Statistical Methods & Applications (2022) 31:365–386
https://doi.org/10.1007/s10260-021-00575-5

ORIGINAL PAPER

Chunk-wise regularised PCA-based imputation of missing data

A. Iodice D'Enza[1] · A. Markos[2] · F. Palumbo[1]

Received: 25 October 2020 / Accepted: 13 June 2021 / Published online: 25 June 2021

Abstract

Standard multivariate techniques like Principal Component Analysis (PCA) are based on the eigendecomposition of a matrix and therefore require complete data sets. Recent comparative reviews of PCA algorithms for missing data showed the regularised iterative PCA algorithm (RPCA) to be effective. This paper presents two chunk-wise implementations of RPCA suitable for the imputation of "tall" data sets, that is, data sets with many observations. A "chunk" is a subset of the whole set of available observations. In particular, one implementation is suitable for distributed computation as it imputes each chunk independently. The other implementation, instead, is suitable for incremental computation, where the imputation of each new chunk is based on all the chunks analysed that far. The proposed procedures were compared to batch RPCA considering different data sets and missing data mechanisms. Experimental results showed that the distributed approach had similar performance to batch RPCA for data with entries missing completely at random. The incremental approach showed appreciable performance when the data is missing not completely at random, and the first analysed chunks contain sufficient information on the data structure.

Keywords Principal components · Missing data · Eigenspace arithmetics

✉ A. Iodice D'Enza
iodicede@unina.it

A. Markos
amarkos@eled.duth.gr

F. Palumbo
fpalumbo@unina.it

[1] Dipartimento di Scienze Politiche, Univeristà degli studi di Napoli Federico II, Naples, Italy

[2] Department of Primary Education, Democritus University of Thrace, Alexandroupolis, Greece

1 Introduction

Missing data are a common and pervasive problem in almost all kinds of studies that also complicate the execution and interpretation of any supervised or unsupervised learning technique. In the present work, the focus is on missing data in the context of principal component analysis (PCA; Jolliffe 2002). Two general strategies are most common in practice, (i) listwise or pairwise deletion, with the drawback of possibly discarding a considerable amount of observations and (ii) missing data imputation before conducting PCA. Imputation by the mean is a simple approach, where missing values are replaced with the mean value of the attribute: such approach is straightforward and suitable in case of a small number of missing entries, but it reduces the variance of the attribute in question. Furthermore, mean imputation affects the correlation structure considerably (Little and Rubin 2019).

Such strategies, however, do not address two important aspects that need to be taken into account when dealing with missing data: the underlying relationship structure of the data set and the missing data mechanism. Concerning the latter aspect, we refer to missing completely at random (MCAR) and missing not completely at random (MNCAR) mechanisms (Rubin 1976). Under the MCAR mechanism, the probability that a given entry is missing does not depend on the attribute, or any attribute, value. For the MNCAR mechanism, we refer to the Loisel and Takane (2019) definition: the missing values characterising a *target* attribute depend on the values of one (or more) *agent* attributes, that may or may not be part of the considered data. The present study treats both MAR (missing at random) and MNAR (missing not at random) under the umbrella of MNCAR.

Therefore, enhanced strategies to deal with missings are designed to preserve the underlying data structure, assuming a specific data mechanism. Under the assumption of data sampled from a multivariate normal distribution, a general approach for the imputation of missing values is the so-called joint modeling: an expectation-maximisation algorithm (EM, Dempster et al. 1977) provides the likelihood estimates of the corresponding parameters; the missing values are then imputed via linear regression (for details, see, e.g., Schafer 1997).

Other strategies for PCA in case of missing data also exist: e.g., to obtain a PCA solution of incomplete data by skipping the missing entries. For instance, the PCA on the matrix of Euclidean distances between observations provides the scores for the observations: distance computation refers to the complete entries only (Gower 1971). Likewise, the PCA loadings can be obtained by analysing the covariance matrix computed on the complete entries only. These methods are somewhat complementary, as they respectively provide the PCA observation scores and attribute scores: however, they present drawbacks such as the possibility of negative eigenvalues when it comes to the decomposition of either the distance matrix or the correlation matrix.

More sophisticated methods to compute PCA in the presence of missing data have also been developed, see Dray and Josse (2015), Folch-Fortuny et al. (2015), Van Ginkel et al. (2014), Geraci and Farcomeni (2018) and Loisel and Takane (2019) for a thorough description of (most of) the different implementations and a

comparison of their performance on simulated and real data sets. A common aspect characterising efficient PCA algorithms for missing data is their implementation via iterative procedures. A regularised iterative PCA algorithm (RPCA, Josse et al. 2009) seems to outperform other approaches, as pointed out in a recent review by Loisel and Takane (2019).

Iterative procedures may be very efficient for small data sets, but their application becomes impractical when dealing with *tall* incomplete data sets, that is, data sets with a large number of observations compared to attributes. More generally, having to deal with large and high dimensional data, the feasible application of PCA is undermined by the computational limitations that affect eigenvalue decomposition (EVD) and singular value decomposition (SVD) which are the core step of PCA. There exist several approaches in the literature that aim to enhance and extend the applicability of SVD; depending on the specific aim, *batch* and *incremental* are two classes of such approaches. Batch methods aim to increase the computational efficiency of SVD, and therefore, its applicability to matrices of increased size. On the contrary, incremental methods aim to process the whole data set as a sequence of incoming chunks and to update the current (SVD) solution as a new chunk comes in. Throughout the paper, the term "chunk" here refers to a subset of the whole set of available observations. Splitting data into chunks and processing them sequentially or in parallel can be a convenient option and, in some cases, even necessary. This is when the data might be too large to be stored in memory or produced at a high rate, as in the case of data flows, where the data set is never available as a whole. Incremental SVD (and, consequently, PCA) approaches are widely used in rather recent application fields, from recommender systems, e.g., the Netflix competition (Ilin and Raiko 2010), to image recognition, as in eigenfaces, to deal with the problem of human face recognition (Navarrete and Ruiz-del-Solar 2002), to name but a few. In extreme scenarios such as data flows, online procedures are applied that constantly update the solution: we refer the reader to the notable review of sub-space tracking of data flows with missings proposed by Balzano et al. (2018).

In this paper, we consider the situation where a large amount of data is generated according to a steady correlation structure, whereas the missing data mechanism may change. Such condition is characteristic of process data sets (Severson et al. 2017): PCA is, in fact, widely used as multivariate statistical process control, and missing values may arise, e.g., from sensor failures. By focusing on a computationally efficient class of incremental eigendecomposition methods with desirable properties that ease their embedding in PCA (Cardot and Degras 2018; Markos and Iodice D'Enza 2018), this work proposes two novel chunk-wise RPCA implementations for the analysis of tall data sets containing missings. One implementation is referred to as *naive*[1] chunk-wise RPCA (*naive* CW-RPCA) in which RPCA runs over every single chunk, and then the chunk-based solutions are merged; the other

[1] we use the word *naive* as in the naive Bayes classifier: the Bayes classifier is naive because it assumes, within each class, the features to be independent of each other; such assumption is generally not true, yet, it simplifies the estimation dramatically (see, e.g., Hastie et al. 2009). Similarly, the naive version of the proposed approach processes each chunk independently from the others, and the imputed values will depend on that data chunk only.

implementation is referred to as CW-RPCA (non-*naive*) in which the data chunks are imputed sequentially (as in data flows), and the RPCA solution is updated as each new chunk is processed. Experiments to assess the performance of the proposed procedures compared to the standard RPCA are carried out on different simulated data sets, considering different missing data generation mechanisms, as well as on a benchmark process data set.

The rest of the paper is structured as follows: Sect. 2 presents the definition of PCA and reviews two efficient iterative approaches to impute missing data in the PCA context. Section 3 first summarises two incremental eigendecomposition methods that are then exploited to derive chuck-wise iterative PCA implementations for single imputation of missing data. Experimental results are reported in Sects. 4, and 5 concludes the paper.

2 PCA with missing data

Let $\mathbf{X}$ be an $I \times J$ data matrix, where I is the number of observations, and J is the number of quantitative attributes. Depending on whether the attributes are scaled to a unit variance or not, we refer to correlation PCA or covariance PCA, respectively (see, e.g., Borgognone et al. 2001; Jolliffe 2002).

In case of complete data (that is, no missing entries), the PCA solution of $\mathbf{X}$ is obtained via the singular value decomposition (SVD) of the following matrix (see, e.g., Greenacre 2010, Equation (6.2), page 60)

$$\mathbf{S} = I^{-1/2}(\mathbf{X} - \mathbf{M})J^{-1/2} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^{\mathsf{T}} \tag{1}$$

where $\mathbf{M} = \mathbf{I} - I^{-1}\mathbf{1}\mathbf{1}^{\mathsf{T}}$ is the centring operator and $\mathbf{1}$ is an I-dimensional vector of ones; $\mathbf{V}$ is a $I \times J$ orthonormal matrix with left singular vectors on columns, $\boldsymbol{\Sigma}$ is a diagonal matrix containing the J singular values $\sqrt{\lambda_j}$, $j = 1, \ldots, J$, and $\mathbf{U}$ is a $J \times J$ matrix of right singular vectors; λ_j is the *jth* eigenvalue of the matrix $\mathbf{S}^{\mathsf{T}}\mathbf{S}$. Therefore, the *jth* singular value corresponds to the standard deviation along the direction of the j^{th} singular vector, $j = 1, \ldots, J$.

Let $\hat{\mathbf{V}}, \hat{\mathbf{U}}$ and $\hat{\boldsymbol{\Sigma}}$ be the first d singular vectors and values; we refer to $\hat{\mathbf{F}} = I^{1/2}\hat{\mathbf{V}}\hat{\boldsymbol{\Sigma}}$ as row principal coordinates, and to $\hat{\mathbf{G}} = J^{1/2}\hat{\mathbf{U}}$ as standard column coordinates. In particular, row principal coordinates on the *jth* dimension are such that their average squared sum equals the eigenvalue λ_j, that is

$$I^{-1}\hat{\mathbf{F}}^{\mathsf{T}}\hat{\mathbf{F}} = I^{-1}\left(\hat{\boldsymbol{\Sigma}}\hat{\mathbf{V}}^{\mathsf{T}}I^{1/2}\right)I^{1/2}\hat{\mathbf{V}}\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Sigma}}^2;$$

similarly, the standard columns coordinates are such that their average sum of squares equals one

$$J^{-1}\hat{\mathbf{G}}^{\mathsf{T}}\hat{\mathbf{G}} = J^{-1}\left(\hat{\mathbf{U}}^{\mathsf{T}}J^{1/2}\right)J^{1/2}\hat{\mathbf{U}} = \mathbf{I}.$$

By the Eckart and Young (1973) theorem, $\hat{\mathbf{F}}\hat{\mathbf{G}}^{\mathsf{T}}$ represents the best rank-$\mathcal{D}$ approximation of $\mathbf{X}$ (centred) in the least squares sense:

$$\hat{\mathbf{F}}\hat{\mathbf{G}}^{\mathsf{T}} = I^{1/2}\hat{\mathbf{V}}\hat{\mathbf{\Sigma}}\hat{\mathbf{U}}^{\mathsf{T}}J^{1/2} = n^{1/2}\mathbf{S}J^{1/2\hat{}} = \hat{\mathbf{X}} - \mathbf{M} \rightarrow \hat{\mathbf{X}} = \mathbf{M} + \hat{\mathbf{F}}\hat{\mathbf{G}}^{\mathsf{T}}. \tag{2}$$

Therefore, the PCA loss function

$$\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 = \|\mathbf{X} - \mathbf{M} - \hat{\mathbf{F}}\hat{\mathbf{G}}^{\mathsf{T}}\|_F^2 \tag{3}$$

is referred to as the low-rank approximation criterion, where $\|\cdot\|_F$ is the Frobenius norm.

In a model-based perspective, PCA is defined as a bilinear fixed-effect model, with the data being characterised by a $\mathcal{D}$-rank plus Gaussian noise structure; formally, the general element of $\mathbf{X}$ is

$$x_{ij} = m_j + \sum_{d=1}^{\mathcal{D}} f_{id}g_{jd} + \epsilon_{ij} = m_j + \sum_{d=1}^{\mathcal{D}} \sqrt{\lambda_d}v_{id}u_{jd} + \epsilon_{ij}, \tag{4}$$

with $i = 1, \ldots, I$, $j = 1, \ldots, J$ and $\epsilon_{ij} \sim N(0, \sigma^2)$. Note that the maximum likelihood estimates of the PCA model correspond to the least squares solution.

2.1 Single imputation via iterative PCA

In order to account for the presence of missing values in PCA, an $I \times J$ weight matrix $\mathbf{W}$ is defined that has general element $w_{ij} = 0$ if the value for the *ith* observation of the *jth* attribute is missing, $w_{ij} = 1$ otherwise. Then the criterion is

$$\sum_{i=1}^{I}\sum_{i=1}^{J} w_{ij}\left(x_{ij} - m_j - \sum_{d=1}^{\mathcal{D}} f_{id}g_{jd}\right)^2, \tag{5}$$

showing that the least squares criterion is minimised by only considering the non-missing entries.

In algebraic form, the loss function in Eq. (3) can be modified as follows

$$\|\mathbf{W} * \left(\mathbf{X} - \mathbf{M} - \hat{\mathbf{F}}\hat{\mathbf{G}}^{\mathsf{T}}\right)\|^2 \tag{6}$$

where the operator '$*$' indicates the Hadamard product. Equivalently, by defining

$$\tilde{\boldsymbol{X}} = \mathbf{W} * \mathbf{X} + (1 - \mathbf{W}) * \hat{\mathbf{F}}\hat{\mathbf{G}}^{\mathsf{T}},$$

the loss function can be further re-stated as

$$\|\tilde{\boldsymbol{X}} - \mathbf{M} - \hat{\mathbf{F}}\hat{\mathbf{G}}^{\mathsf{T}}\|^2, \tag{7}$$

as pointed out by Loisel and Takane (2019).

The optimisation of the criterion in Eq. (6) is not possible via a direct solution. Kiers (1997) proposed a general approach to the weighted least squares (WLS) fitting procedure by iteratively performing ordinary least square fitting: it can be viewed as a majorisation-minimisation problem. In particular, the minimisation of the loss function in Eq. (6) is obtained via repeated minimisations of the majorising function, which is simpler to optimise than the loss function. The procedure,

introduced by Kiers (1997) and further described by Josse and Husson (2012), consists of the following steps:

step 1: Initialise the iteration counter $\ell = 0$. Replace each missing entry in $\mathbf{X}$ with some initialisation values, e.g. the mean of the complete values of the *jth* attribute, obtaining the starting $\tilde{\boldsymbol{X}}^{\ell}$;

step 2: Perform a PCA on $\tilde{\boldsymbol{X}}^{\ell}$ to obtain $\hat{\mathbf{F}}^{\ell}$ and $\hat{\mathbf{G}}^{\ell}$. Use the reconstruction formula

$$\hat{x}_{ij}^{\ell} = \sum_{d=1}^{\mathcal{D}} \sqrt{\hat{\lambda}_d^{\ell}} \hat{v}_{id}^{\ell} \hat{u}_{jd}^{\ell} = \sum_{d=1}^{\mathcal{D}} \hat{f}_{id} \hat{g}_{jd} \qquad \forall i, j \tag{8}$$

to obtain $\hat{\mathbf{X}}^{\ell}$;

step 3: Impute the missing entries in $\tilde{\boldsymbol{X}}^{\ell}$ with the corresponding values of $\hat{\mathbf{X}}^{\ell}$, formally $\tilde{\boldsymbol{X}}^{\ell} = \mathbf{W} * \mathbf{X} + (1 - \mathbf{W}) * \hat{\mathbf{X}}^{\ell}$; and update the counter $\ell = \ell + 1$;

step 4: Repeat steps 2 and 3 until convergence, that is, when the value of Equation (6) does not decrease from an iteration to the next one.

As Kiers (1997) pointed out, the procedure above is monotonically decreasing; as the function value is bounded below by zero, the convergence of the procedure is guaranteed. In general, the procedure is not guaranteed to reach a global minimum of the WLS loss function, and multiple random starts are suggested/required. In the context of handling missing data, the weights are binary, and the missing data entries are usually initialised as described in *step 1* (Josse and Husson 2012). A graphical description of the iterative algorithm behaviour on a toy example with five bivariate points and one missing entry can be found in Josse and Husson (2016).

Interestingly, the above procedure appears in the literature with many different names: iterative PCA (iPCA, Dray and Josse 2015), weighted low-rank approximation (WLRA, Kiers 1997), expectation-maximisation PCA (EM-PCA, Josse and Husson 2012; Geraci and Farcomeni 2018). The last name is due to the definition of the procedure as an EM algorithm returning maximum-likelihood estimates of the parameters for the fixed-effects model in Eq. (4). The iPCA procedure provides both model parameter estimates and missing values imputation. Therefore, it is a single imputation method that takes into account both the similarities among individuals and the correlation structure characterising the attributes. However, iPCA may suffer from overfitting, as both the number of missing entries and the dimensionality of the underlying structure increase.

2.2 Single imputation via Regularised PCA

To overcome the limitations of iPCA, a regularised version of the iPCA algorithm (RPCA) has been proposed by Josse et al. (2009) to deal with the overfitting problem. Since regularisation affects the singular values, the RPCA algorithm differs from iPCA in the reconstruction formula used to impute the missing entries at each iteration. More specifically, RPCA differs from iPCA concerning the following modification of Eq. (8)

$$\hat{x}_{ij}^{\ell} = \sum_{d=1}^{\mathcal{D}} \left(\sqrt{\hat{\lambda}_d^{\ell}} - \frac{(\hat{\sigma}^2)^{\ell}}{\sqrt{\hat{\lambda}_d^{\ell}}} \right) \hat{v}_{id}^{\ell} \hat{u}_{jd}^{\ell} \tag{9}$$

where the singular value $\sqrt{\hat{\lambda}_d^{\ell}}$ is replaced by its shrunk version $\left(\sqrt{\hat{\lambda}_d^{\ell}} - \frac{(\hat{\sigma}^2)^{\ell}}{\sqrt{\hat{\lambda}_d^{\ell}}} \right)$ and $(\hat{\sigma}^2)^{\ell} = \frac{1}{J-\mathcal{D}\sum_{d=\mathcal{D}+1}^{J} \hat{\lambda}_d}$. The idea is to remove the effect of the last dimensions, that are considered to be noise, on the imputation of the missings. The shrinkage depends on the so-called tuning parameter $\mathcal{D}$ and, in particular, on the $J - \mathcal{D}$ dimensions that are assumed to be noise. Josse and Husson (2016) recommend using a cross-validation approach to choose $\mathcal{D}$ (Bro et al. 2008). In this paper, we do not concern ourselves with the choice of the tuning parameter; we consider the number of dimensions $\mathcal{D}$ to be given for all the considered approaches.

A review by Loisel and Takane (2019) compared the performance of RPCA to other methods for PCA with missings that were shown to perform well in previous comparative studies. In particular, the considered methods are missing data passive method (MDP, Takane and Oshima-Takane 2003; Benzécri 1973), trimmed scores regression method (TSR,Folch-Fortuny et al. 2015) and the data augmentation method (DA, Schafer 1997). Except for MDP and DA, all the best-performing methods were iterative. Overall, RPCA was found to be the best performing method in terms of parameter recovery. Furthermore, in the framework of PCA on process data with missings, an interesting comparative review was conducted by Severson et al. (2017). The authors concluded that the method to use might depend on the scenario, which is application domain-specific, yet iterative PCA algorithms demonstrated good performance in that framework, too.

3 Chunk-wise RPCA for missing data

Iterative PCA methods may be very efficient for small incomplete data sets, but their application becomes impractical when dealing with *tall* data sets. In fact, when the number of observations is large, it can be profitable to analyse the dataset chunk-wise and update the solution as new chunks are analysed, more so if each chunk is analysed iteratively.

In this section, we extend the RPCA algorithm to deal with tall incomplete data sets. The general idea of the chunk-wise RPCA approach is to obtain the PCA solution of a chunk split dataset by either merging the chunk-based PCA solutions or by updating the PCA solution incrementally as new chunks are analysed. To accomplish this, we rely on two efficient approaches for incrementally computing the EVD/SVD of a data matrix: the *eigenspace arithmetics* method of Hall et al. (2002) and the *block incremental SVD with mean update*, a method proposed by Ross et al. (2008). The two methods are thoroughly described below.

3.1 Incremental eigendecomposition

Consider the case where the quantitative $I \times J$ data matrix $\mathbf{X}$ is split in K chunks

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \dots \\ \mathbf{X}_K \end{bmatrix}. \tag{10}$$

Eigenspace arithmetics. Given two subsequent chunks $\mathbf{X}_1$ and $\mathbf{X}_2$ we define the corresponding eigenspaces as $\Omega_k = \{\mathbf{U}_k, \Sigma_k, \mathbf{V}_k, \mu_k, I_k\}$, $k = 1, 2$; μ_k and I_k are the chunk mean and size, respectively. The approach proposed by Hall et al. (2002) allows one to merge Ω_1 and Ω_2 to obtain Ω_3, the eigenspace of $\mathbf{X}_3 = [\mathbf{X}_1; \mathbf{X}_2]$, that is $\Omega_3 = \Omega_1 \oplus \Omega_2$, with '$\oplus$' being the *merge* operator. Adding new data to an existing eigenspace makes the eigenvectors (singular vectors) to rotate and it scales the eigenvalues according to data spread. Therefore the eigenvectors in $\mathbf{V}_3$ are linear combinations of the already available, $\mathbf{V}_1$. In order to deal with a change in dimension, a basis sufficient span $\mathbf{V}_3$ is constructed, that is $\mathbf{V}_1$ augmented by $\mathbf{v}$, the latter given by

$$\mathbf{v} = orth(\psi[\mathbf{H}, \mathbf{h}]); \tag{11}$$

the *orth* operator stands for a Gramm-Schmidt orhogonalisation procedure, ψ discards very small column vectors from the matrix, and $\mathbf{v}$ is the set of t eigenvectors that are outside the eigenspace Ω_1; $\mathbf{H}$ is the null space of both $\mathbf{V}_1$ and $\mathbf{V}_2$; $\mathbf{h}$ is the component of the vector joining the means $(\mu_1 - \mu_2)$ that lies in the null space of both subspaces. More specifically,

$$\mathbf{H} = \mathbf{V}_2 - \mathbf{V}_1\mathbf{V}_1^\mathsf{T}\mathbf{V}_2 \text{ and } \mathbf{h} = (\mu_1 - \mu_2) - \mathbf{V}_1\mathbf{V}_1^\mathsf{T}(\mu_1 - \mu_2).$$

Finally, the *merged* eigenvectors are given by $\mathbf{V}_3 = [\mathbf{V}_1, \mathbf{v}]\mathbf{R}$, where $\mathbf{R}$ is an orthonormal matrix obtained from the SVD of the following block matrix:

$$\begin{bmatrix} \mathbf{\Sigma}_1\mathbf{U}_1^\mathsf{T} & \mathbf{V}_1^\mathsf{T}\mathbf{V}_2\mathbf{\Sigma}_2\mathbf{U}_2^\mathsf{T} \\ 0 & \mathbf{v}^\mathsf{T}\mathbf{V}_2\mathbf{\Sigma}_2\mathbf{U}_2^\mathsf{T} \end{bmatrix} + \begin{bmatrix} \mathbf{V}_1^\mathsf{T}(\mu_1 - \mu_3)\mathbf{1}_{I_1} & \mathbf{V}_1^\mathsf{T}(\mu_2 - \mu_3)\mathbf{1}_{I_2} \\ \mathbf{v}^\mathsf{T}(\mu_1 - \mu_3)\mathbf{1}_{I_1} & \mathbf{v}^\mathsf{T}(\mu_2 - \mu_3)\mathbf{1}_{I_2} \end{bmatrix} = \mathbf{R}\mathbf{\Sigma}\mathbf{U}^\mathsf{T}, \tag{12}$$

where $\mathbf{1}_I$ is an I-dimensional vector of ones and $\mu_3 = \frac{1}{I_1+I_2}(\mu_1 I_1 + \mu_2 I_2)$.

The remaining elements of the SVD-based eigenspace $\mathbf{\Omega}_3$ are given by

$$\mathbf{\Sigma}_3 = \mathbf{\Sigma} \text{ and } \mathbf{U}_3 = \mathbf{U}. \tag{13}$$

The PCA observation scores and loadings are given by $\mathbf{F} = I_3^{1/2}\mathbf{V}_3\mathbf{\Sigma}_3$ and $\mathbf{G} = J^{1/2}\mathbf{U}_3$, respectively[2].

[2] Note that in the original paper by Hall et al. (2002), the observations are reported on the columns of the data matrix, which is, therefore, $J \times I$ and it can be referred to as $\mathbf{X}^\mathsf{T}$; in this paper, $\mathbf{X}$ is such that I observations are on rows and J attributes are on columns. As a consequence, in Hall et al. (2002), the reference decomposition is $\mathbf{X}^\mathsf{T} = \mathbf{U}_{(h)}\mathbf{\Sigma}\mathbf{V}_{(h)}^\mathsf{T}$, where the columns of $\mathbf{U}_{(h)}$ represent an orthonormal basis

Block incremental SVD with mean update. The incremental SVD approach by Ross et al. (2008) is based on the following Lemma:

Given the SVD of $\mathbf{X}_1 = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^\mathsf{T}$,

$$\begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{bmatrix} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_1 & \mathbf{0}\\ \mathbf{L} & \mathbf{\Gamma}\mathbf{Q}^\mathsf{T}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1\\ \mathbf{Q}\end{bmatrix}, \tag{14}$$

where $\mathbf{L} = \mathbf{X}_2\mathbf{V}_1^\mathsf{T}$, $\mathbf{Q}$ is the result from the $\mathbf{QR}$-decomposition of $\Gamma = \mathbf{X}_2 - \mathbf{L}\mathbf{V}_1$ and $\mathbf{I}$ is the identity matrix. In order to take into account the varying mean, the updated mean vector μ_3 is added to $\mathbf{X}_2$.

Apply the SVD to the matrix $\begin{bmatrix}\mathbf{\Sigma}_1 & \mathbf{0}\\ \mathbf{L} & \mathbf{K}\mathbf{Q}^\mathsf{T}\end{bmatrix}$ to obtain $\mathbf{U}_m\mathbf{\Sigma}_m\mathbf{V}_m^\mathsf{T}$.

Finally, $\mathbf{U}_3 = \begin{bmatrix}\mathbf{U}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\mathbf{U}_m$, $\mathbf{\Sigma}_3 = \mathbf{\Sigma}_m$, $\mathbf{V}_3 = \mathbf{V}_m\begin{bmatrix}\mathbf{V}_1\\ \mathbf{Q}\end{bmatrix}$.

Note that the PCA solution obtained using both approaches described above is *exact*: the solutions collapse into the ordinary PCA solution on the covariance matrix. If the PCA solution on the correlation matrix is needed, then the attributes need to be scaled in advance. Furthermore, as shown in Ross et al. (2008), the computational complexity of the incremental SVD algorithm is similar to the eigenspace arithmetics approach of Hall et al. (2002): in fact, the incremental SVD incorporates new data directly, without the additional step of computing the eigenvalue decomposition of each new chunk.

At this point, it is also important to outline that there are no differences in computational complexity between incremental SVD and ordinary SVD. In particular, the ordinary SVD of the $I \times J$ matrix $\mathbf{X}$ requires $\mathcal{O}(IJ^2)$ operations (see, e.g., Golub and Van Loan 2012), provided that $I > J$. Levey and Lindenbaum (2000) suggest that the optimal chunk size for the incremental SVD is given by $I_k = \lfloor J/\sqrt{2} \rfloor$, where $\lfloor value \rfloor$ indicates that *value* is rounded down. Therefore, to obtain chunk-wise SVD of $\mathbf{X}$, $k = \lfloor I/I_k \rfloor$ updates are required. Furthermore, the SVD of the central block matrix in Eq. (14) is needed in each update, and it has a complexity of $\mathcal{O}\left((I_k + \mathcal{D})^2 J\right)$, so the computation of the SVD of $\mathbf{X}$ has a complexity $\mathcal{O}\left(k(I_k + \mathcal{D})^2 J\right)$.

Proposition 1 *The complexity of incremental SVD is equivalent to the SVD on the full matrix* $\mathbf{X}$*, that is* $\mathcal{O}(IJ^2)$*, assuming that* $I > J$.

Proof For sake of simplicity, and without loss of generality, let I be a multiple of I_k, thus $k = \frac{I}{I_k} = \sqrt{2}\frac{I}{J}$, and let $\mathcal{D}$ be fixed. The computational complexity for k updates of the SVD can be then re-written as follows

Footnote 2 continued
for the observations space, and the columns of $\mathbf{V}_{(h)}$ are an orthonormal basis for the attributes space. Since we refer to the decomposition of $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$, we consider $\mathbf{U}$ to be the basis of the columns space (attributes) and $\mathbf{V}$ to be the basis of the row space (observations). Hence $\mathbf{U} = \mathbf{V}_{(h)}$ and $\mathbf{V} = \mathbf{U}_{(h)}$.

$$
\begin{aligned}
\mathcal{O}\left(k(I_k+\mathcal{D})^2 J\right) &= \mathcal{O}\left(\sqrt{2}\frac{I}{J}\left(\frac{J}{\sqrt{2}}+\mathcal{D}\right)^2 J\right) = \mathcal{O}\left(\sqrt{2}I\left(\frac{J}{\sqrt{2}}+\mathcal{D}\right)^2\right) \\
&= \mathcal{O}\left(\sqrt{2}I\left(\frac{J^2}{2}+\mathcal{D}^2+2\frac{J}{\sqrt{2}}\mathcal{D}\right)\right) \\
&= \mathcal{O}\left(\sqrt{2}I\frac{J^2}{2}+\sqrt{2}I\mathcal{D}^2+2\sqrt{2}I\frac{J}{\sqrt{2}}\mathcal{D}\right) \\
&= \mathcal{O}\left(I\frac{J^2}{\sqrt{2}}+\sqrt{2}I\mathcal{D}^2+2IJ\mathcal{D}\right)
\end{aligned}
\tag{15}
$$

Dropping constant and non-dominant (linear) terms from the left-hand side expression of Eq. (15), leads to

$$
\mathcal{O}\left(I\frac{J^2}{\sqrt{2}}+\sqrt{2}I\mathcal{D}^2+2IJ\mathcal{D}\right) = \mathcal{O}(IJ^2)
$$

□

As it will be illustrated in the next section, *eigenspace arithmetics* is better suited to merge chunk-based RPCA solutions in a naive way, whereas *block incremental SVD with mean update* is better suited for incremental updates of the RPCA solution.

3.2 Chunk-wise single imputation via RPCA

Naive CW-RPCA. An incomplete data chunk, $\mathbf{X}_i$, needs to be imputed before the current PCA solution can be updated. A simple and straightforward strategy is to *(i)* impute each single chunk with RPCA and store the corresponding eigenspace, and *(ii)* merge the chunk-based solution using the *eigenspace arithmetics* approach to obtain the full PCA solution: we refer to this approach as *naive* CW-RPCA since the RPCA-based imputation of a chunk is independent of the other chunks. The advantage of naive CW-RPCA is two-fold: it can be easily parallelised as chunk imputations are independent of each other; the procedure iterates over every single chunk, and not on the full matrix. The drawback is that the underlying structure used to impute the missings is estimated on the single chunk and not on the whole data matrix. This may harm the accuracy of the results, depending on the scenario, as pointed out in Sect. 4.

CW-RPCA. As opposed to the naive implementation, the underlying data structure used to impute each new chunk is based not only on the chunk itself, but also on the chunks analysed that far. In particular, the CW-RPCA procedure is an embedding of the *incremental SVD* and of a suitably modified version of RPCA. In order to ease the description of the procedure, we will assume the data to be centred and equally scaled.

Recalling from Sect. 2.1 that, once RPCA has converged, $\hat{\mathbf{X}}_i$ is the rank-$\mathcal{D}$ approximation of the chunk $\tilde{\boldsymbol{X}}_i$, and $\tilde{\boldsymbol{X}}_i = \mathbf{W} * \mathbf{X}_i + (1-\mathbf{W}) * \hat{\mathbf{X}}_i$, let $\boldsymbol{\Omega}$ be the

current eigenspace, based on all the chunks insofar processed. Standard RPCA is applied to the first chunk, $\mathbf{X}_1$. The CW-RPCA procedure, for the general chunk $\mathbf{X}_i$ and $i > 1$, can be summarised as follows:

step 1 Apply a modified version of the RPCA algorithm, based on block incremental SVD with mean update on $\mathbf{X}_i$ to obtain $\tilde{\boldsymbol{X}}_i$;

step 2 Update the current eigenspace $\boldsymbol{\Omega}$ according to the obtained $\tilde{\boldsymbol{X}}_i$.

The RPCA algorithm used in step 1 is modified as follows: in the general iteration ℓ, the singular vectors and values used to obtain $\hat{\mathbf{X}}_i^{\ell}$ are elements of the current $\boldsymbol{\Omega}$ updated by $\tilde{\boldsymbol{X}}^{\ell}$ using the *incremental* SVD; therefore, they are not just resulting from the SVD of $\tilde{\boldsymbol{X}}^{\ell}$ (as in the standard implementation of RPCA).

For a further intuition on how CW-RPCA compares to RPCA, consider a new chunk $\mathbf{X}_2$: the application of CW-RPCA on $\mathbf{X}_2$ is equivalent to the application of RPCA on the matrix $\left[\tilde{\boldsymbol{X}}_1; \mathbf{X}_2\right]$, where $\tilde{\boldsymbol{X}}_1$ is the imputed version of $\mathbf{X}_1$. More generally, CW-RPCA of a chunk $\mathbf{X}_i$ is equivalent to the application of RPCA on $\left[\tilde{\boldsymbol{X}}_1; \tilde{\boldsymbol{X}}_2, \ldots, \mathbf{X}_i\right]$. The advantage of CW-RPCA over RPCA is that the CW-RPCA iterates over the chunk $\mathbf{X}_i$ only and not over $\left[\tilde{\boldsymbol{X}}_1; \tilde{\boldsymbol{X}}_2; \ldots; \mathbf{X}_i\right]$. At the same time, the CW-RPCA-based imputation of $\mathbf{X}_i$ takes into account the correlation structure characterising chunks $\tilde{\boldsymbol{X}}_1$ to $\tilde{\boldsymbol{X}}_{i-1}$.

4 Experiments

A simulation study is implemented to assess *(i)* how the performances of the CW-RPCA procedures compare to ordinary RPCA; *(ii)* how the performance of CW-RPCA compares to *naive* CW-RPCA. The methods in question are applied to a synthetic dataset with a fixed correlation structure and a benchmark sensor data set.

4.1 Simulation setup

To generate the synthetic data set, we partially refer to the simulation setup proposed by Dray and Josse (2015). Each data chunk is generated according to a block-wise correlation structure, with three blocks of 4, 3 and 2 attributes that are characterised by a correlation of $\rho = \{0.7, 0.75, 0.8\}$, respectively (see Fig. 1 left-hand side). The number of considered data chunks is 25, each characterised by 500 observations.

The considered missing data mechanisms are missing completely at random (MCAR) and missing not completely at random (MNCAR) (Rubin 1976). Under the MCAR mechanism, the probability of a missing entry does not depend on the attribute, or any attribute, value. In this scenario, 20% of entries are rendered missing for each attribute in each data chunk.

Under the MNCAR mechanism case, the missing values characterising a *target* attribute depend on the values of one (or more) *agent* attribute that may or may not be part of the considered data (Loisel and Takane 2019). Among the several ways of non-randomness in missing data (Josse et al. 2013), we refer to two nonresponse

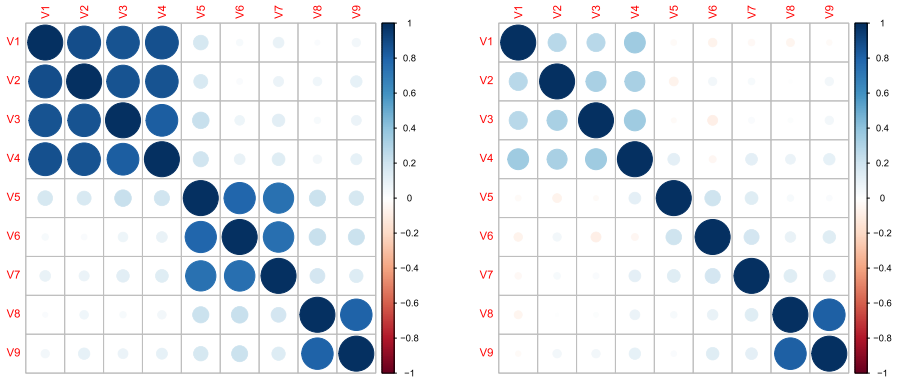

Fig. 1 Correlation structure of a complete data chunk (left); MNCAR mechanism undermining the correlation structure of the first two blocks of attributes of the same chunk

mechanisms: a logistic regression model-based mechanism and a correlation-based mechanism.

Logistic regression model-based mechanism. We consider two different scenarios: in the first one, the non response mechanism of the general attribute X depends on the attribute itself (in other words, the *target* and the *agent* coincide). Let X be the predictor in the logistic regression model, and the goal is to generate the binary response Y. When $Y_i = 1$, then the *ith* observation of X is rendered missing. We perform a grid search of plausible values for $\hat{\beta}_0$ and $\hat{\beta}_1$, and pick up a combination of values that leads to a target proportion of missings. The proportion of i's such that $Y_i = 1$ is set to 10% with a 2% tolerance. Formally, the missings in X depend on the values of X itself according to the logistic function

$$P(Y = 1 \mid X_i) = \frac{exp\ (\hat{\beta}_0 + \hat{\beta}_1 X_i)}{1 + exp\ (\hat{\beta}_0 + \hat{\beta}_1 X_I)}; \tag{16}$$

if $P(Y = 1 \mid X_i) > .5$ then $Y_i = 1$, and the *ith* value of X is rendered missing.

In the second scenario, a *target* attribute X_j is randomly chosen within each block of correlated attributes, and the *agents* are the other attributes from the same block of X_j . The rationale is the same as before, we fix a proportion of missings, and we perform a grid search for the values of $\hat{\boldsymbol{\beta}}$, the parameters vector. We use the multiple logit function

$$P(Y = 1 \mid \mathbf{X}_{-j}) = \frac{exp(\mathbf{X}_{-j}\hat{\boldsymbol{\beta}})}{1 + exp(\mathbf{X}_{-j}\hat{\boldsymbol{\beta}})} \tag{17}$$

where $\mathbf{X}_{-j}$ is the matrix of the agent attributes, that is, all the attributes in the block but j. If $P(Y = 1 \mid \mathbf{X}_{-j}(i)) > .5$ then $Y_i = 1$, and the i^{th} value of X_j is rendered missing.

Pairwise correlation-based mechanism. This scenario can be referred to as the *worst case*, since the nonresponse mechanism weakens the correlation between the attributes rendered missing. In particular, given a pair of centred attributes X_a and

X_b, let the *target* be either X_a or X_b; also, let the agent attribute X_{ab} be the product of X_a and X_b, that is $X_{ab} = X_a X_b$. Since X_a and X_b are centered, $mean(X_{ab}) = cov(X_a, X_b)$. The values of the target attribute (X_a or X_b) are rendered missing if the corresponding positions in X_{ab} are such that $X_{ab} \geq Q_3(X_{ab})$, where $Q_3(X_{ab})$ indicates the third quartile of the distribution of X_{ab}. A toy example is reported in Table 1, with X_a and X_b such that $corr(X_a, X_b) = 0.75$; the rendered missing target attribute is $X_a^\star$ and $Q_3(X_{ab}) = 0.81$: in this setting, the correlation is undermined by the presence of missing values; in fact, it results that $corr(X_a^\star, X_b) = 0.4$.

Under such MNCAR mechanism, up to three blocks of attributes contain missing values that hide the correlation structure: an illustration of an incomplete data chunk correlation structure, with missing entries in attributes from the first and second block, is reported in Fig. 1 (right-hand side).

The imputation error is used to assess the performance of RPCA and of both the implementations of CW-RPCA. In particular, the imputation error is given by the mean absolute difference between the *true* and *imputed* values. We decided to use the imputation error instead of the PCA parameter recovery (as in Loisel and Takane 2019) to have a more granular measure of performance.

4.2 MCAR experiment

The chunk-wise approach is compared to the batch approach to impute the MCAR data; in particular, the RPCA is applied to the data chunks as a whole (that is, up to a 12500×9 data set) using the `imputePCA` function from the R package `missMDA` (Josse and Husson 2016). The experiment runs over an increasing number of chunks, which is $\{5, 10, \ldots, 25\}$, and each analysis is repeated 20 times per number of chunks.

Figure 2 illustrates the results: the performance of *naive* CW-RPCA is comparable with the batch version of RPCA, whereas the non-*naive* implementation gives worse results, especially in terms of variability over the 20 replicates. The difference in performance between the two implementations of the CW-RPCA is

Table 1 Example of agent/target missing rendering: the agent attribute X_{ab} values highlighted are greater than $Q_3(X_{ab}) = 0.81$, $X_a^\star$ is the rendered missing version of X_a

X_a	X_b	X_{ab}	$X_a^\star$
− 0.20	0.47	− 0.09	− 0.20
− 0.89	− 0.93	0.83	NA
− 0.82	− 1.46	1.20	NA
0.46	1.62	0.75	0.46
0.85	− 0.12	− 0.10	0.85
− 0.13	− 0.69	0.09	− 0.13
− 0.32	− 0.04	0.01	− 0.32
− 0.69	− 0.25	0.17	− 0.69
− 0.62	− 0.20	0.12	− 0.62
2.36	1.60	3.78	NA

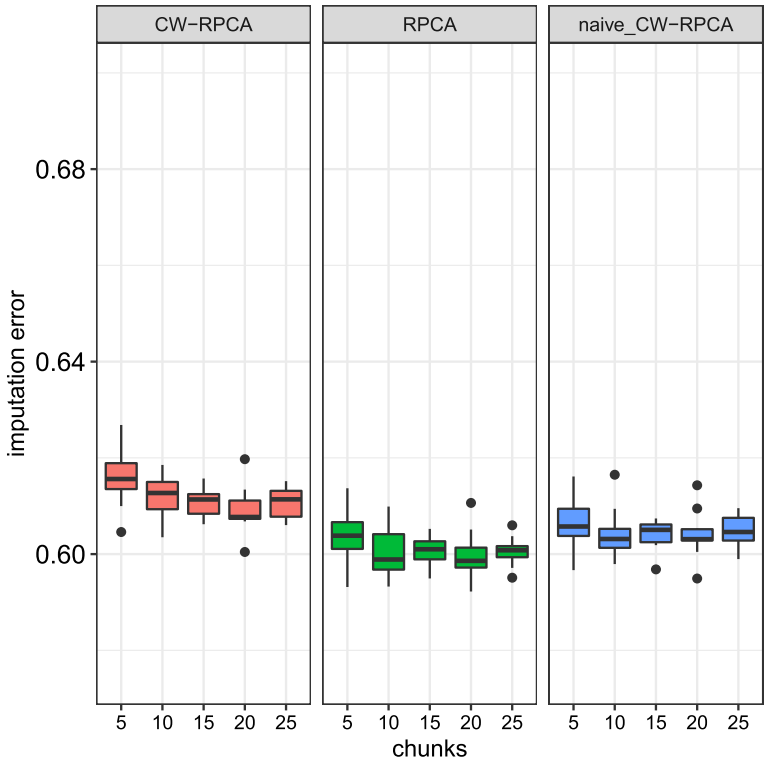

Fig. 2 Results on the MCAR considered scenario: imputation errors (mean absolute difference between the *true* and *imputed* values) over 20 replicates for 5 to 25 analysed chunks for the evaluated methods: CW-RPCA, iPCA and *naive* CW-RPCA

due to the MCAR mechanism. The underlying correlation structure is constant throughout the data chunks, and the missing entries do not alter it: therefore, the local, chunk-based, imputation obtained via *naive* CW-RPCA is just as good as the global imputation obtained via RPCA. In fact, each incomplete chunk contains information on the correlation structure that can be captured by CW-RPCA and used to impute the chunk itself properly. The non-*naive* CW-RPCA keeps track of the already processed data to impute the current chunk, and the results show that, in this case, such information is not needed.

4.3 MNCAR experiment: logistic regression model-based

The two logistic regression model-based scenarios are applied on 23 out of 25 chunks; the first two chunks are rendered missing using a MCAR mechanism, and their block-wise correlation structure is preserved (see Sect. 4.2). The right-hand side of Fig. 3 highlights the position of the MCAR chunks. The results for the two scenarios are reported in the left and right-hand side of Fig. 4, respectively. The results in scenario 1 show a substantially similar performance of CW-RPCA and RPCA. Furthermore, the higher the number of processed chunks, the more CW-

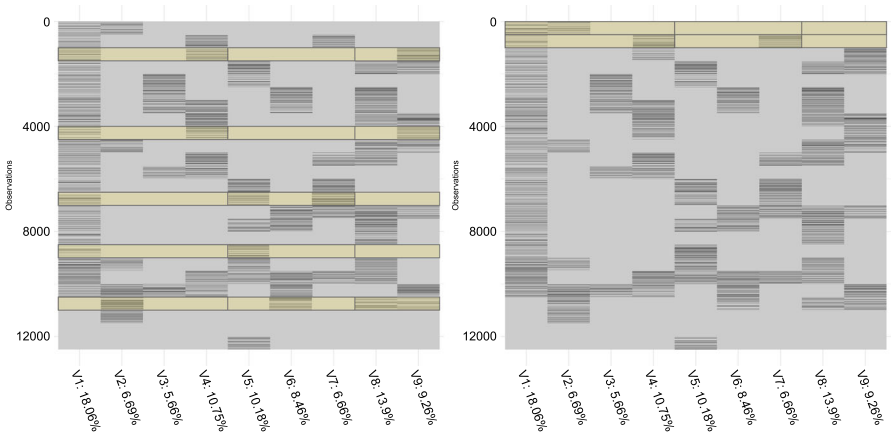

Fig. 3 MNCAR data structures: highlighted chunks are MCAR, with a preserved correlation structure. The left-hand side of the picture shows the scenario 1 of the correlation-based mechanism, with MCAR chunks randomly positioned in the data set; the right-hand side shows the logistic regression model-based scenarios as well as the scenario 2 of the correlation-based mechanism, with structured chunks positioned in the first rows of the considered data

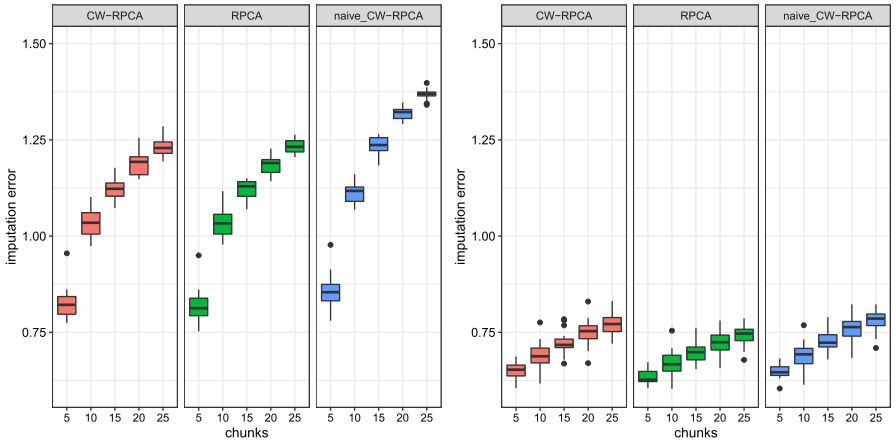

Fig. 4 Results on logistic regression model-based MNCAR scenario 1 (left-hand side) and 2: imputation errors (mean absolute difference between the *true* and *imputed* values) over 20 replicates for 5 to 25 analysed chunks for the evaluated methods: CW-RPCA, RPCA and *naive* CW-RPCA

RPCA and RPCA outperform the *naive* CW-RPCA. The results in scenario 2 show similar performance for the three methods: this is somewhat expected, and it depends on the missingness mechanism used. In fact, for each block of correlated attributes, just one of them is randomly selected and rendered missing: this results in a lower proportion of missing entries overall, and it is easier for each of the RPCA methods to capture the underlying correlation structure and impute the missing entries accordingly.

4.4 MNCAR experiment: correlation-based

For the correlation-based MNCAR mechanism experiment, two different scenarios are considered. In the first scenario, 5 out of 25 chunks contain missings according to an MCAR mechanism, whereas the other chunks contain MNCAR values (see Fig. 3, left-hand side). The rationale is that MCAR chunks preserve the correlation structure, as opposed to MNCAR chunks, whose structure is hidden away by missings: therefore, MCAR chunks make the CW-RPCA to *learn* the underlying data structure, which is an advantage when it comes to imputing incoming MNCAR data chunks. Of course, for the *naive* CW-RPCA that performs an independent chunk-wise imputation, no gain is expected.

The left-hand side of Fig. 5 shows the results referred to the first considered MNCAR scenario: the CW-RPCA slightly outperforms RPCA, albeit with an increased variability over the 20 replicates. The naive CW-RPCA shows a higher imputation error, albeit with less variability.

In the second MNCAR scenario, the first two chunks contain MCAR entries, and all the following chunks contain MNCAR entries (Fig. 3, right-hand side). The results are displayed in the right-hand side Fig. 5 and show how the CW-RPCA outperforms both the RPCA and *naive* CW-RPCA. This result confirms that, given a data set with a steady correlation structure, training the CW-RPCA on a complete data chunk, or on an incomplete data chunk whose missings do not hide away the correlation structure (as in the MCAR case), leads to improved results compared to the data set processed as a whole.

4.5 An application on the tennessee eastman problem dataset

The Tennessee Eastman Problem (TEP) data is a sensor data benchmark simulating an industrial chemical process (see, e.g., Severson et al. 2017). Indeed, PCA is

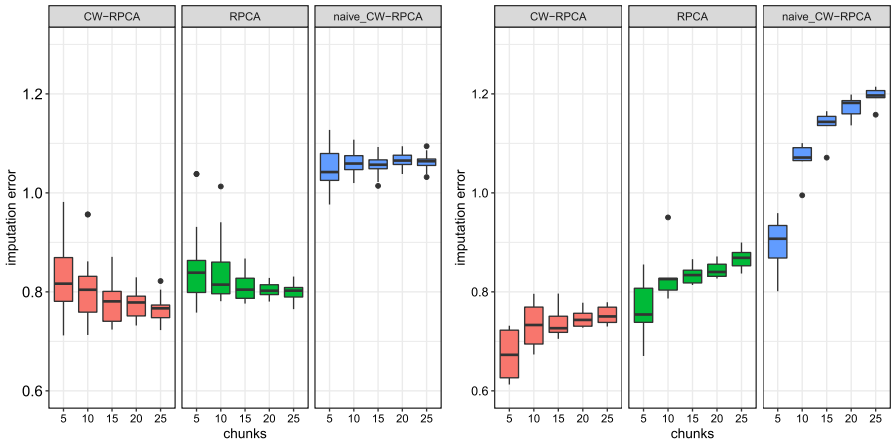

Fig. 5 Results on correlation-based MNCAR scenario 1 (left-hand side) and 2: imputation errors (mean absolute difference between the *true* and *imputed* values) over 20 replicates for 5 to 25 analysed chunks for the evaluated methods: CW-RPCA, RPCA and *naive* CW-RPCA

successfully applied to process data as a tool for multivariate process control. Missing values in process control data are fairly common, and they might be due to sensor failures, for example. The benchmark data we refer to is available in `.RData` format (Rieth et al. 2017) and it consists of observations of both normal operation of the process and different failures. The attributes mostly (52 out of 55) refer to sensors that monitor the process in question. While the faulty process detection is beyond the scope of this paper, we used the fault-free training data set that contains a total of 250 thousand observations.

In a pre-processing phase, we selected a subset of 24 attributes: in particular, only one attribute from each collinear pair; furthermore, the attributes with limited to none correlation were discarded. The pre-processing phase led to a mildly defined correlation structure depicted in the left-hand side of Fig. 6. The same missing data mechanism described in Sect. 4.1 was used to generate the missings that hide away the underlying correlation structure, see the right-hand side of Fig. 6. The proportion of missing values per attribute ranges from 0 to above 50%, see Fig. 7.

Consistently with the experiment described in Sect. 4.4, we considered a chunk-size of 500 observations and an increasing number of analysed chunks; more specifically, the number of chunks is such that $I_{ch} \in \{5, 10, 15, 20, 25\}$. The total number of observations in each scenario is, therefore, $500 \times I_{ch}$.

An appraisal of the CW-RPCA performance compared to naive CW-RPCA and RPCA was carried out with regard to *(i)* the imputation error, as shown in the previous section, and *(ii)* the accuracy of the obtained PCA solution. The second considers on the RV coefficient (Escoufier 1973), calculated to assess the similarity between the PCA solutions obtained on the data with and without missings. In particular, the RV coefficient measures the *closeness* of two configurations, and it can be considered as a multivariate generalisation of Pearson's correlation coefficient, ranging from 0 to 1 (Robert and Escoufier 1976). Let $\mathbf{F}$ and $\mathbf{F}^{\star}$ be the $I \times d$ matrices of object scores resulting from the PCA on the data with and without missings, respectively. Then the RV coefficient is

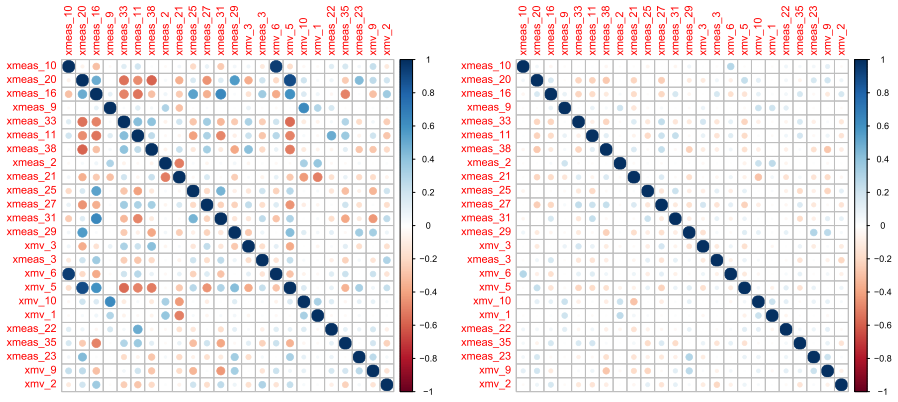

Fig. 6 Tennessee Eastman Problem dataset complete data correlation structure (left) vs correlation structure with missings

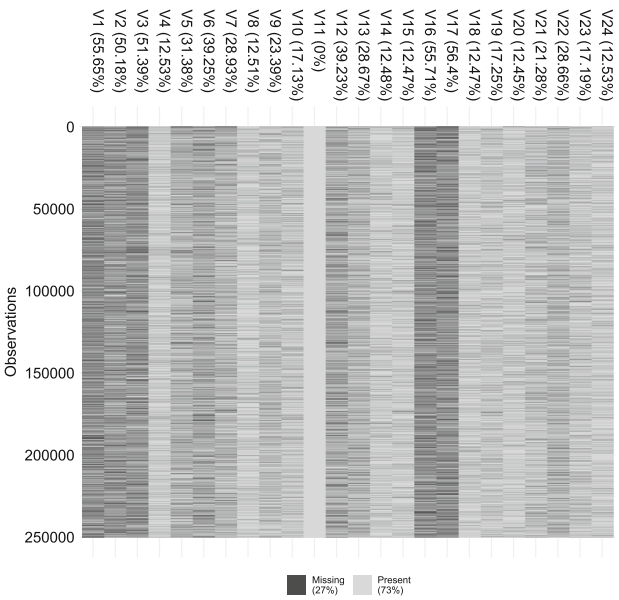

Fig. 7 Tennessee Eastman Problem dataset with missings: 27.2% average proportion of missings, yet single attributes have up to 56% of missings

$$\mathrm{RV}\left(\mathbf{F}, \mathbf{F}^{\star}\right) = \frac{tr\left(\mathbf{F}^{\mathsf{T}}\mathbf{F}\ \mathbf{F}^{\star\mathsf{T}}\mathbf{F}^{\star}\right)}{tr\left(\mathbf{F}^{\mathsf{T}}\mathbf{F}\right)^{2} tr\left(\mathbf{F}^{\star\mathsf{T}\mathbf{F}^{\star}}\right)^{2}}; \tag{18}$$

When $\mathrm{RV}\left(\mathbf{F}, \mathbf{F}^{\star}\right) = 1$ it means that the two configurations of points are superimposed, i.e., the PCA solutions on data with and without missing entries coincide.

The first chunk of observations is characterised by an MCAR mechanism that is not detrimental to the data correlation structure, whereas each further chunk comes from the MNCAR version of the TEP dataset. The imputation error results, referring to 20 replicates of the experiment, are reported in Fig. 8 (top): a similar pattern for the three methods is evident, with the imputation error increasing with the number of analysed chunks. Such a result is not surprising as the proportion of observations with defined correlation structure (that is, observations from the MCAR chunk) decreases as more MNCAR chunks are analysed. Furthermore, CW-RPCA is characterised by a lower imputation error compared to RPCA and, more so, to the naive CW-RPCA. The RV's for observations and attributes are depicted in the bottom left and right Fig. 8: the results confirm that the CW-RPCA outperforms the other methods in terms of parameter recovery.

Analogously to previous experiments, the CW-RPCA *learns* the correlation structure by analysing the first MCAR chunk and takes it into account when processing the forthcoming chunks. Instead, the RPCA processes the observations as a whole, and so the correlation structure information carried by the MCAR chunks is diluted in the full set of observations. Finally, the naive CW-RPCA that processes each chunk independently has the highest imputation error. The RV index

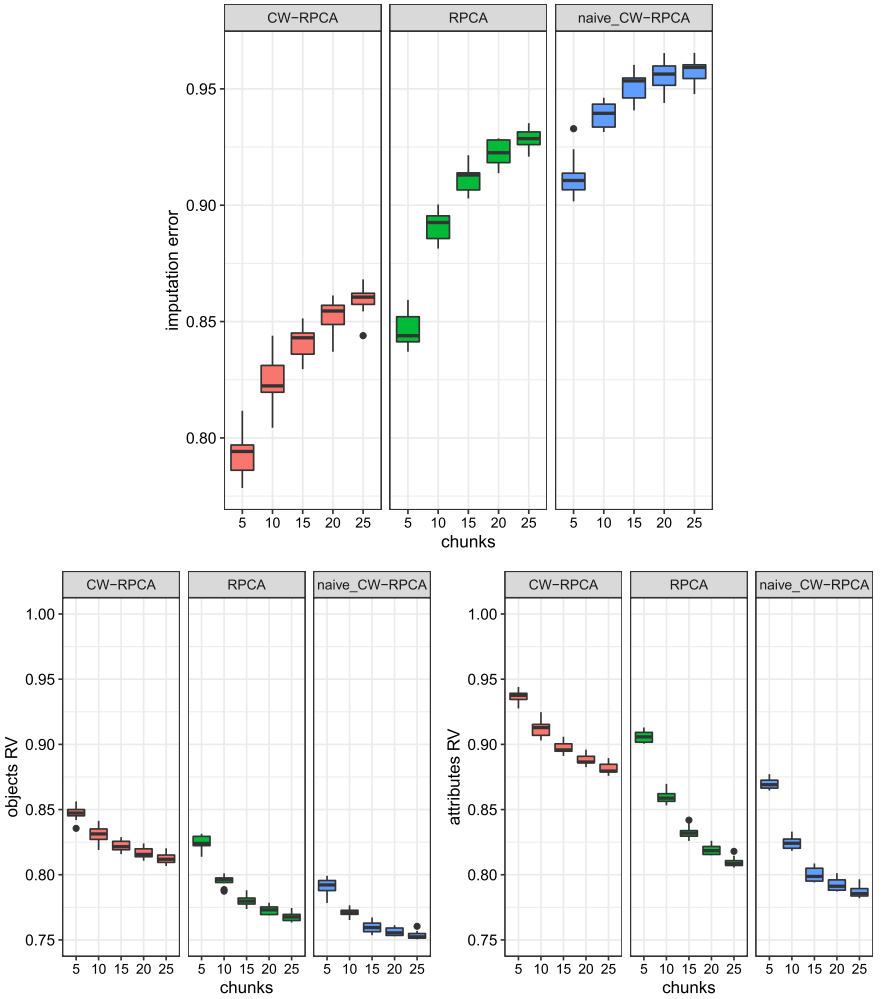

Fig. 8 Results on TEP data: imputation errors (top) (mean absolute difference between the *true* and *imputed* values) over 20 replicates for 5–25 analysed chunks for CW-RPCA, RPCA and *naive* CW-RPCA; RV index for observations (bottom-left) and attributes (bottom-right) PCA scores

results confirm the behaviour of the methods, with the CW-RPCA showing a better performance in terms of solution recovery: a further aspect to point out is that the methods performance decays as the number of chunks increases, but such decay is lower in the CW-RPCA case.

5 Conclusion and future work

This work presented a chunk-wise extension of RPCA for data sets with missings. The general idea is grounded on the imputation of the missing entries of a chunk using the low-rank structure of all the chunks insofar analysed, together with the current one. The performance of CW-RPCA was compared with RPCA on the full data set and with a *naive* version of CW-RPCA. The *naive* approach consists in applying RPCA on each chunk and then simply merge the chunk-based RPCA solutions (Iodice D'Enza et al. 2018).

The results of the MCAR experiment on synthetic data sets with a well-defined correlation structure showed that the *naive* CW-RPCA performed better than CW-RPCA and similar to RPCA on the full data set, as expected. Each MCAR chunk had a correlation structure similar to the full data set. Therefore, tracking the low-rank structure of the analysed chunks did not provide an imputation performance gain; in fact, it was detrimental. The logistic regression model-based MNCAR scenarios can be referred to as MAR, since the non response mechanism of a attribute depends on the values of the attribute itself, or of other observed attributes. In scenario 1, CW-RPCA and RPCA had similar performance, and both outperformed the naive implementation of CW-RPCA. This is not surprising since the naive CW-RPCA learns the correlation structure using a single chunk, as opposed to CW-RPCA that learns the structure from all the previously processed chunks, and to RPCA that learns the structure from all the available observations. In the scenario 2 the three methods performed equally well because the proportion of entries rendered missing was limited: since one attribute per block was considered as target, with the other attributes from the same block being agents.

The pairwise correlation-based MNCAR scenarios can be considered *worst case* since the missing entries alter the correlation structure, and it is realistic in some application domains, such as sensor data. The CW-RPCA has shown appreciable performance, mainly when the first analysed chunks were informative about the correlation structure (Sects. 4.4 and 4.5).

While different non response mechanism have been considered, other MNCAR mechanisms may be at work, depending on the application domain. As pointed out by Severson et al. (2017), the choice of the best method to apply PCA on data with missings may depend on the missing data mechanism, the proportion of missings, and the available computational resources. While the latter two aspects are easily determined, the identification of the missing data mechanism is non-trivial: it involves the determination of why some data are missing. Recent approaches proposed by Geraci and Farcomeni (2016) and Sportisse et al. (2018) aim to model the MNCAR mechanisms explicitly.

We suggest two main directions for future research: *(i)* generalize the application of CW-RPCA to categorical and mixed-type data sets, by embedding in a chunk-based setting PCA-related methods such as multiple correspondence analysis (MCA, Greenacre 2017), and factor analysis of mixed data (FAMD, Pagès 2004); *(ii)* discuss the CW-RPCA model selection procedure and evaluate its performance

in case of attributes on different scales.[3]

Funding Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

References

Balzano L, Chi Y, Lu YM (2018) Streaming pca and subspace tracking: The missing data case. Proc IEEE 106(8):1293–1310

Benzécri JP (1973) L'analyse des données. L'analyse des correspondances, Dunod, Tome II

Borgognone MG, Bussi J, Hough G (2001) Principal component analysis in sensory analysis: covariance or correlation matrix? Food Qual Preference 12(5–7):323–326

Bro R, Kjeldahl K, Smilde AK, Kiers HAL (2008) Cross-validation of component model: a critical look at current methods. Analy Bioanal Chem 390:1241–1251

Cardot H, Degras D (2018) Online principal component analysis in high dimension: which algorithm to choose? Int Stat Rev 86(1):29–50

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soci Ser B 39(1):1–38

Dray S, Josse J (2015) Principal component analysis with missing values: a comparative survey of methods. Plant Ecol 216(5):657–667

Eckart C, Young G (1973) The approximation of one matrix by another of lower rank. Psychometrika 1:211–218

Escoufier Y (1973) Le traitement des variables vectorielles. Biometrics pp 751–760

Folch-Fortuny A, Arteaga F, Ferrer A (2015) PCA model building with missing data: new proposals and a comparative study. Chemom Intell Lab Syst 146:77–88

Geraci M, Farcomeni A (2016) Probabilistic principal component analysis to identify profiles of physical activity behaviours in the presence of non-ignorable missing data. J R Stat Soc Ser C (Appl Stat) 65(1):51–75

Geraci M, Farcomeni A (2018) Principal component analysis in the presence of missing data. Advances in Principal Component Analysis. Springer, New York, pp 47–70

Golub GH, Van Loan CF (2012) Matrix computations, vol 3. JHU Press, Maryland

Gower JC (1971) Statistical methods of comparing different multivariate analyses of the same data. Mathematics in the archaeological and historical science. pp 138–149

Greenacre M (2017) Correspondence analysis in practice. Chapman and Hall/CRC, NewYork

Greenacre MJ (2010) Biplots in practice. Fundacion BBVA, Spain

Hall P, Marshall D, Martin R (2002) Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition. Image Vis Comput 20(13–14):1009–1016

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction, 2nd edn. Springer, NewYork

Hegde A, Principe JC, Erdogmus D, Ozertem U, Rao YN, Peddaneni H (2006) Perturbation-based eigenvector updates for on-line principal components analysis and canonical correlation analysis. J VLSI Signal Process Syst Signal Image Video Technol 45(1–2):85–95

[3] The GitHub repository available at https://github.com/amarkos/CW-RPCA-Experiments contains the supplementary material needed to replicate the experiments described in Sect. 4. Basic guidelines to replicate the experiments can be found in the `Supplementary_script`, available in both .html and .Rmd formats

Ilin A, Raiko T (2010) Practical approaches to principal component analysis in the presence of missing values. J Mach Learn Res 11:1957–2000
Iodice D'Enza A, Markos A, Buttarazzi D (2018) The idm package: incremental decomposition methods in R. J Stat Softw Code Snippets 86(4):1–24
Jolliffe IT (2002) Principal Component Analysis, 2nd edn. Springer-Verlag, New York
Josse J, Husson F (2012) Handling missing values in exploratory multivariate data analysis methods. J Soc Fr Stat 153(2):79–99
Josse J, Husson F, Pagès J (2009) Gestion des données manquantes en analyse en composantes principales. J Soci Fr Stat 150(2):28–51
Josse J, Timmerman ME, Kiers HA (2013) Missing values in multi-level simultaneous component analysis. Chemom Intell Lab Syst 129:21–32
Josse J, Husson F et al (2016) missMDA: a package for handling missing values in multivariate data analysis. J Stat Softw 70(1):1–31
Kiers HA (1997) Weighted least squares fitting using ordinary least squares algorithms. Psychometrika 62(2):251–266
Levey A, Lindenbaum M (2000) Sequential karhunen-loeve basis extraction and its application to images. IEEE Trans Image Process 9(8):1371–1374
Little RJ, Rubin DB (2019) Statistical analysis with missing data. John Wiley & Sons, Hoboken
Loisel S, Takane Y (2019) Comparisons among several methods for handling missing data in principal component analysis (PCA). Adv Data Anal Classif 13(2):495–518
Markos A, Iodice D'Enza A (2018) A framework for the incremental update of the MCA solution. Ital J Appl Stat 29(2–3):217–231
Navarrete P, Ruiz-del-Solar J (2002) Analysis and comparison of eigenspace-based face recognition approaches. Int J Pattern Recognit Artif Intell 16(07):817–830
Pagès J (2004) Analyse factorielle de données mixtes. Revue de Stat Appl 52(4):93–111
Rieth CA, Amsel BD, Tran R, Cook MB (2017). Additional Tennessee Eastman process simulation data for anomaly detection evaluation. https://doi.org/10.7910/DVN/6C3JR1
Robert P, Escoufier Y (1976) A unifying tool for linear multivariate statistical methods: the RV-coefficient. Appl Stat 25(3):257–265
Ross DA, Lim J, Lin RS, Yang MH (2008) Incremental learning for robust visual tracking. Int J Comput Vis 77(1–3):125–141
Rubin DB (1976) Inference and missing data. Biometrika 63(3):581–592
Schafer JL (1997) Analysis of incomplete multivariate data. Chapman and Hall/CRC, New York
Severson KA, Molaro MC, Braatz RD (2017) Principal component analysis of process datasets with missing values. Processes 5(3):38
Sportisse A, Boyer C, Josse J (2020) Imputation and low-rank estimation with Missing Not At Random data. Stat Comput 30(6):1629–1643
Takane Y, Oshima-Takane Y (2003) Relationships between two methods for dealing with missing data in principal component analysis. Behaviormetrika 30(2):145–154
Van Ginkel JR, Kroonenberg PM, Kiers HAL (2014) Missing data in principal component analysis of questionnaire data: a comparison of methods. J Stat Comput Simul 84(11):2298–2315