UNIVERSITEIT GENT

FACULTY OF SCIENCES

On methods for prediction based on complex data with missing values and robust principal component analysis

Holger Cevallos Valdiviezo

Thesis to obtain the degree of
Doctor in Statistical Data Analysis
Academic year 2016-2017

Promotor:
Prof. Stefan Van Aelst

Faculty of Sciences
Department of Applied Mathematics, Computer science and Statistics
Ghent University
Krijgslaan 281, B-9000 Gent

“Everything is possible for one who believes.”

Jesus Christ

Samenvatting

Massale hoeveelheden gegevens worden momenteel geproduceerd op verbazingwekkende snelheid. Technologische ontwikkelingen maken het goedkoper en toegankelijk voor bedrijven/instellingen om grote stromen van data te verkrijgen of te genereren. Deze gegevens kunnen verschillende types van complexiteiten bevatten zoals niet-geobserveerde waarden, onlogische waarden, extreme waarnemingen, en vele anderen. Anderzijds ervaren onderzoekers soms beperkingen om steeproefgegevens te bekomen. Zo kan het kostbaar zijn om een organisme te laten groeien in een lab. Daarom kan een onderzoeker ervoor kiezen om er slechts enkele te laten groeien, ten koste van een lagere kwaliteit van de resultaten. Bij dit soort gegevens wordt vaak een groot aantal eigenschappen gemeten in slechts een klein aantal waarnemingen, zodat de dimensie van de data veel groter is dan de omvang. Denk bijvoorbeeld aan microarray data. Heel vaak zijn beoefenaars meer bezorgd over de correcte inning van de gegevens dan het eigenlijke uitvoeren van een correcte data-analyse. In dit werk bespreken we methoden voor twee relevante stappen in de data-analyse. We kijken eerst naar methoden voor de verkennende stap waarbij de beoefenaar wil navigeren doorheen de grote stroom aan informatie in de data om te beginnen met het begrijpen van hun structuur en eigenschappen. Vervolgens bespreken wij methoden voor statistische data-analyse, gericht op een van de belangrijkste taken in deze stap: het voorspellen van een uitkomst. In dit werk willen we ook vaak voorkomende complexiteiten van real data toepassingen zoals hoog-dimensionale gegevens, atypische data en ontbrekende waarden aanpakken. Meer specifiek begint het proefschrift met een bespreking van methoden voor hoofdcomponentenanalyse, n van de meest populaire experimentele technieken. Deze methoden zijn uitbreidingen van de klassieke benadering van hoofdcomponenten analyse die bestand zijn tegen atypische gegevens. Hoofdstuk 1 beschrijft de Multivariate S- en de Multivariate Least Trimmed Squares schatters voor de principale componenten en stelt een algoritme voor dat meer robuuste resultaten kan opleveren en computationeel sneller is voor hoog-dimensionale problemen dan bestaande algoritmen voor deze methoden en andere robuuste methoden. We tonen aan dat de overeenkomstige functionalen Fisher-consistent zijn voor elliptische verdelingen. Bovendien bestuderen we de robuustheidseigenschappen van de Multivariate S-schatter door zijn invloedsfunctie af te leiden. De Multivariate S- en de Multivariate Least Trimmed Squares schatters richten zich echter alleen op uitschietende observaties (casewise outliers), dit wil zeggen waarnemingen zijn ofwel regulier ofwel uitschietend. Hoofdstuk 2 introduceert een nieuwe methode voor principale componenten waarvan aangetoond wordt dat ze beter tegen uitschietende metingen bestand is: de coordinatewise Least Trimmed Squares schatter. In het bijzonder kan ons voorstel cellwise uitschieters behandelen, die heel gebruikelijk zijn in moderne hoog-dimensionale datasets. We pasten ons algoritme voor multivariate methoden aan voor de coordinatewise Least Trimmed Squares schatter zodat deze snel kan berekend worden in hogere dimensies. Bovendien introduceren wij de functionaalvorm van de schatter en tonen aan dat deze Fisher-consistent is voor elliptische verdelingen. Hoofdstuk 3 breidt deze drie methoden uit naar de setting met functionele gegevens en laat zien dat deze uitbreidingen de robuustheidskenmerken van de methoden in de multivariate setting behouden. Het laatste hoofdstuk van het proefschrift handelt over het maken van voorspellingen in aanwezigheid van ontbrekende gegevens. Om voorspellingen te maken gebruiken we boom-gebaseerde methoden. Bomen zijn een populaire data mining techniek die het mogelijk maakt om voorspellingen te maken over gegevens van verschillende aard en in aanwezigheid van ontbrekende waarden. We vergelijken de voorspellingsprestaties van boom-gebaseerde technieken als de beschikbare trainingsdata variabelen met ontbrekende waarden bevatten. De ontbrekende waarden worden ofwel behandeld met behulp van surrogaat beslissingen in de bomen ofwel door de combinatie van een imputatiemethode met een boom-gebaseerde methode. Zowel classificatie- als regressieproblemen worden beschouwd. Over het algemeen tonen onze resultaten dat voor kleinere fracties van ontbrekende gegevens een ensemble methode gecombineerd met surrogaat beslissingen of enkelvoudige imputatie volstaat. Voor matige tot grote fracties van ontbrekende waarden, tonen ensemble methoden op basis van voorwaardelijke inferentiebomen in combinatie met meervoudige imputatie de beste prestaties, terwijl voorwaardelijke bagging gebruikmakend van surrogaten een goed alternatief is voor hoog-dimensionale voorspelling problemen. Theoretische resultaten bevestigen de potentieel betere voorspellingsprestaties van meervoudige imputatie ensembles.

Acknowledgements

First of all I would like thank to God, who under His Mercy, has allowed me to reach this important moment in my life. He has taught me that everything is possible and has confirmed His promise that He is with us everyday until the end.

I would like to thank my supervisor, Prof. Stefan Van Aelst, whose comments, explanations, and suggestions were valuable for the production of this work. Thank you for your patience and for having allowed me to have this enriching experience in which I have learned a lot and have shared knowledge with colleagues and professors from other universities and institutions. Likewise, I would like to thank Prof. Matias Salibian-Barrera for sharing his research ideas with me and for letting me work on them. Thank you also for letting me work with your computational facilities, they were of big help to get calculations faster and on time. I would like to take this opportunity to thank the other members of the examination committee, Prof. Luc Duchateau, Prof. Gentiane Haesbroeck, Prof. Tim Verdonck and Prof. Dries Benoit. Thank you very much for the effort you made for reading the entire thesis and for the very interesting comments I got from you all. They have improved the quality of this work. I would also like to thank Prof. Olivier Thas, who as the chairman of the examination committee, guided me in the process of getting all the administrative steps fulfilled to be able to defend my thesis.

I would like to give special thanks to my parents, who has supported me all the time. Gracias por escucharme, aconsejarme, estar pendiente todo el tiempo de mi, a la distancia. Gracias por hacer el esfuerzo de venir hasta acá. Gracias mami por orar por mi todos los días, se que lo haces. Los amo mucho y le doy gracias a Dios por permitirme compartir estos momentos junto a ustedes. Les dedico especialmente esta tesis. Thank you my little brother Daniel and thank you Diana for having come as well. Los amo. Quisiera también agradecer a mis tías: Elsy, Carmen, Azucena, Chela. Gracias por orar por mi y por apoyarme en mis metas. I would also like to give special thanks to my dear Ariana. Me has mostrado que el amor existe, que este todo lo soporta a pesar de todas las dificultades. No tengo palabras para agradecerte mi amor. Estaré muy feliz de regresar a casa y de compartir este logro contigo. Te amo.

I would also like to thank the professors from the Department of Statistics of Ghent University for sharing their knowledge with us, the students. To my office mates, thank you for tolerating me. I know it is very difficult! Thanks to all my dear friends that I have met during my stay in Ghent. It is difficult to name them all, but I keep everyone of you in my heart.

Finally, I would like to thank the financial support from the Interuniversity Attraction Poles Programme (IAP-network P7/06), Belgian Science Policy Office, which has allowed me to have this great experience as Ph.D. student.

Contents

List of Figures

List of Tables

To Holger and Patricia, with love.

Chapter 1

Robust multivariate subspace estimation for high-dimensional data

The content of this chapter is work in progress for future publication.

1.1 Introduction

Principal component analysis (PCA) is a popular exploratory tool for multivariate data. Classical principal component analysis can be formulated in several ways. One such formulation is as follows. Classical PCA aims to find an optimal lower-dimensional subspace in the sense of minimizing the mean of the squared euclidean distances between the original observations and their orthogonal projections onto the subspace. It is well-known that the directions spanning this optimal subspace correspond to the first eigenvectors of the sample covariance matrix. Hence, classical PCA also finds the directions of maximum variability of the data. PCA estimates are often used to visualize multivariate data and to quickly learn about the main sources of variation in the data. However, this classical approach to PCA is very sensitive to atypical data. In particular, the subspace found by minimizing the squared error loss can easily be pulled towards outliers.

There have been several proposals to robustify PCA. The earliest and easiest approach consists of taking the eigenvectors and eigenvalues of a robust scatter estimate instead of the standard sample covariance matrix. M-estimates, minimum volume ellipsoid (MVE) and S-estimates have been proposed for this purpose. However, this approach cannot be used for high-dimensional data because calculating high-dimensional robust scatter matrices is computationally complex. Moreover, while the effciency of robust scatter estimators increases with dimension (that is, their variance at elliptical distributions decreases), this comes at the expense of a loss of robustness. Therefore, Locantore et al. (1999) introduced spherical PCA which uses the covariance matrix of the data projected onto the unit sphere and can be calculated fast. Another alternative obtains robust PCA estimates by finding univariate directions that maximize a robust estimator of scale and are orthogonal to each other. This approach is known as robust projection pursuit (PP) and has been studied by Li and Chen (1985); Hubert et al. (2005) for example. A combination of both PP and robust scatter estimation was proposed by Hubert et al. (2005).

Instead of looking for one direction at a time as in PP, one can seek an optimal lower-dimensional subspace directly. To this end, Liu et al. (2003) replaced the squared error loss of classical PCA by the absolute value of the errors. Croux et al. (2003) proposed a weighted version of this procedure to reduce the effect of high-leverage points. Maronna (2005) considered robustly estimating the best lower-dimensional linear manifold by minimizing either an M-estimator of scale or a least trimmed squares (LTS) scale of the euclidean distances. Maronna called these approaches S-M and S-L and proposes an iterative algorithm to compute the solutions. He characterizes the solution by all directions orthogonal to the subspace and shows that these directions correspond to the eigenvectors associated with the smallest eigenvalues of a weighted covariance matrix. This may be a potential disadvantage when looking for a small dimensional subspace of high-dimensional data. In that case a high dimensional covariance matrix is still needed and a large number of eigenvectors is required. Computing all these directions in high-dimensional settings will take more time compared to computing only the first few directions comprising the subspace. Croux et al. (in press) investigated theoretical properties of the Maronna (2005) method based on the LTS scale.

In this chapter we re-investigate the methods of Maronna (2005) based on M and LTS scales, which we call Multivariate S-estimator and Multivariate least trimmed squares estimator respectively. We start with a short review of relevant properties of classical PCA in Section 1.1.1. We then give the definition of the Multivariate S-estimator estimator and corresponding estimating equations in section 1.2. Here, we also introduce the functional corresponding to the estimator. We show that the functional is Fisher-consistent at elliptical distributions and derive its influence function. In section 1.3 we give the definition of the Multivariate least trimmed squares estimator and introduce the corresponding functional. We show that the functional is Fisher-consistent at elliptical distributions. In section 1.4 algorithms for both estimators are proposed that are better suited for high-dimensional data. These algorithms directly determine the directions of the low-dimensional optimal subspace rather than a basis of directions orthogonal to the subspace. Moreover, our iterative algorithm uses estimating equations derived from first order conditions in order to update these directions, instead of computing high-dimensional covariance matrices as in the algorithm proposed by Maronna (2005). These modifications make it possible to calculate the S-M and S-L solutions faster in high-dimensional settings. We consider two choices for the starting values. The first uses random initial orthogonal matrices as in Maronna (2005) and aims to find the global minimum. The second uses a few deterministic starting values inspired by Hubert et al. (2012) and then finds the best local minimum that can be reached from these initial robust starting solutions. With deterministic starting values the algorithm is certainly faster and can make it feasible to calculate the solutions for even larger high-dimensional data. However, this approach is only useful if it does not jeopardize its performance. Section 1.5 discusses a fast strategy to choose the dimension of the subspace based on the proportion of unexplained variability. Finally, in section 1.6 we compare the performance of both algorithms through an extensive simulation study and a real data application.

1.1.1 Classical PCA approach

We first formalize our approach to the classical PCA problem. Consider a sample $Z_n = \{\mathbf{x}_i,\ i = 1, \ldots, n\} \subset \mathbb{R}^p$ and denote the corresponding data matrix by $\mathbf{X} = (\mathbf{x}_1 \ldots \mathbf{x}_n)^t$. Let $\overline{\mathbf{x}}(Z_n) = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ and $\widehat{\boldsymbol{\Sigma}}(Z_n) = \frac{1}{n-1}\sum_1^n (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^{\mathrm{T}}$ be the corresponding sample mean and sample covariance matrix. In this work, we consider principal component analysis as a method that looks for $q < p$ orthogonal unit vectors $\mathbf{b}^{(l)} \in \mathbb{R}^p$, $1 \le l \le q$, which span the linear subspace that gives the best approximation to the data set Z_n. Let $\mathbf{B}_q \in \mathbb{R}^{p\times q}$ be an orthogonal matrix with columns $\mathbf{B}_q = (\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(q)})$, i.e. $\mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q$, and rows $\mathbf{b}_j^{\mathrm{T}}$, $j = 1, \ldots, p$. Let $\mathbf{A}_q \in \mathbb{R}^{n\times q}$ be the matrix with rows $\mathbf{a}_i^{\mathrm{T}}$, $i = 1, \ldots, n$, and $\mathbf{m} \in \mathbb{R}^p$. The corresponding approximations of the observations are given by $\widehat{\mathbf{x}}_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) \equiv \widehat{\mathbf{x}}_i = \mathbf{m} + \mathbf{B}_q\mathbf{a}_i$, or elementwise $\hat{x}_{ij} = m_j + \mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j$. The associated multivariate residuals are given by $\mathbf{r}_i = \mathbf{x}_i - \widehat{\mathbf{x}}_i \in \mathbb{R}^p$. Its Euclidean norm, i.e. the Euclidean distance between $\mathbf{x}_i$ and its approximation $\widehat{\mathbf{x}}_i$, is denoted by $d_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = d_i = \|\mathbf{r}_i\|_{\mathbb{R}^p}$. The classical principal components solution is now found by minimizing

$$\min_{\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}} \sum_{i=1}^n \|\mathbf{x}_i - \widehat{\mathbf{x}}_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\|_{\mathbb{R}^p}^2 = \min_{\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}} \sum_{i=1}^n d_i^2(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) \tag{1.1}$$

over all $\mathbf{B}_q \in \mathbb{R}^{p\times q}$ orthogonal, $\mathbf{A}_q \in \mathbb{R}^{n\times q}$ and $\mathbf{m} \in \mathbb{R}^p$. The solution to this problem is obtained from the eigenvectors and eigenvalues of $\widehat{\boldsymbol{\Sigma}}(Z_n)$. Let $\widehat{\mathbf{B}}_q(Z_n)$ be the orthogonal matrix such that $\widehat{\mathbf{B}}_q(Z_n)^{\mathrm{T}}\widehat{\mathbf{\Sigma}}(Z_n)\widehat{\mathbf{B}}_q(Z_n) = \widehat{\mathbf{\Lambda}}(Z_n) = \mathrm{diag}(\widehat{\lambda}_1(Z_n), \widehat{\lambda}_2(Z_n), \ldots, \widehat{\lambda}_q(Z_n))$, where $\widehat{\lambda}_1(Z_n) \geq \widehat{\lambda}_2(Z_n) \geq \ldots \geq \widehat{\lambda}_q(Z_n) \geq 0$ are the q largest eigenvalues of $\widehat{\mathbf{\Sigma}}(Z_n)$. Then, the solution to (1.1) is given by $\widehat{\mathbf{B}}_{\mathrm{LS}} = \widehat{\mathbf{B}}_q(Z_n)$, $\widehat{\mathbf{m}}_{\mathrm{LS}} = \overline{\mathbf{x}}(Z_n)$ and $\widehat{\mathbf{A}}_{\mathrm{LS}}$ whose rows are given by $\widehat{\mathbf{a}}_{i,\mathrm{LS}}^{\mathrm{T}} = (\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{LS}})^{\mathrm{T}}\widehat{\mathbf{B}}_q(Z_n)$, $i = 1, \ldots, n$. Note that the vectors $\widehat{\mathbf{a}}_{i,\mathrm{LS}}$ are the scores of the observations $\mathbf{x}_i$ on the columns of $\widehat{\mathbf{B}}_{\mathrm{LS}}$. If we assume that $\widehat{\lambda}_q(Z_n) > \widehat{\lambda}_{q+1}(Z_n)$ then the PCA solution is unique (see Seber, 1984, Theorem 5.3).

Unfortunately, classical principal component analysis can be very sensitive to the presence of outliers. Since classical PCA is a least squares problem, outliers can pull the PCA subspace towards them. As a result, incorrect approximations for the regular data are obtained while the outliers cannot be detected because they do not appear as atypical points with unusually large Euclidean distance from the estimated subspace. Therefore, it is crucial to investigate approaches for PCA that can better resist the effect of these outliers.

1.2 The Multivariate S-estimator for PCA (MVS) in $\mathbb{R}^p$

1.2.1 The estimator

From (1.1) it is easy to see that the classical PCA solution is found by minimizing a scale estimate $\widehat{\sigma}^2(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ of the Euclidean distances of the residuals $\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = (d_1, \ldots, d_n)$, given by

$$\widehat{\sigma}^2(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})) = \frac{1}{n}\sum_{i=1}^{n} d_i^2(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}). \tag{1.2}$$

This classical scale estimator based on a quadratic loss function is clearly not robust against outliers. Maronna (2005) proposed to robustify the classical approach by replacing $\widehat{\sigma}$ by an M-estimator of scale which is defined as follows. For a real vector $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ an M-scale estimator $\widehat{\sigma}_{\mathrm{M}}(\mathbf{u})$ is the solution in s which satisfies

$$\frac{1}{n}\sum_{i=1}^{n} \rho_c\left(\frac{u_i}{s}\right) = b \tag{1.3}$$

where $\rho_c(t) = \rho(t/c)$ with $c > 0$ and where $\rho : \mathbb{R} \to \mathbb{R}_+$ is an even function such that $\rho(0) = 0$ and $\rho(t)$ is nondecreasing for $t > 0$ (see e g. Maronna (2005)). The constants c and b are tuning parameters which can be chosen by the user. These constants control consistency and robustness/efficiency of the estimator. For instance with $c = 1.54764$ and $b = 0.5$ the estimator is consistent at the normal distribution and has the maximum breakdown point of 50%.

The multivariate S-estimator for PCA can now be defined as the solution $(\widehat{\mathbf{B}}_{\text{MVS}}, \widehat{\mathbf{A}}_{\text{MVS}}, \widehat{\mathbf{m}}_{\text{MVS}})$ of the minimization problem

$$\min_{\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}} \widehat{\sigma}_{\text{M}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})), \tag{1.4}$$

where $\mathbf{B}_q \in \mathbb{R}^{p \times q}$ again needs to be an orthogonal matrix (i.e. $\mathbf{B}_q^{\text{T}} \mathbf{B}_q = \mathbf{I}_q$), and $\widehat{\sigma}_{\text{M}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ is the solution in s of the equation

$$\frac{1}{n} \sum_{i=1}^{n} \rho_c \left(\frac{d_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})}{s} \right) = b. \tag{1.5}$$

Note that in Maronna (2005) the Euclidean distance between each observation $\mathbf{x}_i$ and its projection $\widehat{\mathbf{x}}_i$ onto the q-dimensional subspace is measured in the $p - q$ dimensional orthogonal subspace which is equivalent to our current formulation in the p-dimensional space. We now write the MVS estimator defined in (1.4) in terms of the corresponding linear subspaces. Let $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\text{MVS}}}$ be the $q-$dimensional linear subspace spanned by the columns of $\widehat{\mathbf{B}}_{\text{MVS}}$. That is, $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\text{MVS}}}$ is the minimizer of

$$\min_{\dim(\mathcal{L}_{\mathbf{B}_q})=q} \quad \widehat{\sigma}_{\text{M}}(\mathbf{d}(\mathcal{L}_{\mathbf{B}_q})) \tag{1.6}$$

over all linear subspaces $\mathcal{L}_{\mathbf{B}_q}$ of dimension q where $\mathbf{d}(\mathcal{L}_{\mathbf{B}_q}) = (d_1(\mathcal{L}_{\mathbf{B}_q}), \ldots, d_n(\mathcal{L}_{\mathbf{B}_q}))$ are the Euclidean distances to the subspace and $\widehat{\sigma}_{\text{M}}(\mathbf{d}(\mathcal{L}_{\mathbf{B}_q}))$ is the solution in s of (1.5) analogous to $\widehat{\sigma}_{\text{M}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$.

By implicitly differentiating the M-scale in (1.5) we obtain first order conditions for the MVS estimator which will be useful to develop an iterative procedure to find local minima of the MVS optimization problem. Let us denote the coordinates of the multivariate residuals $\mathbf{r}_i$ by $r_{ij} = x_{ij} - m_j - \mathbf{a}_i^{\text{T}} \mathbf{b}_j$ such that

$$d_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = \|\mathbf{x}_i - \mathbf{m} - \mathbf{B}_q \mathbf{a}_i\| = \left[\sum_{j=1}^{p} (x_{ij} - m_j - \mathbf{a}_i^{\text{T}} \mathbf{b}_j)^2 \right]^{1/2}. \tag{1.7}$$

Then, the derivatives of $\widehat{\sigma}_{\mathrm{M}}$ with respect to $\mathbf{a}_i$, $\mathbf{b}_j$ and m_j become

$$\frac{\partial \widehat{\sigma}_{\mathrm{M}}}{\partial \mathbf{a}_i} = -\frac{\sum_{j=1}^{p} \rho'\left(\frac{d_i}{\widehat{\sigma}_{\mathrm{M}}}\right)\left(\frac{\widehat{\sigma}_{\mathrm{M}}}{d_i}\right) r_{ij}\mathbf{b}_j}{\sum_{i=1}^{n} \rho'\left(\frac{d_i}{\widehat{\sigma}_{\mathrm{M}}}\right) d_i}\,, \qquad i = 1, \ldots, n\,,$$

$$\frac{\partial \widehat{\sigma}_{\mathrm{M}}}{\partial \mathbf{b}_j} = -\frac{\sum_{i=1}^{n} \rho'\left(\frac{d_i}{\widehat{\sigma}_{\mathrm{M}}}\right)\left(\frac{\widehat{\sigma}_{\mathrm{M}}}{d_i}\right) r_{ij}\mathbf{a}_i}{\sum_{i=1}^{n} \rho'\left(\frac{d_i}{\widehat{\sigma}_{\mathrm{M}}}\right) d_i}$$

$$\frac{\partial \widehat{\sigma}_{\mathrm{M}}}{\partial m_j} = -\frac{\sum_{i=1}^{n} \rho'\left(\frac{d_i}{\widehat{\sigma}_{\mathrm{M}}}\right)\left(\frac{\widehat{\sigma}_{\mathrm{M}}}{d_i}\right) r_{ij}}{\sum_{i=1}^{n} \rho'\left(\frac{d_i}{\widehat{\sigma}_{\mathrm{M}}}\right) d_i}\,, \qquad j = 1, \ldots, p\,.$$

By setting the above equations to zero and writing

$$w_i = \rho'\left(\frac{d_i}{\widehat{\sigma}_{\mathrm{M}}}\right)\frac{\widehat{\sigma}_{\mathrm{M}}}{d_i} \tag{1.8}$$

we obtain the following estimating equations:

$$\sum_{j=1}^{p} (x_{ij} - m_j)\,\mathbf{b}_j = \left(\sum_{j=1}^{p} \mathbf{b}_j\,\mathbf{b}_j^{\mathrm{T}}\right)\mathbf{a}_i\,, \qquad 1 \le i \le n\,, \tag{1.9}$$

$$\sum_{i=1}^{n} w_i\,(x_{ij} - m_j)\,\mathbf{a}_i = \left(\sum_{i=1}^{n} w_i\,\mathbf{a}_i\,\mathbf{a}_i^{\mathrm{T}}\right)\mathbf{b}_j\,, \tag{1.10}$$

$$\sum_{i=1}^{n} w_i\,(x_{ij} - \mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j) = \sum_{i=1}^{n} w_i\,m_j\,, \qquad 1 \le j \le p. \tag{1.11}$$

These estimating equations naturally suggests an iterative reweighted least squares procedure to converge to local minima of the objective function which will be used in our algorithm of the estimator. From (1.9) we obtain that $\mathbf{a}_i = \mathbf{B}_q^T(\mathbf{x}-\mathbf{m})$. Hence, once $\mathbf{B}_q$ and $\mathbf{m}$ are known, the corresponding scores $\mathbf{a}_i$ of the observations are easily obtained. By combining this with (1.11) we can also see that $\mathbf{m} = \sum_{i=1}^{n} w_i\,\mathbf{x}_i / (\sum_{i=1}^{n} w_i)$. Note that if we put $w_i = 1$ for all observations, then the solution of these equations becomes the classical PCA solution.

By combining the estimating equations it can also be seen that the MVS-PCA solutions $(\widehat{\mathbf{B}}_{\mathrm{MVS}}, \widehat{\mathbf{m}}_{\mathrm{MVS}})$ satisfy the equation:

$$\sum_{i=1}^{n} w_i(\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{MVS}})(\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{MVS}})^{\mathrm{T}}\widehat{\mathbf{B}}_{\mathrm{MVS}} = \widehat{\mathbf{B}}_{\mathrm{MVS}}\,\widehat{\Lambda} \tag{1.12}$$

where $\widehat{\Lambda} = \widehat{\mathbf{B}}_{\mathrm{MVS}}^{\mathrm{T}} \sum_{i=1}^{n} w_i(\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{MVS}})(\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{MVS}})^{\mathrm{T}}\widehat{\mathbf{B}}_{\mathrm{MVS}}$ and w_i is given in (1.8).

From (1.12) it follows that the columns of $\widehat{\mathbf{B}}_{\mathrm{MVS}}$ can be taken as the first q eigenvectors of the weighted covariance matrix $\mathbf{C}(\widehat{\mathbf{m}}_{\mathrm{MVS}}, \widehat{\mathbf{B}}_{\mathrm{MVS}})$:

$$\mathbf{C}(\widehat{\mathbf{m}}_{\mathrm{MVS}}, \widehat{\mathbf{B}}_{\mathrm{MVS}}) = \frac{1}{n}\sum_{i=1}^{n} w_i(\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{MVS}})(\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{MVS}})^{\mathrm{T}}, \tag{1.13}$$

This is in accordance with expression (9) in Maronna (2005).

1.2.2 The functional

To investigate asymptotic properties of the MVS estimator, we first introduce the functional corresponding to the estimator. Consider a p-dimensional random variable $\mathbf{x}$ with a continuous distribution G. We assume that the distribution G has location parameter $\boldsymbol{\mu}$ and dispersion parameter $\boldsymbol{\Sigma} \in \mathrm{SPSD}(p)$, i.e. $\boldsymbol{\Sigma}$ belongs to the class of symmetric positive semi-definite matrices of size p. It follows that $\boldsymbol{\Sigma}$ can be decomposed as $\boldsymbol{\Sigma} = \boldsymbol{\beta}\boldsymbol{\Lambda}\boldsymbol{\beta}^{\mathrm{T}}$ where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$ and $\boldsymbol{\beta}$ is an orthogonal $p \times p$ matrix with columns $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(p)}$.

Similarly as for the MVS estimator, the MVS functionals $\mathbf{m}_{\mathrm{MVS}}(G)$, $\mathbf{B}_{\mathrm{MVS}}(G)$ and $\mathbf{a}_{\mathrm{MVS}}(G)$ satisfy $\mathbf{a}_{\mathrm{MVS}}(G) = \mathbf{B}^{T}_{\mathrm{MVS}}(G)\,(\,\mathbf{x} - \mathbf{m}_{\mathrm{MVS}}(G))$. To simplify notation, in what follows we drop G from the functionals. Therefore, we now focus on the functionals $(\mathbf{m}_{\mathrm{MVS}}, \mathbf{B}_{\mathrm{MVS}})$ which are the solution of the minimization problem

$$\min_{\mathbf{m},\, \mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q} \sigma_{\mathrm{M}}(d_G(\mathbf{x}, \mathbf{m}, \mathbf{B}_q)), \tag{1.14}$$

where $d_G(\mathbf{x}, \mathbf{m}, \mathbf{B}_q) = \left\|\mathbf{x} - \mathbf{m} - \mathbf{B}_q\mathbf{B}_q^{\mathrm{T}}\mathbf{x}\right\|$ and the M-scale functional σ_{M} satisfies

$$\int \rho\left(\frac{d_G(\mathbf{x}, \mathbf{m}, \mathbf{B}_q)}{\sigma_{\mathrm{M}}(d_G(\mathbf{x}, \mathbf{m}, \mathbf{B}_q))}\right) dG(\mathbf{x}) = b \tag{1.15}$$

The MVS functional can be written in a more general way as follows. Given an orthogonal matrix $\mathbf{B}_q$, let $\mathcal{L}_{\mathbf{B}_q}$ be the $q-$dimensional linear space spanned by the columns of $\mathbf{B}_q$. To simplify the presentation, assume that the functional $\mathbf{m}_{\mathrm{MVS}}$ is known. In addition, denote as $\pi(\mathbf{y}, \mathcal{L}_{\mathbf{B}_q})$ the orthogonal projection of $\mathbf{y}$ onto the subspace $\mathcal{L}_{\mathbf{B}_q}$. Therefore, the MVS functional $\mathcal{L}_{\mathbf{B}_{\mathrm{MVS}}}$ corresponding to the definition in (1.6) is the solution of the minimization problem

$$\min_{\dim(\mathcal{L}_{\mathbf{B}_q}) = q} \sigma_{\mathrm{M}}(d_G(\mathbf{x}, \mathcal{L}_{\mathbf{B}_q})), \tag{1.16}$$

over all linear subspaces $\mathcal{L}_{\mathbf{B}_q}$ of dimension q, where $d_G(\mathbf{x}, \mathcal{L}_{\mathbf{B}_q}) = \left\|\mathbf{x} - \mathbf{m} - \pi(\mathbf{x} - \mathbf{m}, \mathcal{L}_{\mathbf{B}_q})\right\|$ and the M-scale functional σ_{M} satisfies (1.15) for $d_G(\mathbf{x}, \mathcal{L}_{\mathbf{B}_q})$.

An analogous derivation as for (1.13) shows that the columns of the functional $\mathbf{B}_{\mathrm{MVS}}$ can be taken as the first q eigenvectors of the weighted covariance matrix $\mathbf{C}(G, \mathbf{m}_{\mathrm{MVS}}, \mathbf{B}_{\mathrm{MVS}})$ which is defined as:

$$\mathbf{C}(G, \mathbf{m}_{\mathrm{MVS}}, \mathbf{B}_{\mathrm{MVS}}) = \int w(\mathbf{x} - \mathbf{m}_{\mathrm{MVS}})(\mathbf{x} - \mathbf{m}_{\mathrm{MVS}})^{\mathrm{T}} dG(\mathbf{x}) \tag{1.17}$$

with weights

$$w = \rho' \left(\frac{d_G(\mathbf{x}, \mathbf{m}_{\mathrm{MVS}}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{M}}(d_G(\mathbf{x}, \mathbf{m}_{\mathrm{MVS}}, \mathbf{B}_{\mathrm{MVS}}))} \right) \frac{\sigma_{\mathrm{M}}(d_G(\mathbf{x}, \mathbf{m}_{\mathrm{MVS}}, \mathbf{B}_{\mathrm{MVS}}))}{d_G(\mathbf{x}, \mathbf{m}_{\mathrm{MVS}}, \mathbf{B}_{\mathrm{MVS}})}$$

and MVS location functional

$$\mathbf{m}_{\mathrm{MVS}} = \frac{\int w\, \mathbf{x}\, dG(\mathbf{x})}{\int w\, dG(\mathbf{x})}$$

Without loss of generality we can assume that $\boldsymbol{\mu} = \mathbf{0}$. We consider the case where $\mathbf{x}$ has a model distribution $G = F_{\boldsymbol{\Sigma}}$ with density

$$f_{\boldsymbol{\Sigma}}(\mathbf{x}) = \frac{g(\mathbf{x}^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \mathbf{x})}{\sqrt{\det(\boldsymbol{\Sigma})}}, \tag{1.18}$$

The function g is assumed to have a strictly negative derivative g' such that $F_{\boldsymbol{\Sigma}}$ is a unimodal elliptically symmetric distribution around the origin ($\boldsymbol{\mu} = \mathbf{0}$). To guarantee uniqueness of the best $q-$dimensional subspace $\mathcal{L}_q$ spanned by the columns of $\boldsymbol{\beta}_q = (\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(q)})$, we need a condition on the eigenvalues of $\boldsymbol{\Sigma}$. In particular, we need that $\lambda_q > \lambda_{q+1}$. The other eigenvalues may have the same value. Using (1.17) it can now be shown that the MVS-PCA functional $\mathcal{L}_{\mathbf{B}_{\mathrm{MVS}}}(G)$ is Fisher-consistent at unimodal elliptical distributions $F_{\boldsymbol{\Sigma}}$.

Theorem 1.1. *Let $\mathbf{x} \sim F_{\boldsymbol{\Sigma}}$, a p-dimensional elliptically distributed random variable with location $\mathbf{0}$ and scatter $\boldsymbol{\Sigma}$ such that $\boldsymbol{\Sigma} = \boldsymbol{\beta}\boldsymbol{\Lambda}\boldsymbol{\beta}^{\mathrm{T}}$ where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$, and $\boldsymbol{\beta}$ is an orthogonal matrix with columns $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(p)}$. Denote as $\mathcal{L}_q$ the linear space spanned by $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(q)}$. Assume that $\lambda_q > \lambda_{q+1}$. Then, $\mathcal{L}_{\mathbf{B}_{\mathrm{MVS}}}(F_{\boldsymbol{\Sigma}})$ is a Fisher-consistent functional for $\mathcal{L}_q$ at the model distribution $F_{\boldsymbol{\Sigma}}$, i.e.*

$$\mathcal{L}_{\mathbf{B}_{\mathrm{MVS}}}(F_{\boldsymbol{\Sigma}}) = \mathcal{L}_q \tag{1.19}$$

In order to assess the effect on the estimator of a small amount of contamination at a single point we derive the influence function of the corresponding functional. More specifically, the influence function of a functional T at a distribution G measures the effect on T of an infinitesimal contamination at a single point Hampel et al. (1986). Let us denote the point mass at a point $\mathbf{x}_0$ by $\Delta_{\mathbf{x}_0}$ and consider the contaminated distribution $G_{\epsilon,\mathbf{x}_0} = (1-\epsilon)G + \epsilon\Delta_{\mathbf{x}_0}$, then the influence function is given by

$$IF(\mathbf{x}_0, T, G) = \lim_{\epsilon\to 0} \frac{T(G_{\epsilon,\mathbf{x}_0}) - T(G)}{\epsilon} = \frac{\partial}{\partial\epsilon} T(G_{\epsilon,\mathbf{x}_0})|_{\epsilon=0}. \tag{1.20}$$

Let us assume w.l.o.g. that $\boldsymbol{\mu}$ is known. We consider the influence function of $\mathbf{B}_{\mathrm{MVS}}(G)$ at elliptical distributions. Let us assume, without loss of generality, that the columns of the functional $\mathbf{B}_{\mathrm{MVS}}(G)$ are ordered according to decreasing eigenvalues. To derive this influence function we start with the influence function for the functional $\mathbf{C}(G, \mathbf{B}_{\mathrm{MVS}})$ defined in (1.17). To simplify notation let us write $\mathbf{C}(G) = \mathbf{C}(G, \mathbf{B}_{\mathrm{MVS}})$, $\mathbf{B}_{\mathrm{MVS}} = \mathbf{B}_{\mathrm{MVS}}(G)$ and $\sigma_{\mathrm{S}} = \sigma_{\mathrm{M}}(d_G(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}}))$, the S-scale functional. Moreover, let $\mathbf{b}^{(j)}_{\mathrm{MVS}}$ denote the jth column of $\mathbf{B}_{\mathrm{MVS}}$, $j = 1, \ldots, q$. Recall that $\mathbf{b}^{(j)}_{\mathrm{MVS}}$ is the jth eigenvector of the weigthed covariance matrix $\mathbf{C}(G)$. Furthermore, let $\lambda_j(G)$ denote the jth eigenvalue of $\mathbf{C}(G)$. It can easily be seen that the MVS-PCA functional $\mathbf{B}_{\mathrm{MVS}}$ is orthogonal equivariant. Therefore, it suffices to compute the influence function at elliptical distributions $F_{\boldsymbol{\Sigma}}$ where $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma}$ is a diagonal matrix.

Theorem 1.2. *Let $F_{\boldsymbol{\Sigma}}$ be a p-dimensional elliptical distribution with location $\boldsymbol{\mu} = \mathbf{0}$ and scatter $\boldsymbol{\Sigma} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$. For the diagonal elements of $\mathbf{C}(F_{\boldsymbol{\Sigma}})$ it holds that:*

$$\begin{aligned} IF(\mathbf{x}_0, \mathbf{C}, F_{\boldsymbol{\Sigma}})_{ii} &= u\left(\frac{d_{F_{\boldsymbol{\Sigma}}}(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) x_{0i}^2 - \lambda_i(F_{\boldsymbol{\Sigma}}) \\ &\quad - \mathrm{E}_{F_{\boldsymbol{\Sigma}}}\left[u'\left(\frac{d_{F_{\boldsymbol{\Sigma}}}(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) d_{F_{\boldsymbol{\Sigma}}}(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, x_i^2\right] \cdot \frac{IF(\mathbf{x}_0, \sigma_{\mathrm{S}}, F_{\boldsymbol{\Sigma}})}{\sigma_S^2}, \end{aligned} \tag{1.21}$$

where $u(t) = \rho'(t)/t$ and the IF of the S-scale functional σ_{S} is given by

$$IF(\mathbf{x}_0, \sigma_{\mathrm{S}}, F_{\boldsymbol{\Sigma}}) = \frac{\sigma_{\mathrm{S}}^2\left(\rho\left(\frac{d_{F_{\boldsymbol{\Sigma}}}(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) - b\right)}{2b - 2b\sigma_{\mathrm{S}} + \sigma_{\mathrm{S}}^2\,\mathrm{E}_{F_{\boldsymbol{\Sigma}}}\left[\rho'\left(\frac{d_{F_{\boldsymbol{\Sigma}}}(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) d_{F_{\boldsymbol{\Sigma}}}(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\right]}.$$

For the (i,j) elements with $i = 1, \ldots, q$, $j = q+1, \ldots, p$, we have that

$$IF(\mathbf{x}_0, \mathbf{C}, F_{\boldsymbol{\Sigma}})_{ij} = \frac{[\lambda_j(F_{\boldsymbol{\Sigma}}) - \lambda_i(F_{\boldsymbol{\Sigma}})] \cdot u\left(\frac{d_{F_{\boldsymbol{\Sigma}}}(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) x_{0i} x_{0j}}{\lambda_j(F_{\boldsymbol{\Sigma}}) - \lambda_i(F_{\boldsymbol{\Sigma}}) - H_{ij}(\mathbf{B}_{\mathrm{MVS}})}. \tag{1.22}$$

For the (i,j) elements with $i = q+1, \ldots, p$, $j = 1, \ldots, q$, we have that

$$IF(\mathbf{x}_0, \mathbf{C}, F_{\boldsymbol{\Sigma}})_{ij} = \frac{[\lambda_j(F_{\boldsymbol{\Sigma}}) - \lambda_i(F_{\boldsymbol{\Sigma}})] \cdot u\left(\frac{d_{F_{\boldsymbol{\Sigma}}}(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) x_{0i} x_{0j}}{\lambda_j(F_{\boldsymbol{\Sigma}}) - \lambda_i(F_{\boldsymbol{\Sigma}}) + H_{ij}(\mathbf{B}_{\mathrm{MVS}})}. \tag{1.23}$$

Finally, for the (i,j) elements with $i = 1, \ldots, q$ and $j = 1, \ldots, q$, or with $i = q+1, \ldots, p$ and $j = q+1, \ldots, p$ such that $i \neq j$, we have that

$$IF(\mathbf{x}_0, \mathbf{C}, F_{\mathbf{\Sigma}})_{ij} = u\left(\frac{d_{F_{\mathbf{\Sigma}}}(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) x_{0i}x_{0j}, \tag{1.24}$$

with $d_{F_{\mathbf{\Sigma}}}(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}) = \|\mathbf{x}_0 - \mathbf{B}_{\mathrm{MVS}}\mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\mathbf{x}_0\|$. Finally, the expression for $H_{ij}(\mathbf{B}_{\mathrm{MVS}})$ is given in the Appendix.

Note that the influence functions are not bounded. However, they are non-increasing which means that the effect of a point $\mathbf{x}_0$ on the MVS-PCA functional decreases as the distance from the point to its projection $\mathbf{B}_{\mathrm{MVS}}\mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\mathbf{x}_0$ on the subspace increases. Thus, only good leverage points, i.e. outliers in the direction of the linear subspace, may have a large influence on the estimator. On the other hand, the influence of outliers w.r.t. the subspace is bounded, and smoothly redescends to zero for the non-diagonal elements.

Using the theorem above, one can immediately obtain the influence functions for the columns of $\mathbf{B}_{\mathrm{MVS}}$, i.e. the eigenvectors of $\mathbf{C}(F_{\mathbf{\Sigma}})$. Note that with the assumption of a diagonal $\mathbf{\Sigma}$ with distinct eigenvalues, it follows from the proof of Theorem 1.1 in Appendix A that $\mathbf{B}_{\mathrm{MVS}}(F_{\mathbf{\Sigma}}) = \boldsymbol{\beta}_q$. Then, Lemma 3 of Croux and Haesbroeck (2000) yields:

$$IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F_{\mathbf{\Sigma}})_{ij} = \frac{IF(\mathbf{x}_0, \mathbf{C}, F_{\mathbf{\Sigma}})_{ij}}{\lambda_j(F_{\mathbf{\Sigma}}) - \lambda_i(F_{\mathbf{\Sigma}})}(1 - \delta_{ij}) \tag{1.25}$$

where δ_{ij} is a boolean that takes value 1 when $i = j$ and 0 otherwise. Therefore, the diagonal elements of the influence function of $\mathbf{B}_{\mathrm{MVS}}$ are zero, i.e. $IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F_{\mathbf{\Sigma}})_{ii} = 0$, and only $i \neq j$ elements contribute to the IF of the columns $\mathbf{b}_{\mathrm{MVS}}^{(j)}$, $j = 1, \ldots, q$. For any $j = 1, \ldots, q$, we therefore obtain

$$IF(\mathbf{x}_0, \mathbf{b}_{\mathrm{MVS}}^{(j)}, F_{\mathbf{\Sigma}}) = \sum_{\substack{i=1\\ i\neq j}}^{q} \frac{u\left(\frac{d_{F_{\mathbf{\Sigma}}}(\mathbf{x}_0, \boldsymbol{\beta}_q)}{\sigma_{\mathrm{S}}}\right) x_{0i}x_{0j}}{\lambda_j(F_{\mathbf{\Sigma}}) - \lambda_i(F_{\mathbf{\Sigma}})} \boldsymbol{\beta}^{(i)} + \sum_{\substack{i=q+1\\ i\neq j}}^{p} \frac{u\left(\frac{d_{F_{\mathbf{\Sigma}}}(\mathbf{x}_0, \boldsymbol{\beta}_q)}{\sigma_{\mathrm{S}}}\right) x_{0i}x_{0j}}{\lambda_j(F_{\mathbf{\Sigma}}) - \lambda_i(F_{\mathbf{\Sigma}}) + H_{ij}(\boldsymbol{\beta}_q)} \boldsymbol{\beta}^{(i)} \tag{1.26}$$

since $\mathbf{B}_{\mathrm{MVS}}(F_{\mathbf{\Sigma}}) = \boldsymbol{\beta}_q$. The vector $\boldsymbol{\beta}^{(i)}$ is the ith eigenvector of Σ, $i = 1, \ldots, p$.

Note that these influence functions can be used to calculate asymptotic variances of the estimators or to look for influential points.

1.3 The Multivariate least trimmed squares estimator for PCA (MVLTS) in $\mathbb{R}^p$

1.3.1 The estimator

Another alternative to make PCA resistant to outliers is to replace the scale $\widehat{\sigma}$ of the classical approach by a least trimmed squares (LTS) scale. The LTS scale of the Euclidean distances of the residuals is defined as

$$\widehat{\sigma}^2_{\text{LTS}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})) = \frac{1}{h}\sum_{i=1}^{h} d^2_{(i:n)}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) \tag{1.27}$$

where $d_{(1:n)}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) \leq \ldots \leq d_{(n:n)}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})$ is the ordered sequence of Euclidean distances and $h = n - \lfloor n\alpha \rfloor$, $0 \leq \alpha \leq 1$. The multivariate LTS-estimator for PCA can now be defined as the solution $(\widehat{\mathbf{B}}_{\text{MVLTS}}, \widehat{\mathbf{A}}_{\text{MVLTS}}, \widehat{\mathbf{m}}_{\text{MVLTS}})$ of the minimization problem

$$\min_{\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}} \widehat{\sigma}^2_{\text{LTS}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})), \tag{1.28}$$

where $\mathbf{B}_q \in \mathbb{R}^{p\times q}$ is an orthogonal matrix. By discarding a portion α of the data the MVLTS estimator tries to exclude observations that are extreme and can represent outliers. Note that the formulation of this problem by Maronna (2005) in the $p-q$ dimensional orthogonal subspace is again equivalent to our formulation in the original $p-$dimensional space.

We now write the MVLTS problem (1.28) in terms of the corresponding linear subspaces. Let $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\text{MVLTS}}}$ be the $q-$dimensional linear subspace spanned by the columns of $\widehat{\mathbf{B}}_{\text{MVLTS}}$. That is, $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\text{MVLTS}}}$ is the minimizer of

$$\min_{\dim(\mathcal{L}_{\mathbf{B}_q})=q} \quad \widehat{\sigma}^2_{\text{LTS}}(\mathbf{d}(\mathcal{L}_{\mathbf{B}_q})) = \frac{1}{h}\sum_{i=1}^{h} d^2_{(i:n)}(\mathcal{L}_{\mathbf{B}_q}) \tag{1.29}$$

over all linear subspaces $\mathcal{L}_{\mathbf{B}_q}$ of dimension q, where $d_{(1:n)}(\mathcal{L}_{\mathbf{B}_q}) \leq \ldots \leq d_{(n:n)}(\mathcal{L}_{\mathbf{B}_q})$ is the ordered sequence of Euclidean distances to the subspace and $h = n - \lfloor n\alpha \rfloor$, $0 \leq \alpha \leq 1$. It can be seen from the definition of the LTS-scale that the MVLTS estimator tries to find at the same time an $h-$subset and the corresponding $q-$dimensional linear subspace that gives the smallest orthogonal distances of the h residuals. Hence, we now give an equivalent formulation to (1.29). Suppose that no h points of the dataset $Z_n = \{\mathbf{x}_i,\ i = 1, \ldots, n\} \subset \mathbb{R}^p$ lie in the same subspace of $\mathbb{R}^p$. Formally, this means that for all $\beta, \gamma \in \mathbb{R}^p$ it holds that

$$\#\left\{\mathbf{x}_i \mid \beta^{\mathrm{T}}\mathbf{x} + \gamma = 0\right\} < h \tag{1.30}$$

unless if β and γ are both zero vectors. Let $\mathcal{S} = \{H \subset \{1, \ldots, n\} \mid \#H = h\}$ be the collection of all subsets of size h. For any $H \in \mathcal{S}$ denote by $\overline{\mathbf{x}}(H) = \frac{1}{h}\sum_{i \in H}\mathbf{x}_i$ and $\widehat{\mathbf{\Sigma}}(H) = \frac{1}{h}\sum_{i \in H}(\mathbf{x}_i - \overline{\mathbf{x}}(H))(\mathbf{x}_i - \overline{\mathbf{x}}(H))^{\mathrm{T}}$ the mean and covariance matrix of the h observations $\{\mathbf{x}_i;\ i \in H\}$. Let $\widehat{\mathbf{B}}_{\mathrm{LS}}(H) \in \mathbb{R}^{p \times q}$ be the classical PCA solution based solely on the observations in H. Furthermore, let $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H)$ be the $q-$dimensional classical subspace spanned by the columns of $\widehat{\mathbf{B}}_{\mathrm{LS}}(H)$. The optimal $h-$subset is defined as the solution $\widehat{H}$ that minimizes

$$\min_{H \in \mathcal{S}} \sum_{i \in H} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H)) \tag{1.31}$$

where $d_i(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H))$ is the Euclidean distance from the ith observation $(i \in H)$ to that linear subspace.

Proposition 1. With the notation above, for datasets satisfying (1.30) we have that

$$\left\{\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widehat{H}) \mid \widehat{H} \in \operatorname*{arg\,min}_{H \in \mathcal{S}} \sum_{i \in H} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H))\right\} = \\ \left\{\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}} \in \operatorname*{arg\,min}_{\dim(\mathcal{L}_{\mathbf{B}_q}) = q} \widehat{\sigma}^2_{\mathrm{LTS}}(\mathbf{d}(\mathcal{L}_{\mathbf{B}_q}))\right\}. \tag{1.32}$$

Proposition (1) shows that for any subset H which obtains the best classical PCA approximation, its classical linear subspace is also a solution of the minimization problem in (1.29). In case that the solution is unique, we can write (1.32) as

$$\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{MVLTS}}} = \widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widehat{H}) \ \text{ where } \ \widehat{H} \in \operatorname*{arg\,min}_{H \in \mathcal{S}} \sum_{i \in H} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H)). \tag{1.33}$$

The columns of $\widehat{\mathbf{B}}_{\mathrm{LS}}(\widehat{H})$ are therefore the eigenvectors of the covariance matrix $\widehat{\mathbf{\Sigma}}(\widehat{H})$ corresponding to the q largest eigenvalues $\widehat{\lambda}_1(\widehat{H}) \geq \widehat{\lambda}_2(\widehat{H}) \geq \ldots \geq \widehat{\lambda}_q(\widehat{H}) \geq 0$. The corresponding estimates are $\widehat{\mathbf{m}}_{\mathrm{LS}}(\widehat{H}) = \overline{\mathbf{x}}(\widehat{H})$ and $\widehat{\mathbf{A}}_{\mathrm{LS}}(\widehat{H})$ whose rows are $\widehat{\mathbf{a}}^{\mathrm{T}}_{i,\mathrm{LS}}(\widehat{H}) = \left[\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{LS}}(\widehat{H})\right]^{\mathrm{T}} \widehat{\mathbf{B}}_{\mathrm{LS}}(\widehat{H})$. Note that (1.32) implies that $(\widehat{\mathbf{B}}_{\mathrm{LS}}(\widehat{H}), \widehat{\mathbf{A}}_{\mathrm{LS}}(\widehat{H}), \widehat{\mathbf{m}}_{\mathrm{LS}}(\widehat{H}))$ is also a solution of the MVLTS minimization problem in (1.28).

1.3.2 The functional

The functional form of the MVLTS estimator can be defined as follows. Consider a p-dimensional random variable $\mathbf{x}$ with a continuous distribution G. We assume again that the distribution G has location parameter $\boldsymbol{\mu}$ and dispersion parameter $\mathbf{\Sigma} \in \mathrm{SPSD}(p)$, decomposed as $\mathbf{\Sigma} = \boldsymbol{\beta}\mathbf{\Lambda}\boldsymbol{\beta}^{\mathrm{T}}$ where $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$ and $\boldsymbol{\beta}$ is an orthogonal $p \times p$ matrix with columns $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(p)}$. To define the MVLTS functional at the distribution G we need that

$$P_G(\beta^{\mathrm{T}}\mathbf{x} = 0) < 1 - \alpha \tag{1.34}$$

for all $\beta \in \mathbb{R}^p$ not equal to zero. Denote by $0 < \alpha < 1$ the probability mass of G not determining the MVLTS-PCA solution and define

$$\mathcal{D}_G(\alpha) = \{E \mid E \subset \mathbb{R}^p \text{ measurable and bounded with } P_G(E) = 1 - \alpha\}\,. \tag{1.35}$$

Let $(\mathbf{B}_{\mathrm{LS},E}(G), \mathbf{m}_{\mathrm{LS},E}(G))$ be the classical PCA functional for any subset $E \in \mathcal{D}_G(\alpha)$. To simplify notation we again drop G from the functionals in the remainder. Then, for any $E \in \mathcal{D}_G(\alpha)$, $(\mathbf{m}_{\mathrm{LS},E}, \mathbf{B}_{\mathrm{LS},E})$ is the solution of the minimization problem

$$\min_{\mathbf{m}, \mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q} \frac{1}{1-\alpha} \int_E d_G^2(\mathbf{x}, \mathbf{m}, \mathbf{B}_q)\ dG(\mathbf{x}), \tag{1.36}$$

where $d_G(\mathbf{x}, \mathbf{m}, \mathbf{B}_q) = \left\|\mathbf{x} - \mathbf{m} - \mathbf{B}_q\mathbf{B}_q^{\mathrm{T}}\mathbf{x}\right\|$ as before. An optimal subset $\widehat{E}$ satisfies that

$$\int_{\widehat{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\widehat{E}}, \mathbf{B}_{\mathrm{LS},\widehat{E}})\ dG(\mathbf{x}) \leq \int_E d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},E}, \mathbf{B}_{\mathrm{LS},E})\ dG(\mathbf{x}), \tag{1.37}$$

for all $E \in \mathcal{D}_G(\alpha)$. The MVLTS functionals are then defined as

$$\mathbf{B}_{\mathrm{MVLTS}} = \mathbf{B}_{\mathrm{LS},\widehat{E}} \qquad \text{and} \qquad \mathbf{m}_{\mathrm{MVLTS}} = \mathbf{m}_{\mathrm{LS},\widehat{E}}. \tag{1.38}$$

From the classical PCA estimator we know that $\mathbf{m}_{\mathrm{LS},\widehat{E}} = \frac{1}{1-\alpha}\int_{\widehat{E}} \mathbf{x}\ dG(\mathbf{x})$ and that the columns of $\mathbf{B}_{\mathrm{LS},\widehat{E}}$ are the first q eigenvectors of the covariance matrix functional computed at $\widehat{E}$:

$$\mathbf{\Sigma}_{\widehat{E}}(G) = \frac{1}{1-\alpha}\int_{\widehat{E}} (\mathbf{x} - \mathbf{m}_{\mathrm{LS},\widehat{E}})(\mathbf{x} - \mathbf{m}_{\mathrm{LS},\widehat{E}})^{\mathrm{T}}\, dG(\mathbf{x}). \tag{1.39}$$

The MVLTS functional can also be written in terms of linear subspaces. To simplify the presentation, assume that the functional $\mathbf{m}_{\mathrm{MVLTS}}$ is known. Let $\pi(\mathbf{x} - \mathbf{m}, \mathcal{L}_{\mathbf{B}_q})$ be the orthogonal projection of $(\mathbf{x} - \mathbf{m})$ onto the subspace $\mathcal{L}_{\mathbf{B}_q}$ and define $d_G(\mathbf{x}, \mathcal{L}_{\mathbf{B}_q}) = \left\|\mathbf{x} - \mathbf{m} - \pi(\mathbf{x} - \mathbf{m}, \mathcal{L}_{\mathbf{B}_q})\right\|$. Furthermore, let $\mathcal{L}_{\mathbf{B}_{\mathrm{LS},E}}$ be the linear subspace spanned by the columns of $\mathbf{B}_{\mathrm{LS},E}$, with $E \in \mathcal{D}_G(\alpha)$. Analogous to Equation (1.37), an optimal subset $\widehat{E}$ satisfies that

$$\int_{\widehat{E}} d_G^2(\mathbf{x}, \mathcal{L}_{\mathbf{B}_{\mathrm{LS},\widehat{E}}})\ dG(\mathbf{x}) \leq \int_E d_G^2(\mathbf{x}, \mathcal{L}_{\mathbf{B}_{\mathrm{LS},E}})\ dG(\mathbf{x}), \tag{1.40}$$

for all $E \in \mathcal{D}_G(\alpha)$. Then, the MVLTS functional corresponding to the definition of the estimator in (1.33) is defined as

$$\mathcal{L}_{\mathbf{B}_{\mathrm{MVLTS}}} = \mathcal{L}_{\mathbf{B}_{\mathrm{LS},\widehat{E}}} \tag{1.41}$$

The following proposition states that the MVLTS solution can be taken in a region $\mathcal{E}$.

Lemma 1.3. *Consider a distribution G satisfying condition (1.34) and an MVLTS solution $\mathcal{L}_{\mathbf{B}_{\mathrm{LS},\widehat{E}}}$, with $\widehat{E} \in \mathcal{D}_G(\alpha)$. Define the region $\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^p;\, d_G^2(\mathbf{x}, \mathcal{L}_{\mathbf{B}_{\mathrm{LS},\widehat{E}}}) \leq D_\alpha^2\}$ where D_α^2 is chosen such that $P_G(\mathcal{E}) = 1 - \alpha$. Then it holds that*

$$\mathcal{L}_{\mathbf{B}_{\mathrm{LS},\mathcal{E}}} = \mathcal{L}_{\mathbf{B}_{\mathrm{LS},\widehat{E}}} \tag{1.42}$$

Next, we show that the MVLTS estimator inherits the orthogonal equivariance property from the classical principal component estimator.

Lemma 1.4. *Let $\boldsymbol{\Upsilon} \in \mathbb{R}^{p\times p}$ be any orthogonal matrix. Without loss of generality assume that the true location $\boldsymbol{\mu}$ is known and equal to $\mathbf{0}$. Consider the orthogonal transformation $\boldsymbol{\Upsilon}\mathbf{x}$ of the p-dimensional random vector $\mathbf{x}$. Then the MVLTS functional $\mathbf{B}_{\mathrm{MVLTS}}$ is orthogonally equivariant in the sense that*

$$\mathbf{B}_{\mathrm{MVLTS}}(\boldsymbol{\Upsilon}\mathbf{x}) = \boldsymbol{\Upsilon}\mathbf{B}_{\mathrm{MVLTS}}(\mathbf{x}) \tag{1.43}$$

In the context of PCA orthogonal equivariance is sufficient since the classical PCA procedure is only orthogonal equivariant.

For the MVLTS we also consider the case where $\mathbf{x}$ has a unimodal elliptically symmetric model distribution that is centered around the origin, i.e. $G = F_{\boldsymbol{\Sigma}}$ with density given by (1.18). To guarantee uniqueness of the best $q-$dimensional subspace $\mathcal{L}_q$we need the condition on the eigenvalues of $\boldsymbol{\Sigma}$ that $\lambda_q > \lambda_{q+1}$. Using (1.39) and Lemma 1.3 it can now be shown that the MVLTS-PCA functional $\mathcal{L}_{\mathbf{B}_{\mathrm{MVLTS}}}(G)$ is Fisher-consistent at $F_{\boldsymbol{\Sigma}}$.

Theorem 1.5. *Let $\mathbf{x} \sim F_{\boldsymbol{\Sigma}}$, a p-dimensional elliptically distributed random variable with location $\mathbf{0}$ and scatter $\boldsymbol{\Sigma}$ such that $\boldsymbol{\Sigma} = \boldsymbol{\beta}\boldsymbol{\Lambda}\boldsymbol{\beta}^{\mathrm{T}}$ where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$, and $\boldsymbol{\beta}$ is an orthogonal matrix with columns $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(p)}$. Denote as $\mathcal{L}_q$ the linear space spanned by $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(q)}$. Assume that $\lambda_q > \lambda_{q+1}$. Then, $\mathcal{L}_{\mathbf{B}_{\mathrm{MVLTS}}}(F_{\boldsymbol{\Sigma}})$ is a Fisher-consistent functional for $\mathcal{L}_q$ at the model distribution $F_{\boldsymbol{\Sigma}}$, i.e.*

$$\mathcal{L}_{\mathbf{B}_{\mathrm{MVLTS}}}(F_{\boldsymbol{\Sigma}}) = \mathcal{L}_q \tag{1.44}$$

Croux et al. (in press) derived the influence function of the MVLTS functional $\mathbf{B}_{\mathrm{MVLTS}}$ which turn out to be bounded for bad leverage points. However, good leverage points still may have an unbounded influence.

1.4 The algorithm

We start with a description of the algorithm for the MVS and MVLTS estimators in pseudo-code. Our algorithm depends on the initial choices of $\mathbf{B}_q$ and $\mathbf{m}$ as well as on the tuning parameters N_1, N_2, N_{pc} and *tol* and can by summarized as follows:

```
1. Set $it \leftarrow 0$.
    a. Compute $\mathbf{a}_i^{\mathrm{T}} = (\mathbf{x}_i - \mathbf{m})^{\mathrm{T}}\mathbf{B}_q$, $i = 1, \dots, n$, and append these vectors to the rows of $\mathbf{A}_q$.
    b. Compute residual distances $d_i = \|\mathbf{r}_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\|$, $i = 1, \dots, n$, from (1.7).
    c. Compute $\widehat{\sigma}(\mathbf{d})$:
        • For the MVS estimator: $\widehat{\sigma}(\mathbf{d}) = \widehat{\sigma}_{\mathrm{M}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ from (1.5).
        • For the MVLTS estimator: $\widehat{\sigma}(\mathbf{d}) = \widehat{\sigma}_{\mathrm{LTS}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ from (1.27).
    d. Set $\widehat{\sigma}_0^2 = \widehat{\sigma}^2(\mathbf{d})$.
    e. Set $it = 1$.
2. Do until $it = N_1 + N_2$ or $\Delta \le tol$.
    a. Compute $w_i$ and update the location $\mathbf{m} = \frac{\sum_{i=1}^n w_i \mathbf{x}_i}{\sum_{i=1}^n w_i}$.
        • For the MVS estimator: compute $w_i$ from (1.8).
        • For the MVLTS estimator: take $w_i = \begin{cases} 1 & \text{for } d_{(1:n)} \le \dots \le d_{(h:n)} \\ 0 & \text{otherwise} \end{cases}$
    b. If $it > N_1$:
        (1) Set $iter \leftarrow 1$ and $\widehat{s}_0^2 = \widehat{\sigma}^2(\mathbf{d})$ (current squared scale).
        (2) Do until $iter = N_{pc}$ or $\tilde{\Delta} \le tol$
            i. Compute $\mathbf{a}_i$, $i = 1, \dots, n$, $\mathbf{b}_j$ and $m_j$, $j = 1, \dots, p$, using the estimating equations in (1.9)-(1.11).
            ii. Append the vectors $\mathbf{b}_j^{\mathrm{T}}$, $j = 1, \dots, p$, to the rows of $\mathbf{B}_q$ and the vectors $\mathbf{a}_i^{\mathrm{T}}$, $i = 1, \dots, n$ to the rows of $\mathbf{A}_q$.
            iii. Compute new residual distances $d_i = \|\mathbf{r}_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\|$, $i = 1, \dots, n$, from (1.7).
            iv. Set $\widehat{s}^2 = \frac{1}{n}\sum_{i=1}^n d_i^2$.
```

```
                v. Set $iter = iter + 1$, $\tilde{\Delta} \leftarrow 1 - \widehat{s}^2/\widehat{s}_0^2$ and $\widehat{s}_0^2 \leftarrow \widehat{s}^2$.
            (3) End do.
        c. Compute $\mathbf{a}_i^{\mathrm{T}}$, $i = 1, \ldots, n$, using equation (1.9) and append these vectors to
           the rows of $\mathbf{A}_q$.
        d. Compute new residual distances $d_i = \|\mathbf{r}_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\|$, $i = 1, \ldots, n$, from
           (1.7).
        e. Compute $\widehat{\sigma}(\mathbf{d})$:
            • For the MVS estimator: $\widehat{\sigma}(\mathbf{d}) = \widehat{\sigma}_{\mathrm{M}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ from (1.5).
            • For the MVLTS estimator: $\widehat{\sigma}(\mathbf{d}) = \widehat{\sigma}_{\mathrm{LTS}}(\mathbf{d}(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ from (1.27).
        f. Set $\widehat{\sigma}^2 = \widehat{\sigma}^2(\mathbf{d})$.
        g. Set $\Delta \leftarrow 1 - \widehat{\sigma}^2/\widehat{\sigma}_0^2$ and $\widehat{\sigma}_0^2 \leftarrow \widehat{\sigma}^2$.
        h. Set $it = it + 1$.
    3. End do.
```

This algorithm is inspired by the algorithm of Maronna (2005). However, there is an important difference. Maronna (2005) used the weighted covariance matrix in (1.13) to compute the eigenvectors, which reduces to the empirical covariance matrix of the current $h-$subset in case of the MVLTS estimator. We have replaced the computation of eigenvectors from this covariance matrix by an iterative process based on the estimating equations (1.9)-(1.11) in step 2b. This idea is similar to the reweighted least squares algorithm of Boente and Salibian-Barrera (2015) for the coordinatewise S-estimator. Extensive experiments showed that iterating the estimating equations only 2 or 3 times is enough to obtain results close to the eigenvectors of the weighted covariance matrix. Note that computing eigenvectors of a covariance matrix can be very time-consuming in higher dimensions or even unfeasible. On the other hand, our approach only requires vector operations in step 2b, iterated a small number of times, and thus will be more suitable for high-dimensional settings.

The new algorithm yields the same solution as Maronna's algorithm if both algorithms start with the same initial $\mathbf{B}_q$ and the same orthogonal equivariant location estimate $\mathbf{m}$. The reason for the latter condition is that Maronna computes the solution on the orthogonal space. However, in the experiments we have used the spatial median as initial location estimator in our algorithm and the (original) coordinatewise median for Maronna's algorithm. Similarly as in Maronna's algorithm, we initially fixed $\mathbf{B}_q$ for N_1 iterations to improve the initial location estimate. Maronna states that this also ensures orthogonal invariance of the resulting estimates. In the experiments we assessed different choices for the tuning parameters. For the MVS estimator we used the Tukey biweight loss function $\rho(y) = \min(3y^2 - 3y^4 + y^6, 1)$ with tuning parameters $c = 1.54764$, $b = 0.5$ corresponding to the maximal breakdown point of 50% and $c = 3$, $b = 0.2426$ which yields a better compromise between efficiency and robustness. For the MVLTS we considered $\alpha = 0.5$ which trims half of the data with largest orthogonal distance and $\alpha = 0.25$ which trims only a quarter with largest distance. Using a similar proof as in Maronna (2005), it can easily be shown that the M-scale $\widehat{\sigma}_M$ and LTS scale $\widehat{\sigma}_{\text{LTS}}$ decrease in each iteration of our algorithm.

1.4.1 Strategy to find the global minimum

To search the global minimum in (1.4) random starting values are generated and iterated. The best local minimum that is reached is then the approximation for the global optimum. This strategy showed good results in Rousseeuw and Driessen (1999), Maronna (2005) and Salibian-Barrera and Yohai (2006) However, a sufficiently large number of initial points has to be used to obtain a good approximation.

The details of the general strategy to approximate the global minimum are as follows. Take a number N_{cand} of initial candidates, run the above updating algorithm for each of them with parameters N_1, N_2, N_{pc} and *tol*, and keep N_{keep} of the resulting estimates with lowest robust scale $\widehat{\sigma}$. For each of these N_{keep} cases the algorithm continues running with parameters N_1', N_2', N_{pc}' and tol'. The initial location estimate $\mathbf{m}$ is the spatial median of the data matrix $\mathbf{X}$ and the N_{cand} initial $\mathbf{B}_q$'s are random orthogonal matrices. To generate these orthogonal matrices we use the method of Stewart (1980) which consists of orthogonalizing a matrix of normal random numbers. For the tuning parameters we used the same choices as Maronna (2005), that is $N_{\text{cand}} = 50$, $N_{\text{keep}} = 10$, $N_1 = 3$, $N_2 = 2$, $N_1' = 0$, $N_2' = 10$ and $tol' = 0.001$. It sufficed to iterate the estimating equations in step 2b of the algorithm $N_{\text{pc}} = N_{\text{pc}}' = 3$ times to obtain stable results. Note that for Maronna's algorithm we kept the parameter values advocated in his paper and also used the coordinatewise median for the initial location estimator as he proposed.

1.4.2 Strategy to find a good local minimum

As an alternative to searching the global minimum, we adapt the ideas of the deterministic MCD algorithm in Hubert et al. (2012). The rationale is that one could start with a few well-chosen robust starting values that are in the neighborhood of a robust local minimum of the objective function in (1.4). Hence, we attempt to explore only that part of the space that gives good solutions. As a consequence, we do not need many starting values and the convergence may be faster as well, leading to a considerably lower computation time which allows us to handle larger problems. We have adapted five of the deterministic starting values proposed by Hubert et al. (2012) to the context of PCA, such that they can be calculated in high-dimensional problems. We now describe the procedure to obtain these five starting values

I. Standardize each variable X_j, $j = 1, \dots, p$, by substracting its median and dividing by the Q_n scale estimator of Rousseeuw and Croux (1993). The standardized data is denoted by the $n \times p$ matrix $\mathbf{Z}$ with rows $\mathbf{z}_i^{\mathrm{T}}$, $i = 1, \dots, n$, and columns Z_j, $j = 1, \dots, p$.

II. In a first step reduce the effect of potential outliers by one of the following manipulations:

   1) Compute the hyperbolic tangent (sigmoid) of each column Z_j, i.e. $U_{j,1} = \tanh(Z_j)$, $j = 1, \dots, p$. We then form the matrix $\mathbf{U}_1$ with columns $U_{j,1}$, $j = 1, \dots, p$.

   2) Let R_j be the ranks of the column Z_j. Then form the matrix $\mathbf{U}_2$ with columns R_j, $j = 1, \dots, p$.

   3) Compute normal scores from the ranks R_j: $T_j = \Phi^{-1}\left[(R_j - 1/3)/(n + 1/3)\right]$, where $\Phi(.)$ is the normal cumulative distribution function. Then, form $\mathbf{U}_3$ with columns T_j, $j = 1, \dots, p$.

   4) Following the fourth initial scatter estimate of Hubert et al. (2012) that is based on the spatial sign covariance matrix Visuri et al. (2000), we project the data points onto the unit sphere with center $\widehat{\mathbf{m}}$ and define those projections as $\mathbf{u}_{i,4} = \mathbf{z}_i / \left\|\mathbf{z}_i\right\|$, $i = 1, \dots, n$. We then form $\mathbf{U}_4$ with rows $\mathbf{u}_{i,4}^{\mathrm{T}}$, $i = 1, \dots, n$. Note that these are not the usual projected data for computing the spatial sign covariance matrix since $\widehat{\mathbf{m}}$ here is the coordinatewise median instead of the spatial median to make the procedure faster.

   5) Take as rows of $\mathbf{U}_5$ the $\lceil n/2 \rceil$ observations $\mathbf{x}_i$ with smallest euclidean norm of the standardized observations $\mathbf{z}_i^{\mathrm{T}}$. Note that for this case $\mathbf{U}_5$ is a matrix of size $(\lceil n/2 \rceil \times p)$.

III. To further reduce the effect of potential outliers, apply a second step on $\mathbf{U}_k$, $k = 1, 2, 3, 4$, which is similar to 5). We first standardize each column $U_{j,k}$ by substracting its median and dividing by the Q_n scale estimator. Denote this standardized data matrix by $\widetilde{\mathbf{Z}}_k$. Take as rows of the final matrix $\widetilde{\mathbf{U}}_k$ the $\lceil n/2 \rceil$ observations $\mathbf{x}_i^{\mathrm{T}}$ with smallest euclidean norm of the standardized observations $\widetilde{\mathbf{z}}_{i,k}^{\mathrm{T}}$, for $k = 1, 2, 3, 4$. Note that we take $\widetilde{\mathbf{U}}_5 = \mathbf{U}_5$.

IV. For ease of notation let $\mathbf{B}_k = \mathbf{B}_{q,k}$. For each $k = 1, \dots, 5$, obtain initial estimates by computing classical PCA on the data $\widetilde{\mathbf{U}}_k$ with the following iterative procedure:

(1) Start with $\mathbf{B}_k = (\mathbf{e}_1, \ldots, \mathbf{e}_q)$, i.e. the canonical casis, and $\mathbf{m}_k = \frac{1}{n}\sum_{i=1}^{n} \widetilde{\mathbf{u}}_{i,k}$

(2) Compute $\mathbf{a}_{i,k}^{\mathrm{T}} = (\widetilde{\mathbf{u}}_{i,k} - \mathbf{m}_k)^{\mathrm{T}}\mathbf{B}_k$, $i = 1, \ldots, n$, and append these vectors to the rows of the matrix $\mathbf{A}_k$.

(3) Compute residual distances from $\widetilde{\mathbf{u}}_{i,k}$ to the subspace: $d_{i,k} = \|\mathbf{r}_{i,k}(\mathbf{B}_k, \mathbf{A}_k, \mathbf{m}_k)\|$, $i = 1, \ldots, n$.

(4) Set iter $\leftarrow$ 1 and $\widehat{s}_{0,k}^2 = \frac{1}{n}\sum_{i=1}^{n} d_{i,k}^2$.

(5) Do until iter $= N_{pc0}$ or $\tilde{\Delta} \leq tol0$

i. With $\widetilde{\mathbf{U}}_k$ compute $\mathbf{a}_{i,k}$, $i = 1, \ldots, n$, $\mathbf{b}_{j,k}$ and $m_{j,k}$, $j = 1, \ldots, p$, from the estimating equations in (1.9)-(1.11) with weights $w_i = 1$, $i = 1, \ldots, n$.

ii. Append the vectors $\mathbf{b}_{j,k}^{\mathrm{T}}$, $j = 1, \ldots, p$, to the rows of $\mathbf{B}_k$ and the vectors $\mathbf{a}_{i,k}^{\mathrm{T}}$, $i = 1, \ldots, n$ to the rows of $\mathbf{A}_k$.

iii. Compute residual distances from $\widetilde{\mathbf{u}}_{i,k}$ to the subspace: $d_{i,k} = \|\mathbf{r}_{i,k}(\mathbf{B}_k, \mathbf{A}_k, \mathbf{m}_k)\|$, $i = 1, \ldots, n$.

iv. Set $\widehat{s}_k^2 = \frac{1}{n}\sum_{i=1}^{n} d_{i,k}^2$.

v. Set iter $=$ iter $+ 1$, $\tilde{\Delta} \leftarrow 1 - \widehat{s}_k^2/\widehat{s}_{0,k}^2$ and $\widehat{s}_{0,k}^2 \leftarrow \widehat{s}_k^2$.

(6) End do.

V. Use $\mathbf{B}_k$ and $\mathbf{m}_k$, $k = 1, \ldots, 5$, as initial $\mathbf{B}_q$ and initial $\mathbf{m}$ in the algorithm above.

Note that Hubert et al. (2012) used the raw OGK estimator as a sixth initial scatter in their deterministic algorithm to calculate the minumum covariance determinant estimator of multivariate location and scatter. However, it seems not possible to adapt that proposal to obtain initial PCA estimates without having to calculate the full p-dimensional robust covariance estimate of Gnanadesikan and Kettenring (1972). We want to avoid this in high-dimensional data sets, so we discard this proposal from our list of deterministic starts. Note that we used $N_{pc0} = 5$ iterations to calculate the initial deterministic estimates and $tol0 = 0.001$

1.5 Number of components

In some applications the dimension of the linear subspace is known. For instance, users may want to use the estimated PCA to visualize high-dimensional data in much lower dimensions. Most of the time however the number of components are chosen according to the proportion of unexplained variability. At a fixed dimension q, the best linear subspace is the one that attains the smallest possible unexplained variance u_q. At the true distribution G of $\mathbf{x} \in \mathbb{R}^p$ this minimal variance is attained by the eigenvectors $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(q)}$ of the underlying scatter matrix Σ which yields

$$u_q = \frac{\sum_{j=q+1}^{p} \lambda_j}{\sum_{j=1}^{p} \lambda_j}, \tag{1.45}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$ are the corresponding eigenvalues. Maronna (2005) proposed to estimate the proportion of unexplained variability by:

$$\widehat{u}_q = \frac{\widehat{\sigma}_q^2}{\widehat{\sigma}_0^2}. \tag{1.46}$$

In the case of the MVS estimator $\widehat{\sigma}_q$ is the S-scale estimate, i.e. the M-scale estimate corresponding to the MVS estimates in (1.4), while $\widehat{\sigma}_0$ is the minimum of $\widehat{\sigma}_{\mathrm{M}}(\mathbf{d}_0(\mathbf{m}))$ over all $\mathbf{m} \in \mathbb{R}^p$, with $\mathbf{d}_0(\mathbf{m}) = (\|\mathbf{x}_1 - \mathbf{m}\|, \|\mathbf{x}_2 - \mathbf{m}\|, \ldots, \|\mathbf{x}_n - \mathbf{m}\|)$. For the MVLTS estimator $\widehat{\sigma}_q$ is the LTS scale estimate that corresponds to the MVLTS estimator in (1.28) while $\widehat{\sigma}_0^2$ is the minimum of $\widehat{\sigma}_{\mathrm{LTS}}^2(\mathbf{d}_0(\mathbf{m}))$ over all $\mathbf{m} \in \mathbb{R}^p$. Note that in both cases $\widehat{\sigma}_0^2$ is a squared robust scale estimate for the cases that no principal components are fitted and thus yields an estimate of the total variance in the data. Proposition 2.2 in Maronna (2005) can be used to show that $\widehat{u}_q$ consistently estimates u_q.

We now propose a strategy to choose the dimension q of the subspace that is an adaptation from the approach in Maronna (2005) and is very similar to the strategy in Boente and Salibian-Barrera (2015) for the coordinatewise S-PCA estimator. Let $u_{\max}$ be the maximum proportion of unexplained variability that the problem allows. Denote as $q_{\max}$ the maximum dimension of the subspace that we are willing to accept. We look for the smallest q such that $q \leq q_{\max}$ and $\widehat{u}_q \leq u_{\max}$. This goal could be attained by solving (1.4) or (1.28) for $q_{\max}$, $q_{\max} - 1$, and so forth, but this may be time-consuming. We now describe a marginal strategy to solve (1.4) or (1.28) when increasing q by 1, which is much faster.

We first run the algorithm with $q = 1$. If $\widehat{u}_{q=1} \leq u_{\max}$ then we are done. Otherwise, take the solutions $\widehat{\mathbf{B}}_1 \in \mathbb{R}^{p\times 1}$, $\widehat{\mathbf{A}}_1 \in \mathbb{R}^{n\times 1}$ and $\widehat{m}_j^{(1)}$ obtained for $q = 1$ and proceed as follows. Let $r_{ij}^{(1)}$ be the corresponding elementwise residuals, $j = 1, \ldots, p$, $i = 1, \ldots, n$. Set $q = 2$ and define the matrices $\mathbf{B}_2 = (\widehat{\mathbf{B}_1, \mathfrak{B}}) \in \mathbb{R}^{p\times 2}$ with $\mathfrak{B} = (\mathfrak{b}_1, \ldots, \mathfrak{b}_p)^T$ and $\mathbf{A}_2 = (\widehat{\mathbf{A}}_1, \mathfrak{A}) \in \mathbb{R}^{n\times 2}$ with $\mathfrak{A} = (\mathfrak{a}_1, \ldots, \mathfrak{a}_n)^T$. The corresponding predictions are then obtained by $\hat{x}_{ij}^{(2)} = \hat{x}_{ij}^{(1)} + \mathfrak{b}_j \mathfrak{a}_i$ with residual distance $\left\|\mathbf{r}_i^{(2)}\right\| = \sqrt{\sum_{j=1}^{p} \left(r_{ij}^{(1)} - \mathfrak{b}_j \mathfrak{a}_i\right)^2}$. The marginal optimization problem now becomes

$$\min_{\widehat{\mathbf{B}}_1^{\mathrm{T}} \mathfrak{B} = \mathbf{0},\, \mathfrak{A}} \widehat{\sigma}_{\mathrm{M}} \left(\left\|\mathbf{r}_1^{(2)}(\mathfrak{B}, \mathfrak{A})\right\|, \ldots, \left\|\mathbf{r}_n^{(2)}(\mathfrak{B}, \mathfrak{A})\right\| \right), \tag{1.47}$$

over $\mathfrak{B}, \mathfrak{A}$, with $\widehat{\mathbf{B}}_1^{\mathrm{T}}\mathfrak{B} = \mathbf{0}$. For the MVS estimator a system of estimating equations analogous as in Section 1.2 can be derived and then used in the iterative algorithm of Section 1.4 to find a solution. The MVLTS solution can be found by using 0-1 weights in the estimating equations of the iterative algorithm. Once the optimal solutions $\widehat{\mathfrak{B}}$ and $\widehat{\mathfrak{A}}$ according to (1.47) are found, we optimize the robust scale of the residual distances w.r.t. $\mathbf{m}$ to obtain an updated estimate $\widehat{\mathbf{m}}^{(2)}$. This marginal approach is faster than directly solving (1.4) or (1.28) for $q = 2$. With $\widehat{\mathbf{B}}_2 = (\widehat{\mathbf{B}}_1, \widehat{\mathfrak{B}})$ and $\widehat{\mathbf{A}}_2 = (\widehat{\mathbf{A}}_1, \widehat{\mathfrak{A}})$, we then compute $\widetilde{u}_2 = \frac{\widehat{\sigma}_{\mathrm{M}}(\widehat{\mathbf{B}}_2, \widehat{\mathbf{A}}_2, \widehat{\mathbf{m}}_2)}{\widehat{\sigma}_0^2}$. If $\widetilde{u}_2 \leq u_{\max}$ the procedure stops. Otherwise, we increase q by 1 and repeat the procedure until $\widetilde{u}_q \leq u_{\max}$ or until $q = q_{\max}$. Note that if $\widehat{u}_{q_{\max}} > u_{\max}$, then the chosen $q_{\max}$ does not allow to explain enough of the variance in the data and we will have to modify our goals (increase $q_{\max}$ or $u_{\max}$). Note that the quantity $\widetilde{u}_q$ is typically larger than $\widehat{u}_q$ so that we make a safe choice for q when $\widetilde{u}_q \leq u_{\max}$.

1.6 Simulation study

We want to assess the performance of our iterative algorithms to calculate the MVS-PCA and MVLTS-PCA estimators based on the estimating equations. We consider both strategies for the starting values. Either the algorithm starts with several random orthogonal matrices or its starts with the five well-chosen deterministic starting solutions as described in Section 1.4. We compare our algorithms for the MVS-PCA and MVLTS-PCA estimators with the S-M and S-L algorithms of Maronna (2005). Our algorithms with random orthogonal matrices are therefore expected to give similar results as those of Maronna. However, we expect that our algorithms can be computed faster in high-dimensional settings.

Moreover, we also compare with other methods to estimate the q-dimensional subspace. In particular, we consider the Projection pursuit (PP) method of Li and Chen (1985), the Spherical PCA of Locantore et al. (1999) and the classical PCA. To implement the Projection pursuit estimator in our experiments we used the approximate algorithm of Croux and Ruiz-Gazen (1996, 2005). We consider three different scales that are maximized, the modified MAD (used by Croux and Ruiz-Gazen), the M-scale of the absolute deviations from the median (used in Maronna (2005)) with the Tukey's biweight function ($c = 1.54764$ and $b = 0.5$) and the LTS scale of the absolute deviations from the median with $\alpha = 0.5$. The two latter variants thus use the same scale estimators as the MVS and MVLTS methods. We denote these procedures as PPMD (PP with MAD scale), PPME (PP with M-scale) and PPLTS (PP with LTS scale), respectively.

To investigate the robustness of these methods at finite samples we replicate the simulations in Maronna (2005). Hence, we generate $M = 200$ samples of size $n = 100$ and dimension $p = 10$, where $n - \lfloor n\epsilon \rfloor$ of the data are generated by the model distribution $N(\mathbf{0}, \mathbf{\Sigma})$, with $\mathbf{\Sigma} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$. Two designs of diagonal elements for $\mathbf{\Sigma}$ were considered as in Maronna (2005) that represents:

a) an abrupt increase of the eigenvalues: $\lambda_j = 1 + 0.1j$ for $1 \leq j \leq (p - q)$ and $\lambda_j = 20(1 + 0.5(j - p + q))$ for $(p - q + 1) \leq j \leq p$.

b) a smooth increase of the eigenvalues: $\lambda_j = 2^{j-1}$ for $1 \leq j \leq p$.

The remaining $\lfloor n\epsilon \rfloor$ of the data are outliers which are generated from $N(k\mathbf{x}_0, 0.25\mathbf{\Sigma})$, where $\mathbf{x}_0$ is a vector of length p with $x_{0j} = 1$ for $j \leq (p - q)$ and 0 otherwise. The value of k runs between 0 and 20 with steps of 0.5. Fractions $\epsilon = 10\%$ and $\epsilon = 20\%$ of outliers are considered. We also consider the scenario with only regular data, i.e. $\epsilon = 0\%$. In all experiments the fitted PCA techniques try to estimate the best linear subspace of dimension $q = 2$.

Performance measures

As performance criterion for an estimator $\widehat{\mathbf{B}}_q$ we use a predictive approach analogous to Maronna (2005). Essentially, we measure the proportion of variance in independent regular data that remains unexplained by the estimated subspace. More formally, let $\mathbf{x}$ be a $N(\mathbf{0}, \mathbf{\Sigma})$ vector independent of the random sample used to obtain $\widehat{\mathbf{B}}_q$. Then, the variability of $\mathbf{x}$ around the subspace generated by $\widehat{\mathbf{B}}_q$ is

$$E\|\mathbf{x} - \widehat{\mathbf{B}}_q\widehat{\mathbf{B}}_q^{\mathrm{T}}\mathbf{x}\|^2 = \text{tr}\big[\mathbf{\Sigma}\big] - \text{tr}\big[\widehat{\mathbf{B}}_q^{\mathrm{T}}\mathbf{\Sigma}\widehat{\mathbf{B}}_q\big],$$

and the prediction proportion of unexplained variance is:

$$u_q^{\text{pred}} = \frac{E\|\mathbf{x} - \widehat{\mathbf{B}}_q\widehat{\mathbf{B}}_q^{\mathrm{T}}\mathbf{x}\|^2}{\text{tr}\big[\mathbf{\Sigma}\big]} = 1 - \frac{\text{tr}\big[\widehat{\mathbf{B}}_q^{\mathrm{T}}\mathbf{\Sigma}\widehat{\mathbf{B}}_q\big]}{\text{tr}\big[\mathbf{\Sigma}\big]}. \tag{1.48}$$

Note that for the Maronna's S-M and S-L methods characterize the subspace by an estimate $\widehat{\mathbf{B}}_{p-q} \in \mathbb{R}^{p\times(p-q)}$ of its orthogonal complement. Therefore, in this case the corresponding prediction proportion of unexplained variability becomes:

$$u_q^{\text{pred}} = \frac{\text{tr}\big[\widehat{\mathbf{B}}_{p-q}^{\mathrm{T}}\,\mathbf{\Sigma}\,\widehat{\mathbf{B}}_{p-q}\big]}{\text{tr}\big[\mathbf{\Sigma}\big]}. \tag{1.49}$$

We want to keep u_q^{pred} as small as possible. At a fixed dimension q the lowest possible proportion of unexplained variability obtained by the best linear subspace is given by (1.45). The performance of the subspace generated by the estimate $\widehat{\mathbf{B}}_q$ is then compared to the best subspace with a measure of relative prediction error:

$$e_{\text{pred}} = \frac{u_q^{\text{pred}}}{u_q^{\text{opt}}} - 1. \tag{1.50}$$

While the dimension q of the subspace is chosen according to the proportion of unexplained variability in many applications, its quantity given by (1.48) or (1.49) cannot be obtained in practice since we do not know the scatter matrix Σ that generated the data. Hence, we need to estimate the proportion of unexplained variance. For the MVS-PCA, MVLTS-PCA, S-M and S-L procedures we can obviously estimate (1.48) or (1.49) by (1.46). For the PP, spherical PCA and classical PCA we can estimate u_q^{pred} by

$$\widehat{u}_q = \widehat{u}_q(\widehat{\lambda}) = \frac{\sum_{j=q+1}^{p} \widehat{\lambda}_j}{\sum_{j=1}^{p} \widehat{\lambda}_j},$$

where $\widehat{\lambda}_j$, $j = 1, \ldots, p$, are the eigenvalues or variances as estimated by the respective methods. Clearly, we do not want too small values or too large values of $\widehat{u}_q$ since that would lead to under-estimating or over-estimating the dimension of the subspace. Thus, similarly as in Maronna (2005), we also measure the relative estimation error of $\widehat{u}_q$ by:

$$e_{\text{est}} = \max\left(\frac{\widehat{u}_q}{u_q^{\text{pred}}}, \frac{u_q^{\text{pred}}}{\widehat{u}_q}\right) - 1.$$

Results

Table 1.1 shows the mean relative prediction errors $\bar{e}_{\text{pred}}$ over $M = 200$ samples for the methods that showed the best performance throughout the different scenarios analyzed. Although the PP methods do not perform well in these scenarios, we also included the results of the PPLTS method in Table 1.1 for comparison purposes. The values of k that have been included in Table 1.1 are those values at which some estimators attain their maximum, i.e. their worst performance. More detailed results for all methods can be found in Tables B.1-B.1 in the Appendix A. For $\epsilon = 20\%$ of contamination the evolution of the mean relative prediction error with the outlier distance k is displayed in Figures 1.1 and 1.2. From $k = 10$ onwards these prediction errors stabilize so we only show results up to $k = 10$. To make the plots easier to read, the results for the second design are displayed in two panels in Figure 1.2. Only techniques with good performance are

Table 1.1: Mean relative prediction errors $\overline{e}_{pred}$ for the techniques with best performance and for PPLTS

Design	ϵ	k	PPLTS	S-M	MVS	S-L	MVLTS	MVS	MVLTS
				(c=1.5)	(c=1.5)	(α=0.25)	(α=0.25)	det(c=1.5)	det(α=0.5)
a)	0	0	0.32	0.02	0.02	0.04	0.04	0.02	0.06
	10%	1	0.41	0.03	0.03	0.04	0.04	0.03	0.08
		2.5	0.62	0.03	0.03	0.03	0.03	0.03	0.07
		20	0.58	0.03	0.03	0.03	0.03	0.03	0.07
	20%	1.5	0.78	0.03	0.03	0.08	0.07	0.03	0.29
		5	1.74	0.03	0.03	0.03	0.03	0.03	0.06
		20	0.65	0.03	0.03	0.03	0.03	0.03	0.06
b)	0	0	0.27	0.04	0.04	0.06	0.06	0.04	0.11
	10%	1.5	0.40	0.09	0.08	0.14	0.12	0.08	0.14
		2	0.46	0.10	0.09	0.11	0.09	0.07	0.12
		4	0.49	0.05	0.05	0.06	0.06	0.05	0.11
	20%	2	0.78	0.67	0.66	0.68	0.67	0.38	0.35
		3	0.73	0.71	0.71	0.66	0.67	0.27	0.15
		5	0.57	0.69	0.69	0.17	0.17	0.07	0.11
		19	0.30	0.04	0.04	0.05	0.05	0.05	0.11

shown in the plots. Plots for $\epsilon = 10\%$ are not shown because for most methods there is much less variability in this case.

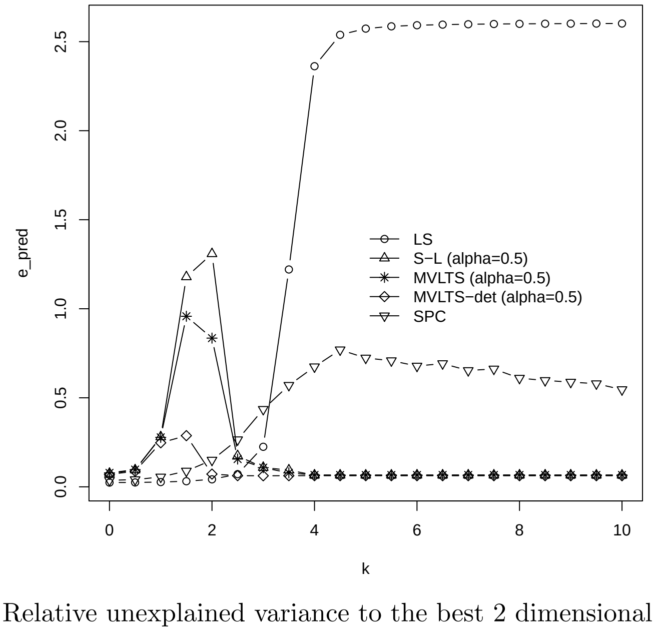


Figure 1.1: Relative unexplained variance to the best 2 dimensional subspace ($\overline{e}_{\text{pred}}$) as a function of k for eigenvalue configuration a) and $\epsilon = 20\%$.

The classical PCA (LS) shows the best performance when there are only regular data ($\epsilon = 0\%$), but as expected it breaks down when outliers are introduced and then becomes the worst technique by far. PPLTS shows some advantage in comparison to the other PP approaches but in general all PP approaches give poor results in these settings.

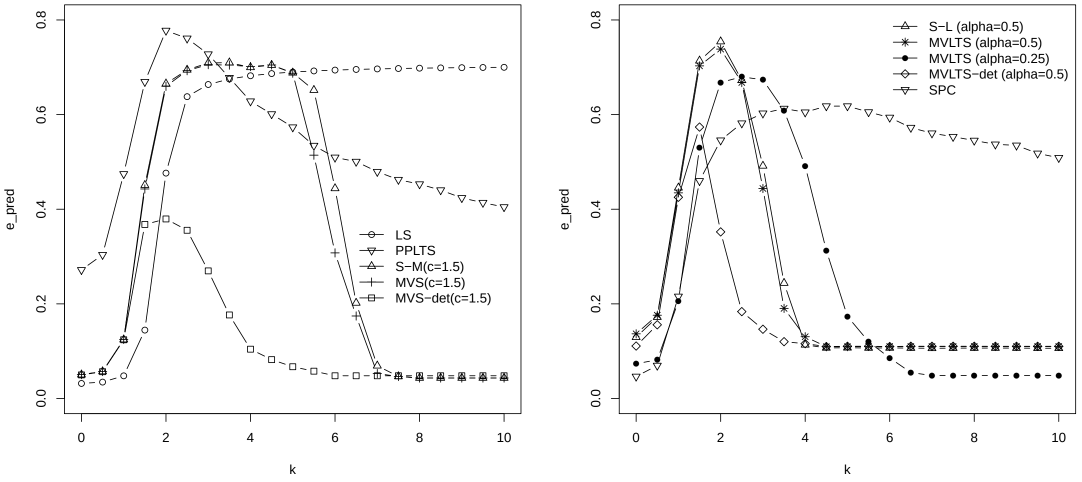


Figure 1.2: Relative unexplained variance to the best 2 dimensional subspace ($\bar{e}_{\text{pred}}$) as a function of k for eigenvalue configuration b) and $\epsilon = 20\%$.

Spherical PCA is in general better than the PP approaches even though its performance decreases considerably when $\epsilon = 20\%$ (see Table B.2).

The S-L and S-M algorithms of Maronna (2005) yield very similar results to our MVS and MVLTS counterparts when using random orthogonal matrices. This can be seen for instance in Figures 1.1 and 1.2 for the S-L and MVLTS methods with $\alpha = 0.5$ and in Figure 1.2 for the S-M and MVS methods with $c = 1.5$. therefore, by referring to S-PCA and LTS-PCA we mean the MVS and MVLTS estimators, either calculated with Maronna's algorithm or with our algorithm using random orthogonal matrices. LTS-PCA with $\alpha = 0.5$ shows very good results. On the other hand, S-PCA with $c = 3$ breaks down when $\epsilon = 20\%$ and performs as bad as the classical PCA in these scenarios. In general, S-PCA with $c = 1.5$ and LTS-PCA with $\alpha = 0.25$ both give excellent results.

Table 1.2: Mean estimation errors e_{est} for data without contamination

	Design a)	Design b)
	$\epsilon = 0\%$	$\epsilon = 0\%$
LS	0.09	0.11
PPMD	0.11	0.12
SPC	0.17	0.16
MVS (c=3)	0.07	0.10
MVS (c=1.5)	0.12	0.12
MVLTS ($\alpha = 0.5$)	0.35	0.29
MVLTS ($\alpha = 0.25$)	0.19	0.15
MVS-det. (c=3)	0.07	0.10
MVS-det. (c=1.5)	0.12	0.12
MVLTS-det ($\alpha = 0.5$)	0.36	0.25

The MVS and MVLTS algorithms with deterministic starts aim to find a robust local minimum. The simulation results show that in both designs the algorithms succeed well in this. The MVS algorithm with deterministic starting values and $c = 1.5$ (MVS-det,

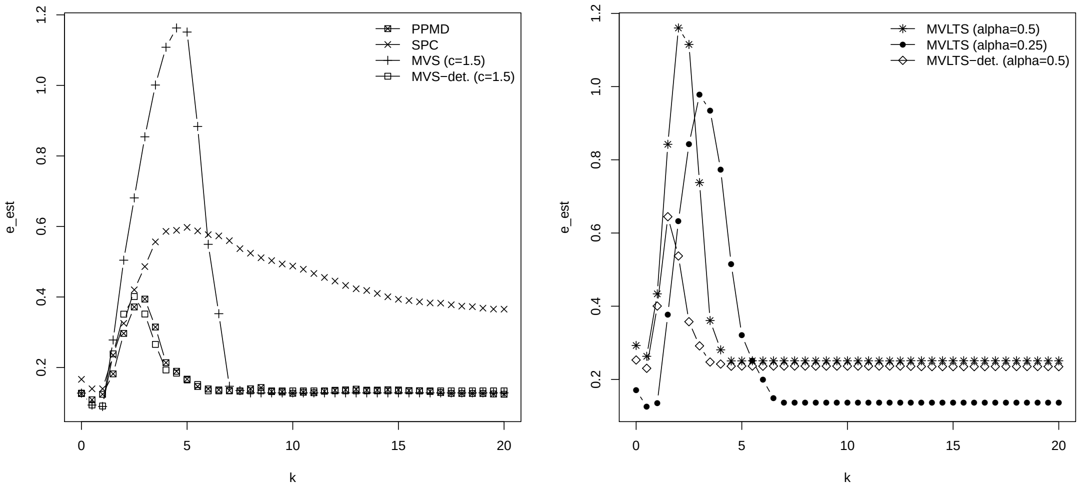


Figure 1.3: Mean estimation error $\overline{e}_{est}$ as a function of k for eigenvalue configuration b) and $\epsilon = 20\%$.

c=1.5) clearly performs best in all scenarios explored. This can be seen for example for design b) with $\epsilon = 20\%$ in Figure 1.2. Results of MVS-det (c=1.5) are omitted from Figure 1.1 since they form almost a straight line with estimates very close to 0. However, these results are shown for all scenarios and some values of k in Table 1.1. In general when outliers are close or at a moderate distance of the clean data (i.e. small or moderate k values) we find that it is more beneficial to start the algorithm with deterministic starting values than with random orthogonal matrices. Most robust PCA techniques do not succeed in correctly identifying outliers when they are close to the regular data. These simulation results suggest that our algorithm with deterministic starts is not only faster, but it can also better discriminate between outliers and good data. Even for the very hard design b) with $\epsilon = 20\%$ (Figure 1.2) this approach avoids starting with outliers in more cases than when using random orthogonal matrices.

Figure 1.3 shows the results of mean estimation errors of $\widehat{u}_q$ as a function of the outlier distance k for design b) with $\epsilon = 20\%$ of contamination. The results for the other scenarios analyzed are pretty similar. We again show results up to $k = 10$ since the errors stabilize after that k level. S-M and S-L yield similar results to MVS and MVLTS, respectively, when using random orthogonal matrices, so they are not shown in Figure 1.3. Likewise, all projection pursuit methods showed similar behavior, so we only show the results for PPMD. As can be seen in table 1.2, Classical PCA is one of the best techniques to estimate u_q^{pred} for data without outliers. However, it shows very high mean estimation errors when outliers are present, as expected. With outliers at moderate or large distance to the regular data, the estimation error of classical PCA reaches values up to 20 and so they are not shown in Figure 1.3. With $\epsilon = 20\%$, also S-PCA with $c = 3$ (results not shown) performs poorly for estimating u_q^{pred} and yields results as bad as classical PCA for large k values. Spherical PCA (SPC) estimates u_q^{pred} comparatively well only for sufficiently large k values. It can clearly be seen from Figure 1.3 that MVS with deterministic starts and c=1.5 as well as projection pursuit (PPMD) both are able to estimate u_q^{pred} satisfactorily, even when outliers are very close to the regular data. The MVLTS algorithm with deterministic starting values and $\alpha = 0.5$ also performs quite well. Therefore, our algorithm with deterministic starting values, next to the aforementioned advantages, is also able to accurately estimate the amount of unexplained variance of the model. This means that at least in these settings our algorithm with deterministic starts can effectively choose the dimension of the subspace based on the estimator $\widehat{u}_q$ in (1.46).

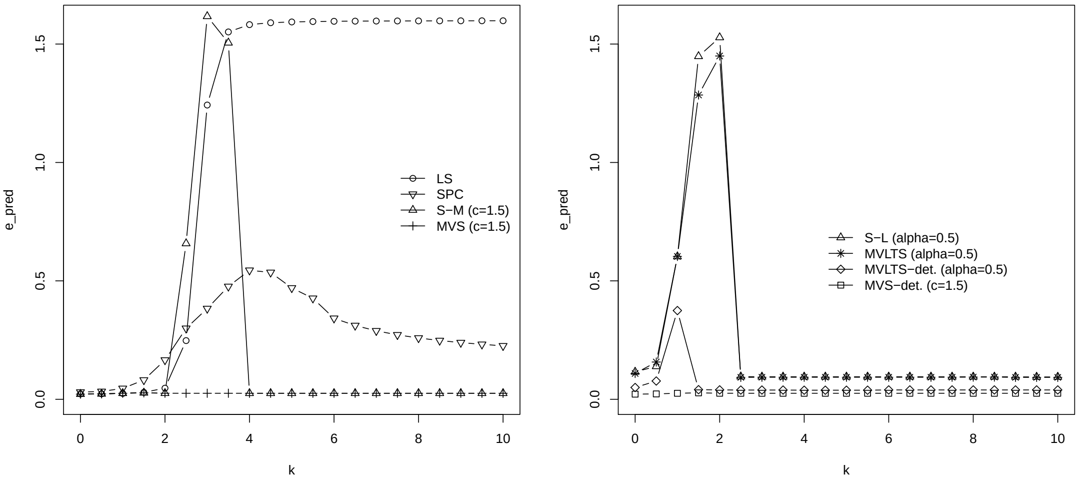


Figure 1.4: Mean relative prediction errors $\bar{e}_{pred}$ for the simulations with $p = 750$ and $\epsilon = 20\%$ of contamination

1.7 Simulations with high dimensional data

We now want to see if the performance results of the previous section still hold when we go to a high-dimensional setting. We therefore consider a data generating model similar to design a) of the previous section. More specifically, we generated clean data from the same model distribution. We keep $n = 100$ fixed but increase the dimension to $p = 200$, $p = 500$ or $p = 750$. As in the previous experiments we consider a $q = 2$ dimensional subspace estimation. To generate eigenvalues we slightly modified design a) to ensure that the two main directions of $\mathbf{\Sigma}$ explain about 80% of the total variability. We therefore used:

a) For $p = 200$:
$\lambda_j = 1+.001j$ for $1 \leq j \leq (p-q)$ and $\lambda_j = 20(1+0.5(j-p+q))$ for $(p-q+1) \leq j \leq p$.

b) For $p = 500$:
$\lambda_j = 1 + .00015j$ for $1 \leq j \leq (p - q)$ and for $(p - q + 1) \leq j \leq p$ the eigenvalues λ_j are the same as a).

c) For $p = 750$:
$\lambda_j = 1 + .00007j$ for $1 \leq j \leq (p - q)$ and for $(p - q + 1) \leq j \leq p$ the eigenvalues λ_j are the same as a).

For these simulations we took the same model to generate contaminated data as in the previous section and fixed $\epsilon = 20\%$. To assess the performance of the PCA methods we used the average of relative prediction errors computed from (1.50) over $M = 200$ replications. All three high-dimensional cases analyzed showed similar results so we only present the results for $p = 750$ in Figure 1.4. Relative prediction errors stabilize from $k = 10$ onwards so we only present results up to $k = 10$. Results of the projection-pursuit approaches, the LTS-PCA methods with $\alpha = 0.25$ and the MVS-PCA methods with $c = 3$ have a similar behaviour as in the low dimensional case so they are not displayed in Figure 1.4. We immediately see that in general, prediction errors are larger as compared to the low dimensional case when the contaminated data is close to the clean data (small or moderate k values). Classical PCA as expected breaks down with outliers. The results of spherical principal components (SPC) are similar to the low dimensional scenario (see Figure 1.2) but in this case they look relatively competitive given the complexity of the problem. Maronna methods still do not show an advantage over our procedures based on random orthogonal starts. The performance of S-M even deteriorates for moderate k values while MVS carries its excellent performance from the low dimensional to the high-dimensional setting. We also notice that our algorithm with deterministic starting values has an excellent performance in this setting as well. In particular, the performance of MVLTS with deterministic starts do not decrease as compared to the low-dimensional case when the contaminated data is close to the clean data while S-L and MVS with random orthogonal starts do show a decrease in performance. MVS with deterministic starts performs equally well as the MVS algorithm with random orthogonal starts.

1.8 Computational time

We now compare the computational time of our algorithms with those of the algorithms of Maronna. For this purpose we used the low-dimensional and the high-dimensional experiments described in previous sections in which we kept the size of the data fixed to $n = 100$ while we let the dimension grow. The estimators were implemented in the

R statistical software and run on a single Intel i7 CPU (3.4GHz) machine running Windows 7. The average of computational times in seconds over $M = 200$ replications are summarized on Table 1.3 as dimension p increases. Note that $p = 10$ refers to the original design a) with $\epsilon = 20\%$ of contamination. Not surprisingly, we see that the spherical PCA implementation is the fastest by far while PPLTS is the slowest in higher dimensions. Projection-pursuit calculates one direction at a time which is a disadvantage for computational time issues. Methods with the Maronna algorithm (S-M and S-L) can be computed fast in problems with small dimensions since it only needs a few iterations to give good results. However, as soon as we go to a larger dimension the algorithm becomes the second and the third slowest in the comparisons of Table 1.3. This result was expected since the algorithm of Maronna computes the last $p - q$ eigenvectors of a covariance matrix which is very time-consuming in higher dimensions. We carried out additional experiments in which we let p increase even more and after $p = 1500$ it was already not possible to compute these directions. On the other hand, our algorithm with random orthogonal matrix is the slowest for small dimensions ($p = 10$), but it becomes faster in relation to the algorithm of Maronna after $p = 600$. Our algorithm of section 1.4 replaces the computation of eigenvectors from a covariance matrix with simple vector operations from the estimating equations in (1.9)-(1.11) which consequently makes the whole algorithm faster. In these experiments however our algorithm with five deterministic starting values yields the best tradeoff between performance and computational speed. It does not only show the best performance in the low-dimensional experiments of section 1.6 but also in the high-dimensional ones of section 1.7 while keeping its fairly low computational time. These methods are also able to accurately estimate the amount of unexplained variance of their models and therefore they can effectively choose the dimension of the subspace based on the estimator $\widehat{u}_q$ in (1.46). This shows that five robust starts are enough to stay in the neighborhood of a robust local solution for these experiments and we do not need to spend more time looking in other parts of the space. Overall, deterministic starting values with MVS shows better performance than with MVLTS in these experiments and it only requires a few more seconds of computational time.

Table 1.3: Computational time in seconds as the dimension p increases

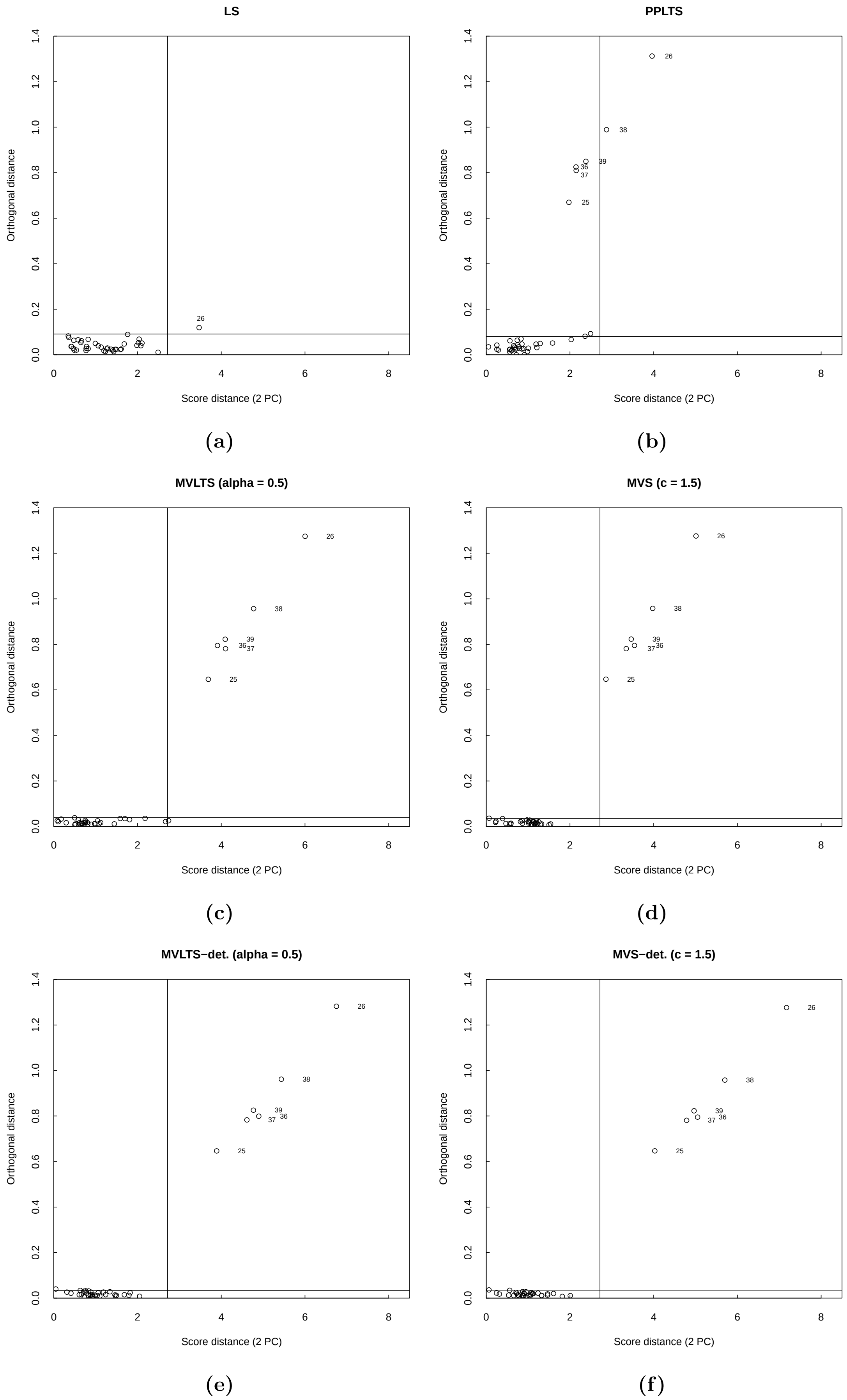
	p=10	p=200	p=500	p=750
PPLTS	0.58	22.64	89.27	196.59
SPC	0.02	0.05	0.09	0.12
S-M	0.44	7.25	49.22	140.40
S-L	0.32	6.16	45.72	130.17
MVS	2.87	22.99	66.96	129.82
MVLTS	2.80	10.81	37.83	82.90
MVS-det.	0.75	5.95	12.22	20.70
MVLTS-det.	0.76	2.28	4.89	7.07

1.9 Real data example

In this section we illustrate the performance of our MVS and MVLTS algorithms on the Octane dataset introduced in Hubert et al. (2005). In particular, it consists of near-infrared (NIR) absorbance spectra of $n = 39$ gasoline samples with certain octane numbers over $p = 226$ wavelengths. Hence, this is a high-dimensional data with $p >> n$. It is well known that six of the samples contain added alcohol, so that they are potential outliers. These are observations 25, 26, and 36-39. With $q = 2$ components classical PCA explains 98% of the total variability while the considered robust PCA techniques explain more than 96% of the total variability. Thus, as in Hubert et al. (2005), we retain a 2 dimensional subspace.

Figure 1.5 shows the diagnostic plots corresponding to six different PCA estimates to reduce the data to dimension $q = 2$. The six considered methods are classical PCA (LS), projection pursuit with LTS scale (PPLTS), MVLTS with orthogonal random starts and MVLTS with deterministic starts (both with $\alpha = 0.5$) and MVS with orthogonal random starts and MVS with deterministic starts (both with $c = 1.5$). A diagnostic plot for PCA was introduced by Hubert et al. (2005) and is a very popular tool to identify outliers in principal component analysis with high-dimensional data. Essentially, it computes orthogonal distances from the observations to the estimated subspace as well as robust distances in the subspace in order to identify three types of outliers. An observation with small orthogonal distance to the subspace but far from the regular data within the subspace is called a *good leverage point*. Moreover, an observation is called an *orthogonal outlier* if it lies far from the subspace, but its projection on the subspace is close to the typical projections. The worse types of outliers are the so called *bad leverage points* which are observations that lie far from the subspace and have projections that are also remote from the regular points in the subspace. Hubert et al. (2005) proposed cutoff values for both the robust orthogonal distances and the robust score distances that allow to identify unusually large distances. Computing the robust score distances requires a robust estimate for the variances according to the basis directions within the subspace. Since our algorithm yields the basis directions of the subspace and corresponding scores of the data, but does not yield estimates of the variability, we estimate these variances robustly by computing univariate LTS or M-scales of the scores corresponding to these directions.

We focus on the six alcohol samples which are potential outliers. The classical diagnostic plot in Figure 1.5a shows that classical PCA only identifies observation 26 as mildly outlying. This observation only falls just above the cutoff lines which suggests that the six alcohol samples do not deviate from the other observations. On the other hand, the diagnostic plots for the five robust PCA methods show a completely different



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 1.5: Diagnostic plots of the Octane dataset based on six two-dimensional PCA estimates.

picture. All the robust PCA methods identify the six samples with added alcohol as outliers. In particular, projection pursuit with LTS scale identifies observations 25, 36, 37, 39 as orthogonal outliers while observations 26 and 38 are flagged as bad leverage points. MVLTS methods and MVS methods flag the six samples with added alcohol as bad leverage points that lie far from the two-dimensional subspace which corresponds to the conclusions of Hubert et al. (2005). Note that while all robust methods yield approximately the same orthogonal distances for the 6 alcohol samples, the score distances are not equally large for all methods. In particular, the computationally fast algorithms with deterministic starts also exhibit large score distances such that the samples with added alcohol are most clearly identified as bad leverage points in this case.

1.10 Discussion and conclusions

In this Chapter we discussed two methods that aim to estimate the best lower-dimensional subspace in a robust way, namely the Multivariate S-estimator (MVS) and the Multivariate LTS estimator (MVLTS). These methods were introduced by Maronna (2005). We refer to them as multivariate methods since they look at entire observations by minimizing a robust scale of the residual norms. MVS minimizes a M-scale and MVLTS minimizes a LTS scale of the residual norms. We introduced the corresponding functionals and showed that they are Fisher-consistent at elliptical model distributions. We also studied the robustness properties of the MVS-PCA estimator by deriving the influence functions which turn out to be bounded for outliers w.r.t. the subspace and smoothly redescends to zero for the non-diagonal elements of the functional. Good leverage points may have a large influence on the estimator. In the last part of this chapter we proposed an iterative algorithm for both methods which uses the corresponding estimating equations derived from first order conditions to update the directions. This algorithm is suitable for high-dimensional problems since we only compute vector operations from the estimating equations. For the starting values of the algorithm we considered two choices. The first uses random orthogonal matrices as in Maronna (2005) and aims to find the global minimum. The second uses a few well-chosen robust starting values and then finds the best local minimum that can be obtained from these initial robust solutions. Our algorithm with deterministic starts can be computed faster since we do not need many starting values and we do not need many iterations before the algorithm converges. This algorithm can therefore allow us to handle larger problems. Experiments with low and high-dimensional data confirmed a lower computational time of our algorithm with deterministic starts when compared to the algorithm of Maronna (2005) or with our algorithm starting with random orthogonal matrices. These simulations also show that our algorithm with random orthogonal matrices yields very close results to the algorithm of Maronna and it shows better results than other robust procedures like projection-pursuit and spherical principal components. However, starting the algorithm with deterministic starting values gives better results, even in the complicated scenario where outliers are close to the regular data. In particular, the MVS algorithm with deterministic starts and $c = 1.5$ clearly performs best. Additional experiments suggest that our algorithm with deterministic starting values, next to the aforementioned advantages, is also able to accurately estimate the amount of unexplained variability of the model and to carry its excellent performance to high-dimensional settings. We closed with an example that used a high-dimensional real dataset. The example confirmed the good performance of our algorithm, in particular when it starts with deterministic starting values.

Chapter 2

Coordinatewise subspace estimation for high-dimensional data

The content of this chapter is work in progress for future publication. This was a joint work with Prof. Matias Salibian-Barrera from the University of British Columbia (Canada).

2.1 Introduction

For many years robust statistics has devoted its attention to the case where a majority of the observations is regular while the remaining minority may be atypical. Therefore, most of the existing robust methods in any context aim to identify the minority of outlying cases and downweight them. The Tukey-Huber contamination model is the standard contamination model that describes this contamination pattern. More specifically, the Tukey-Huber contamination model assumes that a large fraction $(1-\epsilon)$ of the data is generated from a postulated statistical model with well-behaved random noise while the remaining fraction may be affected by abnormal noise that is left unspecified. However, this paradigm may not be satisfactory for modern high-dimensional datasets. In particular, when $p > n$ even a small percentage of outlying cells can affect a large percentage of observations when the cells are contaminated at random. Thus, in high-dimensional data the fraction of observations that are completely free of contamination can become very small and downweighting entire observations can be wasteful if only a small part of the cells of an observation are actually contaminated. Random outlying cells can thus be devastating for any affine equivariant high-breakdown estimator since they cannot handle more than 50% of contaminated observations. Consider for example a data generating model with a $(0-1)$ contamination indicator variable O_j for every feature j, where $O_1, O_2, \ldots, O_p$ are independent. Consider the case where every variable has the same probability ϵ for a contaminated measurement, i.e. $\Pr(O_1 = 1) = \ldots = \Pr(O_p = 1) = \epsilon$. Then, the probability of having a completely clean observation under this model is only $(1-\epsilon)^p$. Thus, even for small values of ϵ this probability quickly decreases as p increases, and for large values of p this probability lies well below the critical value 0.5. For example, for $\epsilon = 0.01$, the probability lies below 50% for $p \geq 69$. Alqallaf et al. (2009) called this contamination model the fully independent contamination model (FICM). However, most of the contributions in robust statistics have targeted the problem of casewise outliers, i.e. observations are either regular or outlying. Alqallaf et al. (2009) investigate the performance and theoretical properties of such robust estimators of multivariate location for the FICM model. They showed that these methods lose their robustness in presence of cellwise outliers. Therefore, in the last five years, some methods targeting problems with cellwise outliers have been proposed. Van Aelst et al. (2011); Van Aelst et al. (2012) and Van Aelst (2016) present adaptations of the Stahel-Donoho estimator to better measure outlyingness of observations for high-dimensional settings with cellwise outliers. Agostinelli et al. (2015) presented a complex procedure to deal with cellwise and casewise outliers in the multivariate location and scatter model. In the regression context, Oellerer et al. (2013) proposed the shooting S-estimator while Leung et al. (2016) proposed a three-step regression procedure to handle cellwise and casewise outliers.

Boente and Salibian-Barrera (2015) introduced an S-estimator for functional principal component analysis. They also define the estimator for multivariate data and discuss some theoretical properties in this setting. While the estimators presented in Chapter 1 look at entire observations by minimizing a robust scale of the residual norms, the estimator proposed by Boente and Salibian-Barrera (2015) minimizes the sum of the M-scales of the coordinates of the residuals. Therefore, the estimator is also suitable to handle problems with cellwise outliers although this was not the focus in Boente and Salibian-Barrera (2015). We refer to this estimator as the Coordinatewise S-estimator.

In this chapter we introduce the least trimmed squares equivalent of the Coordinatewise S-estimator, which we call the Coordinatewise least trimmed squares estimator for PCA (CooLTS-PCA). It adapts the Coordinatewise S-estimator by replacing the minimization of the sum of M-scales by the minimization of the sum of least trimmed squares scales to estimate the best $q-$dimensional linear space. It is expected that both procedures show good results in high-dimensional datasets with cellwise outliers while the multivariate S and LTS for PCA quickly loose their robustness in this setting. In section 2.2 we recall the definition of the Coordinatewise S-estimator and introduce the functional corresponding to the estimator. Next, in section 2.3 we define the Coordinatewise least trimmed squares estimator and derive the corresponding estimating equations. We also define the functional corresponding to the estimator. Following arguments in Boente and Salibian-Barrera (2015) it can also be shown that the CooLTS-PCA functional is Fisher-consistent and the estimator is consistent at elliptical distributions. In section 2.4 algorithms for both estimators are presented which are obtained by adapting the algorithms for the multivariate PCA methods in section 1.4. In section 2.5 we describe the fast strategy of Boente and Salibian-Barrera (2015) to choose the dimension of the subspace based on the proportion of unexplained variability and adapt it to our CooLTS estimator. In Section 2.6 the coordinatewise PCA procedures are assessed against the multivariate PCA procedures of Chapter 1 in a simulation study. Finally, in Section 2.7 we adapt the approach of Rousseeuw and Van den Bossche (2016) to flag cellwise outliers for the coordinatewise PCA methods and compare their results against a purely outlying detection method on a real data example. We also compare the outlying detection of coordinatewise PCA methods against that of multivariate-PCA methods using the Octane dataset of Section 1.9.

2.2 The coordinatewise S-estimator in $\mathbb{R}^p$

2.2.1 The estimator

As before, consider a sample $Z_n = \{\mathbf{x}_i,\ i = 1, \dots, n\} \subset \mathbb{R}^p$ and let $\mathbf{B}_q \in \mathbb{R}^{p\times q}$ be an orthogonal matrix with columns $\mathbf{B}_q = (\mathbf{b}^{(1)}, \dots, \mathbf{b}^{(q)})$, i.e. $\mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q$, and rows $\mathbf{b}_j^{\mathrm{T}}$, $j = 1, \dots, p$. Let $\mathbf{A}_q \in \mathbb{R}^{n\times q}$ be the matrix with rows $\mathbf{a}_i^{\mathrm{T}}$, $i = 1, \dots, n$, and $\mathbf{m} \in \mathbb{R}^p$. The corresponding approximations of the observations are given by $\widehat{\mathbf{x}}_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) \equiv \widehat{\mathbf{x}}_i = \mathbf{m} + \mathbf{B}_q\mathbf{a}_i$, or elementwise $\hat{x}_{ij} = m_j + \mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j$. The associated cellwise residuals are given by $r_{ij} = x_{ij} - \widehat{x}_{ij}$. Consider the vector $\mathbf{r}_j = (r_{1j}, r_{2j}, \dots, r_{nj})$ of the residuals corresponding to the jth variable.

Boente and Salibian-Barrera (2015) noted that the classical PCA problem in (1.1) can be rewritten as

$$\min_{\mathbf{B}_q,\mathbf{A}_q,\mathbf{m}} \sum_{i=1}^{n} \|\mathbf{r}_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\|^2 = \min_{\mathbf{B}_q,\mathbf{A}_q,\mathbf{m}} \sum_{i=1}^{n}\sum_{j=1}^{p} r_{ij}^2(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})$$

Since $\sum_{i=1}^{n} r_{ij}^2$ is proportional to s_j^2, the standard estimator of the variance of the residual vector $\mathbf{r}_j$, the classical principal components problem can also be formulated as

$$\min_{\mathbf{B}_q,\mathbf{A}_q,\mathbf{m}} \sum_{j=1}^{p} s_j^2(\mathbf{B}_q,\mathbf{A}_q,\mathbf{m}) \tag{2.1}$$

A robust alternative for classical PCA can thus be obtained by replacing the standard nonrobust variance of the residual vectors by a robust estimator of scale. The coordinatewise S-estimator of Boente and Salibian-Barrera (2015) uses an M-estimator of scale. Similarly as in (1.3), the M-scale estimate $\widehat{\sigma}_{\mathrm{M}}(\mathbf{r}_j)$ of the residual vector $\mathbf{r}_j$ is defined as the solution in s of the equation

$$\frac{1}{n}\sum_{i=1}^{n} \rho_c\left(\frac{r_{ij}}{s}\right) = b \tag{2.2}$$

The coordinatewise S-estimator for PCA (CooS-PCA) can now be defined as the solution $(\widehat{\mathbf{B}}_{\mathrm{CooS}}, \widehat{\mathbf{A}}_{\mathrm{CooS}}, \widehat{\mathbf{m}}_{\mathrm{CooS}})$ of the minimization problem

$$\min_{\mathbf{B}_q,\mathbf{A}_q,\mathbf{m}} \sum_{j=1}^{p} \widehat{\sigma}_{\mathrm{M}}^2\left(\mathbf{r}_j(\mathbf{B}_q,\mathbf{A}_q,\mathbf{m})\right) \tag{2.3}$$

Boente and Salibian-Barrera (2015) obtained explicit first-order conditions for the CooS-PCA estimator by differentiating (2.3) with respect to $\mathbf{a}_i$, $\mathbf{b}_j$ and μ_j. This yields:

$$\frac{\partial}{\partial \mathbf{a}_i}\left(\sum_{j=1}^{p} \hat{\sigma}_{\mathrm{M},j}^2\right) = -2\sum_{j=1}^{p} \hat{\sigma}_{\mathrm{M},j} h_j^{-1} \rho'\left(\frac{r_{ij}}{\hat{\sigma}_{\mathrm{M},j}}\right)\mathbf{b}_j\,, \qquad i = 1,\ldots,n\,,$$

$$\frac{\partial}{\partial \mathbf{b}_j}\left(\sum_{j=1}^{p} \hat{\sigma}_{\mathrm{M},j}^2\right) = -2\hat{\sigma}_{\mathrm{M},j} h_j^{-1} \sum_{i=1}^{n} \rho'\left(\frac{r_{ij}}{\hat{\sigma}_{\mathrm{M},j}}\right)\mathbf{a}_j\,, \qquad j = 1,\ldots,p\,,$$

$$\frac{\partial}{\partial \mu_j}\left(\sum_{j=1}^{p} \hat{\sigma}_{\mathrm{M},j}^2\right) = -2\hat{\sigma}_{\mathrm{M},j} h_j^{-1} \sum_{i=1}^{n} \rho'\left(\frac{r_{ij}}{\hat{\sigma}_{\mathrm{M},j}}\right)\,, \qquad j = 1,\ldots,p\,.$$

where $h_j = \sum_{i=1}^{n} \rho'\left(\frac{r_{ij}}{\hat{\sigma}_{\mathrm{M},j}}\right)\frac{r_{ij}}{\hat{\sigma}_{\mathrm{M},j}}$. Setting these to zero they obtained a system of equations which they rewrote as re-weighted least-squares problems. Setting the weights w_{ij} as

$$w_{ij} = \hat{\sigma}_{\mathrm{M},j} h_j^{-1} r_{ij}^{-1} \rho'\left(\frac{r_{ij}}{\hat{\sigma}_{\mathrm{M},j}}\right) \tag{2.4}$$

they wrote

$$\sum_{j=1}^{p} w_{ij}\,(x_{ij}-\mu_j)\,\mathbf{b}_j = \left(\sum_{j=1}^{p} w_{ij}\,\mathbf{b}_j\,\mathbf{b}_j^{\mathrm{T}}\right)\mathbf{a}_i\,, \qquad 1\le i\le n\,, \tag{2.5}$$

$$\sum_{i=1}^{n} w_{ij}\,(x_{ij}-\mu_j)\,\mathbf{a}_i = \left(\sum_{i=1}^{n} w_{ij}\,\mathbf{a}_i\,\mathbf{a}_i^{\mathrm{T}}\right)\mathbf{b}_j\,, \qquad 1\le j\le p\,, \tag{2.6}$$

$$\sum_{i=1}^{n} w_{ij}\,(x_{ij}-\mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j) = \sum_{i=1}^{n} w_{ij}\,\mu_j\,, \qquad 1\le j\le p\,. \tag{2.7}$$

This formulation naturally suggests an iterative re-weighted least square procedure to converge to local minima of the objective function which will be used in the algorithm of the estimator in section 2.4.

2.3 The coordinatewise LTS estimator in $\mathbb{R}^p$

2.3.1 The estimator

The coordinatewise least trimmed squares estimator uses univariate LTS scale estimators instead of sample variances in (2.1) to prevent the influence of outliers on the estimation of the PCA subspace. Similar as in 1.27, the LTS scale estimate $\hat{\sigma}^2_{\mathrm{LTS}}(\mathbf{r}_j)$ of the residual vector $\mathbf{r}_j$ is defined as

$$\hat{\sigma}^2_{\mathrm{LTS}}(\mathbf{r}_j) = \frac{1}{h}\sum_{i=1}^{h}(r^2_{ij})_{i:n} = \frac{1}{h}\sum_{i=1}^{n} w_{ij}(x_{ij}-m_j-\mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j)^2 \tag{2.8}$$

where the weights w_{ij} are:

$$w_{ij} = \begin{cases} 1 & \text{if } r^2_{ij} \le (r^2_{ij})_{h:n} \\ \\ 0 & \text{if } r^2_{ij} > (r^2_{ij})_{h:n} \end{cases} \tag{2.9}$$

The corresponding coordinatewise LTS-estimator for PCA (CooLTS-PCA) can now be defined as the solution $(\widehat{\mathbf{B}}_{\mathrm{CooLTS}}, \widehat{\mathbf{A}}_{\mathrm{CooLTS}}, \widehat{\mathbf{m}}_{\mathrm{CooLTS}})$ of the minimization problem

$$= \min_{\mathbf{B}_q,\mathbf{A}_q,\mathbf{m}} \sum_{j=1}^{p} \hat{\sigma}^2_{\mathrm{LTS}}\left(\mathbf{r}_j(\mathbf{B}_q,\mathbf{A}_q,\mathbf{m})\right), \tag{2.10}$$

over all orthogonal matrices $\mathbf{B}_q$, $\mathbf{A}_q$, and $\mathbf{m}$.

Explicit first-order conditions for the CooLTS-PCA estimator can be obtained by differentiating (2.10) with respect to $\mathbf{a}_i$, $\mathbf{b}_j$ and μ_j. This yields

$$\frac{\partial}{\partial \mathbf{a}_i}\left(\sum_{j=1}^{p} \hat{\sigma}^2_{\mathrm{LTS},j}\right) = -\frac{2}{h}\sum_{j=1}^{p} w_{ij}\, r_{ij}\, \mathbf{b}_j\,, \qquad i = 1,\dots,n\,,$$

$$\frac{\partial}{\partial \mathbf{b}_j}\left(\sum_{j=1}^{p} \hat{\sigma}^2_{\mathrm{LTS},j}\right) = -\frac{2}{h}\sum_{i=1}^{n} w_{ij}\, r_{ij}\, \mathbf{a}_i\,, \qquad j = 1,\dots,p\,,$$

$$\frac{\partial}{\partial \mu_j}\left(\sum_{j=1}^{p} \hat{\sigma}^2_{\mathrm{LTS},j}\right) = -\frac{2}{h}\sum_{i=1}^{n} w_{ij}\, r_{ij}\,, \qquad j = 1,\dots,p\,.$$

Setting these to zero we obtain the following system of equations:

$$\begin{aligned}\sum_{j=1}^{p} w_{ij}\,(x_{ij} - \mu_j - \mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j)\,\mathbf{b}_j &= \mathbf{0}\,, && 1 \le i \le n\,,\\ \sum_{i=1}^{n} w_{ij}\,(x_{ij} - \mu_j - \mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j)\,\mathbf{a}_i &= \mathbf{0}\,, && 1 \le j \le p\,,\\ \sum_{i=1}^{n} w_{ij}\,(x_{ij} - \mu_j - \mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j) &= 0\,, && 1 \le j \le p\,.\end{aligned}$$

Similarly as in Boente and Salibian-Barrera (2015), these estimating equations can be re-expressed as re-weighted least squares problems. More specifically, we obtain the equations:

$$\sum_{j=1}^{p} w_{ij}\,(x_{ij} - \mu_j)\,\mathbf{b}_j = \left(\sum_{j=1}^{p} w_{ij}\,\mathbf{b}_j\,\mathbf{b}_j^{\mathrm{T}}\right)\mathbf{a}_i\,, \qquad 1 \le i \le n\,, \tag{2.11}$$

$$\sum_{i=1}^{n} w_{ij}\,(x_{ij} - \mu_j)\,\mathbf{a}_i = \left(\sum_{i=1}^{n} w_{ij}\,\mathbf{a}_i\,\mathbf{a}_i^{\mathrm{T}}\right)\mathbf{b}_j\,, \qquad 1 \le j \le p\,, \tag{2.12}$$

$$\sum_{i=1}^{n} w_{ij}\,(x_{ij} - \mathbf{a}_i^{\mathrm{T}}\mathbf{b}_j) = \sum_{i=1}^{n} w_{ij}\,\mu_j\,, \qquad 1 \le j \le p\,. \tag{2.13}$$

Note that the weights w_{ij} are the (0-1) weights of 2.9. Equations above therefore suggests an iterative re-weighted least squares procedure which is used in the algorithm of the CooLTS estimator detailed in section 2.4. Note that the re-weighted least squares problems of the CooLTS-PCA are analogous to those of the CooS-PCA estimator in equations (2.5), (2.6) and (2.7). They only differ in the weights w_{ij}.

2.3.2 The functional

As before, consider a p-dimensional random variable $\mathbf{x}$ with a continuous distribution G with location $\boldsymbol{\mu}$ and scatter $\boldsymbol{\Sigma} \in$ SPSD. The scatter matrix $\boldsymbol{\Sigma}$ can be decomposed as $\boldsymbol{\Sigma} = \boldsymbol{\beta}\boldsymbol{\Lambda}\boldsymbol{\beta}^{\mathrm{T}}$ where $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$, and $\boldsymbol{\beta}$ is an orthogonal matrix with columns $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(p)}$. Without loss of generality we may again assume that $\boldsymbol{\mu} = \mathbf{0}$.

We can now define the functionals corresponding to the CooLTS-PCA estimator. Note that we still have that $\mathbf{a}_{\text{CooLTS}}(G) = \mathbf{B}_{\text{CooLTS}}(G)^T(\mathbf{x} - \mathbf{m}_{\text{CooLTS}}(G))$.Therefore, we focus on the functionals $\mathbf{B}_{\text{CooLTS}}(G)$ and $\mathbf{m}_{\text{CooLTS}}(G)$ Given a vector $\mathbf{m} \in \mathbb{R}^p$ and a matrix $\mathbf{B}_q \in \mathbb{R}^{p\times q}$ with $\mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q$, let $K_j(\mathbf{m}, \mathbf{B}_q)$ denote the distribution of $r_j(\mathbf{x}, \mathbf{m}, \mathbf{B}_q)$ where $\mathbf{r}(\mathbf{x}, \mathbf{m}, \mathbf{B}_q) = \mathbf{x} - \mathbf{m} - \mathbf{B}_q\mathbf{B}_q^{\mathrm{T}}\mathbf{x}$. Then, the functionals $(\mathbf{m}_{\text{CooLTS}}(G), \mathbf{B}_{\text{CooLTS}}(G))$ are the solution of the minimization problem

$$\min_{\mathbf{m}, \mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q} \quad \Psi(\mathbf{m}, \mathbf{B}_q), \tag{2.14}$$

where $\Psi(\mathbf{m}, \mathbf{B}_q) = \sum_{j=1}^{p} \sigma^2_{\text{LTS}}(K_j(\mathbf{m}, \mathbf{B}_q))$ with σ^2_{LTS} the LTS scale functional which is defined as follows. Consider a univariate continuous distribution K and $0 < \alpha < 1$ the probability mass of K not determining the LTS scale solution and define

$$\mathcal{J}_K(\alpha) = \{S \mid S \subset \mathbb{R},\ \text{measurable and bounded with } P_K(S) = 1 - \alpha\}\,.$$

Then, the LTS scale functional at distribution K is defined as

$$\sigma^2_{\text{LTS}}(K) = \min_{S \in \mathcal{J}_K(\alpha)} \sigma^2(K_S), \tag{2.15}$$

where $\sigma^2(K_S) = \frac{1}{1-\alpha}\int_S u^2\, dK(u)$ for any subset $S \in \mathcal{J}_K(\alpha)$. Hence, $\sigma^2(K_S)$ is the functional corresponding to the classical residual variance estimator for the subset S.

The CooLTS functional can also be written in terms of linear subspaces. To simplify the presentation, assume that the functional $\mathbf{m}_{\text{CooLTS}}(G)$ is known. Let $\pi(\mathbf{x} - \mathbf{m}, \mathcal{L}_{\mathbf{B}_q})$ be the orthogonal projection of $(\mathbf{x} - \mathbf{m})$ onto the subspace $\mathcal{L}_{\mathbf{B}_q}$. In addition, let $K_j(\mathcal{L}_{\mathbf{B}_q})$ denote the distribution of $r_j(\mathbf{x}, \mathcal{L}_{\mathbf{B}_q})$ where $\mathbf{r}(\mathbf{x}, \mathcal{L}_{\mathbf{B}_q}) = \mathbf{x} - \mathbf{m} - \pi(\mathbf{x} - \mathbf{m}, \mathcal{L}_{\mathbf{B}_q})$. Then, the CooLTS functional $\mathcal{L}_{\mathbf{B}_{\text{CooLTS}}}(G)$ can be defined as the minimizer of:

$$\min_{\dim(\mathcal{L}_{\mathbf{B}_q}) = q} \quad \Psi(\mathcal{L}_{\mathbf{B}_q}), \tag{2.16}$$

where $\Psi(\mathcal{L}_{\mathbf{B}_q}) = \sum_{j=1}^{p} \sigma^2_{\text{LTS}}(K_j(\mathcal{L}_{\mathbf{B}_q}))$ and σ^2_{LTS} is the LTS scale functional defined in (2.15).

Since the factor $c_\alpha = \left(\frac{1}{1-\alpha}\int_{-q}^{q} u^2 dF\right)^{-1}$ with $q_\alpha = F^{-1}(1-\alpha)$ makes the LTS scale estimator in (2.15) Fisher-consistent at elliptical distributions F, trivial adjustements to Proposition 2.1 in Boente and Salibian-Barrera (2015) proofs Fisher consistency of the functional $\mathcal{L}_{\mathbf{B}_{\text{CooLTS}}}(G)$ for elliptically distributed random vectors. Therefore, $\mathcal{L}_{\mathbf{B}_{\text{CooLTS}}}(G)$ is a Fisher-consistent functional for the parameter $\mathcal{L}_q$ when G is assumed to be an elliptical distribution as (1.18), i.e. $\mathcal{L}_{\mathbf{B}_{\text{CooLTS}}}(G) = \mathcal{L}_q$.

Let $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\text{CooLTS}}}$ be the CooLTS subspace estimator. That is, $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\text{CooLTS}}}$ is the minimizer of $\sum_{j=1}^{p} \hat{\sigma}^2_{\text{LTS}}\left(\mathbf{r}_j(\mathcal{L}_{\mathbf{B}_\text{q}})\right)$ over all linear subspaces $\mathcal{L}_{\mathbf{B}_\text{q}}$. Proposition 2.2 in Boente and Salibian-Barrera (2015) can be used to show consistency for $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\text{CooLTS}}}$ for elliptical random vectors because the M-scales in this proposition can directly be replaced by LTS scales. Hence, the CooLTS estimator $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\text{CooLTS}}}(Z_n)$ converges to $\mathcal{L}_q$ as the size of Z_n goes to infinity, i.e. as $n \to \infty$.

2.4 The algorithm

In order to solve the minimization problems in (2.3) and in (2.10) we adapt the algorithm in section 1.4 with the corresponding scales and the estimating equations (2.11), (2.12), (2.13). With initial choices for $\mathbf{B}_q$ and $\mathbf{m}$ as well as with choices for the tuning parameters N_1, N_2, N_{pc} and *tol*, the algorithm for the coordinatewise methods can by summarized as follows:

1. Set $it \leftarrow 0$.

   a. Compute $\mathbf{a}_i^{\text{T}} = (\mathbf{x}_i - \mathbf{m})^{\text{T}}\mathbf{B}_q$, $i = 1, \ldots, n$, and append these vectors to the rows of $\mathbf{A}_q$.

   b. Compute residuals $r_{ij} = x_{ij} - m_j - \mathbf{a}_i^{\text{T}}\mathbf{b}_j$, $i = 1, \ldots, n$, $j = 1, \ldots, p$.

   c. Take the vector $\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = (r_{1j}, r_{2j}, \ldots, r_{nj})$ and compute $\Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})$:

   - For the CooS estimator: $\Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = \sum_{j=1}^{p} \hat{\sigma}^2_{\text{M}}\left(\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\right)$ where $\hat{\sigma}^2_{\text{M}}\left(\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\right)$ is computed from (2.2).
   - For the CooLTS estimator: $\Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = \sum_{j=1}^{p} \hat{\sigma}^2_{\text{LTS}}\left(\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\right)$ where $\hat{\sigma}^2_{\text{LTS}}\left(\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\right)$ is computed from (2.8).

   d. Set $\hat{\sigma}_0^2 = \Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})$.

   e. Set $it = 1$.

2. Do until $it = N_1 + N_2$ or $\Delta \leq tol$.

```
    a. Compute $w_i$ and update the location $\mathbf{m} = \frac{\sum_{i=1}^n w_i \mathbf{x}_i}{\sum_{i=1}^n w_i}$.
        • For the CooS estimator: compute $w_i$ from (2.4).
        • For the CooLTS estimator: compute $w_i$ from (2.9).
    b. If $it > N_1$:
        (1) Set $iter \leftarrow 1$ and $\widehat{s}_0^2 = \Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})$ (current objective function value).
        (2) Do until $iter = N_{pc}$ or $\tilde{\Delta} \leq tol$
            i. Compute $\mathbf{a}_i$, $i = 1, \ldots, n$, $\mathbf{b}_j$ and $m_j$, $j = 1, \ldots, p$, using the estimating equations in (2.11)-(2.13) or (2.5)-(2.7).
            ii. Append the vectors $\mathbf{b}_j^{\mathrm{T}}$, $j = 1, \ldots, p$, to the rows of $\mathbf{B}_q$ and the vectors $\mathbf{a}_i^{\mathrm{T}}$, $i = 1, \ldots, n$ to the rows of $\mathbf{A}_q$.
            iii. Compute residual distances $d_i = \|\mathbf{r}_i(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})\| = \|\mathbf{x} - \mathbf{m} - \mathbf{B}_q \mathbf{a}_i\|$, $i = 1, \ldots, n$.
            iv. Set $\widehat{s}^2 = \frac{1}{n} \sum_{i=1}^n d_i^2$.
            v. Set $iter = iter + 1$, $\tilde{\Delta} \leftarrow 1 - \widehat{s}^2 / \widehat{s}_0^2$ and $\widehat{s}_0^2 \leftarrow \widehat{s}^2$.
        (3) End do.
    c. Compute $\mathbf{a}_i^{\mathrm{T}}$, $i = 1, \ldots, n$, using equation (2.5) or (2.11) and append these vectors to the rows of $\mathbf{A}_q$.
    d. Compute new residuals $r_{ij} = x_{ij} - m_j - \mathbf{a}_i^{\mathrm{T}} \mathbf{b}_j$, $i = 1, \ldots, n$, $j = 1, \ldots, p$.
    e. Take the new vector $\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = (r_{1j}, r_{2j}, \ldots, r_{nj})$ and compute the new $\Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})$:
        • For the CooS estimator: $\Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = \sum_{j=1}^p \hat{\sigma}_{\mathrm{M}}^2 (\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ where $\hat{\sigma}_{\mathrm{M}}^2 (\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ is computed from (2.2).
        • For the CooLTS estimator: $\Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}) = \sum_{j=1}^p \hat{\sigma}_{\mathrm{LTS}}^2 (\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ where $\hat{\sigma}_{\mathrm{LTS}}^2 (\mathbf{r}_j(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m}))$ is computed from (2.8).
    f. Set $\widehat{\sigma}^2 = \Psi(\mathbf{B}_q, \mathbf{A}_q, \mathbf{m})$.
    g. Set $\Delta \leftarrow 1 - \widehat{\sigma}^2 / \widehat{\sigma}_0^2$ and $\widehat{\sigma}_0^2 \leftarrow \widehat{\sigma}^2$.
    h. Set $it = it + 1$.
3. End do.
```

Note that this algorithm follows the recommendation of Maronna (2005) of fixing $\mathbf{B}_q$ for N_1 iterations to improve the location estimate. This algorithm is also suitable for high-dimensional settings since we compute eigenvectors from the estimating equations in (2.11)-(2.13) which only involve vector operations.

Similar to the MVS and MVLTS estimators, to search the global minimum in (2.3) or in (2.10) we generate N_{cand} random orthogonal matrices yielding N_{cand} initial $\mathbf{B}_q$'s. The initial location estimate $\mathbf{m}$ is the spatial median of the data matrix $\mathbf{X}$. Next, we run the above updating algorithm for each initial candidate with parameters N_1, N_2, N_{pc} and *tol*, and keep N_{keep} of the resulting estimates with lowest robust scale $\widehat{\sigma}$. For each of these N_{keep} cases the algorithm continues running with parameters N_1', N_2', N_{pc}' and tol'. Among the final candidates we then select the one with smallest robust scale $\widehat{\sigma}$ which is the approximation for the global minimum and the resulting estimates are the solution to (2.3) or to (2.10) of this strategy. Here we also use the method of Stewart (1980) to generate random orthogonal matrices. For the experiments of section 2.6 we kept the same parameter values as Maronna (2005) which gave good results. These values are detailed in section 1.4.1.

We are also interested in investigating the strategy with deterministic starting values for the coordinatewise methods. Recall that this strategy already showed excellent results and one of the lowest computational times in the experiments and in the application of Chapter 1 for the MVS and for the MVLTS estimators. As described in section 1.4.2, this strategy aims to find a robust local minimum by starting from five well-chosen starting values for $\mathbf{B}_q$ and $\mathbf{m}$ in the algorithm above. The steps to generate these starting values are also described in section 1.4.2. We used this strategy for the coordinatewise PCA methods in chapter 3 where we extend the estimators to accomodate functional data. In the experiments and applications of chapter 3 we used the same tuning parameter values as in the experiments of Chapter 1. The whole procedure will certainly be faster with deterministic starting values than with random orthogonal starts and therefore the former strategy is more suitable to handle larger problems like in chapter 3.

2.5 Number of components

To choose the number of components for the Coordinatewise LTS procedure we can use the strategy formulated by Boente and Salibian-Barrera (2015) which we now describe. Let us consider a fixed dimension q for the subspace. Then, at the true distribution G of $\mathbf{x} \in \mathbb{R}^p$ the smallest possible unexplained variance u_q is attained by the eigenvectors $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(q)}$ of the underlying scatter matrix Σ which yields

$$u_q = \frac{\sum_{j=q+1}^{p} \lambda_j}{\sum_{j=1}^{p} \lambda_j}, \tag{2.17}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$ are the corresponding eigenvalues.
Let $\Psi(\widetilde{\mathbf{B}}_{\text{CooLTS}}, \widetilde{\mathbf{A}}_{\text{CooLTS}}, \widetilde{\mathbf{m}}_{\text{CooLTS}}) = \sum_{j=1}^{p} \hat{\sigma}^2_{\text{LTS}} \left(\mathbf{r}_j(\widetilde{\mathbf{B}}_{\text{CooLTS}}, \widetilde{\mathbf{A}}_{\text{CooLTS}}, \widetilde{\mathbf{m}}_{\text{CooLTS}}) \right)$ be the sum of the LTS scale estimates of the coordinates of the residuals corresponding to the CooLTS estimates $(\widetilde{\mathbf{B}}_{\text{CooLTS}}, \widetilde{\mathbf{A}}_{\text{CooLTS}}, \widetilde{\mathbf{m}}_{\text{CooLTS}})$ obtained in (2.10). Furthermore let

$\Psi_0(\widetilde{\mathbf{m}})$ be the minimum of $\sum_{j=1}^{p} \hat{\sigma}^2_{\text{LTS}}\left(\mathbf{r}_j(\mathbf{m})\right)$ over all $\mathbf{m} \in \mathbb{R}^p$ where the vector $\mathbf{r}_j(\mathbf{m}) = (r_{1j}(\mathbf{m}), r_{2j}(\mathbf{m}), \ldots, r_{nj}(\mathbf{m}))$ is the jth coordinate of the residuals $\mathbf{r}_i(\mathbf{m}) = \mathbf{x}_i - \mathbf{m} = (r_{i1}(\mathbf{m}), \ldots, r_{ip}(\mathbf{m}))$. Analogously to (1.46) we can estimate the proportion of unexplained variability for the CooLTS estimator by:

$$\widehat{u}_q = \frac{\Psi(\widetilde{\mathbf{B}}_{\text{CooLTS}}, \widetilde{\mathbf{A}}_{\text{CooLTS}}, \widetilde{\mathbf{m}}_{\text{CooLTS}})}{\Psi_0(\widetilde{\mathbf{m}})}. \tag{2.18}$$

Note that $\Psi_0(\widetilde{\mathbf{m}})$ is the sum of LTS scale estimates of the coordinates of the residuals for the case where no principal components are fitted and thus yields an estimate of the total variance in the data. Proposition 2.2 in Boente and Salibian-Barrera (2015) can be used to show that $\widehat{u}_q$ consistently estimates u_q.

As before, let $u_{\max}$ be the maximum proportion of unexplained variability that the problem allows and denote as $q_{\max}$ the maximum dimension of the subspace that we are willing to accept. We therefore look for the smallest q such that $q \leq q_{\max}$ and $\widehat{u}_q \leq u_{\max}$. The strategy that we now describe is faster than solving (2.10) for $q_{\max}$, $q_{\max} - 1$, and so forth.

The procedure starts with $q = 1$. If $\widehat{u}_{q=1} \leq u_{\max}$ then we are done. Otherwise, take the CooLTS solutions $\widehat{\mathbf{B}}_1 \in \mathbb{R}^{p \times 1}$, $\widehat{\mathbf{A}}_1 \in \mathbb{R}^{n \times 1}$ and $\widehat{m}_j^{(1)}$ obtained for $q = 1$ and proceed as follows. Let $r_{ij}^{(1)}$ be the corresponding elementwise residuals, $j = 1, \ldots, p$, $i = 1, \ldots, n$. Set $q = 2$ and define the matrices $\mathbf{B}_2 = (\widehat{\mathbf{B}_1}, \mathfrak{B}) \in \mathbb{R}^{p \times 2}$ with $\mathfrak{B} = (\mathfrak{b}_1, \ldots, \mathfrak{b}_p)^T$ and $\mathbf{A}_2 = (\widehat{\mathbf{A}}_1, \mathfrak{A}) \in \mathbb{R}^{n \times 2}$ with $\mathfrak{A} = (\mathfrak{a}_1, \ldots, \mathfrak{a}_n)^T$. The corresponding predictions are then obtained by $\hat{x}_{ij}^{(2)} = \hat{x}_{ij}^{(1)} + \mathfrak{b}_j \mathfrak{a}_i$ and note that the residuals satisfy $r_{ij}^{(2)} = r_{ij}^{(1)} - \mathfrak{b}_j \mathfrak{a}_i$. The marginal optimization problem now becomes

$$\min_{\widehat{\mathbf{B}}_1^{\mathrm{T}} \mathfrak{B} = \mathbf{0}, \mathfrak{A}} \sum_{j=1}^{p} \hat{\sigma}^2_{\text{LTS}} \left(\mathbf{r}_j^{(2)}(\mathfrak{B}, \mathfrak{A}) \right) \tag{2.19}$$

over $\mathfrak{B}, \mathfrak{A}$, with $\widehat{\mathbf{B}}_1^{\mathrm{T}} \mathfrak{B} = \mathbf{0}$ and $\mathbf{r}_j^{(2)}(\mathfrak{B}, \mathfrak{A}) = (r_{1j}^{(1)} - \mathfrak{b}_j \mathfrak{a}_1, \ldots, r_{nj}^{(1)} - \mathfrak{b}_j \mathfrak{a}_n)$.

A system of estimating equations analogous as in Section 2.3 can be derived and then used in the iterative algorithm of Section 2.4 to find a solution. Once the optimal solutions $\widehat{\mathfrak{B}}$ and $\widehat{\mathfrak{A}}$ according to (2.19) are found, we optimize the sum of LTS scales of the coordinates of the residuals w.r.t. $\mathbf{m}$ to obtain an updated estimate $\widehat{\mathbf{m}}_2$. With $\widehat{\mathbf{B}}_2 = (\widehat{\mathbf{B}}_1, \widehat{\mathfrak{B}})$, $\widehat{\mathbf{A}}_2 = (\widehat{\mathbf{A}}_1, \widehat{\mathfrak{A}})$ and $\widehat{\mathbf{m}}_2$ we then compute $\widetilde{u}_2 = \frac{\Psi(\widehat{\mathbf{B}}_2, \widehat{\mathbf{A}}_2, \widehat{\mathbf{m}}_2)}{\Psi_0(\widetilde{\mathbf{m}})}$. If $\widetilde{u}_2 \leq u_{\max}$ the procedure stops. Otherwise, we increase q by 1 and repeat the procedure until $\widetilde{u}_q \leq u_{\max}$ or until $q = q_{\max}$. Note that it should hold that $\widehat{u}_{q_{\max}} \leq u_{\max}$ otherwise the problem cannot be solved and we will have to modify our goals (increase $q_{\max}$ or $u_{\max}$). As we remarked before the quantity $\widetilde{u}_q$ is typically larger than $\widehat{u}_q$ so that we make a safe choice for q when $\widetilde{u}_q \leq u_{\max}$.

2.6 Simulation study

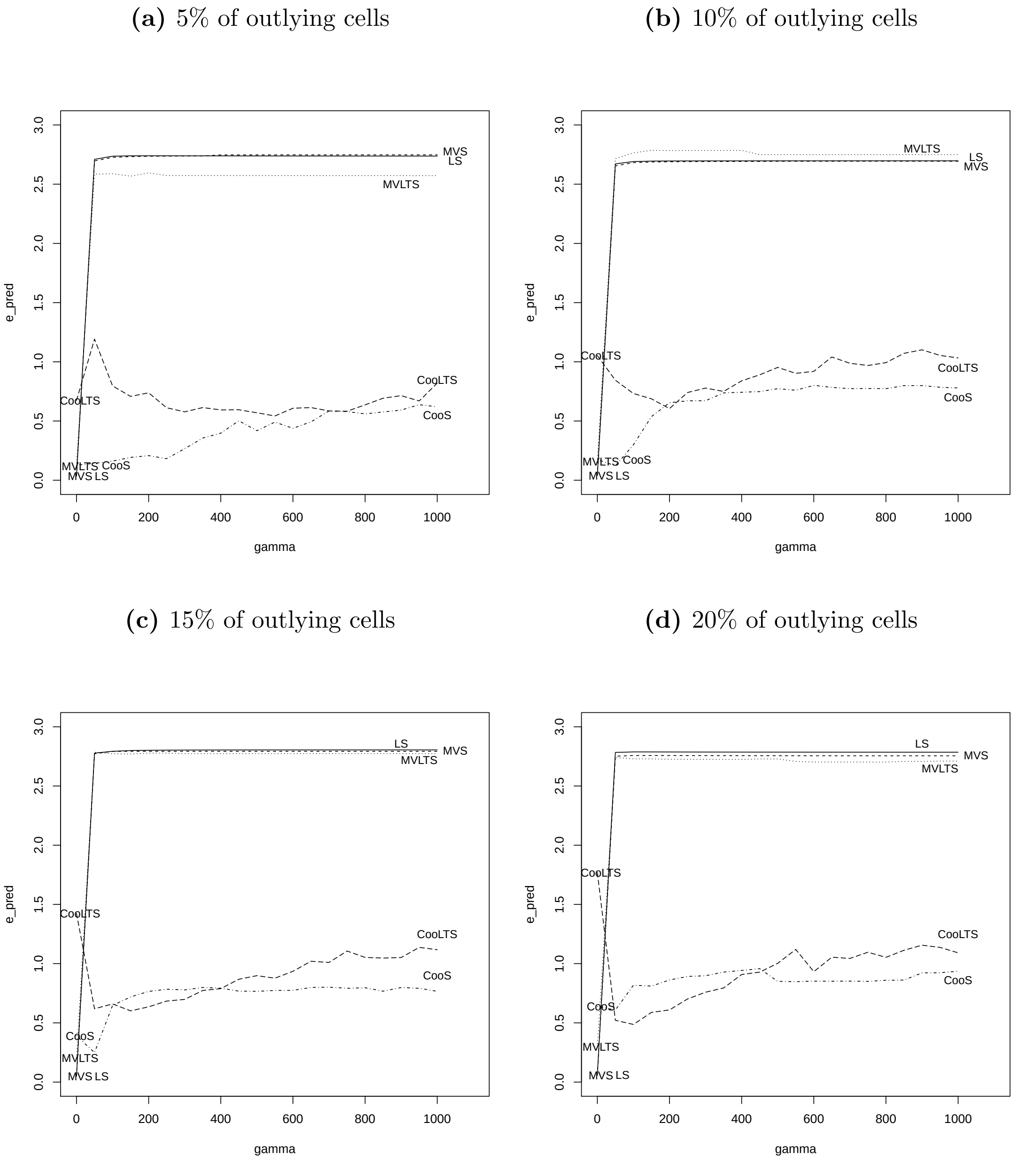
We want to assess the performance of the coordinatewise PCA procedures against the multivariate PCA procedures when data contain cellwise outliers. For the comparison we consider the MVS-PCA, the MVLTS-PCA, the CooS-PCA and our CooLTS-PCA estimator. To calculate these estimators we use the iterative algorithms of sections 1.4 and 2.4 and use the same parameter values as in the experiments in section 1.6. We only consider the strategy that generates random orthogonal matrices for $\mathbf{B}_q$ since it looks to approximate the global minimum. For the MVS-PCA and for the CooS-PCA estimator we used the Tukey biweight loss function with tuning parameters $c = 1.54764$, $b = 0.5$. For the MVLTS-PCA and for the CooLTS-PCA estimator we considered $\alpha = 0.5$.

To assess the effect of cellwise outliers on the estimators we replicate one of the simulations in Rousseeuw and Van den Bossche (2016) that generates contamination at random. First, we generate multivariate data of size $n = 100$ and dimension $p = 20$ from the multivariate gaussian distribution with mean zero and A09 correlation matrix which is given by $\rho_{jh} = (-0.9)^{|h-j|}$. The A09 correlation matrix yields low and high correlations. Next, these clean data are contaminated. Outlying cells are generated by replacing a random subset of the $n \times p$ cells by a value γ which was varied to see its effect. In our experiments we consider fractions of contamination of 5%, 10%, 15% and 20%. From 100 experiments these fractions of outlying cells produce corresponding fractions of contaminated observations of 64%, 88%, 96% and 99%, on average. This shows how harmful only a small percentage of outlying cells in the number of contaminated observations can be. The value of γ runs between 0 and 1000 with steps of 50. Therefore, we also consider the case of contamination with extreme values. In all experiments we tried to estimate the best linear subspace of dimension $q = 2$. To assess the robust performance of the methods we used the relative prediction error defined in (1.50). We replicate the experiments $M = 200$ times and report the mean relative prediction error $\bar{e}_{\text{pred}}$ over those replications.

Figure 2.1 shows the results of these experiments for the four cases of outlying cells considered.

2.6.1 Results

As shown in Table 2.1, for the case of only clean data classical PCA (LS) shows the smallest prediction error while the multivariate-PCA methods have clearly lower errors than their coordinatewise counterparts.


(a) 5% of outlying cells

(b) 10% of outlying cells

(c) 15% of outlying cells

(d) 20% of outlying cells

Figure 2.1: Mean relative prediction errors $\overline{e}_{\text{pred}}$ over $M = 200$ replications as a function of the contamination values γ. For $\gamma = 0$, the $\overline{e}_{\text{pred}}$ value is indicated with the name of the method. Panels (a) to (d) shows the results with 5%, 10%, 15% and 20% of outlying cells.

Table 2.1: Mean relative prediction errors $\overline{e}_{\text{pred}}$ for data without contamination

	$\epsilon = 0\%$
LS	0.02
MVS	0.03
MVLTS	0.08
CooS	0.09
CooLTS	0.42

A similar pattern is found when cells are randomly replaced by a value $\gamma = 0$. However, the performance of the coordinatewise-PCA methods get much worse as the fraction of outliers increase to 15% or 20%, especially the performance of CooLTS (see Panels 2.1c and 2.1d). Even though $\gamma = 0$ corresponds to contaminating with the mean value, the PCA solution by coordinatewise methods seem to get more biased than that of the multivariate methods when the fraction of contamination becomes larger. Coordinatewise-PCA methods may pick up many of these 0 valued cells directly to estimate its solution. The impact on multivariate-PCA methods may be more mild since they pick up entire observations by searching those with the smallest euclidean distances.

Figure 2.1 reveals however that multivariate-PCA methods break down with clear outliers ($\gamma = 50, 100, \ldots, 1000$) in any of the fractions considered. The coordinatewise methods show robust results in those cases. In fact, Multivariate-PCA methods perform as bad as the classical PCA. The poor performance of multivariate-PCA methods was expected since even a fraction of 5% of cellwise outliers leads to a percentage of contaminated observations that exceeds the critical value of 50%, on average.

We note that the performance of CooLTS is decreased with higher values of contamination for fractions of 10%-20% of outliers (see Panels 2.1b - 2.1d). CooS also gets its performance decreased with higher values of contamination for a fraction of 5% of outliers. For fractions of 10%-20% of outliers CooS shows a prediction error which increases at the beginning but then levels off at some point. However, the effect of extreme values of contamination is still small for both methods and they still look robust compared to the multivariate-PCA methods. We also note that the prediction errors of coordinatewise-PCA methods show a somewhat wiggly pattern. This may be due to the random contamination introduced which does not ensure that every column has the same fraction of outliers in every experiment. On the other hand, for clear outliers multivariate-PCA methods show a stable pattern for its large prediction error because in all these experiments we always introduce contamination in more than 50% of observations.

Overall, the Coordinatewise LTS and the Coordinatewise S estimators alternatively beat each other in the different scenarios analyzed. These experiments also show that in general coordinatewise-PCA methods yield more robust results than multivariate-PCA methods when the data contain large fractions of contaminated observations.

2.7 Real data examples

We now illustrate the coordinatewise-PCA methods on two real data examples. In the first example we compare cellwise outlier detection by coordinatewise-PCA methods to the results of *DetectDeviatingCells* (DDC), which is purely an outlier detection method. *DetectDeviatingCells* was introduced in Rousseeuw and Van den Bossche (2016) and detects cellwise outliers by taking correlations between the variables into account. For this purpose, we use the Top Gear data analyzed in Rousseeuw and Van den Bossche (2016). In the second example we compare outlier detection by coordinatewise-PCA methods to that by multivariate-PCA methods. For this purpose we use the Octane dataset which was already analyzed in Section 1.9.

In order to decide whether a cell is outlying or not according to our coordinatewise-PCA methods, we use a similar approach as in Rousseeuw and Van den Bossche (2016) based on quantiles of the χ^2 distribution. In particular, the following steps are taken:

1. Check that every variable in the dataset is approximately Gaussian. If there are very non-Gaussian variables they should be transformed to approximate Gaussianity, e.g. by taking a logarithmic transformation.

2. Fit the coordinatewise-PCA method and obtain estimates $(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}})$. The corresponding approximations are $\widehat{\mathbf{x}}_i = \widehat{\mathbf{m}} + \widehat{\mathbf{B}}_q \widehat{\mathbf{a}}_i$, $i = 1, \ldots, n$.

3. Compute cellwise residuals $r_{ij}(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}}) \equiv r_{ij} = x_{ij} - \widehat{x}_{ij}$. Note that $\mathbf{r}_j = (r_{1j}, r_{2j}, \ldots, r_{nj})$ is the vector of residuals for the jth variable.

4. Calculate standardized cell residuals:

$$z_{ij} = \frac{r_{ij}(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}})}{\widehat{\sigma}(\mathbf{r}_j(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}}))} \tag{2.20}$$

   where $\widehat{\sigma}(\mathbf{r}_j(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}}))$ is the coordinatewise robust scale estimate. That is, $\widehat{\sigma}(\mathbf{r}_j(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}})) = \widehat{\sigma}_{\mathrm{M}}(\mathbf{r}_j(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}}))$ for the CooS estimator and $\widehat{\sigma}(\mathbf{r}_j(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}})) = \widehat{\sigma}_{\mathrm{LTS}}(\mathbf{r}_j(\widehat{\mathbf{B}}_q, \widehat{\mathbf{A}}_q, \widehat{\mathbf{m}}))$ for the CooLTS estimator. These scale estimates are part of the output of the algorithm (see Section 2.4) so that (2.20) can be computed without additional computational effort.

5. Take as cutoff value $c = \sqrt{\chi^2_{1,p}}$, where $\chi^2_{1,p}$ is the pth quantile of the chi-squared distribution with one degree of freedom. As in Rousseeuw and Van den Bossche (2016) we take a probability tolerance $p = 99\%$ for the examples in this section.

6. Finally, in each column j, flag as an outlier all cells with $|r_{ij}| > c$.

Note that in step 2 PCA methods give lower dimensional approximations while *DetectDeviatingCells* gives estimated values for the cells. We adopt a similar strategy as in Rousseeuw and Van den Bossche (2016) for the coordinatewise-PCA methods to flag an entire observation if it contains too many cells with anomalous behaviour. With this approach, coordinatewise-PCA methods can also flag outlying cases similar to multivariate-PCA methods. The proposed approach in Rousseeuw and Van den Bossche (2016) is based on noting that under the null hypothesis of clean multivariate Gaussian data the distribution of the z_{ij} is close to standard Gaussian, so that the cdf of z_{ij}^2 is approximately the cdf F of χ^2_1. Rousseeuw and Van den Bossche (2016) first computes the criterion:

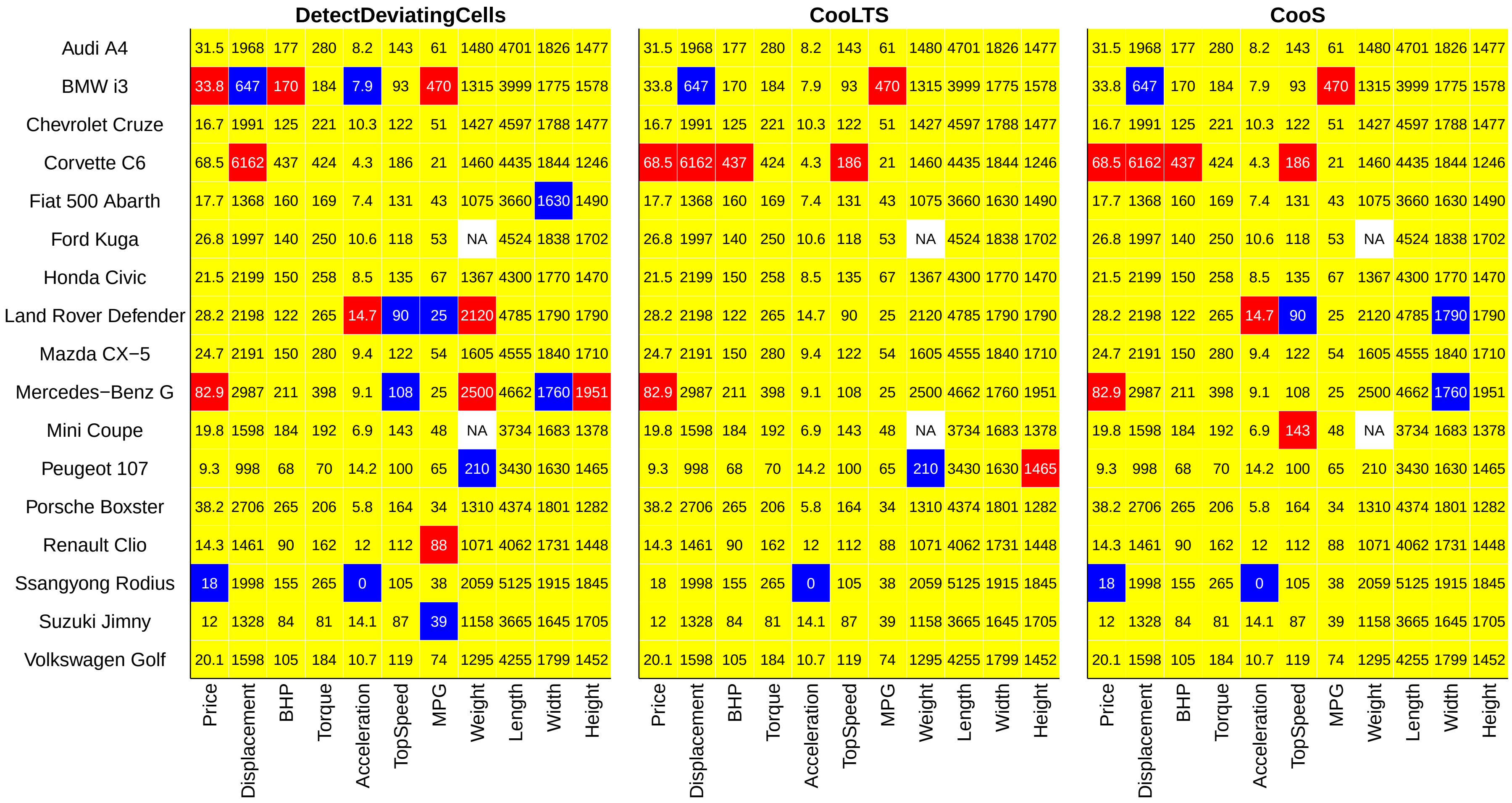
$$T_i = \frac{1}{p}\sum_{j=1}^{p} F(z_{ij}^2) \;-\; \frac{1}{2}.$$

It is easy to see that T_i lies between -0.5 and 0.5. Next, we robustly standardize the T_i's. We just use the median and the MAD for the standardization step instead of using the robust estimates proposed in Rousseeuw and Van den Bossche (2016). Finally, we flag the observations i for which the standardized T_i exceed the cutoff c defined before.

2.7.1 Top gear data

This dataset was included in Alfons (2016) and contains information on cars featured on the website of the popular British television show “Top Gear”. More specifically, there are 32 variables about 297 cars. The dataset also contains a few missing values. To make it possible the fitting of *DetectDeviatingCells*, CooS and CooLTS, we removed all non-numerical variables, which left us with only 11 variables. Furthermore, to make *DetectDeviatingCells* work well, we also set aside rows with more than 20% of missing values. To make the comparison fair, we carried out the same step before fitting CooS and CooLTS. This left us with 280 observations on 11 variables containing a few missing values. Five variables were rather skewed so we logarithmically transformed them. Namely: “price”, “displacement”, “BHP”, “torque” and “topspeed”. Since our PCA methods cannot handle missing values yet, we imputed the remaining missing values once with the MICE procedure (i.e. with the parameter $m = 1$, see Table 4.3) before running CooS and CooLTS. To fit *DetectDeviatingCells* we kept those missing values since DDC can handle missing data well. Except for the imputation of missing values, CooS and CooLTS were fitted on the same final dataset as *DetectDeviatingCells*. This allows a fair comparison of the methods.

The left panel of Figure 2.2 shows the cell map obtained by applying *DetectDeviatingCells*. Here we plot the same rows as in Rousseeuw and Van den Bossche (2016) since they represent interesting cases. Note that because we used the same tolerance value of 99%, the plot on the left panel is an exact reproduction of the one displayed in Rousseeuw and Van den Bossche (2016) for DDC. The panel in the middle shows the results of CooLTS while the right panel shows the results of CooS. Here we show the results of the algorithm starting with random orthogonal matrices that retain $q = 2$ components (with 2 components both CooLTS and CooS explain 99% of the total variability). However, similar results were found for a 2-dimensional approximation with the algorithm that starts with deterministic values (with explanation of 99% of the total variability by both methods). This result is included in Figure D.1 of Appendix D. Cells in yellow represent regular cells while those colored red or blue represent outlying cells. If the observed cell is much higher than the estimated/approximated cell value, then it is colored red. If it is much lower it is colored blue. Note that missing data are shown in white color and labeled NA (from ‘not available’). Moreover, if an entire observation is outlying we color it black. However, as can be seen from Figure 2.2, for the selected rows none of the methods detected an outlying observation.



DetectDeviatingCells

	Price	Displacement	BHP	Torque	Acceleration	TopSpeed	MPG	Weight	Length	Width	Height
Audi A4	31.5	1968	177	280	8.2	143	61	1480	4701	1826	1477
BMW i3	🟥 33.8	🟦 647	🟥 170	184	🟦 7.9	93	🟥 470	1315	3999	1775	1578
Chevrolet Cruze	16.7	1991	125	221	10.3	122	51	1427	4597	1788	1477
Corvette C6	68.5	🟥 6162	437	424	4.3	186	21	1460	4435	1844	1246
Fiat 500 Abarth	17.7	1368	160	169	7.4	131	43	1075	3660	🟦 1630	1490
Ford Kuga	26.8	1997	140	250	10.6	118	53	NA	4524	1838	1702
Honda Civic	21.5	2199	150	258	8.5	135	67	1367	4300	1770	1470
Land Rover Defender	28.2	2198	122	265	🟥 14.7	🟦 90	🟦 25	🟥 2120	4785	1790	1790
Mazda CX−5	24.7	2191	150	280	9.4	122	54	1605	4555	1840	1710
Mercedes−Benz G	🟥 82.9	2987	211	398	9.1	🟦 108	25	🟥 2500	4662	🟦 1760	🟥 1951
Mini Coupe	19.8	1598	184	192	6.9	143	48	NA	3734	1683	1378
Peugeot 107	9.3	998	68	70	14.2	100	65	🟦 210	3430	1630	1465
Porsche Boxster	38.2	2706	265	206	5.8	164	34	1310	4374	1801	1282
Renault Clio	14.3	1461	90	162	12	112	🟥 88	1071	4062	1731	1448
Ssangyong Rodius	🟦 18	1998	155	265	🟦 0	105	38	2059	5125	1915	1845
Suzuki Jimny	12	1328	84	81	14.1	87	🟦 39	1158	3665	1645	1705
Volkswagen Golf	20.1	1598	105	184	10.7	119	74	1295	4255	1799	1452

CooLTS

	Price	Displacement	BHP	Torque	Acceleration	TopSpeed	MPG	Weight	Length	Width	Height
Audi A4	31.5	1968	177	280	8.2	143	61	1480	4701	1826	1477
BMW i3	33.8	🟦 647	170	184	7.9	93	🟥 470	1315	3999	1775	1578
Chevrolet Cruze	16.7	1991	125	221	10.3	122	51	1427	4597	1788	1477
Corvette C6	🟥 68.5	🟥 6162	🟥 437	424	4.3	🟥 186	21	1460	4435	1844	1246
Fiat 500 Abarth	17.7	1368	160	169	7.4	131	43	1075	3660	1630	1490
Ford Kuga	26.8	1997	140	250	10.6	118	53	NA	4524	1838	1702
Honda Civic	21.5	2199	150	258	8.5	135	67	1367	4300	1770	1470
Land Rover Defender	28.2	2198	122	265	14.7	90	25	2120	4785	1790	1790
Mazda CX−5	24.7	2191	150	280	9.4	122	54	1605	4555	1840	1710
Mercedes−Benz G	🟥 82.9	2987	211	398	9.1	108	25	2500	4662	1760	1951
Mini Coupe	19.8	1598	184	192	6.9	143	48	NA	3734	1683	1378
Peugeot 107	9.3	998	68	70	14.2	100	65	🟦 210	3430	1630	🟥 1465
Porsche Boxster	38.2	2706	265	206	5.8	164	34	1310	4374	1801	1282
Renault Clio	14.3	1461	90	162	12	112	88	1071	4062	1731	1448
Ssangyong Rodius	18	1998	155	265	🟦 0	105	38	2059	5125	1915	1845
Suzuki Jimny	12	1328	84	81	14.1	87	39	1158	3665	1645	1705
Volkswagen Golf	20.1	1598	105	184	10.7	119	74	1295	4255	1799	1452

CooS

	Price	Displacement	BHP	Torque	Acceleration	TopSpeed	MPG	Weight	Length	Width	Height
Audi A4	31.5	1968	177	280	8.2	143	61	1480	4701	1826	1477
BMW i3	33.8	🟦 647	170	184	7.9	93	🟥 470	1315	3999	1775	1578
Chevrolet Cruze	16.7	1991	125	221	10.3	122	51	1427	4597	1788	1477
Corvette C6	🟥 68.5	🟥 6162	🟥 437	424	4.3	🟥 186	21	1460	4435	1844	1246
Fiat 500 Abarth	17.7	1368	160	169	7.4	131	43	1075	3660	1630	1490
Ford Kuga	26.8	1997	140	250	10.6	118	53	NA	4524	1838	1702
Honda Civic	21.5	2199	150	258	8.5	135	67	1367	4300	1770	1470
Land Rover Defender	28.2	2198	122	265	🟥 14.7	🟦 90	25	2120	4785	🟦 1790	1790
Mazda CX−5	24.7	2191	150	280	9.4	122	54	1605	4555	1840	1710
Mercedes−Benz G	🟥 82.9	2987	211	398	9.1	108	25	2500	4662	🟦 1760	1951
Mini Coupe	19.8	1598	184	192	6.9	🟥 143	48	NA	3734	1683	1378
Peugeot 107	9.3	998	68	70	14.2	100	65	210	3430	1630	1465
Porsche Boxster	38.2	2706	265	206	5.8	164	34	1310	4374	1801	1282
Renault Clio	14.3	1461	90	162	12	112	88	1071	4062	1731	1448
Ssangyong Rodius	🟦 18	1998	155	265	🟦 0	105	38	2059	5125	1915	1845
Suzuki Jimny	12	1328	84	81	14.1	87	39	1158	3665	1645	1705
Volkswagen Golf	20.1	1598	105	184	10.7	119	74	1295	4255	1799	1452

Figure 2.2: Cell maps for selected rows of the Top gear data when detecting cellwise outliers with *DetectDeviatingCells* (left-hand side), with CooLTS (center) and with CooS (right-hand side).

We see that CooLTS and CooS yield similar results to DDC. Both CooLTS and CooS are able to flag the unusual high gas mileage (MPG) of the BMW i3. This is in fact an electric vehicle with only a small additional gas engine. For both coordinatewise-PCA methods the Corvette's displacement is not the only feature that stands out, but rather several features are unusual. The Corvette's displacement is actually not unusual by itself, but it is high in relation to other features of the car. Our CooLTS is able to pick the abnormal weight of 210 kg for the Peugeot 107 while both coordinatewise-PCA methods flag the acceleration time of zero seconds for the Ssangyong Rodius vehicle as outlying. Actually, an acceleration time of zero seconds from 0 to 62 mph is physically impossible, so this is a clear outlying cell.

Coordinatewise-PCA methods are different in nature compared to *DetectDeviatingCells*. The primary goal of PCA methods is to estimate the best lower-dimensional subspace while DDC is a purely oulier detection method. However, with this example we show that robust PCA by coordinatewise methods mostly detects the same cellwise outliers detected by *DetectDeviatingCells*.

2.7.2 Octane data

We now revisit the Octane data to fit coordinatewise-PCA methods. Recall that this dataset consists of near-infrared (NIR) absorbance spectra of 39 gasoline samples with some octane numbers over 226 wavelenghts. In Section 1.9 we estimated the best 2-dimensional subspace by multivariate-PCA methods. For the sake of comparison we also consider a 2-dimensional approximation based on the coordinatewise-PCA methods. With 2 components CooLTS and CooS explain about 98% of the total variability. Previous results of MVS and MVLTS clearly flagged the six samples with added alcohol as outliers, namely observations 25, 26 and 36-39. Figure 2.3 shows these results for the multivariate-PCA methods (top panel) as well as the cellwise outliers detected by CooLTS and CooS with random orthogonal starts (middle and bottom panel respectively). Both CooLTS and CooS flag the observations with added alcohol as outliers (black color in the horizontal lines). To keep cellwise outliers visible, we have superimposed the red and blue colors on the plot. We now see which data cells are actually responsible for flagging observations 25, 26 and 36-39 as outliers. Moreover, one can see that a few other samples contain cellwise outliers. By looking at the wavelengths of these outlying cells we can actually know which chemical elements are responsible for the deviating measurements. CooLTS and CooS with deterministic starting values yield similar results but do not flag the alcohol samples as outlying observations. These results together with the result of *DetectDeviatingCells* are shown in Figure D.2 of Appendix D.

This example shows us that it can easily happen for an ‘outlying observation’ that many of its data cells are regular and only a few of its cells are actually outlying. Discarding an entire observation can therefore lead to a considerable loss of good information, especially in high-dimensional settings where we dispose of only a few observations. Coordinatewise-PCA methods are a good alternative to detect which are the cells responsible for the ‘outlying’ behavior of an observation.

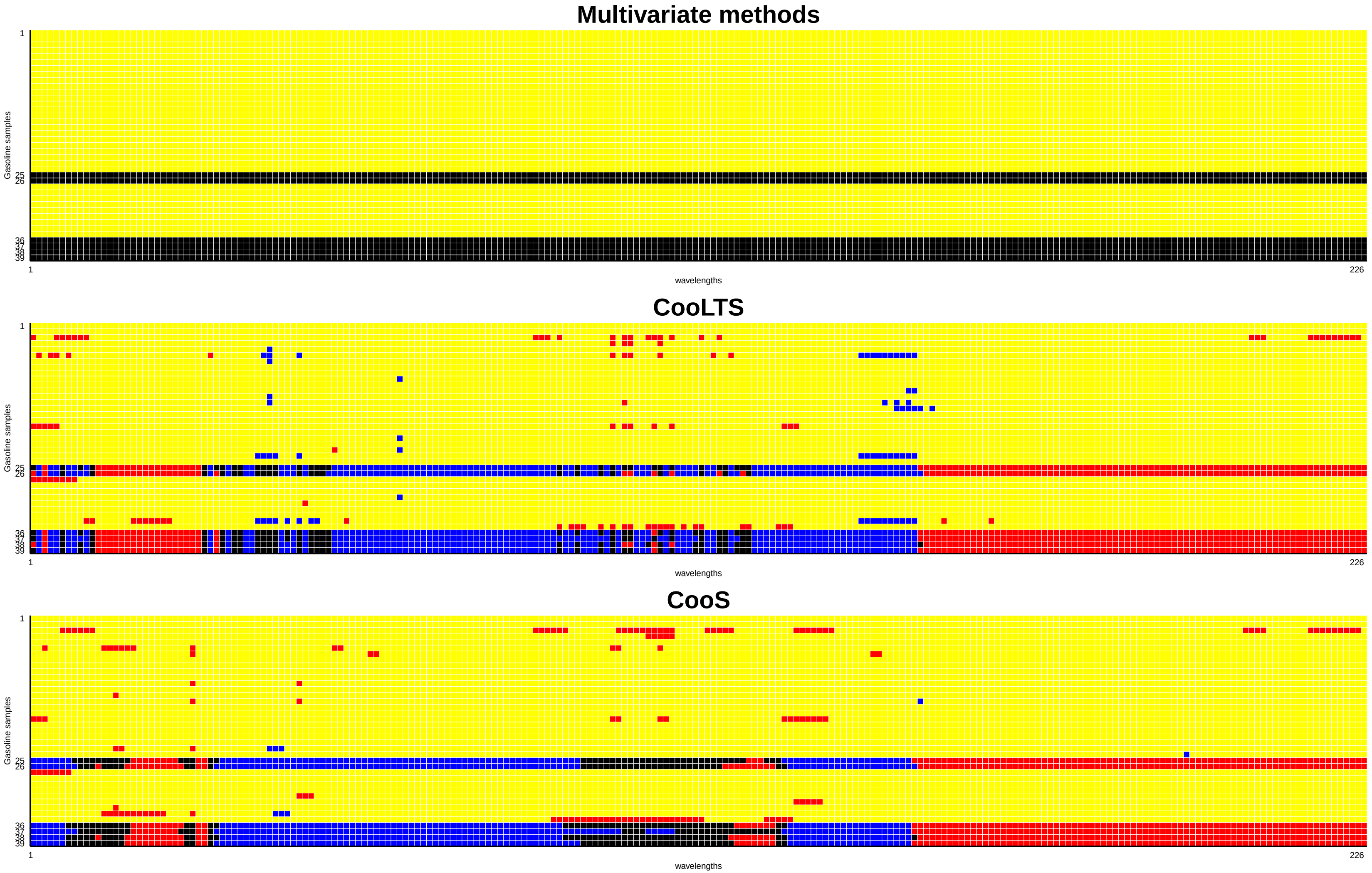


Figure 2.3: Cell maps for the Octane dataset with $n = 39$ gasoline samples and $p = 226$ wavelengths: when detecting casewise outliers with a multivariate-PCA method (top panel), when using CooLTS (middle panel) and when using CooS (bottom panel).

2.8 Discussion and conclusions

In this Chapter we introduced the Coordinatewise least trimmed squares (CooLTS) estimator for principal components. Our proposal is the least trimmed squares equivalent of the Coordinatewise S-estimator (CooS) of Boente and Salibian-Barrera (2015). In particular, CooLTS replaces the minimization of the sum of M-scales by the minimization of the sum of least trimmed squares scales to estimate the best $q-$dimensional linear space. Therefore, our estimator is suitable to handle problems with cellwise outliers in contrast to the multivariate methods of Chapter 1 which only targets casewise outliers. We then introduced the functional of the estimator which is Fisher-consistent at elliptical distributions. The latter can be proved using similar arguments as in Boente and Salibian-Barrera (2015). We obtained estimating equations derived from first order conditions and used them in our iterative algorithm proposed in chapter 1 which was adapted to fit coordinatewise PCA methods, and in particular our CooLTS estimator. Since the coordinatewise algorithm uses estimating equations it can also handle high-dimensional problems.

Results of a experiment confirm that coordinatewise-PCA methods are more suitable than multivariate-PCA methods in datasets with large fractions of contaminated observations. We also assessed outlier detection by coordinatewise-PCA methods on two real data examples. To decide whether a cell is outlying or not we used a similar approach as in Rousseeuw and Van den Bossche (2016). In the first example we showed that coordinatewise-PCA methods are able to detect most of the cellwise outliers detected by a purely outlier detection method. In the second example we looked at a realistic scenario where some of the cells of 'outlying observations' are actually regular cells. We showed that while multivariate-PCA methods completely discard those 'outlying observations' coordinatewise-PCA methods is a good alternative to identify those regular cells and flag only the outlying cells responsible for the 'outlying' behavior of those observations. Therefore, coordinatewise-PCA methods can also be used as an outlier detection tool, especially in high-dimensional settings where we dispose of only a few observations.

Chapter 3

Functional data setting

The content of this chapter is work in progress for future publication. This was a joint work with Prof. Matias Salibian-Barrera from the University of British Columbia (Canada).

3.1 Introduction

Principal component analysis was originally developed for multivariate data and later successfully adapted to accommodate functional data. This is known in the literature as functional principal component analysis (FPCA). Analogously to the multivariate case, FPCA also has the property of providing optimal approximations in the L^2 sense. Therefore, one of the main applications of FPCA is to obtain finite dimensional approximations of functional curves. Similarly as in the multivariate setting, FPCA may also be used to gain insight in the functional data by identifying the most important sources of variation of functional data. However, due to the squared loss function in its optimization problem classical FPCA approach is also very sensitive to abnormal data. However, there are not yet many proposals in the literature for robust functional principal component analysis. Locantore et al. (1999) seem to be one of the first to study this problem with spherical PCA. Gervini (2008) introduced a fully functional approach to robust spherical principal components. Later, Bali et al. (2011) proposed a robust projection-pursuit FPCA approach with raw estimation and various strategies to smooth principal components. They also showed consistency of the estimators for the eigenfunctions and eigenvalues of the underlying process. More recently, Sawant et al. (2012) adapted the BACONPCA estimator to the functional data setting and Boente and Salibian-Barrera (2015) introduced an S-estimator for functional principal components.

In this section we extend the Multivariate S-estimator, the Multivariate LTS estimator and the Coordinatewise LTS robust principal component estimators as presented in Chapters 1 and 2 to the functional data setting. These extensions use smoothed robust principal components by the Sieves method introduced in Bali et al. (2011). The Sieves smoothing method uses $B-$splines as a smoothing tool. Hence, we first project the functional data on a finite dimensional space by using appropriate basis functions, then we estimate the principal components in the finite dimensional space and finally we transform the solution back to the original functional space. The advantages of using smoothed FPCA have been shown in e.g. Rice and Silverman (1991), Ramsay and Silverman (2005) and Bali et al. (2011).

As discussed in the previous chapter, even with a small fraction of outlying cells high-dimensional datasets can contain only a small number of observations that are completely free of outliers. Since functional data are essentially infinite dimensional, this problem can become even worse in this setting. Therefore, we are particularly interested in studying the extension of the coordinatewise method based on LTS scales to the functional setting.

First, we present the extension of the MVLTS and MVS methods to the functional setting and introduce the corresponding functionals in the Hilbert space in sections 3.2.1 and 3.3.1. We then extend the coordinatewise LTS estimator to functional data in section 3.4 and define the corresponding functional. Empirical results of these methods in functional data with complicated patterns of contamination are presented in the simulation study of section 3.5. We also compare the proposed methods with other existing robust methods for functional data such as the Coordinatewise S-estimator of Boente and Salibian-Barrera (2015) and the Sieve-projection pursuit of Bali et al. (2011). Simulation results show that all robust methods perform equally well in general for small fractions of contaminated functional observations. When a large fraction of the curves is contaminated at some positions along its trajectory, then the multivariate methods break down. On the other hand, the coordinatewise methods still behave robustly if the curves are not all contaminated at the same positions. Finally, we illustrate that MVS, MVLTS, CooS and CooLTS are able to identify observations corresponding to anomalous events in a real data example in section 3.6.

3.2 The Multivariate least trimmed squares estimator for PCA (MVLTS) in the functional setting

We can extend the estimator defined in Section 1.3 for random vectors to accommodate functional data. The simplest setting corresponds to observations that are realizations of a stochastic process $X \in L^2(\mathcal{I})$ with $\mathcal{I}$ an interval of the real line which can be assumed to be between 0 and 1, i.e. $\mathcal{I} = [0,1]$. A more general setting corresponds to observations that are realizations of a random element on a separable Hilbert space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and norm $\| \cdot \|_{\mathcal{H}}$. Classical principal components for functional data is defined via the Karhunen-Loève decomposition of the covariance function of the stochastic process X. It has the property of providing the best lower-dimensional approximation in the L^2 sense.

In general, one rarely observes entire curves but instead observes only a finite set of discrete values for each of the curves. Moreover, in many applications the curves are observed at different design points t_{ij}, $1 \leq j \leq m_i$, $1 \leq i \leq n$. This means that the functional data for observation i usually correspond to values $x_{i1}, \dots, x_{im_i}$ with $x_{ij} = X_i(t_{ij})$, $1 \leq j \leq m_i$. Similarly as in Boente and Salibian-Barrera (2015), we assume that the number of points where each trajectory is observed increases with the sample size n, and that in the limit these points cover the whole interval $[0,1]$. Using the Sieves method of Bali et al. (2011) , each observed point in $\mathcal{H}$ is identified with the vector formed by its coordinates on an appropriate finite set of functional basis elements which increases with the sample size. Then, the procedure in Section 1.4 can be applied to these finite-dimensional vectors to obtain the estimate of the q-dimensional subspace, which can then be mapped back into $\mathcal{H}$.

More specifically, let $\delta_1, \dots, \delta_p$ be a set of orthogonal basis elements in $\mathcal{H}$ spanning the linear space $\mathcal{H}_p$. Let $x_{ij} = \langle X_i, \delta_j \rangle_{\mathcal{H}}$ be the coefficient of the ith curve on the jth element of the basis. After calculating this inner product in $\mathcal{H}$ for all elements of the basis, $1 \leq j \leq p$, we can form the p-dimensional vector $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^{\mathrm{T}}$. We can apply the procedure described in Section 1.4 to the multivariate set $Z_n = \{\mathbf{x}_i,\ i = 1, \dots, n\} \subset \mathbb{R}^p$ to obtain the q-dimensional linear space estimate $\mathcal{L}_{\widehat{\mathbf{B}}_{\mathrm{MVLTS}}}(Z_n)$ spanned by the orthogonal vectors $(\widehat{\mathbf{b}}^{(1)}, \dots, \widehat{\mathbf{b}}^{(q)})$ and the location estimate $\widehat{\mathbf{m}}_{\mathrm{MVLTS}}(Z_n) = (\widehat{m}_1, \dots, \widehat{m}_p)$, with scores $\widehat{a}_{il} = \widehat{\mathbf{b}}^{(l)\,\mathrm{T}}(\mathbf{x}_i - \widehat{\mathbf{m}}_{\mathrm{MVLTS}}(Z_n))$ and corresponding approximations $\widehat{\mathbf{x}}_i = \widehat{\mathbf{m}}_{\mathrm{MVLTS}}(Z_n) + \sum_{l=1}^{q} \widehat{a}_{il}\, \widehat{\mathbf{b}}^{(l)}$ Then, we can transform these MVLTS estimates back to the original Hilbert space $\mathcal{H}$. Hence, the MVLTS location estimate in $\mathcal{H}$ becomes $\widehat{\mu}_{\mathrm{MVLTS}} = \sum_{j=1}^{p} \widehat{m}_j\, \delta_j$ and the associated MVLTS estimates of the basis functions of the q-dimensional linear space are given by $\widehat{\phi}^{(l)}_{\mathrm{MVLTS}} = \sum_{j=1}^{p} \widehat{b}_{lj}\, \delta_j / \| \sum_{j=1}^{p} \widehat{b}_{lj}\, \delta_j \|_{\mathcal{H}}$, for $1 \leq l \leq q$. Finally, the approximations in $\mathcal{H}$ are $\widehat{X}_i = \widehat{\mu}_{\mathrm{MVLTS}} + \widehat{a}_{il}\, \widehat{\phi}^{(l)}_{\mathrm{MVLTS}}$.

3.2.1 The functional in $\mathcal{H}$

Before defining the functional corresponding to our estimator in $\mathcal{H}$ we introduce some notation. Denote the tensor product in $\mathcal{H}$ by $\otimes$. For any two elements $u, v \in \mathcal{H}$ the operator $u \otimes v : \mathcal{H} \to \mathcal{H}$ is defined as $(u \otimes v)w = \langle v, w\rangle\, u$ for $w \in \mathcal{H}$. Let X be a random element in a separable Hilbert space $\mathcal{H}$ and let $O_p : \mathcal{H} \to \mathbb{R}^p$ be any linear and bounded operator. Denote $\mathbf{x} \in \mathbb{R}^p$ the random vector defined by $\mathbf{x} = O_p\, X$ with $O_p : \mathcal{H} \to \mathbb{R}^p$ defined as

$$O_p = \sum_{j=1}^{p} \mathbf{e}_j \,\otimes\, \delta_j, \tag{3.1}$$

where $\mathbf{e}_j$, $1 \le j \le p$, are the elements of the canonical basis of $\mathbb{R}^p$. This means that $O_p\, X$ consists of the p coefficients of X on the basis $\delta_1, \ldots, \delta_p$. In general, let $\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(q)}$ denote the columns of the matrix $\mathbf{B}$ and let $\phi^{(l)}(\mathbf{B}) \in \mathcal{H}$ be given by

$$\phi^{(l)}(\mathbf{B}) = \sum_{j=1}^{p} b_{lj}\delta_j = (\sum_{j=1}^{p} \delta_j \otimes \mathbf{e}_j)\, \mathbf{b}^{(l)}, \qquad 1 \le l \le q. \tag{3.2}$$

We denote as $\mathcal{H}_{\mathbf{B}}$ the linear space spanned by the orthonormal elements $\phi^{(1)}(\mathbf{B}), \ldots, \phi^{(q)}(\mathbf{B})$.

Let $O_p^* : \mathbb{R}^p \to \mathcal{H}$ denote the adjoint operator of the linear and bounded operator O_p. By Definition 3.1 in Boente and Salibian-Barrera (2015) it follows that X has an elliptical distribution with parameters $\mu_{\mathcal{H}} \in \mathcal{H}$ and $\mathbf{\Gamma} : \mathcal{H} \to \mathcal{H}$, where $\mathbf{\Gamma}$ is a self-adjoint, positive semidefinite and compact operator, that is $X \sim P_{(\mu_{\mathcal{H}}, \mathbf{\Gamma})}$, if and only if the vector O_pX has a p-variate elliptical distribution with location parameter $O_p\, \mu_{\mathcal{H}}$ and scatter matrix $O_p\, \mathbf{\Gamma}\, O_p^*$, that is $O_p\, X \sim F_{(O_p\mu_{\mathcal{H}},\, O_p\mathbf{\Gamma} O_p^*)}$.

In what follows we assume without loss of generality that X follows a distribution with location $\mu_{\mathcal{H}} = 0$. Furthermore, to derive the Fisher consistency of this Sieves$-$approach we also assume that $O_p\, X \sim F_{(O_p\mathbf{\Gamma} O_p^*)}$. The functional corresponding to our estimator at a distribution $P_{(\mathbf{\Gamma})}$ of the functional random variable X is obtained by first projecting X onto a finite set of basis functions $\delta_1, \ldots, \delta_p$. Then, the problem turns into finding the MVLTS functional at the distribution $F_{(O_p\mathbf{\Gamma} O_p^*)}$. As in Section 1.3.2 we call G the distribution $F_{(O_p\mathbf{\Gamma} O_p^*)}$ and define the MVLTS functional at G as

$$\mathcal{L}_{\mathbf{B}_{\text{MVLTS}}}(G) = \mathcal{L}_{\mathbf{B}_{\text{LS},\widehat{E}}}(G) \in \min_{E \in \mathcal{D}_G(\alpha)} \Psi_p(\mathcal{L}_{\mathbf{B}_{\text{LS},E}}(G)) \tag{3.3}$$

where $\Psi_p(\mathcal{L}_{\mathbf{B}_{\text{LS},E}}(G)) = \int_E\, d_G^2(\mathbf{x}, \mathbf{B}_{\text{LS},E})\; dG(\mathbf{x})$ for subset $E \in \mathcal{D}_G(\alpha)$ with $\mathcal{D}_G(\alpha)$ defined in (1.35). The subscript p in (3.3) emphasizes that we are working at the level of the p-dimensional distribution G. Then, the MVLTS-PCA functional at $P_{(\mathbf{\Gamma})}$ is obtained by transforming the solution in (3.3) back onto $\mathcal{H}$.

Consider the spectral decomposition of the scale operator $\mathbf{\Gamma} = \sum_{j=1}^{\infty} \lambda_j\, \phi^{(j)} \otimes \phi^{(j)}$, where λ_j denotes the jth largest eigenvalue with associated eigenfunction $\phi^{(j)}$, $j \ge 1$. Assume that $\lambda_q > \lambda_{q+1}$ and that $\sum_{j\ge1} \lambda_j < \infty$. The best approximating q-dimensional linear space corresponding to distribution $P_{(\mathbf{\Gamma})}$ is then spanned by $\phi^{(1)}, \dots, \phi^{(q)}$. Proposition 3.1 in Boente and Salibian-Barrera (2015) can be used to show that the functional $\mathcal{L}_{\mathbf{B}_{\text{MVLTS}}}(G)$ in (3.3) when transformed back to the original Hilbert space $\mathcal{H}$ is equal to the linear space spanned by $\phi^{(1)}, \dots, \phi^{(q)}$ when the dimension p of the projection grows to infinity.

More specifically, let $\mathcal{H}_p$ be the linear space spanned by $\{\delta_1, \dots, \delta_p\}$ and $\Pi_p : \mathcal{H} \to \mathcal{H}_p$ be the projection operator over $\mathcal{H}_p$, that is, $\Pi_p = \sum_{j=1}^p \delta_j \otimes \delta_j$. In addition, define $O : \mathcal{H} \to \mathbb{R}^p$ by $O = \sum_{j=1}^p \mathbf{e}_j \otimes \delta_j$ with adjoint operator $O^* : \mathbb{R}^p \to \mathcal{H}$. It can be shown that if $\mathbf{u} \in \mathbb{R}^p$ is an eigenvector of $\mathbf{\Sigma}$ related to an eigenvalue γ, then $v = O^*\,\mathbf{u}$ is an eigenfunction of the compact operator $\Omega_p = \Pi_p \mathbf{\Gamma} \Pi_p^*$ associated to γ. Similarly, if v is an eigenfunction of Ω_p with eigenvalue γ, then Ov is an eigenvector of $\mathbf{\Sigma}$ associated with the same eigenvalue γ. Thus, the p largest eigenvalues of Ω_p are those of $\mathbf{\Sigma}$, where Ω_p has at most p non-null eigenvalues. Denote $\lambda_j(\Omega_p)$ the jth largest eigenvalue of the operator Ω_p. Then, by Theorem 1.5 which shows Fisher consistency of the functional $\mathcal{L}_{\mathbf{B}_{\text{MVLTS}}}(G)$ at $F_{\mathbf{\Sigma}}$, it follows that

$$\min_{E \in \mathcal{D}_G(\alpha)} \mathcal{L}_{\mathbf{B}_{\text{LS},E}}(G) = \mathcal{L}_{\mathbf{B}_{\text{LS},\widehat{E}}}(G) = \operatorname{tr}(\Omega_p) - \sum_{j=1}^{q} \lambda_j(\Omega_p), \tag{3.4}$$

since $\operatorname{tr}(\mathbf{\Sigma}) = \operatorname{tr}(\Omega_p)$. Then, $\mathbf{B}_{\text{LS},\widehat{E}}$ can be transformed back to the original variables by using (3.2), this yields the linear space $\mathcal{H}_{\mathbf{B}_{\text{LS},\widehat{E}}}$ spanned by $\phi^{(1)}(\mathbf{B}_{\text{LS},\widehat{E}}), \dots, \phi^{(q)}(\mathbf{B}_{\text{LS},\widehat{E}})$. By (3.4) we have that $\phi^{(j)}(\mathbf{B}_{\text{LS},\widehat{E}}) = \phi^{(j)}(\Omega_p)$. The following result can be derived from Proposition 3.1 in Boente and Salibian-Barrera (2015):

$$\lim_{p \to \infty} \Psi_p(\mathcal{L}_{\mathbf{B}_{\text{LS},\widehat{E}}}(G)) = \operatorname{tr}(\mathbf{\Gamma}) - \sum_{j=1}^{q} \lambda_j, \tag{3.5}$$

and therefore the linear space spanned by $\phi^{(1)}(\Omega_p), \dots, \phi^{(q)}(\Omega_p)$ converges to that spanned by $\phi^{(1)}, \dots, \phi^{(q)}$. In other words, the MVLTS linear space functional is Fisher consistent at distribution $P_{(\mathbf{\Gamma})}$, i.e. for elliptically distributed random elements in $\mathcal{H}$. The first part of Proposition 3.1 in Boente and Salibian-Barrera (2015) shows that Fisher consistency can be proved directly after transformation with (3.2) when assuming that the orthonormal basis δ_j is the basis $\phi^{(j)}$ for eigenfunctions of $\mathbf{\Gamma}$.

3.3 The Multivariate S estimator for PCA (MVS) in the functional setting

The MVS estimator in the functional setting is obtained analogously to the MVLTS estimator. We now use similar notation as in section 3.2. We first project the data on a sufficiently rich space of dimension p with orthonormal basis $\delta_1, \dots, \delta_p$ and obtain the p-dimensional dataset $Z_n = \{\mathbf{x}_i,\ i = 1, \dots, n\} \subset \mathbb{R}^p$ with coordinates $x_{ij} = \langle X_i, \delta_j \rangle_{\mathcal{H}}$. We then apply the procedure in section 1.4 and obtain the estimates $\widehat{\mathbf{B}}_{\text{MVS}}(Z_n) = (\widehat{\mathbf{b}}^{(1)}, \dots, \widehat{\mathbf{b}}^{(q)})$ and $\widehat{\mathbf{m}}_{\text{MVS}}(Z_n) = (\widehat{m}_1, \dots, \widehat{m}_p)$ with scores $\widehat{a}_{il} = \widehat{\mathbf{b}}^{(l)\,\text{T}}\,(\mathbf{x}_i - \widehat{\mathbf{m}}_{\text{MVS}}(Z_n))$. Finally, we map back the estimates to the original Hilbert space $\mathcal{H}$ by $\widehat{\mu}_{\text{MVS}} = \sum_{j=1}^{p} \widehat{m}_j\, \delta_j$ and for the orthogonal basis we have $\widehat{\phi}^{(l)}_{\text{MVS}} = \sum_{j=1}^{p} \widehat{b}_{lj}\, \delta_j / \| \sum_{j=1}^{p} \widehat{b}_{lj}\, \delta_j \|_{\mathcal{H}}$, for $1 \leq l \leq q$. This yields the MVS approximations in $\mathcal{H}$: $\widehat{X}_i = \widehat{\mu}_{\text{MVS}} + \widehat{a}_{il}\, \widehat{\phi}^{(l)}_{\text{MVS}}$.

3.3.1 The functional in $\mathcal{H}$

In this section we use the same definitions and notation of section 3.3.1. We define the functional of the MVS estimator at the elliptical distribution $P_{(\boldsymbol{\Gamma})}$ of the functional random variable X. We also assume that the multivariate vector $O_p X$ has a $p-$variate elliptical distribution $F_{(O_p \boldsymbol{\Gamma} O_p^*)}$. Since the procedure first make projections on a finite set of basis functions we first consider the functional at the distribution $F_{(O_p \boldsymbol{\Gamma} O_p^*)}$. As in section 1.14 we call G the distribution $F_{(O_p \boldsymbol{\Gamma} O_p^*)}$. The MVS functional at G is therefore defined as

$$\mathcal{L}_{\mathbf{B}_{\text{MVS}}}(G) \in \min_{\mathbf{B}_q^{\text{T}} \mathbf{B}_q = \mathbf{I}_q} \sigma_{\text{M}}(d_G(\mathbf{x}, \mathbf{B}_q)), \tag{3.6}$$

where $d_G(\mathbf{x}, \mathbf{B}_q) = \left\| \mathbf{x} - \mathbf{B}_q \mathbf{B}_q^{\text{T}} \mathbf{x} \right\|$ and the M-scale functional σ_{M} satisfies

$$\int \rho \left(\frac{d_G(\mathbf{x}, \mathbf{B}_q)}{\sigma_{\text{M}}(d_G(\mathbf{x}, \mathbf{B}_q))} \right) dG(\mathbf{x}) = b$$

Then, the MVS-PCA functional at $P_{(\boldsymbol{\Gamma})}$ is obtained by transforming the solution in (3.6) back onto $\mathcal{H}$. Using Proposition 3.1 in Boente and Salibian-Barrera (2015) and the Fisher consistency result in Theorem 1.1 it can be shown that the functional when transformed back to the original Hilbert space $\mathcal{H}$ is equal to the linear space spanned by $\phi^{(1)}, \dots, \phi^{(q)}$ when the dimension p of the projection grows to infinity.

3.4 Componentwise least trimmed squares estimator (CooLTS) in the functional setting

To define the CooLTS-PCA estimator for functional data, we consider as in Section 3.2 the general situation where curves are only partially observed at different design points t_{ij}, $1 \leq j \leq m_i$, $1 \leq i \leq n$, i.e. $x_{ij} = X_i(t_{ij})$. We again assume that the observed measurements cover the whole support of the curves when the sample size n increases to infinity. We consider the Sieves−approach for the for functional principal components, but we replace the MLTS-PCA that was used in Section 3.2 by the CooLTS-PCA procedure. Similar notations and definitions as for the MVLTS-PCA functional estimates in Section 3.2 apply therefore for functional CooLTS-PCA with obvious modifications.

3.4.1 The functional in $\mathcal{H}$

We define the CooLTS-PCA functional of a functional random element X in a separable Hilbert space $\mathcal{H}$ with distribution $X \sim P_{(\mathbf{\Gamma})}$. We again assume without loss of generality that X has location $\mu_{\mathcal{H}} = 0$. Furthermore, to derive the Fisher consistency of this Sieves−approach we also assume that $O_p X \sim F_{(O_p \mathbf{\Gamma} O_p^*)}$. The functional corresponding to our estimator is again obtained by first projecting X on a finite set of basis functions $\delta_1, \dots, \delta_p$. Then, the problem turns into finding the CooLTS-PCA functional at the distribution $F_{(O_p \mathbf{\Gamma} O_p^*)}$. In Section 2.3 the definition of the CooLTS-PCA functional at $G = F_{(O_p \mathbf{\Gamma} O_p^*)}$ is given by (2.14). It follows that the CooLTS-PCA functional at $P_{(\mathbf{\Gamma})}$ is obtained by transforming the solution in (2.14) back onto $\mathcal{H}$.

Similarly as for the MVLTS case, Proposition 3.1 in Boente and Salibian-Barrera (2015) can be adapted to show that the functional $\mathcal{L}_{\mathbf{B}_{\text{CoLTS}}}(G)$ corresponding to (2.14) when transformed back to the original Hilbert space $\mathcal{H}$ is equal to the linear space spanned by $\phi^{(1)}, \dots, \phi^{(q)}$ when the dimension p of the projection grows to infinity. This result can then be used again to show that the CooLTS-PCA linear space functional is Fisher consistent for random elements X in $\mathcal{H}$ with an elliptical distribution $P_{(\mathbf{\Gamma})}$.

3.5 Simulation

We consider the same designs as in Boente and Salibian-Barrera (2015) to investigate the finite-sample properties of the coordinatewise and multivariate estimators in the functional data setting. We are particularly interested in the performance of our Coordinatewise LTS estimator in functional data in presence of casewise or cellwise outliers. As in Boente and Salibian-Barrera (2015) the performance of the methods is assessed by measuring how well the estimators approximate regular curves on the one hand and correctly detect outlying curves on the other hand. We use deterministic starting values for our PCA algorithms since in the previous chapters this strategy showed the best performance for multivariate data settings at a lower computational time. We compare CooLTS with the CooS estimator of Boente and Salibian-Barrera (2015), the MVS estimator, the MVLTS estimator, the robust sieve projection-pursuit approach (PP) proposed in Bali et al. (2011) and the classical PCA approach (LS). We also consider the best $q-$dimensional linear space (True) according to the data generating process as a benchmark for all methods. Of course, this is not an estimator but a kind of oracle method that cannot be used in practice.

To calculate the functional PCA estimates based on CooLTS, CooS, MVLTS and MVS, we used the algorithm outline in Section 1.4 with deterministic starting values using the same parameter values as in the experiments in section 1.6. For the S-estimates we consider the Tukey's bisquare function for ρ with constants $c = 1.54764, b = 0.50$ and $c = 3, b = 0.2426$. For the LTS estimates we consider $\alpha = 0.5$.

3.5.1 Simulation design

We now describe the simulation designs in more detail. To investigate the influence of different outlier configurations on our estimators we consider the three different models used in Boente and Salibian-Barrera (2015). The first two models were constructed from a finite-rank process while the third follows an infinite-rank process. In all cases $n = 70$ functional observations were generated where each curve was observed at $m = 100$ equidistant instants in the interval $[0, 1]$. A total of 500 replications was generated for each setting. A cubic $B-$spline basis was used to project the functional data which in general does not show periodic patterns. The dimension of the basis was chosen to $p = 50$ to represent a realistic situation where the sample size is similar to the dimension of the data. In each model we consider different settings of contaminated data.

Model 1

This model was generated from a two-dimensional scatter operator so that regular curves follow a smooth trajectory. In particular, the non-contaminated curves $X_i \sim X$, $1 \leq i \leq n$, follow the model

$$X(t_s) = 10 + \mu(t_s) + \xi_1\phi_1(t_s) + \xi_2\phi_2(t_s) + z_s, \quad s = 1, \dots, 100,$$

where z_s are i.i.d additive errors that follow $N(0,1)$. The scores ξ_1 and ξ_2 are independent of each other and independent of z_s with $\xi_1 \sim N(0, 25/4)$, $\xi_2 \sim N(0, 1/4)$. The basis functions $\phi_1(t) = \sqrt{2}\cos(2\pi t)$ and $\phi_2(t) = \sqrt{2}\sin(2\pi t)$ correspond to the Fourier basis. The mean function is

$$\mu(t) = 5 + 10\text{sin}(4\pi t)\text{exp}(-2t) + 5\text{sin}(\pi t/3) + 2\text{cos}(\pi t/2).$$

To assess the performance of robust functional PCA methods a mixture of clean and contaminated trajectories is generated from the model:

$$X^{(c)}(t_s) = X(t_s) + VY(t_s), \quad s = 1, \dots, 100,$$

where $V \sim Bi(1, \epsilon_1)$ is independent of X and Y. The contamination process Y is given by $Y(t_s) = W_s\tilde{z}_s$ with $W_s \sim Bi(1, \epsilon_2)$ and $\tilde{z}_s \sim N(\mu^{(c)}, 0.01)$. W_s and $\tilde{z}_s$ are all independent. Observations without contamination correspond to $\epsilon_1 = 0$. Therefore with this model any trajectory $X(t_s)$ has a probability ϵ_1 of being contaminated and any cell t_s of the contaminated trajectories has a probability ϵ_2 of being shifted vertically. The shift is random normally distributed and tightly centered around $\mu^{(c)} = 30$ (upwards shift). For our simulations we considered the settings $\epsilon_1 = 0.10$ and $\epsilon_1 = 0.30$ with $\epsilon_2 = 0.30$ in both cases. For the worst scenario with $\epsilon_1 = \epsilon_2 = 0.30$, this means that we expect about $70 \times 0.3 \times 0.3 \approx 6$ outliers in each time instant of the functional data. An example of this scenario is shown in Figure 3.1. We also examined the amount of potential coordinatewise outliers in the projected data. We used as a criterion in each coordinate the highest value and the lowest value of the clean projected data. Experiments showed that at most 11 cells went outside these bounds in a coordinate. The actual fraction of contamination in each coordinate is thus still rather low although 30% of the curves is contaminated.

Model 2

This model was also generated from a two-dimensional scatter operator but with a slightly different process. The non-contaminated trajectories $X_i \sim X$ were generated as

$$X(t_s) = 150 - 2\mu(t_s) + \xi_1\phi_1(t_s) + \xi_2\phi_2(t_s) + z_s, \quad s = 1, \dots, 100,$$

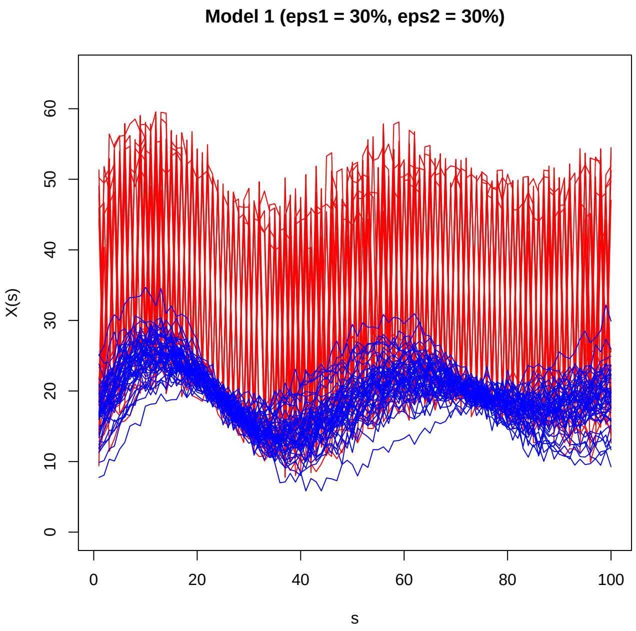


Figure 3.1: An example of Model 1 with $\epsilon_1 = \epsilon_2 = 0.30$. Regular curves are shown in blue color while contaminated curves are shown in red color

where z_s, ξ_1, ξ_2, μ, ϕ_1 and ϕ_2 are as in Model 1. To assess robustness of the methods a mixture of clean and contaminated trajectories is generated from the model

$$X^{(c)}(t_s) = \begin{cases} X(t_s) + VY(t_s) & \text{when } t_s < 0.4 \\ X(t_s) & \text{when } t_s \geq 0.4. \end{cases}$$

Hence, the contaminated curve are only contaminated in the first part of their trajectory, i.e. when $t_s < 0.4$. We take $V \sim Bi(1, \epsilon_1)$ independent of X and Y. The contamination process Y is generated by $Y(t_s) = W_s \tilde{z}_s + 2\mu(t_s)$ with $W_s \sim Bi(1, \epsilon_2)$, $\tilde{z}_s \sim N(\mu^{(c)}, 0.01)$ and $\mu^{(c)} = -5$. W_s and $\tilde{z}_s$ are all independent. Observations without contamination correspond to $\epsilon_1 = 0$. Contaminated curves start with a deviating trajectory in the first part of their range and then join smoothly with the trajectory of the regular curves. For our simulations we considered the settings $\epsilon_1 = 0.10$ and $\epsilon_1 = 0.30$ with $\epsilon_2 = 0.90$ in both cases. For the worst scenario of $\epsilon_1 = 0.30$, this means that we expect $70 \times 0.3 = 21$ outliers for each time instant when $t_s < 0.4$. An example of this scenario is shown in Figure 3.2. Similarly as in Model 1, for the projected data we found that at most 12 cells that lie outside the minimum and maximum bound of the regular data in each coordinate.

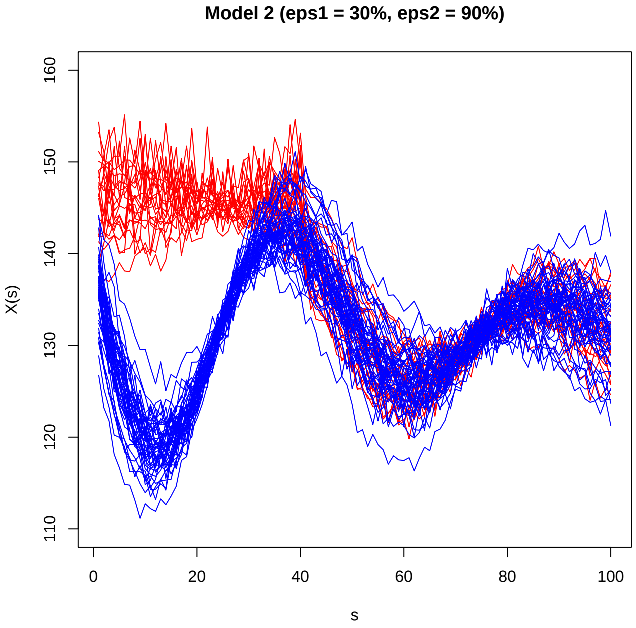


Figure 3.2: An example of Model 2 with $\epsilon_1 = 0.3$, $\epsilon_2 = 0.90$. Regular curves are shown in blue color while contaminated curves are shown in red color

Model 3

This model follows an infinite-rank stochastic process. Regular curves were generated from a Gaussian process with covariance kernel $\gamma_X(s,t) = 10\min(s,t)$. The eigenfunctions of the covariance operator are $\phi_j(t) = \sqrt{2}\sin((2j-1)(\pi/2)t)$, $j \geq 1$, with associated eigenvalues $\lambda_j = 10\,(2/\,[d(2j-1)\pi])^2$. To form data with good and contaminated trajectories we consider the model

$$X_i^{(c)}(s) = X_i(s) + V_i\,D_i\,\mathrm{M}\,\mathbb{I}_{\{T_i<s<T_i+\ell\}},$$

where $V_i \sim Bi(1,\epsilon)$, $\Pr(D_i = 1) = \Pr(D_i = -1) = 1/2$, $T_i \sim \mathcal{U}(0, 1-\ell)$, $\ell < 1/2$, with V_i, X_i, D_i and T_i independent of each other. We fix $\ell = 1/15$ and $\mathrm{M} = 30$. We consider different ϵ values for the model with these settings, namely $\epsilon = 0.10, 0.30$. An example of this configuration for $\epsilon = 0.30$ is shown in Figure 3.3. We see that with $\ell = 1/15$ contaminated curves make random jumps for about six time instants and then they return to the regular pattern. We also consider a configuration that uses the same model but fixes $D = 1$ (i.e. $\Pr(D = 1) = 1$) so that contaminated curves only have upwards shifts. For this configuration we set $\epsilon = 0.90$. An example of this configuration is shown in Figure 3.4. Even though we contaminate a large majority of 90% of the curves in this configuration, the amount of outliers in each time instant does not exceed 50% yet. This illustrates that even with a large fraction of contaminated curves, the data may still contain a lot of useful information from which the functional principal components can be estimated robustly by suitable methods.

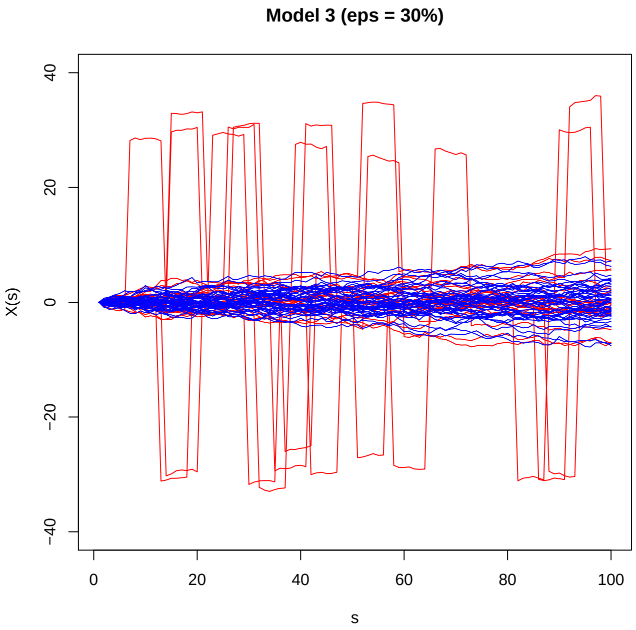


Figure 3.3: An example of Model 3 with $\epsilon = 0.30$. Regular curves are shown in blue color while contaminated curves are shown in red color

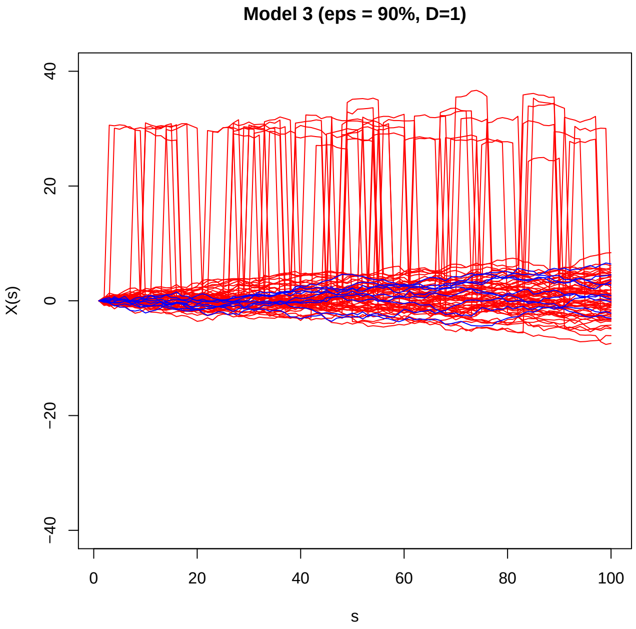


Figure 3.4: An example of Model 3 with $\epsilon = 0.90$ and $D = 1$. Regular curves are shown in blue color while contaminated curves are shown in red color

As in Boente and Salibian-Barrera (2015) we estimated a low dimensional approximation of dimension $q = 1$ for Models 1 and 2 since they were generated from a two-dimensional scatter operator and we used dimension $q = 4$ for Model 3 because this choice explains 95% of the variance of the underlying infinite-rank process.

3.5.2 Results

The measure to assess the quality of the approximation for each curve obtained by the FPCA methods is the squared residual norm in the original functional space $\|X_i - \hat{X}_i\|_{\mathcal{H}}^2$. The average squared residual norm over the regular curves in a data set assesses quality of the approximations for the regular data. Obviously, lower values of this average means a better approximation for these data. On the other hand, the average squared residual norm over outlying curves assesses to what extend the FPCA estimator is affected by the outliers. An FPCA method that is robust will give a high average for the outlying curves, indicating that this method succeeds better in identifying the outliers. Let γ_i be the indicator variable taking the value 1 when a curve X_i is an outlier and 0 otherwise. Then, the proportion of the total mean squared prediction error due to contaminated curves and clean curves respectively are:

$$\mathrm{PE}_{\mathcal{H},\mathrm{OUT}} = \frac{1}{n}\sum_{i=1}^{n} \gamma_i \|X_i - \hat{X}_i\|_{\mathcal{H}}^2 \tag{3.7}$$

$$\mathrm{PE}_{\mathcal{H},\mathrm{CLEAN}} = \frac{1}{n}\sum_{i=1}^{n} (1-\gamma_i) \|X_i - \hat{X}_i\|_{\mathcal{H}}^2. \tag{3.8}$$

The average squared residual norms over contaminated trajectories and over clean trajectories separately are:

$$\overline{\mathrm{PE}}_{\mathcal{H},\mathrm{OUT}} = \frac{\sum_{i=1}^{n} \gamma_i \|X_i - \hat{X}_i\|_{\mathcal{H}}^2}{\sum_{i=1}^{n} \gamma_i} \tag{3.9}$$

and

$$\overline{\mathrm{PE}}_{\mathcal{H},\mathrm{CLEAN}} = \frac{\sum_{i=1}^{n} (1-\gamma_i) \|X_i - \hat{X}_i\|_{\mathcal{H}}^2}{\sum_{i=1}^{n} (1-\gamma_i)} \tag{3.10}$$

respectively. To calculate the prediction errors for the best lower-dimensional predictions X_i^{True} according to the data generating process, we just replace the estimates $\hat{X}_i$ by the approximations X_i^{True} based on the optimal subspace according to this process in (3.7), (3.9) and (3.10). The average values over the 500 datasets of the performance measures $\mathrm{PE}_{\mathcal{H},\mathrm{OUT}}$, $\mathrm{PE}_{\mathcal{H},\mathrm{CLEAN}}$, $\overline{\mathrm{PE}}_{\mathcal{H},\mathrm{OUT}}$ and $\overline{\mathrm{PE}}_{\mathcal{H},\mathrm{CLEAN}}$ are reported in the tables below, using the labels "Out", "Clean", "$\overline{\text{Out}}$" and "$\overline{\text{Clean}}$" respectively.

Tables 3.1 and 3.2 summarize the results for Model 1 and 2 respectively, while Tables 3.3 and 3.4 show the results for the configurations of Model 3.

Table 3.1: Mean prediction errors over 500 replications for Model 1

	$\epsilon_1 = \epsilon_2 = 0.00$	$\epsilon_1 = 0.10$				$\epsilon_1 = 0.30$			
	Clean	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$
True	1.27	26.93	1.14	269.32	1.26	80.40	0.89	269.72	1.27
LS	1.25	18.96	5.06	193.37	5.68	56.59	5.07	189.69	7.21
CooLTS	1.40	27.34	1.22	270.95	1.36	79.30	0.93	269.69	1.31
CooS(c=1.5)	1.31	26.87	1.27	268.94	1.42	78.20	1.79	263.04	2.60
CooS(c=3)	1.25	26.92	1.13	269.24	1.25	75.07	1.68	254.76	2.50
MVLTS	1.29	27.31	1.16	270.60	1.29	79.39	0.89	269.99	1.25
MVS(c=1.5)	1.25	27.30	1.12	270.54	1.24	79.38	0.86	269.94	1.22
MVS(c=3)	1.24	27.30	1.12	270.51	1.24	58.53	4.13	203.85	6.00
PP	1.34	26.54	1.33	265.79	1.49	73.85	2.21	249.54	3.22

Table 3.2: Mean prediction errors over 500 replications for Model 2

	$\epsilon_1 = \epsilon_2 = 0.00$	$\epsilon_1 = 0.10$				$\epsilon_1 = 0.30$			
	Clean	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$
True	1.36	10.06	1.22	100.59	1.36	29.95	0.95	100.51	1.36
LS	1.34	1.60	4.03	19.53	4.51	2.52	4.12	8.48	5.87
CooLTS	1.49	10.10	1.37	100.45	1.52	29.48	1.03	100.22	1.45
CooS(c=1.5)	1.40	9.64	2.05	97.21	2.30	24.57	3.35	83.26	4.81
CooS(c=3)	1.35	9.84	1.38	99.23	1.54	4.11	3.86	16.23	5.55
MVLTS	1.38	10.07	1.24	99.85	1.38	29.30	0.96	99.40	1.35
MVS(c=1.5)	1.34	10.15	1.20	100.71	1.33	29.56	0.93	100.62	1.31
MVS(c=3)	1.34	10.14	1.20	100.63	1.33	4.96	3.50	22.59	5.08
PP	1.43	8.92	1.43	90.70	1.59	15.65	2.03	55.22	2.94

Table 3.3: Mean prediction errors over 500 replications for Model 3

	$\epsilon_1 = \epsilon_2 = 0.00$	$\epsilon_1 = 0.10$				$\epsilon_1 = 0.30$			
	Clean	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$
True	0.30	4.41	0.27	44.16	0.30	13.49	0.21	44.11	0.30
LS	0.29	2.07	0.66	18.46	0.74	9.55	0.72	30.95	1.04
CooLTS	0.48	5.20	0.44	44.87	0.50	14.37	0.30	45.58	0.44
CooS(c=1.5)	0.35	4.47	0.32	44.67	0.35	13.63	0.25	44.57	0.36
CooS(c=3)	0.30	4.41	0.27	44.15	0.30	13.48	0.21	44.05	0.30
MVLTS	0.33	5.21	0.29	44.93	0.33	14.11	0.21	44.77	0.30
MVS(c=1.5)	0.29	5.13	0.25	44.18	0.28	14.04	0.19	44.55	0.28
MVS(c=3)	0.29	5.12	0.25	44.13	0.28	11.14	0.48	35.55	0.71
PP	0.38	4.44	0.35	44.40	0.39	13.59	0.29	44.43	0.42

Let us first look at the results for Model 1 in Table 3.1. Without contamination the classical FPCA approach (LS) does a good job while the robust methods perform a little bit worse, indicating that their efficiency is lower. However, in presence of contamination the classical PCA does not perform well anymore while the other methods show robust behavior, with some advantage for the LTS methods and for the MVS estimator with $c = 1.5$. For Model 2 we see a similar behaviour in Table 3.2. However, when the fraction of contamination becomes larger ($\epsilon_1 = 30\%$) the differences between the LTS and MVS ($c = 1.5$) methods and the other procedures becomes larger. While we focused on a one dimensional approximations in Models 1 and 2, we consider four-dimensional approximations for the infinite rank process in Model 3. The results in Table 3.3 show that without contamination the classical PCA is again the best. However, as expected

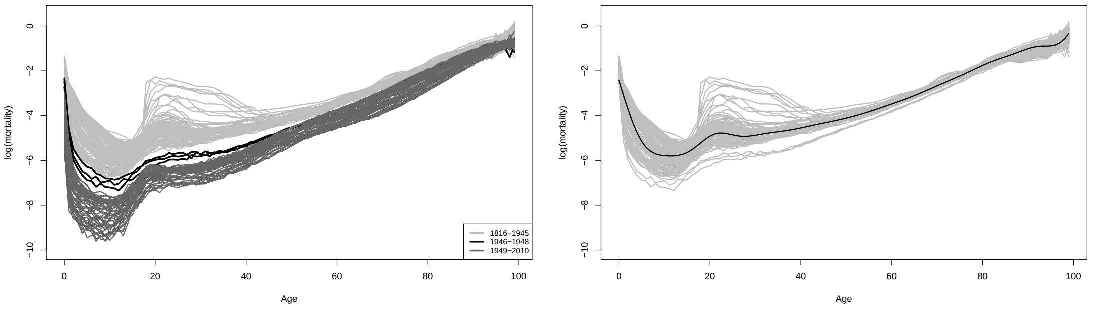
Table 3.4: Mean prediction errors over 500 replications for Model 3

	$\epsilon_1 = 0.90$, (D=1)			
	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$
True	39.71	0.03	44.11	0.31
LS	32.65	0.41	36.28	4.31
CooLTS	40.26	0.16	45.53	1.45
CooS(c=1.5)	40.34	0.05	44.81	0.52
CooS(c=3)	39.49	0.04	43.87	0.39
MVLTS	37.45	0.59	42.37	5.84
MVS(c=1.5)	33.06	0.43	37.41	3.95
MVS(c=3)	32.30	0.45	36.55	4.05
PP	40.24	0.04	44.69	0.38

the classical approach again quickly deteriorates in the presence of contamination. On the other hand, the robust methods show robust performance with similar results in Table 3.3. However, Model 3 seems not to produce severe contamination since even the classical PCA can discriminate between contaminated and regular curves in some cases. The configuration of Model 3 considered in Table 3.4 with contamination in 90% of the curves is a more challenging situation. Here, we clearly see that the multivariate methods MVS and MVLTS loose their robustness because too many curves are contamination. On the other hand, the coordinatewise approaches CooLTS and CooS remain robust. This is because even with this high fraction of contaminated curves, the fraction of contamination in each coordinate still remains below 50% as explained before. Therefore, the coordinatewise methods can still withstand this amount of contamination. We see that in particular the coordinatewise S-estimator shows the best performance in this setting. Note that also the projection pursuit (PP) approach shows competitive results in this scenario. For functional PCA based on CooLTS, CooS, MVLTS and MVS results based on the algorithms with initial estimates calculated from random subsets of size $q+1$ were also obtained (see Tables B.4-B.7 in the Appendix) and lead to similar conclusions as with deterministic starting values.

3.6 Real data example

In this section we present an application of the coordinatewise methods and the multivariate methods for functional data on real data. The goal is to obtain robust FPCA estimates and to identify potential atypical observations by examining the functional PCA approximations given by the methods. The dataset was analyzed in Boente and Salibian-Barrera (2015) with the Coordinatewise S estimator. We now re-analyze this data with the other procedures, namely the MVS, the MVLTS and our CooLTS estimator. Given the excellent results in previous sections we use our algorithms with deterministic starting values.



(a) Mortality data for the period 1816-2010 **(b)** Mortality data for the period of interest between 1816 and 1948

Figure 3.5: Mortality data. Panel (a) contains the curves for the years 1816 to 2010. Three periods can be distinguished and are marked with different gray scale colors. Panel (b) only depicts the curves corresponding to the period of interest from 1816 to 1948. On top the median curve is plotted.

3.6.1 Mortality data

The human mortality data is available on-line from the Human Mortality database (Human Mortality Database, 2013). Figure 3.5 shows the trajectories for this dataset. Panel 3.5a shows the entire dataset while panel 3.5b shows the period of interest. Every curve represents a different year and represents the death rate per age group for men in France. In particular, the logarithm of the death rate of people between the ages of 0 and 99 is shown. From 3.5a we can observe a clear difference in patterns of mortality before and after the second world war. This phenomenon may be attributed to technological advances and the change in quality of life in Europe after 1945. One can also notice a transitional period (1946-1948) where mortality curves lies between the two main periods. For the analysis we focus on the period between 1816 and 1948 that includes the pre-war time as well as the the transition period as shown in Panel 3.5b. The purpose of this analysis is to identify years with an atypical pattern of mortality. We computed the classical FPCA, FPCA based on S-estimators (MVS and CooS) with tuning constant $c = 3$ and FPCA based on LTS estimators (MVLTS and CooLTS) with $\alpha = 0.5$ to estimate the best 2 dimensional approximations. In the algorithms of the robust methods we used a projection onto a cubic $B-$spline basis of dimension $p = 20$. Figure 3.6 contrasts the robust fits with the classical FPCA fit.

From Figure 3.6 we can see that the robust methods identify mortality curves with a peak from ages 20 to 40 as outliers while the classical approach tries to accommodate these curves as well as possible. To detect outlying curves we use orthogonal distances between the observations and their projection on the estimated subspace. We use the cutoff value of Hubert et al. (2005) for the orthogonal distances in order to flag outliers.

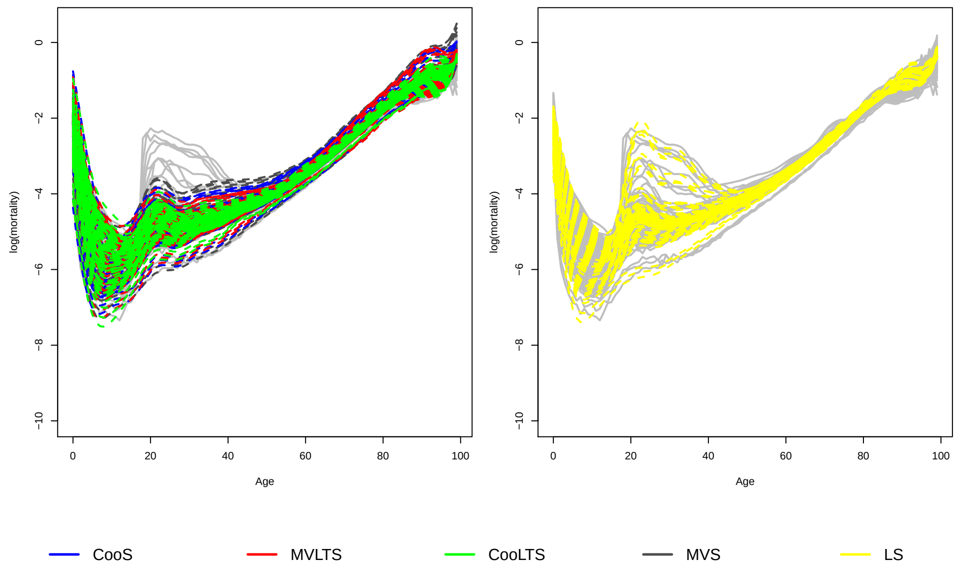


Figure 3.6: Robust approximations vs classical PCA approximations

Figure 3.7 shows for each of the methods the orthogonal distance of the observations with the corresponding cutoff to identify outlying observations.

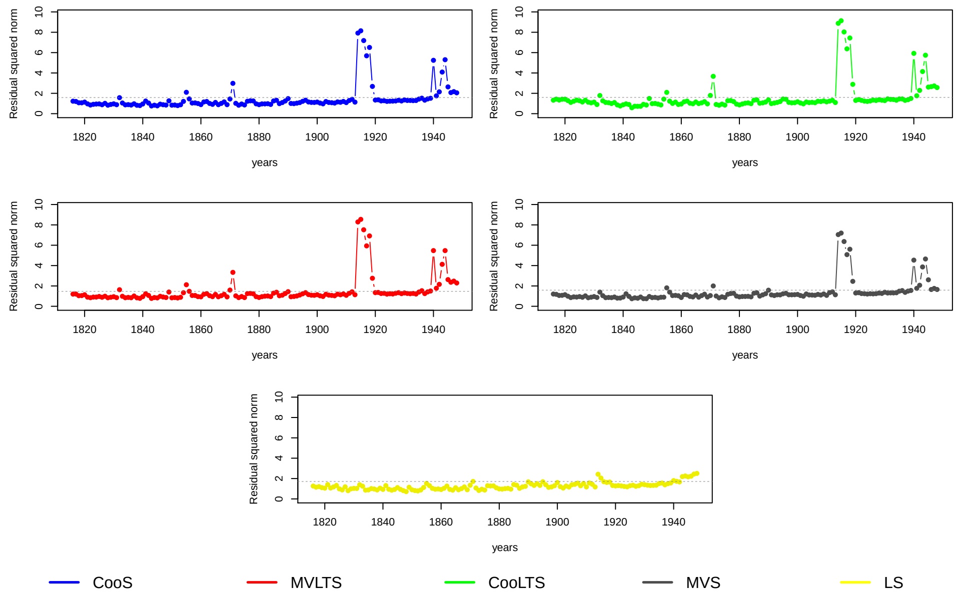


Figure 3.7: Outlier detection based on orthogonal distances

Figure 3.6 shows that all robust methods identify the following years as outliers: 1855, 1871, 1914-1919 and 1940-1948. All robust methods except MVS also identify year 1832 as a border case. On the other hand, classical PCA only identifies the years 1871, 1914-1915 1940, 1941 and 1943-1948 as mildly atypical. Similar results were obtained in Boente and Salibian-Barrera (2015) who applied the adjusted boxplot of Hubert and Vandervieren (2008) on the squared residual distances to identify outliers. It is interesting to see that years identified by the robust methods correspond to years of important events as pointed out by Boente and Salibian-Barrera (2015). In 1855 France was involved in the Crimean War and in 1871 in the Prussian War. The period 1914-1919 corresponds to World War I and the Spanish Flu epidemic. France falls to German occupation in 1941 and from there on France was involved in World War II until its end in 1945. The period 1946-1948 was a transitional period after the war. Note that classical FPCA is not able to detect the Crimean War, the last episodes of the World War I, the spanish flu and the early World War II casualties in France (1940 and 1942). Figure 3.8 shows the observed and the approximated curves for these events. We can clearly see that while the classical FPCA tries to fit these curves as well as possible, the robust methods are not attracted by these observations. Therefore, all robust estimators in this exampleare able to identify years with atypical events that have affected mortality rates in France.

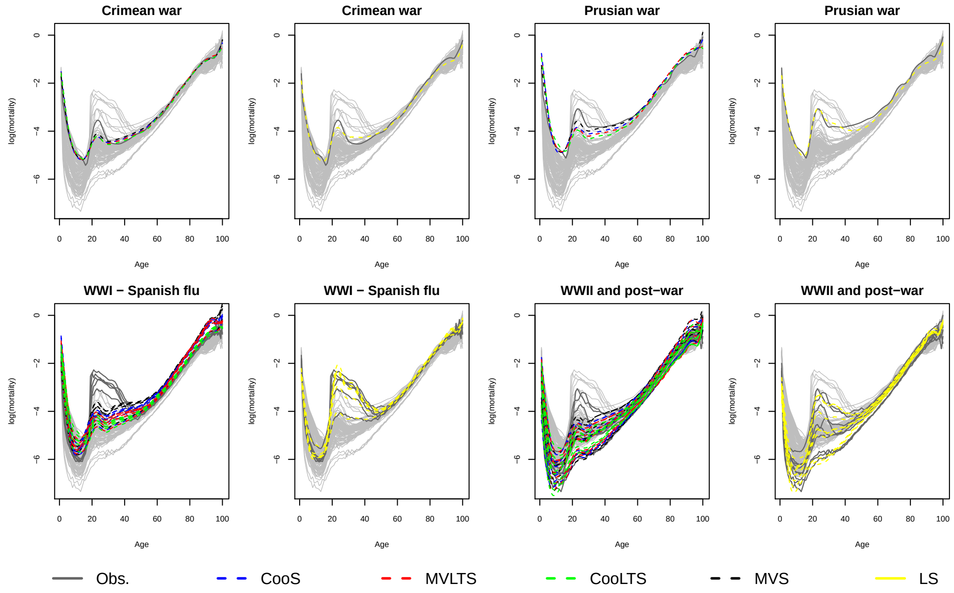


Figure 3.8: Observed curves and PCA approximations by the methods analyzed for years of important events in France

3.7 Discussion and conclusions

We have extended the MVS-PCA, the MVLTS-PCA and the CooLTS-PCA estimators to accomodate functional data. We calculate solutions for these extensions by using smoothed functional PCA according to the Sieves approach of Bali et al. (2011). Therefore, the functional data is first projected on a finite set of sufficiently rich basis functions, then the solutions of the estimators are obtained with the algorithms of chapter 1 and chapter 2 in the finite dimensional space and then these solutions are transformed back to the original functional space to obtain final estimates. Later, we introduced the functionals of the three extensions in the Hilbert space.

Experiments with complicated patterns of contamination but small fractions of outliers showed that the MVS-PCA, the MVLTS-PCA and the CooLTS-PCA estimators for functional PCA perform equally well when compared to other existing robust methods for functional PCA such as the Coordinatewise S-estimator and the Sieves-projection pursuit of Bali et al. (2011). When a large fraction of the curves is contaminated at some points along its trajectory, then the multivariate methods break down. On the other hand, the coordinatewise methods still behave robustly if the trajectories do not have much contamination at the same positions. This later result confirms the findings of the experiment in chapter 2 that coordinatewise methods, and in particular our CooLTS, are able to handle cellwise contamination. Finally, we show in a real example that the MVS-PCA, the MVLTS-PCA, the CooLTS-PCA and the CooS-PCA extensions for functional PCA are able to identify curves that correspond to atypical events.

Chapter 4

Tree-based prediction on incomplete data using imputation or surrogate decisions

The work in this chapter was published in Cevallos Valdiviezo and Van Aelst (2015).

4.1 Introduction

Many real datasets with predictive applications face the problem of missing values on useful features. Evidently, this complicates the predictive modeling process since predictive power may depend heavily on the way missing values are treated. In principle, missing data can occur in the training data only, in the individual test cases only, or in both the training data and test cases. In practice, however, missing data appear most often in both training and test set. Consider for instance customer data that is used to predict important outcomes such as buying preferences for individual costumers (based on their past actions). This type of data frequently contains missing values in both the training data and test cases, because the same amount of information is not available for all customers.

Most of the research work so far has addressed the problem of missing values in the training data (see e.g. Rubin (1987); Schafer (1997); Feelders (1999); Dempster et al. (1977); Batista and Monard (2003); Hapfelmeier and Ulm (2014)). On the other hand, Saar-Tsechansky and Provost (2007) is one of the only contributions in which the prediction accuracy of classification techniques is compared when only test cases contain missing values. Tree-based classifiers have been investigated for test cases with data missing completely at random (MCAR), i.e. test cases with missingness which does not depend on any value of the data. The performance of prediction methods for different missing data strategies when missing data occur in both the training and test set has been assessed in Hapfelmeier et al. (2012); Kapelner and Bleich (2013); Rieger et al. (2010). However, in Rieger et al. (2010) k-nearest neighbors (kNN) imputation was applied separately on the training and test samples. This is a potential weakness for practical purposes because the kNN imputation is impossible for test cases that appear on a case-by-case basis. Similarly, in Hapfelmeier et al. (2012) and Kapelner and Bleich (2013) imputation models were applied separately to the training and test cases. Moreover, the response variable was used in the imputation model for the training data so that the same imputation scheme cannot be applied to test cases arriving one-by-one. In this study, we are interested in methods that can deal with missingness in both training and test cases. Moreover, the methods should be able to handle test cases that appear one-by-one, because this case is often encountered in practical applications. Think for example of new potential patients for which a prediction needs to be made as soon as possible on a case-by-case basis, using the available information of the patient (such as clinical test results).

In this chapter we compare several strategies to handle missing data when using tree-based prediction methods. We focus on trees because they have several advantages and few limitations compared to other prediction techniques. Firstly, trees allow to handle data of different type (categorical, discrete, continuous). Other features that make trees highly popular among practitioners are their ability to capture important dependencies and interactions. Moreover, tree-based ensembles such as random forests can easily handle high dimensional problems and often show good performance without the need to fine-tune parameters. Trees also include a built-in methodology to process observations with missing data, called surrogate splits Breiman et al. (1984).

Evidently, if the missing data issue is not addressed correctly, misleading predictions may be obtained. Thus, one aims for prediction rules that have low bias (accurate enough) and low variability (stable enough) and at the same time take into account the additional uncertainty caused by missing values. Among the strategies to handle the missing values are:

1. Discard observations with any missing values in the training data
2. Rely on the learning algorithm to deal with missing values in the training phase
3. Impute all missing values before training the prediction method

Approach 1 encompasses ad-hoc procedures like complete case and available case analysis. They have been shown to work for relatively small amounts of missing data and under certain restrictive conditions Vach (1994); White and Carlin (2010). However, this approach is not applicable when missing values are present in test cases. Tree methods with surrogate splits are an example of the second approach. An advantage of strategy 2 is that incomplete data need not be treated prior to model fitting. For most learning techniques, the third approach is necessary to handle incomplete values or it simply helps to improve predictive capability. Many imputation methods have been developed to address the missing data issue in general. Imputation methods have been studied extensively with regard to inference: unbiasedness of estimates, efficiency, coverage and length of confidence intervals or power of tests (see e.g. Little and Rubin (2002); Burgette and Reiter (2010); Shah et al. (2014); Doove et al. (2014)). Other works study the performance of imputation methods when estimating the true values of the missing data, without considering the subsequent statistical analysis (see e.g. Liao et al. (2014); Stekhoven and Bühlmann (2012)). However, there is much less known about the properties of imputation methods in the context of prediction. An advantage of Approach 3 is that it completely separates the missing data problem from the prediction problem. This strategy thus gives freedom to (third party) analysts to apply any appropriate data mining method to the imputed data.

A few comparisons of approach 2 and 3 have already been considered in the literature. For instance in Feelders (1999) CART using surrogates was compared to CART preceded by single or multiple imputation. Two classification problems were considered. Multiple imputation performed clearly better than both single imputation and surrogates. Single imputation outperformed surrogates for a fraction of missingness above 10%. No ensemble methods were considered.

The predictive performance of conditional random forests Hothorn et al. (2011) with missing data was investigated in Rieger et al. (2010). Conditional random forests (CondRF) combined with surrogates was compared to CondRF with prior kNN imputation. Both classification and regression problems were considered. No difference in performance was found between handling missing values by surrogates or with prior kNN imputation. Recently, Hapfelmeier et al. (2012) compared the predictive performance of CART, conditional inference tree (CondTree) and CondRF in combination with surrogates or Multiple Imputation by Chained Equations (MICE) to handle the missing data. Real datasets with and without missing cells were used. The complete data were used for a simulation study in which missing values were introduced completely at random. For the real data with missing values MICE did not show a convincing improvement compared to surrogates, while in their simulation study MICE was beneficial for large amounts of missing data introduced in many variables. However, the authors argue that their simulation results may lack generalizability due to restrictive and artificial simulation patterns. Therefore, it is suggested to extend their simulations to a wider range of patterns.

So far, there is no clear conclusion in the literature about which combinations of tree-based prediction method and missing data strategy yield the most satisfactory predictions. It seems that an answer to this question may depend on the structure of the predictors, the type of relationship between predictors and response variable, and the pattern and fraction of missing data.

The contribution of this chapter of the thesis is threefold. First, we provide a theoretical comparison of prediction techniques that can be constructed from incomplete training data and can be applied directly on individual test cases with missing values, as this corresponds to most of the practical applications. Secondly, we set up a framework for the empirical comparison of these prediction techniques. Thirdly, using this framework, we provide some insight into the effect of different missing data patterns on the performance of 26 of these techniques based on trees.

In our comparison we consider as learning methods CART, CondTree, Random Forest (RF), CondRF, Bagging and Conditional Bagging (CondBagging). The procedures to handle missing data are surrogates, single imputation by median/mode, proximity matrix or kNN, and multiple imputation by MICE or Multiple Imputation by Sequential Regression Trees (MIST). Not all combinations have been implemented in R R Development Core Team (2011) which we use for our investigation. The 26 techniques in our comparison are summarized in Table 4.1.

Our comparison incorporates recent tree-based methods and imputation procedures for which there are almost no research results available about their predictive performance in presence of missing values. Any analysis or discussion of the situations under which the different techniques predict well or poorly is still lacking. Our empirical comparison shows that for moderate to large amounts of missing data, multiple imputation by MICE or MIST followed by CondRF is advisable, although these techniques are expensive in terms of computation time. Their better performance is due to the mutual effort of the imputation strategy and prediction method to average out sampling variability and variability due to missing data. This result of our empirical comparison is confirmed by the theoretical derivations. CondBagging using surrogate decisions emerges as an alternative with good performance and much lower computation time. For small amounts of missing data, any ensemble method with surrogate decisions or preceded by single imputation suffices to get a good prediction performance at a cheaper computational cost.

Table 4.1: Overview of the 26 techniques investigated in this study. Each mark ‘×’ corresponds to a technique. The second mark in the MIST + RF box corresponds to a special case of this technique that consists of imputing bootstrap samples by MIST + RF. N/I stands for “not implemented”.

Strategy for miss. data	Imputation method	CART	CondTree	RF	CondRF	Bagg.	CondBagg.
Surrogates	None	×	×	N/I	×	×	×
Single Imp.	Median/mode	×	×	×	×	N/I	N/I
	Prox.matrix	×	×	×	×	N/I	N/I
	*k*NN	×	×	×	×	N/I	N/I
Multiple Imp.	MICE	×	×	×	×	N/I	N/I
	MIST	×	×	××	×	N/I	N/I

4.2 Methodology

4.2.1 Tree-based methods

The Classification and Regression Tree (CART) algorithm proposed by Breiman et al. (1984) is a popular technique to fit trees. While it is an intuitively appealing procedure, it also has some drawbacks: it is known to be highly unstable due to its hierarchical nature Marshall and Kitsantas (2012); Hastie et al. (2009) and it tends to produce selection bias towards continuous and categorical features with many possible splits and missing values. Aiming to solve the latter problem, Hothorn et al. (2006) proposed the conditional inference tree (CondTree) algorithm which utilizes a unified framework for conditional inference. More specifically, CondTree allows for unbiased selection of the splitting variable by using univariate P-values which can be directly compared among covariates measured at different scales. However, CondTree might still be an unstable procedure due to its hierarchical nature.

With the aim of reducing the prediction variance of single trees, Bagging was proposed Breiman (1996a). It fits the noisy CART algorithm many times to bootstrap-sampled versions of the data Efron (1979) and averages for each observation the outcomes of individual trees to obtain a final prediction. However, overfitting may arise because trees are fitted on modified versions of the same original sample. This limits the benefits of Bagging. Hence, Random Forest Breiman (2001) was developed to further improve the prediction variance reduction of Bagging by decreasing the correlation among trees. This is established by adjusting the splitting process during the growing of the tree. Instead of considering all features for each split, only a number $g \leq p$ of predictors selected at random are considered as candidates for a split.

In the same spirit, Conditional Bagging and Conditional inference Forests were developed to combine the benefit of unbiased variable selection with reduction of the prediction variance Hothorn et al. (2011).

Surrogate splits, as introduced in Breiman et al. (1984), are an attempt to mimic the primary split of a region in terms of the number of cases sent down the same way. For any observation with a missing value for the primary split variable, we can find among all variables with nonmissing value for that case the predictor and corresponding split point producing the best surrogate split (i.e. the split yielding the most similar results as the best split). Quinlan (1993) considers surrogate splits as a special case of predictive value imputation. All tree-based methods can in theory handle missing predictor values by using the principle of surrogate splits. However, the implementation of RF in the R package randomForest Liaw and Wiener (2002) cannot be used on incomplete data. More information about tree-based methods is given in the Appendix C.

4.2.2 Imputation methods

An imputation can be the mean or a random draw from a predictive distribution that is specifically modeled for each missing entry Little and Rubin (2002). Thus, an imputation method is required to estimate these predictive distributions based on the observed data. In general, an advantage of using an imputation strategy is that it separates the missing data problem from the prediction problem. Hence, a completed dataset(s) can be used for the prediction problem. This allows to apply the most appropriate prediction method on the imputed dataset(s). We now give a short description of the imputation methods used in this chapter.

Single imputation (SI) methods

A rapid and simple fix to the problem of missing predictor values consists of just replacing them with the column median or mode, depending on the type of predictor variable. However, this method might distort the covariate distribution by underestimating its variance and also the relations between the covariates may be disturbed.

A more elaborate method consists of imputing based on the *proximity matrix* Liaw and Wiener (2002), which is a $N \times N$ matrix (N being the size of the training sample) that comes “for free” in the output of the Random Forest implementation in R. Each cell of this matrix contains the proportion of the total number of trees in the forest in which the respective pair of training observations share a terminal region. The proximity matrix algorithm starts with a median/mode imputation. Then, Random Forest is called with the completed data. The imputed values are updated according to the current proximity matrix. For continuous predictors the imputation update is the weighted average of the initially non-missing observations, where the weights are the proximities. For categorical predictors the imputation update is the category with the largest average proximity. This process is repeated iteratively, usually five times. Thus, the intuitive idea is to give a larger weight to cases that are more like the case with missing data.

Another single imputation method is kNN imputation Troyanskaya et al. (2001). This procedure looks for the k nearest neighbors of the missing observation with respect to their Euclidean distance computed from the remaining observed variables. Eventually, the missing value is replaced by a weighted mean of the k nearest neighbors, where the weights are based on the kNN euclidean distances.

After imputation by a SI method, the filled-in data are treated as if they were actually observed. The additional uncertainty caused by missing data on top of the already "available" sampling variance is thus ignored. As a consequence, the whole prediction rule may lose stability and hence prediction performance.

Multiple imputation (MI) methods

One way to take into account the variability caused by missing data is through multiple imputations Rubin (1987, 1996). This creates several training datasets differing only in the imputed fields. The variability across these completed versions of the data reflects the uncertainty underlying the imputed values.

Let M denote the data matrix and D the total number of imputed datasets by MI. As described in Little and Rubin (2002), multiple imputation draws the missing values for the oth imputed dataset ($o = 1, \dots, D$) as:

$$M^{(o)}_{\text{mis}} \sim Pr(M_{\text{mis}}|M_{\text{obs}}), \tag{4.1}$$

with

$$Pr(M_{\text{mis}}|M_{\text{obs}}) = \int Pr(M_{\text{mis}}|M_{\text{obs}}, \boldsymbol{\theta}) Pr(\boldsymbol{\theta}|M_{\text{obs}})\, \mathrm{d}\boldsymbol{\theta}. \tag{4.2}$$

That is, the imputed values are random draws from the joint posterior distribution of the missing data given the observed data. However, it is often difficult to draw from this predictive distribution due to the requirement of integrating over the model parameters $\boldsymbol{\theta}$ in (4.2). In the univariate case, Data Augmentation Tanner and Wong (1987) accomplishes this by iteratively drawing a sequence of values of the parameters and missing data until convergence. More specifically, data augmentation can be run independently D times to generate D iid draws from the approximate posterior distribution involving D estimates $\boldsymbol{\theta}^{*(1)}, \boldsymbol{\theta}^{*(2)}, \dots, \boldsymbol{\theta}^{*(D)}$ from $Pr(\boldsymbol{\theta}|M_{\text{obs}})$ which are subsequently used in the conditional distributions $Pr(M_{\text{mis}}|M_{\text{obs}};\boldsymbol{\theta}^{*(o)})$ to draw D imputations. However, in situations with multivariate data involving nonlinear relationships, building one coherent model for the joint distribution of the variables may be difficult. In those situations, simpler methods that approximate draws from (4.1) should be considered. We now discuss two such methods which are used in our comparison.

Multivariate Imputation by chained equations (MICE)

In real multivariate settings with more than one variable containing missing values, we might be able to approximate draws from (4.1) by specifying for each incomplete variable a conditional model for the missing data given a set of other variables. Essentially, for each variable containing missing values MICE draws values for the parameters and imputations from the corresponding conditional model and iterates this procedure through the other incomplete variables. Hence, the procedure splits the p-dimensional problem into p one-dimensional problems. By modeling only conditional distributions many complexities of real-life multivariate data such as predictors of different type, existence of nonlinear relations or interactions between variables and circular dependence can be addressed Burgette and Reiter (2010); Doove et al. (2014); Van Buuren (2012); Shah et al. (2014). These complexities are difficult to handle if a joint modeling approach Schafer (1997) is adopted. The reason is that in joint modeling an explicit multivariate distribution for the missing data needs to be specified to derive conditional models for imputations. Thus, distributional assumptions are imposed which may lack flexibility to address the above mentioned complexities. On the other hand, MICE (also called fully conditional specification [FCS] by Van Buuren et al. (2006)) directly specifies conditional models without the need of an explicit multivariate model for the entire dataset. Instead, the algorithm assumes that an underlying multivariate model exists and that draws from it can be generated by iteratively sampling from the conditionally specified imputation models.

Let $\mathbf{X}$ be the $N \times p$ matrix that contains the partially observed values for the p predictor variables. Then, $Pr(\mathbf{X}_{\text{mis}}|\mathbf{X}_{\text{obs}})$ denotes the joint *multivariate* posterior where $\mathbf{X}_{\text{mis}}$ and $\mathbf{X}_{\text{obs}}$ are the missing and observed parts of $\mathbf{X}$, respectively. Assume that the multivariate distribution of $\mathbf{X}$ is completely specified by $\boldsymbol{\theta}$, a p-dimensional vector of unknown parameters. MICE aims to obtain the posterior distribution of $\boldsymbol{\theta}$ through chained equations which form parametric models for the conditional distributions. More precisely, if all p predictors contain missing data, then starting from a simple draw from the observed marginal distributions the tth iteration of chained equations is a Gibbs sampler that successively draws:

$$\begin{aligned} &\theta_1^{*(t)} \sim Pr(\theta_1 | x_1^{\text{obs}}, x_2^{t-1}, \ldots, x_p^{t-1}) \\ &x_1^{*(t)} \sim Pr(x_1^{\text{mis}} | x_1^{\text{obs}}, x_2^{t-1}, \ldots, x_p^{t-1}, \theta_1^{*(t)}) \\ &\vdots \\ &\theta_p^{*(t)} \sim Pr(\theta_p | x_p^{\text{obs}}, x_1^{t}, x_2^{t}, \ldots, x_{p-1}^{t}) \\ &x_p^{*(t)} \sim Pr(x_p^{\text{mis}} | x_p^{\text{obs}}, x_1^{t}, x_2^{t}, \ldots, x_{p-1}^{t}, \theta_p^{*(t)}), \end{aligned} \tag{4.3}$$

where $x_j^{(t)} = (x_j^{\text{obs}}, x_j^{*(t)})$ is the jth imputed feature at iteration t and $\theta_1, \ldots, \theta_p$ are the components of $\boldsymbol{\theta}$ (see Van Buuren and Groothuis-Oudshoorn (2011); Van Buuren et al. (2006)).

MICE deviates from Markov Chain Monte Carlo (MCMC) approaches in that the sequences of univariate regressions are applied to cases with observed x_j. After convergence, it is implicitly assumed that the Gibbs sampler in (4.3) provides a draw $\boldsymbol{\theta}^*$ from its posterior which can be used to draw values $\mathbf{X}^*$ to impute $\mathbf{X}_{\text{mis}}$. Van Buuren and Groothuis-Oudshoorn (2011) states that convergence of the algorithm can be quite fast (10 iterations might be enough) since previous imputations $x_j^{*(t-1)}$ only enter $x_j^{*(t)}$ through their relation with other variables. This procedure can be run in parallel D times to generate D imputations. Various authors have shown the satisfactory performance of this method in a variety of simulation studies (e.g. Van Buuren et al. (2006); Horton and Kleinman (2007); Hapfelmeier and Ulm (2014)). As mentioned earlier, MICE also gives the user flexibility to specify a convenient imputation model for each variable in order to help preserving important characteristics of the data. Due to its construction, this approach is suitable for data missing at random (MAR), i.e. data whose missingness depends only on the observed data, although Van Buuren and Groothuis-Oudshoorn (2011) argues that MICE can also handle data missing not at random (MNAR) under additional modeling assumptions. Data MNAR occur when the missingness depends on unobserved data.

Despite the mentioned benefits, the MICE algorithm also has some shortcomings. For instance, it is not guaranteed that the specified conditional models in the Gibbs sampler will eventually converge to an existing stationary distribution. This problem is known as incompatibility of the conditionals which however is not considered a serious problem in practice Van Buuren et al. (2006). Another issue is that the standard MICE implementation uses parametric (generalized) linear models to estimate the conditional distributions in (4.3). Therefore, it might not be able to capture complex relations among variables, especially when having a large number of predictors.

Multivariate Imputation by Sequential Regression Trees (MIST)

MIST has been proposed in Burgette and Reiter (2010) with the goal of better capturing interactions and nonlinear relations among predictors when imputing missing values. MIST uses CART to model the conditional distribution of each missing predictor in (4.3). The authors justify their choice for CART by stressing that it is sufficiently flexible to capture complex structures without parametric assumptions or data transformations. After convergence, approximate draws from the predictive distribution of the incomplete targeted predictor can be taken by sampling elements from the final region that corresponds to the covariate values of the case of interest. A Bayesian bootstrap Rubin (1981) is performed within each final region before sampling in order to reflect the uncertainty about the population conditional distributions Burgette and Reiter (2010). Another benefit of this strategy is that potential problems that may arrive when imputing, such as nonsensical or impossible imputations, are avoided because MIST imputations come from the observed values.

Summary

There are two sources of uncertainty that might prevent us to produce good prediction results when using data with incomplete features: one is the inherent sampling variability and the other is the additional uncertainty caused by missing data. The former is well-known and can affect the performance of highly data-driven prediction methods such as single tree methods. The latter can make the prediction rule unreliable if not treated adequately, even if the prediction method itself is very stable. For instance, if the imputation is poor then the predictions can become unreliable no matter how well the learning method performs. This can happen when applying a single imputation prior to the learning method.

Procedures that combine MI with an ensemble of trees might potentially yield superior results, thanks to the mutual effort of the imputation strategy and prediction method to reduce variability of predictions. In particular, they tend to average out not only the variability present between trees (intra-forest variability), but also the variability due to the missing data by fitting a forest for each of the D imputed datasets (between-forest variability). Our theoretical derivation in Section 4.3 confirm the high potential of MI with ensembles to give accurate predictions. In our empirical investigation (Sections 4.4 and 4.5) we examine to what extent these procedures can indeed outperform the other alternatives in practical settings.

We also consider an alternative to multiple imputation, as introduced in He (2006), that also aims to take into account the variability due to missing data. Since this procedure showed good results in He (2006), we investigate its performance in our study. The technique first constructs B bootstrap samples from the original incomplete sample. Next, each of these bootstrap samples is imputed once. Although only a single imputation is applied on each bootstrap sample, we end up with B imputed bootstrap samples which may reproduce the variability of the imputation model. In He (2006) Gaussian, Logistic or Poisson regression is used to generate imputations, but we adapted the procedure by using MIST to impute the bootstrap samples (which thus yields MIST imputed bootstrap samples). This implies that no initial imputation is needed in contrast to the original procedure. RF is then applied on each of the imputed bootstrap samples, resulting in an ensemble of B forests. Finally, the results of all forests are averaged to obtain the final predictions. Similar to the previous strategy both intra-forest variability and between-forest variability is averaged out so that both sampling variability and missing data variability might be taken into account.

4.3 Theoretical properties

The derivations in this section form a basis to theoretically compare the properties of the methods analyzed in this study. Let us denote by $\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})$ a single tree predictor at $\mathrm{X} = \mathbf{x}$ after imputation of missing values in the training set by a single random draw from their predictive distribution. Here, $\mathcal{L}_{\text{miss}}$ denotes the missing part of the training set and ϕ the single imputation on those data by a given imputation method. For a regression problem, consider the expected generalization error at $\mathrm{X} = \mathbf{x}$ according to the squared error loss function:

$$\mathrm{E}_{\mathcal{L}}\{\mathrm{Err}\,\{\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})\}\} = \mathrm{E}_{\mathcal{L}}\{\mathrm{E}_{\mathrm{Y}|\mathrm{X}=\mathbf{x}}\{(Y - \varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x}))^2\}\}, \tag{4.4}$$

where $\mathcal{L}$ denotes the random training set. By rewriting the above expression with respect to the optimal Bayes model φ_{B}, it can be shown that in general the expected generalization error for the prediction at $\mathrm{X} = \mathbf{x}$ additively decomposes into a bias, a variance and a noise component as follows:

$$\begin{aligned}\mathrm{E}_{\mathcal{L}}\{\mathrm{Err}\,\{\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})\}\} &= (\varphi_{\mathrm{B}}(\mathbf{x}) - \mathrm{E}_{\mathcal{L}}\{\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})\})^2 + \mathrm{E}_{\mathcal{L}}\{(\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x}) - \mathrm{E}_{\mathcal{L}}\{\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})\})^2\} \\ &\quad + \mathrm{Err}(\varphi_{\mathrm{B}}(\mathbf{x})) \\ &= \mathrm{bias}^2(\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})) + \mathrm{var}(\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})) + \mathrm{Err}(\varphi_{\mathrm{B}}(\mathbf{x})) \end{aligned} \tag{4.5}$$

The bias term measures the difference between the average prediction over all possible random training sets and the prediction of the optimal Bayes model. The variance term measures the variability of the predictions generated by $\varphi_{\mathcal{L}_{\text{miss}},\phi}(\mathbf{x})$. Lastly, the third term, $\text{Err}(\varphi_{\text{B}}(\mathbf{x}))$, represents the irreducible error or noise in the data. It is independent of both the prediction method and the training set. This bias-variance decomposition of the expected generalization error was first introduced in Geman et al. (1992).

For classification problems a similar decomposition is more difficult to obtain in general. However, several proposals can be found in the literature for the expected generalization error based on the zero-one loss function that give a similar insight into the nature of misclassification error (see e.g.Dietterich and Kong (1995); Breiman (1996b); Tibshirani (1996); Louppe (2014)). Moreover, *soft voting*, i.e. averaging class probability estimates and then predicting the most likely class, provide an easy framework to study the generalization error of classification methods by just plugging averaged estimates into (4.5). This approach yields nearly identical results as *majority voting* Breiman (1996a).

First, we review the results showing when ensemble learning is advantageous in comparison to single model learning in regression. We then adapt these results to show the theoretical advantage of multiple imputation regression trees over single imputation regression trees, given an incomplete training set $\mathcal{L}_{\text{miss}}$. Finally, we extend our results to discuss the theoretical benefit of MI combined with ensembles of trees with respect to SI with an ensemble, MI with a single tree and SI with a single tree.

Ensemble learning

Louppe (2014) provided theoretical derivations using the bias-variance decomposition to show the superior prediction results of an ensemble of randomized models compared to its single counterpart, given complete training sets. Specifically, let $\mu_{\mathcal{L},\theta}$ denote the expectation of a single randomized predictor $\varphi_{\mathcal{L},\theta}(\mathbf{x})$ (e.g. CART) with randomization parameter θ. θ is considered to be a random variable inducing randomness between the models in an ensemble. Further, let $\sigma^2_{\mathcal{L},\theta}$ denote the variance of such predictor. Now, consider an ensemble of T randomized models (e.g. a forest) $\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T}\varphi_{\mathcal{L},\theta_i}(\mathbf{x})$ with $\theta_1,\cdots,\theta_T$ i.i.d. random variables. Louppe (2014) shows that such an ensemble keeps the same bias as its single model counterpart, but is able to decrease its variability depending on the size of the ensemble T and the correlation $\rho(\mathbf{x})$ between the models in the ensemble. Indeed, we have that

$$\text{E}_{\mathcal{L},\theta_1,\cdots,\theta_T}\{\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x})\} = \mu_{\mathcal{L},\theta},$$

and thus it follows that

$$\text{bias}^2(\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x})) = (\varphi_{\mathrm{B}}(\mathbf{x}) - \mu_{\mathcal{L},\theta})^2. \tag{4.6}$$

Hence, an ensemble of randomized models and its single model counterpart have the same bias.

Therefore, ensemble methods can only reduce prediction error by reducing their variance. For the prediction variance of the ensemble we obtain that (see e.g. Louppe (2014))

$$\text{var}_{\mathcal{L},\theta}\{\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x})\} = \rho(\mathbf{x})\sigma^2_{\mathcal{L},\theta}(\mathbf{x}) + \sigma^2_{\mathcal{L},\theta}(\mathbf{x})\left(\frac{1-\rho(\mathbf{x})}{T}\right) \tag{4.7}$$

with

$$\rho(\mathbf{x}) = \frac{\mathrm{E}_{\mathcal{L},\theta',\theta''}\{\varphi_{\mathcal{L},\theta'}(\mathbf{x})\varphi_{\mathcal{L},\theta''}(\mathbf{x})\} - \mu^2_{\mathcal{L},\theta}(\mathbf{x})}{\sigma^2_{\mathcal{L},\theta}(\mathbf{x})}. \tag{4.8}$$

If we can make the variance of the ensemble, $\text{var}_{\mathcal{L},\theta}\{\psi_{\mathcal{L},\theta_1,\cdots,\theta_T}(\mathbf{x})\}$, smaller than the single model variance $\sigma^2_{\mathcal{L},\theta}(\mathbf{x})$, then the ensemble improves the prediction performance. As the ensemble gets large, i.e. $T \to \infty$, the variance of the ensemble predictor reduces to $\rho(\mathbf{x})\sigma^2_{\mathcal{L},\theta}(\mathbf{x})$. Hence, large ensembles decrease prediction error when building more decorrelated trees (i.e. with a larger randomization effect). Moreover, for $\rho(\mathbf{x}) \to 0$ the prediction variance reduces to $\frac{\sigma^2_{\mathcal{L},\theta}(\mathbf{x})}{T}$, which again reduces with increasing size T of the ensemble. Notice that when the predictors show no randomization effect at all, i.e. $\rho(\mathbf{x}) \to 1$, then building an ensemble brings no benefit (because all models in the ensemble yield exactly the same prediction in the limit).

Multiple imputation (MI) versus single imputation (SI) for a single tree

The above results can be extended to the case when MI is combined with single tree prediction given a missing training set $\mathcal{L}_{\text{miss}}$. We assume that the imputed datasets are all obtained by the same imputation strategy but each make a different random draw from the predictive distribution, yielding the prediction $\psi_{\mathcal{L}_{\text{miss}},\phi_1,\ldots,\phi_D}(\mathbf{x}) = \frac{1}{D}\sum_{j=1}^{D}\varphi_{\mathcal{L}_{\text{miss}},\phi_j}(\mathbf{x})$. We can decompose the prediction error as in (4.5) and similarly as in Louppe (2014) the expected value and variance of the multiple imputation prediction can be rewritten as:

$$\mathrm{E}_{\mathcal{L},\phi_1,\cdots,\phi_D}\{\psi_{\mathcal{L}_{\text{miss}},\phi_1,\ldots,\phi_D}(\mathbf{x})\} = \mu_{\mathcal{L},\phi},$$

with $\mu_{\mathcal{L},\phi} = \mathrm{E}_{\mathcal{L},\phi}\{\varphi_{\mathcal{L}_{\text{miss}},\phi}\}$. Hence, the bias does not reduce by considering multiple imputations. Therefore, the only source available to reduce prediction error is again the variance of the predictor:

$$\mathrm{var}_{\mathcal{L},\phi_1,\cdots,\phi_D}\{\psi_{\mathcal{L}_{\text{miss}},\phi_1,\ldots,\phi_D}(\mathbf{x})\} = \rho_B(\mathbf{x})\sigma^2_{\mathcal{L},\phi}(\mathbf{x}) + \sigma^2_{\mathcal{L},\phi}(\mathbf{x})\left(\frac{1-\rho_B(\mathbf{x})}{D}\right), \tag{4.9}$$

where $\sigma^2_{\mathcal{L},\phi}$ is the prediction variance of a single tree with single imputation, and $\rho_B(\mathbf{x})$ is the correlation of trees corresponding to different imputations of the same dataset, namely:

$$\rho_B(\mathbf{x}) = \frac{\mathrm{E}_{\mathcal{L},\phi',\phi''}\{\varphi_{\mathcal{L},\phi'}(\mathbf{x})\varphi_{\mathcal{L},\phi''}(\mathbf{x})\} - \mu^2_{\mathcal{L},\phi}(\mathbf{x})}{\sigma^2_{\mathcal{L},\phi}(\mathbf{x})} \tag{4.10}$$

Similar conclusions as before can be obtained now. Multiple imputation improves the performance of single imputation increasingly when the number of imputations D increases and when the correlation $\rho_B(\mathbf{x})$ among prediction models on the different imputed datasets decreases. Note therefore the importance of drawing independent imputations to reduce correlation among the different prediction models.

MI + ensemble methods

Now we discuss when MI combined with an ensemble method yields an improvement in prediction performance. The final prediction in this case can be written as

$$\psi_{\mathcal{L}_{\text{miss}},\Lambda}(\mathbf{x}) = \frac{1}{D}\sum_{d=1}^{D}\frac{1}{T}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\text{miss}}\theta_{t_d},\phi_d}(\mathbf{x}),$$

where Λ denotes a hyperparameter that includes all random parameters θ_{t_d} for growing trees and all random parameters ϕ_d for random imputations.

We consider again the bias-variance decomposition in (4.5). As before, we assume that the imputed datasets are all obtained by the same imputation strategy but make a different random draw from the predictive distribution. Moreover, we assume again that the randomization parameters θ are i.i.d. random variables. For the bias we obtain that

$$\mathrm{E}_{\mathcal{L},\Lambda}\{\psi_{\mathcal{L}_{\text{miss}},\Lambda}(\mathbf{x})\} = \mathrm{E}_{\mathcal{L},\theta,\phi}\{\varphi_{\mathcal{L}_{\text{miss}},\theta,\phi}\} = \mu_{\mathcal{L},\theta,\phi}.$$

Therefore bias remains the same as when a single predictor with single imputation is used. The component that we address to reduce prediction error is therefore again the variance. We now derive the prediction variance for MI with ensembles.

$$
\begin{aligned}
\mathrm{var}_{\mathcal{L},\Lambda}\{\psi_{\mathcal{L}_{\mathrm{miss}},\Lambda}(\mathbf{x})\} &= \mathrm{var}_{\mathcal{L},\Lambda}\{\frac{1}{D}\sum_{d=1}^{D}\frac{1}{T}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\mathrm{miss}},\theta_{t_d},\phi_d}(\mathbf{x})\} \\
&= \frac{1}{D^2}\frac{1}{T^2}\left[\mathrm{E}_{\mathcal{L},\Lambda}\{(\sum_{d=1}^{D}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\mathrm{miss}},\theta_{t_d},\phi_d}(\mathbf{x}))^2\} - \mathrm{E}_{\mathcal{L},\Lambda}\{\sum_{d=1}^{D}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\mathrm{miss}},\theta_{t_d},\phi_d}(\mathbf{x})\}^2\right] \\
&= \frac{1}{D^2}\frac{1}{T^2}\left[\mathrm{E}_{\mathcal{L},\Lambda}\{\sum_{d,e}\sum_{t_d,u_e}\varphi_{\mathcal{L}_{\mathrm{miss}},\theta_{t_d},\phi_d}(\mathbf{x})\varphi_{\mathcal{L}_{\mathrm{miss}},\theta_{u_e},\phi_e}(\mathbf{x})\} - (TD\mu_{\mathcal{L},\theta,\phi}(\mathbf{x}))^2\right] \\
&= \frac{1}{D^2}\frac{1}{T^2}\left[\sum_{d,e}\mathrm{E}_{\mathcal{L},\theta,\phi_d,\phi_e}\{\sum_{t_d,u_e}\varphi_{\mathcal{L}_{\mathrm{miss}},\theta_{t_d},\phi_d}(\mathbf{x})\varphi_{\mathcal{L}_{\mathrm{miss}},\theta_{u_e},\phi_e}(\mathbf{x})\} - T^2D^2\mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})\right] \\
&= \frac{1}{D^2}\frac{1}{T^2}\Bigg[D\Big(T\,\mathrm{E}_{\mathcal{L},\theta,\phi}\{\varphi_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}(\mathbf{x})^2\} + (T^2-T)\,\mathrm{E}_{\mathcal{L},\theta',\theta'',\phi}\{\varphi_{\mathcal{L}_{\mathrm{miss}},\theta',\phi}(\mathbf{x})\varphi_{\mathcal{L}_{\mathrm{miss}},\theta'',\phi}(\mathbf{x})\}\Big) \\
&\quad + (D^2-D)\Big(T^2\,\mathrm{E}_{\mathcal{L},\theta',\theta'',\phi',\phi''}\{\varphi_{\mathcal{L}_{\mathrm{miss}},\theta',\phi'}(\mathbf{x})\varphi_{\mathcal{L}_{\mathrm{miss}},\theta'',\phi''}(\mathbf{x})\}\Big) - T^2D^2\mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})\Bigg] \\
&= \frac{1}{D^2}\frac{1}{T^2}\Bigg[D\Big(T(\sigma^2_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}(\mathbf{x}) + \mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})) + (T^2-T)(\rho_W(\mathbf{x})\sigma^2_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}(\mathbf{x}) + \mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x}))\Big) \\
&\quad + (D^2-D)\Big(T^2(\rho_B(\mathbf{x})\sigma^2_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}(\mathbf{x}) + \mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x}))\Big) - T^2D^2\mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})\Bigg] \\
&= \frac{\rho_W(\mathbf{x})\sigma^2_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}(\mathbf{x})}{D} + \sigma^2_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}(\mathbf{x})\left(\frac{1-\rho_W(\mathbf{x})}{D\cdot T}\right) + \rho_B(\mathbf{x})\sigma^2_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}(\mathbf{x})\left(1-\frac{1}{D}\right)
\end{aligned}
\tag{4.11}
$$

where $\rho_W(\mathbf{x})$ is the correlation of trees fitted on the same imputed dataset. More specifically:

$$
\rho_W(\mathbf{x}) = \frac{\mathrm{E}_{\mathcal{L},\theta',\theta'',\phi}\{\varphi_{\mathcal{L},\theta',\phi}(\mathbf{x})\varphi_{\mathcal{L},\theta'',\phi}(\mathbf{x})\} - \mu^2_{\mathcal{L},\theta,\phi}(\mathbf{x})}{\sigma^2_{\mathcal{L},\theta,\phi}(\mathbf{x})} \tag{4.12}
$$

Note that $\sigma^2_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}$ is the variance of a single tree after single imputation. If we can make the variance in (4.11) smaller than $\sigma^2_{\mathcal{L}_{\mathrm{miss}},\theta,\phi}$, then the prediction error of MI ensembles will be lower than that of SI with single trees. Remark that the first two terms in (4.11) are related to the sampling variability of the predictions while the last term is related to the extra variability in the predictions caused by the missing values. From (4.11) we can also see that if we take the number of imputations D large enough, having a low correlation $\rho_W(\mathbf{x})$ among the trees in each ensemble is not a necessary condition to decrease prediction error. It then suffices to decrease the correlations among imputations $\rho_B(\mathbf{x})$. This is in correspondence with our previous findings for MI + single trees.

The variance of MI with ensembles can be linked to the variance of MI with single trees in (4.9) by rewriting (4.11) as follows.

$$\begin{aligned}\operatorname{var}_{\mathcal{L},\Lambda}\{\psi_{\mathcal{L}_{\text{miss}},\Lambda}(\mathbf{x})\} = \rho_B(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x}) + \frac{\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})}{T}\left(\frac{1-\rho_B(\mathbf{x})T}{D}\right) \\ + \frac{\rho_W(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})}{D} - \frac{\rho_W(\mathbf{x})\sigma^2_{\mathcal{L}_{\text{miss}},\theta,\phi}(\mathbf{x})}{D\cdot T}\end{aligned} \quad (4.13)$$

Comparing the expression in (4.9) to (4.13) reveals that a lower prediction variance for MI with ensembles can be achieved by fitting a large number of decorrelated trees on a large number of decorrelated imputed datasets. While MI with single trees only reduces variability in the predictions due to missing data, MI with ensembles also reduces the sampling variability of the predictions.

A similar comparison can be carried out for SI followed by an ensemble which yields the predictor $\psi_{\mathcal{L}_{\text{miss}},\theta_1,\ldots,\theta_T,\phi}(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T}\varphi_{\mathcal{L}_{\text{miss}},\theta_t,\phi}(\mathbf{x})$. This predictor again has the same bias as the SI + single tree predictor. Moreover, the prediction variance of the SI with ensemble predictor becomes:

$$\operatorname{var}_{\mathcal{L},\theta_1,\cdots,\theta_T,\phi}\{\psi_{\mathcal{L}_{\text{miss}},\theta_1,\ldots,\theta_T,\phi}(\mathbf{x})\} = \rho_W(\mathbf{x})\sigma^2_{\mathcal{L},\theta,\phi}(\mathbf{x}) + \sigma^2_{\mathcal{L},\theta,\phi}(\mathbf{x})\left(\frac{1-\rho_W(\mathbf{x})}{T}\right), \quad (4.14)$$

By comparing (4.14) to (4.11) it is immediately clear that the performance of SI with ensemble can be improved by imputing the training data several times with decorrelated imputations (i.e. with MI ensembles).

Finally, we conclude that MI with ensembles gives superior results to SI with an ensemble, MI with a single tree and SI with a single tree. Surrogate splits can be considered as a special case of single imputation Quinlan (1993), so we can expect that MI with ensembles will also yield better performance than surrogates. Therefore, theoretically MICE + CondRF forms an ideal combination. Indeed, by construction MICE attempts to make independent draws for the imputations while at the same time CondRF attempts to grow decorrelated trees. Using CondRF also helps to improve the whole technique by reducing bias.

4.4 Simulation study

In order to compare the use of surrogates versus imputation, and more in general the predictive performance of the 26 methods considered (see Table 4.1), empirical studies similar to those in Feelders (1999); Rieger et al. (2010); Hapfelmeier et al. (2012) were performed. Next to comparing the empirical performance with the theoretical conclusions, it is of interest to compare our findings with the results in these previous studies, especially with those in the most recent work Hapfelmeier et al. (2012). Hence, all four real-life datasets without missing data selected in Hapfelmeier et al. (2012) have also been used in our studies. They comprise datasets available in R R Development Core Team (2011) and datasets from the UCI Machine Learning Repository Asuncion and Newman (2007). Two of these datasets concern classification and the other two are regression problems. In addition, we also considered a simulated regression dataset where the response follows the data generating model (DGM) of the simulation study in Burgette and Reiter (2010). An overview of the total number of observations and predictors in each dataset can be found in Table 4.2. We now give a short summary of these datasets.

- The *Haberman's Survival Dataset* contains 306 cases from a study conducted on patients who had undergone surgery for breast cancer. It can be obtained from the UCI Machine Learning Repository Asuncion and Newman (2007). We aim to predict the 5-year survival status of a patient based on the three available predictor variables.

- The *Statlog (Heart) Disease Dataset* was collected from 270 patients at four different hospitals. It is provided by the UCI Machine Learning Repository Asuncion and Newman (2007). Our objective is to predict the presence of heart disease based on 13 clinical measurements of the patients.

- The *Swiss Fertility and Socioeconomic Indicators Dataset* was collected at 47 French-speaking provinces of Switzerland around 1888. It is provided by R R Development Core Team (2011) and is used to predict a standardized fertility measure from a set of 5 socio-economic indicators.

- The *Infant Birth Weight Dataset* was gathered from 189 newborns at the Baystate Medical Center, Springfield, Mass, during the year 1986. It is available in the R package MASS and is used to predict the baby's birth weight in grams from 8 risk factors.

- A large regression dataset was generated in order to assess our research questions in a possibly more complex context that might be present in real-life situations. In particular, a dataset with 500 observations and ten continuous predictors was created. Predictors were generated from linear models in a way that complexities such as circular dependence, multicollinearity and interactions may be present. To generate the response variable, the DGM specified in the simulation study of Burgette and Reiter (2010) was used with the same parameter values. The noise to total variability ratio was kept lower than 10% throughout the generation of the variables. Detailed information on the DGM for this dataset can be found in Appendix C.

Table 4.2: Number of observations and predictors listed for each dataset used in this study

Dataset	Obs.	Var.
Survival	306	3
Heart	270	13
Fertility	47	5
Birthweight	189	8
Simulated	500	10

To make our findings comparable to those in Hapfelmeier et al. (2012), missing values were introduced in a similar way as in their paper, although only in the training data. We only considered complete test cases for evaluation purposes, to avoid an extra source of variability in the performance measures. In accordance with the recommendation in Hapfelmeier et al. (2012) to investigate a wider range of patterns we did not only introduce missing values completely at random (MCAR) but also at random (MAR) and not at random (MNAR). We now discuss the missing data mechanisms used in our study in more detail.

Real-life datasets

The following steps were used to introduce missing data in the real datasets according to the different missing data mechanisms and schemes:

1. Randomly split dataset: 80% training set, 20% test set.
2. Fix the fraction of missing data for each variable with missing values as $\eta = 10\%$, 20%, 30%, or 40%.
3. Insert missing data in the training set according to one of the following missing data mechanisms and schemes.
   - *Under MCAR mechanism*:
     First scheme: Randomly induce missing data in ALL (p) variables. In each variable a fraction η of missing values is inserted at random.
     Second scheme: Induce missing data in ONE THIRD of all variables ($p/3$) chosen at random. In each of these variables a fraction η of missing values is inserted at random.

- *Under MAR mechanism*:

  First scheme: Randomly choose one "determining variable" x_{det} to induce missing data in the remaining $p-1$ variables. In each variable a fraction η of missing values is inserted. To this end, the value of the determining variable is transformed into a probability by a logistic function. A missingness indicator is then generated from a Bernoulli distribution with this probability.

  Second scheme: Induce missing data in ONE THIRD of all variables ($p/3$) chosen at random. In each of these variables a fraction η of missing values is inserted. The remaining two thirds of variables now form the "determining variables". The values of these determining variables are transformed into a probability by a logistic function. The missingness indicator is then generated from a Bernoulli distribution with this probability.

- *Under MNAR mechanism*:

  First scheme: Induce missing data in ALL (p) variables. In each variable a fraction η of missing values is inserted based on its upper or lower η quantile (we change this from dataset to dataset), i.e. in every variable a missing status is given to observations that are above (or below) its upper (or lower) η quantile.

  Second scheme: Induce missing data in ONE THIRD of all variables ($p/3$) chosen at random. In each variable a fraction η of missing values is inserted based on its upper or lower η quantile (we change this from dataset to dataset), i.e. in every variable a missing status is given to observations that are above (or below) its upper (or lower) η quantile.

Note that in the first scheme of MCAR and MNAR it can happen that an observation has missing values for all the predictor variables. Such observations were removed from the dataset because they cause problems for several imputation methods.

Simulated dataset

A similar design was used for this dataset. More specifically, these are the steps taken for the introduction of missing data in our simulated data:

1. Randomly split dataset: 80% training set, 20% test set.
2. Fix the fraction of missing data for each variable with missing values as $\eta = 10\%$, 20%, 30%, or 40%.

3. Insert missing data in the training set according to the different mechanisms and schemes.

    - *Under MCAR mechanism*:
      First scheme: Randomly induce missing data in the first 8 variables $(x_1, x_2, \ldots, x_8)$. In each variable a fraction η of missing values is inserted at random.
      Second scheme: Randomly induce missing data in ONE THIRD of all variables $(p/3)$ chosen at random. In each variable a fraction η of missing values is inserted at random.
    - *Under MAR mechanism*:
      First scheme: Use x_9 and x_{10} as potential "determining variables" to induce missing data in $(x_1, x_2, \ldots, x_8)$. In each variable a fraction η of missing values is inserted by randomly selecting one of the following three strategies:
        - insert missing values based on the upper η quantile of one randomly chosen "determining variable" among x_9 and x_{10}, i.e. in every variable a missing status is given to observations that correspond with those of the chosen "determining variable" that are above this upper η quantile.
        - insert missing values as in the previous strategy but now using the lower η quantile.
        - use both x_9 and x_{10} as determining variables and transform their values into a probability by a logistic function. A missingness indicator is then generated from a Bernoulli distribution with this probability.

      Second scheme: Induce missing data in ONE THIRD of all variables $(p/3)$ chosen at random. In each variable a fraction η of missing values is inserted based on the potential "determining variables" x_9 and x_{10} following the same procedure as in the previous scheme.
    - *Under MNAR mechanism*:
      First scheme: Induce missing data in the first 8 variables $(x_1, x_2, \ldots, x_8)$. In each variable a fraction η of missing values is inserted based on its upper η quantile. That is, a missing status is given to observations that are above this upper quantile.
      Second scheme: Induce missing data in ONE THIRD of all variables $(p/3)$ chosen at random. In each variable a fraction η of missing values is inserted based on its upper η quantile. That is, a missing status is given to observations that are above this upper quantile.

General

As in related studies, predictive performance was assessed via the *mean squared prediction error* (MSPE) for regression or its equivalent *misclassification error* (MER) for classification. The procedure to generate datasets with missing values was repeated $1,000$ times for each mechanism and scheme. The mean root MSPE (RMSPE) or the mean MER across these $1,000$ iterations is reported as a final measure of predictive performance. Moreover, a measure for the performance improvement with an imputation strategy compared to surrogate decisions is calculated as in Hapfelmeier et al. (2012):

$$\text{rel.impr.} = \frac{\text{MSPE}_{\text{Sur.}} - \text{MSPE}_{\text{Imp.}}}{\text{MSPE}_{\text{Sur.}}}. \tag{4.15}$$

Hence, we report the mean relative improvement to assess the performance of an imputation method compared to surrogates.

All simulations were implemented in the R statistical software R Development Core Team (2011). To allow a fair comparison with Hapfelmeier et al. (2012), R function settings in their paper were replicated in our study. An overview of all the settings for the methods used in our empirical studies is given in Table 4.3. As mentioned earlier, the R package randomForest Liaw and Wiener (2002) does not support the use of surrogate decisions. Therefore, no comparison between surrogates and imputation could be made for RF.

It has to be emphasized that our comparisons were made among 26 techniques with fixed modeling strategies. Issues like estimation of parameters that yield the best tree structure or setting the best possible imputation model for a given imputation strategy are outside the scope of this study. These settings were specified to allow comparability.

As in Hapfelmeier et al. (2012), the recommendations of Klebanoff and Cole (2008) on the proper publication of imputation methods were followed in work. They are outlined in this Section and described in more detail in the Appendix C. This allows researchers in the field to evaluate the impact of these methods in practice by looking at every result and the particular situation(s) in which they hold.

4.5 Results and Discussion

A summary of mean RMSPE/MER values as the percentage of missing data increases can be found in Figures 4.1-4.3 for all datasets analyzed in this study. To make the plots more informative, we decided to remove all methods with overall low performance.

Table 4.3: R function and its corresponding package name, package reference paper and settings for the implementation of each of the methods included in this study

Technique	R function	R package	Reference	R Settings
CART	rpart()	rpart	Therneau and Atkinson (2011)	maxsurrogate = min(3, variables available)
CondTree	ctree()	party	Hothorn et al. (2011)	maxsurrogate = min(3, variables available)

RF	randomForest()	randomForest	Liaw and Wiener (2002)	ntree = 500,
				mtry = min(5, variables available)

CondRF	cforest()	party	Hothorn et al. (2011)	ntree = 500, mtry = min(5, variables available),
				maxsurrogate = min(3, variables available)

Bagging	bagging()	ipred	Peters et al. (2002)	nbagg= 500,
				maxsurrogate = min(3, variables available)

CondBagging	cforest()	party	Hothorn et al. (2011)	ntree = 500,
				maxsurrogate = min(3, variables available)

Median/mode	na.roughfix()	randomForest	Liaw and Wiener (2002)	none

Prox. matrix	rfImpute()	randomForest	Liaw and Wiener (2002)	ntree = 500,
				mtry = min(5, variables available), iter = 5 [a]

MICE	mice()	mice	VanBuuren and Groothuis-Oudshoorn (2011)	m = 5,
				defaultMethod = c(”norm”,”logreg”,”polyreg”)

MIST	treeMI()	treeMI	Burgette and Reiter (2010)	ITER = 20
*k*NN	kNNImpute() [b]	imputation	Troyanskaya et al. (2001)	k=5

[a] Error messages were displayed frequently when running the rfImpute() routine with iter = 5 on datasets with large amounts of values MAR or MNAR. To obtain imputation in these cases, we ran this routine exceptionally with iter = 1 combined with median/mode imputation when no convergence of the Prox. matrix algorithm was attained at some cells.

[b] Since the kNNImpute() routine only allows numeric data as input, techniques with prior *k*NN imputation could not be included in the comparisons made on datasets with at least one categorical predictor (Heart and Birthweight datasets).

We only show the results for schemes with all variables containing missing values (first schemes). The performance of all methods when a random third of the variables contains missing values is quite stable and resembles that of 10% missingness in all variables. Note that in the plots we have used the same point characters for techniques based on the same tree prediction method. Different line types and colors (gray scale) correspond to the different missing data treatments. In addition to these graphical results, a general summary of mean relative improvement values can be found in Table 4.4 and 4.5. More extensive numerical reports of mean MSPE/MER values and mean relative improvement values can be found in the Appendix C.

4.5.1 Comparison of techniques

The lower lines in Figures 4.1-4.3 mostly correspond to ensemble methods. This confirms the theoretical result in Section 4.3 that the usage of ensemble methods is advisable when the goal is prediction. These methods benefit from their ensemble nature to average out sampling variability. The same result was empirically obtained by Hapfelmeier et al. (2012) and corresponds to what has been broadly shown by several authors, e.g. Bühlmann and Yu (2002); Breiman (1996a). Throughout our simulations CondRF and RF methods as well as CondBagging performed in general superior to single tree methods. Among these ensemble methods, one especially finds that the combinations MICE/MIST + CondRF, MICE/MIST + RF, Prox. matrix + RF and CondBagging alternatively beat each other throughout the datasets and scenarios analyzed.

For small amounts of missing data, the plots in Figures 4.1-4.3 show that CondRF/-CondBagging with surrogates or RF/CondRF with a previous single imputation suffices in general to obtain good prediction results. Therefore, we do not need to make more intensive multiple imputation computations to obtain satisfactory predictions under this scenario. In particular, CondRF with surrogates (dotted lines with triangle point-down symbols) performs as well as other CondRF combinations in three out of the four real-life datasets: Survival, Heart and Birthweight. Similarly, a SI + RF strategy is sufficient to obtain competitive prediction results in the Fertility, Heart and simulated datasets. For instance, Prox. matrix + RF (long-dashed lines with triangle point-up symbols) performs very well for these datasets.

When the amount of missing values is large under the MCAR or MAR patterns MICE/MIST + CondRF/RF methods perform well throughout the datasets analyzed. MICE/MIST + CondRF are shown in Figures 4.1-4.3 with the triangle point-down joined with solid thick lines for MICE as opposed to solid thin lines for MIST. Both methods show competitive performance in comparison to the other techniques in three real-life datasets: Survival, Heart and Birthweight. RF methods (portrayed by the triangle point-up) achieve the first place in the Fertility dataset, with all missing data methods performing equally well, while in the simulated dataset Prox. matrix + RF performs best; in both cases with a clear difference over the other techniques.

When the amount of missingness becomes large under the MNAR pattern, simulations show that MICE/MIST + CondRF again produces satisfactory results in general. In most instances of the real-life datasets they are at least competitive to the other methods. On the other hand, in our studies the performance of RF methods systematically deteriorates relative to the other methods. This is particularly the case for all RF methods in the Survival and Fertility datasets, for RF combined with SI in the Heart and

Birthweight datasets and for MICE + RF in the simulated dataset. In Figure 4.2 one can note that for the Fertility dataset RF methods goes down from being the best techniques under MCAR and MAR mechanisms to become incredibly the worse techniques at large amounts of data MNAR. Likewise, the plots for the Survival, Birthweight and simulated datasets report more robustness in terms of predictive performance for CondRF methods compared to RF methods under this complex missing data scenario. This tendency of RF methods was further observed in other simulation studies whose results are not shown here. Most likely, the difference in performance between CondRF and RF methods resides on the different strategy that is used to select a splitting covariate in each region. The CART procedure in RF might bias the selection of splitting variables while the conditional trees in CondRF aim to prevent this. As a result, CondRF methods can be more successful in extracting valuable information from the (imputed) predictor variables than RF methods, especially in situations of high uncertainty caused by a complex missing data structure.

Interestingly, CondBagging with surrogate decisions (dotted lines with + symbol) also yielded quite competitive results for all datasets and different scenarios analyzed. Moreover, it is also computationally much faster than MICE/MIST + CondRF/RF (see subsection 4.5.4), which is an extra advantage. Bagging, however, always showed worse performance than CondBagging, even for small amounts of missing data.

For SI, it turns out that imputation by the Prox. matrix performs in general comparable to kNN imputation (results shown in the Appendix C). The new method of MIST imputed bootstrap samples + RF (in gray solid line in Figures 4.1-4.3) shows good results in general in comparison to techniques that combine a single tree with surrogates or to techniques that combine RF with single imputation. This is in line with the results in He (2006). However, when compared to CondRF procedures or RF combined with MICE or MIST, it turns out that it has a comparable or slightly worse performance, but never yields a real improvement.

A comparison between the multiple imputation methods MICE (in solid thick lines) and MIST (in solid thin lines) in Figures 4.1-4.3 reveals that they mostly yield quite similar prediction results, with a slight advantage for MICE in the real datasets. Hence, in most cases the extra flexibility by using trees in MIST does not lead to better imputations, due to the higher variability of this procedure. However, the high flexibility of MIST may be useful to capture complicated structures in complex datasets. This is the case for the simulated dataset where MIST yielded better results in comparison to MICE. Doove et al. (2014) showed similar results for MIST with complex simulated datasets involving different types of interactions, considering both categorical and continuous responses.

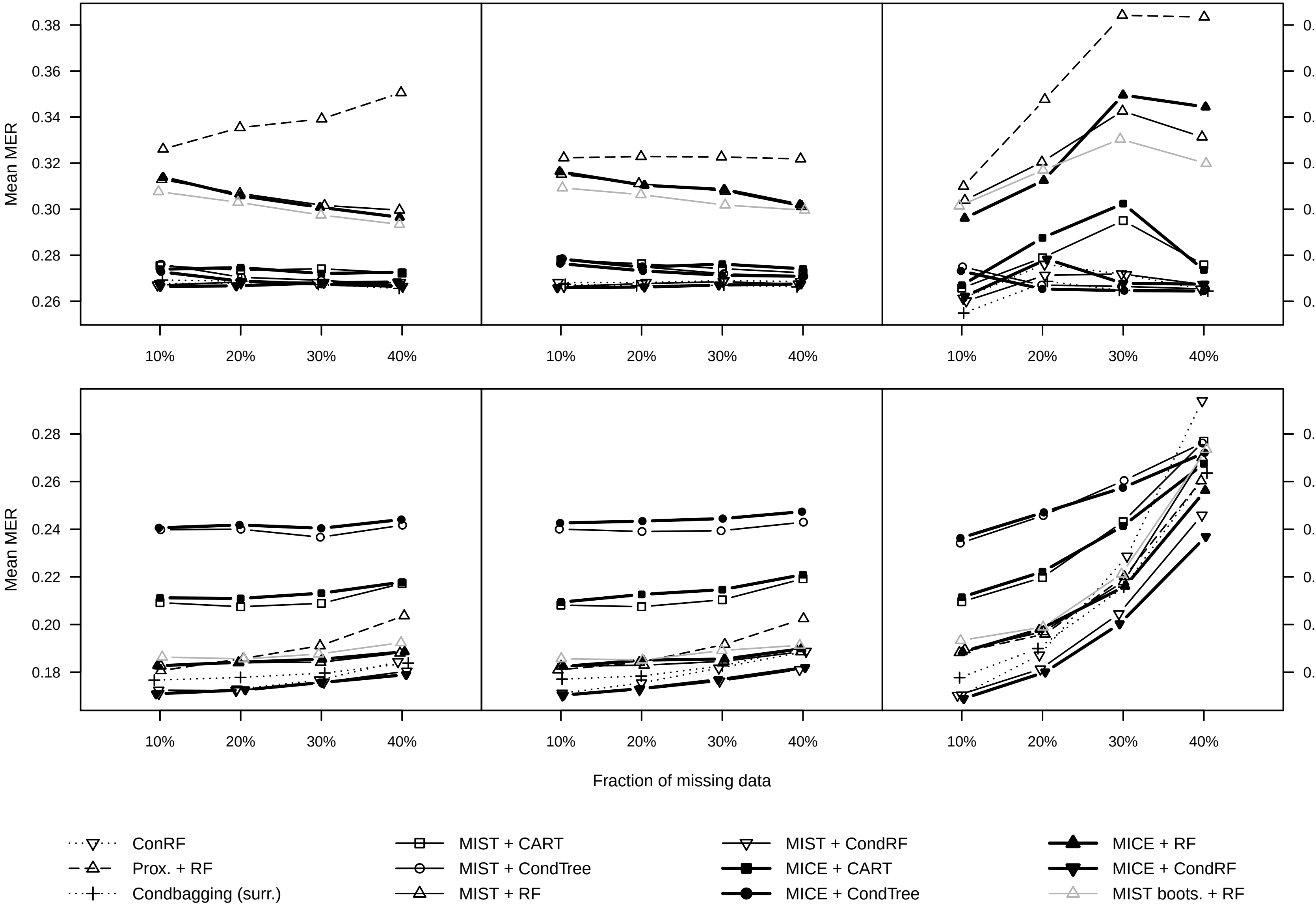


Figure 4.1: Mean MER results for the Survival data (top row) and Heart disease data (bottom row). Results are shown for data MCAR (left panel), MAR (middle panel) and MNAR (right panel).

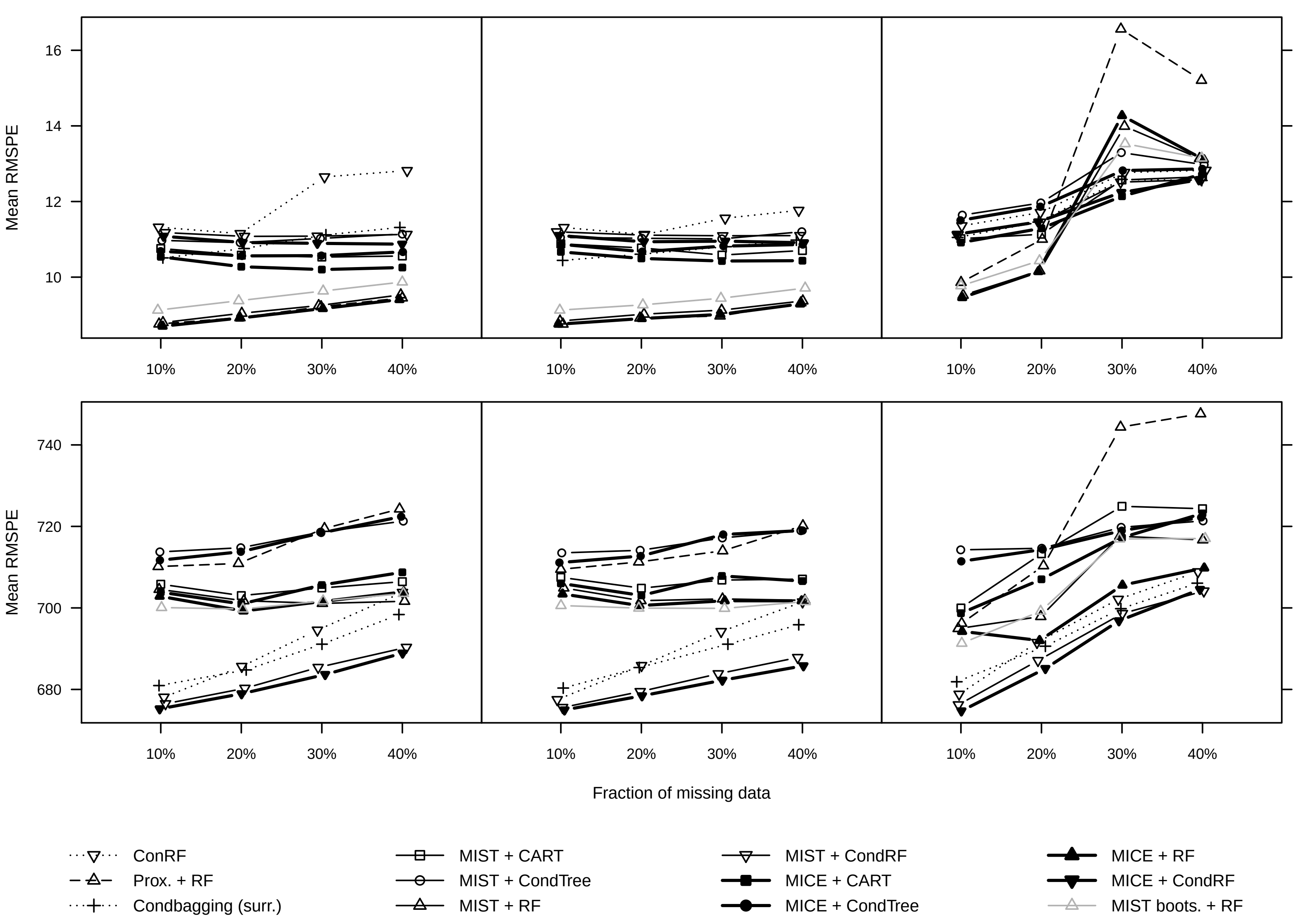


Figure 4.2: Mean RMSPE results for the Fertility data (top row) and Birthweight data (bottom row). The values for the Birthweight dataset have been divided by 10^5. Results are shown for data MCAR (left panel), MAR (middle panel) and MNAR (right panel).

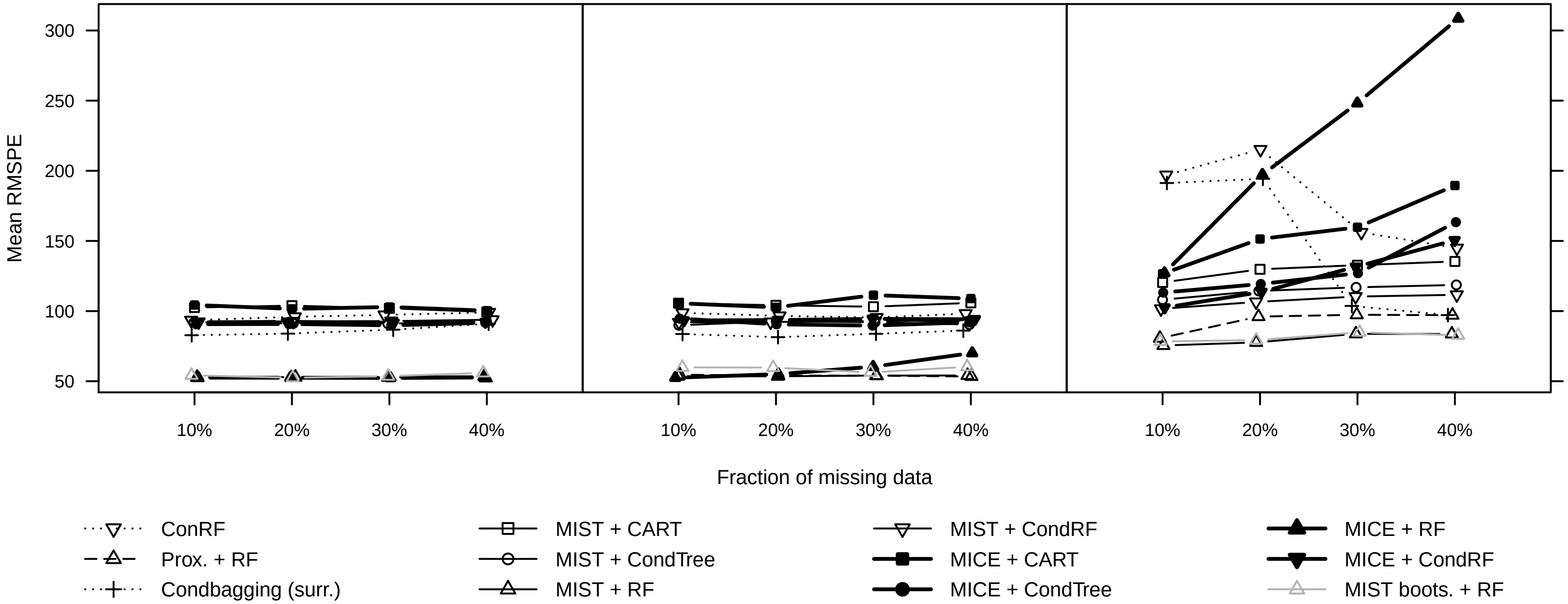


Figure 4.3: Mean RMSPE results for the simulated data. Results are shown for data MCAR (left), MAR (middle) and MNAR (right).

4.5.2 Effects of sample size and dimension

We carried out experiments to investigate the effect of different sample sizes and dimensions on the performance of the techniques under investigation. We used the design of the simulated dataset and the MNAR scenario. This is the most complex scenario and exhibits the largest differences across the different percentages of missing data. Performance patterns for MCAR and MAR mechanisms were quite similar, but with lower error rates than for MNAR values. First, the sample size was extended to 750, 1000 and 2000 observations while the dimension remained fixed at 10. Figure 4.4 shows the results. When the sample size increases, the prediction errors of the methods change very little with a slight tendency to decrease for some methods. The general performance pattern of the methods is almost not changed. Thus, Prox. matrix + RF, MIST + RF and MIST imputed bootstrap samples + RF keep their good performance when the sample size increases.

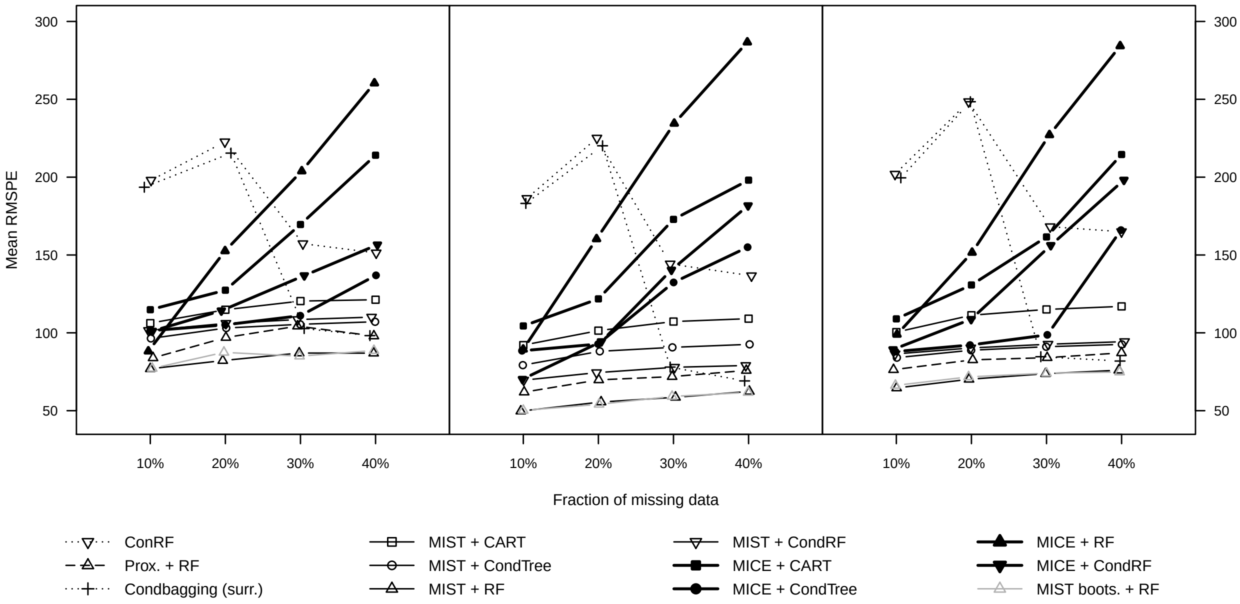


Figure 4.4: Effect of sample size on the prediction performance for simulated datasets under the MNAR mechanism. Results are shown for sample sizes with 750 observations (left panel), 1000 observations (middle panel) and 2000 observations (right panel) with dimension fixed at $p = 10$.

Secondly, we kept the sample size fixed at 500, but increased the dimension to 15, 20 and 50 continuous predictors, by adding noise predictors to our simulated dataset. Missing data were also generated for these noise predictors. These results are shown in Figure 4.5. When the dimension grows, most prediction errors slightly increase when compared to the original simulated dataset. In contrast, CART or CondTree combined with multiple imputation by MICE yield better prediction errors in higher dimensions, which become

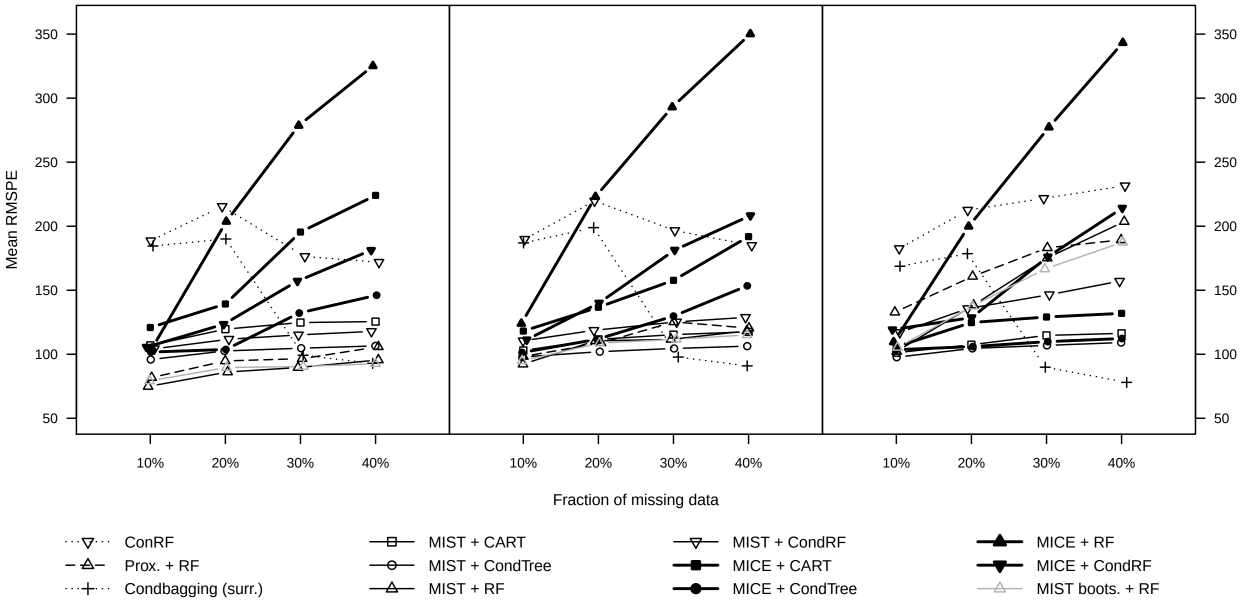


Figure 4.5: Effect of dimension on the prediction performance for simulated datasets under the MNAR mechanism. Results are shown for dimensions with 15 (left panel), 20 (middle panel) and 50 continuous predictors (right panel) with sample size fixed at $N = 500$.

comparable to those of their counterparts using MIST. While MIST performed clearly better than MICE in the original 10 dimensional dataset, the difference between both approaches becomes smaller as the dimension grows. Although this effect seems small for the range of dimension we consider, intuitively this effect could be expected. MICE uses more rigid linear models to make imputations while MIST uses flexible models to make its imputations (i.e. CART models). The linear models in MICE are more biased but less variable than the models in MIST. In higher dimensions with a lot of noise variables, the lower variance of the linear models helps MICE to introduce stability in the imputations and therefore stability in the predictions. Note also that CondBagging and CondRF (both with surrogates) keep their performance stable as the dimension grows (with better performance for CondBagging) in contrast to some methods based on imputation which show a clear increase in their prediction error (e.g. MIST + RF/CondRF). This may imply that the variability reduction by averaging in MI is exceeded by the extra noise introduced in the imputation process, due to the many noise predictors.

We also investigated the combined effect of sample size and dimension by generating datasets of size 1000 in 50 dimensions. The effect on prediction error (results not shown) was less pronounced than for sample size 500. Therefore, CondBagging can be advised for large dimensional datasets, that likely contain noisy variables.

4.5.3 Initial imputation versus surrogates

We first compare SI to the use of surrogate decisions. Table 4.4 shows ranges of mean relative improvement values over all techniques with SI and all missing data fractions. These ranges are specified for the three missing data mechanisms: MCAR, MAR and MNAR. In general there is no clear improvement by SI. As can be seen in Table 4.4, sometimes large improvements occur but they are not regular throughout the analysis. For instance, for the Heart and Survival datasets no SI method yields a clear improvement, except at a few instances with MNAR data (e.g. Prox. matrix + CondRF with 31%). For the simulated dataset, single imputation by Prox. matrix or kNN sometimes yields an improvement for moderate to large fractions of missing data (e.g. kNN + CondRF with 80%), while in the Birthweight dataset only single imputation by Prox. matrix combined with CondRF slightly improves on surrogates (e.g. 4% under MAR). For the Fertility dataset the largest improvement rates are obtained by kNN imputation (22% for MCAR and 5% for MAR). Overall, there is no guarantee that SI will be superior to surrogates in real-life applications and in fact it can turn out to be much worse as can be seen from the large negative lower bounds in Table 4.4.

Table 4.4: Ranges of mean relative improvement for single imputation over all techniques and missing data fractions. Note that the first result line corresponds to the MCAR pattern, the second to the MAR and the third to the MNAR pattern. Only CondRF, CondTree and CART were taken into account for these comparisons.

Fraction of var. miss.	Real-life Datasets				Simulated dataset
	Haberman's Survival	Heart Disease	Swiss Fertility	Birthweight	
1/1	-6% to 1%	-17% to 0%	-11% to 22%	-11% to 3%	-46% to 44%
	-4% to 1%	-11% to -1%	-14% to 5%	-9% to 4%	-14% to 29%
	-20% to 4%	-15% to 31%	-53% to 4%	-12% to 1%	-63% to 80%
1/3	-2% to 0%	-17% to -1%	-15% to 0%	-5% to 1%	-3% to 2%
	-2% to 0%	-17% to -2%	-15% to 1%	-5% to 1%	-2% to 1%
	-3% to 1%	-14% to 0%	-20% to 1%	-5% to 1%	-12% to 1%

Multiple imputation followed by a single tree method (CART or CondTree) in general performs better than surrogates when having a high fraction of data missing on all features under any pattern. However, as in Hapfelmeier et al. (2012), we emphasize that the comparison between MI and surrogates for single trees is not fair. The reason is that MI combined with single trees already has an ensemble nature as seen in Section 4.3. Therefore, it is more honest to look at improvement rates by MI when using an ensemble method instead of single trees.

Table 4.5 contains the mean improvement rates of MI with CondRF with respect to CondRF with surrogates. MI followed by an ensemble method does not yield a distinct benefit over surrogates when the amount of incomplete data is small. This is mostly indicated by the rates on the left extremes of the ranges in Table 4.5. However, it can yield an improvement when the amount of missingness increases. For instance, for the real datasets MI + CondRF often yields much better results than surrogates at large amounts of data missing in all covariates (1/1) and for any pattern. The latter is mostly shown by the rates on the right extremes of the ranges in Table 4.5. In the simulated dataset only MIST + CondRF performs clearly better than surrogates, reaching an improvement rate of 80%, while MICE + CondRF performs clearly worse (see Figure 4.3).

Table 4.5: Ranges of mean relative improvement for multiple imputation with CondRF vs CondRF with surrogates over all missing data fractions. Note that the first result line corresponds to the MCAR pattern, the second to the MAR and the third to the MNAR pattern.

Fraction of var. miss.	Real-life Datasets				Simulated dataset
	Haberman's Survival	Heart Disease	Swiss Fertility	Birthweight	
1/1	-1% to 0%	-2% to 1%	0% to 27%	0% to 4%	5% to 20%
	0% to 1%	0% to 2%	-1% to 11%	1% to 4%	-28% to 7%
	-1% to 1%	-1% to 19%	2% to 8%	1% to 2%	-108% to 80%
1/3	-1% to 0%	-1% to 3%	1% to 3%	0% to 2%	-1% to 1%
	-1% to 0%	-1% to 2%	0% to 4%	0% to 2%	-6% to 1%
	-1% to 0%	-1% to 2%	-1% to 2%	0% to 2%	-52% to 1%

Note that our results for the MCAR pattern differ from those obtained in Hapfelmeier et al. (2012) with real-life datasets originally containing missing values. Authors in Hapfelmeier et al. (2012) concluded that there was no convincing improvement when using MI compared to surrogates. However, there are some differences with our study concerning the modeling of the imputation distributions, as discussed in Section 4.1, which can explain the difference in results.

4.5.4 Computational issues

The good and safe performance of MICE/MIST + CondRF techniques comes at a cost in terms of computation time. CondBagging arises as the best alternative when one is interested in making a tradeoff between performance and computational speed. It showed quite good results throughout the simulation study and it shows a fairly stable computation time even with increasing amounts of missing data. Plots of predictive performance versus average computation time (in CPU seconds) are shown in Figure 4.6 for the Birthweight, Heart, Fertility and simulated datasets under the MNAR mechanism. This is the scenario that shows the largest differences in performance and computational cost. However, similar conclusions can be obtained with other scenarios. The different points in the plots indicate the different percentages of missing data introduced (10%, 20%, 30% and 40%). Note that the computation times are expressed in seconds and were obtained on a single Intel i7 CPU (3.4GHz) machine running Windows 7.

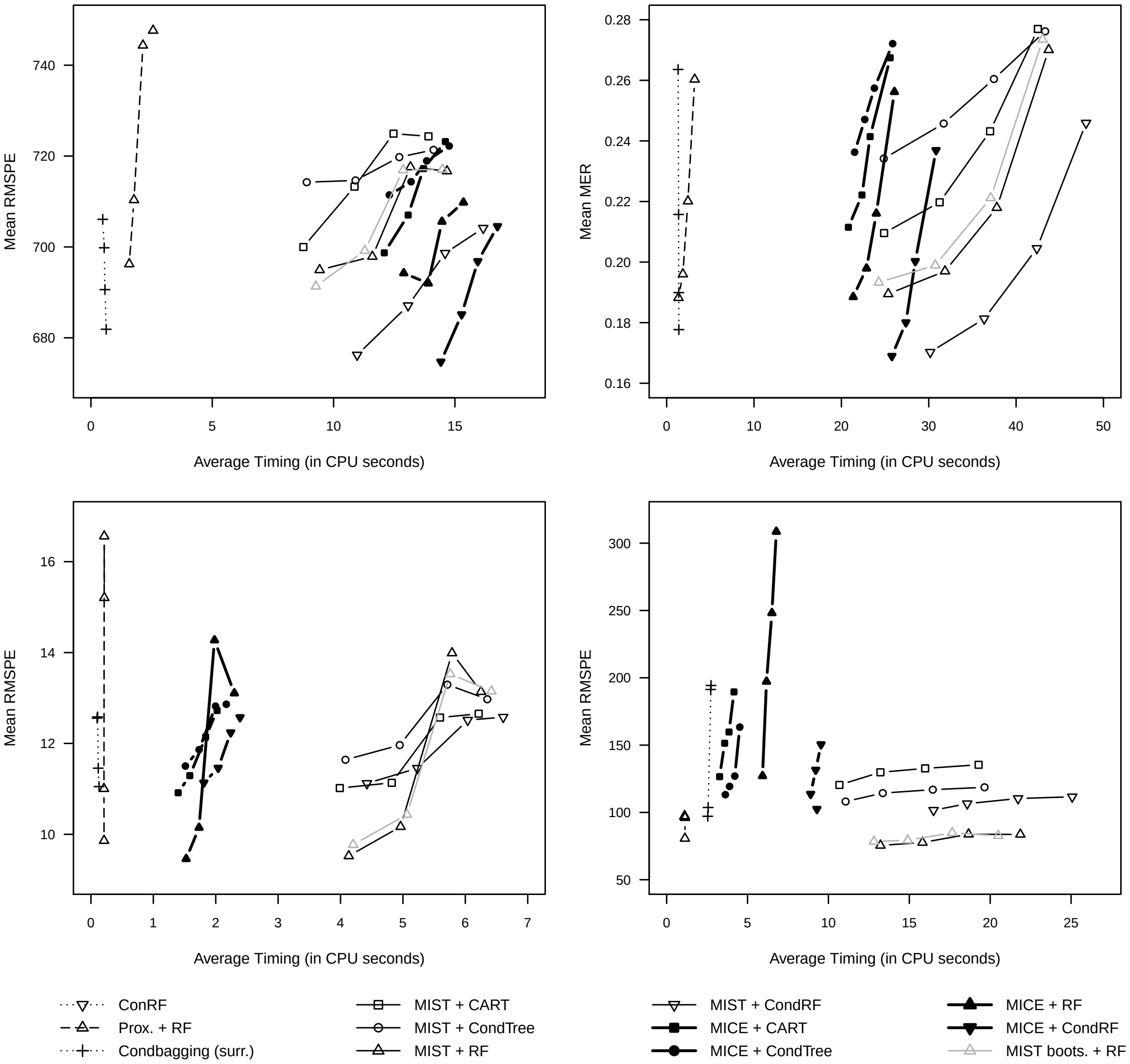


Figure 4.6: Performance vs computation time for the Birthweight (top left), Heart (top right), Fertility (bottom left) and simulated (bottom right) datasets. The mean RMSPE values for the Birthweight dataset have been divided by 10^5.

From the plots, the relatively fast computation of CondBagging is evident (see dotted lines with + symbol). Compared to MIST + CondRF, it runs at least 10 times faster for the Birthweight, 30 times faster for the Heart, 6 times faster for the Fertility and 10 times faster for the simulated dataset, while both methods show similar performance in many cases. The nice trade-off between performance and computation time for CondBagging may become less important when the practitioner has access to multiple processor machines because the MICE/MIST + CondRF procedures can easily be parallelized.

The plots in Figure 4.6 also suggest that multiple imputation by MICE is faster than by MIST. Other datasets and scenarios revealed the same behavior. MICE was also shown to be faster than MIST in Doove et al. (2014). The reason is that MIST uses non-parametric tree models with Bayesian bootstrap which can make it more difficult for the algorithm to converge, compared to the standard MICE that uses linear parametric models. Therefore, MICE + CondRF makes a better trade-off between performance and computational cost than MIST + CondRF. Only for complex datasets with strong nonlinear dependencies MIST + CondRF gives a better performance at the cost of a higher computation time.

Next to CondBagging, single imputation methods and procedures with surrogates have stable computational time across scenarios as well, but they may only work for small amounts of missing data. Only results of Prox. matrix + RF are shown in the plots. In general, methods show the lowest computation times under MAR and the largest times under MNAR mechanism. The latter is especially the case for methods with MI. This is not surprising since, given the complexity of the MNAR mechanism, methods with MI will need more time until they achieve convergence.

We also inspected the time evolution of the different techniques as sample size and/or dimension increases. In particular, we recorded the computation time for the experiments described in subsection 4.5.2. Figure 4.7 shows the time evolution in CPU minutes for datasets with 40% of the values MNAR on all features. When the sample size increases we note from Figure 4.7A that for almost all methods time increases quasi exponentially. Overall, CondBagging consistently has a lower computation time compared to MICE/MIST + CondRF techniques, with the highest differences for MIST + CondRF. MICE is faster than MIST in all scenarios. In general techniques take longer to be computed under data MNAR (plots for MCAR and MAR are not shown here).

When the dimension grows (Figure 4.7B), the computation time of MICE/MIST methods clearly increases faster than for growing sample size, while CondBagging keeps a similar speed in both experiments. MICE is still faster than MIST and methods take longer to be computed under the MNAR mechanism. On datasets of size 1000 in 50 dimensions MIST combined with ensembles required the longest computation time, reaching even around 8 minutes. The increase in computation time as size and/or dimension grows could be expected, especially for the methods with MI. However, computing times of around 8 minutes on a standard machine are still manageable. Hence, MI + CondRF can be computed at a reasonable time even for larger datasets.

4.6 Conclusions and Future work

If in real-life applications the practitioner does not know the mechanism that generated the missing data, as often is the case in practice, we recommend the following strategies

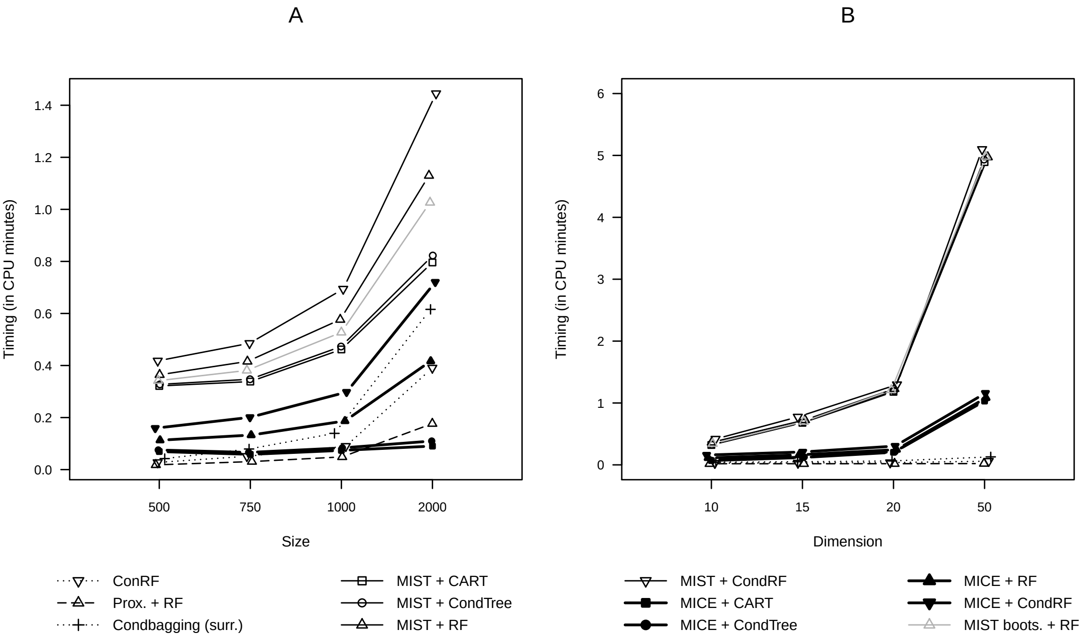


Figure 4.7: A: Computation time vs sample size for simulated data; B: Computation time vs dimension for simulated data

when building a prediction model using tree-based methods:

- If small amounts of missingness are present it suffices to apply any ensemble method with surrogates or with a previous single imputation.
- If the data contains moderate to large amounts of missing values, then multiple imputation by MICE or MIST followed by CondRF is the safest option.
- For high dimensional datasets, CondBagging with surrogate decisions yields a good compromise between performance and computation time.

Multiple imputation is preferred over single imputation because the latter often does not yield any improvement over surrogates. The new method of MIST imputed bootstrap samples when combined with RF is also not able to outperform MI. Multiple imputation ensembles in general showed good results in our comparisons, especially when the amount of missing data was large. These scenarios potentially lead to high prediction variability. Thus, it is crucial that the prediction rule has the ability to average out these sources of variability. Thanks to their ensemble nature in both the imputation step and the prediction model, MI ensembles are able to cancel out both the sampling variability and the variability caused by the missing data. Our theoretical derivations support the empirical findings. However, our studies showed that prediction performance of MI + RF may deteriorate compared to MI + CondRF for large fractions of data MNAR. Most likely the strategy used to select a splitting variable in each region by the individual conditional trees in CondRF allows to better extract valuable information from the imputations than RF in this complex scenario. Therefore, MI + CondRF is a safer and more robust method in terms of prediction performance.

In high dimensions, variability reduction by averaging in MI methods like MICE/MIST + CondRF can be exceeded by the extra noise introduced in the imputation process due to a large number of noisy predictors. In these cases CondBagging emerges as a very good and computationally cheaper alternative.

As with all empirical studies it is not possible to make broad generalizations of our results to other real-life settings. Our conclusions will be applicable to datasets of similar structure (correlations, size, dimension,...) when using the same settings for the tuning parameters of the methods as in our case (see Table 4.3). However, in our opinion these conclusions form a good reference more generally, as the various real-life datasets were selected from different scientific fields with variation in the number of observations and variables. Moreover, the artificial dataset was simulated with a very complex structure for its predictor variables and their relation with the outcome variable. The large comparison of several techniques across many missing data scenarios, which were at times extreme, also enriches the utility and relevance of this study as an element of reference.

In this study, we have combined missing data procedures with tree-based prediction methods in such a way that the whole procedure can first be learned on the training data and then be used to make predictions for individual test cases, when both contain missing values. In our evaluation of the techniques, we only considered complete test cases to avoid an extra source of variability in the performance measures. In principle, all methods considered in this study can handle test cases with missing values, but the currently available implementations in R are not flexible enough yet to obtain the predictions in practice. For example, for the imputation approaches it would be necessary that an incomplete test case can be imputed on the basis of the imputation model from the training data before entering the tree model. Implementations of imputation methods like mice() or rfImpute() currently do not have the feature to “predict” the missing data in a new case based on the imputation fit(s) of the training data. Therefore, current implementations need to be updated and extended with an associated predict function to make them applicable in practice.

Appendix A

Proofs of theorems and additional lemmas

Proof of Theorem 1.1. Due to orthogonal equivariance of the estimator we may assume that $\mathbf{\Sigma} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ with $\lambda_1 > \lambda_2 > \ldots > \lambda_p$, so that $\boldsymbol{\beta}_q = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_q)$, i.e. the canonical basis. This implies that $\lambda_q > \lambda_{q+1}$ and our parameter of interest $\mathcal{L}_q$ is uniquely defined. It now suffices to show that $\mathbf{C}(F_{\mathbf{\Sigma}}, \mathbf{B}_{\text{MVS}}) = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$. Due to orthogonal equivariance, we can assume w.l.o.g. that $\mathbf{C}(F_{\mathbf{\Sigma}}, \mathbf{B}_{\text{MVS}}) = \widetilde{\mathbf{\Lambda}} = \text{diag}(\widetilde{\lambda}_1, \widetilde{\lambda}_2, \ldots, \widetilde{\lambda}_p)$. Using the notation $u(t) = \rho_c'(t)/t$ we have that

$$\widetilde{\mathbf{\Lambda}} = \eta \int u\left(\frac{d_G(\mathbf{x}, \mathbf{B}_{\text{MVS}})}{\sigma_{\text{S}}}\right) \mathbf{x}\mathbf{x}^{\text{T}} g(\lambda_1^{-1}x_1^2 + \lambda_2^{-1}x_2^2 + \ldots + \lambda_p^{-1}x_p^2)\; d\mathbf{x},$$

for some $\eta > 0$. By using the transformation $\mathbf{y} = \widetilde{\mathbf{\Lambda}}^{-1/2}\mathbf{x}$ it is sufficient to show that all solutions of

$$\mathbf{I}_p = \int u\left(\frac{\|\mathbf{r}_{\mathbf{y}}(\mathbf{B}_{\text{MVS}})\|}{\sigma_{\text{S}}}\right) \mathbf{y}\mathbf{y}^{\text{T}} g\left(\frac{\widetilde{\lambda}_1}{\lambda_1}y_1^2 + \frac{\widetilde{\lambda}_2}{\lambda_2}y_2^2 + \ldots + \frac{\widetilde{\lambda}_p}{\lambda_p}y_p^2\right)\; d\mathbf{y}$$

with $\|\mathbf{r}_{\mathbf{y}}(\mathbf{B}_{\text{MVS}})\| = \left\|\widetilde{\mathbf{\Lambda}}^{1/2}\mathbf{y} - \mathbf{B}_{\text{MVS}}\mathbf{B}_{\text{MVS}}^{\text{T}}\widetilde{\mathbf{\Lambda}}^{1/2}\mathbf{y}\right\|$, satisfy $\frac{\widetilde{\lambda}_1}{\lambda_1} = \ldots = \frac{\widetilde{\lambda}_p}{\lambda_p}$. We have that

$$\int u\left(\frac{\|\mathbf{r}_{\mathbf{y}}(\mathbf{B}_{\text{MVS}})\|}{\sigma_{\text{S}}}\right) y_1^2\, g\left(\sum_{j=1}^{p}\frac{\widetilde{\lambda}_j}{\lambda_j}y_j^2\right)\; d\mathbf{y} = \int u\left(\frac{\|\mathbf{r}_{\mathbf{y}}(\mathbf{B}_{\text{MVS}})\|}{\sigma_{\text{S}}}\right) y_2^2\, g\left(\sum_{j=1}^{p}\frac{\widetilde{\lambda}_j}{\lambda_j}y_j^2\right)\; d\mathbf{y} \tag{A.1}$$

and hence

$$\int u\left(\frac{\|\mathbf{r}_{\mathbf{y}}(\mathbf{B}_{\text{MVS}})\|}{\sigma_{\text{S}}}\right)(y_1^2 - y_2^2)\left[g\left(\frac{\widetilde{\lambda}_1}{\lambda_1}y_1^2 + \frac{\widetilde{\lambda}_2}{\lambda_2}y_2^2 + \sum_{j=3}^{p}\frac{\widetilde{\lambda}_j}{\lambda_j}y_j^2\right) - g\left(\frac{\widetilde{\lambda}_2}{\lambda_2}y_1^2 + \frac{\widetilde{\lambda}_1}{\lambda_1}y_2^2 + \sum_{j=3}^{p}\frac{\widetilde{\lambda}_j}{\lambda_j}y_j^2\right)\right]\; d\mathbf{y} = 0 \tag{A.2}$$

as may be seen by interchanging the roles of y_1 and y_2. Note that there is a contribution to the integral above only if in $u(t)$ we have $t \leq c$. Suppose that $\frac{\widetilde{\lambda}_1}{\lambda_1} > \frac{\widetilde{\lambda}_2}{\lambda_2}$. Then if $y_1^2 > y_2^2$ we have that $\frac{\widetilde{\lambda}_1}{\lambda_1} y_1^2 + \frac{\widetilde{\lambda}_2}{\lambda_2} y_2^2 > \frac{\widetilde{\lambda}_2}{\lambda_2} y_1^2 + \frac{\widetilde{\lambda}_1}{\lambda_1} y_2^2$. Likewise, if $y_1^2 < y_2^2$ then $\frac{\widetilde{\lambda}_1}{\lambda_1} y_1^2 + \frac{\widetilde{\lambda}_2}{\lambda_2} y_2^2 < \frac{\widetilde{\lambda}_2}{\lambda_2} y_1^2 + \frac{\widetilde{\lambda}_1}{\lambda_1} y_2^2$. Recall that g is strictly decreasing. Thus, if $\frac{\widetilde{\lambda}_1}{\lambda_1} > \frac{\widetilde{\lambda}_2}{\lambda_2}$ the integral in (A.2) is always non-positive and strictly negative at some y_1, y_2. This contradicts (A.1) showing that $\frac{\widetilde{\lambda}_1}{\lambda_1} = \frac{\widetilde{\lambda}_2}{\lambda_2}$ and in general $\frac{\widetilde{\lambda}_1}{\lambda_1} = \ldots = \frac{\widetilde{\lambda}_p}{\lambda_p}$ □

Proof of Theorem 1.2. Due to orthogonal equivariance, we can restrict ourselves to elliptical distributions $F_{\mathbf{\Sigma}}$ with scatter $\mathbf{\Sigma} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$, $\lambda_1 > \lambda_2 > \ldots > \lambda_p$. From (1.17), we can rewrite

$$\mathbf{C}(F_{\mathbf{\Sigma}}) = \mathbf{C}(F_{\mathbf{\Sigma}}, \mathbf{B}_{\text{MVS}}) = \int u\left(\frac{d_{F_{\mathbf{\Sigma}}}(\mathbf{x}, \mathbf{B}_{\text{MVS}})}{\sigma_S}\right) \mathbf{x}\mathbf{x}^{\mathrm{T}} dF_{\mathbf{\Sigma}}(\mathbf{x}) \tag{A.3}$$

with $u(t) = \rho'(t)/t$. Fisher consistency implies that $\mathbf{C}(F_{\mathbf{\Sigma}}) = \operatorname{diag}(\lambda_1(F_{\mathbf{\Sigma}}), \ldots, \lambda_p(F_{\mathbf{\Sigma}})) = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$. Recall that the columns of $\mathbf{B}_{\text{MVS}}$ are the ordered eigenvectors of $\mathbf{C}(F_{\mathbf{\Sigma}})$. To simplify the notation, we write $F = F_{\mathbf{\Sigma}}$. Let us denote the point mass at a point $\mathbf{x}_0$ by $\Delta_{\mathbf{x}_0}$ and consider the contaminated distribution $F_{\epsilon,\mathbf{x}_0} = (1-\epsilon)F + \epsilon\Delta_{\mathbf{x}_0}$. First, we derive the influence function of the weighted covariance matrix $\mathbf{C}(F_{\mathbf{\Sigma}})$. We have that

$$\begin{aligned}\mathbf{C}_\epsilon = \mathbf{C}(F_{\epsilon,\mathbf{x}_0}) = (1-\epsilon) \int u\left(\frac{d_{F_{\epsilon,\mathbf{x}_0}}(\mathbf{x}, \mathbf{B}_{\text{MVS}}(F_{\epsilon,\mathbf{x}_0}))}{\sigma_S(F_{\epsilon,\mathbf{x}_0})}\right) \mathbf{x}\mathbf{x}^{\mathrm{T}} dF(\mathbf{x}) \\ +\epsilon\; u\left(\frac{d_{F_{\epsilon,\mathbf{x}_0}}(\mathbf{x}_0, \mathbf{B}_{\text{MVS}}(F_{\epsilon,\mathbf{x}_0}))}{\sigma_S(F_{\epsilon,\mathbf{x}_0})}\right) \mathbf{x}_0\mathbf{x}_0^{\mathrm{T}}\end{aligned} \tag{A.4}$$

Using the definition of the influence function we have that

$$IF(\mathbf{x}_0, \mathbf{C}, F) = \left.\frac{\partial \mathbf{C}_\epsilon}{\partial \epsilon}\right|_{\epsilon=0}$$

Differentiating (A.4) gives

$$\begin{aligned}IF(\mathbf{x}_0, \mathbf{C}, F) = \int u'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\text{MVS}})}{\sigma_S}\right) \frac{\partial}{\partial \epsilon} \left.\frac{d_{F_{\epsilon,\mathbf{x}_0}}(\mathbf{x}, \mathbf{B}_{\text{MVS}}(F_{\epsilon,\mathbf{x}_0}))}{\sigma_S(F_{\epsilon,\mathbf{x}_0})}\right|_{\epsilon=0} \mathbf{x}\mathbf{x}^{\mathrm{T}} dF(\mathbf{x}) \\ - \mathbf{C}(F) + u\left(\frac{d_F(\mathbf{x}_0, \mathbf{B}_{\text{MVS}})}{\sigma_S}\right) \mathbf{x}_0\mathbf{x}_0^{\mathrm{T}}\end{aligned} \tag{A.5}$$

It holds that

$$
\begin{aligned}
&\frac{\partial}{\partial \epsilon} \frac{d_{F_{\epsilon,\mathbf{x}_0}}(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}}(F_{\epsilon,\mathbf{x}_0}))}{\sigma_S(F_{\epsilon,\mathbf{x}_0})}\Big|_{\epsilon=0} = \\
&= \frac{\partial}{\partial \epsilon} \frac{\left[\left(\mathbf{x} - \mathbf{B}_{\mathrm{MVS}}(F_{\epsilon,\mathbf{x}_0})\mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}(F_{\epsilon,\mathbf{x}_0})\mathbf{x}\right)^{\mathrm{T}}\left(\mathbf{x} - \mathbf{B}_{\mathrm{MVS}}(F_{\epsilon,\mathbf{x}_0})\mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}(F_{\epsilon,\mathbf{x}_0})\mathbf{x}\right)\right]^{1/2}}{\sigma_S(F_{\epsilon,\mathbf{x}_0})}\Big|_{\epsilon=0} \\
&= \frac{1}{2\, d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \sigma_S} \cdot \\
&\cdot \Big[\Big(- IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)\mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\mathbf{x} - \mathbf{B}_{\mathrm{MVS}} IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)^{\mathrm{T}}\mathbf{x}\Big)^{\mathrm{T}}\Big(\mathbf{x} - \mathbf{B}_{\mathrm{MVS}}\mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\mathbf{x}\Big) \\
&+ \Big(\mathbf{x} - \mathbf{B}_{\mathrm{MVS}}\mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\mathbf{x}\Big)^{\mathrm{T}}\Big(- IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)\mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\mathbf{x} - \mathbf{B}_{\mathrm{MVS}} IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)^{\mathrm{T}}\mathbf{x}\Big)\Big] \\
&- IF(\mathbf{x}_0, \sigma_S, F) \cdot d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}}) \cdot \sigma_S^{-2} \\
&= \frac{\mathbf{x}^{\mathrm{T}}\, \mathbf{B}_{\mathrm{MVS}}\, \mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\, IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)\, \mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\, \mathbf{x}}{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \sigma_S} - \frac{\mathbf{x}^{\mathrm{T}}\, IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)\, \mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\, \mathbf{x}}{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \sigma_S} \\
&- IF(\mathbf{x}_0, \sigma_S, F) \cdot d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}}) \cdot \sigma_S^{-2} \qquad \text{(A.6)}
\end{aligned}
$$

Inserting (A.6) in (A.5) we obtain:

$$
\begin{aligned}
IF(\mathbf{x}_0, \mathbf{C}, F) = &\int u'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right) \frac{\mathbf{x}^{\mathrm{T}}\, \mathbf{B}_{\mathrm{MVS}}\, \mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\, IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)\, \mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\, \mathbf{x}\, \mathbf{x}\, \mathbf{x}^{\mathrm{T}}}{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \sigma_S}\, dF(\mathbf{x}) \\
&- \int u'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right) \frac{\mathbf{x}^{\mathrm{T}}\, IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)\, \mathbf{B}_{\mathrm{MVS}}^{\mathrm{T}}\, \mathbf{x}\, \mathbf{x}\, \mathbf{x}^{\mathrm{T}}}{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \sigma_S}\, dF(\mathbf{x}) \\
&- \int u'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right) d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \mathbf{x}\, \mathbf{x}^{\mathrm{T}}\, dF(\mathbf{x})\, IF(\mathbf{x}_0, \sigma_S, F) \cdot \sigma_S^{-2} \\
&- \mathbf{C}(F) + u\left(\frac{d_F(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right) \mathbf{x}_0 \mathbf{x}_0^{\mathrm{T}} \qquad \text{(A.7)}
\end{aligned}
$$

Using a derivation as in Van Aelst et al. (2013), we obtain that

$$
IF(\mathbf{x}_0, \sigma_S, F) = \frac{\sigma_{\mathrm{S}}^2 \left[\rho\left(\frac{d_F(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) - b\right]}{2b - 2b\sigma_{\mathrm{S}} + \sigma_{\mathrm{S}}^2\, \mathrm{E}_F\left[\rho'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\right]}
$$

Since $\mathbf{B}_{\mathrm{MVS}} = (\mathbf{e}_1, \ldots, \mathbf{e}_q)$, i.e. the canonical basis, we can rewrite (A.7) as:

$$
\begin{aligned}
IF(\mathbf{x}_0, \mathbf{C}, F) = &\sum_{k,l=1}^{q} \int u'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right) \cdot \frac{1}{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \sigma_S}\, x_k\, IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)_{kl}\, x_l\, \mathbf{x}\, \mathbf{x}^{\mathrm{T}}\, dF(\mathbf{x}) \\
&- \sum_{l=1}^{q} \sum_{k=1}^{p} \int u'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right) \frac{1}{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \sigma_S}\, x_k\, IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)_{kl}\, x_l\, \mathbf{x}\, \mathbf{x}^{\mathrm{T}}\, dF(\mathbf{x}) \\
&- \int u'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right) d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\, \mathbf{x}\, \mathbf{x}^{\mathrm{T}}\, dF(\mathbf{x}) \cdot \frac{\sigma_{\mathrm{S}}^2 \left[\rho\left(\frac{d_F(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) - b\right] \cdot \sigma_S^{-2}}{2b - 2b\sigma_{\mathrm{S}} + \sigma_{\mathrm{S}}^2\, \mathrm{E}_F\left[\rho'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right) d_F(\mathbf{x}, \mathbf{B}_{\mathrm{MVS}})\right]} \\
&- \mathbf{C}(F) + u\left(\frac{d_F(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right) \mathbf{x}_0 \mathbf{x}_0^{\mathrm{T}}
\end{aligned}
$$

Simplfying the first and second terms we obtain

$$\begin{aligned}
IF(\mathbf{x}_0, \mathbf{C}, F) = &-\sum_{l=1}^{q}\sum_{k=q+1}^{p}\int u'\left(\frac{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right)\cdot\frac{1}{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})\,\sigma_S}\;x_k\,IF(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}},F)_{kl}\,x_l\,\mathbf{x}\,\mathbf{x}^{\mathrm{T}}\,dF(\mathbf{x})\\
&-\int u'\left(\frac{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right)d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})\,\mathbf{x}\,\mathbf{x}^{\mathrm{T}}\,dF(\mathbf{x})\cdot\frac{\sigma_{\mathrm{S}}^2\left[\rho\left(\frac{d_F(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right)-b\right]\cdot\sigma_S^{-2}}{2b-2b\sigma_{\mathrm{S}}+\sigma_{\mathrm{S}}^2\,\mathrm{E}_F\left[\rho'\left(\frac{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right)d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})\right]}\\
&-\mathbf{C}(F)+u\left(\frac{d_F(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right)\mathbf{x}_0\mathbf{x}_0^{\mathrm{T}}
\end{aligned}$$

We now consider an (i, j)th element of $IF(\mathbf{x}_0, \mathbf{C}, F)$. By symmetry of the integration domain, non-zero contributions in the first integral come from $i = k$, $j = l$, or, $i = l$, $j = k$. Therefore, for any (i, j)th element with $i = 1, \ldots, q$, $j = 1, \ldots, q$, or with $i = q + 1, \ldots, p$, $j = q + 1, \ldots, p$, there is no contribution in the first integral. In the second term the integrand is an odd function if $i \neq j$ and the contribution is zero in that case. Since $\mathbf{C}(F)$ is a diagonal matrix, for any (i, j)th element with $i = q+1, \ldots, p$, $j = 1, \ldots, q$, we have that

$$\begin{aligned}
IF(\mathbf{x}_0, \mathbf{C}, F)_{ij} = &-IF(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}},F)_{ij}\int u'\left(\frac{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right)\cdot\frac{1}{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})\,\sigma_S}\;x_i^2x_j^2\;dF(\mathbf{x})\\
&+u\left(\frac{d_F(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right)x_{0i}x_{0j} \qquad \text{(A.8)}
\end{aligned}$$

Similarly, for any (i, j)th element with $i = 1, \ldots, q$, $j = q + 1, \ldots, p$, we get

$$\begin{aligned}
IF(\mathbf{x}_0, \mathbf{C}, F)_{ij} = &-IF(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}},F)_{ji}\int u'\left(\frac{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right)\cdot\frac{1}{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})\,\sigma_S}\;x_i^2x_j^2\;dF(\mathbf{x})\\
&+u\left(\frac{d_F(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right)x_{0i}x_{0j} \qquad \text{(A.9)}
\end{aligned}$$

For any (i, j)th element with $i = 1, \ldots, q$, $j = 1, \ldots, q$, or, $i = q+1, \ldots, p$, $j = q+1, \ldots, p$, with $i \neq j$ we get

$$IF(\mathbf{x}_0, \mathbf{C}, F)_{ij} = u\left(\frac{d_F(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}})}{\sigma_S}\right)x_{0i}x_{0j}$$

And when $i = j$ we get

$$\begin{aligned}
IF(\mathbf{x}_0, \mathbf{C}, F_{\mathbf{\Sigma}})_{ii} = u\left(\frac{d_F(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right)x_{0i}^2-\lambda_i(F)-\int u'\left(\frac{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right)d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})\,x_i^2\,dF(\mathbf{x})\cdot\\
\cdot\frac{\sigma_{\mathrm{S}}^2\left[\rho\left(\frac{d_F(\mathbf{x}_0,\mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right)-b\right]\cdot\sigma_S^{-2}}{2b-2b\sigma_{\mathrm{S}}+\sigma_{\mathrm{S}}^2\,\mathrm{E}_F\left[\rho'\left(\frac{d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})}{\sigma_{\mathrm{S}}}\right)d_F(\mathbf{x},\mathbf{B}_{\mathrm{MVS}})\right]}
\end{aligned}$$

Using Lemma 3 of Croux and Haesbroeck (2000) and the diagonality of $\mathbf{C}(F)$ it holds that the diagonal elements $IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)_{ii}$ are zero, and that the non-diagonal elements are given by

$$IF(\mathbf{x}_0, \mathbf{B}_{\mathrm{MVS}}, F)_{ij} = \frac{IF(\mathbf{x}_0, \mathbf{C}, F)_{ij}}{\lambda_j(F) - \lambda_i(F)}$$

Using this result in (A.8) and (A.9) and after rearranging terms we obtain

$$IF(\mathbf{x}_0, \mathbf{C}, F_{\mathbf{\Sigma}})_{ij} = \frac{[\lambda_j(F) - \lambda_i(F)] \cdot u\left(\frac{d_F(\mathbf{x}_0, \mathbf{B}_{\text{MVS}})}{\sigma_{\text{S}}}\right) x_{0i}x_{0j}}{\lambda_j(F) - \lambda_i(F) + H_{ij}(\mathbf{B}_{\text{MVS}})}, \quad i = q+1, \dots, p, \; j = 1, \dots, q$$

and

$$IF(\mathbf{x}_0, \mathbf{C}, F_{\mathbf{\Sigma}})_{ij} = \frac{[\lambda_j(F) - \lambda_i(F)] \cdot u\left(\frac{d_F(\mathbf{x}_0, \mathbf{B}_{\text{MVS}})}{\sigma_{\text{S}}}\right) x_{0i}x_{0j}}{\lambda_j(F) - \lambda_i(F) - H_{ij}(\mathbf{B}_{\text{MVS}})}, \quad i = 1, \dots, q, \; j = q+1, \dots, p$$

with $H_{ij}(\mathbf{B}_{\text{MVS}}) = \int u'\left(\frac{d_F(\mathbf{x}, \mathbf{B}_{\text{MVS}})}{\sigma_S}\right) \cdot \frac{1}{d_F(\mathbf{x}, \mathbf{B}_{\text{MVS}})\,\sigma_S}\; x_i^2 x_j^2 \; dF(\mathbf{x})$. □

Lemma A.1. *Let* $\mathbf{x}$ *be a* $p-$*dimensional random vector having any distribution* G *with location* $\boldsymbol{\mu}$ *and scale* $\mathbf{\Sigma} \in \text{SPSD}(p)$*. Assume w.l.o.g. that* $\boldsymbol{\mu} = \mathbf{0}$*. Let* $\boldsymbol{\beta}_q$ *be an orthogonal matrix such that* $\boldsymbol{\beta}_q^{\text{T}} \mathbf{\Sigma} \boldsymbol{\beta}_q = \mathbf{\Lambda}_q = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_q)$*, where* $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_q \geq 0$ *are the* q *largest eigenvalues of* $\mathbf{\Sigma}$*. Assume* $\lambda_q > \lambda_{q+1}$*. For any orthogonal matrix* $\mathbf{B}_q \in \mathbb{R}^{p \times q}$ *it follows that*

$$E_G\left[\|\mathbf{x} - \mathbf{B}_q\mathbf{B}_q^{\text{T}}\mathbf{x}\|^2\right] \geq \sum_{j=q+1}^{p} \lambda_j. \tag{A.10}$$

The unique solution which attains this lower bound is $\boldsymbol{\beta}_q$*.*

Proof of Lemma A.1. For any orthogonal matrix $\mathbf{B}_q \in \mathbb{R}^{p \times q}$ we obtain:

$$\begin{aligned} E_G\left[\|\mathbf{x} - \mathbf{B}_q\mathbf{B}_q^{\text{T}}\mathbf{x}\|^2\right] &= E_G\left[(\mathbf{x} - \mathbf{B}_q\mathbf{B}_q^{\text{T}}\mathbf{x})^{\text{T}}(\mathbf{x} - \mathbf{B}_q\mathbf{B}_q^{\text{T}}\mathbf{x})\right] \\ &= \text{tr}\; E_G\left[(\mathbf{x} - \mathbf{B}_q\mathbf{B}_q^{\text{T}}\mathbf{x})(\mathbf{x} - \mathbf{B}_q\mathbf{B}_q^{\text{T}}\mathbf{x})^{\text{T}}\right] \\ &= \text{tr}\left(E_G\left[\mathbf{x}\mathbf{x}^{\text{T}}\right] - E_G\left[\mathbf{x}\mathbf{x}^{\text{T}}\right]\mathbf{B}_q\mathbf{B}_q^{\text{T}} - \mathbf{B}_q\mathbf{B}_q^{\text{T}}E_G\left[\mathbf{x}\mathbf{x}^{\text{T}}\right]\right. \\ &\quad \left. + \mathbf{B}_q\mathbf{B}_q^{\text{T}}E_G\left[\mathbf{x}\mathbf{x}^{\text{T}}\right]\mathbf{B}_q\mathbf{B}_q^{\text{T}}\right) \\ &= \text{tr}\left(\mathbf{\Sigma} - \mathbf{B}_q^{\text{T}}\mathbf{\Sigma}\mathbf{B}_q\right) = \text{tr}\left(\mathbf{\Sigma}\right) - \text{tr}\left(\mathbf{B}_q^{\text{T}}\mathbf{\Sigma}\mathbf{B}_q\right) \end{aligned}$$

since the trace of a matrix is invariant under orthogonal transformations. We first show the following matrix result. Let $\eta_i\left[\cdot\right]$ represent the $i-$th largest eigenvalue. Then

$$\eta_i\left[\mathbf{\Sigma} - \mathbf{B}_q^{\text{T}}\mathbf{\Sigma}\mathbf{B}_q\right] \geq \begin{cases} \eta_{q+i}\left[\mathbf{\Sigma}\right] = \lambda_{q+i} & (i = 1, 2, \dots, p-q) \\ 0 & (i = p-q+1, \dots, p) \end{cases} \tag{A.11}$$

Using (A.11) we have that

$$\text{tr}\left(\mathbf{\Sigma} - \mathbf{B}_q^{\text{T}}\mathbf{\Sigma}\mathbf{B}_q\right) \geq \sum_{j=q+1}^{p} \lambda_j \tag{A.12}$$

We know that $\operatorname{tr}(\boldsymbol{\beta}_q^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\beta}_q) = \sum_{j=1}^{q}\lambda_j$. Therefore, we conclude that we attain the lower bound in (A.12) when $\mathbf{B}_q = \boldsymbol{\beta}_q$ (see Seber, 1984, Theorem 5.3), i.e.

$$\operatorname{tr}(\boldsymbol{\Sigma} - \boldsymbol{\beta}_q^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\beta}_q) = \sum_{j=q+1}^{p}\lambda_j \tag{A.13}$$

Since $\lambda_q > \lambda_{q+1}$ we have also proved the uniqueness part. □

Proof of Proposition 1. Take $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widehat{H})$ where $\widehat{H} \in \min_{H\in\mathcal{S}} \sum_{i\in H} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H))$. We first prove that $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widehat{H})$ minimizes $\widehat{\sigma}^2_{\mathrm{LTS}}(\mathbf{d}(\mathcal{L}_{\mathbf{B}_q}))$. Take $\mathbf{B}_q \in \mathbb{R}^{p\times q}$ arbitrarily and denote $H^1 = \left\{i \mid d_i^2(\mathcal{L}_{\mathbf{B}_q}) \le d^2_{(h:n)}(\mathcal{L}_{\mathbf{B}_q})\right\} \in \mathcal{S}$ the set of indices corresponding to the first h ordered squared Euclidean distances of the residuals. Then we can write $\sum_{i\in H^1} d_i^2(\mathcal{L}_{\mathbf{B}_q}) = \sum_{i=1}^{h} d^2_{(i:n)}(\mathcal{L}_{\mathbf{B}_q})$. Without loss of generality we assume that $\boldsymbol{\mu}$ is known and equal to $\mathbf{0}$. Using the property of traces and of eigenvalues in (A.11) it follows that

$$\frac{1}{h}\sum_{i\in H^1} d_i^2(\mathcal{L}_{\mathbf{B}_q}) = \frac{1}{h}\sum_{i\in H^1}\|\mathbf{r}_i(\mathcal{L}_{\mathbf{B}_q})\|^2 = \frac{1}{h}\sum_{i\in H^1}\|\mathbf{x}_i - \mathbf{B}_q\mathbf{B}_q^{\mathrm{T}}\mathbf{x}_i\|^2 \ge \sum_{j=q+1}^{p}\widehat{\lambda}_j(H^1) \tag{A.14}$$

where $\widehat{\lambda}_j(H^1)$ is the jth eigenvalue of $\widehat{\boldsymbol{\Sigma}}(H^1)$, the covariance matrix based on the observations $\left\{\mathbf{x}_i;\ i \in H^1\right\}$. Since the data satisfies condition (1.30), Lemma A.1 can be applied:

$$\begin{aligned}\widehat{\sigma}^2_{\mathrm{LTS}}(\mathbf{d}(\mathcal{L}_{\mathbf{B}_q})) = \sum_{i=1}^{h} d^2_{(i:n)}(\mathcal{L}_{\mathbf{B}_q}) = \sum_{i\in H^1} d_i^2(\mathcal{L}_{\mathbf{B}_q}) &\ge \sum_{i\in H^1} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H^1))\\ &\ge \sum_{i\in\widehat{H}} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widehat{H}))\\ &= \sum_{i=1}^{h} d^2_{(i:n)}(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widehat{H}))\\ &= \widehat{\sigma}^2_{\mathrm{LTS}}(\mathbf{d}(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widehat{H})),\end{aligned}$$

where we applied the definition of $\widehat{H}$.
We conclude that $\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widehat{H}) = \min_{\dim(\mathcal{L}_{\mathbf{B}_q})=q}\ \widehat{\sigma}^2_{\mathrm{LTS}}(\mathbf{d}(\mathcal{L}_{\mathbf{B}_q}))$.

On the other hand, take now $\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q} \in \min_{\dim(\mathcal{L}_{\mathbf{B}_q})=q}\ \widehat{\sigma}^2_{\mathrm{LTS}}(\mathbf{d}(\mathcal{L}_{\mathbf{B}_q}))$ with $\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q}$ spanned by the columns of $\widetilde{\mathbf{B}}_q$. Take $\widetilde{\mathbf{B}}_q$ and denote $\widetilde{H} = \left\{i \mid d_i^2(\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q}) \le d^2_{(h:n)}(\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q})\right\} \in \mathcal{S}$ the set of indices corresponding to the first h ordered squared Euclidean distances of the residuals. We can write $\sum_{i\in\widetilde{H}} d_i^2(\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q}) = \sum_{i=1}^{h} d^2_{(i:n)}(\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q})$. Then we have $\sum_{i\in\widetilde{H}} d_i^2(\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q}) \le \sum_{i\in\widetilde{H}} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widetilde{H}))$. But since (A.14) also holds for the pair $(\widetilde{H}, \widetilde{\mathbf{B}}_q)$, the uniqueness part of Lemma A.1 gives $\widetilde{\mathbf{B}}_q = \widehat{\mathbf{B}}_{\mathrm{LS}}(\widetilde{H})$. It then follows that for any other $H \in \mathcal{S}$ we have

$$\begin{aligned}\sum_{i\in H} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H)) \geq \sum_{i=1}^{h} d_{(i:n)}^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H)) &\geq \sum_{i=1}^{h} d_{(i:n)}^2(\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q}) \\ &= \sum_{i\in\widetilde{H}} d_i^2(\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q}) \\ &= \sum_{i\in\widetilde{H}} d_i^2(\widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widetilde{H})).\end{aligned}$$

Hence, it follows that $\widetilde{\mathcal{L}}_{\widetilde{\mathbf{B}}_q} = \widehat{\mathcal{L}}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(\widetilde{H})$, where $\widetilde{H} \in \min_{H\in\mathcal{S}} \; \sum_{i\in H} d_i^2(\mathcal{L}_{\widehat{\mathbf{B}}_{\mathrm{LS}}}(H))$, which ends the proof. □

Proof of Lemma 1.3. Clearly we have that $\mathcal{E} \in \mathcal{D}_G(\alpha)$. From (1.37) we have that

$$\int_{\widehat{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\widehat{E}}, \mathbf{B}_{\mathrm{LS},\widehat{E}}) \; dG(\mathbf{x}) \leq \int_{E} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},E}, \mathbf{B}_{\mathrm{LS},E}) \; dG(\mathbf{x}) \tag{A.15}$$

for all $E \in \mathcal{D}_G(\alpha)$. By definition of $\mathcal{E}$ we also know that

$$\int_{\mathcal{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\widehat{E}}, \mathbf{B}_{\mathrm{LS},\widehat{E}}) \; dG(\mathbf{x}) \leq \int_{\widehat{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\widehat{E}}, \mathbf{B}_{\mathrm{LS},\widehat{E}}) \; dG(\mathbf{x}) \tag{A.16}$$

By lemma A.1 we obtain

$$\int_{\mathcal{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\mathcal{E}}, \mathbf{B}_{\mathrm{LS},\mathcal{E}}) \; dG(\mathbf{x}) \leq \int_{\mathcal{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\widehat{E}}, \mathbf{B}_{\mathrm{LS},\widehat{E}}) \; dG(\mathbf{x}) \tag{A.17}$$

Combining (A.16) and (A.17) it holds that

$$\int_{\mathcal{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\mathcal{E}}, \mathbf{B}_{\mathrm{LS},\mathcal{E}}) \; dG(\mathbf{x}) \leq \int_{\widehat{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\widehat{E}}, \mathbf{B}_{\mathrm{LS},\widehat{E}}) \; dG(\mathbf{x}) \tag{A.18}$$

Finally, combining (A.15) and (A.18) we obtain

$$\int_{\mathcal{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\mathcal{E}}, \mathbf{B}_{\mathrm{LS},\mathcal{E}}) \; dG(\mathbf{x}) = \int_{\widehat{E}} d_G^2(\mathbf{x}, \mathbf{m}_{\mathrm{LS},\widehat{E}}, \mathbf{B}_{\mathrm{LS},\widehat{E}}) \; dG(\mathbf{x}) \tag{A.19}$$

and thus we conclude that $\mathcal{L}_{\mathbf{B}_{\mathrm{LS},\mathcal{E}}} = \mathcal{L}_{\mathbf{B}_{\mathrm{LS},\widehat{E}}}$. □

Proof of Lemma 1.4. The MVLTS solution $\mathbf{B}_{\mathrm{LS},\widehat{E}}$ satisfies $\mathbf{B}_{\mathrm{LS},\widehat{E}}^{\mathrm{T}} \mathbf{\Sigma}_{\widehat{E}} \mathbf{B}_{\mathrm{LS},\widehat{E}} = \mathbf{\Lambda}_{\mathrm{LS},\widehat{E}}$, where $\mathbf{\Lambda}_{\mathrm{LS},\widehat{E}} \in \mathbb{R}^{q\times q}$ is the diagonal matrix that contains the q largest eigenvalues of the covariance functional $\mathbf{\Sigma}_{\widehat{E}}$. Let us now rewrite the MVLTS problem after the orthogonal transformation $\boldsymbol{\Upsilon}\mathbf{x}$, $\boldsymbol{\Upsilon} \in \mathbb{R}^{p \times p}$:

$$\begin{aligned}
& \min_{\mathbf{B}_q,\, \mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q} \frac{1}{1-\alpha} \int_{\widehat{E}} \left\|\boldsymbol{\Upsilon}\mathbf{x} - \mathbf{B}_q\mathbf{B}_q^{\mathrm{T}}\boldsymbol{\Upsilon}\mathbf{x}\right\|^2 dH(\mathbf{x}) \\
= & \min_{\mathbf{B}_q,\, \mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q} \frac{1}{1-\alpha} \int_{\widehat{E}} \|\mathbf{x}\|^2 - \left\|\mathbf{B}_q\mathbf{B}_q^{\mathrm{T}}\boldsymbol{\Upsilon}\mathbf{x}\right\|^2 dH(\mathbf{x}) \\
= & \max_{\mathbf{B}_q,\, \mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q} \frac{1}{1-\alpha} \int_{\widehat{E}} \left\|\mathbf{B}_q\mathbf{B}_q^{\mathrm{T}}\boldsymbol{\Upsilon}\mathbf{x}\right\|^2 dH(\mathbf{x}) \\
= & \max_{\mathbf{B}_q,\, \mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q} \frac{1}{1-\alpha} \int_{\widehat{E}} \mathbf{x}^{\mathrm{T}}\boldsymbol{\Upsilon}^{\mathrm{T}}\mathbf{B}_q\mathbf{B}_q^{\mathrm{T}}\boldsymbol{\Upsilon}\mathbf{x}\ dH(\mathbf{x}) \\
= & \max_{\mathbf{B}_q,\, \mathbf{B}_q^{\mathrm{T}}\mathbf{B}_q = \mathbf{I}_q} \operatorname{tr}(\mathbf{B}_q^{\mathrm{T}}\boldsymbol{\Gamma}_{\widehat{E}}(G)\mathbf{B}_q), \qquad \text{(A.20)}
\end{aligned}$$

by using properties of traces and that $\boldsymbol{\Gamma}_{\widehat{E}}(G) = \frac{1}{1-\alpha}\int_{\widehat{E}} \boldsymbol{\Upsilon}\mathbf{x}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Upsilon}^{\mathrm{T}}\, dG(\mathbf{x}) = \boldsymbol{\Upsilon}\boldsymbol{\Sigma}_{\widehat{E}}(G)\boldsymbol{\Upsilon}^{\mathrm{T}}$. Let us define $\widetilde{\mathbf{B}}_{\widehat{E}}(G) = \boldsymbol{\Upsilon}\mathbf{B}_{\mathrm{LS},\widehat{E}}$. Since $\widetilde{\mathbf{B}}_{\widehat{E}}^{\mathrm{T}}(G)\boldsymbol{\Gamma}_{\widehat{E}}(G)\widetilde{\mathbf{B}}_{\widehat{E}}(G) = \boldsymbol{\Lambda}_{\mathrm{LS},\widehat{E}}$, we have by lemma A.1 that $\widetilde{\mathbf{B}}_{\widehat{E}}(G)$ is the solution to (A.20). Therefore, after transformation we have that

$$\mathbf{B}_{\mathrm{MVLTS}}(\boldsymbol{\Upsilon}\mathbf{x}) = \widetilde{\mathbf{B}}_{\widehat{E}} = \boldsymbol{\Upsilon}\mathbf{B}_{\mathrm{LS},\widehat{E}} = \boldsymbol{\Upsilon}\mathbf{B}_{\mathrm{MVLTS}}(\mathbf{x}),$$

and conclude orthogonal equivariance of the MLTS-PCA estimator. □

We can now proof the Fisher-consistent result of Theorem 1.5.

Proof of Theorem 1.5. Using the result of Lemma 1.4 we may assume that $\boldsymbol{\Sigma} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ with $\lambda_1 > \lambda_2 > \ldots > \lambda_p$, so that $\boldsymbol{\beta}_q = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_q)$, i.e. the canonical basis. This implies that $\lambda_q > \lambda_{q+1}$ so that our parameter of interest $\mathcal{L}_q$ is uniquely defined. It now suffices to show that $\mathbf{B}_{\mathrm{LS},\widehat{E}}(F_{\boldsymbol{\Sigma}}) = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_q)$ and that its columns correspond to the q largest eigenvalues of $\boldsymbol{\Sigma}_{\widehat{E}}(F_{\boldsymbol{\Sigma}})$: $\widetilde{\lambda}_1 > \widetilde{\lambda}_2 > \ldots > \widetilde{\lambda}_q$. Lemma 1.3 shows that $\boldsymbol{\Sigma}_{\widehat{E}}(F_{\boldsymbol{\Sigma}})$ is the covariance matrix based solely on the region $\mathcal{E} = \left\{\mathbf{x} \in \mathbb{R}^p; d_G^2(\mathbf{x}, \mathcal{L}_{\mathbf{B}_{\mathrm{LS},\widehat{E}}}) \le D_\alpha^2\right\}$. Due to orthogonal equivariance we can assume w.l.o.g. that $\boldsymbol{\Sigma}_{\widehat{E}}(F_{\boldsymbol{\Sigma}}) = \widetilde{\boldsymbol{\Lambda}} = \operatorname{diag}(\widetilde{\lambda}_1, \widetilde{\lambda}_2, \ldots, \widetilde{\lambda}_p)$. We have

$$\widetilde{\boldsymbol{\Lambda}} = \eta \int_{\mathcal{E}} \mathbf{x}\mathbf{x}^{\mathrm{T}} g(\lambda_1^{-1}x_1^2 + \lambda_2^{-1}x_2^2 + \ldots + \lambda_p^{-1}x_p^2)\, d\mathbf{x},$$

for some $\eta > 0$. We are left to show that $\widetilde{\lambda}_1 > \widetilde{\lambda}_2 > \ldots > \widetilde{\lambda}_p$. For this, it is sufficient to show that any pair $(\widetilde{\lambda}_j, \widetilde{\lambda}_{j+1})$ satisfies the condition $\widetilde{\lambda}_j > \widetilde{\lambda}_{j+1}$, for $j = 1, \ldots, p-1$. We have

$$\widetilde{\lambda}_2 \int_{\mathcal{E}} x_1^2\, g(\sum_{i=1}^p \lambda_i^{-1}x_i^2)\, d\mathbf{x} \;=\; \widetilde{\lambda}_1 \int_{\mathcal{E}} x_2^2\, g(\sum_{i=1}^p \lambda_i^{-1}x_i^2)\, d\mathbf{x} \qquad \text{(A.21)}$$

and hence

$$\begin{aligned}&\widetilde{\lambda}_1 \left[\int_{\mathcal{E}} x_2^2 \left(g(\ \lambda_1^{-1}x_1^2 + \lambda_2^{-1}x_2^2 + \sum_{i=3}^{p} \lambda_i^{-1}x_i^2\ ) - g(\ \lambda_2^{-1}x_1^2 + \lambda_1^{-1}x_2^2 + \sum_{i=3}^{p} \lambda_i^{-1}x_i^2\ ) \right) d\mathbf{x} \right] \\ &-\widetilde{\lambda}_2 \left[\int_{\mathcal{E}} x_1^2 \left(g(\ \lambda_1^{-1}x_1^2 + \lambda_2^{-1}x_2^2 + \sum_{i=3}^{p} \lambda_i^{-1}x_i^2\ ) - g(\ \lambda_2^{-1}x_1^2 + \lambda_1^{-1}x_2^2 + \sum_{i=3}^{p} \lambda_i^{-1}x_i^2\ ) \right) d\mathbf{x} \right] = 0\end{aligned} \tag{A.22}$$

as may be seen by interchanging the roles of x_1 and x_2. Equation (A.22) can be rewritten as

$$\widetilde{\lambda}_1\ I(\mathbf{x})\ -\ \widetilde{\lambda}_2\ K(\mathbf{x}) = 0$$

with

$$I(\mathbf{x}) = \int_{\mathcal{E}} x_2^2 \left(g(\ \lambda_1^{-1}x_1^2 + \lambda_2^{-1}x_2^2 + \sum_{i=3}^{p} \lambda_i^{-1}x_i^2\ ) - g(\ \lambda_2^{-1}x_1^2 + \lambda_1^{-1}x_2^2 + \sum_{i=3}^{p} \lambda_i^{-1}x_i^2\ ) \right) d\mathbf{x}$$

and

$$K(\mathbf{x}) = \int_{\mathcal{E}} x_1^2 \left(g(\ \lambda_1^{-1}x_1^2 + \lambda_2^{-1}x_2^2 + \sum_{i=3}^{p} \lambda_i^{-1}x_i^2\ ) - g(\ \lambda_2^{-1}x_1^2 + \lambda_1^{-1}x_2^2 + \sum_{i=3}^{p} \lambda_i^{-1}x_i^2\ ) \right) d\mathbf{x}$$

We know that $\lambda_1 > \lambda_2$. Then if $x_1^2 > x_2^2$ we have that $\lambda_1^{-1}x_1^2 + \lambda_2^{-1}x_2^2\ <\ \lambda_2^{-1}x_1^2 + \lambda_1^{-1}x_2^2$ and since g is strictly decreasing this implies $K(\mathbf{x}) > I(\mathbf{x})$. Similarly, if $x_1^2 < x_2^2$ then $\lambda_1^{-1}x_1^2 + \lambda_2^{-1}x_2^2\ >\ \lambda_2^{-1}x_1^2 + \lambda_1^{-1}x_2^2$ implying $K(\mathbf{x}) > I(\mathbf{x})$. Thus, $K(\mathbf{x})$ will always be larger than $I(\mathbf{x})$. This contradicts (A.21) unless $\widetilde{\lambda}_1 > \widetilde{\lambda}_2$ and in general $\widetilde{\lambda}_1 > \widetilde{\lambda}_2 > \ldots > \widetilde{\lambda}_p$. □

Appendix B

Additional tables

Table B.1: Additional results of the simulation study in section 1.6

Design	ϵ	k	LS	S-M (c=3)	S-M (c=1.5)	S-L (α=0.5)	S-L (α=0.25)	PPMD	PPME
a)	0	0	0.02	0.02	0.02	0.07	0.04	0.39	0.23
	10%	1	0.02	0.02	0.03	0.10	0.04	0.43	0.26
		2.5	0.03	0.03	0.03	0.07	0.03	0.49	0.34
		20	2.60	0.02	0.03	0.07	0.03	0.61	0.54
	20%	1.5	0.03	0.03	0.03	1.18	0.08	0.55	0.38
		5	2.57	2.60	0.03	0.07	0.03	1.99	1.96
		20	2.60	2.62	0.03	0.07	0.03	0.77	0.70
b)	0	0	0.03	0.03	0.04	0.14	0.06	0.25	0.16
	10%	1.5	0.05	0.06	0.09	0.24	0.14	0.33	0.28
		2	0.11	0.11	0.10	0.15	0.11	0.38	0.34
		4	0.66	0.64	0.05	0.11	0.06	0.47	0.47
	20%	2	0.48	0.56	0.67	0.75	0.68	0.73	0.66
		3	0.66	0.67	0.71	0.49	0.66	0.79	0.75
		5	0.69	0.69	0.69	0.11	0.17	0.62	0.61
		19	0.70	0.70	0.04	0.11	0.05	0.33	0.31

Table B.2: Additional results of the simulation study in section 1.6

Design	ϵ	k	PPLTS	SPC	MVS	MVS	MVLTS	MVLTS
					(c=3)	(c=1.5)	(α=0.5)	(α=0.25)
a)	0	0	0.32	0.04	0.02	0.02	0.07	0.04
	10%	1	0.41	0.04	0.02	0.03	0.11	0.04
		2.5	0.62	0.05	0.03	0.03	0.07	0.03
		20	0.58	0.05	0.02	0.03	0.07	0.03
	20%	1.5	0.78	0.09	0.03	0.03	0.96	0.07
		5	1.74	0.72	2.60	0.03	0.07	0.03
		20	0.65	0.42	2.62	0.03	0.07	0.03
b)	0	0	0.27	0.05	0.03	0.04	0.14	0.06
	10%	1.5	0.40	0.13	0.06	0.08	0.20	0.12
		2	0.46	0.16	0.10	0.09	0.16	0.09
		4	0.49	0.19	0.55	0.05	0.12	0.06
	20%	2	0.78	0.55	0.56	0.66	0.74	0.67
		3	0.73	0.60	0.67	0.71	0.44	0.67
		5	0.57	0.62	0.69	0.69	0.11	0.17
		19	0.30	0.46	0.72	0.04	0.11	0.05

Table B.3: Additional results of the simulation study in section 1.6

Design	ϵ	k	MVS-	MVS-	MVLTS-	MVLTS-
			det(c=3)	det(c=1.5)	det(α=0.5)	det(α=0.25)
a)	0	0	0.02	0.02	0.06	0.04
	10%	1	0.02	0.03	0.08	0.04
		2.5	0.03	0.03	0.07	0.03
		20	0.02	0.03	0.07	0.03
	20%	1.5	0.03	0.03	0.29	0.06
		5	0.28	0.03	0.06	0.03
		20	0.04	0.03	0.06	0.03
b)	0	0	0.03	0.04	0.11	0.07
	10%	1.5	0.06	0.08	0.14	0.12
		2	0.09	0.07	0.12	0.09
		4	0.24	0.05	0.11	0.07
	20%	2	0.46	0.38	0.35	0.62
		3	0.67	0.27	0.15	0.65
		5	0.69	0.07	0.11	0.66
		19	0.04	0.05	0.11	0.69

	$\epsilon_1 = \epsilon_2 = 0.00$	$\epsilon_1 = 0.10$				$\epsilon_1 = 0.20$				$\epsilon_1 = 0.30$			
Method	Clean	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$
True	1.266	26.930	1.138	269.316	1.264	53.780	1.013	269.685	1.265	80.399	0.888	269.717	1.265
LS	1.246	18.961	5.065	193.372	5.679	37.429	5.682	187.461	7.104	56.593	5.069	189.685	7.214
CoLTS	1.349	26.937	1.204	269.382	1.338	53.792	1.065	269.755	1.329	80.405	0.925	269.757	1.317
MVLTS	1.316	26.935	1.169	269.350	1.298	53.794	1.026	269.749	1.281	80.421	0.889	269.790	1.265
S(3)	1.253	26.922	1.126	269.245	1.252	53.425	1.081	268.453	1.361	75.067	1.685	254.757	2.503
S(1.5)	1.308	26.872	1.270	268.937	1.417	53.241	1.464	267.400	1.850	78.196	1.794	263.041	2.600
PP	1.335	26.536	1.335	265.791	1.486	51.845	1.559	260.972	1.972	73.853	2.206	249.538	3.222

Table B.4: Mean prediction errors over 500 replications for Model 1

	$\epsilon_1 = \epsilon_2 = 0.00$	$\epsilon_1 = 0.10$				$\epsilon_1 = 0.20$				$\epsilon_1 = 0.30$			
Method	Clean	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$
True	1.359	10.063	1.222	100.589	1.358	20.054	1.087	100.598	1.358	29.950	0.953	100.506	1.358
LS	1.339	1.597	4.032	19.528	4.512	1.840	4.482	9.505	5.610	2.517	4.119	8.478	5.868
CoLTS	1.441	10.099	1.294	100.998	1.438	20.191	1.152	101.266	1.438	29.965	1.007	100.674	1.435
MVLTS	1.411	10.040	1.254	100.517	1.393	20.054	1.100	100.546	1.374	29.860	0.962	100.261	1.374
S(3)	1.346	9.839	1.380	99.230	1.541	12.427	2.357	69.919	3.035	4.110	3.861	16.235	5.545
S(1.5)	1.401	9.638	2.047	97.207	2.296	17.916	2.891	90.648	3.645	24.572	3.353	83.262	4.809
PP	1.428	8.922	1.427	90.696	1.589	14.865	1.618	76.535	2.039	15.653	2.026	55.221	2.937

Table B.5: Mean prediction errors over 500 replications for Model 2

	$\epsilon = 0.00$	$\epsilon_1 = 0.10$				$\epsilon_1 = 0.20$				$\epsilon_1 = 0.30$			
Method	Clean	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$
True	0.304	4.411	0.274	44.163	0.304	8.842	0.243	44.088	0.304	13.491	0.211	44.105	0.304
LS	0.285	2.074	0.660	18.457	0.736	5.599	0.711	27.363	0.893	9.550	0.721	30.954	1.045
CoLTS	0.432	4.534	0.389	45.404	0.433	9.062	0.347	45.176	0.434	13.796	0.307	45.106	0.443
MVLTS	0.327	4.434	0.289	44.384	0.321	8.900	0.249	44.387	0.312	13.573	0.209	44.367	0.300
S(3)	0.301	4.412	0.269	44.148	0.299	8.846	0.237	44.113	0.297	13.476	0.205	44.053	0.296
S(1.5)	0.354	4.465	0.318	44.674	0.354	8.931	0.284	44.535	0.355	13.633	0.248	44.574	0.358
PP	0.385	4.439	0.355	44.397	0.394	8.913	0.321	44.430	0.402	13.592	0.290	44.432	0.419

Table B.6: Mean prediction errors over 500 replications for Model 3

	$\epsilon_1 = 0.90$ (D = 1)				$\epsilon_1 = 0.48$				$\epsilon_1 = 0.90$ (fixed T)			
Method	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$	Out	Clean	$\overline{\text{Out}}$	$\overline{\text{Clean}}$
True	39.705	0.031	44.111	0.308	21.345	0.157	44.096	0.304	39.666	0.031	44.066	0.308
LS	32.654	0.415	36.276	4.310	16.496	0.656	33.999	1.280	2.833	0.035	3.147	0.349
CoLTS	41.861	0.122	46.491	1.474	21.701	0.245	44.831	0.478	3.013	0.050	3.347	0.501
MVLTS	37.876	0.585	42.076	6.712	21.192	0.165	43.861	0.329	2.883	0.039	3.202	0.389
S(3)	39.489	0.039	43.874	0.390	21.255	0.153	43.921	0.298	2.850	0.036	3.166	0.364
S(1.5)	40.337	0.051	44.811	0.517	21.508	0.191	44.443	0.370	2.908	0.041	3.230	0.407
PP	40.235	0.042	44.693	0.376	21.446	0.237	44.315	0.461	3.017	0.049	3.351	0.487

Table B.7: Mean prediction errors over 500 replications for Model 3

Appendix C

Tree-based methods

C.1 Tree-based methods

Tree-based methods are known to be flexible enough to capture complex interactive structures and deal comparatively fast with high dimensional settings. Most of them are founded on the notion of the Classification and Regression Tree (CART) algorithm proposed by Breiman et al. (1984). The CART algorithm recursively partitions the predictor space into binary sub-regions so that at each partition the "purest" sub-regions possible are obtained, i.e. the observed values and the fitted values by the tree are as close as possible in each of the sub-regions formed. CART fits a constant in each region, e.g. the average of responses for regression settings or the majority class for classification. Basically, at each region b we seek for the splitting variable j and corresponding point s maximizing the reduction of the impurity measure $\widehat{\Delta Q}_b$:

$$\widehat{\Delta Q}_b = \widehat{Q}_b - \left(\widehat{Q}_{Lb} + \widehat{Q}_{Rb}\right) \tag{C.1}$$

where $\widehat{Q}_{Lb}$ and $\widehat{Q}_{Rb}$ are the impurity measures at the left child and right child region of b respectively. This is equivalent to minimizing the total impurity measure after performing the split (minimizing the rightmost-hand side term of Equation C.1). For classification problems $\widehat{Q}_{Lb}$ and $\widehat{Q}_{Rb}$ in Equation C.1 need to be weighted by N_{Lb} and N_{Rb} respectively, i.e. the number of cases in the corresponding child region of b. Depending on the response type different criteria can be employed for measuring region impurity. The residual sum of squares is often used for continuous response type and the Gini Index for binary response type.

The growth of the tree continues until a stopping criterion is met, e.g. a minimum size for a region. To prevent overfitting issues this large tree is pruned. The optimal tree size is found by means of the cost complexity criterion $C_\alpha(T)$ which is defined as follows:

$$C_\alpha(T) = \sum_{b=1}^{|T|} N_b Q_b(T) + \alpha\,|T| \tag{C.2}$$

where T is any subtree that can be obtained by pruning the initial large tree, $|T|$ is the number of terminal regions in T, N_b the number of observations and $Q_b(T)$ the impurity measure in terminal region b and α the tuning parameter that controls the tradeoff between tree size and its goodness of fit to the data. A finite sequence of subtrees can be formed via *weakest link pruning*: we successively collapse the internal region that produces the smallest increase in the impurity measure until the single-region tree is produced. It can be shown that for each α there is a unique smallest subtree in that sequence that minimizes $C_\alpha(T)$. The optimal value of α and thus the optimal tree size is achieved via cross-validation, commonly by selecting the least complex tree whose estimated error lies below one standard error above the minimum error value.

Despite being an intuitively appealing procedure, CART has some drawbacks: it is a highly unstable procedure due to its hierarchical nature Marshall and Kitsantas (2012); Hastie et al. (2009) and it tends to produce selection bias towards continuous and categorical features with many possible splits and missing values. Aiming to solve the latter problem, Hothorn et al. (2006) proposed the conditional inference tree algorithm (CondTree) which utilizes a unified framework for conditional inference developed by Strasser and Weber (1999). More specifically, recursive binary partitioning is implemented in two steps:

1. Test the global null hypothesis of independence between any of the p features and the response variable by means of permutation tests after multiplicity adjustment (e.g. with Bonferroni procedure). The algorithm is stopped if this hypothesis cannot be rejected. Otherwise, the feature with the strongest association to the response is selected. The association of each of the p features to the response variable is measured by the P-value corresponding to the test for the partial null hypothesis of that single feature and the response.

2. The best split point s for the predictor j selected in step 1 is determined.

The growth of the tree continues until the algorithm is stopped. Hothorn et al. (2006) showed that such statistically motivated stopping criterion ensures that the right sized tree is grown and therefore no form of pruning or cross-validation is needed. P-values for the conditional distribution of test statistics are used to allow an unbiased variable selection since they can be directly compared among covariates measured at different scales.

Despite these advantages, conditional inference tree might still be an unstable procedure due to its hierarchical nature. With the aim of reducing prediction variance, Bagging Breiman (1996a) was proposed. It fits the noisy CART algorithm many times to bootstrap-sampled versions of the data Efron (1979) and average individual tree outcomes on each observation to obtain a final prediction. For classification tasks, “majority voting” over the committee of trees each casting a class “vote” observation-wise is implemented. However, overfitting may arise because trees are fitted on modified versions of the same original sample. This limits the benefits of Bagging. Random Forests (RF), introduced by Breiman (2001), was developed to further improve the prediction variance reduction of Bagging by decreasing the correlation among trees. This is established by adjusting the splitting process during the growing of the tree. Instead of considering all features for each split, only a number $g \leq p$ of predictors selected at random are considered as candidates for a split. The issue of overfitting might be prevented since we don’t provide the algorithm with all the available information, but only with a random part of it. In contrast, more prediction bias might be introduced. A suitable selection of g as well as of the number of bootstrap samples should be done in order to make a proper trade-off between bias and variance.

In the same spirit, Conditional inference Forests (CondRF) and Conditional Bagging (CondBagging) were developed to combine the benefit of unbiased variable selection with reduction of the prediction variance Hothorn et al. (2011). CondRF follows the principle of random feature selection of RF, but fits conditional inference tree instead of CART on the bootstrap samples generated from the training data. CondBagging does not perform random feature selection just like in Bagging (all variables are considered for each split), although it uses conditional inference tree as the base learner too.

All these tree-based methods can in theory handle missing predictor values by using surrogate decisions Breiman et al. (1984). However, the implementation of RF in the R package randomForest Liaw and Wiener (2002) cannot be used on incomplete data. In presence of missingness the best split is chosen by considering only observed cases in every variable. The problem arises either in the training phase or during prediction when the best split variable contains non-observed values at some cells and thus it is not defined to which way (left or right) those cases should be sent down the tree. It is there when alternative or *surrogate splits* play a role. For any observation with a missing value for the primary split variable we can find among all variables with nonmissing value for that case the predictor and corresponding split point producing the best surrogate split, i.e. the split yielding the most similar results as the best split of the training data. Surrogate splits are an attempt to mimic the primary split of a region in terms of the number of cases sent down the same way. The resemblance between surrogate and primary splits is calculated on cases with both the best split and the alternative split variable observed. Once found, such case is sent down the tree according to the best surrogate split rule. If such observation is missing all potential surrogates splits, then the case is simply sent to the child with the largest relative frequency at that region.

C.2 Real-life datasets

- The *Haberman's Survival Dataset* contains 306 cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on patients who had undergone surgery for breast cancer. It can be obtained from the UCI Machine Learning Repository Asuncion and Newman (2007). We aim to predict the 5-year survival status of a patient. Three predictor variables are available for this purpose, namely: age of patient at time of operation, patient's year of operation and number of positive axillary nodes detected.

- The *Statlog (Heart) Disease Dataset* was collected from patients at four clinics: the Cleveland Clinic Foundation, the Hungarian Institute of Cardiology (Budapest), the V.A. Medical Center (Long Beach, CA) and the University Hospital (Zurich, Switzerland). It is provided by the UCI Machine Learning Repository Asuncion and Newman (2007). It contains 270 observations. Our objective is to predict the presence of heart disease based on 13 clinical measurements of the patients: age, gender, chest pain type, resting blood pressure, serum cholestoral in mg/dl, a fasting blood sugar assessment (>120 mg/dl), resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels (0-3) colored by fluoroscopy and thallium scan status information.

- The *Swiss Fertility and Socioeconomic Indicators Dataset* was collected at 47 French-speaking provinces of Switzerland around 1888. It is provided by R R Development Core Team (2011) and is used to predict a standardized fertility measure from a set of 5 socio-economic indicators, namely: males involved in agriculture as occupation, draftees receiving highest mark on army examination, draftees with education beyond primary school, catholic population and the infant mortality within the first year of life.

- The *Infant Birth Weight Dataset* was gathered from 189 newborns at the Baystate Medical Center, Springfield, Mass, during the year 1986. It is available in the R package MASS. It is used to predict the baby's birth weight in grams from 8 risk factors available: the mother's age in years, the mother's weight in pounds at last menstrual period, the mother's race, smoking status during pregnancy, the number of previous premature labours, history of hypertension, presence of uterine irritability and the number of physician visits during the first trimester.

C.3 DGM for the Simulated dataset

$$y_i = 0 + 0.5x_{1,i} + 0.5x_{2,i} + 0.5x_{3,i} + 0.5x_{8,i} + 0.5x_{9,i} + 0.5x_{3,i}^2 + 1x_{1,i}x_{2,i} + 1x_{8,i}x_{9,i} + \varepsilon_{y,i} \tag{C.3}$$

$$\varepsilon_{y,i} \overset{iid}{\sim} N(0,1); i = 1, \ldots, 500$$

$$\begin{aligned}
x_{1,i} &= 0 + 0.1x_{9,i} + 0.1x_{10,i} + 0.08x_{9,i}x_{10,i} + \varepsilon_{x_1,i} \qquad (C.4)\\
x_{2,i} &= 0 + 0.001x_{1,i} + 0.001x_{9,i} + 0.001x_{10,i} + 0.05x_{1,i}x_{9,i} + 0.05x_{9,i}x_{10,i} + 0.05x_{1,i}x_{10,i}\\
&\quad + 0.05x_{1,i}x_{9,i}x_{10,i} + \varepsilon_{x_2,i}\\
x_{3,i} &= 0 + 0.001x_{1,i} + 0.001x_{2,i} + 0.001x_{9,i} + 0.001x_{10,i} + 0.05x_{1,i}x_{2,i} + 0.05x_{1,i}x_{9,i} + 0.05x_{1,i}x_{10,i}\\
&\quad + 0.05x_{2,i}x_{9,i} + 0.05x_{9,i}x_{10,i} + \varepsilon_{x_3,i}\\
x_{4,i} &= 0 + 0.001x_{1,i} + 0.001x_{2,i} + 0.001x_{3,i} + 0.001x_{9,i} + 0.001x_{10,i} + 0.05x_{1,i}x_{2,i} + 0.05x_{1,i}x_{3,i}\\
&\quad + 0.05x_{1,i}x_{9,i} + 0.05x_{1,i}x_{10,i} + 0.05x_{2,i}x_{3,i} + 0.05x_{9,i}x_{10,i} + \varepsilon_{x_4,i}\\
x_{5,i} &= 0 + 0.001x_{1,i} + 0.001x_{2,i} + 0.001x_{3,i} + 0.001x_{4,i} + 0.001x_{9,i} + 0.001x_{10,i} + 0.005x_{1,i}x_{2,i}\\
&\quad + 0.005x_{1,i}x_{3,i} + 0.005x_{1,i}x_{4,i} + 0.005x_{1,i}x_{9,i} + 0.005x_{1,i}x_{10,i} + 0.005x_{3,i}x_{4,i}\\
&\quad + 0.005x_{9,i}x_{10,i} + \varepsilon_{x_5,i}\\
x_{6,i} &= 0 + 0.001x_{1,i} + 0.001x_{2,i} + 0.001x_{3,i} + 0.001x_{4,i} + 0.001x_{5,i} + 0.005x_{1,i}x_{2,i} + 0.005x_{1,i}x_{3,i}\\
&\quad + 0.005x_{1,i}x_{5,i} + 0.005x_{9,i}x_{10,i} + 0.005x_{4,i}x_{9,i} + 0.005x_{4,i}x_{10,i} + 0.005x_{3,i}x_{5,i} + \varepsilon_{x_6,i}\\
x_{7,i} &= 0 + 0.001x_{1,i} + 0.001x_{2,i} + 0.001x_{3,i} + 0.001x_{4,i} + 0.001x_{5,i} + 0.001x_{6,i} + 0.001x_{9,i}\\
&\quad + 0.001x_{10,i} + 0.005x_{1,i}x_{2,i} + 0.005x_{2,i}x_{3,i} + 0.005x_{1,i}x_{6,i} + 0.005x_{1,i}x_{9,i}\\
&\quad + 0.005x_{6,i}x_{9,i} + 0.005x_{9,i}x_{10,i} + \varepsilon_{x_7,i}\\
x_{8,i} &= 0 + 0.001x_{1,i} + 0.001x_{2,i} + 0.001x_{3,i} + 0.001x_{4,i} + 0.001x_{5,i} + 0.001x_{6,i} + 0.001x_{7,i}\\
&\quad + 0.001x_{9,i} + 0.001x_{10,i} + 0.005x_{4,i}x_{7,i} + 0.005x_{1,i}x_{4,i} + 0.005x_{1,i}x_{7,i} + 0.005x_{2,i}x_{5,i}\\
&\quad + 0.005x_{3,i}x_{6,i} + 0.005x_{9,i}x_{10,i} + \varepsilon_{x_8,i}\\
\varepsilon_{x_j,i} &\overset{iid}{\sim} N(0,0.4); i = 1, \ldots, 500; j = 1, \ldots, 8
\end{aligned}$$

Covariates x_9 and x_{10} were drawn from a bivariate normal distribution with means 10 and 7 respectively, variances equal to 1 and a correlation of 0.9.

Table C.1: Summary of mean MSPE/MER values for the real-life datasets. The values for the Birthweight dataset have been divided by 10^4. Results of techniques with prior kNN imputation, Bagging and MIST imputed bootstrap samples + RF are not shown here. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all p variables ($p-1$ for MAR) with missing values (1/1) and only $p/3$ variables with missing values (1/3). Note that the first result line of each technique corresponds to the MCAR pattern, the second to the MAR and the third to the MNAR pattern. N/I stands for "not implemented".

Type	Data	Technique	Missing values										
			0%	10% to 40% Surrogates		10% to 40% Median/mode		10% to 40% Prox. Matrix		10% to 40% MICE		10% to 40% MIST	
				1/1	1/3	1/1	1/3	1/1	1/3	1/1	1/3	1/1	1/3
Classif.	Survival	CondRF	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
				0.27	0.26 - 0.27	0.27	0.26 - 0.27	0.27	0.27	0.27	0.26 - 0.27	0.27	0.26 - 0.27
				0.26 - 0.28	0.26 - 0.27	0.26 - 0.28	0.26 - 0.27	0.26 - 0.30	0.26 - 0.27	0.26 - 0.28	0.27	0.26 - 0.27	0.26 - 0.27
		CondTree	0.28	0.27	0.28	0.27 - 0.28	0.28	0.27 - 0.28	0.28	0.27	0.28	0.27 - 0.28	0.28
				0.27	0.28	0.27 - 0.28	0.28	0.28	0.28	0.27 - 0.28	0.28	0.27 - 0.28	0.28
				0.27 - 0.28	0.28	0.26 - 0.27	0.28	0.27 - 0.28	0.28	0.26 - 0.27	0.28	0.27 - 0.27	0.28
		CART	0.28	0.28	0.28	0.28 - 0.29	0.28	0.28 - 0.30	0.28 - 0.29	0.27	0.28	0.27 - 0.28	0.28
				0.28	0.28	0.28	0.28	0.28 - 0.29	0.28	0.27 - 0.28	0.28	0.27 - 0.28	0.28
				0.27 - 0.32	0.28 - 0.29	0.28 - 0.34	0.28 - 0.29	0.27 - 0.33	0.28 - 0.29	0.27 - 0.30	0.28	0.27 - 0.30	0.28 - 0.29
		RF	0.32			0.32	0.31 - 0.32	0.33 - 0.35	0.32 - 0.33	0.30 - 0.31	0.31 - 0.32	0.30 - 0.31	0.31 - 0.32
				N/I	N/I	0.31 - 0.32	0.31 - 0.32	0.32	0.32	0.30 - 0.32	0.31 - 0.32	0.30 - 0.32	0.31 - 0.32
						0.30 - 0.38	0.31 - 0.33	0.31 - 0.38	0.31 - 0.33	0.30 - 0.35	0.31 - 0.32	0.30 - 0.34	0.31 - 0.32
		CondBagging	0.26	0.27	0.26 - 0.27								
				0.27	0.26 - 0.27	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
				0.25 - 0.27	0.26								
	Heart	CondRF	0.17	0.17 - 0.18	0.17 - 0.18	0.18 - 0.21	0.17 - 0.18	0.17 - 0.19	0.17 - 0.18	0.17 - 0.18	0.17 - 0.18	0.17 - 0.18	0.17 - 0.18
				0.17 - 0.19	0.17 - 0.18	0.18 - 0.20	0.17 - 0.18	0.17 - 0.19	0.17 - 0.18	0.17 - 0.18	0.17 - 0.18	0.17 - 0.18	0.17 - 0.18
				0.17 - 0.29	0.17 - 0.18	0.17 - 0.25	0.17 - 0.19	0.17 - 0.20	0.17 - 0.18	0.17 - 0.24	0.17 - 0.18	0.17 - 0.25	0.17 - 0.18
		CondTree	0.24	0.25 - 0.27	0.24 - 0.25	0.25 - 0.28	0.25	0.25 - 0.27	0.24 - 0.26	0.24	0.24 - 0.25	0.24	0.24 - 0.25
				0.25 - 0.27	0.24 - 0.25	0.25 - 0.27	0.25	0.25 - 0.27	0.25 - 0.27	0.24 - 0.25	0.24 - 0.25	0.24	0.24 - 0.25
				0.24 - 0.31	0.24 - 0.25	0.23 - 0.30	0.24 - 0.26	0.24 - 0.27	0.24 - 0.26	0.24 - 0.27	0.24 - 0.25	0.23 - 0.28	0.24 - 0.25
		CART	0.21	0.22 - 0.28	0.22 - 0.23	0.23 - 0.27	0.22 - 0.23	0.23 - 0.27	0.22 - 0.25	0.21 - 0.22	0.21 - 0.22	0.21 - 0.22	0.21 - 0.22
				0.22 - 0.26	0.22 - 0.23	0.23 - 0.26	0.22 - 0.23	0.22 - 0.27	0.22 - 0.25	0.21 - 0.22	0.21 - 0.22	0.21 - 0.22	0.21 - 0.22
				0.22 - 0.33	0.21 - 0.23	0.22 - 0.32	0.22 - 0.24	0.22 - 0.28	0.21 - 0.25	0.21 - 0.27	0.21 - 0.22	0.21 - 0.28	0.21 - 0.22
		RF	0.18			0.19 - 0.21	0.18 - 0.19	0.18 - 0.20	0.18 - 0.19	0.18 - 0.19	0.18 - 0.19	0.18 - 0.19	0.18 - 0.19
				N/I	N/I	0.19 - 0.21	0.18 - 0.19	0.18 - 0.20	0.18 - 0.19	0.18 - 0.19	0.18 - 0.19	0.18 - 0.19	0.18 - 0.19
						0.19 - 0.31	0.18 - 0.20	0.19 - 0.26	0.18 - 0.19	0.19 - 0.26	0.18 - 0.19	0.19 - 0.27	0.18 - 0.19
		CondBagging	0.18	0.18	0.18 - 0.19								
				0.18 - 0.19	0.18 - 0.19	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
				0.18 - 0.26	0.18 - 0.19								
Regr.	Fertility	CondRF	128	124 - 164	127 - 128	127 - 132	127 - 128	125 - 127	126 - 127	118 - 123	123 - 126	123 - 125	125 - 127
				124 - 138	127 - 128	127 - 129	127 - 128	124 - 127	124 - 127	119 - 123	122 - 126	123 - 125	125 - 127
				129 - 164	127 - 129	132 - 171	128 - 129	131 - 172	128 - 129	124 - 158	126 - 128	124 - 158	126 - 128
		CondTree	130	138 - 146	131 - 133	137 - 147	132 - 135	133 - 135	130 - 131	112 - 114	121 - 124	119 - 124	125 - 126
				135 - 140	130 - 132	132 - 141	128 - 132	131	129 - 131	114 - 118	123 - 125	121 - 125	125 - 127
				144 - 188	130 - 133	146 - 216	132 - 134	151 - 197	129 - 134	132 - 165	129 - 133	135 - 177	129 - 133
		CART	130	125 - 134	124 - 129	126 - 138	123 - 127	127 - 130	128 - 131	104 - 111	113 - 121	111 - 116	118 - 123
				124 - 128	124 - 126	125 - 134	127 - 131	126 - 129	128 - 132	109 - 114	116 - 123	112 - 118	120 - 126
				125 - 185	126 - 128	130 - 228	126 - 132	133 - 236	126 - 148	119 - 162	124 - 128	121 - 160	123 - 128
		RF	74			77 - 91	75 - 79	77 - 89	75 - 80	76 - 88	75 - 78	77 - 91	76 - 80
				N/I	N/I	76 - 87	76 - 79	77 - 87	76 - 82	77 - 86	76 - 81	78 - 88	76 - 79
						95 - 250	82 - 95	97 - 274	83 - 95	89 - 204	79 - 90	91 - 196	80 - 93
		CondBagging	106	110 - 128	107 - 111								
				109 - 121	107 - 111	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
				122 - 158	111 - 113								
	Birthweight	CondRF	45.26	45.98 - 49.53	45.56 - 46.93	46.01 - 48.78	45.53 - 46.51	45.59 - 47.70	45.40 - 46.21	45.59 - 47.46	45.40 - 45.92	45.76 - 47.66	45.43 - 46.09
				45.90 - 49.21	45.58 - 46.92	45.88 - 48.40	45.55 - 46.48	45.50 - 47.33	45.40 - 46.25	45.56 - 47.03	45.45 - 45.99	45.64 - 47.31	45.50 - 46.11
				46.08 - 50.25	45.52 - 47.46	45.66 - 50.32	45.36 - 47.21	45.40 - 50.80	45.28 - 46.85	45.53 - 49.64	45.26 - 46.69	45.73 - 49.58	45.35 - 46.80
		CondTree	51.73	51.61 - 52.77	51.69 - 52.25	52.29 - 53.27	51.87 - 52.49	51.32 - 52.66	51.56 - 52.04	50.65 - 52.19	51.01 - 51.46	50.94 - 52.03	51.24 - 51.45
				51.67 - 52.62	51.69 - 52.33	52.15 - 53.32	52.00 - 52.17	51.29 - 52.37	51.81 - 52.31	50.57 - 51.69	51.19 - 51.61	50.91 - 51.69	51.29 - 51.56
				51.54 - 52.08	51.60 - 52.51	51.71 - 52.36	51.70 - 52.25	51.05 - 52.40	51.42 - 51.91	50.61 - 52.16	50.99 - 51.71	51.02 - 52.04	51.17 - 51.75
		CART	52.32	52.53 - 54.21	52.17 - 52.98	53.33 - 55.63	52.73 - 53.95	53.82 - 58.89	53.13 - 55.13	49.14 - 50.23	50.42 - 50.80	49.42 - 49.91	50.65 - 51.04
				52.58 - 53.55	52.40 - 53.12	53.10 - 55.34	52.59 - 54.00	53.47 - 57.62	53.10 - 55.28	49.43 - 50.11	50.45 - 50.73	49.68 - 50.06	50.71 - 50.92
				51.14 - 57.84	51.82 - 54.45	51.70 - 61.57	52.19 - 56.18	51.81 - 63.50	52.01 - 56.20	48.82 - 52.30	50.29 - 51.54	49.00 - 52.55	50.43 - 51.82
		RF	50.46			50.67 - 53.12	50.63 - 51.41	50.43 - 52.46	50.36 - 50.83	48.89 - 49.55	49.73 - 50.05	49.16 - 49.64	49.82 - 50.17
				N/I	N/I	50.62 - 52.46	50.70 - 51.21	50.34 - 51.87	50.38 - 50.80	49.08 - 49.48	49.57 - 50.08	49.25 - 49.69	49.73 - 50.21
						49.58 - 56.36	50.13 - 52.82	48.47 - 55.90	49.66 - 52.11	47.89 - 50.38	49.46 - 49.99	48.29 - 51.49	49.53 - 50.86
		CondBagging	46.04	46.37 - 48.77	46.14 - 47.04								
				46.29 - 48.42	46.20 - 47.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
				46.50 - 49.85	46.14 - 47.70								

Table C.2: Summary of mean MSPE/MER values for the simulated dataset. Results of techniques with prior kNN imputation, Bagging and MIST imputed bootstrap samples + RF are not shown here. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: 8 variables with missing values and only 8/3 variables with missing values. Note that the first result line of each technique corresponds to the MCAR pattern, the second to the MAR and the third to the MNAR pattern. N/I stands for "not implemented".

Type	Data	Technique	Missing values										
			0%	10% to 40% Surrogates		10% to 40% Median/mode		10% to 40% Prox. Matrix		10% to 40% MICE		10% to 40% MIST	
				8	8/3	8	8/3	8	8/3	8	8/3	8	8/3
Regr.	Simulated	CondRF	77	79 - 93	77	80 - 101	77	76 - 77	76 - 77	77	77	77 - 80	77
				83 - 91	77	78 - 82	77	77 - 78	77	78 - 81	77 - 78	77 - 78	76 - 77
				205 - 467	76 - 77	130 - 177	76 - 77	102 - 128	76 - 77	100 - 232	77 - 84	98 - 118	76 - 77
		CondTree	77	77 - 112	78 - 79	85 - 124	78 - 79	77 - 79	77 - 78	74 - 77	76 - 77	70 - 76	77 - 78
				85 - 93	78 - 79	81 - 86	78 - 79	77 - 81	78 - 80	75 - 77	77 - 78	76 - 79	77 - 78
				125 - 521	79	115 - 130	79	109 - 130	79	111 - 251	76 - 84	100 - 123	79
		CART	99	111 - 164	98 - 101	121 - 164	98 - 100	99	98 - 99	87 - 92	94 - 96	80 - 91	93 - 97
				114 - 131	99 - 101	101 - 114	99 - 101	100 - 102	98 - 101	89 - 98	95 - 97	92 - 96	97 - 99
				152 - 172	100 - 101	193 - 230	100 - 102	157 - 195	101 - 109	142 - 413	91 - 101	125 - 161	98 - 100
		RF	22			23 - 27	22	22	22	22	22	22 - 23	22
				N/I	N/I	23 - 24	22	23 - 24	22 - 24	22 - 39	22 - 23	23 - 24	22
						124 - 129	22 - 23	73 - 88	22 - 24	123 - 940	22 - 28	46 - 64	22 - 23
		CondBagging	60	61 - 73	60								
				61 - 67	60	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
				84 - 384	60								

Table C.3: For each real-life dataset analyzed, we show the average percentage of missing data and the average percentage of complete observations across $1,000$ simulations in which missingness is introduced according to a fixed pattern, fraction and scheme of missing values.

Data	Pattern	% miss./var	# var. miss.	average	average
				% miss. data	% complete obs
Survival	MCAR	10%	3/3	7.65%	72.39%
		20%		15.00%	51.20%
		30%		22.65%	34.00%
		40%		30.00%	21.55%
		10%	1/3	2.55%	89.80%
		20%		5.00%	80.00%
		30%		7.55%	69.80%
		40%		10.00%	60.00%
	MAR	10%	2/3	4.90%	81.45%
		20%		10.00%	64.21%
		30%		15.10%	49.04%
		40%		20.00%	36.43%
		10%	1/3	2.55%	89.80%
		20%		5.00%	80.00%
		30%		7.55%	69.80%
		40%		10.00%	60.00%
	MNAR	10%	3/3	7.65%	72.69%
		20%		15.00%	51.97%
		30%		22.65%	34.30%
		40%		30.00%	20.62%
		10%	1/3	2.55%	89.80%
		20%		5.00%	80.00%
		30%		7.55%	69.80%
		40%		10.00%	60.00%
Heart	MCAR	10%	13/13	9.46%	24.80%
		20%		18.49%	5.54%
		30%		27.94%	0.93%
		40%		36.97%	0.14%
		10%	4/13	2.91%	65.12%
		20%		5.69%	41.08%
		30%		8.60%	23.83%
		40%		11.38%	13.11%
	MAR	10%	12/13	8.73%	30.68%
		20%		17.06%	10.23%
		30%		25.79%	3.42%
		40%		34.13%	1.52%
		10%	4/13	2.91%	65.49%
		20%		5.69%	42.06%
		30%		8.60%	25.33%
		40%		11.38%	14.35%
	MNAR	10%	13/13	9.46%	25.76%
		20%		18.49%	8.20%
		30%		27.94%	2.16%
		40%		36.97%	0.49%
		10%	4/13	2.91%	65.27%
		20%		5.69%	42.76%
		30%		8.60%	25.95%
		40%		11.38%	14.15%

Data	Pattern	% miss./var	# var. miss.	average	average
				% miss. data	% complete obs
Fertility	MCAR	10%	5/5	8.77%	57.19%
		20%		17.54%	30.79%
		30%		26.32%	14.84%
		40%		32.89%	8.29%
		10%	2/5	3.51%	80.02%
		20%		7.02%	62.33%
		30%		10.53%	46.80%
		40%		13.16%	36.85%
	MAR	10%	4/5	7.02%	64.53%
		20%		14.04%	40.21%
		30%		19.30%	27.37%
		40%		26.32%	15.58%
		10%	2/5	3.51%	81.63%
		20%		7.02%	67.29%
		30%		10.53%	55.78%
		40%		13.16%	47.97%
	MNAR	10%	5/5	8.77%	62.76%
		20%		17.54%	28.54%
		30%		26.32%	11.08%
		40%		32.89%	4.67%
		10%	2/5	3.51%	80.53%
		20%		7.02%	61.51%
		30%		10.53%	44.81%
		40%		13.16%	34.00%
Birthweight	MCAR	10%	8/8	8.83%	43.15%
		20%		17.66%	16.94%
		30%		26.49%	5.91%
		40%		35.32%	1.71%
		10%	3/8	3.31%	73.07%
		20%		6.62%	51.40%
		30%		9.93%	34.51%
		40%		13.25%	21.95%
	MAR	10%	7/8	7.73%	48.16%
		20%		15.45%	21.25%
		30%		23.18%	8.40%
		40%		30.91%	3.00%
		10%	3/8	3.31%	74.23%
		20%		6.62%	54.11%
		30%		9.93%	38.55%
		40%		13.25%	25.62%
	MNAR	10%	8/8	8.83%	42.98%
		20%		17.66%	16.79%
		30%		26.49%	5.84%
		40%		35.32%	1.61%
		10%	3/8	3.31%	72.97%
		20%		6.62%	51.46%
		30%		9.93%	34.48%
		40%		13.25%	21.43%

Table C.4: For the simulated dataset, we show the average percentage of missing data and the average percentage of complete observations across $1,000$ simulations in which missingness is introduced according to a fixed pattern, fraction and scheme of missing values.

Data	Pattern	% miss./var	# var. miss.	average % miss. data	average % complete obs
Simulated	MCAR	10%	8/10	7.27%	43.11%
		20%		14.55%	16.92%
		30%		21.82%	5.75%
		40%		29.09%	1.64%
		10%	3/10	2.73%	72.78%
		20%		5.45%	51.12%
		30%		8.18%	34.36%
		40%		10.91%	21.67%
	MAR	10%	8/10	7.27%	57.98%
		20%		14.55%	32.86%
		30%		21.82%	15.17%
		40%		29.09%	4.65%
		10%	3/10	2.73%	76.02%
		20%		5.45%	56.12%
		30%		8.18%	39.75%
		40%		10.91%	23.43%
	MNAR	10%	8/10	7.27%	88.29%
		20%		14.55%	76.69%
		30%		21.82%	65.35%
		40%		29.09%	54.04%
		10%	3/10	2.73%	89.17%
		20%		5.45%	78.28%
		30%		8.18%	67.59%
		40%		10.91%	56.92%

Table C.5: Summary of mean MER values for the Survival dataset. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 3 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 2$) and only $p/3$ variables with missing values. N/I stands for "not implemented".

		Missing		Surrogates			Median/mode			Prox. Matrix			MICE			MIST			kNN		
Data	Technique	# Var.	%	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR
Survival	CondRF		0%	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
		m	10%	0.27	0.27	0.26	0.27	0.27	0.26	0.27	0.27	0.26	0.27	0.27	0.26	0.27	0.27	0.26	0.26	0.26	0.26
			20%	0.27	0.27	0.28	0.27	0.27	0.27	0.27	0.27	0.29	0.27	0.27	0.28	0.27	0.27	0.27	0.26	0.26	0.29
			30%	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.30	0.27	0.27	0.27	0.27	0.27	0.27	0.26	0.26	0.28
			40%	0.27	0.27	0.27	0.27	0.27	0.28	0.27	0.27	0.30	0.27	0.27	0.27	0.27	0.27	0.27	0.26	0.27	0.29
		1	10%	0.27	0.26	0.26	0.27	0.26	0.26	0.27	0.27	0.26	0.27	0.27	0.27	0.27	0.27	0.26	0.27	0.26	0.26
			20%	0.27	0.26	0.27	0.27	0.26	0.27	0.27	0.27	0.27	0.27	0.26	0.27	0.27	0.27	0.27	0.27	0.26	0.27
			30%	0.27	0.27	0.27	0.27	0.27	0.26	0.27	0.27	0.27	0.27	0.26	0.27	0.27	0.26	0.27	0.27	0.26	0.26
			40%	0.27	0.27	0.27	0.27	0.26	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.26	0.27
	CondTree		0%	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
		m	10%	0.27	0.27	0.28	0.28	0.28	0.27	0.28	0.28	0.28	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.28	0.27
			20%	0.27	0.27	0.28	0.27	0.28	0.26	0.27	0.28	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
			30%	0.27	0.27	0.27	0.28	0.28	0.26	0.27	0.28	0.27	0.27	0.27	0.26	0.27	0.27	0.27	0.27	0.27	0.26
			40%	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.28	0.27	0.27	0.27	0.26	0.27	0.27	0.27	0.27	0.27	0.26
		1	10%	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
			20%	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
			30%	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
			40%	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
	CART		0%	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
		m	10%	0.28	0.28	0.27	0.28	0.28	0.28	0.28	0.28	0.27	0.27	0.28	0.27	0.28	0.28	0.27	0.28	0.28	0.28
			20%	0.28	0.28	0.30	0.28	0.28	0.31	0.29	0.28	0.29	0.27	0.27	0.29	0.27	0.28	0.28	0.28	0.28	0.33
			30%	0.28	0.28	0.32	0.28	0.28	0.34	0.29	0.28	0.33	0.27	0.28	0.30	0.27	0.27	0.30	0.28	0.28	0.34
			40%	0.28	0.28	0.28	0.29	0.28	0.30	0.30	0.29	0.33	0.27	0.27	0.27	0.27	0.27	0.28	0.28	0.28	0.32
		1	10%	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
			20%	0.28	0.28	0.29	0.28	0.28	0.29	0.28	0.28	0.29	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.29
			30%	0.28	0.28	0.29	0.28	0.28	0.29	0.29	0.28	0.29	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.29
			40%	0.28	0.28	0.29	0.28	0.28	0.29	0.28	0.28	0.29	0.28	0.28	0.28	0.28	0.28	0.29	0.28	0.28	0.29
	RF		0%	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
		m	10%	N/I	N/I	N/I	0.32	0.32	0.30	0.33	0.32	0.31	0.31	0.32	0.30	0.31	0.32	0.30	0.31	0.31	0.31
			20%	N/I	N/I	N/I	0.32	0.32	0.33	0.34	0.32	0.35	0.31	0.31	0.31	0.31	0.31	0.32	0.31	0.31	0.34
			30%	N/I	N/I	N/I	0.32	0.32	0.38	0.34	0.32	0.38	0.30	0.31	0.35	0.30	0.31	0.34	0.31	0.31	0.39
			40%	N/I	N/I	N/I	0.32	0.31	0.35	0.35	0.32	0.38	0.30	0.30	0.34	0.30	0.30	0.33	0.31	0.31	0.37
		1	10%	N/I	N/I	N/I	0.32	0.32	0.31	0.32	0.32	0.31	0.32	0.32	0.31	0.32	0.32	0.31	0.32	0.32	0.31
			20%	N/I	N/I	N/I	0.32	0.32	0.33	0.32	0.32	0.33	0.32	0.31	0.32	0.31	0.31	0.32	0.31	0.31	0.33
			30%	N/I	N/I	N/I	0.31	0.31	0.33	0.32	0.32	0.33	0.31	0.31	0.32	0.31	0.31	0.32	0.31	0.31	0.33
			40%	N/I	N/I	N/I	0.31	0.31	0.32	0.33	0.32	0.32	0.31	0.31	0.32	0.31	0.31	0.31	0.31	0.31	0.32
	Bagging		0%	0.32	0.32	0.32	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	0.27	0.27	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.27	0.27	0.27	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.27	0.27	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.28	0.27	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		1	10%	0.27	0.27	0.27	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.27	0.27	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.27	0.26	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.27	0.27	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	0.26	0.26	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	0.27	0.27	0.25	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.27	0.27	0.27	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.27	0.27	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.27	0.27	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		1	10%	0.26	0.26	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.27	0.26	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.27	0.27	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.27	0.26	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	m	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.31	0.31	0.30	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.30	0.31	0.32	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.30	0.30	0.33	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.29	0.30	0.32	N/I	N/I	N/I
	Boot. RF	1	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.31	0.31	0.31	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.31	0.31	0.31	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.31	0.31	0.31	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.30	0.31	0.31	N/I	N/I	N/I

Table C.6: Summary of standard error (SE) estimates of each of the MER estimates for the Survival dataset. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 3 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 2$) and only $p/3$ variables with missing values. N/I stands for "not implemented".

		Missing		Surrogates			Median/mode			Prox. Matrix			MICE			MIST			kNN		
Data	Technique	# Var.	%	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR
Survival	CondRF		0%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		m	10%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06
			40%	0.05	0.05	0.05	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		1	10%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			40%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
	CondTree		0%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		m	10%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			40%	0.05	0.05	0.05	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		1	10%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			40%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
	CART		0%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		m	10%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	0.05	0.05	0.06	0.05	0.05	0.07	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.07
			30%	0.06	0.06	0.08	0.06	0.06	0.09	0.06	0.06	0.07	0.05	0.05	0.07	0.05	0.05	0.07	0.06	0.06	0.09
			40%	0.06	0.06	0.06	0.06	0.06	0.07	0.06	0.06	0.07	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.05	0.08
		1	10%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06	0.05
			40%	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.06
	RF		0%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		m	10%	N/I	N/I	N/I	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.06
			20%	N/I	N/I	N/I	0.05	0.05	0.06	0.06	0.05	0.07	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.06
			30%	N/I	N/I	N/I	0.06	0.05	0.07	0.06	0.05	0.07	0.05	0.05	0.07	0.05	0.05	0.07	0.06	0.05	0.07
			40%	N/I	N/I	N/I	0.06	0.06	0.08	0.06	0.06	0.08	0.05	0.05	0.07	0.06	0.05	0.07	0.06	0.05	0.08
		1	10%	N/I	N/I	N/I	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	N/I	N/I	N/I	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.05	0.05	0.06
			30%	N/I	N/I	N/I	0.05	0.05	0.07	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.06	0.05	0.05	0.07
			40%	N/I	N/I	N/I	0.05	0.05	0.06	0.05	0.06	0.06	0.05	0.05	0.06	0.05	0.06	0.06	0.05	0.05	0.06
	Bagging		0%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.06	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		1	10%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		1	10%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	m	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.06	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.06	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.07	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.07	N/I	N/I	N/I
	Boot. RF	1	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.05	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.06	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.06	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.06	N/I	N/I	N/I

Table C.7: Summary of mean relative improvement values with an imputation strategy compared to surrogate decisions through different missing data scenarios for the Survival dataset. Only CondRF, CondTree and CART were taken into account for these comparisons (because RF implementation in R -randomForest()- cannot be fitted on incomplete data). Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 3 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 2$) and only $p/3$ variables with missing values.

						Median/mode						Prox. Matrix						MICE			
		Missing		MCAR		MAR		MNAR		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Survival	CondRF		0%																		
		m	10%	-0.01	0.10	0.00	0.09	0.00	0.10	0.00	0.10	0.00	0.10	-0.01	0.11	0.00	0.09	0.01	0.08	-0.01	0.08
			20%	-0.01	0.14	0.00	0.13	0.02	0.13	-0.01	0.14	-0.01	0.13	-0.06	0.15	0.00	0.10	0.00	0.11	-0.01	0.11
			30%	0.00	0.16	0.00	0.14	0.00	0.13	-0.01	0.16	-0.01	0.14	-0.10	0.19	0.00	0.11	0.00	0.11	0.01	0.09
			40%	-0.01	0.17	0.00	0.14	-0.06	0.17	-0.02	0.18	-0.01	0.15	-0.13	0.21	-0.01	0.11	0.00	0.10	-0.01	0.05
		1	10%	0.00	0.07	0.00	0.07	0.00	0.07	0.00	0.08	-0.01	0.08	-0.01	0.08	0.00	0.06	0.00	0.07	-0.01	0.06
			20%	-0.01	0.10	0.00	0.09	0.00	0.09	-0.01	0.10	-0.01	0.11	-0.03	0.09	0.00	0.07	0.00	0.08	-0.01	0.08
			30%	0.00	0.10	0.00	0.10	0.00	0.07	-0.01	0.12	-0.01	0.11	-0.02	0.10	0.00	0.08	0.00	0.09	0.00	0.06
			40%	0.00	0.11	0.00	0.11	0.00	0.08	-0.01	0.13	-0.01	0.12	-0.01	0.09	0.00	0.08	-0.01	0.09	0.00	0.07
	CondTree		0%																		
		m	10%	-0.04	0.16	-0.03	0.13	0.02	0.11	-0.04	0.17	-0.03	0.13	-0.01	0.15	-0.02	0.12	-0.01	0.10	0.01	0.11
			20%	-0.04	0.17	-0.03	0.14	0.04	0.12	-0.04	0.19	-0.03	0.16	0.02	0.13	-0.01	0.10	-0.01	0.09	0.03	0.11
			30%	-0.05	0.18	-0.03	0.15	0.02	0.08	-0.04	0.18	-0.03	0.15	0.00	0.14	-0.01	0.10	0.00	0.08	0.02	0.08
			40%	-0.05	0.19	-0.02	0.15	-0.01	0.13	-0.03	0.19	-0.03	0.15	-0.01	0.11	-0.01	0.07	0.00	0.07	0.00	0.04
		1	10%	-0.01	0.09	-0.01	0.09	0.01	0.08	-0.02	0.10	-0.01	0.09	0.00	0.08	0.00	0.08	-0.01	0.08	0.00	0.07
			20%	-0.01	0.10	-0.02	0.10	0.01	0.05	-0.01	0.10	-0.02	0.10	0.00	0.08	0.00	0.06	0.00	0.07	0.01	0.06
			30%	-0.01	0.10	-0.01	0.10	0.00	0.02	-0.01	0.11	-0.02	0.10	0.00	0.06	0.00	0.05	-0.01	0.08	0.00	0.04
			40%	-0.02	0.13	-0.01	0.09	0.00	0.02	-0.02	0.12	-0.02	0.10	0.00	0.03	0.00	0.05	0.00	0.06	0.00	0.04
	CART		0%																		
		m	10%	-0.01	0.14	-0.01	0.13	-0.03	0.16	-0.02	0.15	-0.01	0.14	-0.02	0.17	0.02	0.14	0.01	0.13	0.02	0.14
			20%	-0.02	0.14	-0.01	0.13	-0.07	0.24	-0.05	0.19	-0.02	0.17	0.01	0.21	0.00	0.16	0.01	0.15	0.02	0.19
			30%	-0.02	0.15	-0.01	0.14	-0.07	0.23	-0.05	0.20	-0.02	0.15	-0.09	0.28	0.02	0.16	0.00	0.17	0.03	0.20
			40%	-0.02	0.16	-0.02	0.15	-0.07	0.25	-0.06	0.24	-0.04	0.18	-0.20	0.30	0.02	0.17	0.01	0.17	0.01	0.13
		1	10%	0.00	0.10	-0.01	0.09	-0.02	0.12	-0.01	0.11	-0.01	0.11	-0.01	0.11	0.00	0.11	0.00	0.11	0.00	0.10
			20%	0.00	0.10	0.00	0.12	-0.01	0.13	-0.02	0.13	-0.01	0.14	-0.01	0.14	0.00	0.13	0.00	0.14	0.00	0.13
			30%	0.00	0.11	-0.01	0.11	-0.01	0.13	-0.02	0.16	-0.02	0.14	-0.02	0.16	0.01	0.13	0.00	0.14	0.00	0.13
			40%	-0.01	0.12	-0.01	0.12	0.00	0.11	-0.02	0.17	-0.02	0.15	-0.01	0.15	0.01	0.15	0.00	0.14	0.01	0.10

						MIST						kNN			
		Missing		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Survival	CondRF		0%												
		m	10%	0.00	0.09	0.00	0.08	0.00	0.08	0.01	0.11	0.01	0.10	0.01	0.09
			20%	-0.01	0.11	0.00	0.11	0.01	0.11	0.01	0.13	0.01	0.12	-0.05	0.17
			30%	0.00	0.11	0.00	0.11	-0.01	0.10	0.01	0.13	0.01	0.13	-0.05	0.19
			40%	-0.01	0.11	0.00	0.10	-0.01	0.06	0.00	0.14	0.01	0.13	-0.09	0.17
		1	10%	0.00	0.06	0.00	0.07	0.00	0.06	0.00	0.08	0.00	0.08	-0.01	0.06
			20%	-0.01	0.08	-0.01	0.08	0.00	0.07	0.00	0.09	0.00	0.09	-0.01	0.09
			30%	0.00	0.08	0.00	0.09	0.00	0.06	0.00	0.10	0.00	0.10	0.01	0.07
			40%	0.00	0.09	0.00	0.09	-0.01	0.08	0.00	0.10	0.00	0.11	-0.01	0.08
	CondTree		0%												
		m	10%	-0.03	0.12	-0.02	0.11	0.00	0.09	-0.02	0.14	-0.01	0.12	0.01	0.10
			20%	-0.02	0.12	-0.02	0.11	0.03	0.12	-0.01	0.13	-0.01	0.12	0.03	0.12
			30%	-0.02	0.11	-0.01	0.08	0.02	0.09	-0.01	0.13	-0.01	0.11	0.02	0.08
			40%	-0.01	0.09	0.00	0.07	0.00	0.05	-0.01	0.12	0.00	0.10	0.00	0.06
		1	10%	-0.01	0.08	-0.01	0.08	0.00	0.06	-0.01	0.08	-0.01	0.09	0.00	0.06
			20%	-0.01	0.08	-0.01	0.08	0.01	0.06	0.00	0.08	0.00	0.07	0.01	0.05
			30%	-0.01	0.06	-0.01	0.09	0.00	0.04	0.00	0.07	0.00	0.07	0.00	0.03
			40%	0.00	0.05	0.00	0.06	0.00	0.06	0.00	0.08	0.00	0.07	0.00	0.03
	CART		0%												
		m	10%	0.01	0.13	0.01	0.13	0.02	0.14	0.00	0.16	0.01	0.14	-0.03	0.17
			20%	0.01	0.15	0.01	0.14	0.05	0.19	-0.01	0.18	0.00	0.16	-0.11	0.26
			30%	0.01	0.16	0.01	0.16	0.04	0.22	-0.01	0.18	-0.01	0.19	-0.10	0.28
			40%	0.03	0.16	0.02	0.16	0.00	0.14	-0.01	0.22	-0.01	0.19	-0.16	0.30
		1	10%	0.01	0.11	0.01	0.11	0.01	0.11	0.00	0.12	0.00	0.12	-0.01	0.12
			20%	0.01	0.12	0.00	0.12	0.01	0.13	-0.01	0.14	0.00	0.14	-0.01	0.15
			30%	0.02	0.13	0.01	0.14	0.00	0.14	0.00	0.15	-0.01	0.15	-0.01	0.13
			40%	0.01	0.14	0.01	0.15	0.00	0.12	0.00	0.16	-0.01	0.15	0.00	0.12

Table C.8: Summary of mean MER values for the Heart dataset. Techniques with prior kNN imputation could not be fitted since this dataset contains categorical predictors. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 13 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 12$) and only $p/3$ variables with missing values. N/I stands for "not implemented".

		Missing			Surrogates			Median/mode			Prox. Matrix			MICE			MIST	
Data	Technique	# Var.	%	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
Heart	CondRF		0%	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
		m	10%	0.17	0.17	0.17	0.18	0.18	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
			20%	0.17	0.18	0.19	0.19	0.18	0.19	0.17	0.17	0.17	0.17	0.17	0.18	0.17	0.17	0.18
			30%	0.18	0.18	0.23	0.20	0.19	0.22	0.18	0.18	0.18	0.18	0.18	0.20	0.18	0.18	0.20
			40%	0.18	0.19	0.29	0.21	0.20	0.25	0.19	0.19	0.20	0.18	0.18	0.24	0.18	0.18	0.25
		4	10%	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
			20%	0.17	0.17	0.17	0.18	0.18	0.18	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
			30%	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.17	0.17	0.17	0.17	0.17	0.17	0.17
			40%	0.18	0.18	0.18	0.18	0.18	0.19	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
	CondTree		0%	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
		m	10%	0.25	0.25	0.24	0.25	0.25	0.23	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.23
			20%	0.26	0.26	0.26	0.26	0.26	0.27	0.26	0.26	0.25	0.24	0.24	0.25	0.24	0.24	0.25
			30%	0.27	0.26	0.28	0.27	0.27	0.27	0.26	0.27	0.26	0.24	0.24	0.26	0.24	0.24	0.26
			40%	0.27	0.27	0.31	0.28	0.27	0.30	0.27	0.27	0.27	0.24	0.25	0.27	0.24	0.24	0.28
		4	10%	0.24	0.24	0.24	0.25	0.25	0.24	0.24	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24
			20%	0.24	0.25	0.24	0.25	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24
			30%	0.25	0.24	0.25	0.25	0.25	0.25	0.26	0.25	0.25	0.24	0.24	0.25	0.25	0.24	0.24
			40%	0.25	0.25	0.25	0.25	0.25	0.26	0.26	0.27	0.26	0.25	0.25	0.25	0.25	0.25	0.25
	CART		0%	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
		m	10%	0.22	0.22	0.22	0.23	0.23	0.22	0.23	0.22	0.22	0.21	0.21	0.21	0.21	0.21	0.21
			20%	0.24	0.23	0.24	0.24	0.24	0.26	0.24	0.23	0.24	0.21	0.21	0.22	0.21	0.21	0.22
			30%	0.26	0.25	0.30	0.26	0.25	0.30	0.25	0.25	0.25	0.21	0.21	0.24	0.21	0.21	0.24
			40%	0.28	0.26	0.33	0.27	0.26	0.32	0.27	0.27	0.28	0.22	0.22	0.27	0.22	0.22	0.28
		4	10%	0.22	0.22	0.21	0.22	0.22	0.22	0.22	0.22	0.21	0.21	0.21	0.21	0.21	0.21	0.21
			20%	0.22	0.22	0.22	0.22	0.22	0.23	0.22	0.22	0.22	0.21	0.21	0.22	0.21	0.21	0.22
			30%	0.22	0.22	0.22	0.23	0.23	0.23	0.23	0.23	0.24	0.22	0.22	0.22	0.22	0.21	0.22
			40%	0.23	0.23	0.23	0.23	0.23	0.24	0.25	0.25	0.25	0.22	0.22	0.22	0.22	0.22	0.22
	RF		0%	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
		m	10%	N/I	N/I	N/I	0.19	0.19	0.19	0.18	0.18	0.19	0.18	0.18	0.19	0.18	0.18	0.19
			20%	N/I	N/I	N/I	0.20	0.19	0.22	0.19	0.18	0.20	0.18	0.19	0.20	0.18	0.18	0.20
			30%	N/I	N/I	N/I	0.20	0.20	0.25	0.19	0.19	0.22	0.19	0.19	0.22	0.18	0.18	0.22
			40%	N/I	N/I	N/I	0.21	0.21	0.31	0.20	0.20	0.26	0.19	0.19	0.26	0.19	0.19	0.27
		4	10%	N/I	N/I	N/I	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
			20%	N/I	N/I	N/I	0.19	0.19	0.19	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
			30%	N/I	N/I	N/I	0.19	0.19	0.20	0.19	0.18	0.19	0.18	0.18	0.19	0.18	0.18	0.19
			40%	N/I	N/I	N/I	0.19	0.19	0.20	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
	Bagging		0%	0.19	0.19	0.19	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	0.23	0.23	0.23	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.48	0.39	0.51	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.49	0.45	0.55	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.50	0.48	0.55	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		4	10%	0.20	0.20	0.20	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.22	0.22	0.23	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.24	0.23	0.24	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.25	0.25	0.29	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	0.18	0.18	0.18	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	0.18	0.18	0.18	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.18	0.18	0.19	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.18	0.18	0.22	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.18	0.19	0.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		4	10%	0.18	0.18	0.18	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.18	0.18	0.18	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.18	0.18	0.18	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.19	0.19	0.19	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	m	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.19	0.19	0.19
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.19	0.19	0.20
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.19	0.19	0.22
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.19	0.19	0.27
	Boot. RF	4	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.18	0.18	0.18
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.18	0.19	0.19
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.19	0.19	0.19
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.19	0.19	0.19

Table C.9: Summary of standard error (SE) estimates of each of the MER estimates for the Heart dataset. Techniques with prior kNN imputation could not be fitted since this dataset contains categorical predictors. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 13 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 12$) and only $p/3$ variables with missing values. N/I stands for "not implemented".

		Missing			Surrogates			Median/mode			Prox. Matrix			MICE			MIST	
Data	Technique	# Var.	%	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
Heart	CondRF		0%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		m	10%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			40%	0.05	0.05	0.07	0.06	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.06	0.05	0.05	0.06
		4	10%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			40%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
	CondTree		0%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
		m	10%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
			20%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
			30%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
			40%	0.06	0.06	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
		4	10%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
			20%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
			30%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
			40%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	CART		0%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
		m	10%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.05	0.06	0.05	0.06	0.06
			20%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.06	0.06	0.06	0.06	0.05
			30%	0.06	0.06	0.07	0.06	0.06	0.07	0.06	0.06	0.06	0.05	0.05	0.06	0.05	0.06	0.06
			40%	0.07	0.06	0.08	0.07	0.06	0.07	0.07	0.06	0.08	0.05	0.06	0.06	0.06	0.06	0.06
		4	10%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
			20%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.05	0.06	0.06	0.06	0.06	0.06
			30%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
			40%	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	RF		0%	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
		m	10%	N/I	N/I	N/I	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	N/I	N/I	N/I	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	N/I	N/I	N/I	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			40%	N/I	N/I	N/I	0.05	0.05	0.07	0.05	0.05	0.07	0.05	0.05	0.06	0.05	0.05	0.06
		4	10%	N/I	N/I	N/I	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			20%	N/I	N/I	N/I	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			30%	N/I	N/I	N/I	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
			40%	N/I	N/I	N/I	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
	Bagging		0%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	0.06	0.06	0.06	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.08	0.13	0.12	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.08	0.10	0.06	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.08	0.08	0.06	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		4	10%	0.05	0.06	0.06	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.06	0.06	0.06	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.06	0.06	0.07	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.07	0.07	0.11	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.05	0.05	0.06	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.05	0.05	0.06	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		4	10%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	0.05	0.05	0.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	m	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.05
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.05
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.05
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.06
	Boot. RF	4	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.05
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.05
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.05
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	0.05	0.05	0.05

Table C.10: Summary of mean relative improvement values with an imputation strategy compared to surrogate decisions through different missing data scenarios for the Heart dataset. Only CondRF, CondTree and CART were taken into account for these comparisons (because RF implementation in R -randomForest()- cannot be fitted on incomplete data). In addition, kNN imputation could not be implemented since this dataset contains categorical predictors. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 13 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 12$) and only $p/3$ variables with missing values.

						Median/mode						Prox. Matrix						MICE			
		Missing		MCAR		MAR		MNAR		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Heart	CondRF		0%																		
		m	10%	-0.05	0.18	-0.06	0.17	-0.01	0.19	-0.01	0.16	-0.01	0.16	0.01	0.17	-0.01	0.14	0.00	0.13	0.00	0.14
			20%	-0.10	0.24	-0.07	0.23	-0.05	0.26	-0.02	0.18	-0.01	0.22	0.07	0.18	-0.01	0.17	0.00	0.18	0.03	0.15
			30%	-0.15	0.27	-0.09	0.24	0.02	0.27	-0.04	0.22	-0.02	0.24	0.19	0.16	-0.01	0.19	0.01	0.19	0.12	0.14
			40%	-0.17	0.28	-0.11	0.28	0.13	0.21	-0.03	0.23	-0.03	0.29	0.31	0.15	0.01	0.23	0.01	0.21	0.19	0.13
		4	10%	-0.02	0.16	-0.02	0.13	-0.01	0.16	-0.02	0.18	-0.02	0.17	0.00	0.15	-0.01	0.14	0.00	0.12	0.00	0.12
			20%	-0.02	0.16	-0.03	0.16	-0.04	0.17	-0.02	0.28	-0.03	0.25	-0.02	0.24	0.00	0.16	0.01	0.14	0.00	0.14
			30%	-0.02	0.14	-0.02	0.15	-0.05	0.17	-0.04	0.30	-0.03	0.27	-0.03	0.28	0.02	0.15	0.01	0.14	0.01	0.14
			40%	-0.01	0.16	-0.02	0.15	-0.04	0.14	-0.03	0.34	-0.06	0.33	-0.02	0.33	0.03	0.15	0.02	0.14	0.02	0.14
	CondTree		0%																		
		m	10%	-0.05	0.27	-0.03	0.26	-0.03	0.34	-0.03	0.26	-0.03	0.28	-0.04	0.31	0.00	0.25	0.01	0.22	-0.04	0.28
			20%	-0.06	0.33	-0.05	0.36	-0.08	0.31	-0.03	0.28	-0.05	0.45	-0.01	0.28	0.02	0.27	0.01	0.33	0.01	0.27
			30%	-0.06	0.32	-0.05	0.29	0.00	0.28	-0.03	0.28	-0.05	0.31	0.06	0.26	0.07	0.26	0.05	0.25	0.07	0.22
			40%	-0.09	0.32	-0.05	0.32	-0.02	0.30	-0.05	0.30	-0.06	0.34	0.09	0.25	0.06	0.28	0.05	0.26	0.09	0.20
		4	10%	-0.04	0.22	-0.03	0.20	-0.03	0.23	-0.04	0.26	-0.05	0.24	-0.03	0.25	-0.02	0.18	-0.02	0.19	-0.03	0.21
			20%	-0.05	0.20	-0.03	0.19	-0.04	0.20	-0.06	0.28	-0.07	0.29	-0.02	0.28	-0.01	0.19	-0.01	0.19	-0.01	0.18
			30%	-0.03	0.18	-0.04	0.20	-0.04	0.19	-0.08	0.34	-0.09	0.35	-0.05	0.32	0.00	0.19	-0.02	0.20	-0.01	0.20
			40%	-0.03	0.17	-0.02	0.17	-0.03	0.13	-0.09	0.36	-0.12	0.38	-0.09	0.30	-0.01	0.15	0.01	0.17	0.00	0.14
	CART		0%																		
		m	10%	-0.06	0.30	-0.08	0.29	-0.06	0.37	-0.05	0.30	-0.04	0.30	-0.06	0.35	0.02	0.25	0.02	0.24	-0.02	0.28
			20%	-0.08	0.32	-0.10	0.35	-0.15	0.38	-0.05	0.33	-0.06	0.33	-0.02	0.34	0.07	0.28	0.03	0.29	0.05	0.27
			30%	-0.06	0.30	-0.06	0.30	-0.07	0.34	-0.04	0.34	-0.05	0.33	0.11	0.28	0.13	0.27	0.10	0.27	0.16	0.22
			40%	0.00	0.29	-0.03	0.31	-0.02	0.32	0.00	0.31	-0.07	0.34	0.11	0.29	0.19	0.25	0.13	0.27	0.17	0.21
		4	10%	-0.03	0.23	-0.03	0.19	-0.04	0.24	-0.04	0.28	-0.03	0.25	-0.04	0.26	0.00	0.19	0.00	0.17	-0.02	0.20
			20%	-0.05	0.23	-0.05	0.23	-0.08	0.25	-0.07	0.34	-0.08	0.34	-0.07	0.33	0.00	0.21	0.00	0.22	-0.03	0.22
			30%	-0.05	0.23	-0.05	0.22	-0.07	0.23	-0.11	0.41	-0.12	0.38	-0.13	0.38	0.00	0.22	-0.01	0.22	-0.01	0.20
			40%	-0.06	0.22	-0.03	0.19	-0.05	0.19	-0.17	0.44	-0.17	0.43	-0.14	0.36	0.00	0.21	0.01	0.20	0.01	0.18

						MIST			
		Missing		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD
Heart	CondRF		0%						
		m	10%	-0.02	0.14	0.00	0.13	-0.01	0.14
			20%	-0.01	0.17	0.00	0.18	0.02	0.15
			30%	-0.01	0.19	0.02	0.17	0.10	0.15
			40%	0.00	0.21	0.02	0.20	0.16	0.13
		4	10%	-0.01	0.14	-0.01	0.12	-0.01	0.13
			20%	0.00	0.15	0.01	0.13	0.00	0.13
			30%	0.01	0.15	0.01	0.14	0.01	0.14
			40%	0.02	0.15	0.02	0.14	0.01	0.14
	CondTree		0%						
		m	10%	0.01	0.25	0.02	0.23	-0.03	0.27
			20%	0.03	0.25	0.03	0.32	0.02	0.25
			30%	0.08	0.25	0.07	0.24	0.06	0.22
			40%	0.07	0.28	0.07	0.28	0.08	0.20
		4	10%	-0.02	0.19	-0.02	0.18	-0.03	0.22
			20%	-0.02	0.20	-0.01	0.18	-0.01	0.19
			30%	-0.01	0.19	-0.02	0.20	-0.01	0.21
			40%	-0.01	0.16	0.00	0.16	0.00	0.12
	CART		0%						
		m	10%	0.03	0.26	0.02	0.26	-0.01	0.28
			20%	0.09	0.26	0.06	0.27	0.06	0.27
			30%	0.15	0.25	0.12	0.26	0.15	0.24
			40%	0.19	0.26	0.13	0.26	0.14	0.21
		4	10%	-0.01	0.24	0.00	0.18	-0.02	0.20
			20%	0.01	0.21	0.00	0.22	-0.02	0.23
			30%	0.01	0.21	0.00	0.21	0.00	0.19
			40%	0.01	0.21	0.02	0.18	0.01	0.16

Table C.11: Summary of mean MSPE values for the Fertility dataset. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 5 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 4$) and only $p/3$ variables with missing values. N/I stands for "not implemented".

		Missing			Surrogates			Median/mode			Prox. Matrix			MICE			MIST			kNN	
Data	Technique	# Var.	%	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR
Fertility	CondRF		0%	127.77	127.77	127.86	127.77	127.77	127.86	127.77	127.77	127.86	127.77	127.77	127.86	127.77	127.77	127.86	127.77	127.77	127.86
		m	10%	128.19	127.95	129.00	127.58	127.17	132.37	127.03	126.58	131.21	122.79	123.24	123.96	124.94	125.41	123.67	125.56	126.13	132.18
			20%	124.35	123.84	137.13	127.45	126.96	143.15	125.06	124.98	143.71	119.18	119.59	131.33	122.73	123.46	131.19	124.73	124.68	151.72
			30%	160.05	133.61	163.05	129.27	129.16	171.03	125.64	124.71	171.18	118.68	119.94	149.81	122.86	123.07	156.57	125.90	124.56	169.92
			40%	164.29	138.43	164.48	132.15	129.21	154.69	125.46	124.03	172.26	118.20	119.01	158.08	123.89	123.18	158.22	124.47	124.70	160.74
		2	10%	128.11	128.21	128.85	127.97	127.76	129.37	127.33	127.25	127.59	126.42	125.61	126.14	127.13	126.92	126.56	127.51	127.18	126.77
			20%	127.60	126.92	127.38	128.20	127.94	127.72	126.87	125.69	127.56	124.94	123.49	126.05	125.91	125.74	126.14	126.96	125.40	126.88
			30%	127.08	127.48	127.78	126.51	127.76	128.89	126.41	125.79	128.98	123.36	122.20	127.76	124.85	124.93	128.42	125.85	125.80	128.16
			40%	127.71	128.26	127.98	127.62	127.39	128.30	126.86	124.48	128.38	123.29	121.51	127.77	125.42	124.62	128.16	126.02	126.62	128.54
	CondTree		0%	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09	130.09
		m	10%	138.10	135.32	144.17	136.59	132.47	146.26	133.09	131.32	150.54	114.18	118.11	132.30	120.39	122.10	135.50	128.47	131.63	151.31
			20%	142.63	135.32	157.02	137.64	134.36	168.60	134.34	131.25	170.10	111.56	113.92	140.73	119.09	121.61	143.12	128.75	129.65	175.81
			30%	143.86	138.41	188.49	144.19	137.64	215.86	134.65	130.52	197.26	111.69	117.09	164.34	121.52	121.26	176.74	133.25	130.15	182.40
			40%	146.31	139.92	169.33	146.75	141.06	181.69	133.31	131.00	186.53	113.79	118.01	165.45	124.08	125.46	168.29	131.56	132.10	168.53
		2	10%	131.86	131.35	132.73	132.09	128.10	133.71	130.01	131.14	132.91	124.38	125.10	131.48	126.26	127.40	130.42	130.21	130.41	131.53
			20%	132.86	132.48	130.38	134.69	131.50	131.59	130.46	129.24	129.26	122.27	123.25	129.29	124.61	126.61	128.57	131.72	132.88	129.53
			30%	131.33	130.46	131.80	132.19	129.34	132.54	130.20	129.48	132.74	121.55	122.86	131.47	124.98	124.87	131.78	130.48	130.86	131.77
			40%	131.54	131.47	133.14	131.72	130.52	133.35	131.18	130.64	133.56	121.07	123.01	133.12	125.47	126.48	133.06	131.21	132.24	134.12
	CART		0%	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24	130.24
		m	10%	129.66	126.75	124.63	125.54	126.68	129.62	127.08	126.34	133.11	111.04	113.81	119.13	115.73	118.38	121.39	126.06	127.73	130.98
			20%	125.30	125.83	155.77	128.67	128.75	159.44	128.99	128.34	163.67	105.59	110.14	127.49	111.97	116.05	123.96	127.94	129.90	174.41
			30%	132.52	123.64	185.30	134.55	125.50	227.94	126.94	126.76	203.60	104.05	108.71	147.37	110.91	112.03	157.98	130.19	128.35	182.12
			40%	134.26	128.49	173.22	138.34	133.72	184.14	129.69	129.24	236.01	105.16	108.89	161.91	111.49	114.56	160.12	133.20	128.94	181.46
		2	10%	128.09	126.36	127.71	127.16	126.95	126.94	128.19	128.84	131.57	120.93	123.30	127.36	123.29	125.96	128.46	129.21	129.71	133.29
			20%	128.54	125.40	125.56	127.48	130.45	125.58	131.00	129.59	125.71	119.33	120.24	124.37	122.54	125.22	122.74	129.92	131.96	127.13
			30%	125.04	123.81	127.38	124.78	128.65	132.41	129.71	127.54	132.21	114.94	118.14	125.95	119.15	122.10	127.07	127.89	132.37	132.67
			40%	124.09	124.64	126.35	122.70	130.52	129.38	129.64	132.08	148.09	113.21	116.30	127.59	117.70	120.09	126.23	127.60	130.47	135.48
	RF		0%	74.25	74.25	74.26	74.25	74.25	74.26	74.25	74.25	74.26	74.25	74.25	74.26	74.25	74.25	74.26	74.25	74.25	74.26
		m	10%	N/I	N/I	N/I	76.82	76.21	95.48	76.75	76.70	97.24	75.74	76.66	89.48	77.41	78.14	90.68	77.40	78.02	107.47
			20%	N/I	N/I	N/I	79.59	79.08	116.98	79.68	79.55	121.06	79.52	79.29	102.96	81.82	81.39	103.37	82.47	81.72	129.43
			30%	N/I	N/I	N/I	84.90	83.00	249.93	84.97	80.52	274.27	84.08	81.27	203.72	85.53	83.31	195.71	88.95	84.21	281.98
			40%	N/I	N/I	N/I	90.89	87.46	180.06	89.32	86.90	231.20	88.50	86.32	171.77	90.86	87.83	172.60	92.11	87.89	194.39
		2	10%	N/I	N/I	N/I	75.18	75.59	81.92	75.01	75.75	82.67	75.38	75.53	79.23	75.96	76.21	80.00	75.95	75.95	84.08
			20%	N/I	N/I	N/I	76.60	77.91	82.10	77.27	77.98	83.72	76.90	77.71	82.14	77.92	78.13	82.43	77.38	78.24	83.84
			30%	N/I	N/I	N/I	77.50	79.09	95.03	78.76	79.85	94.41	77.12	79.47	90.07	79.23	78.86	93.15	78.48	80.72	94.36
			40%	N/I	N/I	N/I	78.56	79.44	90.40	79.99	81.90	94.91	78.02	80.75	87.03	79.87	79.14	88.80	79.16	80.90	91.05
	Bagging		0%	94.58	94.58	94.62	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	116.84	114.42	134.54	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	174.11	171.60	170.40	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	193.31	174.73	178.66	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	225.10	199.47	228.91	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		2	10%	104.15	107.93	118.66	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	112.78	118.35	128.83	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	164.40	136.63	175.54	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	172.36	157.66	187.25	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	105.68	105.68	105.88	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	110.44	109.06	122.11	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	115.70	112.49	131.23	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	123.63	116.74	158.33	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	128.01	120.55	157.67	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		2	10%	107.44	107.27	111.52	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	109.62	108.91	111.43	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	110.31	109.83	113.38	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	111.44	110.73	113.34	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	m	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	83.29	83.31	95.49	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	87.92	85.88	108.88	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	92.81	89.14	183.04	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	97.44	94.31	172.87	N/I	N/I	N/I
	Boot. RF	2	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	86.22	82.27	86.52	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	83.29	82.04	87.75	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	84.42	84.68	97.10	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	85.33	85.80	95.19	N/I	N/I	N/I

Table C.12: Summary of standard error (SE) estimates of each of the MSPE estimates for the Fertility dataset. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 5 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 4$) and only $p/3$ variables with missing values. N/I stands for "not implemented".

		Missing		Surrogates			Median/mode			Prox. Matrix			MICE			MIST			kNN		
Data	Technique	# Var.	%	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR
Fertility	CondRF		0%	61.56	61.56	61.60	61.56	61.56	61.60	61.56	61.56	61.60	61.56	61.56	61.60	61.56	61.56	61.60	61.56	61.56	61.60
		m	10%	62.34	61.97	72.64	63.93	61.95	69.96	63.59	61.24	71.52	58.46	60.19	67.73	62.06	61.86	66.68	58.58	60.49	77.99
			20%	62.81	61.68	69.05	62.80	64.23	68.61	61.52	61.58	73.61	58.38	58.51	71.02	60.84	61.92	66.90	60.13	60.36	86.28
			30%	74.32	69.18	74.98	66.18	67.32	77.04	65.16	63.21	81.36	59.49	59.98	70.74	63.40	62.75	72.62	63.23	59.34	74.85
			40%	75.33	70.60	75.36	66.39	65.40	67.94	63.06	61.14	80.29	59.34	61.39	73.25	62.38	62.56	70.14	59.48	61.28	74.65
		2	10%	62.55	62.43	64.19	62.46	61.74	63.61	60.73	62.40	62.48	61.75	61.12	62.53	62.51	61.99	62.09	61.30	61.30	62.58
			20%	63.31	62.75	65.35	63.25	62.78	64.76	61.88	60.98	65.67	61.09	60.35	64.14	62.71	61.00	63.93	61.61	58.97	64.42
			30%	61.67	63.18	63.85	61.60	62.52	63.75	61.62	63.15	64.75	60.00	60.28	63.49	61.43	61.88	63.48	61.00	62.67	63.28
			40%	63.30	64.99	62.06	62.93	63.69	62.17	62.44	62.79	62.23	61.17	60.23	61.98	62.53	63.48	61.85	60.97	63.99	62.35
	CondTree		0%	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06	78.06
		m	10%	73.49	75.42	77.54	74.71	74.54	89.68	79.00	74.90	87.61	63.59	66.88	69.47	70.77	71.78	72.49	71.20	75.79	82.60
			20%	72.64	72.33	86.51	70.70	75.83	87.93	75.78	75.51	88.31	61.16	62.97	75.85	65.72	68.91	74.87	69.20	73.00	93.67
			30%	74.22	75.83	99.69	78.56	75.52	119.96	79.77	77.96	111.60	59.55	64.06	79.32	66.28	66.71	86.85	71.95	67.59	91.38
			40%	74.79	69.69	74.63	76.45	72.75	77.80	75.30	72.42	101.19	59.60	62.97	75.28	65.44	65.31	74.43	65.26	69.59	75.21
		2	10%	74.91	75.93	77.28	76.28	75.19	75.23	76.72	78.29	77.46	74.73	72.29	73.18	75.80	75.50	73.33	76.89	76.62	75.28
			20%	76.91	75.99	73.19	76.48	79.44	72.28	77.42	75.47	75.17	73.34	69.45	73.39	74.64	72.13	71.92	76.06	72.55	74.00
			30%	74.14	72.29	73.17	75.66	72.69	75.35	75.10	75.35	77.78	67.28	67.02	72.79	70.68	69.46	74.07	72.68	73.45	73.32
			40%	70.43	73.08	70.74	70.84	74.37	71.11	74.76	77.03	71.30	68.12	64.83	70.71	68.91	71.57	70.54	70.56	74.33	71.69
	CART		0%	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57	70.57
		m	10%	70.72	67.07	64.77	70.96	69.80	65.88	74.11	70.19	72.26	63.51	63.37	63.40	67.82	64.92	61.09	70.07	71.00	77.73
			20%	67.98	70.86	78.49	72.45	75.98	79.32	76.88	77.20	89.32	60.08	63.76	73.69	64.93	66.51	60.26	75.39	78.00	99.02
			30%	74.60	67.97	106.62	81.45	69.90	111.96	75.69	71.97	117.44	57.31	61.24	76.85	61.72	62.58	79.80	77.21	73.20	98.24
			40%	70.82	68.75	86.63	74.94	72.38	81.75	75.92	74.46	135.16	56.81	60.39	80.33	61.49	61.72	72.79	75.85	72.65	85.96
		2	10%	69.65	71.48	67.33	70.39	73.24	68.37	71.08	73.30	72.42	68.94	68.78	68.38	70.42	71.62	68.05	72.13	73.31	71.75
			20%	70.28	67.72	66.82	71.68	75.14	67.43	72.57	73.93	71.15	67.82	66.06	67.78	67.84	69.18	65.37	72.88	73.03	68.53
			30%	66.32	64.43	65.72	67.61	71.42	77.60	74.67	73.72	80.56	62.85	63.40	67.06	65.12	66.92	66.83	69.41	73.66	72.96
			40%	64.28	67.57	64.91	64.91	72.65	68.17	71.79	79.08	99.41	62.18	64.65	70.85	63.14	66.48	65.73	69.25	71.49	81.54
	RF		0%	39.62	39.62	39.75	39.62	39.62	39.75	39.62	39.62	39.75	39.62	39.62	39.75	39.62	39.62	39.75	39.62	39.62	39.75
		m	10%	N/I	N/I	N/I	44.24	40.80	53.08	44.64	41.58	57.85	39.73	40.95	50.20	41.74	41.43	49.35	41.00	43.77	65.75
			20%	N/I	N/I	N/I	43.67	46.82	62.14	46.02	46.21	69.04	41.48	42.88	56.95	41.86	43.39	47.98	48.12	48.00	81.24
			30%	N/I	N/I	N/I	49.17	46.03	170.36	51.34	43.82	179.50	43.99	41.98	139.11	44.16	43.36	129.82	52.78	46.70	182.43
			40%	N/I	N/I	N/I	51.98	49.03	80.03	50.01	48.88	110.73	46.14	45.64	77.07	46.36	45.33	75.34	51.91	50.47	85.60
		2	10%	N/I	N/I	N/I	40.75	41.46	46.41	40.66	41.45	47.53	40.74	40.51	43.76	41.01	40.61	44.10	41.50	41.94	49.05
			20%	N/I	N/I	N/I	42.21	41.66	46.80	42.32	41.84	47.45	41.31	41.12	45.74	41.41	40.85	44.07	42.26	40.48	47.17
			30%	N/I	N/I	N/I	41.03	43.48	65.02	44.14	44.13	64.24	40.20	41.54	55.54	41.24	41.56	57.20	42.48	44.73	61.88
			40%	N/I	N/I	N/I	41.54	43.62	48.28	42.27	45.64	53.69	40.90	42.39	47.35	40.67	42.00	46.74	40.93	44.05	50.17
	Bagging		0%	54.86	54.86	54.77	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	64.94	64.75	79.79	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	83.13	80.63	79.23	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	99.87	80.60	83.80	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	158.51	103.99	129.38	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		2	10%	59.74	61.45	75.71	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	61.28	64.27	76.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	78.81	70.12	84.91	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	79.58	76.39	84.94	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	61.72	61.72	61.97	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	62.41	61.75	72.89	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	63.42	62.67	72.30	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	65.68	65.84	72.05	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	65.93	64.69	71.22	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		2	10%	62.56	61.95	64.40	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	63.34	62.68	65.26	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	61.65	62.30	63.07	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	62.00	63.09	60.91	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	m	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	43.78	43.74	51.85	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	44.67	45.36	52.47	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	47.09	45.81	104.86	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	49.39	49.19	77.23	N/I	N/I	N/I
	Boot. RF	2	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	46.02	43.94	47.58	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	42.17	43.41	46.10	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	42.07	45.22	54.22	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	42.60	45.50	49.31	N/I	N/I	N/I

Table C.13: Summary of mean relative improvement values with an imputation strategy compared to surrogate decisions through different missing data scenarios for the Fertility dataset. Only CondRF, CondTree and CART were taken into account for these comparisons (because RF implementation in R -randomForest()- cannot be fitted on incomplete data). Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 5 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 4$) and only $p/3$ variables with missing values.

				Median/mode						Prox. Matrix						MICE					
		Missing		MCAR		MAR		MNAR		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Fertility	CondRF		0%																		
		m	10%	0.00	0.09	0.00	0.08	-0.05	0.16	0.01	0.08	0.01	0.09	-0.03	0.14	0.04	0.11	0.03	0.10	0.03	0.09
			20%	-0.04	0.17	-0.04	0.15	-0.06	0.14	-0.03	0.18	-0.04	0.21	-0.06	0.12	0.03	0.14	0.02	0.16	0.05	0.13
			30%	0.18	0.22	0.00	0.25	-0.08	0.32	0.19	0.28	0.01	0.34	-0.07	0.26	0.25	0.20	0.07	0.25	0.08	0.14
			40%	0.19	0.17	0.04	0.25	0.04	0.17	0.20	0.30	0.04	0.37	-0.07	0.27	0.27	0.19	0.11	0.26	0.03	0.13
		2	10%	0.00	0.07	0.00	0.06	-0.01	0.06	0.00	0.09	0.01	0.07	0.01	0.04	0.01	0.07	0.02	0.08	0.02	0.09
			20%	-0.01	0.10	-0.02	0.12	-0.01	0.04	-0.01	0.18	-0.01	0.17	0.00	0.05	0.01	0.11	0.02	0.13	0.01	0.08
			30%	-0.01	0.14	-0.02	0.17	-0.01	0.09	-0.03	0.28	-0.01	0.24	-0.01	0.07	0.02	0.17	0.03	0.18	0.00	0.05
			40%	-0.01	0.11	0.00	0.12	0.00	0.04	-0.02	0.26	0.00	0.24	0.00	0.03	0.03	0.12	0.04	0.16	0.00	0.03
	CondTree		0%																		
		m	10%	-0.04	0.34	-0.02	0.34	-0.04	0.39	0.01	0.33	-0.04	0.68	-0.08	0.42	0.13	0.31	0.06	0.51	0.06	0.23
			20%	-0.02	0.41	-0.03	0.34	-0.15	0.49	0.02	0.41	-0.02	0.44	-0.16	0.46	0.18	0.31	0.12	0.36	0.07	0.27
			30%	-0.06	0.46	-0.04	0.34	-0.16	0.39	0.02	0.42	-0.01	0.51	-0.06	0.25	0.19	0.26	0.11	0.36	0.09	0.19
			40%	-0.06	0.39	-0.05	0.39	-0.11	0.36	0.02	0.48	0.00	0.44	-0.11	0.45	0.19	0.28	0.13	0.30	0.02	0.11
		2	10%	-0.03	0.35	0.00	0.23	-0.04	0.25	-0.03	0.47	-0.03	0.41	-0.02	0.23	0.03	0.31	0.01	0.32	-0.03	0.35
			20%	-0.06	0.37	-0.01	0.27	-0.03	0.21	-0.04	0.44	-0.04	0.55	0.00	0.24	0.05	0.37	0.02	0.39	-0.01	0.28
			30%	-0.04	0.30	-0.02	0.28	-0.01	0.21	-0.06	0.52	-0.07	0.58	-0.01	0.16	0.04	0.32	0.01	0.40	-0.01	0.15
			40%	-0.03	0.29	-0.01	0.23	0.00	0.10	-0.06	0.52	-0.05	0.43	0.00	0.08	0.06	0.22	0.01	0.38	0.00	0.07
	CART		0%																		
		m	10%	-0.03	0.40	-0.05	0.46	-0.15	0.61	-0.04	0.50	-0.07	0.54	-0.17	0.55	0.10	0.42	0.06	0.36	-0.02	0.43
			20%	-0.11	0.53	-0.08	0.48	-0.11	0.48	-0.11	0.58	-0.12	0.60	-0.10	0.45	0.10	0.40	0.07	0.39	0.15	0.34
			30%	-0.09	0.52	-0.08	0.45	-0.46	0.94	-0.06	0.57	-0.14	0.65	-0.21	0.60	0.15	0.36	0.07	0.38	0.12	0.36
			40%	-0.11	0.54	-0.11	0.46	-0.19	0.57	-0.07	0.60	-0.12	0.66	-0.53	1.05	0.16	0.35	0.10	0.35	-0.01	0.42
		2	10%	-0.02	0.26	-0.04	0.34	-0.02	0.24	-0.04	0.37	-0.07	0.43	-0.07	0.43	0.04	0.28	-0.01	0.32	-0.03	0.40
			20%	-0.02	0.25	-0.09	0.50	-0.02	0.26	-0.09	0.50	-0.11	0.56	-0.03	0.34	0.04	0.30	0.00	0.37	-0.01	0.32
			30%	-0.02	0.28	-0.09	0.56	-0.06	0.61	-0.12	0.72	-0.10	0.59	-0.05	0.43	0.05	0.32	0.00	0.40	0.00	0.21
			40%	-0.01	0.21	-0.08	0.37	-0.04	0.29	-0.15	0.73	-0.14	0.58	-0.20	0.77	0.06	0.32	0.02	0.39	-0.02	0.26

				MIST						kNN					
		Missing		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Fertility	CondRF		0%												
		m	10%	0.03	0.09	0.02	0.08	0.03	0.10	0.01	0.09	0.01	0.08	-0.02	0.19
			20%	0.00	0.13	-0.01	0.14	0.04	0.12	-0.03	0.17	-0.03	0.19	-0.09	0.15
			30%	0.22	0.23	0.05	0.24	0.03	0.21	0.19	0.28	0.01	0.32	-0.07	0.21
			40%	0.23	0.21	0.07	0.26	0.02	0.16	0.22	0.25	0.05	0.32	0.01	0.18
		2	10%	0.01	0.06	0.01	0.06	0.01	0.05	0.00	0.07	0.00	0.07	0.01	0.05
			20%	0.01	0.10	0.00	0.14	0.01	0.05	-0.01	0.14	-0.01	0.17	0.00	0.05
			30%	0.01	0.16	0.00	0.19	-0.01	0.08	-0.01	0.22	-0.01	0.22	-0.01	0.05
			40%	0.01	0.12	0.02	0.15	0.00	0.06	0.00	0.16	0.00	0.17	-0.01	0.07
	CondTree		0%												
		m	10%	0.11	0.26	0.05	0.61	0.04	0.22	0.01	0.51	-0.05	0.69	-0.16	0.62
			20%	0.14	0.32	0.07	0.30	0.06	0.24	0.04	0.48	-0.01	0.43	-0.19	0.43
			30%	0.13	0.28	0.08	0.30	0.03	0.17	0.02	0.39	-0.01	0.45	0.01	0.21
			40%	0.12	0.28	0.08	0.26	0.00	0.12	0.03	0.42	0.00	0.43	0.00	0.14
		2	10%	0.02	0.28	0.00	0.33	-0.01	0.24	-0.02	0.38	-0.03	0.42	-0.02	0.27
			20%	0.04	0.28	0.00	0.33	0.00	0.21	-0.04	0.45	-0.09	0.60	-0.01	0.20
			30%	0.02	0.26	0.01	0.29	-0.01	0.19	-0.04	0.40	-0.06	0.44	0.00	0.09
			40%	0.03	0.28	0.01	0.31	-0.01	0.18	-0.04	0.39	-0.04	0.38	-0.02	0.29
	CART		0%												
		m	10%	0.07	0.40	0.03	0.33	-0.04	0.44	-0.05	0.53	-0.09	0.55	-0.13	0.55
			20%	0.06	0.41	0.03	0.41	0.16	0.28	-0.11	0.54	-0.12	0.59	-0.16	0.51
			30%	0.10	0.38	0.04	0.37	0.07	0.37	-0.10	0.65	-0.13	0.58	-0.09	0.54
			40%	0.11	0.38	0.06	0.35	-0.01	0.42	-0.10	0.64	-0.10	0.59	-0.16	0.52
		2	10%	0.02	0.26	-0.03	0.33	-0.05	0.40	-0.05	0.39	-0.08	0.42	-0.11	0.50
			20%	0.02	0.28	-0.04	0.40	0.00	0.24	-0.06	0.41	-0.15	0.65	-0.04	0.33
			30%	0.02	0.34	-0.02	0.37	-0.02	0.46	-0.09	0.62	-0.15	0.62	-0.06	0.41
			40%	0.03	0.26	0.01	0.30	-0.01	0.21	-0.09	0.54	-0.13	0.58	-0.08	0.44

Table C.14: Summary of mean MSPE values for the Birthweight dataset. Techniques with prior kNN imputation could not be fitted since this dataset contains categorical predictors. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 8 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 7$) and only $p/3$ variables with missing values. N/I stands for "not implemented".

		Missing		Surrogates			Median/mode			Prox. Matrix			MICE			MIST		
Data	Technique	# Var.	%	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
Birth weight	CondRF		0%	452584.3	452584.3	452349.3	452584.3	452584.3	452349.3	452584.3	452584.3	452349.3	452584.3	452584.3	452349.3	452584.3	452584.3	452349.3
		m	10%	459813.2	458992.8	460798.3	460084.9	458842.1	456592.0	455864.1	454975.1	454021.8	455948.7	455580.4	455251.7	457636.4	456398.5	457278.8
			20%	470114.8	470404.5	478195.6	467515.1	467241.1	475207.1	460091.9	459926.5	475636.0	460952.3	460268.7	469403.0	462783.4	461678.9	472087.5
			30%	482392.0	481917.0	492959.4	477648.5	474148.1	496061.5	469665.0	466360.6	500344.1	467393.9	465472.0	485580.4	469770.2	467643.0	488075.3
			40%	495339.9	492056.7	502489.0	487839.4	483971.3	503222.9	477049.5	473259.6	508027.8	474596.4	470321.5	496365.5	476550.7	473110.8	495810.1
		3	10%	455552.5	455768.8	455229.1	455279.8	455451.4	453641.1	453961.3	454047.9	452832.4	454049.1	454449.9	452611.6	454312.0	454960.0	453520.0
			20%	459479.2	460493.0	462969.7	457503.2	458951.3	461101.7	454927.1	456868.1	460182.8	455058.2	456325.9	458050.3	455504.3	457254.0	458861.5
			30%	465069.8	464313.9	468813.0	461167.3	460616.6	467871.2	458286.4	458454.7	466690.4	458022.9	457405.1	462856.6	458638.5	458629.0	464084.0
			40%	469347.9	469171.4	474591.6	465090.5	464801.6	472121.7	462060.3	462456.3	468462.3	459197.7	459908.0	466857.9	460931.9	461054.4	467954.3
	CondTree		0%	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7	517320.7
		m	10%	516077.1	516714.5	515440.7	522882.1	521478.3	517123.5	513178.7	512893.4	510503.9	506521.2	505691.9	506135.5	509433.3	509088.7	510155.9
			20%	517041.0	520764.8	515499.1	527386.1	525012.5	523624.2	516440.4	517922.0	516052.8	509498.9	507979.2	510297.9	510890.7	509988.7	510701.2
			30%	523261.6	526201.1	520818.0	531561.9	528104.6	523539.2	521859.6	518856.1	518815.3	516110.8	515540.8	516876.2	516374.4	514333.6	518068.3
			40%	527684.5	525408.3	520307.0	532709.7	533196.8	522524.6	526562.4	523704.1	523954.2	521850.2	516941.5	521592.5	520270.0	516899.1	520359.4
		3	10%	516856.9	516902.3	516024.1	518702.3	520081.8	516956.0	515554.3	518056.7	514174.2	510059.5	511859.1	509915.7	512391.3	513461.1	511655.5
			20%	519470.9	520737.1	519686.2	521794.6	519993.4	520140.0	516544.2	518265.1	515753.2	511821.1	512334.3	510223.2	513075.1	512876.2	512320.4
			30%	522358.9	521951.5	522279.9	522550.0	521479.4	522506.9	518112.1	519440.9	519052.6	513364.3	512225.0	512324.6	513928.0	514382.9	514497.6
			40%	522458.0	523250.5	525096.0	524867.3	521677.6	521520.6	520393.3	523142.9	518793.1	514623.2	516121.2	517098.8	514469.5	515626.5	517474.6
	CART		0%	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9	523200.9
		m	10%	525344.7	525841.1	511444.4	533349.2	531027.9	517028.2	538200.7	534694.7	518118.6	495420.1	498458.5	488186.1	498178.3	500559.6	489973.0
			20%	527392.1	534232.5	544938.0	539896.3	542494.9	565500.1	555437.5	550276.0	583683.7	491372.9	494295.0	499879.1	494249.0	496806.7	508755.8
			30%	538780.1	532592.9	576179.0	548640.1	545449.7	604052.0	575947.6	563886.9	626916.9	497888.3	501050.8	514442.3	496874.3	499612.7	525536.0
			40%	542070.8	535502.0	578406.1	556261.3	553376.2	615713.7	588939.0	576184.4	635028.6	502318.0	499256.4	522998.4	499064.6	499952.2	524674.5
		3	10%	521682.3	523984.0	518219.5	527339.3	525864.4	521884.9	531308.8	530976.4	520138.1	507545.5	507302.6	502864.7	509911.2	507665.4	504270.6
			20%	524738.8	528971.0	535673.3	531793.1	537098.9	545358.9	532804.8	543160.4	548015.0	504150.8	506618.7	507967.7	506479.7	509156.0	512757.8
			30%	527032.8	528807.8	535861.7	536272.5	533979.9	554750.5	538329.0	547277.6	558336.8	506277.1	504458.2	513819.7	507429.0	507063.8	516095.4
			40%	529764.0	531217.4	544492.2	539481.6	540036.5	561837.9	551281.1	552781.4	561971.2	507973.4	507071.9	515379.1	510390.5	507760.1	518194.2
	RF		0%	504581.2	504581.2	504759.5	504581.2	504581.2	504759.5	504581.2	504581.2	504759.5	504581.2	504581.2	504759.5	504581.2	504581.2	504759.5
		m	10%	N/I	N/I	N/I	506671.5	506231.2	495833.9	504319.0	503369.4	484744.4	494028.5	494796.4	481970.7	496403.7	496882.5	482940.6
			20%	N/I	N/I	N/I	513642.2	513171.1	510777.1	505383.7	505908.2	504587.1	488947.0	490805.9	478863.2	492460.1	492493.1	487043.3
			30%	N/I	N/I	N/I	521534.8	517847.8	553346.0	517679.7	509804.8	554084.8	492216.2	492463.6	497852.5	491557.6	493090.6	514925.1
			40%	N/I	N/I	N/I	531205.7	524561.3	563622.7	524572.9	518657.6	558955.6	495534.0	492431.8	503813.5	492282.9	492612.9	513650.7
		3	10%	N/I	N/I	N/I	506343.2	507041.7	501345.0	504239.0	504644.2	496557.9	500467.9	500839.9	494567.0	501709.0	502146.3	495308.6
			20%	N/I	N/I	N/I	507023.0	507638.1	508202.2	503577.6	504245.1	501625.3	497271.4	498123.3	494610.1	499202.4	499567.0	498343.4
			30%	N/I	N/I	N/I	509959.0	508533.1	523810.8	504174.7	503836.4	517206.8	497296.9	495696.1	499699.5	498179.4	497288.3	508573.4
			40%	N/I	N/I	N/I	514054.2	512123.1	528201.7	508266.3	507980.1	521149.4	497495.6	496951.1	499911.3	499185.5	497370.2	507960.7
	Bagging		0%	469856.8	469926.2	469451.7	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	499907.9	493056.7	494686.4	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	542382.7	535212.5	560208.4	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	599570.4	571437.5	608223.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	782016.0	681922.1	872861.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		3	10%	477541.8	477184.5	475940.5	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	492257.4	489998.0	497194.7	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	508186.7	503863.1	517245.8	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	532981.8	526042.3	545106.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	460448.6	460443.2	460549.6	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	463687.5	462866.9	464962.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	468936.9	469742.9	476940.5	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	477626.4	477626.9	489741.8	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	487727.6	484238.1	498541.2	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		3	10%	461370.1	462029.9	461439.6	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	463671.9	465040.2	468152.8	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	467362.7	466800.9	473146.8	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	470411.9	470481.3	476965.2	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	m	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	490130.2	490812.3	477936.3
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	489616.8	489928.9	488850.7
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	492411.0	489847.1	513976.6
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	494961.3	492170.8	514146.1
	Boot. RF	3	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	492230.9	491771.8	486736.9
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	491033.7	491528.9	491588.1
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	492891.6	490494.8	501376.8
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	491322.4	492009.9	503441.4

Table C.15: Summary of standard error (SE) estimates of each of the MSPE estimates for the Birthweight dataset. Techniques with prior kNN imputation could not be fitted since this dataset contains categorical predictors. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 8 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 7$) and only $p/3$ variables with missing values. N/I stands for "not implemented".

		Missing		Surrogates			Median/mode			Prox. Matrix			MICE			MIST		
Data	Technique	# Var.	%	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR	MCAR	MAR	MNAR
Birth	CondRF		0%	93781.7	93781.7	93451.8	93781.7	93781.7	93451.8	93781.7	93781.7	93451.8	93781.7	93781.7	93451.8	93781.7	93781.7	93451.8
weight		m	10%	95510.2	95361.1	94661.6	94626.5	94326.8	93461.5	93904.8	93349.3	92241.3	94715.2	94595.4	93567.2	95009.1	94337.7	93864.1
			20%	97077.3	97519.1	96666.3	95302.3	95046.1	96329.2	92591.9	93337.9	96132.9	95553.4	94652.9	95288.1	94774.1	94980.9	95600.9
			30%	98658.0	99114.0	99256.0	98894.5	96855.4	99386.0	96345.9	94868.8	101150.0	96050.0	97166.7	98433.6	96651.9	96582.6	98271.9
			40%	100124.0	100590.7	100328.1	98574.4	97463.0	100696.1	96409.7	96921.7	103229.2	97759.5	96436.6	100768.5	96836.1	96903.3	99961.6
		3	10%	94112.6	94349.1	94700.7	93524.6	93727.0	93641.3	93235.8	93178.4	93417.9	93550.4	94152.4	93679.4	93625.0	94361.3	94007.3
			20%	95580.9	95126.4	95994.3	94175.7	94247.8	94549.7	93360.4	93613.4	94605.1	94165.9	94302.6	95020.8	94220.6	94186.9	94653.4
			30%	98160.8	96641.6	97984.3	96111.8	94740.9	96233.4	94804.5	93927.2	96486.8	95309.4	95079.3	96249.8	96044.2	95304.9	96805.4
			40%	97996.1	97902.2	99634.3	96280.9	95486.0	97465.1	95627.7	95178.8	96211.3	95977.1	96101.4	97459.3	95644.4	95297.3	97518.7
	CondTree		0%	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9	97991.9
		m	10%	100267.7	102096.2	98426.2	101601.4	102732.9	97404.8	99270.3	101319.4	98648.7	100134.4	98922.7	99214.5	99963.6	98431.2	97698.2
			20%	101441.2	102001.2	102075.5	101721.2	101365.0	99285.9	100911.2	101783.0	103398.8	101176.7	99153.6	101691.3	99949.9	99444.8	100256.0
			30%	103184.1	102018.8	104909.3	104326.7	102481.6	99530.3	105535.6	102018.8	104456.0	101761.0	102925.4	102298.3	102099.0	102124.3	101538.2
			40%	104676.5	103685.6	103124.4	102040.9	105364.5	99194.4	105512.8	104340.5	105782.4	102585.9	102194.7	102597.5	102774.1	101754.7	102096.1
		3	10%	99637.9	98749.3	99023.4	99679.8	98302.1	97578.4	100181.3	98611.2	98250.7	98515.1	98259.0	97052.5	98126.5	97739.1	98460.0
			20%	101031.4	99652.2	101002.7	98649.2	99400.7	98828.3	99835.5	99096.2	100307.3	98879.8	98979.6	99399.7	98264.4	99452.2	98962.8
			30%	105552.5	103367.4	103535.0	102950.4	102511.4	99843.1	104120.0	103068.6	100321.6	101635.7	101603.0	99838.4	100028.9	102078.5	100836.8
			40%	102518.5	100549.9	105550.3	101687.8	100312.7	100404.6	102159.8	103394.4	99278.4	99513.5	99440.1	102254.4	98969.3	98333.9	101330.3
	CART		0%	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3	118534.3
		m	10%	116809.1	117853.0	117315.9	118253.0	118973.7	114921.8	120660.3	122270.2	113702.2	107606.1	107803.9	105023.7	106191.1	108171.2	103306.2
			20%	113630.5	119074.1	114703.0	119234.0	118804.1	121157.2	122367.7	122529.6	133543.7	107464.9	106131.9	104997.7	103508.8	104874.1	105370.8
			30%	121617.5	117426.1	124033.5	122910.7	121471.5	123072.9	130073.3	122981.7	139768.2	106253.1	106928.7	106489.5	105417.6	104863.2	107050.9
			40%	119768.4	115009.4	121172.4	125089.2	120985.3	127278.3	130080.0	133103.5	138228.2	105752.0	106002.3	110231.2	103336.2	103331.7	105908.2
		3	10%	117670.2	118257.7	115105.6	115900.1	117096.5	117782.3	116666.8	118058.5	117351.4	110718.8	111361.5	110110.1	110940.7	111165.5	112382.6
			20%	119413.6	113506.5	117073.8	117341.0	113323.1	121812.9	119185.7	118858.1	122918.4	109814.3	107992.2	109602.0	110963.6	107071.1	109915.1
			30%	118079.3	118715.9	117537.1	120583.5	116181.0	117864.8	119909.9	121197.5	119224.7	111911.4	110368.3	111548.5	113571.4	109893.4	111871.8
			40%	120009.0	121241.4	122212.1	119767.5	122535.7	123749.0	125481.1	124354.7	128790.7	112198.0	111243.7	113274.5	111765.8	112115.1	112093.1
	RF		0%	97319.4	97319.4	97564.5	97319.4	97319.4	97564.5	97319.4	97319.4	97564.5	97319.4	97319.4	97564.5	97319.4	97319.4	97564.5
		m	10%	N/I	N/I	N/I	98553.5	98023.1	93931.6	100787.5	99062.8	94237.7	98746.1	97978.5	95767.4	98720.5	98097.4	94022.0
			20%	N/I	N/I	N/I	99856.2	99802.8	100168.3	100110.2	101154.1	103430.1	99689.3	99789.8	97002.0	97117.1	98539.3	96942.9
			30%	N/I	N/I	N/I	104648.5	102953.9	108031.9	108190.1	106078.4	113258.1	101743.8	103376.5	100642.7	99578.5	100318.3	100842.3
			40%	N/I	N/I	N/I	104439.2	103139.2	114723.5	106064.5	106781.0	119498.5	102067.7	100146.2	107009.5	97003.1	98218.3	106794.7
		3	10%	N/I	N/I	N/I	98400.0	98758.1	95771.5	98166.1	98481.9	96382.7	97647.0	97596.0	96471.0	98016.8	97933.3	96674.4
			20%	N/I	N/I	N/I	98403.8	97463.8	98128.1	99762.4	98624.8	99349.3	99462.3	97998.1	97503.3	97669.0	96769.2	97044.8
			30%	N/I	N/I	N/I	99466.4	98568.1	105054.1	101573.7	98724.0	104927.2	100325.3	99759.1	101792.1	100629.4	98060.7	103524.4
			40%	N/I	N/I	N/I	103558.0	101307.6	102676.7	104692.5	101893.5	104292.5	101504.4	101313.3	102043.1	101046.8	99396.1	102507.5
	Bagging		0%	98471.2	98326.5	98417.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	109096.5	106912.6	108058.3	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	120419.2	117040.8	126035.1	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	154982.3	127705.2	163402.2	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	418046.5	302205.5	700133.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		3	10%	102920.4	103787.1	102653.5	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	110954.5	107398.5	108747.3	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	113654.3	109596.5	115762.4	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	117435.9	118901.8	119415.8	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	93576.6	93574.1	93604.8	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		m	10%	95965.7	95619.4	95304.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	96680.0	97744.1	97207.7	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	97862.8	99402.6	99716.7	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	99125.9	99520.0	100382.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		3	10%	94422.8	94391.3	94712.9	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	95846.9	95959.4	96649.6	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	98522.1	96957.7	98332.7	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	97882.2	98505.4	99742.0	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	m	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	98924.9	100034.8	95438.6
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	99212.9	99292.8	99584.4
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	101508.8	100370.5	103139.5
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	98440.4	99845.0	105183.3
	Boot. RF	3	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	98311.0	99949.3	97186.9
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	99233.9	99401.2	97928.9
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	100029.7	98802.6	103666.9
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	101262.3	98819.1	102677.9

Table C.16: Summary of mean relative improvement values with an imputation strategy compared to surrogate decisions through different missing data scenarios for the Birthweight dataset. Only CondRF, CondTree and CART were taken into account for these comparisons (because RF implementation in R -randomForest()- cannot be fitted on incomplete data). In addition, kNN imputation could not be implemented since this dataset contains categorical predictors. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: all $p = 8 = m$ variables with missing values (for MAR pattern: $m = p - 1 = 7$) and only $p/3$ variables with missing values.

				Median/mode						Prox. Matrix						MICE					
		Missing		MCAR		MAR		MNAR		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Birthweight	CondRF		0%																		
		m	10%	0.00	0.03	0.00	0.03	0.01	0.03	0.01	0.04	0.01	0.03	0.01	0.04	0.01	0.02	0.01	0.02	0.01	0.02
			20%	0.00	0.04	0.01	0.04	0.01	0.04	0.02	0.05	0.02	0.05	0.00	0.05	0.02	0.03	0.02	0.03	0.02	0.03
			30%	0.01	0.05	0.01	0.05	-0.01	0.05	0.02	0.06	0.03	0.06	-0.02	0.06	0.03	0.04	0.03	0.04	0.01	0.03
			40%	0.01	0.05	0.01	0.05	0.00	0.06	0.03	0.07	0.04	0.07	-0.01	0.07	0.04	0.04	0.04	0.04	0.01	0.04
		3	10%	0.00	0.02	0.00	0.02	0.00	0.02	0.00	0.02	0.00	0.02	0.00	0.02	0.00	0.01	0.00	0.01	0.01	0.01
			20%	0.00	0.03	0.00	0.03	0.00	0.03	0.01	0.04	0.01	0.04	0.01	0.04	0.01	0.02	0.01	0.02	0.01	0.02
			30%	0.01	0.03	0.01	0.03	0.00	0.04	0.01	0.05	0.01	0.04	0.00	0.05	0.01	0.03	0.01	0.03	0.01	0.03
			40%	0.01	0.04	0.01	0.04	0.00	0.04	0.01	0.05	0.01	0.05	0.01	0.05	0.02	0.03	0.02	0.03	0.02	0.03
	CondTree		0%																		
		m	10%	-0.02	0.08	-0.01	0.08	-0.01	0.08	0.00	0.08	0.00	0.08	0.01	0.09	0.02	0.06	0.02	0.06	0.02	0.06
			20%	-0.02	0.09	-0.01	0.08	-0.02	0.09	0.00	0.10	0.00	0.10	0.00	0.08	0.01	0.07	0.02	0.07	0.01	0.07
			30%	-0.02	0.08	-0.01	0.08	-0.01	0.09	0.00	0.10	0.01	0.10	0.00	0.09	0.01	0.06	0.02	0.06	0.01	0.06
			40%	-0.01	0.08	-0.02	0.09	-0.01	0.10	0.00	0.10	0.00	0.10	-0.01	0.10	0.01	0.05	0.01	0.06	0.00	0.06
		3	10%	0.00	0.05	-0.01	0.06	0.00	0.06	0.00	0.06	0.00	0.06	0.00	0.06	0.01	0.05	0.01	0.05	0.01	0.05
			20%	-0.01	0.07	0.00	0.07	0.00	0.07	0.00	0.08	0.00	0.08	0.01	0.07	0.01	0.05	0.01	0.06	0.02	0.05
			30%	0.00	0.07	0.00	0.06	0.00	0.07	0.00	0.09	0.00	0.09	0.00	0.07	0.02	0.05	0.02	0.05	0.02	0.05
			40%	-0.01	0.07	0.00	0.07	0.00	0.07	0.00	0.09	0.00	0.09	0.01	0.07	0.01	0.05	0.01	0.05	0.01	0.05
	CART		0%																		
		m	10%	-0.02	0.13	-0.02	0.13	-0.02	0.14	-0.03	0.15	-0.03	0.15	-0.03	0.17	0.05	0.12	0.04	0.12	0.03	0.12
			20%	-0.03	0.14	-0.03	0.14	-0.05	0.16	-0.07	0.19	-0.05	0.19	-0.08	0.19	0.06	0.13	0.06	0.13	0.07	0.12
			30%	-0.03	0.15	-0.04	0.16	-0.06	0.17	-0.09	0.21	-0.08	0.20	-0.11	0.21	0.06	0.14	0.05	0.14	0.10	0.13
			40%	-0.04	0.16	-0.04	0.16	-0.08	0.16	-0.11	0.22	-0.09	0.22	-0.12	0.22	0.06	0.15	0.06	0.14	0.09	0.14
		3	10%	-0.02	0.10	-0.01	0.10	-0.01	0.10	-0.03	0.13	-0.02	0.12	-0.01	0.11	0.02	0.09	0.03	0.09	0.03	0.08
			20%	-0.02	0.10	-0.02	0.11	-0.02	0.12	-0.03	0.15	-0.04	0.15	-0.03	0.14	0.03	0.11	0.04	0.10	0.05	0.11
			30%	-0.02	0.12	-0.02	0.12	-0.04	0.13	-0.03	0.15	-0.05	0.17	-0.05	0.15	0.03	0.11	0.04	0.11	0.03	0.11
			40%	-0.03	0.12	-0.02	0.12	-0.04	0.13	-0.05	0.18	-0.05	0.17	-0.04	0.17	0.03	0.12	0.04	0.12	0.05	0.12

				MIST					
		Missing		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD
Birthweight	CondRF		0%						
		m	10%	0.00	0.02	0.01	0.02	0.01	0.02
			20%	0.01	0.03	0.02	0.03	0.01	0.03
			30%	0.03	0.03	0.03	0.03	0.01	0.03
			40%	0.04	0.04	0.04	0.04	0.01	0.04
		3	10%	0.00	0.01	0.00	0.01	0.00	0.01
			20%	0.01	0.02	0.01	0.02	0.01	0.02
			30%	0.01	0.03	0.01	0.03	0.01	0.03
			40%	0.02	0.03	0.02	0.03	0.01	0.03
	CondTree		0%						
		m	10%	0.01	0.06	0.01	0.07	0.01	0.06
			20%	0.01	0.07	0.02	0.06	0.01	0.06
			30%	0.01	0.06	0.02	0.06	0.00	0.06
			40%	0.01	0.05	0.01	0.06	0.00	0.06
		3	10%	0.01	0.05	0.01	0.05	0.01	0.04
			20%	0.01	0.06	0.01	0.06	0.01	0.05
			30%	0.01	0.06	0.01	0.05	0.01	0.05
			40%	0.01	0.06	0.01	0.05	0.01	0.05
	CART		0%						
		m	10%	0.04	0.11	0.04	0.12	0.03	0.11
			20%	0.05	0.13	0.06	0.13	0.06	0.12
			30%	0.07	0.14	0.05	0.14	0.08	0.13
			40%	0.07	0.14	0.05	0.14	0.08	0.13
		3	10%	0.02	0.09	0.03	0.09	0.02	0.08
			20%	0.03	0.11	0.03	0.11	0.04	0.10
			30%	0.03	0.11	0.03	0.11	0.03	0.11
			40%	0.03	0.12	0.04	0.12	0.04	0.11

Table C.17: Summary of mean MSPE values for the simulated dataset. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: 8 variables with missing values and only 8/3 variables with missing values. N/I stands for "not implemented".

		Missing		Surrogates			Median/mode			Prox. Matrix			MICE			MIST			kNN		
Data	Technique	# Var.	%	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR
Simu	CondRF		0%	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80	76.80
lated		8	10%	78.92	91.08	380.46	79.51	78.41	129.73	76.46	77.17	101.88	76.73	77.77	99.86	76.85	76.80	98.00	77.39	77.07	96.44
			20%	83.23	89.96	467.49	87.39	80.18	149.66	76.99	76.93	111.75	76.80	79.95	123.53	77.76	76.74	107.73	78.07	77.39	105.52
			30%	87.54	82.95	234.38	97.40	80.90	168.46	76.59	77.52	119.40	76.72	81.37	174.69	78.12	77.57	113.89	80.22	78.16	113.79
			40%	93.17	82.89	204.51	101.08	81.73	176.84	76.55	77.18	128.49	76.61	79.76	232.30	79.56	78.05	117.55	84.74	78.00	114.79
		3	10%	77.15	76.98	76.68	76.73	77.31	76.32	76.79	76.98	76.29	76.85	76.77	77.06	76.75	76.77	76.27	76.89	76.64	76.51
			20%	76.70	76.73	76.54	76.89	76.64	76.16	76.62	76.83	76.56	76.88	77.39	78.18	76.88	76.40	76.51	77.03	76.14	76.58
			30%	76.82	77.13	76.33	77.14	77.11	76.61	77.00	76.91	76.51	76.69	77.48	80.37	76.91	76.94	76.32	77.44	77.24	76.75
			40%	77.20	76.95	76.69	76.71	77.02	76.46	76.49	77.03	76.35	76.82	78.12	83.69	77.07	76.73	76.35	77.21	76.79	76.15
	CondTree		0%	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38	77.38
		8	10%	77.48	91.59	370.98	85.15	82.87	115.03	77.17	81.28	109.25	76.64	77.13	111.08	75.75	79.08	100.36	78.56	80.76	104.73
			20%	86.37	92.52	520.88	93.72	80.82	126.51	77.94	79.09	120.45	75.65	75.67	127.17	71.71	76.54	113.61	81.12	78.70	114.26
			30%	103.28	85.13	152.61	116.46	84.40	129.60	79.32	77.37	126.61	74.81	75.29	178.93	72.83	75.68	118.31	84.87	78.57	120.39
			40%	112.46	85.55	125.16	123.71	86.39	130.13	78.62	79.15	130.13	73.62	76.80	250.56	70.31	78.16	122.52	91.98	81.11	124.09
		3	10%	78.62	77.74	78.68	78.75	77.74	78.68	77.30	78.31	78.66	76.23	77.79	76.25	76.69	77.43	78.65	77.44	78.16	78.66
			20%	79.10	78.32	78.72	79.25	78.37	78.73	77.75	79.63	78.72	76.50	77.48	78.69	78.46	77.94	78.70	79.08	78.67	78.71
			30%	77.93	78.68	78.80	77.99	78.72	78.81	77.24	78.74	78.79	76.46	76.58	78.42	77.15	78.31	78.78	77.27	78.72	78.79
			40%	78.55	78.61	78.84	78.56	78.65	78.88	77.32	78.42	78.87	75.59	77.65	83.96	78.23	78.49	78.83	78.23	78.44	78.84
	CART		0%	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53	98.53
		8	10%	110.50	114.10	151.51	121.05	101.36	230.11	99.38	100.86	156.62	91.91	89.04	141.64	91.20	93.94	124.52	99.43	100.54	131.92
			20%	138.88	123.57	165.65	144.64	110.64	192.82	98.79	102.01	166.51	91.22	90.65	233.70	87.82	95.60	143.33	103.13	103.81	150.18
			30%	153.40	129.81	170.10	155.63	113.84	213.45	98.83	100.52	178.35	86.96	98.19	299.51	83.00	92.81	152.80	103.67	100.88	156.73
			40%	163.66	131.48	171.95	163.79	113.44	211.95	98.94	99.56	194.64	86.97	93.93	412.76	80.36	92.29	160.73	110.47	101.97	161.83
		3	10%	99.14	98.85	100.38	100.13	98.82	100.38	98.51	99.44	109.19	96.47	95.54	92.50	97.42	97.34	98.17	99.07	99.33	98.33
			20%	97.97	99.73	100.49	98.08	99.60	100.49	98.92	98.70	100.61	94.03	95.24	91.32	93.07	98.98	99.23	98.81	99.32	98.47
			30%	100.64	100.37	100.57	100.16	100.08	101.94	98.20	98.31	100.53	95.44	96.90	93.88	94.38	98.25	100.41	99.36	100.75	100.40
			40%	99.51	100.53	100.61	99.51	100.56	100.53	99.19	100.89	101.16	94.93	95.71	101.37	96.80	99.01	100.43	99.04	101.01	100.43
	RF		0%	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78	21.78
		8	10%	N/I	N/I	N/I	22.53	23.02	129.31	21.72	24.45	72.96	21.80	22.35	123.14	22.11	22.54	46.34	21.90	30.21	52.76
			20%	N/I	N/I	N/I	23.23	23.71	124.17	21.89	23.26	88.07	21.96	23.91	355.72	22.21	22.98	55.55	22.54	26.81	65.87
			30%	N/I	N/I	N/I	24.82	24.36	126.39	21.69	23.08	88.23	21.90	38.16	644.79	22.44	23.35	61.13	24.21	28.48	69.74
			40%	N/I	N/I	N/I	26.85	24.32	123.56	21.95	22.73	88.38	21.96	39.15	939.87	22.86	23.70	64.39	27.07	27.61	67.38
		3	10%	N/I	N/I	N/I	22.42	21.93	22.56	21.73	23.55	23.66	21.76	22.25	22.43	21.90	22.18	22.71	21.91	22.27	22.22
			20%	N/I	N/I	N/I	22.30	22.43	22.37	21.63	22.28	22.17	21.89	21.95	23.56	21.93	22.22	22.49	21.92	22.45	27.76
			30%	N/I	N/I	N/I	22.11	22.05	22.52	21.76	21.98	22.62	21.86	22.21	25.06	22.06	22.04	22.75	21.97	22.25	26.81
			40%	N/I	N/I	N/I	22.32	22.26	22.42	21.67	22.09	22.56	21.87	22.70	28.41	22.01	22.13	22.58	22.31	22.61	23.63
	Bagging		0%	55.30	55.30	55.30	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		8	10%	102.74	373.06	432.45	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	238.96	597.39	680.83	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	651.19	868.40	901.43	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	1454.06	1187.34	1112.51	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		3	10%	71.25	258.40	412.10	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	90.43	402.46	652.18	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	132.33	551.55	864.46	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	200.28	725.34	1069.24	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	59.54	59.54	59.54	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		8	10%	61.26	66.60	360.22	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	64.18	64.49	384.24	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	67.10	61.37	96.74	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	72.66	62.28	83.99	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		3	10%	60.20	59.88	59.79	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	60.03	60.05	59.70	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	60.29	60.01	59.94	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	60.41	60.10	60.12	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	8	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	25.85	27.81	50.77	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	24.25	27.04	59.25	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	26.49	26.32	63.93	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	26.33	28.46	66.51	N/I	N/I	N/I
	Boot. RF	3	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	24.38	26.77	24.95	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	25.66	26.96	26.06	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	25.05	26.08	26.75	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	27.13	26.41	25.28	N/I	N/I	N/I

Table C.18: Summary of standard error (SE) estimates of each of the MSPE estimates for the simulated dataset. Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: 8 variables with missing values and only 8/3 variables with missing values. N/I stands for "not implemented".

		Missing		Surrogates			Median/mode			Prox. Matrix			MICE			MIST			kNN		
Data	Technique	# Var.	%	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR	MCAR	MAR	NMAR
Simu	CondRF		0%	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25	175.25
lated		8	10%	175.12	184.44	315.20	174.17	176.91	210.50	174.79	176.96	188.89	175.64	175.40	188.74	175.27	175.64	187.74	176.08	175.77	185.41
			20%	177.85	183.49	336.28	174.62	176.57	220.45	175.70	175.98	187.38	175.50	175.53	182.37	176.59	175.64	190.26	176.40	178.22	190.37
			30%	179.96	183.45	253.34	180.47	178.23	232.34	176.44	176.89	194.71	175.55	175.31	180.60	176.69	176.03	193.36	176.27	177.81	196.39
			40%	184.34	180.69	247.73	184.00	177.11	233.89	176.12	175.88	195.84	175.44	176.25	191.23	176.32	176.29	195.90	178.88	176.34	193.87
		3	10%	176.12	176.07	176.21	175.22	175.83	175.30	175.29	174.87	176.12	175.67	176.01	175.40	175.82	175.84	175.58	175.41	175.64	176.02
			20%	175.92	176.29	175.39	175.94	176.07	175.39	175.78	177.02	175.64	175.66	176.42	175.99	175.74	175.57	176.19	176.41	175.16	175.32
			30%	174.89	175.37	175.48	175.18	176.43	176.26	176.30	175.70	176.03	175.56	175.34	175.62	175.66	175.39	175.66	175.70	175.93	176.72
			40%	175.84	175.07	175.70	175.60	174.45	175.44	174.98	176.17	175.75	175.75	176.03	176.09	175.58	175.07	175.20	175.84	175.20	175.82
	CondTree		0%	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49	157.49
		8	10%	164.93	177.73	307.56	167.14	169.54	169.93	158.62	169.05	168.77	163.04	162.09	168.74	167.08	169.12	167.25	158.73	168.96	166.66
			20%	157.11	170.68	342.80	159.72	157.69	168.26	157.76	158.07	168.91	159.30	157.83	172.21	157.69	158.00	168.22	158.25	157.59	168.31
			30%	165.67	154.41	206.78	159.99	158.00	167.76	167.17	157.49	168.67	160.56	157.52	215.38	159.79	157.35	168.39	158.69	157.87	168.03
			40%	168.87	156.03	168.45	168.70	158.61	168.65	167.60	157.65	168.79	161.51	159.21	314.42	160.67	165.32	168.07	160.40	158.29	167.91
		3	10%	157.56	157.21	169.20	157.55	157.21	169.20	157.49	157.49	169.20	157.57	157.55	163.72	157.48	157.51	169.20	157.20	157.32	169.20
			20%	158.47	157.27	169.20	158.51	157.26	169.19	157.17	167.34	169.20	157.48	159.13	158.45	158.32	157.35	169.20	157.98	157.20	169.20
			30%	157.43	157.12	169.19	157.44	157.11	169.19	156.45	157.14	169.19	157.44	157.58	160.98	157.41	157.18	169.19	157.38	157.10	169.19
			40%	157.43	156.21	169.19	157.42	156.20	169.19	156.46	156.21	169.19	157.82	156.37	164.75	157.43	156.21	169.19	157.41	156.24	169.20
	CART		0%	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57	157.57
		8	10%	161.35	158.02	167.68	159.81	157.02	263.70	157.41	158.61	174.65	156.81	154.93	169.29	163.06	156.93	166.25	158.34	157.27	165.74
			20%	170.57	157.14	167.19	171.02	170.56	176.42	157.58	169.51	171.83	166.85	170.58	242.81	167.39	170.83	168.39	169.50	169.33	167.94
			30%	175.05	156.95	167.11	165.54	157.94	171.69	157.17	157.88	172.21	162.69	157.88	298.98	157.04	158.45	167.85	169.01	159.00	173.40
			40%	162.21	156.87	166.95	161.91	157.80	171.70	157.51	157.73	176.23	162.44	156.80	381.48	160.45	163.59	167.85	158.84	158.31	167.48
		3	10%	169.91	157.48	169.68	169.74	157.48	169.68	157.58	157.34	175.00	157.73	156.54	157.17	164.19	157.66	157.57	157.44	157.44	157.54
			20%	157.22	157.30	169.67	157.20	157.34	169.67	157.57	157.58	169.64	158.11	158.19	158.19	156.09	158.06	157.49	158.62	157.42	157.52
			30%	169.35	169.60	169.68	169.45	169.66	169.45	157.37	157.59	169.67	167.25	166.78	169.76	163.84	169.69	169.62	157.43	169.83	169.64
			40%	157.42	169.68	169.68	157.42	169.70	169.69	157.42	169.56	169.69	156.03	168.94	171.10	157.90	169.86	169.65	156.96	169.72	169.65
	RF		0%	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80	113.80
		8	10%	N/I	N/I	N/I	114.81	114.29	208.43	114.26	114.82	134.38	114.07	114.20	184.50	114.63	114.58	127.55	114.24	131.64	130.26
			20%	N/I	N/I	N/I	114.19	117.08	171.83	114.88	113.99	137.62	114.69	114.27	447.14	114.79	114.89	133.51	113.50	121.16	144.57
			30%	N/I	N/I	N/I	116.01	116.58	166.69	113.71	114.97	149.23	114.65	141.88	650.45	114.24	116.69	134.28	116.96	129.49	148.45
			40%	N/I	N/I	N/I	119.03	114.64	160.40	113.39	114.85	133.27	114.17	118.66	824.56	115.05	115.20	138.50	123.52	121.64	136.47
		3	10%	N/I	N/I	N/I	115.25	113.57	115.13	113.39	122.22	115.72	113.91	116.18	113.93	114.25	115.68	116.97	114.40	113.44	113.87
			20%	N/I	N/I	N/I	114.68	115.30	113.95	113.83	114.65	113.89	114.35	113.94	114.18	114.22	114.55	115.04	115.14	114.95	124.86
			30%	N/I	N/I	N/I	114.26	114.45	114.31	114.15	114.56	115.25	114.36	113.98	113.65	114.89	114.66	115.69	114.14	113.86	116.33
			40%	N/I	N/I	N/I	114.79	115.13	114.69	113.85	114.64	114.14	113.92	114.41	114.55	114.32	114.46	115.48	116.07	117.22	117.68
	Bagging		0%	155.36	155.36	155.36	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		8	10%	181.88	315.20	328.45	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	257.49	363.98	385.87	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	483.06	460.79	422.09	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	723.86	498.42	449.31	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		3	10%	173.55	291.63	323.67	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	176.76	339.34	377.00	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	225.48	422.89	414.69	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	254.71	537.55	444.39	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	CondBagging		0%	157.65	157.65	157.65	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		8	10%	160.55	162.17	308.70	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	164.84	160.58	306.33	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	165.18	157.89	167.93	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	166.63	157.51	164.10	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
		3	10%	159.52	158.41	158.53	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			20%	158.91	158.07	157.44	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			30%	157.76	158.40	158.82	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
			40%	157.31	157.92	158.73	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I
	Boot. RF	8	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	119.71	117.17	138.14	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	117.06	123.45	146.87	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	121.55	114.54	145.90	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	119.85	124.45	145.27	N/I	N/I	N/I
	Boot. RF	3	10%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	120.19	126.02	112.24	N/I	N/I	N/I
			20%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	126.24	125.72	120.87	N/I	N/I	N/I
			30%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	120.71	116.48	115.92	N/I	N/I	N/I
			40%	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	N/I	131.82	121.34	118.04	N/I	N/I	N/I

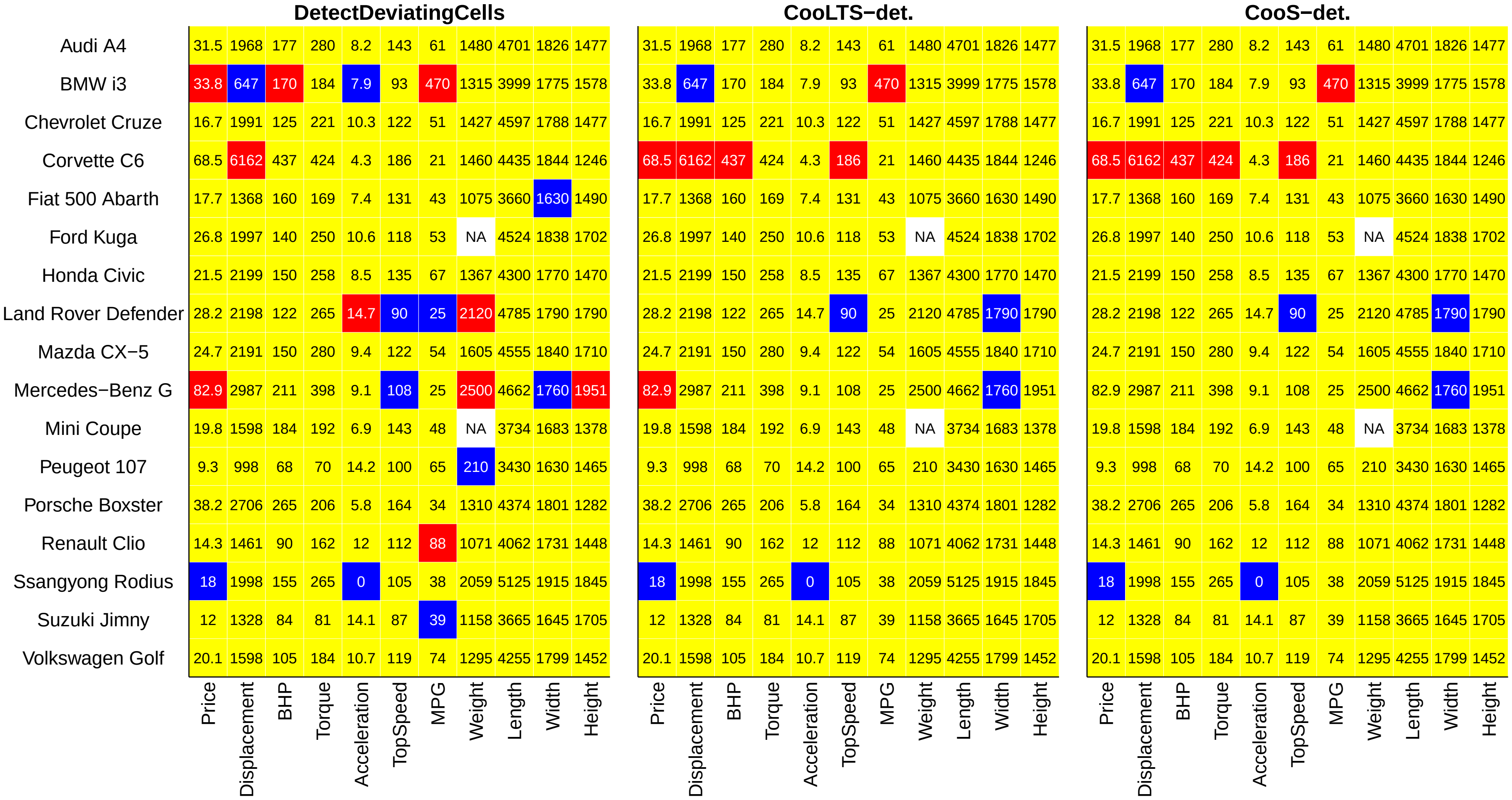
Table C.19: Summary of mean relative improvement values with an imputation strategy compared to surrogate decisions through different missing data scenarios for the simulated dataset. Only CondRF, CondTree and CART were taken into account for these comparisons (because RF implementation in R -randomForest()- cannot be fitted on incomplete data). Missing data was induced under MCAR, MAR and MNAR patterns at different fractions and following 2 schemes: 8 variables with missing values and only 8/3 variables with missing values.

				Median/mode						Prox. Matrix						MICE					
		Missing		MCAR		MAR		MNAR		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Simulated	CondRF		0%																		
		8	10%	-0.09	0.25	-0.11	1.43	0.72	0.14	0.05	0.15	-0.10	1.36	0.78	0.12	0.05	0.15	-0.17	1.84	0.79	0.13
			20%	-0.24	0.48	-0.05	1.00	0.71	0.17	0.12	0.25	0.07	0.54	0.79	0.13	0.11	0.27	-0.06	0.66	0.74	0.14
			30%	-0.38	0.56	-0.06	0.37	0.31	0.20	0.10	0.95	-0.05	0.72	0.48	0.28	0.09	0.89	-0.28	1.27	-0.15	1.16
			40%	-0.28	0.41	-0.11	0.38	0.15	0.13	0.22	0.52	-0.06	0.65	0.33	0.41	0.20	0.55	-0.18	0.86	-1.08	2.26
		3	10%	0.00	0.07	-0.01	0.08	-0.01	0.08	-0.01	0.11	-0.01	0.09	0.01	0.07	-0.01	0.10	0.00	0.10	-0.10	0.57
			20%	-0.01	0.08	-0.01	0.10	0.01	0.07	0.00	0.13	0.00	0.11	0.00	0.09	0.00	0.11	-0.02	0.20	-0.18	0.80
			30%	-0.01	0.07	0.00	0.08	0.00	0.08	0.01	0.10	0.01	0.08	0.00	0.07	0.01	0.11	-0.06	0.45	-0.29	0.92
			40%	0.00	0.08	-0.01	0.08	0.01	0.08	0.02	0.09	0.00	0.11	0.01	0.07	0.01	0.09	-0.05	0.23	-0.52	1.42
	CondTree		0%																		
		8	10%	-0.30	0.56	-0.10	0.79	0.66	0.24	-0.01	0.36	-0.14	0.99	0.68	0.24	0.02	0.36	-0.05	1.00	0.68	0.23
			20%	-0.33	0.77	-0.08	0.68	0.69	0.30	0.05	0.50	-0.03	0.63	0.70	0.30	0.07	0.57	0.07	0.64	0.69	0.32
			30%	-0.46	1.12	0.02	0.38	0.01	0.26	0.20	0.58	0.16	0.39	0.05	0.24	0.27	0.55	0.19	0.43	-0.63	1.97
			40%	-0.26	0.48	-0.09	1.20	-0.06	0.09	0.29	0.73	0.03	1.23	-0.06	0.13	0.32	0.76	0.09	1.33	-1.90	6.37
		3	10%	-0.01	0.04	0.00	0.02	0.00	0.00	0.02	0.17	-0.02	0.17	0.00	0.00	0.06	0.15	0.01	0.18	-0.01	0.35
			20%	0.00	0.02	0.00	0.01	0.00	0.01	-0.02	0.33	0.00	0.10	0.00	0.01	0.04	0.19	-0.01	0.36	-0.09	0.68
			30%	0.00	0.01	0.00	0.02	0.00	0.00	0.00	0.13	0.00	0.05	0.00	0.00	-0.05	0.79	0.05	0.18	-0.02	0.40
			40%	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.17	0.00	0.01	0.00	0.00	0.06	0.18	0.02	0.13	-0.22	0.96
	CART		0%																		
		8	10%	-0.16	0.34	0.12	0.30	-0.63	1.12	0.11	0.29	0.14	0.37	-0.06	0.41	0.21	0.29	0.25	0.29	0.05	0.41
			20%	-0.07	0.21	0.14	0.29	-0.20	0.32	0.33	0.27	0.22	0.31	0.00	0.28	0.41	0.27	0.34	0.26	-0.62	2.01
			30%	-0.07	0.26	0.14	0.30	-0.33	0.31	0.39	0.28	0.28	0.31	-0.08	0.41	0.50	0.28	0.28	0.37	-1.18	3.00
			40%	-0.02	0.16	0.14	0.36	-0.32	0.41	0.44	0.29	0.29	0.37	-0.18	0.59	0.54	0.28	0.33	0.39	-1.95	3.58
		3	10%	-0.02	0.11	0.00	0.02	0.00	0.00	-0.02	0.12	-0.01	0.08	-0.12	0.51	0.01	0.16	0.04	0.09	0.09	0.13
			20%	0.00	0.01	0.00	0.02	0.00	0.00	-0.01	0.13	0.01	0.10	0.00	0.03	0.07	0.11	0.07	0.11	0.09	0.21
			30%	0.01	0.05	0.00	0.03	-0.03	0.23	0.01	0.10	0.01	0.04	0.00	0.01	0.08	0.12	0.04	0.27	0.05	0.35
			40%	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.08	-0.02	0.18	-0.01	0.05	0.05	0.08	0.06	0.12	-0.05	0.51

				MIST						kNN					
		Missing		MCAR		MAR		MNAR		MCAR		MAR		MNAR	
Data	Technique	# Var.	%	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Simulated	CondRF		0%												
		8	10%	0.05	0.12	-0.08	1.32	0.79	0.12	0.03	0.17	-0.07	1.23	0.80	0.12
			20%	0.11	0.22	0.07	0.49	0.80	0.13	0.09	0.24	0.07	0.53	0.80	0.13
			30%	0.09	0.81	-0.04	0.61	0.54	0.19	-0.01	0.99	-0.02	0.44	0.55	0.20
			40%	0.19	0.43	-0.05	0.55	0.44	0.20	-0.01	0.67	-0.01	0.39	0.45	0.20
		3	10%	0.00	0.09	0.00	0.06	0.01	0.06	-0.01	0.11	0.00	0.09	-0.01	0.11
			20%	-0.01	0.09	0.01	0.08	0.00	0.05	-0.01	0.11	0.01	0.11	-0.01	0.09
			30%	0.01	0.05	0.01	0.07	0.00	0.05	0.00	0.09	0.00	0.09	0.00	0.08
			40%	0.00	0.06	0.00	0.06	0.01	0.07	-0.01	0.08	0.00	0.10	0.01	0.11
	CondTree		0%												
		8	10%	0.05	0.35	-0.05	0.94	0.72	0.22	-0.05	0.42	-0.12	0.98	0.71	0.22
			20%	0.19	0.51	0.05	0.58	0.73	0.26	0.02	0.54	-0.04	0.66	0.73	0.26
			30%	0.36	0.46	0.20	0.37	0.15	0.20	0.11	0.76	0.15	0.38	0.12	0.21
			40%	0.44	0.56	0.07	1.22	0.03	0.05	0.12	0.80	0.02	1.23	0.01	0.07
		3	10%	0.05	0.13	0.02	0.08	0.00	0.00	0.01	0.26	-0.02	0.17	0.00	0.00
			20%	0.02	0.15	0.01	0.10	0.00	0.00	-0.03	0.31	0.00	0.09	0.00	0.01
			30%	0.03	0.07	0.01	0.04	0.00	0.00	0.01	0.12	0.00	0.04	0.00	0.00
			40%	0.01	0.03	0.00	0.01	0.00	0.00	0.00	0.07	0.00	0.01	0.00	0.00
	CART		0%												
		8	10%	0.22	0.29	0.22	0.28	0.22	0.23	0.11	0.29	0.12	0.45	0.14	0.27
			20%	0.45	0.27	0.30	0.31	0.16	0.18	0.30	0.30	0.21	0.32	0.11	0.24
			30%	0.53	0.27	0.35	0.32	0.13	0.09	0.36	0.30	0.28	0.31	0.10	0.13
			40%	0.59	0.26	0.37	0.36	0.09	0.07	0.37	0.33	0.27	0.37	0.07	0.09
		3	10%	0.02	0.11	0.02	0.06	0.01	0.04	-0.03	0.15	-0.01	0.07	0.01	0.04
			20%	0.06	0.14	0.02	0.08	-0.01	0.18	-0.01	0.10	0.01	0.03	0.01	0.04
			30%	0.07	0.12	0.03	0.06	0.00	0.04	0.00	0.08	-0.01	0.12	0.00	0.02
			40%	0.04	0.07	0.02	0.06	0.00	0.03	0.00	0.06	-0.01	0.13	0.00	0.01

Appendix D

Additional Figures



DetectDeviatingCells

	Price	Displacement	BHP	Torque	Acceleration	TopSpeed	MPG	Weight	Length	Width	Height
Audi A4	31.5	1968	177	280	8.2	143	61	1480	4701	1826	1477
BMW i3	🟥 33.8	🟦 647	🟥 170	184	🟦 7.9	93	🟥 470	1315	3999	1775	1578
Chevrolet Cruze	16.7	1991	125	221	10.3	122	51	1427	4597	1788	1477
Corvette C6	68.5	🟥 6162	437	424	4.3	186	21	1460	4435	1844	1246
Fiat 500 Abarth	17.7	1368	160	169	7.4	131	43	1075	3660	🟦 1630	1490
Ford Kuga	26.8	1997	140	250	10.6	118	53	NA	4524	1838	1702
Honda Civic	21.5	2199	150	258	8.5	135	67	1367	4300	1770	1470
Land Rover Defender	28.2	2198	122	265	🟥 14.7	🟦 90	🟦 25	🟥 2120	4785	1790	1790
Mazda CX−5	24.7	2191	150	280	9.4	122	54	1605	4555	1840	1710
Mercedes−Benz G	🟥 82.9	2987	211	398	9.1	🟦 108	25	🟥 2500	4662	🟦 1760	🟥 1951
Mini Coupe	19.8	1598	184	192	6.9	143	48	NA	3734	1683	1378
Peugeot 107	9.3	998	68	70	14.2	100	65	🟦 210	3430	1630	1465
Porsche Boxster	38.2	2706	265	206	5.8	164	34	1310	4374	1801	1282
Renault Clio	14.3	1461	90	162	12	112	🟥 88	1071	4062	1731	1448
Ssangyong Rodius	🟦 18	1998	155	265	🟦 0	105	38	2059	5125	1915	1845
Suzuki Jimny	12	1328	84	81	14.1	87	🟦 39	1158	3665	1645	1705
Volkswagen Golf	20.1	1598	105	184	10.7	119	74	1295	4255	1799	1452

CooLTS−det.

	Price	Displacement	BHP	Torque	Acceleration	TopSpeed	MPG	Weight	Length	Width	Height
Audi A4	31.5	1968	177	280	8.2	143	61	1480	4701	1826	1477
BMW i3	33.8	🟦 647	170	184	7.9	93	🟥 470	1315	3999	1775	1578
Chevrolet Cruze	16.7	1991	125	221	10.3	122	51	1427	4597	1788	1477
Corvette C6	🟥 68.5	🟥 6162	🟥 437	424	4.3	🟥 186	21	1460	4435	1844	1246
Fiat 500 Abarth	17.7	1368	160	169	7.4	131	43	1075	3660	1630	1490
Ford Kuga	26.8	1997	140	250	10.6	118	53	NA	4524	1838	1702
Honda Civic	21.5	2199	150	258	8.5	135	67	1367	4300	1770	1470
Land Rover Defender	28.2	2198	122	265	14.7	🟦 90	25	2120	4785	🟦 1790	1790
Mazda CX−5	24.7	2191	150	280	9.4	122	54	1605	4555	1840	1710
Mercedes−Benz G	🟥 82.9	2987	211	398	9.1	108	25	2500	4662	🟦 1760	1951
Mini Coupe	19.8	1598	184	192	6.9	143	48	NA	3734	1683	1378
Peugeot 107	9.3	998	68	70	14.2	100	65	210	3430	1630	1465
Porsche Boxster	38.2	2706	265	206	5.8	164	34	1310	4374	1801	1282
Renault Clio	14.3	1461	90	162	12	112	88	1071	4062	1731	1448
Ssangyong Rodius	🟦 18	1998	155	265	🟦 0	105	38	2059	5125	1915	1845
Suzuki Jimny	12	1328	84	81	14.1	87	39	1158	3665	1645	1705
Volkswagen Golf	20.1	1598	105	184	10.7	119	74	1295	4255	1799	1452

CooS−det.

	Price	Displacement	BHP	Torque	Acceleration	TopSpeed	MPG	Weight	Length	Width	Height
Audi A4	31.5	1968	177	280	8.2	143	61	1480	4701	1826	1477
BMW i3	33.8	🟦 647	170	184	7.9	93	🟥 470	1315	3999	1775	1578
Chevrolet Cruze	16.7	1991	125	221	10.3	122	51	1427	4597	1788	1477
Corvette C6	🟥 68.5	🟥 6162	🟥 437	🟥 424	4.3	🟥 186	21	1460	4435	1844	1246
Fiat 500 Abarth	17.7	1368	160	169	7.4	131	43	1075	3660	1630	1490
Ford Kuga	26.8	1997	140	250	10.6	118	53	NA	4524	1838	1702
Honda Civic	21.5	2199	150	258	8.5	135	67	1367	4300	1770	1470
Land Rover Defender	28.2	2198	122	265	14.7	🟦 90	25	2120	4785	🟦 1790	1790
Mazda CX−5	24.7	2191	150	280	9.4	122	54	1605	4555	1840	1710
Mercedes−Benz G	82.9	2987	211	398	9.1	108	25	2500	4662	🟦 1760	1951
Mini Coupe	19.8	1598	184	192	6.9	143	48	NA	3734	1683	1378
Peugeot 107	9.3	998	68	70	14.2	100	65	210	3430	1630	1465
Porsche Boxster	38.2	2706	265	206	5.8	164	34	1310	4374	1801	1282
Renault Clio	14.3	1461	90	162	12	112	88	1071	4062	1731	1448
Ssangyong Rodius	🟦 18	1998	155	265	🟦 0	105	38	2059	5125	1915	1845
Suzuki Jimny	12	1328	84	81	14.1	87	39	1158	3665	1645	1705
Volkswagen Golf	20.1	1598	105	184	10.7	119	74	1295	4255	1799	1452

Figure D.1: Cell maps for selected rows of the Top gear data: when detecting cellwise outliers with *DetectDeviatingCells* (left-hand side), when using CooLTS with deterministic starts (center) and when using CooS with deterministic starts (right-hand side).

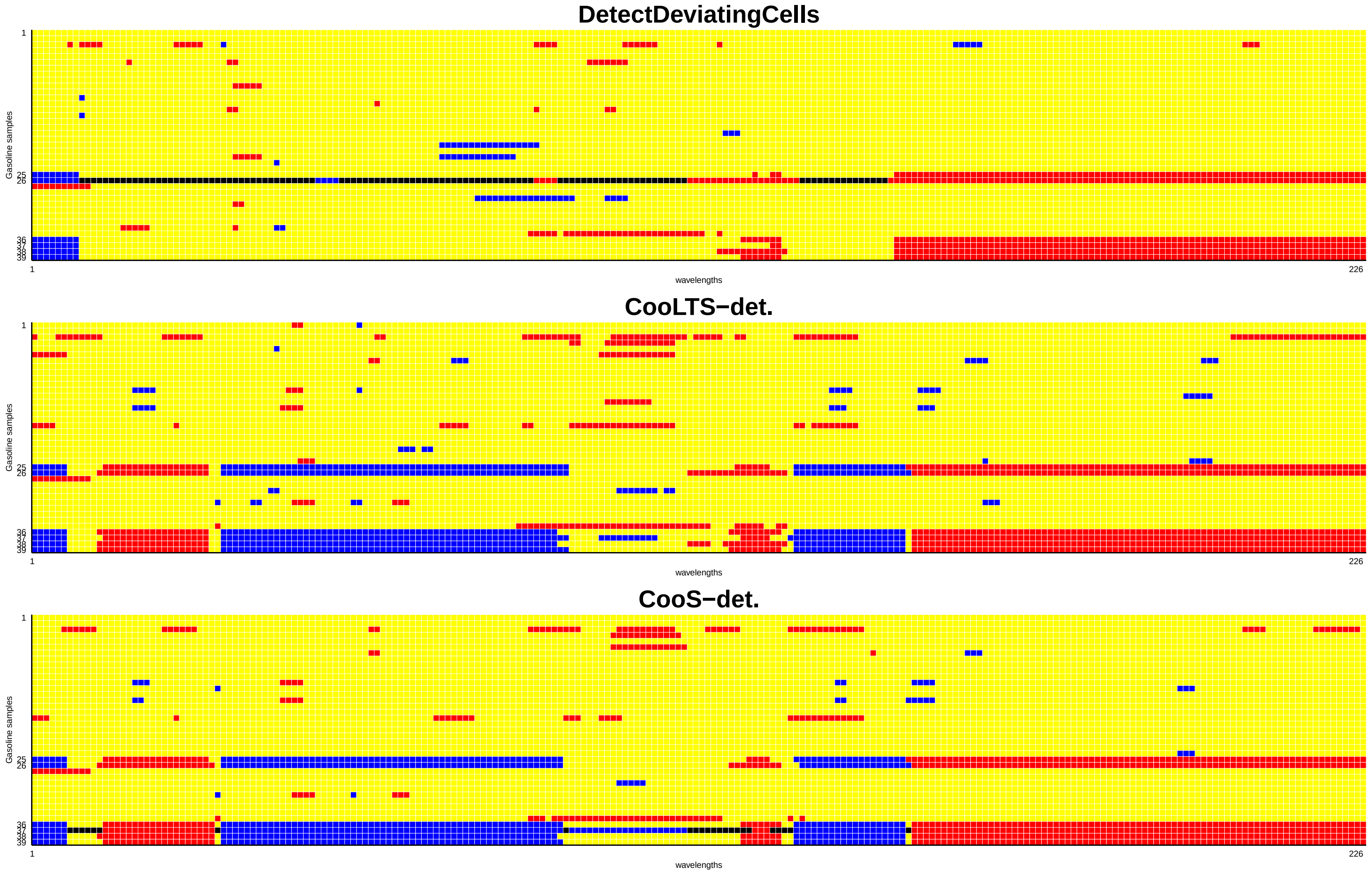


Figure D.2: Cell maps for the Octane dataset with $n = 39$ gasoline samples and $p = 226$ wavelengths: when detecting casewise outliers with a multivariate-PCA method (top panel), when using CooLTS with deterministic starts (middle panel) and when using CooS with deterministic starts (bottom panel).

Bibliography

C. Agostinelli, A. Leung, V. J. Yohai, and R. H. Zamar. Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. *Test*, 24(3):441–461, 2015. ISSN 1863-8260. doi: 10.1007/s11749-015-0450-6. URL http://dx.doi.org/10.1007/s11749-015-0450-6.

A. Alfons. robusthd: Robust methods for high-dimensional data. *R Package Version 0.5.1*, 2016. URL https://cran.r-project.org/web/packages/robustHD/robustHD.pdf.

F. Alqallaf, S. Van Aelst, V. J. Yohai, and R. H. Zamar. Propagation of outliers in multivariate data. *Annals of Statistics*, 37:311–331, 2009. ISSN 00905364. doi: 10.1214/07-AOS588.

A. Asuncion and D. J. Newman. UCI Machine Learning Repository, 2007.

J. L. Bali, G. Boente, D. E. Tyler, and J. L. Wang. Robust functional principal components: A projection-pursuit approach. *Annals of Statistics*, 39:2852–2882, 2011. ISSN 00905364. doi: 10.1214/11-AOS923.

G. E. A. P. A. Batista and M. C. Monard. An analysis of four missing data treatment methods for supervised learning, 2003. ISSN 0883-9514.

G. Boente and M. Salibian-Barrera. S-estimators for functional principal component analysis. *Journal of the American Statistical Association*, 110(511):1100–1111, 2015. doi: 10.1080/01621459.2014.946991. URL http://dx.doi.org/10.1080/01621459.2014.946991.

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL http://www.springerlink.com/index/U0P06167N6173512.pdf.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Chapman and Hall, CRC, 1984. ISBN 0412048418.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996a. ISSN 0885-6125. doi: 10.1007/BF00058655.

L. Breiman. Bias, Variance, and Arcing Classifiers. Technical report, Statistics Department, University of California, Berkeley, CA, 1996b. URL https://www.stat.berkeley.edu/~breiman/arcall96.pdf.

P. Bühlmann and B. Yu. Analyzing bagging. *The Annals of Statistics*, 30:927–961, 2002. ISSN 00905364. doi: 10.1214/aos/1031689014.

L. F. Burgette and J. P. Reiter. Multiple imputation for missing data via sequential regression trees. *American Journal of Epidemiology*, 172(9):1070–1076, November 2010. ISSN 1476-6256. doi: 10.1093/aje/kwq260. URL http://aje.oxfordjournals.org/content/172/9/1070.full.pdf+html.

H. Cevallos Valdiviezo and S. Van Aelst. Tree-based prediction on incomplete data using imputation or surrogate decisions. *Information Sciences*, 311:163–181, August 2015. ISSN 00200255. doi: 10.1016/j.ins.2015.03.018. URL http://linkinghub.elsevier.com/retrieve/pii/S0020025515001838.

C. Croux and G. Haesbroeck. Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87:603–618, 2000. ISSN 0006-3444. doi: 10.1093/biomet/87.3.603. URL http://biomet.oxfordjournals.org/content/87/3/603.short.

C. Croux and A. Ruiz-Gazen. *A Fast Algorithm for Robust Principal Components Based on Projection Pursuit*, pages 211–216. Physica-Verlag HD, Heidelberg, 1996. ISBN 978-3-642-46992-3. doi: 10.1007/978-3-642-46992-3_22. URL http://dx.doi.org/10.1007/978-3-642-46992-3_22.

C. Croux and A. Ruiz-Gazen. High breakdown estimators for principal components: The projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95:206–226, 2005. ISSN 0047259X. doi: 10.1016/j.jmva.2004.08.002.

C. Croux, P. Filzmoser, G. Pison, and P. J. Rousseeuw. Fitting multiplicative models by robust alternating regressions. *Statistics and Computing*, 13(1):23–36, 2003. doi: 10.1023/A:1021979409012. URL http://dx.doi.org/10.1023/A:1021979409012.

C. Croux, L. A. García-Escudero, A. Gordaliza, C. Ruwet, and R. S. Martín. Robust principal component analysis based on trimming around affine subspaces. *Statistica Sinica*, in press.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38,

1977. ISSN 0035-9246. doi: 10.1.1.133.4884. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.4884.

T. G. Dietterich and E. B. Kong. Machine Learning Bias , Statistical Bias , and Statistical Variance of Decision Tree Algorithms. *Machine Learning*, 255:0–13, 1995. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.2702&amp;rep=rep1&amp;type=pdf.

L. L. Doove, S. Van Buuren, and E. Dusseldorp. Recursive partitioning for missing data imputation in the presence of interaction effects. *Computational Statistics and Data Analysis*, 72:92–104, 2014. ISSN 01679473. doi: 10.1016/j.csda.2013.10.025.

B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7:1–26, 1979. ISSN 0090-5364. doi: 10.1214/aos/1176344552.

A. J. Feelders. Handling missing data in trees: surrogate splits or statistical imputation. In *PKDD99: Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, volume 1704 of *Lecture Notes in Computer Science*, pages 329–334. Springer-Verlag, London, UK, 1999. ISBN 3-540-66490-4.

S. Geman, E. Bienenstock, and R. Doursat. Neural Networks and the Bias/Variance Dilemma, 1992. ISSN 0899-7667.

D. Gervini. Robust functional estimation using the spatial median and spherical principal components. *Biometrika*, 95:587–600, 2008.

R. Gnanadesikan and J. R. Kettenring. Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data. *Biometrics*, 28:81–124, 1972. ISSN 0006341X. doi: 10.2307/2528963. URL http://www.jstor.org/stable/2528963.

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

A. Hapfelmeier and K. Ulm. Variable selection by Random Forests using data with missing values. *Computational Statistics & Data Analysis*, 80:129–139, December 2014. ISSN 01679473. doi: 10.1016/j.csda.2014.06.017. URL http://linkinghub.elsevier.com/retrieve/pii/S0167947314001881.

A. Hapfelmeier, T. Hothorn, and K. Ulm. Recursive partitioning on incomplete data using surrogate decisions and multiple imputation. *Computational Statistics and Data Analysis*, 56(6):1552–1565, June 2012. ISSN 01679473. doi: 10.1016/j.csda.2011.09.024. URL http://www.sciencedirect.com/science/article/pii/S0167947311003550.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition.* Springer New York, 2009. ISBN 978-0-387-84857-0. doi: 10.1007/978-0-387-84858-7.

Y. He. *Missing Data Imputation for Tree-Based Models.* PhD thesis, University of California, Los Angeles, 2006.

N. J. Horton and K. P. Kleinman. Much ado about nothing: a comparison of missing data methods and software to fit incomplete data regression models. *The American Statistician*, 61:79–90, 2007. ISSN 0003-1305. doi: 10.1198/000313007X172556. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1839993/pdf/nihms16073.pdf.

T. Hothorn, K. Hornik, and A. Zeileis. Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, 15:651–674, 2006. ISSN 1061-8600. doi: 10.1198/106186006X133933. URL http://pubs.amstat.org/doi/pdf/10.1198/106186006X133933.

T. Hothorn, K. Hornik, C. Strobl, and A. Zeileis. Party: a laboratory for recursive part(y)itioning. *R Package Version 0.9-99996*, 2011. doi: 10.1.1.180.2216.

M. Hubert and E. Vandervieren. An adjusted boxplot for skewed distributions. *Computational Statistics and Data Analysis*, 52:5186–5201, 2008. ISSN 01679473. doi: 10.1016/j.csda.2007.11.008.

M. Hubert, P. J. Rousseeuw, and K. Vanden Branden. ROBPCA: A New Approach to Robust Principal Component Analysis. *Technometrics*, 47:64–79, 2005. ISSN 0040-1706. doi: 10.1198/004017004000000563.

M. Hubert, P. J. Rousseeuw, and T. Verdonck. A deterministic algorithm for robust location and scatter. *Journal of Computational and Graphical Statistics*, 21:618–637, 2012. ISSN 1061-8600. doi: 10.1080/10618600.2012.672100. URL http://dx.doi.org/10.1080/10618600.2012.672100$\delimiter"026E30F$nhttp://www.tandfonline.com/doi/abs/10.1080/10618600.2012.672100#.UdeZC0A-nCM$\delimiter"026E30F$nhttp://www.tandfonline.com/doi/abs/10.1080/10618600.2012.672100.

A. Kapelner and J. Bleich. Prediction with Missing Data via Bayesian Additive Regression Trees. *arXiv.org*, stat.ML, 2013. URL http://arxiv.org/abs/1306.0618$\delimiter"026E30F$npapers2://publication/uuid/CF699DEE-769B-4C53-B5E5-DEDAB0367BFF.

M. A. Klebanoff and S. R. Cole. Use of multiple imputation in the epidemiologic literature. *American Journal of Epidemiology*, 168:355–357, 2008. ISSN 1476-6256. doi:

10.1093/aje/kwn071. URL http://aje.oxfordjournals.org/content/168/4/355.full.pdf+html.

A. Leung, H. Zhang, and R. Zamar. Robust regression estimation and inference in the presence of cellwise and casewise contamination. *Computational Statistics and Data Analysis*, 99:1–11, 2016. ISSN 01679473. doi: 10.1016/j.csda.2016.01.004.

G. Li and Z. Chen. Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and Monte Carlo. *Journal of the American Statistical Association*, 80:759–766, 1985. ISSN 0162-1459. doi: 10.1080/01621459.1985.10478181. URL http://www.tandfonline.com/doi/abs/10.1080/01621459.1985.10478181$\delimiter"026E30F$npapers3://publication/uuid/E7A6D092-1420-4117-A47C-C6F759DA3121.

S. G. Liao, Y. Lin, D. D. Kang, D. Chandra, J. Bon, N. Kaminski, F. C. Sciurba, and G. C. Tseng. Missing value imputation in high-dimensional phenomic data: imputable or not, and how? *BMC bioinformatics*, 15(1):346, November 2014. ISSN 1471-2105. doi: 10.1186/s12859-014-0346-6. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4228077&tool=pmcentrez&rendertype=abstract.

A. Liaw and M. Wiener. Classification and regression by randomForest. *R News*, 2 (December):18–22, 2002.

R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. 2002. ISBN 0471183865. URL http://sfx.libis.be/sfxlcl3?sid=google.

L. Liu, D. M. Hawkins, S. Ghosh, and S. S. Young. Robust singular value decomposition analysis of microarray data. *Proceedings of the National Academy of Sciences of the United States of America*, 100:13167–72, 2003. ISSN 0027-8424. doi: 10.1073/pnas.1733249100. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=263735&tool=pmcentrez&rendertype=abstract.

N. Locantore, J. S. Marron, D. G. Simpson, N. Tripoli, J. T. Zhang, and K. L. Cohen. Robust principal component analysis for functional data. *Test*, 8(1):1–73, 1999. ISSN 1863-8260. doi: 10.1007/BF02595862. URL http://dx.doi.org/10.1007/BF02595862.

G. Louppe. *Understanding random forests from theory to practice*. PhD thesis, University of Liege, 2014. URL http://arxiv.org/abs/1407.7502.

R. A. Maronna. Principal Components and Orthogonal Regression Based on Robust Scales. *Technometrics*, 47:264–273, 2005. ISSN 0040-1706. doi: 10.1198/004017005000000166. URL http://pubs.amstat.org/doi/abs/10.1198/004017005000000166.

R. J. Marshall and P. Kitsantas. Stability and Structure of CART and SPAN Search Generated Data Partitions for the Analysis of Low Birth Weight. *Journal of Data Science*, 10:61–73, 2012.

V. Oellerer, A. Alfons, and C. Croux. The shooting S-estimator for robust regression. *FEB Research Report KBI_1318*, 2013. ISSN 16139658. doi: 10.1007/s00180-015-0593-7. URL https://lirias.kuleuven.be/bitstream/123456789/425555/1/KBI_1318.pdf.

A. Peters, T. Hothorn, and B. Lausen. ipred: Improved predictors. *R News*, 2:33–36, 2002.

J. R. Quinlan. *C4.5: Programs for Machine Learning*, volume 1. 1993. ISBN 1558602380. doi: 10.1016/S0019-9958(62)90649-6. URL http://portal.acm.org/citation.cfm?id=152181.

R Development Core Team. R: a language and environment for statistical computing. Technical report, Vienna, Austria, 2011.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis (Springer Series in Statistics)*. 2005. ISBN 038740080X. doi: 10.1007/b98888. URL http://onlinelibrary.wiley.com/doi/10.1002/0471667196.ess3138/full.

J. A. Rice and B. W. Silverman. Estimating the Mean and Covariance Structure nonparametrically When the Data are Curves. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53:233–243, 1991. ISSN 00359246. doi: 10.2307/2345738.

A. Rieger, T. Hothorn, and C. Strobl. Random Forests with Missing Values in the Covariates. Technical Report 79, Ludwig-Maximilians-Universität Munich, Germany, 2010. URL http://epub.ub.uni-muenchen.de/11481/1/techreport.pdf.

P. J. Rousseeuw and C. Croux. Alternatives to the Median Absolute Deviation. *Journal of the American Statistical Association*, 88:1273–1283, 1993. ISSN 0162-1459. doi: 10.1080/01621459.1993.10476408. URL http://www.jstor.org/stable/2291267.

P. J. Rousseeuw and K. V. Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, August 1999. ISSN 0040-1706. doi: 10.2307/1270566. URL http://dx.doi.org/10.2307/1270566.

P. J. Rousseeuw and W. Van den Bossche. Detecting deviating data cells. *ArXiv e-prints*, January 2016. URL arxiv.org/abs/1601.07251.

D. B. Rubin. Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91:473–489, 1996. ISSN 01621459. doi: 10.2307/2291635.

D. B. Rubin. The Bayesian Bootstrap. *The Annals of Statistics*, 9:130–134, 1981. ISSN 0090-5364. doi: 10.1214/aos/1176345338. URL http://projecteuclid.org/download/pdf_1/euclid.aos/1176345338.

D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, New York, 1987. ISBN 0-471-08705-X.

M. Saar-Tsechansky and F. Provost. Handling Missing Values when Applying Classification Models. *Journal of Machine Learning Research*, 8: 1625–1657, 2007. ISSN 15324435. doi: 10.1.1.72.3271. URL http://ezproxy.stevens.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=34641784&site=eds-live.

M. Salibian-Barrera and V. J. Yohai. A Fast Algorithm for S-Regression Estimates. *Journal of Computational and Graphical Statistics*, 15:414–427, 2006. ISSN 1061-8600. doi: 10.1198/106186006X113629.

P. Sawant, N. Billor, and H. Shin. Functional outlier detection with robust functional principal component analysis. *Computational Statistics*, 27:83–102, 2012. ISSN 09434062. doi: 10.1007/s00180-011-0239-3.

J. L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman & Hall, 1997.

A. D. Shah, J. W. Bartlett, J. Carpenter, O. Nicholas, and H. Hemingway. Comparison of random forest and parametric imputation models for imputing missing data using MICE: A CALIBER study. *American Journal of Epidemiology*, 179:764–774, 2014. ISSN 14766256. doi: 10.1093/aje/kwt312.

D. J. Stekhoven and P. Bühlmann. Missforest-Non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28:112–118, 2012. ISSN 13674803. doi: 10.1093/bioinformatics/btr597.

G. W. Stewart. The Efficient Generation of Random Orthogonal Matrices with an Application to Condition Estimators, 1980. ISSN 0036-1429.

H. Strasser and C. Weber. On the asymptotic theory of permutation statistics. *Mathematical Methods of Statistics*, 2, 1999. ISSN 1066-5307.

M. A. Tanner and W. H. Wong. The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987. doi: 10.1080/01621459.1987.10478458. URL http://www.jstor.org/stable/pdfplus/2289457.pdf.

R. Tibshirani. Bias, variance and prediction error for classification rules. *Monographs of the Society for Research in Child Development*, 79:1–14, 1996. ISSN 1540-5834. doi: 10.1111/mono.12110. URL http://www.ncbi.nlm.nih.gov/pubmed/25102152$\delimiter"026E30F$nhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Bias,+varianceand+prediction+error+for+classification+rules#0.

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17:520–525, 2001. ISSN 1367-4803. doi: 10.1093/bioinformatics/17.6.520. URL http://bioinformatics.oxfordjournals.org/content/17/6/520.full.pdf+html.

W. Vach. *Logistic Regression with Missing Values in the Covariates.* Springer New York, 1994. ISBN 978-0-387-94263-6. doi: 10.1007/978-1-4612-2650-5.

S. Van Aelst. Stahel-Donoho estimation for high-dimensional data. *International Journal of Computer Mathematics*, 93(4):628–639, 2016.

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

S. Van Aelst, E. Vandervieren, and G. Willems. A Stahel-Donoho estimator based on huberized outlyingness. *Computational Statistics and Data Analysis*, 56:531–542, 2012. ISSN 01679473. doi: 10.1016/j.csda.2011.08.014.

S. Van Buuren, J. P. Brand, C. G. Groothuis-Oudshoorn, and D. B. Rubin. Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation*, 76:1049–1064, 2006. ISSN 0094-9655. doi: 10.1080/10629360600810434.

S. Van Buuren. *Flexible Imputation of Missing Data.* Chapman and Hall/CRC 2012, 2012. ISBN 978-1-4398-6824-9. URL http://www.crcnetbase.com/isbn/9781439868256.

S. Van Buuren and K. Groothuis-Oudshoorn. MICE: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45:1–67, 2011. ISSN 15487660. URL http://www.jstatsoft.org/v45/i03/.

S. Visuri, V. Koivunen, and H. Oja. Sign and rank covariance matrices. *Journal of Statistical Planning and Inference*, 91:557–575, 2000. ISSN 03783758. doi: 10.1016/S0378-3758(00)00199-3.

I. R. White and J. B. Carlin. Bias and efficiency of multiple imputation compared with complete-case analysis for missing covariate values. *Statistics in Medicine*, 29:2920–2931, 2010. ISSN 02776715. doi: 10.1002/sim.3944.