\rho}_{|\varkappa|} \left(\frac{\mathrm{MD}^2(\boldsymbol{X}_{[\varkappa]}; \boldsymbol{m}_{[\varkappa]}, \boldsymbol{\Sigma}_{[\varkappa]})}{s(\boldsymbol{m}, \boldsymbol{\Sigma})} \right) \right] = b, \tag{B.10}$$

subject to

$$\sum_{\varkappa \in \mathcal{P}} \pi_{\varkappa} \log \left(\det \left(\boldsymbol{\Sigma}_{[\varkappa]} \right) \right) = 0. \tag{B.11}$$

The function $\tilde{\rho}_{|\varkappa|}(\cdot)$ *depends on the dimension of* $\varkappa$

$$\tilde{\rho}_{|\varkappa|}(d) = \frac{c_{|\varkappa|} k_{|\varkappa|}}{\sum_{\varkappa^*} \pi_{\varkappa^*} c_{|\varkappa^*|} k_{|\varkappa^*|}} \rho_{c_{|\varkappa|}}(d), \tag{B.12}$$

with constants $c_{|\varkappa|}$ *and* $k_{|\varkappa|}$ *such that*

$$\mathbb{E}\Big[\rho_{c_{|\varkappa|}} \left(X_1^2 + \cdots + X_{|\varkappa|}^2 \right) \Big] = b, \text{ and} \tag{B.13}$$

$$k_{|\varkappa|} = \Big\{ \mathbb{E}\Big[c_{|\varkappa|} \rho'_{c_{|\varkappa|}} \left(X_1^2 + \cdots + X_{|\varkappa|}^2 \right) X_1^2 \Big] \Big\}^{-1}. \tag{B.14}$$

Expectations in (B.13) and (B.14) are computed over $X_j \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$*, for* $j = 1, \ldots, p$*.*

Proof of Theorem 1. Let the distribution F of the complete data be multivariate normal with mean $\boldsymbol{m}_0$ and covariance $\boldsymbol{\Sigma}_0$. Then we will show that the scale $s(\boldsymbol{m}, \boldsymbol{\Sigma})$ is locally minimized by $(\boldsymbol{m}_0, a(\boldsymbol{\Sigma}_0)\boldsymbol{\Sigma}_0)$ where $a(\boldsymbol{\Sigma}_0)$ is a standardizing constant computed in (3.22) to make $a(\boldsymbol{\Sigma}_0)\boldsymbol{\Sigma}_0$ satisfy constraint (B.11). Denote $a_0 = a(\boldsymbol{\Sigma}_0)$.

The extended S-estimate is location equivariant and therefore we can assume without loss of generality that $\boldsymbol{m}_0 = \boldsymbol{0}$.

Since the condition in (B.11) is solely multiplicative, we can see the constrained optimization problem in Definiton 2 from a different, unconstrained, point of view. For

any $\boldsymbol{\Sigma} \in \mathrm{SPD}(p)$ and $\boldsymbol{m} \in \mathbb{R}^p$, we can compute the scale $s^*(\boldsymbol{m}, \boldsymbol{\Sigma}) = s(\boldsymbol{m}, a(\boldsymbol{\Sigma})\boldsymbol{\Sigma})$ and attempt to minimize it.

It is easy to see that $s_T^* = s(\mathbf{0}, a_0\boldsymbol{\Sigma}_0) = a_0^{-1}$. Mahalanobis distances $\mathrm{MD}^2(X_{[\varkappa]}; \mathbf{0}_{[\varkappa]}, a_0\boldsymbol{\Sigma}_{0[\varkappa]})$ are distributed as $a_0^{-1}\chi^2_{|\varkappa|}$ and therefore if $s(\mathbf{0}, a_0\boldsymbol{\Sigma}_0) > a_0^{-1}$ then all individual expectation in (B.10) are smaller than $b\frac{c_{|\varkappa|}k_{|\varkappa|}}{\sum_{\varkappa^*}\pi_{\varkappa^*}c_{|\varkappa^*|}k_{|\varkappa^*|}}$ because constants $c_{|\varkappa|}$ are tuned to satisfy (B.13). Thus the summation in (B.10) is smaller than b. Analogously, if $s(\mathbf{0}, a_0\boldsymbol{\Sigma}_0) < a_0^{-1}$ than the summation is larger than b. Therefore they must be equal.

Also, note the following. Suppose we have two pairs of candidate estimates $(\boldsymbol{m}_1, \boldsymbol{\Sigma}_1)$ and $(\boldsymbol{m}_2, \boldsymbol{\Sigma}_2)$ that generate scales s_1^* and s_2^* respectively. Denote the left hand side of the expression in (B.10) by $G(\boldsymbol{m}, \boldsymbol{\Sigma}, s)$. If $G(\boldsymbol{m}_1, a(\boldsymbol{\Sigma}_1)\boldsymbol{\Sigma}_1, s_2^*) \geq b = G(\boldsymbol{m}_1, a(\boldsymbol{\Sigma}_1)\boldsymbol{\Sigma}_1, s_1^*)$ then $s_1^* \geq s_2^*$ because G is a monotonely non-decreasing function of s. Therefore, if we could prove that $H(\boldsymbol{m}, \boldsymbol{\Sigma}) = G(\boldsymbol{m}, a(\boldsymbol{\Sigma})\boldsymbol{\Sigma}, s_T^*)$ is minimized by $(\mathbf{0}, \boldsymbol{\Sigma}_0)$, at which it is equal to b, we would have proven that the scale $s^*(\boldsymbol{m}, \boldsymbol{\Sigma})$ is also minimized by that same pair.

We will take the derivatives of H w.r.t. $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$ and verify that they are equal to zero at $(\mathbf{0}, \boldsymbol{\Sigma}_0)$.

$$\begin{aligned}\frac{\partial H}{\partial \boldsymbol{m}}(\mathbf{0}, \boldsymbol{\Sigma}_0) = \frac{\partial G}{\partial \boldsymbol{m}}(\mathbf{0}, a(\boldsymbol{\Sigma}_0)\boldsymbol{\Sigma}_0, s_T^*) &= \sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \left\langle \mathbb{E}\left[\tilde{\rho}'_{|\varkappa|}(\mathrm{MD}^2/s_T^*)\frac{-2\boldsymbol{\Sigma}_{0[\varkappa]}^{-1}\boldsymbol{X}_{[\varkappa]}}{s_T^*}\right]\right\rangle \\ &= -\frac{2}{s_T^*}\sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \left\langle \boldsymbol{\Sigma}_{0[\varkappa]}^{-1}\mathbb{E}\left[\tilde{\rho}'_{|\varkappa|}(\mathrm{MD}^2/s_T^*)\boldsymbol{X}_{[\varkappa]}\right]\right\rangle = \mathbf{0},\end{aligned}$$

because the distribution of $\boldsymbol{X}$ is symmetric around $\boldsymbol{m}_0 = \mathbf{0}$ and therefore the expectations are equal to zero. The angle brackets $\langle\cdot\rangle$ in the expression above mean that the $|\varkappa|$-vectors need to be expanded to p-vectors by filling the empty cells with zeros.

The derivative w.r.t. $\boldsymbol{\Sigma}$ requires slightly more work. Before we proceed with our proof let us recall two useful expressions (number 48 and 52, respectively) from Petersen and Pedersen (2008) for future use:

$$\frac{\partial \log|\det(\boldsymbol{M})|}{\partial \boldsymbol{M}} = (\boldsymbol{M}^{-1})' = (\boldsymbol{M}')^{-1}, \tag{B.15}$$

$$\frac{\partial\, \boldsymbol{a}'\boldsymbol{M}^{-1}\boldsymbol{b}}{\partial \boldsymbol{M}} = -\boldsymbol{M}^{-1}\boldsymbol{a}\boldsymbol{b}'\boldsymbol{M}^{-1}, \tag{B.16}$$

where $\boldsymbol{M}$ is a square matrix, while $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors of the same dimension.

Now note that

$$\frac{\partial H}{\partial \boldsymbol{\Sigma}}(\mathbf{0}, \boldsymbol{\Sigma}_0) = \frac{\partial G}{\partial \boldsymbol{\Sigma}}(\mathbf{0}, a_0\boldsymbol{\Sigma}_0, s_T^*)\left(\frac{\partial a}{\partial \boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_0)\boldsymbol{\Sigma}_0 + a_0 I_p\right). \tag{B.17}$$

Then denote the left hand side of the constraint (B.11) as $F(\boldsymbol{\Sigma})$. Normalizing constant $a(\boldsymbol{\Sigma})$ is chosen such that $F(a(\boldsymbol{\Sigma})\boldsymbol{\Sigma}) = 0$ for all $\boldsymbol{\Sigma}$. Therefore the derivative of it w.r.t. $\boldsymbol{\Sigma}$

is also equal to zero:

$$\frac{\partial F(a(\boldsymbol{\Sigma})\boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_0) = \frac{\partial F}{\partial \boldsymbol{\Sigma}}(a_0\boldsymbol{\Sigma}_0)\left\{\frac{\partial a}{\partial \boldsymbol{\Sigma}}(\boldsymbol{\Sigma}_0)\boldsymbol{\Sigma}_0 + a_0 I_p\right\} = \mathbf{0}.$$

Note that both terms in the product above are matrices so it is not necessary for one of them to be equal to zero in order for the product to be zero. If we can show that $\frac{\partial G}{\partial \boldsymbol{\Sigma}}(\mathbf{0}, a_0\boldsymbol{\Sigma}_0, s_T^*)$ in (B.17) is proportional (as matrix) to $\frac{\partial F}{\partial \boldsymbol{\Sigma}}(a_0\boldsymbol{\Sigma}_0)$ above then we would have proven that the left hand side in (B.17) is equal to zero because the second terms in both expressions are identical. Now have a closer look at them:

$$\frac{\partial F}{\partial \boldsymbol{\Sigma}}(a_0\boldsymbol{\Sigma}_0) = \sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \frac{\partial \log\det\left(\boldsymbol{\Sigma}_{[\varkappa]}\right)}{\partial \boldsymbol{\Sigma}}(a_0\boldsymbol{\Sigma}_0) = a_0^{-1}\sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \left\langle \boldsymbol{\Sigma}_{0[\varkappa]}^{-1}\right\rangle, \tag{B.18}$$

where we used expression (B.15). Angle brackets now mean that the $|\varkappa|\times|\varkappa|$ matrices need to be expanded to size $p\times p$ the same way as it was done with vectors above. Using (B.16), we obtain

$$\begin{aligned}
&\frac{\partial G}{\partial \boldsymbol{\Sigma}}(\mathbf{0}, a_0\boldsymbol{\Sigma}_0, s_T^*) \\
&\quad= \sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \mathbb{E}\left[\tilde{\rho}'_{|\varkappa|}\left(\frac{\mathrm{MD}^2(\boldsymbol{X}_{[\varkappa]};\mathbf{0}_{[\varkappa]}, a_0\boldsymbol{\Sigma}_{0[\varkappa]})}{s_T^*}\right)\frac{\partial \mathrm{MD}^2(\boldsymbol{X}_{[\varkappa]};\boldsymbol{m}_{[\varkappa]},\boldsymbol{\Sigma}_{[\varkappa]})}{\partial \boldsymbol{\Sigma}}(\mathbf{0}, a_0\boldsymbol{\Sigma}_0)\frac{1}{s_T^*}\right] \\
&\quad= \frac{1}{s_T^*}\sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \mathbb{E}\left[\tilde{\rho}'_{|\varkappa|}\left(\frac{a_0^{-1}\mathrm{MD}^2(\boldsymbol{X}_{[\varkappa]};\mathbf{0}_{[\varkappa]},\boldsymbol{\Sigma}_{0[\varkappa]})}{a_0^{-1}}\right)\left\langle a_0^{-2}\boldsymbol{\Sigma}_{0[\varkappa]}^{-1}(\boldsymbol{X}_{[\varkappa]}\boldsymbol{X}'_{[\varkappa]})\boldsymbol{\Sigma}_{0[\varkappa]}^{-1}\right\rangle\right] \\
&= \frac{1}{(s_T^*)^3}\sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \left\langle \boldsymbol{\Sigma}_{0[\varkappa]}^{-1/2}\mathbb{E}\left[\tilde{\rho}'_{|\varkappa|}\left(\boldsymbol{X}'_{[\varkappa]}\boldsymbol{\Sigma}_{0[\varkappa]}^{-1}\boldsymbol{X}_{[\varkappa]}\right)\right)(\boldsymbol{\Sigma}_{0[\varkappa]}^{-1/2}\boldsymbol{X}_{[\varkappa]})(\boldsymbol{\Sigma}_{0[\varkappa]}^{-1/2}\boldsymbol{X}_{[\varkappa]})'\right]\boldsymbol{\Sigma}_{0[\varkappa]}^{-1/2}\right\rangle \\
&\qquad= \frac{1}{(s_T^*)^3}\sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \left\langle \boldsymbol{\Sigma}_{0[\varkappa]}^{-1/2}(kI_{|\varkappa|})\boldsymbol{\Sigma}_{0[\varkappa]}^{-1/2}\right\rangle = \frac{k}{(s_T^*)^3}\sum_{\varkappa\in\mathcal{P}} \pi_\varkappa \left\langle \boldsymbol{\Sigma}_{0[\varkappa]}^{-1}\right\rangle,
\end{aligned} \tag{B.19}$$

where $k = (\sum_{\varkappa\in\mathcal{P}} \pi_\varkappa c_{|\varkappa|}k_{|\varkappa|})^{-1}$ as in (B.12). The most important part of the derivation above is the fact that all expectations (for all $\varkappa$) are equal to $kI_{|\varkappa|}$. They have different dimensions but all diagonal element are the same. All expectation matrices are diagonal simply because $\boldsymbol{\Sigma}_{0[\varkappa]}^{-1/2}\boldsymbol{X}_{[\varkappa]}$ has a spherical distribution and therefore all off-diagonal cross-products are equal to zero due to symmetry. Diagonal elements, however, are equalized due to the specific choice of constants $k_{|\varkappa|}$ in (B.14). This is the most important reason for taking k_i in (3.19) the way we did — to counterbalance the discrepancies in these expectations across different dimensions.

Comparing (B.19) and (B.18) we can see that they are indeed proportional to each other and therefore the gradient $\frac{\partial H}{\partial \boldsymbol{\Sigma}}(\mathbf{0}, \boldsymbol{\Sigma}_0)$ is equal to zero. □

Remark 1. Note that the proof only uses the fact that F is multivariate normal to compute expectations in (B.13) and (B.14). Everything else will hold for any elliptical

distribution with scatter matrix $\boldsymbol{\Sigma}_0$ and location $\boldsymbol{m}_0$. Therefore, if the data come from another elliptical family, then Definition 2 and, accordingly, Definition 1 can be modified so that the estimate is Fisher-consistent under that family. The only required change is to replace the distribution in the expressions defining $c_{|\varkappa|}$ and $k_{|\varkappa|}$ (or c_i and k_i in terms of the definition for finite samples).

Remark 2. In order to have Fisher-consistency, probabilities $\{\pi_\varkappa\}$ must be such that, for each pair of variables, the probability of observing them together is greater than zero. Otherwise the element of the covariance matrix that corresponds to the completely missing pair is unidentifiable. The same is true for the finite sample data. If there is a pair of variables that is never observed together then the corresponding element of the covariance matrix cannot be estimated.

B.5 Proof of theorem 2

Proof. Let Y be the transformed version of X such that $\boldsymbol{y}_i = \boldsymbol{D}\boldsymbol{x}_i + \boldsymbol{b}$, where D is a non-singular diagonal matrix. Let $(\boldsymbol{m}_x, s_x\boldsymbol{\Sigma}_x)$ be the estimate based on the original x-data and then we will show that $(\boldsymbol{D}\boldsymbol{m}_x + \boldsymbol{b}, \boldsymbol{D}(s_x\boldsymbol{\Sigma}_x)\boldsymbol{D}')$ is the estimate based on the y-data.

First, look at the dimension-specific Mahalanobis distances in (3.14). Since $\boldsymbol{D}$ is diagonal, missing values do not propagate and, since all factors are different from zeros, they do not disappear under the transformation — missingness pattern remains the same. Note that $\boldsymbol{y}_i^{\text{obs}} = \boldsymbol{D}_{[i]}\boldsymbol{x}_i^{\text{obs}} + \boldsymbol{b}_{[i]}$ and therefore, for any p-vector $\boldsymbol{m}$, $p \times p$ matrix $\boldsymbol{\Sigma}$ and $a \in \mathbb{R}$,

$$\text{MD}_i^2(\boldsymbol{y}_i^{\text{obs}}; \boldsymbol{D}\boldsymbol{m} + \boldsymbol{b}, a\boldsymbol{D}\boldsymbol{\Sigma}\boldsymbol{D}') = \frac{1}{a}\text{MD}_i^2(\boldsymbol{x}_i^{\text{obs}}; \boldsymbol{m}, \boldsymbol{\Sigma}). \tag{B.20}$$

Consequently, the scale s, the solution to (3.14) for a given $(\boldsymbol{m}, \boldsymbol{\Sigma})$, is such that

$$s_y(\boldsymbol{D}\boldsymbol{m} + \boldsymbol{b}, a\boldsymbol{D}\boldsymbol{\Sigma}\boldsymbol{D}') = \frac{1}{a}s_x(\boldsymbol{m}, \boldsymbol{\Sigma}). \tag{B.21}$$

Let us define

$$a_D = \exp\left(-\frac{2\sum_{i=1}^n \log(\det(\boldsymbol{D}_{[i]}))}{\sum_{i=1}^n p_i}\right), \tag{B.22}$$

such that constraint (3.15) is satisfied simultaneously for $\boldsymbol{\Sigma}$ and $a_D\boldsymbol{D}\boldsymbol{\Sigma}\boldsymbol{D}'$. This concludes the part of the proof that is specific to the extended version of the estimate that we defined in this paper — the rest is generic and would work for any non-singular $\boldsymbol{D}$ if there were no missing values in the data.

Consider arbitrary pair $(\tilde{\boldsymbol{m}}, \widetilde{\boldsymbol{\Sigma}})$ satisfying constraint (3.15) and look at the scale they

generate if used with the y-data:

$$s_y(\tilde{\boldsymbol{m}}, \widetilde{\boldsymbol{\Sigma}}) = \frac{1}{a_D} s_x(\boldsymbol{D}^{-1}(\tilde{\boldsymbol{m}} - \boldsymbol{b}), \frac{1}{a_D}\boldsymbol{D}^{-1}\widetilde{\boldsymbol{\Sigma}}(\boldsymbol{D}')^{-1}) \\ \geq \frac{1}{a_D} s_x(\boldsymbol{m}_x, \boldsymbol{\Sigma}_x) = s_y(\boldsymbol{D}\boldsymbol{m}_x + \boldsymbol{b}, a_D\boldsymbol{D}\boldsymbol{\Sigma}_x\boldsymbol{D}), \quad \text{(B.23)}$$

which proves that $s_y()$ achieves its minimum at $(\boldsymbol{D}\boldsymbol{m}_x + \boldsymbol{b}, a_D\boldsymbol{D}\boldsymbol{\Sigma}_x\boldsymbol{D})$. Therefore the extended S-estimate computed on y-data is

$$(\boldsymbol{D}\boldsymbol{m}_x + \boldsymbol{b}, \frac{1}{a_D} s_x(\boldsymbol{m}_x, \boldsymbol{\Sigma}_x) a_D\boldsymbol{D}\boldsymbol{\Sigma}\boldsymbol{D}) = (\boldsymbol{D}\boldsymbol{m}_x + \boldsymbol{b}, s_x\boldsymbol{D}\boldsymbol{\Sigma}_x\boldsymbol{D}). \quad \text{(B.24)}$$

□