Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uasa20

The Cellwise Minimum Covariance Determinant Estimator

Jakob Raymaekers & Peter J. Rousseeuw

To cite this article: Jakob Raymaekers & Peter J. Rousseeuw (22 Nov 2023): The Cellwise Minimum Covariance Determinant Estimator, Journal of the American Statistical Association, DOI: 10.1080/01621459.2023.2267777

To link to this article: https://doi.org/10.1080/01621459.2023.2267777

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2023, VOL. 00, NO. 0, 1–12: Theory and Methods
https://doi.org/10.1080/01621459.2023.2267777

The Cellwise Minimum Covariance Determinant Estimator

Jakob Raymaekers[a] and Peter J. Rousseeuw[b]

[a]Department of Quantitative Economics, Maastricht University, Maastricht, The Netherlands; [b]Section of Statistics and Data Science, University of Leuven, Leuven, Belgium

ABSTRACT

The usual Minimum Covariance Determinant (MCD) estimator of a covariance matrix is robust against casewise outliers. These are cases (that is, rows of the data matrix) that behave differently from the majority of cases, raising suspicion that they might belong to a different population. On the other hand, cellwise outliers are individual cells in the data matrix. When a row contains one or more outlying cells, the other cells in the same row still contain useful information that we wish to preserve. We propose a cellwise robust version of the MCD method, called cellMCD. Its main building blocks are observed likelihood and a penalty term on the number of flagged cellwise outliers. It possesses good breakdown properties. We construct a fast algorithm for cellMCD based on concentration steps (C-steps) that always lower the objective. The method performs well in simulations with cellwise outliers, and has high finite-sample efficiency on clean data. It is illustrated on real data with visualizations of the results. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received July 2022
Accepted September 2023

KEYWORDS
Cellwise outliers; Covariance matrix; Likelihood; Missing values

1. Motivation

Any practicing statistician or data scientist knows that real datasets often contain outliers. One definition of outliers says that they are cases that do not obey the fit suggested by the majority of the data, which raises suspicion that they may have been generated by a different mechanism. Since cases typically correspond to rows of the data matrix, they are sometimes called rowwise outliers. They may be the result of gross errors, but they can also be nuggets of valuable information. In either case, it is important to find them. In computer science this is called anomaly detection, and in some areas it is known as exception mining. In statistics several approaches were tried, such as testing for outliers and the computation of outlier diagnostics. In our experience the approach working best is that of robust statistics, which aims to fit the majority of the data first, and then flags outliers by their large deviation from that fit.

In this article we focus on single-class multivariate numerical data without a response variable (although the results are relevant for classification and regression too). The goal is to robustly estimate the central location of the point cloud as well as its covariance matrix, and at the same time flag the outliers that may be present. The underlying model is that the data come from a multivariate Gaussian distribution, after which some data has been replaced by outliers that can be anywhere.

The Minimum Covariance Determinant (MCD) estimator introduced by Rousseeuw (1984, 1985) is highly robust to casewise outliers. Its definition is quite intuitive. Take an integer h that is at least half the sample size n. We then look for the subset containing h cases such that the determinant of its usual covariance matrix is as small as possible. The resulting robust location estimate is then the mean of that subset, and the robust covariance matrix is its covariance matrix multiplied by a consistency factor. One can show that the estimates are not overly affected when there are fewer than $n - h$ outlying cases. The MCD became computationally feasible with the algorithm of Rousseeuw and Van Driessen (1999), followed by even faster algorithms by Hubert, Rousseeuw, and Verdonck (2012) and De Ketelaere et al. (2020). Copt and Victoria-Feser (2004) computed the MCD for incomplete data. The MCD has also been generalized to high dimensions (Boudt et al. 2020), and to non-elliptical distributions using kernels (Schreurs et al. 2021). For a survey on the MCD and its applications see Hubert, Debruyne, and Rousseeuw (2018). The MCD is available in the procedure ROBUSTREG in SAS, in SAS/IML, in Matlab's PLS Toolbox, and in the R packages *robustbase* (Maechler et al. 2022) and *rrcov* (Todorov 2012) on CRAN. In Python one can use `MinCovDet` in `scikit-learn` (Pedregosa et al. 2011).

In recent times a different outlier paradigm has gained prominence, that of *cellwise outliers*, first published by Alqallaf et al. (2009). It assumes that the data were generated from a certain distributional model, after which some individual cells (entries) were replaced by other values. The difference between the casewise and the cellwise paradigm is illustrated in Figure 1. In the left panel the outlying cases are shown as black rows. In the panel on the right the cellwise outliers correspond to fewer black squares in total, but together they contaminate over half of the cases, so the existing methods for casewise outliers may fail.

In reality we do not know in advance *which* cells in the right panel of Figure 1 are outlying (black), unlike the simpler problem of incomplete data where we do know which cells are missing. When the variables have substantial correlations, the

CONTACT Peter J. Rousseeuw *peter@rousseeuw.net* Section of Statistics and Data Science, University of Leuven, Leuven, Belgium.
Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/JASA*.

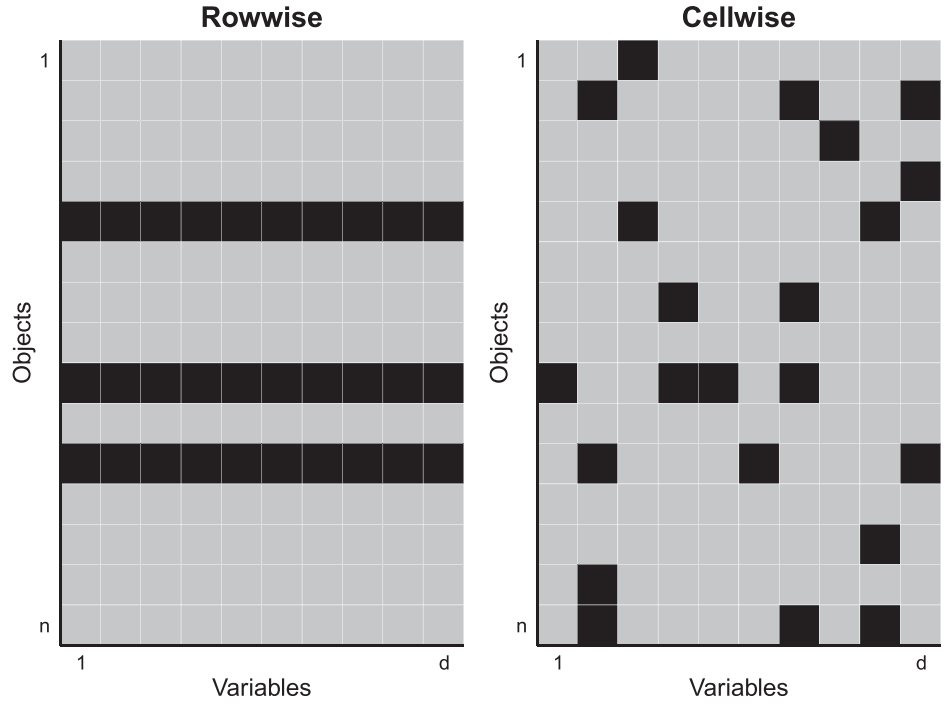

Figure 1. The casewise (left) and cellwise (right) outlier paradigms. (Black means outlying.)

cellwise outliers need not be marginally outlying, and then it can be quite hard to detect them. Van Aelst, Vandervieren, and Willems (2011) proposed one of the first detection methods. Rousseeuw and Van den Bossche (2018) predict the values of all cells and flag the cells that differ much from their prediction.

There has been some work on estimating the underlying covariance matrix in the presence of cellwise outliers. One approach is to compute robust covariances between each pair of variables, and to assemble them in a matrix. To estimate these pairwise covariances, Öllerer and Croux (2015) and Croux and Öllerer (2016) use rank correlations. Tarr, Muller, and Weber (2016) instead use the robust pairwise correlation estimator of Gnanadesikan and Kettenring (1972) in combination with the robust scale estimator Q_n of Rousseeuw and Croux (1993). As the resulting matrix is not necessarily positive semidefinite (PSD), they then compute the nearest PSD matrix by the method of Higham (2002). Raymaekers and Rousseeuw (2021a) obtain a PSD covariance matrix by transforming ("wrapping") the original data variables.

Many cellwise robust methods were developed for settings such as principal components (Hubert, Rousseeuw, and Van den Bossche 2019), discriminant analysis (Aerts and Wilms 2017), clustering (García-Escudero et al. 2021), graphical models (Katayama, Fujisawa, and Drton 2018), low-rank approximation (Maronna and Yohai 2008), regression (Öllerer, Alfons, and Croux 2016; Filzmoser et al. 2020), and variable selection (Su, Tarr, and Muller 2021). Also, isolated outliers in functional data (Hubert, Rousseeuw, and Segaert 2015) can be seen as cellwise outliers.

In the next section we introduce the cellwise MCD estimator. It is the first method with a single objective that combines detection and estimation, unlike some existing methods which do detection and estimation separately. Because of this cellMCD has provable cellwise breakdown properties, see Section 3. There we also derive its consistency. Section 4 describes its algorithm, and proves that it converges. It is faster than the earlier methods. Some illustrations on real data are shown in Section 5. The performance of the method is studied by simulation in Section 6, indicating that it is very robust against adversarial contamination. Section 7 concludes with a discussion.

2. A Cellwise MCD

We first note that the *casewise* MCD can be reformulated in terms of likelihood. The likelihood of a d-variate Gaussian distribution is

$$f(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\mathrm{MD}^2(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})/2} \tag{1}$$

where $\boldsymbol{\mu}$ is a column vector, $\boldsymbol{\Sigma}$ is a positive definite matrix, and the Mahalanobis distance is $\mathrm{MD}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$. For a sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ we put $L(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) := -2\ln(f(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}))$ so the maximum likelihood estimator (MLE) of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ minimizes

$$\sum_{i=1}^{n} L(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \left(\ln|\boldsymbol{\Sigma}| + d\ln(2\pi) + \mathrm{MD}^2(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \right). \tag{2}$$

Let us now look for a subset $H \subset \{1, \ldots, n\}$ with h elements which minimizes (2) where the sum is only over i in H. We can also write this with weights w_i that are 0 or 1 in the objective $\sum_{i=1}^{n} w_i L(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, so we minimize

$$\sum_{i=1}^{n} w_i \left(\ln|\boldsymbol{\Sigma}| + d\ln(2\pi) + \mathrm{MD}^2(\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \right) \tag{3}$$

$$\text{under the constraint that } \sum_{i=1}^{n} w_i = h\,.$$

For the minimizing set of weights w_i we know from maximum likelihood that $\widehat{\boldsymbol{\mu}}$ is the mean of the $\boldsymbol{x}_i$ in H, so it is the weighted mean of all $\boldsymbol{x}_i$, and similarly

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{h}\sum_{i=1}^{n} w_i(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^\top . \tag{4}$$

But then the third term of (3) becomes

$$\sum_{i=1}^{n} w_i(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^\top \widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})$$

$$= \sum_{i=1}^{n} \text{trace}\left(w_i(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^\top \widehat{\boldsymbol{\Sigma}}^{-1}\right) =$$

$$\text{trace}\left(\sum_{i=1}^{n} w_i(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^\top \widehat{\boldsymbol{\Sigma}}^{-1}\right) = \text{trace}\left(h\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\Sigma}}^{-1}\right) = hd$$

which is constant, and so is the second term. Therefore, minimizing (3) is equivalent to minimizing the determinant of (4), which is the definition of the casewise MCD.

In the context of incomplete data, Dempster, Laird, and Rubin (1977) and others defined the *observed likelihood*. Let us denote the missingness pattern of the $n \times d$ data matrix $\boldsymbol{X}$ by the $n \times d$ matrix $\boldsymbol{W}$ with entries w_{ij} that are 0 for missing x_{ij} and 1 otherwise. Its rows $\boldsymbol{w}_i$ take the place of the scalar weights w_i in (3). For the Gaussian model the observed likelihood of the ith observation (Little and Rubin 2020) is given by

$$f(\boldsymbol{x}_i^{(\boldsymbol{w}_i)}, \boldsymbol{\mu}^{(\boldsymbol{w}_i)}, \boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}) := \frac{1}{(2\pi)^{d^{(\boldsymbol{w}_i)}/2}|\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}|^{1/2}} e^{-\text{MD}^2(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})/2} \tag{5}$$

in which

$$\text{MD}(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) := \sqrt{(\boldsymbol{x}_i^{(\boldsymbol{w}_i)} - \boldsymbol{\mu}^{(\boldsymbol{w}_i)})^\top (\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)})^{-1}(\boldsymbol{x}_i^{(\boldsymbol{w}_i)} - \boldsymbol{\mu}^{(\boldsymbol{w}_i)})} \tag{6}$$

is called the *partial Mahalanobis distance* by Danilov, Yohai, and Zamar (2012). Here $\boldsymbol{x}_i^{(\boldsymbol{w}_i)}$ is the vector with only the entries for which $w_{ij} = 1$, and similarly for $\boldsymbol{\mu}^{(\boldsymbol{w}_i)}$. The matrix $\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}$ is the submatrix of $\boldsymbol{\Sigma}$ containing only the rows and columns of the variables j with $w_{ij} = 1$. Finally, $d^{(\boldsymbol{w}_i)}$ is the dimension of $\boldsymbol{x}_i^{(\boldsymbol{w}_i)}$, that is the number of non-missing entries of $\boldsymbol{x}_i$. By convention, a case $\boldsymbol{x}_i$ consisting exclusively of NA's has $d^{(\boldsymbol{w}_i)} = 0$, $\text{MD}(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = 0$ and $|\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}| = 1$. Putting $L(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) := -2\ln(f(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}))$ we see that maximizing the observed likelihood of the entire dataset comes down to minimizing

$$\sum_{i=1}^{n} L(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \Big(\ln|\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}| + d^{(\boldsymbol{w}_i)}\ln(2\pi) + \text{MD}^2(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})\Big). \tag{7}$$

This maximum likelihood estimate of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is typically computed by the EM algorithm (Dempster, Laird, and Rubin 1977).

When constructing a cellwise MCD, the matrix W now describes which cells are flagged: a flagged cell x_{ij} gets $w_{ij} = 0$. The notations $\boldsymbol{x}_i^{(\boldsymbol{w}_i)}$, $d^{(\boldsymbol{w}_i)}$, $\boldsymbol{\mu}^{(\boldsymbol{w}_i)}$, and $\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}$ are interpreted analogously. The matrix W is not given in advance, but will be obtained through the estimation procedure. Now h can no longer apply to the number of unflagged *cases*. Instead, we apply it to the number of unflagged *cells* per column. We could minimize

$$\sum_{i=1}^{n} \Big(\ln|\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}| + d^{(\boldsymbol{w}_i)}\ln(2\pi) + \text{MD}^2(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})\Big) \tag{8}$$

under the constraints $\lambda_d(\boldsymbol{\Sigma}) \geqslant a$ and $||\boldsymbol{W}_{\cdot j}||_0 \geqslant h$ for all $j = 1, \ldots, d$

over $(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{W})$. The first constraint says that the smallest eigenvalue of $\boldsymbol{\Sigma}$ is at least as large as a number $a > 0$, where the eigenvalues of $\boldsymbol{\Sigma}$ are denoted as $\lambda_1(\boldsymbol{\Sigma}) \geqslant \cdots \geqslant \lambda_d(\boldsymbol{\Sigma})$. This ensures that $\boldsymbol{\Sigma}$ is nonsingular, which is required to compute Mahalanobis distances. In the second constraint, $||\boldsymbol{W}_{\cdot j}||_0$ is the number of nonzero entries in the jth column of $\boldsymbol{W}$. Note that we should not choose h too low. Whereas for the casewise MCD we can take h as low as $0.5n$, that would be ill-advised here because it could happen that two variables j and k do not overlap in the sense that $w_{ij}w_{ik} = 0$ for all i, making it impossible to estimate their covariance. We will impose that $h \geqslant 0.75n$ throughout.

However, minimizing (8) typically treats too many cells as outlying. This is because a value of h that is suitable for one variable may be too low for another, and we do not know ahead of time which variables have many outlying cells and which have few or none. To avoid flagging too many cells, we add a penalty counting the number of flagged cells in each column. The objective function of the *cellwise MCD* (cellMCD) then becomes

$$\sum_{i=1}^{n} \Big(\ln|\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}| + d^{(\boldsymbol{w}_i)}\ln(2\pi) + \text{MD}^2(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma})\Big) + \sum_{j=1}^{d} q_j||\boldsymbol{1}_d - \boldsymbol{W}_{\cdot j}||_0 \tag{9}$$

under the constraints $\lambda_d(\boldsymbol{\Sigma}) \geqslant a$ and $||\boldsymbol{W}_{\cdot j}||_0 \geqslant h$ for all $j = 1, \ldots, d$.

The notation $||\boldsymbol{1}_d - \boldsymbol{W}_{\cdot j}||_0$ stands for the number of nonzero elements in this vector, so the number of zero weights in column j of $\boldsymbol{W}$, that is, the number of flagged cells in column j of $\boldsymbol{X}$. The constants q_j for $j = 1, \ldots, d$ are computed (in Section 4) from the desired percentage of flagged cells in the absence of contamination. At the same time we keep the robustness constraint that $||\boldsymbol{W}_{\cdot j}||_0 \geqslant h$. Combining a penalty term with a $||.||_0$ constraint is not new, see the work of She, Wang, and Shen (2022) on casewise robust regression. In our context, the constraint $||\boldsymbol{W}_{\cdot j}||_0 \geqslant h$ will ensure the robustness of the estimator (through Proposition 2), whereas the penalty term $\sum_j q_j||\boldsymbol{1}_d - \boldsymbol{W}_{\cdot j}||_0$ discourages flagging too many cells, which improves the estimation accuracy at clean data as seen in simulations.

The cellMCD method is the first cellwise robust technique that combines the fitting of the parameters and the flagging of outlying cells $(\boldsymbol{W})$ in one objective function. The constraint $||\boldsymbol{W}_{\cdot j}||_0 \geqslant h$ for $j = 1, \ldots, d$ says that we require at least h unflagged cells in each column. In order to avoid a singular covariance matrix, we obviously need $h > d$. Combining these

inequalities we obtain $n > 4d/3$. But the curse of dimensionality implies that many spurious structures can be found in increasing dimensions, so we want a more comfortable ratio of cases per dimension. For the casewise MCD the rule of thumb is $n/d \geqslant 5$ (Rousseeuw and van Zomeren 1990), and we will require that here too.

The cellMCD method defined by (9) is equivariant for permuting the cases, for shifting the data, and for multiplying the variables by nonzero constants. But unlike the casewise MCD it is not equivariant under general nonsingular linear transformations, or even orthogonal transformations. This is because cells are intimately tied to the coordinate system, and an orthogonal transformation changes the cells. This is an important difference between the casewise and cellwise approaches. For instance, consider the standard multivariate Gaussian model in dimension $d = 4$ with the suspicious point $(10, 0, 0, 0)$. By an orthogonal transformation of the data, this point can be moved to $(\sqrt{50}, \sqrt{50}, 0, 0)$ or to $(5, 5, 5, 5)$. The casewise MCD is equivariant to such transformations and will still flag the same case. But in the cellwise paradigm $(10, 0, 0, 0)$ has one outlying cell, $(\sqrt{50}, \sqrt{50}, 0, 0)$ has two, and $(5, 5, 5, 5)$ has four, so cellMCD will react differently, as it should.

3. Theoretical Properties

Alqallaf et al. (2009) define the cellwise breakdown value of a location estimator. Here we will focus on finite-sample breakdown values in the sense of Donoho and Huber (1983) and Lopuhaä and Rousseeuw (1991). The *finite-sample cellwise breakdown value* of an estimator $\widehat{\boldsymbol{\mu}}$ at a dataset $\boldsymbol{X}$ is given by the smallest fraction of cells per column that need to be replaced to carry the estimate outside all bounds. Formally, let $\boldsymbol{X}$ be a dataset of size n, and denote by $\boldsymbol{X}^m$ any corrupted sample obtained by replacing at most m cells in each column of $\boldsymbol{X}$ by arbitrary values. Then the finite-sample cellwise breakdown value of a location estimator $\widehat{\boldsymbol{\mu}}$ at $\boldsymbol{X}$ is given by

$$\varepsilon_n^*(\widehat{\boldsymbol{\mu}}, \boldsymbol{X}) = \min\left\{\frac{m}{n} : \sup_{\boldsymbol{X}^m} \left|\left|\widehat{\boldsymbol{\mu}}(\boldsymbol{X}^m) - \widehat{\boldsymbol{\mu}}(\boldsymbol{X})\right|\right| = \infty\right\}. \quad (10)$$

Analogously to the casewise setting, we can also define the *cellwise explosion breakdown value* of a covariance estimator $\widehat{\boldsymbol{\Sigma}}$ as

$$\varepsilon_n^+(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{X}) = \min\left\{\frac{m}{n} : \sup_{\boldsymbol{X}^m} \lambda_1(\widehat{\boldsymbol{\Sigma}}) = \infty\right\}. \quad (11)$$

Moreover, we define the *cellwise implosion breakdown value* of $\widehat{\boldsymbol{\Sigma}}$ as

$$\varepsilon_n^-(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{X}) = \min\left\{\frac{m}{n} : \inf_{\boldsymbol{X}^m} \lambda_d(\widehat{\boldsymbol{\Sigma}}) = 0\right\}. \quad (12)$$

The definitions of the corresponding casewise breakdown values are very similar, the only difference being that the corrupted samples, let us call them $\widetilde{\boldsymbol{X}}^m$, are obtained by replacing at most m rows of $\boldsymbol{X}$ by arbitrary rows. If we denote the casewise breakdown values by δ_n^*, δ_n^+ and δ_n^- we can formulate the following simple but useful result:

Proposition 1. For all estimators $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ at any dataset $\boldsymbol{X}$ it holds that $\varepsilon_n^*(\widehat{\boldsymbol{\mu}}, \boldsymbol{X}) \leqslant \delta_n^*(\widehat{\boldsymbol{\mu}}, \boldsymbol{X})$, $\varepsilon_n^+(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{X}) \leqslant \delta_n^+(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{X})$, and $\varepsilon_n^-(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{X}) \leqslant \delta_n^-(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{X})$.

The proof consists of realizing that the casewise contaminated samples $\widetilde{\boldsymbol{X}}^m$ can be seen as cellwise contaminated samples $\boldsymbol{X}^m$. It is thus generally true that the cellwise breakdown value is less than or equal to the casewise breakdown value. Therefore, all upper bounds on casewise breakdown values in the literature also hold for cellwise breakdown values.

When proving breakdown values one often assumes that the original dataset $\boldsymbol{X}$ is in *general position*, meaning that no more than d points lie in any $d-1$ dimensional affine subspace. In particular, no three points lie on a line, no 4 points lie on a plane, and so on. When the data are drawn from a continuous distribution, it is in general position with probability 1. Real data have a limited precision, so they are not always in general position.

The inequalities in Proposition 1 can be strict. For instance, the casewise implosion breakdown value of the classical covariance matrix **Cov** at a dataset in general position is very high, in fact it is $(n-d)/n$ which goes to 1 for increasing sample size n. This is because whenever $d+1$ of the original data points are kept, **Cov** remains nonsingular. In stark contrast, its *cellwise* implosion breakdown value is quite low:

$$\varepsilon_n^-(\mathbf{Cov}, \boldsymbol{X}) = \left\lceil \frac{n-d}{d} \right\rceil / n \leqslant \frac{1}{d}. \quad (13)$$

To see why, let us pick d points of $\boldsymbol{X}$ which lie on a hyperplane that is not parallel to any coordinate axis. In the remaining $n-d$ rows we can then replace a single cell such that all of the resulting points lie on the same hyperplane, so **Cov** becomes singular. We can do this by replacing no more than $\lceil (n-d)/d \rceil$ cells in each variable, which is a fraction $\lceil (n-d)/d \rceil / n$ of its n cells.

Raymaekers and Rousseeuw (2023) recently derived a similar upper bound for all affine equivariant estimators $\widehat{\boldsymbol{\Sigma}}$. In order to obtain a higher cellwise breakdown value we are thus forced to leave the realm of affine equivariance. In fact, the constraint $\lambda_d(\widehat{\boldsymbol{\Sigma}}) \geqslant a > 0$ in the definition (9) of cellMCD is not affine invariant, but it keeps $\widehat{\boldsymbol{\Sigma}}$ from imploding. Therefore, the cellwise implosion breakdown value of cellMCD is 1.

We also want to know the breakdown value of its location estimate $\widehat{\boldsymbol{\mu}}$ and the explosion breakdown value of $\widehat{\boldsymbol{\Sigma}}$. These naturally depend on the choice of h.

Proposition 2. If the dataset $\boldsymbol{X}$ is in general position and $h \geqslant \lfloor \frac{n}{2} \rfloor + 1$, the cellMCD estimators $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ satisfy the properties

(a) $\varepsilon_n^-(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{X}) = 1$
(b) $\varepsilon_n^+(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{X}) \geqslant (n-h+1)/n$
(c) $\varepsilon_n^*(\widehat{\boldsymbol{\mu}}, \boldsymbol{X}) \geqslant (n-h+1)/n$
(d) The lower bound $(n-h+1)/n$ is sharp.

Proposition 2 shows that cellMCD is highly robust. Its proof is in Section A.1 of the supplementary material. By Proposition 1, it follows that these lower bounds also hold for the casewise breakdown values. This also implies that the method works on a mix of cellwise and casewise outliers as well. We

do not actually recommend to choose h as low as the proposition allows: as explained before this could lead to some poorly defined covariances and numerical instability. We stick with our earlier recommendation of $h \geqslant 0.75n$, and in fact $h = 0.75n$ is the default in our implementation.

Let us now turn to the asymptotic behavior of cellMCD. At the uncontaminated model distribution and for large n only a small fraction of cells is actually discarded, due to our choice of the constants q_j in the penalty term. In that situation the large-sample behavior of cellMCD is therefore the same as without the columnwise constraint on W. The cellMCD objective can then be written as

$$G(\mu, \Sigma, F) := \int g_{\mu,\Sigma}(x) F(\mathrm{d}x) \tag{14}$$

where

$$g_{\mu,\Sigma}(x) := \min_{w \in \{0,1\}^d} \left\{ \ln \left| \Sigma^{(w)} \right| + d^{(w)} \ln(2\pi) + \mathrm{MD}^2(x, w, \mu, \Sigma) + \boldsymbol{q}\,(\mathbf{1} - w)^\top \right\} \tag{15}$$

in which $\boldsymbol{q} = (q_1, \ldots, q_d)$ and $w = (w_1, \ldots, w_d)$. The cellMCD estimate is then

$$\operatorname*{argmin}_{(\mu,\Sigma)\in\Theta} G(\mu, \Sigma, F_n)$$

with F_n the empirical distribution and Θ the parameter space of (μ, Σ), which incorporates the condition $\lambda_d(\Sigma) \geqslant a$. Denote the set of minimizers as Θ^*. In section A.2 of the supplementary material the following Wald-type consistency result is shown, using work of Van der Vaart (2000):

Proposition 3. Let $(\hat{\mu}_n, \hat{\Sigma}_n)$ be a sequence of estimators which nearly minimize $G(\cdot, \cdot, F_n)$ in the sense that $G(\hat{\mu}_n, \hat{\Sigma}_n, F_n) \leqslant G(\mu^*, \Sigma^*, F_n) + o_P(1)$ for some $(\mu^*, \Sigma^*) \in \Theta^*$. Then it holds for all $\varepsilon > 0$ that

$$P(\, D((\hat{\mu}_n, \hat{\Sigma}_n), \Theta^*) \geqslant \varepsilon\,) \to 0\,,$$

where $D((\hat{\mu}_n, \hat{\Sigma}_n), (\mu^*, \Sigma^*)) := \max(||\hat{\mu}_n - \mu^*||_2, ||\hat{\Sigma}_n - \widehat{\Sigma}^*||_F)$ combines the Euclidean and Frobenius norms.

The population minimizer for Σ is not quite the underlying parameter, since a small fraction of cells is always given weight zero due to the penalty term in the objective. But for the location μ we can prove that the unique minimizer is indeed the underlying parameter vector, so the cellMCD functional for location is Fisher consistent:

Proposition 4. Let F be a strictly unimodal elliptical distribution with center μ and a density function. For any Σ, we then have the unique $\operatorname{argmin}_{m\in\mathbb{R}^d} G(m, \Sigma, F) = \mu$.

4. Algorithm

In the algorithm we will need the following result about decomposing the Mahalanobis distance and the likelihood.

Proposition 5. Let us split the d-variate case $\boldsymbol{x}$ into two nonempty blocks, and split $\boldsymbol{\mu}$ and the $d \times d$ positive definite matrix $\boldsymbol{\Sigma}$ accordingly, like

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \qquad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}.$$

Then $\mathrm{MD}^2(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$ and $L(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -2\ln(f(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}))$ satisfy

$$\mathrm{MD}^2(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathrm{MD}^2(\boldsymbol{x}_1, \widehat{\boldsymbol{x}}_1, \boldsymbol{C}_1) + \mathrm{MD}^2(\boldsymbol{x}_2, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \tag{16}$$

$$L(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = L(\boldsymbol{x}_1, \widehat{\boldsymbol{x}}_1, \boldsymbol{C}_1) + L(\boldsymbol{x}_2, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22}) \tag{17}$$

for $\widehat{\boldsymbol{x}}_1 = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)$ and $\boldsymbol{C}_1 = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$.

The proof can be found in section A.3 in the supplementary material. The proposition can be interpreted as follows. Take a case $\boldsymbol{x}_i$ with some but not all cells missing, and for simplicity assume that its missing components come first. Then put $\boldsymbol{x}_1 = \boldsymbol{x}_i^{(\mathbf{1}-\boldsymbol{w}_i)}$ and $\boldsymbol{x}_2$ the remainder. If $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ are the true underlying parameters, $\widehat{\boldsymbol{x}}_1$ is the conditional expectation $E[\boldsymbol{X}_1 | \boldsymbol{X}_2 = \boldsymbol{x}_2]$ and $\boldsymbol{C}_1$ is the conditional covariance matrix $\mathrm{cov}[\boldsymbol{X}_1 | \boldsymbol{X}_2 = \boldsymbol{x}_2]$. The additivity in (16) and (17) justifies the use of the partial Mahalanobis distances and the observed likelihood in our setting. Moreover, the fact that the difference of two "nested" MD^2 is again an MD^2 and hence nonnegative implies that the MD^2 is monotone for nested sets of variables. In particular, if $\boldsymbol{x}$ is observed fully we can write

$$\begin{aligned} &\mathrm{MD}^2(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \frac{r^2(x_1 | x_2, \ldots, x_d)}{s^2(X_1 | x_2, \ldots, x_d)} + \frac{r^2(x_2 | x_3, \ldots, x_d)}{s^2(X_2 | x_3, \ldots, x_d)} + \cdots \\ &+ \frac{r^2(x_{d-1} | x_d)}{s^2(X_{d-1} | x_d)} + \frac{(x_d - \mu_d)^2}{\Sigma_{dd}} \end{aligned} \tag{18}$$

where each time s^2 is the matrix $\boldsymbol{C}_1$ (which is a scalar here) and the residuals are $r(x_1 | x_2, \ldots, x_d) = x_1 - \widehat{x}_1(x_2, \ldots, x_d)$ and so on. Note that (18) holds for any order of the d variables. However, in each order the relative contribution of variable j to the total $\mathrm{MD}^2(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ may be different. For the likelihood we obtain similarly

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad = \quad & L(x_1, \mu_1, C_{1|2,\ldots,d}) + L(x_2, \mu_2, C_{2|3,\ldots,d}) + \cdots \\ & + L(x_{d-1}, \mu_{d-1}, C_{d-1|d}) + L(x_d, \mu_d, \Sigma_{dd}) \end{aligned} \tag{19}$$

in which the terms do not need to be positive.

If we set $q_j = 0$ in the objective function (9) of cellMCD and use casewise weights, that is, casewise constant w_{ij}, we recover the objective function (3) of the original casewise MCD. The latter is not convex in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, so neither is (9). The crucial ingredient in the algorithm for the casewise MCD is the concentration step (C-step) of Rousseeuw and Van Driessen (1999). After each C-step the new objective value is less than or equal to the old objective value, so iterating C-steps always converges to a stationary point. We will now construct a C-step for cellMCD with the same properties. Let us denote the current solution of cellMCD by $\widehat{\boldsymbol{\mu}}^{(k)}$, $\widehat{\boldsymbol{\Sigma}}^{(k)}$, and $\boldsymbol{W}^{(k)}$. Then the new C-step proceeds as follows.

Part (a) of the C-step. In this part we update the matrix $\boldsymbol{W}$ in (9) while keeping $\widehat{\boldsymbol{\mu}}^{(k)}$ and $\widehat{\boldsymbol{\Sigma}}^{(k)}$ unchanged. We start the new

pattern $\widetilde{\boldsymbol{W}}$ as $\widetilde{\boldsymbol{W}} = \boldsymbol{W}^{(k)}$, and then we modify $\widetilde{\boldsymbol{W}}$ column by column, by cycling over the variables $j = 1, \ldots, d$. The fact that this job can be done by column is advantageous for maintaining the constraint. Assume we are working on column j of $\widetilde{\boldsymbol{W}}$, possibly after having modified other columns of $\widetilde{\boldsymbol{W}}$ already. The current pattern of variable j is $\widetilde{\boldsymbol{W}}_{\cdot j}$ and we want to obtain a new pattern for column j to reduce the objective while leaving the other columns of $\widetilde{\boldsymbol{W}}$ unchanged. Note that we can write the objective (9) as $\sum_{i=1}^{n} \widetilde{L}(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{q})$ where

$$\widetilde{L}(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{q}) = \ln|\boldsymbol{\Sigma}^{(\boldsymbol{w}_i)}| + d^{(\boldsymbol{w}_i)} \ln(2\pi) + \mathrm{MD}^2(\boldsymbol{x}_i, \boldsymbol{w}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \sum_{j=1}^{d} q_j |1 - w_{ij}|$$

with $\boldsymbol{q} = (q_1, \ldots, q_d)$. For each $i = 1, \ldots, n$ we compute the difference in the total objective (9) between putting $\widetilde{w}_{ij} = 1$ and putting $\widetilde{w}_{ij} = 0$, which is

$$\begin{aligned}\Delta_{ij} &= \widetilde{L}(\boldsymbol{x}_i, \widetilde{w}_{ij} = 1, \widehat{\boldsymbol{\mu}}^{(k)}, \widehat{\boldsymbol{\Sigma}}^{(k)}, \boldsymbol{q}) - \widetilde{L}(\boldsymbol{x}_i, \widetilde{w}_{ij} = 0, \widehat{\boldsymbol{\mu}}^{(k)}, \widehat{\boldsymbol{\Sigma}}^{(k)}, \boldsymbol{q}) \\ &= \ln|\boldsymbol{\Sigma}^{(\widetilde{w}_{ij}=1)}| - \ln|\boldsymbol{\Sigma}^{(\widetilde{w}_{ij}=0)}| + \ln(2\pi) \\ &\quad + \mathrm{MD}^2(x_{ij}, \widehat{x}_{ij}, C_{ij}) - q_j \\ &= \ln(C_{ij}) + \ln(2\pi) + (x_{ij} - \widehat{x}_{ij})^2 / C_{ij} - q_j \end{aligned} \tag{20}$$

where the second and third equalities use Proposition 5 in which $\widehat{x}_{ij}$ and C_{ij} are now scalars. Note that $\widehat{x}_{ij} = \widehat{\mu}_j^{(k)} + \widehat{\boldsymbol{\Sigma}}_{j,o}^{(k)} (\widehat{\boldsymbol{\Sigma}}_{o,o}^{(k)})^{-1} (\widehat{\boldsymbol{x}}_{i,o} - \widehat{\boldsymbol{\mu}}_o^{(k)})$ is the conditional expectation of the cell X_{ij} conditional on the observed (subscript 'o') cells in row i, that is, those with $\widetilde{w}_{i\cdot} = 1$, taking into account any earlier modifications to $\widetilde{\boldsymbol{W}}$. Analogously, $C_{ij} = \widehat{\boldsymbol{\Sigma}}_{j,j}^{(k)} - \widehat{\boldsymbol{\Sigma}}_{j,o}^{(k)} (\widehat{\boldsymbol{\Sigma}}_{o,o}^{(k)})^{-1} \widehat{\boldsymbol{\Sigma}}_{o,j}^{(k)}$ is the conditional variance of X_{ij}. We now need to minimize $\sum_{i=1}^{n} \widetilde{L}(\boldsymbol{x}_i, \widetilde{w}_{ij}, \widehat{\boldsymbol{\mu}}^{(k)}, \widehat{\boldsymbol{\Sigma}}^{(k)}, \boldsymbol{q})$ subject to the constraint $\sum_{i=1}^{n} \widetilde{w}_{ij} \geqslant h$. If $\Delta_{ij} \leqslant 0$ holds for h or more i, then the minimum is attained by setting those $\widetilde{w}_{ij}$ to 1 and the others to 0. If not, it is attained by setting $\widetilde{w}_{ij}$ to 1 for the i with the h smallest Δ_{ij} and to 0 otherwise. After cycling through all columns of $\widetilde{\boldsymbol{W}}$ we set $\boldsymbol{W}^{(k+1)} = \widetilde{\boldsymbol{W}}$.

Part (b) of the C-step. Keeping the new pattern $\boldsymbol{W}^{(k+1)}$ fixed we now want to update $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$. As $\boldsymbol{W}^{(k+1)}$ is fixed the penalty term in (9) does not enter the minimization, so we are in the situation of the objective (7) for incomplete data, where the EM algorithm can be used. We first carry out one E-step which computes conditional means and products for the data entries with $\boldsymbol{W}_{ij}^{(k+1)} = 0$, for all rows. Next, we carry out an M-step, followed by imposing the constraint $\lambda_d \geqslant a$ by truncating the eigenvalues of $\widehat{\boldsymbol{\Sigma}}$ from below at a. The C-step ends by reporting $\boldsymbol{W}^{(k+1)}$, $\widehat{\boldsymbol{\mu}}^{(k+1)}$ and $\widehat{\boldsymbol{\Sigma}}^{(k+1)}$.

Proposition 6. (i) Each C-step turns a triplet $(\widehat{\boldsymbol{\mu}}^{(k)}, \widehat{\boldsymbol{\Sigma}}^{(k)}, \boldsymbol{W}^{(k)})$ satisfying the constraints in (9) into a new triplet $(\widehat{\boldsymbol{\mu}}^{(k+1)}, \widehat{\boldsymbol{\Sigma}}^{(k+1)}, \boldsymbol{W}^{(k+1)})$ which satisfies the same constraints and whose objective (9) is less than or equal to before. (ii) Iterating C-steps always converges.

For the proof see section A.3 in the supplementary material, which also contains the pseudocode of the algorithm. Many variations of the C-step are possible, such as cycling through the columns of $\widetilde{\boldsymbol{W}}$ in a different order. We could also cycle through the columns of $\widetilde{\boldsymbol{W}}$ more than once in part (a), and/or run more than one EM-step in part (b). But experiments in section A.6 of the supplementary material show that these changes have a negligible and nonsystematic effect on estimation accuracy, so we stay with the current version which is the fastest.

Note that cellMCD can still be used when the data contains missing cells, indicated by u_{ij} which are 0 for missing cells and 1 elsewhere. In that situation we first have to remove variables with more than $n - h$ missing values. In the C-step it then suffices to force $w_{ij} = 0$ whenever $u_{ij} = 0$.

In order to start our C-steps we need an initial estimator. In our experiments we found that the DDCW estimator of Raymaekers and Rousseeuw (2021b) gives good results and is very fast. It is a combination of the DetectDeviatingCells (DDC) method of Rousseeuw and Van den Bossche (2018) and the fast correlation method in (Raymaekers and Rousseeuw 2021a). DDCW is described in section A.4 of the supplementary material. Instead of starting from a single initial estimate, one could also start from several initial estimates. Iterating C-steps from each (with the same q_j and $a > 0$) until convergence, one can then keep the solution with the lowest objective (9).

The only remaining question is how to select the constants q_j but this is quite simple, we do not need cross-validation or an information criterion. In (20) the term $(x_{ij} - \widehat{x}_{ij})^2 / C_{ij}$ is the square of the residual $x_{ij} - \widehat{x}_{ij}$ standardized robustly. For inlying cells this should be below a cutoff, for which we take the Chi-squared quantile $\chi^2_{1,p}$ with one degree of freedom and probability p. The term $\ln(C_{ij})$ is approximated by using the conditional variance of variable j in the initial estimate $\widehat{\boldsymbol{\Sigma}}_0$, given by $C_j := 1/(\widehat{\boldsymbol{\Sigma}}_0^{-1})_{jj}$. So we set each q_j equal to

$$q_j = \chi^2_{1,p} + \ln(2\pi) + \ln(C_j)\,. \tag{21}$$

The effect of this choice is that a cell x_{ij} is flagged iff it lies outside a robust tolerance interval around its predicted value $\widehat{x}_{ij}$ with coverage probability p. Therefore, we only have to choose a single cutoff probability p to generate all q_j automatically. From simulations and examples we found that $p = 0.99$ was a good choice overall, so it is set as the default. Section A.5 provides more information on the q_j and the choice of p.

The algorithm has been implemented as the R function `cellMCD()`. It starts by checking the data for non-numerical variables, cases with too many NA's and so on. Next, it robustly standardizes the variables, and then computes the initial estimator followed by C-steps until convergence. The constraint $\lambda_d(\widehat{\boldsymbol{\Sigma}}) \geqslant a$ is applied to the standardized data, with default $a = 10^{-4}$. The function also reports the number of flagged cells in each variable. All the plots in the next section were made by the companion function `plot_cellMCD()`. Both functions have been included in the R package *cellWise* on CRAN.

5. Illustration on Real Data

We will illustrate cellMCD on the cars data obtained from the Top Gear website by Alfons (2016), focusing on the 11

numerical variables `price`, `displacement`, `horsepower`, `torque`, `acceleration time`, `top speed`, `miles per gallon`, `weight`, `length`, `width`, and `height`. This dataset is popular because both the variables and the cases (the cars) can easily be interpreted. After removing two cars with mostly NA's we have $n = 295$. We also replaced the highly right-skewed variables `price`, `displacement`, `horsepower`, `torque`, and `top speed` by their logarithms. On these data we ran cellMCD in its default version.

To visualize the results, we first look by variable. Consider variable j, say `horsepower`. Its ith cell has observed value x_{ij} as well as its prediction $\widehat{x}_{ij}$ obtained from the *unflagged* cells in the same row i, as in (20). In (20) we also see the conditional variance C_{ij} of this cell. It is then natural to plot the *standardized cellwise residual*

$$\text{stdres}_{ij} = \frac{x_{ij} - \widehat{x}_{ij}}{\sqrt{C_{ij}}} \tag{22}$$

which is NA when x_{ij} is missing. The left panel of Figure 2 shows the standardized residuals of the variable `horsepower` versus the index (case number) i. This plot was made by the function `plot.cellMCD()`, which also draws a horizontal tolerance band given by $\pm c$ where $c = \sqrt{\chi^2_{1,0.99}} \approx 2.57$. Here, some residuals stick out below the tolerance band. The Renault Twizy and Citroen DS3 are energy savers, whereas the Caterham is a super lightweight fun car. The most extreme outlier is the Chevrolet Volt with a standardized residual below -8. Top Gear lists this car's power as 86 hp, which cellMCD says is very low compared to what would be expected from the other 10 characteristics of this car. Looking it up revealed that the Volt actually has 149 hp. As far as we know this data error was not detected before.

The right panel of Figure 2 plots the standardized residuals of the variable `length` versus the observed `length` itself. The vertical lines are at $T \pm cS$ where T and S are robust univariate location and scale estimates of `length`, obtained from the function `estLocScale()` in the R package *cellWise*. The points to the left and right of such a vertical tolerance band are marginally outlying, that is their `length` stands out by itself without regard to the other variables. In the bottom left region of the plot we see five cars that are marginal outliers to the left and at the same time have outlying negative residuals, so they are short in absolute terms, as well as relative to what would be expected from their other characteristics. The Smart fortwo, Renault Twizy, Toyota IQ, and Aston Martin Cygnet are indeed tiny.

However, not all cellwise outliers are marginal outliers. In the middle bottom part of the plot we marked three cars whose length is not unusual by itself, but that are short relative to what would be expected based on their other 10 variables. They are sports cars, often built small to achieve high speeds. Note that there could also be points that lie inside the horizontal band but (slightly) outside the vertical band. They would correspond to cells that look a bit unusual in the variable j, but whose observed value x_{ij} is not that far from the predicted $\widehat{x}_{ij}$ based on its other variables.

The left panel of Figure 3 plots the standardized residual of each car's `weight` versus its prediction. Since all the points lie within the vertical tolerance band, no predictions are outlying. But we do see some outlying residuals, most of which can easily be explained. The Bentley is a heavy luxury car, and the Mercedes-Benz G an all-terrain vehicle. Below the horizontal tolerance band we see four lightweight sports cars. What remains is the Peugeot 107 which is small but not sporty at all. Top Gear reports its weight as 210 kg, which seems much too light for a car. Based on its other characteristics, cellMCD predicts its weight as 757 kg with a standard error of 89.5 kg. Looking up this car, its actual weight turns out to be 800 kg, so the value in the Top Gear dataset was mistaken.

The right panel of Figure 3 shows the observed value of `top speed` versus its prediction. Below the superimposed $y = x$ line we find some electric cars (BMW i3, Vauxhall Ampera) and some small cars (Smart fortwo and Renault Zoe). The one standing out most is the Renault Twizy, a tiny electric one-seater vehicle. Above the line we see some extremely fast sports cars. Also note that some points appear to lie on a horizontal line. Top Gear reports their top speed as 155 mph, corresponding to

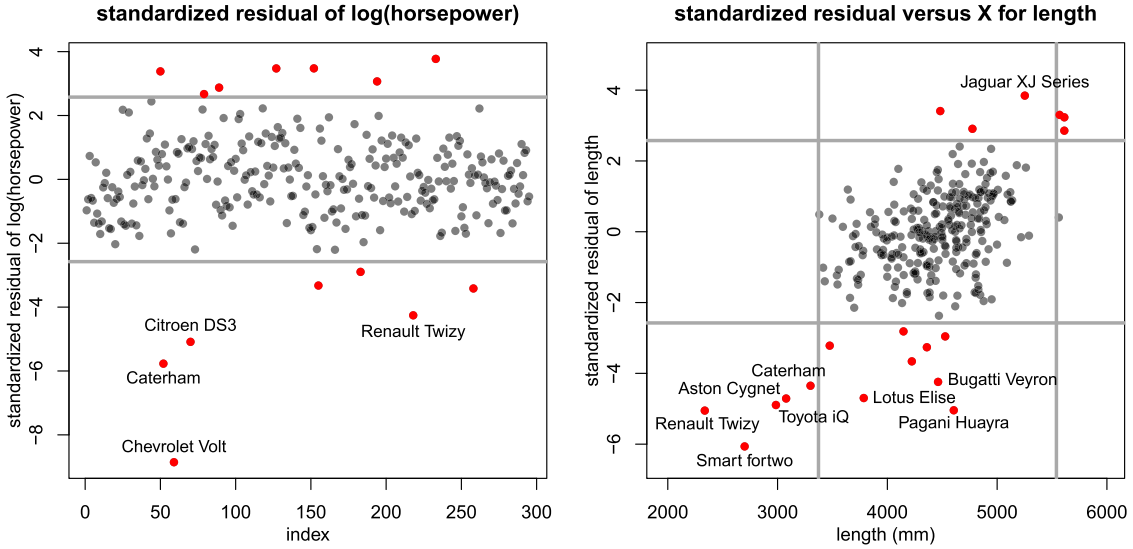

Figure 2. Top Gear data: (left) index plot of the standardized residual of `log(horsepower)`; (right) standardized residual of `length` versus observed `length`.

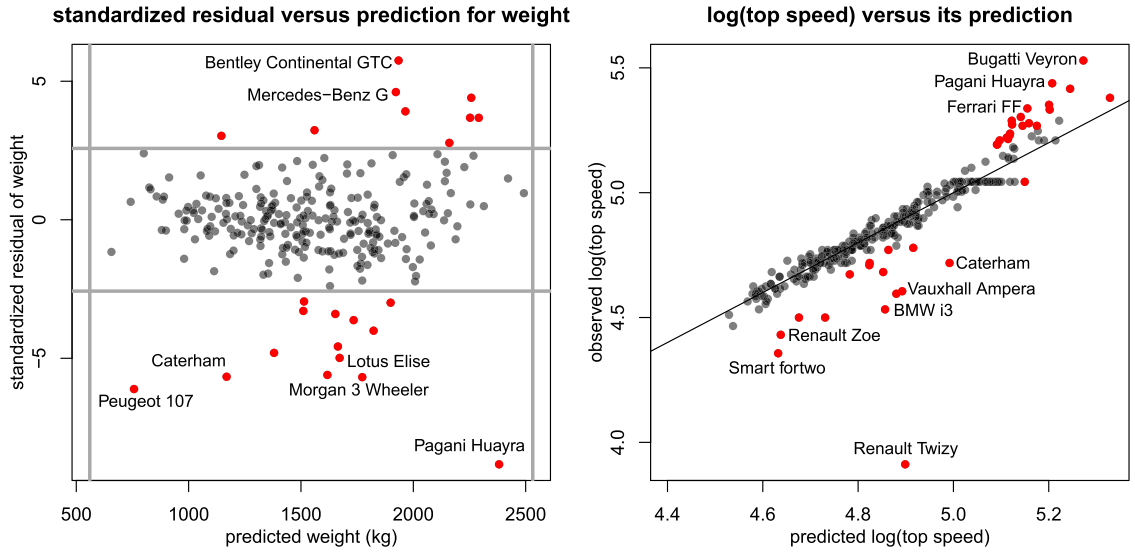

Figure 3. Top Gear data: (left) standardized residual of `weight` versus its prediction; (right) observed `log(top speed)` versus its prediction.

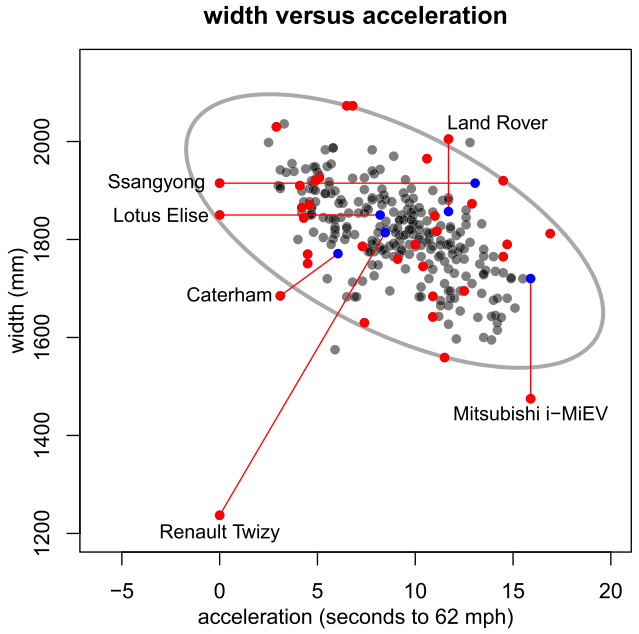

Figure 4. Top Gear data: bivariate plot of `width` versus `acceleration`. The 99% tolerance ellipse is given by the cellMCD estimates $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ restricted to the variables in the bivariate plot, and the red lines go to the predicted points shown in blue.

250 km/hr. Many of these cars were produced by Audi, BMW and Mercedes with a built-in 250 km/hr speed limiter.

The four plot types in Figures 2 and 3 all focus on a single variable. It can also be instructive to look at a pair of variables, say j and k. Figure 4 shows the variables `width` versus `acceleration`. The points for which $w_{ij} = 0$ or $w_{ik} = 0$ or both are automatically plotted in red. The figure also contains an ellipse, given by

$$\begin{bmatrix} x-\widehat{\mu}_j & y-\widehat{\mu}_k \end{bmatrix} \begin{bmatrix} \widehat{\Sigma}_{jj} & \widehat{\Sigma}_{jk} \\ \widehat{\Sigma}_{kj} & \widehat{\Sigma}_{kk} \end{bmatrix}^{-1} \begin{bmatrix} x-\widehat{\mu}_j \\ y-\widehat{\mu}_k \end{bmatrix} = q \qquad (23)$$

where q is the 0.99 quantile of the χ^2_2 distribution with two degrees of freedom. Note that outlyingness in this type of plot differs from cellwise outlyingness, since the former refers to two variables only, whereas the latter uses all 11 variables. So it is not unusual to see some red points inside the ellipse, and some black points outside it.

The width of the Land Rover is flagged as this is a wide all terrain vehicle. The red vertical line connects the observed point (x_{ij}, x_{ik}) to its predicted point $(\widehat{x}_{ij}, \widehat{x}_{ik})$ plotted in blue. That the line is vertical means that the `width` cell was flagged whereas the `acceleration` cell was not. The acceleration of the Ssangyong Rodius and Lotus Elise is outlying on the left. In fact, Top Gear lists their acceleration time as 0 which is physically impossible: presumably the true value was missing and encoded as 0 instead of NA. The same happens for the Renault Twizy. Note that also the `width` cell of the Twizy is flagged, so the red line to its predicted point is slanted instead of horizontal. The Caterham also has both cells flagged, as seen from its slanted line.

6. Simulation Results

In this section we evaluate the performance of cellMCD by a simulation study. The clean data is generated as n points from a d-variate Gaussian distribution with mean $\boldsymbol{\mu} = \mathbf{0}$. Since there is no affine equivariance, letting $\boldsymbol{\Sigma}$ be the identity matrix is not sufficient. Instead we use the types "A09" and "ALYZ". The entries of the A09 correlation matrix are given by $\boldsymbol{\Sigma}_{ij} = 0.9^{|i-j|}$, yielding both small and large correlations. The ALYZ type are randomly generated correlation matrices following the procedure of Agostinelli et al. (2015) and typically have mostly small absolute correlations. We consider three combinations of sample size and dimension (n, d): (100, 10), (400, 20), and (800, 40).

In these clean data, we then replace a fraction ε in $\{0.1, 0.2\}$ of cells by contaminated cells. These are generated as follows. First, for each column in the data matrix we randomly sample $n\varepsilon$ indices of cells to be contaminated. In each row, say $(z_1, \ldots, z_d)$, we then collect the indices of the cells to be contaminated. Denote this set of size k by $K = \{j_1, \ldots, j_k\}$. We

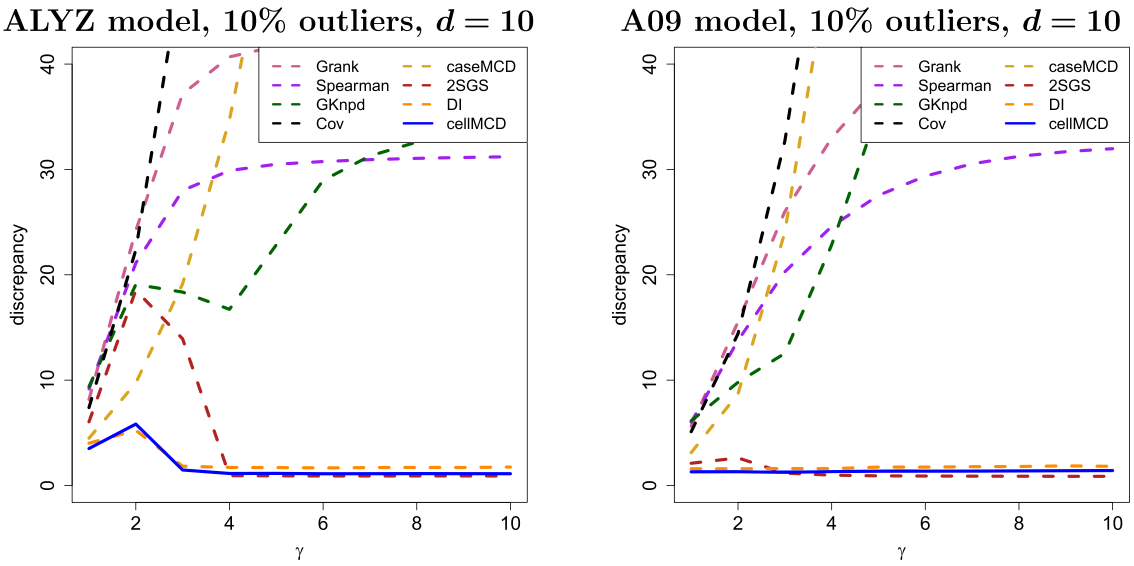

Figure 5. Discrepancy of estimated covariance matrices for $d = 10$ and $n = 100$.

next replace the cells $(z_{j_1}, \ldots, z_{j_k})$ by the k-dimensional vector $\gamma\sqrt{k}\,\boldsymbol{v}_K/\mathrm{MD}(\boldsymbol{v}_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$ where $\boldsymbol{\mu}_K$ and $\boldsymbol{\Sigma}_K$ are $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ restricted to the indices in K. The scalar $\gamma > 0$ quantifies the distance of the outlying cells to the center of the distribution, and we vary γ over $1, \ldots, 10$. The vector $\boldsymbol{v}_K$ is the normed eigenvector of $\boldsymbol{\Sigma}_K$ with the smallest eigenvalue. In each row, the outlying cells are thus structurally outlying in the subspace generated by the variables in K. Therefore, these cells will often not be marginally outlying, especially when $|K|$ is large and γ is relatively small, which makes them hard to detect. The R-package `cellWise` (Raymaekers and Rousseeuw 2022) contains the function `generateData` which generates the contaminated data according to this procedure.

We compare the proposed method cellMCD to the following alternative estimators:

- **Grank**, **Spearman**: the Gaussian and Spearman rank-based estimators used in Öllerer and Croux (2015) and Croux and Öllerer (2016);
- **GKnpd**: the Gnanadesikan-Kettenring estimator used in Tarr, Muller, and Weber (2016);
- **2SGS**: the two-step generalized S-estimator of Agostinelli et al. (2015);
- **DI**: the detection-imputation algorithm of Raymaekers and Rousseeuw (2021b).

In order to evaluate the performance of the different estimators, we compute the Kullback-Leibler discrepancy between the estimated $\widehat{\boldsymbol{\Sigma}}$ and the true $\boldsymbol{\Sigma}$ given by

$$\mathrm{KL}(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma}) = \mathrm{tr}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) - d - \log(\det(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1})) \,.$$

For each setting of the simulation parameters we generate 100 random datasets, and average the Kullback-Leibler discrepancy over these 100 replications. (For the variability around these averages see Section A.6.1.)

Figure 5 presents the results for $d = 10$, $n = 100$ and $\varepsilon = 0.1$. (The results for $\varepsilon = 0.2$ were similar.) Both cellMCD and DI perform well, as does 2SGS provided $\gamma \geqslant 4$. As expected, the classical covariance matrix (Cov) and the casewise MCD (labeled caseMCD) were not robust to these adversarial cellwise outliers. Note that the performances of Grank, Spearman and GKnpd do not improve as γ increases. While these estimators bound the influence that a single cell can have on the estimation, the effect remains substantial as the cell becomes more outlying. This is in contrast to 2SGS, DI and cellMCD in which far outliers get a zero weight.

The top panels of Figure 6 show the results for $n = 400$ and $d = 20$. The relative performances are similar to Figure 5. The 2SGS method still does well when $\gamma > 4$, but now suffers more for low γ. The performances of DI and cellMCD are again very close, with cellMCD often doing slightly better.

The lower panels with $n = 800$ and $d = 40$ are similar, with cellMCD performing best for all values of γ while DI is quite close, and 2SGS only doing well for higher γ.

Table 1 lists the computation times (in seconds) of the methods in the simulation. The first five methods are fast but they performed poorly. The bottom three methods did better. In dimensions 20 and 40 the cellMCD method was the fastest among them.

We are also interested in the performance of these methods on data without outliers. For this we repeated the simulation with $\varepsilon = 0$, again with 100 replications. The variability of each entry of the covariance matrix was measured taking the Fisher information of that entry into account. These results were then averaged over the upper triangular matrix entries including the diagonal. Next we divided the MSE of the classical MLE estimator by that of each robust method, yielding the finite-sample efficiencies in Table 2.

We see that the efficiency of cellMCD averages over 90%, which is excellent for a highly robust covariance estimator. This is similar to 2SGS, and outperforms DI. As expected Grank has a high efficiency, but we just saw that it performed poorly under contamination, as did GKnpd and Spearman. The finite-sample efficiency of cellMCD is much higher than that of the casewise MCD with the same coverage parameter $h = 0.75n$, which is under 0.70 for this range of dimensions d. This is due

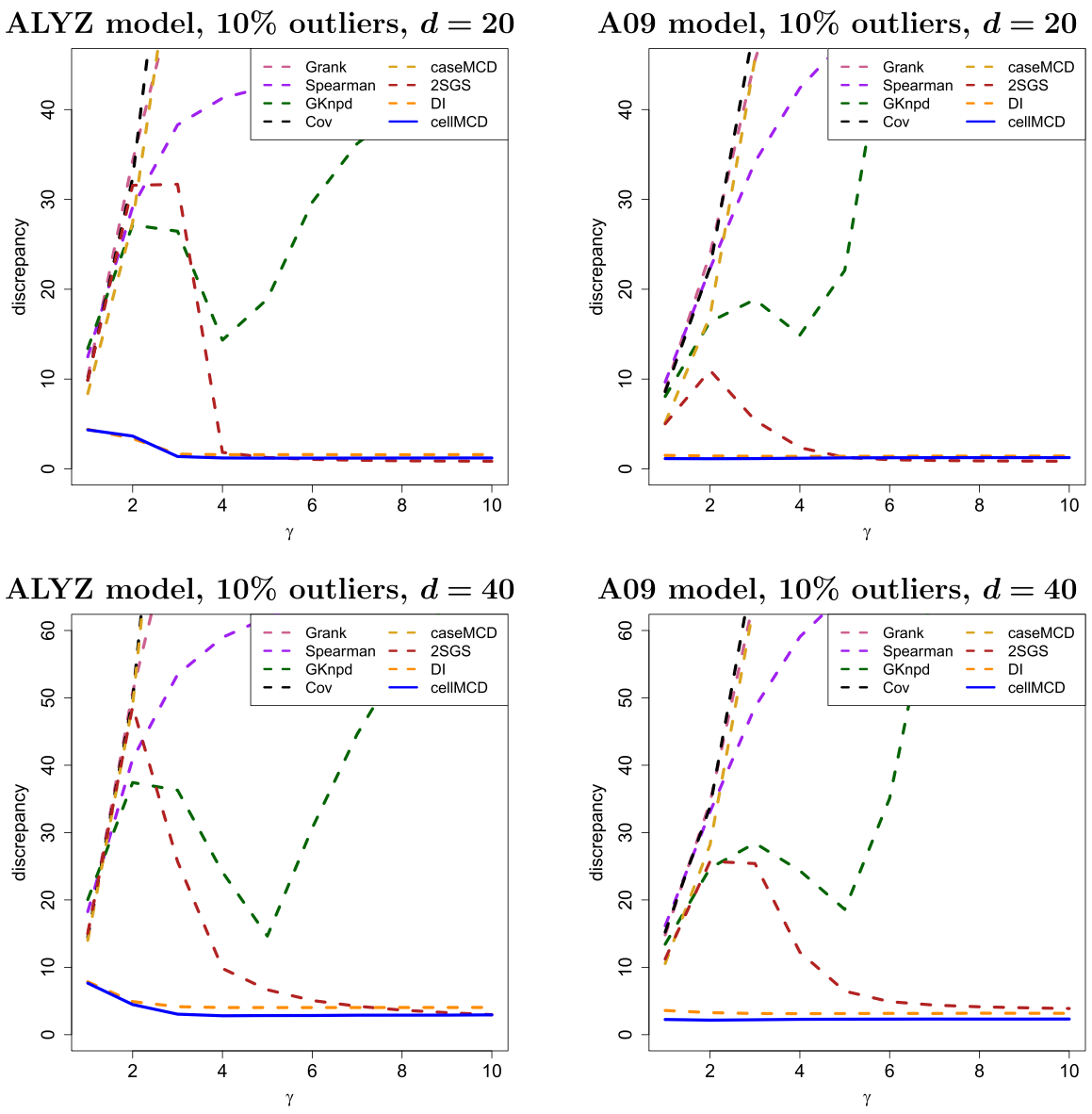

Figure 6. Discrepancy of estimated covariance matrices for $d = 20$ and $n = 400$ (top panels) and for $d = 40$ and $n = 800$ (bottom panels).

Table 1. Computation times of the methods in the simulation.

	$d = 10$	$d = 20$	$d = 40$
Cov	0.00	0.00	0.00
Grank	0.00	0.01	0.05
Spearman	0.01	0.02	0.06
GKnpd	0.90	1.31	4.29
caseMCD	0.04	0.53	2.37
DDCW	0.01	0.03	0.18
2SGS	0.67	6.91	66.88
DI	0.28	4.72	41.41
cellMCD	0.28	1.83	22.47

Table 2. Finite-sample efficiencies of robust covariance estimators.

	ALYZ configuration			A09 configuration		
method	$d = 10$	$d = 20$	$d = 40$	$d = 10$	$d = 20$	$d = 40$
cellMCD	0.90	0.90	0.89	0.89	0.93	0.96
2SGS	0.87	0.94	0.98	0.83	0.91	0.95
DI	0.68	0.61	0.49	0.87	0.90	0.90
GKnpd	0.74	0.80	0.81	0.78	0.77	0.79
Grank	0.90	0.96	0.98	0.88	0.89	0.94
Spearman	0.84	0.88	0.90	0.83	0.82	0.85

to the penalty term in (9), which made the number of actually discarded cells much smaller than $0.25\,n$.

We conclude that cellMCD is about equally robust as DI but with better efficiency, and is about as efficient as 2SGS but with better robustness at contaminated data. Moreover, it does substantially better at contaminated data than the remaining methods.

7. Discussion

The cellMCD method proposed here has an elegant formulation based on a single objective function, making it easier to understand than the earlier 2SGS and DI methods. We proved its good breakdown properties and consistency, and like the casewise MCD it can be computed by an algorithm based on C-steps that always lower the objective function and is guaranteed to converge. We have illustrated cellMCD on a real dataset

where the accompanying graphical displays revealed interesting aspects of the data that aided interpretation. Simulations indicate that cellMCD outperforms earlier cellwise methods, while being conceptually simple and rather fast to compute.

CellMCD is cellwise robust and incorporates a kind of sparsity penalty (on $\mathbf{1}-\boldsymbol{W}$). This naturally brings to mind the work of Candès et al. (2011). The goals are clearly related, but there are also some differences. The first is that their work assumes that the cellwise outlier pattern $\boldsymbol{W}$ is drawn uniformly at random, whereas we adopt the robustness paradigm that the outliers may be placed adversarially. Second, the method of Candès et al. (2011) is equivariant for transposing the data matrix, so it treats cases and variables in the same way, whereas in our setting they have to be treated differently. We do allow for some rows being flagged entirely, whereas we cannot allow flagging an entire column as this would make $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ not identifiable, which motivates our constraint $||\boldsymbol{W}_{.j}||_0 \geqslant h$ for $j = 1, \ldots, d$.

The fact that implosion breakdown can happen easily in the cellwise setting, see (13), was not mentioned in the literature before. We feel that, apart from cellMCD, also other cellwise robust covariance estimators could benefit from a constraint such as $\lambda_d(\widehat{\boldsymbol{\Sigma}}) \geqslant a$, or similarly from a formulation in which $\widehat{\boldsymbol{\Sigma}}$ is a sum of two matrices, one of which is a small multiple of the identity matrix.

The casewise MCD is typically followed by a reweighting step. This works as follows. First, the estimated covariance matrix $\widehat{\boldsymbol{\Sigma}}$ is multiplied by a correction factor $c_{n,d,h}$ such that $c_{n,d,h}\widehat{\boldsymbol{\Sigma}}$ is roughly unbiased when the original data are generated from a Gaussian distribution. Next, one computes the squared robust distances of the data points, given by $\mathrm{RD}_i^2 = (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^\top (c_{n,d,h}\widehat{\boldsymbol{\Sigma}})^{-1}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})$. Each case $\boldsymbol{x}_i$ then gets a weight w_i depending on its RD_i^2. Typically, the weight is set to 1 when RD_i^2 is below some quantile of the χ_d^2 distribution with d degrees of freedom, and to 0 otherwise. The final estimates are then the weighted mean and the weighted covariance matrix (4). This reweighting step increases the finite-sample efficiency of the estimator.

For cellMCD, the analogous reweighting step would compute the standardized residual (22) of every cell x_{ij} and compare its square to a quantile of the χ_1^2 distribution with 1 degree of freedom, yielding zero-one weights w_{ij}. With these w_{ij} one would then run the EM algorithm on the original data. But in fact, the result is not very different from the cellMCD result. This is because all the ingredients are already used in cellMCD, which contains the squared standardized residual in (20), the χ_1^2 quantile in (21), and the partial likelihood on which EM is based in (9). So in some sense the components of a reweighting step are already built into cellMCD itself. This explains its rather high finite-sample efficiency in Table 2.

Supplementary Materials

This is a text with the proofs of the propositions and some additional simulation results, as well as a zipped directory with the R code and vignette.

Acknowledgments

We are grateful for the constructive comments made by the Editor, Associate Editor, and five reviewers.

Data Availability Statement

The cellMCD method is implemented as the function `cellMCD()`, and the plots in Section 5 were drawn by the function `plot_cellMCD()`. Both functions are available in the R package *cellWise* on CRAN. Its vignette `cellMCD_examples` reproduces all results and figures in Section 5.

Disclosure Statement

The authors report there are no competing interests to declare.

ORCID

Peter J. Rousseeuw http://orcid.org/0000-0002-3807-5353

References

Aerts, S., and Wilms, I. (2017), "Cellwise Robust Regularized Discriminant Analysis," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10, 436–447. [2]

Agostinelli, C., Leung, A., Yohai, V. J., and Zamar, R. H. (2015), "Robust Estimation of Multivariate Location and Scatter in the Presence of Cellwise and Casewise Contamination," *Test*, 24, 441–461. [8,9]

Alfons, A. (2016), *robustHD: Robust Methods for High-Dimensional Data*, R package version 0.5.1, CRAN. [6]

Alqallaf, F., Van Aelst, S., Yohai, V. J., and Zamar, R. H. (2009), "Propagation of Outliers in Multivariate Data," *The Annals of Statistics*, 37, 311–331. [1,4]

Boudt, K., Rousseeuw, P. J., Vanduffel, S., and Verdonck, T. (2020), "The Minimum Regularized Covariance Determinant Estimator," *Statistics and Computing*, 30, 113–128. [1]

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011), "Robust Principal Component Analysis?" *Journal of the ACM*, 58, 1–37. [11]

Copt, S., and Victoria-Feser, M.-P. (2004), "Fast Algorithms for Computing High Breakdown Covariance Matrices with Missing Data," in *Theory and Applications of Recent Robust Methods*, eds. M. Hubert, G. Pison, A. Struyf, and S. Van Aelst, pp. 71–82, Basel: Birkhäuser. [1]

Croux, C., and Öllerer, V. (2016), "Robust and Sparse Estimation of the Inverse Covariance Matrix Using Rank Correlation Measures," in *Recent Advances in Robust Statistics: Theory and Applications*, pp. 35–55, New Delhi: Springer. [2,9]

Danilov, M., Yohai, V. J., and Zamar, R. H. (2012), "Robust Estimation of Multivariate Location and Scatter in the Presence of Missing Data," *Journal of the American Statistical Association*, 107, 1178–1186. [3]

De Ketelaere, B., Hubert, M., Raymaekers, J., Rousseeuw, P. J., and Vranckx, I. (2020), "Real-Time Outlier Detection for Large Datasets by RT-DetMCD," *Chemometrics and Intelligent Laboratory Systems*, 199, 103957. [1]

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, 39, 1–22. [3]

Donoho, D., and Huber, P. (1983), "The Notion of Breakdown Point," in *A Festschrift for Erich Lehmann*, eds. P. Bickel, K. Doksum, and J. Hodges, pp. 157–184, Belmont, CA: Wadsworth. [4]

Filzmoser, P., Höppner, S., Ortner, I., Serneels, S., and Verdonck, T. (2020), "Cellwise Robust M Regression," *Computational Statistics & Data Analysis*, 147, 106944. [2]

García-Escudero, L.-A., Rivera-García, D., Mayo-Iscar, A., and Ortega, J. (2021), "Cluster Analysis with Cellwise Trimming and Applications for the Robust Clustering of Curves," *Information Sciences*, 573, 100–124. [2]

Gnanadesikan, R., and Kettenring, J. (1972), "Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data," *Biometrics*, 28, 81–124. [2]

Higham, N. J. (2002), "Computing the Nearest Correlation Matrix – A Problem from Finance," *IMA Journal of Numerical Analysis*, 22, 329–343. [2]

Hubert, M., Debruyne, M., and Rousseeuw, P. J. (2018), "Minimum Covariance Determinant and Extensions," *Wiley Interdisciplinary Reviews: Computational Statistics*, 10, e1421. [1]

Hubert, M., Rousseeuw, P. J., and Segaert, P. (2015), "Multivariate Functional Outlier Detection," *Statistical Methods & Applications*, 24, 177–202. [2]

Hubert, M., Rousseeuw, P. J., and Van den Bossche, W. (2019), "MacroPCA: An All-in-One PCA Method Allowing for Missing Values as Well as Cellwise and Rowwise Outliers," *Technometrics*, 61, 459–473. [2]

Hubert, M., Rousseeuw, P. J., and Verdonck, T. (2012), "A Deterministic Algorithm for Robust Location and Scatter," *Journal of Computational and Graphical Statistics*, 21, 618–637. [1]

Katayama, S., Fujisawa, H., and Drton, M. (2018), "Robust and Sparse Gaussian Graphical Modelling under Cell-Wise Contamination," *Stat*, 7, e181. [2]

Little, R., and Rubin, D. (2020), *Statistical Analysis with Missing Data* (3rd ed.), New York: Wiley. [3]

Lopuhaä, H. P., and Rousseeuw, P. J. (1991), "Breakdown Points of Affine Equivariant Estimators of Multivariate Location and Covariance Matrices," *The Annals of Statistics*, 19, 229–248. [4]

Maechler, M., Rousseeuw, P., Croux, C., Todorov, V., Rückstuhl, A., Salibian-Barrera, M., Verbeke, T., Koller, M., Conceicao, E., and di Palma, M. (2022), *robustbase: Basic Robust Sstatistics*, R package, CRAN. [1]

Maronna, R. A., and Yohai, V. J. (2008), "Robust Low-Rank Approximation of Data Matrices with Elementwise Contamination," *Technometrics*, 50, 295–304. [2]

Öllerer, V., Alfons, A., and Croux, C. (2016), "The Shooting S-estimator for Robust Regression," *Computational Statistics*, 31, 829–844. [2]

Öllerer, V., and Croux, C. (2015), "Robust High-Dimensional Precision Matrix Estimation," in *Modern Nonparametric, Robust and Multivariate Methods*, eds. K. Nordhausen, and S. Taskinen, pp. 325–350, Cham: Springer. [2,9]

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011), "Scikit-learn: Machine Learning in Python," *The Journal of Machine Learning Research*, 12, 2825–2830. [1]

Raymaekers, J., and Rousseeuw, P. J. (2021a), "Fast Robust Correlation for High-Dimensional Data," *Technometrics*, 63, 184–198. [2,6]

——— (2021b), "Handling Cellwise Outliers by Sparse Regression and Robust Covariance," *Journal of Data Science, Statistics, and Visualization*, 1. [6,9]

——— (2022), *cellWise: Analyzing Data with Cellwise Outliers*, R package, CRAN. [9]

——— (2023), "Challenges of Cellwise Outliers," arXiv report 2302.02156. [4]

Rousseeuw, P. J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871–880. [1]

——— (1985), "Multivariate Estimation with High Breakdown Point," in *Mathematical Statistics and Applications*, eds. W. Grossmann, G. Pflug, I. Vincze, and W. Wertz, pp. 283–297, Dordrecht: Reidel. [1]

Rousseeuw, P. J., and Croux, C. (1993), "Alternatives to the Median Absolute Deviation," *Journal of the American Statistical Association*, 88, 1273–1283. [2]

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

Rousseeuw, P. J., and Van den Bossche, W. (2018), "Detecting Deviating Data Cells," *Technometrics*, 60, 135–145. [2,6]

Rousseeuw, P. J., and Van Driessen, K. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, 41, 212–223. [1,5]

Rousseeuw, P. J., and van Zomeren, B. C. (1990), "Unmasking Multivariate Outliers and Leverage Points," *Journal of the American Statistical Association*, 85, 633–651. [4]

Schreurs, J., Vranckx, I., Hubert, M., Suykens, J., and Rousseeuw, P. J. (2021), "Outlier Detection in Non-elliptical Data by Kernel MRCD," *Statistics and Computing*, 31, 1–18. [1]

She, Y., Wang, Z., and Shen, J. (2022), "Gaining Outlier Resistance with Progressive Quantiles: Fast Algorithms and Theoretical Studies," *Journal of the American Statistical Association*, 117, 1282–1295. [3]

Su, P., Tarr, G., and Muller, S. (2021), "Robust Variable Selection Under Cellwise Contamination," arXiv preprint 2110.12406. [2]

Tarr, G., Muller, S., and Weber, N. (2016), "Robust Estimation of Precision Matrices Under Cellwise Contamination," *Computational Statistics & Data Analysis*, 93, 404–420. [2,9]

Todorov, V. (2012), *rrcov: Scalable Robust Estimators with High Breakdown Point*, R package, CRAN. [1]

Van Aelst, S., Vandervieren, E., and Willems, G. (2011), "Stahel-Donoho Estimators with Cellwise Weights," *Journal of Statistical Computation and Simulation*, 81, 1–27. [2]

Van der Vaart, A. (2000), *Asymptotic Statistics*, Cambridge: Cambridge University Press. [5]