




Article

Techniques for Robust Imputation in Incomplete Two-Way Tables

Sergio Arciniegas-Alarcón ¹, Marisol García-Peña ^{2,*}, Camilo Rengifo ¹ and Wojtek J. Krzanowski ³

¹ Facultad de Ingeniería, Universidad de La Sabana, Campus Puente del Común, Km. 7 Autopista Norte, Chía 140013, Colombia; sergio.arciniegas@unisabana.edu.co (S.A.-A.); camilo.rengifo@unisabana.edu.co (C.R.)

² Departamento de Matemáticas, Pontificia Universidad Javeriana, Carrera 7 40-62, Bogotá 110231, Colombia

³ College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, UK; w.j.krzanowski@exeter.ac.uk

* Correspondence: luzmara@gmail.com

Abstract: We describe imputation strategies resistant to outliers, through modifications of the simple imputation method proposed by Krzanowski and assess their performance. The strategies use a robust singular value decomposition, do not depend on distributional or structural assumptions and have no restrictions as to the pattern or missing data mechanisms. They are tested through the simulation of contamination and unbalance, both in artificially generated matrices and in a matrix of real data from an experiment with genotype-by-environment interaction. Their performance is assessed by means of prediction errors, the squared cosine between matrices, and a quality coefficient of fit between imputations and true values. For small matrices, the best results are obtained by applying robust decomposition directly, while for larger matrices the highest quality is obtained by eliminating the singular values of the imputation equation.

Keywords: imputation; missing values; singular value decomposition; genotype-by-environment interaction



Citation: Arciniegas-Alarcón, S.; García-Peña, M.; Rengifo, C.; Krzanowski, W.J. Techniques for Robust Imputation in Incomplete Two-Way Tables. *Appl. Syst. Innov.* **2021**, *4*, 62. <https://doi.org/10.3390/asi4030062>

Academic Editor: Friedhelm Schwenker

Received: 13 July 2021

Accepted: 16 August 2021

Published: 4 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Complete data matrices are required for some statistical analysis techniques, making the imputation of missing values necessary in certain circumstances. For example, the biplot analysis used widely in multivariate exploratory analysis [1] and the additive main effects and multiplicative interaction models—AMMI [2]—used to describe the interaction between genotypes and environments, have as main tool the singular value decomposition—SVD [3]. However, this SVD is not directly calculable if there are missing values [4], and it is necessary to pre-process the information by first replacing the missing data with plausible values.

Modern literature on incomplete information analysis recommends completing matrices using methods that employ either maximum likelihood or multiple imputation [5]. However, these procedures can depend heavily not only on probability distributions (for example, multivariate normal), but also on the missing data mechanisms [6].

There are currently four missing data mechanisms [7]: Missing Completely at Random (MCAR), where the missing data does not depend on the feature variable being considered or any of the other feature variables in the dataset; Missing at Random (MAR), where the missing data in the feature variable is conditional on any other feature variable in the dataset but not on the one being considered; Missing not at Random (MNAR); when the possibility of a feature variable having a missing data entry depends on the value of the feature variable itself (irrespective of any alteration or modification to the values of other feature variables in the dataset); and Missing by Natural Design (MBND), where the missing data occurs because it cannot be measured physically but is relevant in the data analysis procedure [7].

To take into account both dependence on probability distributions and missing data mechanisms, a very useful option is non-parametric imputation [8,9]. A general

method free of structural and distributional assumptions was originally proposed by Krzanowski [10] and recently generalized by Arciniegas-Alarcón et al. [11] to complete matrices from multi-environmental experiments. The method depends on the SVD, which is a least squares technique, but a disadvantage of these types of techniques is that they are highly influenced by untypical values or outliers [12].

To the best of our knowledge, this method has not yet been subjected to a robustness study, so our purpose here is to evaluate its performance in the face of outlier observations and to propose some strategies for robust imputations. We therefore first present an updated version of the SVD method, then consider robust alternatives for imputation, and finally describe numerical studies for the corresponding performance analysis.

2. Material and Methods

2.1. Missing Value Imputation Using the SVD–Method SVD88

The method consists of an updated version of the imputation system of Krzanowski [10] with some minor changes reported in the literature that improve its performance [11]. Consider a matrix Y ($n \times p$) with elements y_{ij} ($i = 1, \dots, n$; $j = 1, \dots, p$) and $p > n$ (if $p < n$ the matrix should be first transposed). First, suppose there is just one missing value y_{ij} in Y . Then, the i -th row from Y is deleted and the SVD for the $((n-1) \times p)$ resulting matrix $Y^{(-i)}$ is calculated as $Y^{(-i)} = \bar{U}\bar{D}\bar{V}^T$, $\bar{U} = (\bar{u}_{sh})$, $\bar{V} = (\bar{v}_{sh})$, $\bar{D} = (\bar{d}_1, \dots, \bar{d}_p)$. The next step is to delete the j -th column from Y and obtain the SVD for the $(n \times (p-1))$ matrix $Y_{(-j)}$ as $Y_{(-j)} = \tilde{U}\tilde{D}\tilde{V}^T$, $\tilde{U} = (\tilde{u}_{sh})$, $\tilde{V} = (\tilde{v}_{sh})$, $\tilde{D} = (\tilde{d}_1, \dots, \tilde{d}_{p-1})$. The matrices \bar{U} , \bar{V} , \tilde{U} and \tilde{V} are orthonormal, while \tilde{D} and \bar{D} are diagonal. Now, combining the two SVDs, $Y^{(-i)}$ and $Y_{(-j)}$, the imputed value is given by:

$$\hat{y}_{ij} = \sum_{h=1}^H \tilde{u}_{ih} \left(\tilde{d}_h \sqrt{p/(p-1)} \right)^{\frac{1}{2}} \bar{v}_{jh} \left(\bar{d}_h \sqrt{n/(n-1)} \right)^{\frac{1}{2}} \quad (1)$$

where H is the optimal number of components of the SVD that can be found by cross-validation methods adapted for matrices with missing data and available in the R statistical environment [13]. In this work, the “bcv” package was used, which has implemented cross-validation for incomplete matrices using an EM algorithm ([13]; available from CRAN; see <https://cran.r-project.org/web/packages/bcv/index.html>, accessed on 25 June 2021).

When there is more than one missing value, an iterative scheme is required as follows. Initially, all missing values are replaced by their respective column means, giving a completed matrix Y and then the columns are standardized by subtracting m_j and dividing the result by s_j (where m_j and s_j represent the mean and the standard deviation of the j -th column calculated only from the observed values). Using the standardized matrix, the imputation for each missing value is recalculated using the Equation (1). Finally, the matrix Y is returned to its original scale, $y_{ij} = m_j + s_j \hat{y}_{ij}$. Then, the process is iterated until stability is achieved in the imputations (i.e., when the values in two successive iterations agree to within a desired level of accuracy). In order to avoid convergence problems, a parity check should be done in each iteration by matching the sign of $\left(\tilde{u}_{ih} \left(\tilde{d}_h \sqrt{p/(p-1)} \right)^{\frac{1}{2}} \right) \left(\bar{v}_{jh} \left(\bar{d}_h \sqrt{n/(n-1)} \right)^{\frac{1}{2}} \right)$ in Equation (1) to the sign of $u_{ih} d_h v_{jh}$ obtained from the SVD of the Y matrix for each $h = 1, \dots, H$ [11].

Pseudocode

- i. Y <- incomplete matrix.
- ii. Y^* <- initial completed matrix (columns mean and standardisation).
- iii. \hat{y}_{ij} <- Imputation update (Equation (1)), $\forall (i, j)$ with missing values.
- iv. Return to step ii. until stability is achieved in the imputations.

2.2. Robust Singular Value Decomposition

The presence of outliers in a data set can reduce the effectiveness of least squares techniques [12], which in this case is the standard SVD. To avoid this behavior, robust lower-rank approximations or equivalent robust SVD could be used. Fortunately, the literature provides several alternatives that can be considered.

Recently, [14] presented a review of four options for producing robust SVD. The first is an approximation of alternating regressions based on a mixture of M estimators and trimmed least squares estimators. For more details on these types of regressions, see [15]. The second consists of a sub-sampling method whose objective is to find a sub-matrix of rows without outliers with high probability and from this sub-matrix to estimate the effects of rows and columns by means of a robust function. For more details see [16]. The third option is to use iterative reweighted least square algorithms to obtain robust estimators in principal component analysis (PCA). A complete description of this approach and extensions can be found in [17] and a computational implementation is available in the R RobRSVD package [13]. The last option is an optimization with restrictions in which the data matrix is assumed to be a sum of three matrices, one of low rank, a sparse matrix to represent the outliers, and a random noise matrix. Algorithms for such optimization and additional details can be found in [14].

Other very useful work on robust SVD has been carried out by [18], who used alternating robust fittings with trimmed least squares to find the SVD and apply it in microarray data analysis; Maronna and Yohai [19] who used alternating M regressions to obtain robust low-rank approximations; and Rodrigues et al. [20,21], who used the SVD proposed by Hawkins et al. [22] for robust singular spectral analysis and for a robust version of AMMI models. The computational implementation of this SVD can be found in the R pcaMethods package ([13]; pcaMethods available from Bioconductor; <https://bioconductor.org/packages/release/bioc/html/pcaMethods.html>, accessed on 25 June 2021).

Given this literature review and taking our objectives into account, we delimit the research and choose an option to produce a robust SVD that can use the SVD88 method. To focus specifically on matrices with genotype-by-environment interaction, we adopt the suggestion of [20]. Thus, we use the Hawkins et al. [22] SVD which is based on an alternating L1 regression algorithm (rSVD01), and the generalization proposed by [23] using weighted least absolute deviation regression (rSVD10). The frequent use of alternating regressions in the literature cited above obliged us to also consider the classic reference of Gabriel and Odoroff [24], whose main objective was to produce resistant low-rank matrices (rSVD84).

We thus initially focused on comparing the performances of rSVD01, rSVD10, and rSVD84 when they are introduced into the SVD88 imputation system. However, their individual performance was also considered, i.e., without introducing them into the imputation system. This is because the SVD from alternating regressions does not need any data imputation as it uses only the observed information. Moreover, a robust low-rank approximation can provide estimates of the missing positions of an incomplete matrix [18] without any additional calculations. The three SVDs were tested, and in the case of rSVD01 and rSVD10, we also tested replacing L1 regressions with M regressions using the rlm function of R ([13]; RobRSVD available from CRAN; see <https://cran.r-project.org/web/packages/RobRSVD/index.html>, accessed on 25 June 2021). These preliminary tests used contaminated and incomplete matrices of different dimensions, starting with matrices of dimension 20×5 and continuing to matrices with many more elements, for example, dimension 100×8 . In these initial experiments, the method that was statistically simplest, computationally most efficient, and with the best results when introduced into the SVD88 was always rSVD84, so it is the method described below.

2.3. rSVD84 (Procedure Based on García-Peña et al., 2021 and Gabriel and Odoroff 1984)

Consider a dimension Y matrix ($n \times p$) with possible missing entries. (i) Using the observed information, calculate the vectors of trimmed means (i.e., the means when a

certain percentage of the most extreme observations are ignored; we use 10% or 20% following [25]) by columns and by rows, $\mathbf{b}(1 \times p)$ and $\mathbf{a}(n \times 1)$ respectively. (ii) Determine the presence of outliers in vector \mathbf{a} by any technique that is preferred for univariate outliers (for example, the quartile method) and if there is any discrepant data replace it with a trimmed mean of the elements of \mathbf{a} . (iii) Update the elements of \mathbf{b} and \mathbf{a} as follows: $b_c = \text{med}\{|y_{r,c}/a_r|; r = 1, \dots, n\}$ and $a_r = \text{med}\{|y_{r,c}/b_c|; c = 1, \dots, p\}$. Repeat steps (ii) and (iii) with the updated vectors \mathbf{b} and \mathbf{a} until some convergence criteria are attained. Once the stability of vectors \mathbf{b} and \mathbf{a} has been achieved, a robust low-rank approximation of \mathbf{Y} is obtained by means of the \mathbf{ab}^T product whose element signals must be the same as those of \mathbf{Y} . To obtain the first singular value and the first right and left singular vector of the robust SVD, the standard SVD over \mathbf{ab}^T is applied and both the first singular value and the first right and left singular vector are recorded. To obtain the second singular value and the second right and left vector, the same procedure described in this section is done, but replacing the matrix \mathbf{Y} with $\mathbf{Y} - \mathbf{ab}^T$. This cyclic strategy deflating the matrix continues until the desired number of robust SVD components-rSVD84 is obtained. In the Appendix A of this manuscript a function is provided in R [13] that calculates the rSVD84 for any complete or incomplete matrix.

Pseudocode

- i. $\mathbf{Y} \leftarrow$ incomplete matrix
- ii. $\mathbf{Y}^* \leftarrow$ initial completed matrix (columns mean)
- iii. Calculate a robust singular value decomposition on \mathbf{Y}^* Section 2.3 (rSVD).
- iv. The rSVD contains the require imputations.

2.4. Alternatives to Obtain Robust Imputation

The SVD88 algorithm depends directly on the standard SVD, so the presence of outliers can affect the performance and decrease the quality of the imputations. To avoid this behavior, we have three possible strategies: (i) Substitute in the SVD88 iterations the classic robust standardization, in which the column standardization is done by subtracting the median M_j and dividing the result by the median absolute deviation-MAD_j. Having thus standardized the matrix, use rSVD84 until stability in the imputations is obtained and then return the matrix to the original scale; (ii) Find a robust low-rank approximation to the incomplete matrix using rSVD84 and then take the values obtained in that approximation as imputations for those matrix positions that were not originally obtained. (iii) Perform the imputation in two stages, first obtaining a robust low-rank approximation of the incomplete matrix and then applying the SVD88 method to refine the imputations.

Some additional implementation details for these options are as follows. In option (i) we tried to obtain a robust version of SVD88 but using the robust standardization and rSVD84 together did not perform well and in preliminary tests it never achieved convergence. However, if we eliminated the singular values of Equation (1) of imputation then convergence was achieved. This robust version of the SVD88 is called M5RobSVD. Option (ii) or rSVD84 has always shown in preliminary tests to be the fastest but for both this and M5RobSVD an appropriate number of components to be used in the imputation must be specified. To do this, the robust SVD was calculated on the incomplete matrix and k was chosen such that the sum of the first k ordered singular squared values divided by the sum of all singular squared values was greater than 0.75 [11]. Finally, option (iii) depends on the sequential application of two procedures, rSVD84 and SVD88. This sequence of procedures in the preliminary tests has always reached convergence. In view of the original authors of the procedures, we call the imputation methods GOKimputation and M5GOKimputation when the singular values were disregarded.

Pseudocode for M5RobSVD

- i. $\mathbf{Y} \leftarrow$ incomplete matrix
- ii. $\mathbf{Y}^* \leftarrow$ completed matrix (columns means and divide by MAD_j).
- iii. $\hat{y}_{ij} \leftarrow$ Imputation update (Equation (1)), eliminate \tilde{a}_h and $\tilde{b}_h, \forall (i, j)$ with missing values.
- iv. Return to step ii. until stability is achieved in the imputations,

2.5. Numerical Example 1: Simulation Study

To evaluate the performance of the five imputation systems (SVD88, M5RobSVD, rSVD84, GOKimputation, and M5GOKimputation), we simulated eight incomplete and contaminated 100×8 size matrices (Table 1) that had the structure of an AMMI model, using the steps given by [20,26] as follows:

1. Create a matrix X with $n = 100$ rows and $p = 8$ columns with observations drawn from a uniform distribution $[-0.5; 0.5]$.
2. Compute the SVD of X to obtain the matrices U , V and D , containing, respectively, the left and right singular vectors and the singular values of X .
3. Simulate the grand mean (μ), the row (α) and column (β) main effects, where $\mu \sim N(15, 3)$, $\alpha \sim N(5, 1)$ and $\beta \sim N(8, 2)$.
4. Simulate a matrix Y with AMMI2 structure (i.e., AMMI with the first two singular values/vectors):

$$Y = \mathbf{1}_n \mathbf{1}_p^T \mu + \alpha_n \mathbf{1}_p^T + \mathbf{1}_n \beta_p^T + 28 \times U[:, 1] D[1, 1] V[:, 1]^T + 15 \times U[:, 2] D[2, 2] V[:, 2]^T$$

where $A[:, i]$ represents i -th column of the matrix A ($i = 1, 2$), $A[i, i]$ represents the element in the row i and column i of the matrix A , $\mathbf{1}_n \mathbf{1}_p^T \mu$ is a matrix ($n \times p$) with the grand mean μ in all positions, $\alpha_n \mathbf{1}_p^T$ is a matrix ($n \times p$) of rows main effects (all matrix rows equal), $\mathbf{1}_n \beta_p^T$ is a matrix ($n \times p$) of columns main effects (all matrix columns equal).

Table 1. Characteristics of the simulated matrices.

| Matrices | % of Missing | % of Outliers |
|----------|--------------|---------------|
| Matrix1 | 10 | 0 |
| Matrix2 | 20 | 0 |
| Matrix3 | 10 | 5 |
| Matrix4 | 10 | 10 |
| Matrix5 | 10 | 15 |
| Matrix6 | 20 | 5 |
| Matrix7 | 20 | 10 |
| Matrix8 | 20 | 15 |

For each complete Y matrix, observations were then randomly deleted at the specified percentages shown in Table 1 and outliers were produced in the remaining data at the companion percentages in Table 1. The positions of the outliers were chosen randomly, and the outliers were generated from $N(\mu_j + 5\sigma_j^2, \sigma_j^2)$, where μ_j and σ_j^2 represent the mean and variance of the j -th column (or j -th environment) of the non-missing values. The five imputation procedures were applied to the eight incomplete and contaminated matrices, the five imputation procedures were applied, having previously recorded the true data that had been randomly removed. This made it possible to assess the performance of the imputations by calculating the prediction error P_e defined by [4] as the square root of the mean squared error (MSE) between the true values and the corresponding imputations.

2.6. Numerical Example 2: Cross-Validation on Real Data from Experiments with Genotype-by-Environment Interaction

The Ontario winter wheat dataset with 18 genotypes in nine environments was published by [27] to show an application of biplot analysis, and recently these data were used to illustrate new proposals for distribution-free multiple imputation [9] and new bootstrap methods to determine the optimal number of multiplicative components in AMMI models [28].

A difficulty in evaluating the behavior of robust imputation systems in complete real data is the lack of a priori knowledge of what “good behavior” should be. For this reason and following the recommendation of [19], we consider adding a small contamination of

the matrix elements and removing some elements. Initially, 10% of the elements of each environment were removed, but unlike the previous simulation study, we used an MNAR mechanism. To obtain non-random deletions, we obtained the 10th percentile of each column ($P10_j; j = 1, \dots, 9$) and any observation that was less than $P10_j$ was eliminated to obtain an incomplete matrix. On the resulting matrix, 10% of positions chosen at random were contaminated following the same contamination process used in the simulation study mentioned above.

When the “Ontario” matrix was incomplete and contaminated, each of the values present in the matrix was eliminated and imputed with the robust strategies presented from the remaining elements. Thus, a complete matrix of imputations for each method was obtained by cross-validation and could be compared with the original matrix using P_e [4]. In addition to the prediction error, we added two more statistics proposed by Gabriel [29] to identify the best method. One of these statistics was the coefficient $GF_1 = 1 - \|O - I\|^2 / \|O\|^2$, where O is the true “Ontario” matrix and I is the imputation matrix obtained by cross-validation. The other statistic was $GF_2 = \cos^2(O, I) = \frac{\text{tr}^2(O^T I)}{\text{tr}(O^T O) \text{tr}(I^T I)}$. The GF_2 statistic will always be in the range $[0, 1]$ and the closer it is to 1, the better are the imputations. On the other hand, GF_1 cannot be greater than GF_2 , but it can take negative values which indicate that the imputations are of lower quality than if matrix I were the zero matrix.

This second numerical study was also used to compare our new robust proposals with a method that has had good results in data science, the missForest method [8]. missForest is an imputation method based on random forests (predictors that consist of a collection of randomized regression trees) that has a high versatility as it is nonparametric, it can be applied to continuous, categorized or mixed data (categorized and continuous) and it has a computational implementation in the statistical language R [30,31].

3. Results and Discussion

Table 2 shows the results of the first numerical example. When data were incomplete (10% and 20% of missing values), but without contamination, the SVD88 algorithm always obtained the smallest prediction errors. This behavior is as expected, since without outliers the procedures based on least squares generally perform well [12]. This situation changes completely when there is contamination of the data with 5, 10 and 15% of outliers. In these cases, it can be seen that as the number of outliers increases, the SVD88 prediction errors also increase. Thus, SVD88 is highly affected by contamination, producing low quality imputations. It can also be seen that in these contaminated scenarios the four proposed strategies of robust imputation always surpass SVD88 in terms of P_e . It remains, then, to analyze in detail the performance of the robust imputations.

In general, using the mean and median of the prediction errors, M5RobSVD provides the best quality of imputations when there are outliers because it minimizes the values of such statistics. However, at the highest imputation and contamination percentages, (20% and 15%, respectively) the M5GOKimputation strategy marginally outperforms M5RobSVD. Moreover, the remaining two strategies, rSVD84 and GOKimputation, are also effective in the presence of outliers because they had lower P_e values than SVD88, but in no case were they able to improve on M5RobSVD or M5GOKimputation. These results show that the elimination of the singular values of the imputation equation within a robust procedure favors convergence and improves the quality of the imputations.

Table 3 shows the results of the second numerical example based on the “Ontario” matrix. It is noteworthy that the four strategies proposed in this article outperformed both the SVD88 and the missForest method, obtaining GF_1 and GF_2 values close to 1 and minimizing P_e . In this cross-validation exercise on real data, SVD88 presented the worst performance with very poor-quality imputations, which can be concluded from the negative value in the GF_1 statistic. This situation can be explained by the lack of convergence in the imputation of some observations.

Table 2. Results of numerical example 1.

| Percentage of Missing Values and Outliers | | | Imputation Method | | | | |
|---|-------|------|-------------------|----------------|----------------|----------------|------------------|
| | | | SVD88 | M5RobSVD | rSVD84 | GOK Imputation | M5GOK Imputation |
| Matrix | %miss | %out | P _e | P _e | P _e | P _e | P _e |
| 1 | 10 | 0 | 0.9721 | 4.4655 | 4.5945 | 4.5903 | 4.4889 |
| 2 | 20 | 0 | 0.7570 | 3.9584 | 4.4752 | 4.4670 | 4.0461 |
| 3 | 10 | 5 | 17.5018 | 3.2894 | 3.8948 | 3.8883 | 3.3602 |
| 4 | 10 | 10 | 36.1932 | 3.2856 | 3.6275 | 3.6235 | 3.4482 |
| 5 | 10 | 15 | 46.3526 | 4.3479 | 4.4465 | 4.4396 | 4.3283 |
| 6 | 20 | 5 | 24.0821 | 4.5323 | 5.1784 | 5.1639 | 4.5994 |
| 7 | 20 | 10 | 36.8853 | 4.0614 | 4.7675 | 4.7563 | 4.1501 |
| 8 | 20 | 15 | 59.2625 | 4.6295 | 4.8285 | 4.8033 | 4.4290 |
| Mean | | | 27.7508 | 4.0713 | 4.4766 | 4.4665 | 4.1063 |
| Median | | | 30.1377 | 4.2047 | 4.5349 | 4.5286 | 4.2392 |
| Standard deviation | | | 20.9011 | 0.5342 | 0.5036 | 0.4993 | 0.4688 |
| Interquartile distance | | | 25.8827 | 0.6911 | 0.4742 | 0.4663 | 0.5473 |

Table 3. Results of numerical example 2.

| Method | GF ₂ | GF ₁ | P _e |
|-----------------|-----------------|-----------------|----------------|
| SVD88 | 0.3202 | −2.0001 | 7.4491 |
| M5RobSVD | 0.9836 | 0.9812 | 0.5891 |
| rSVD84 | 0.9843 | 0.9821 | 0.5756 |
| GOKimputation | 0.9842 | 0.9819 | 0.5788 |
| M5GOKimputation | 0.9833 | 0.9808 | 0.5958 |
| missForest | 0.8730 | 0.3719 | 3.4083 |

On considering the GF₂ and GF₁ statistics, the four methods M5RobSVD, rSVD84, GOKimputation and M5GOKimputation showed values of approximately 0.98 with very small differences only in the third and fourth decimal places. This indicates that the imputations are all high quality and the performance difference among these methods is very small. On the other hand, the prediction error was minimized using rSVD84, so taking into account the values of all three statistics then rSVD84 can be chosen in this case, because it is the fastest computationally robust method.

The numerical results presented so far show that our four robust imputation alternatives (M5RobSVD, rSVD84, GOKimputation and M5GOKimputation) work very well and in the case of data contamination the SVD88 algorithm should be replaced by one of them. The reader may therefore be left with the question: which method to choose according to the results obtained in this study? answer this question, outliers must first be detected and for this reason the “cellWise” package from R ([13]; cellWise available from CRAN; <https://cran.r-project.org/web/packages/cellWise/index.html>, accessed on 25 June 2021) is recommended. Once outliers are detected [32], the dimension of the matrix under analysis must be taken into account. If the matrix is small (for example 18 × 9) rSVD84 can be applied, but if the matrix is larger (for example 100 × 8), the M5RobSVD and M5GOKimputation systems can be considered. In the latter case, if the imputation and contamination percentages were greater than 20 and 10% respectively, it is suggested to test both algorithms. From a computational point of view, in large matrices (100 × 8) M5GOKimputation presents greater speed if compared to M5RobSVD. All the proposals in this article are based on a robust SVD, for that reason in the Appendix A function of R is presented to calculate the rSVD84.

Finally, it is worth answering one more question that may arise for the applied researcher when analyzing contaminated and unbalanced data: what is the maximum percentage of contamination that can be tolerated if an effective imputation of missing values is to be made? There is a lot of literature on the two problems and very useful

references can be found in the recent study by [32]. However, we suggest following a rationale derived from a simulation study described by [33] and from the results obtained by [4]. In a real situation, algorithms based on SVD can work with up to 60% of missing values [4], but if there is a high proportion of outliers in the remaining data, imputations will necessarily be affected by them and will produce low quality of results [33]. Taking this into account, we recommend only considering imputation on incomplete matrices that still have at least half of the observed data without outliers. For example, one rule of thumb might be to add the percentages of missing data and outliers, and only proceed if that sum is not greater than 40%. This will ensure that at least 60% of “clean” data can be effectively used for the purpose of completing the matrix. With less than 60% of “clean” data any imputation results should be treated with caution.

4. Conclusions

In this paper, we have focused on nonparametric imputation based on the singular value (SVD) decomposition [10,11]. SVD is a least-squares technique, and in the presence of outliers it is known that least-squares imputation methods can produce low quality imputations [12]. The main aims of our study were therefore:

1. To consider robust versions of the method that will allow for outliers.
2. To investigate the robustness of all the proposed methods, in what we believe to be the first extensive such robustness study.
3. To provide advice as to which specific method to use in a given practical application requiring imputation.

Our conclusions regarding these aims can be summarized as follows. The basic method, denoted SVD88 in the text, performs well in the absence of outliers but performance deteriorates markedly as the number of outliers increases. Imputations in all cases are indeed of low quality. Four possible robust versions of SVD88 were identified and studied. All improved substantially on SVD88 so in the case of contaminated data one of them should be used instead of SVD88. In a practical application, if the matrix is small then rSVD84 is recommended, but for larger matrices either of M5RobSVD or M5GOK would be preferable. If computational speed is needed, then the latter should be chosen. As for the maximum number of missing or contaminated observations to allow, a suggested rule of thumb is to add the percentages of missing and outlier observations and to proceed only if the sum is not greater than 40%.

Finally, it is worth stressing that the robust singular value decomposition is a flexible tool that allows us to generate imputations resistant to outliers, and strategies have been proposed here that perform well without the use of any structural or distributional assumptions. The proposed imputation schemes can be safely applied in experiments with genotype-by-environment interaction and, being very general, they can be widely used in any research area that has an incomplete two-way or multivariate matrix.

Author Contributions: Conceptualization, S.A.-A., M.G.-P., C.R., and W.J.K.; methodology, S.A.-A., M.G.-P., C.R., and W.J.K.; software, S.A.-A., M.G.-P., C.R., and W.J.K.; validation, S.A.-A., M.G.-P., C.R., and W.J.K.; formal analysis, S.A.-A., M.G.-P., C.R., and W.J.K.; investigation, S.A.-A., M.G.-P., C.R., and W.J.K.; resources, S.A.-A., M.G.-P., C.R., and W.J.K.; data curation, S.A.-A., M.G.-P., C.R., and W.J.K.; writing—original draft preparation, S.A.-A., M.G.-P., C.R., and W.J.K.; writing—review and editing, S.A.-A., M.G.-P., C.R., and W.J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors of this paper acknowledge the High-Performance Computing Center–ZINE of Pontificia Universidad Javeriana for assistance during the simulation study.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. R Functions to Calculate Robust SVD (Procedure Based on García-Peña et al., 2021)

```
###rm(list=ls(all=TRUE)) # Deletes all content from memory
###install.packages("StatMeasures")
library(DescTools)
rSVD84SAM<-function(WojtekOrig){
  indicamissing<-is.na(WojtekOrig)# Indicates the occurrence of missing in the matrix
  posicobservados<-which(indicamissing != 1, TRUE) # Matrix of positions with observed
  totalobservados<-nrow(posicobservados)
  library(StatMeasures)
  ### Creation of trimmed means for rows and columns with missing and complete
  TrimmedColMeansWojtekOrig <- apply(WojtekOrig, 2, mean, trim=0.1, na.rm=T)
  TrimmedRowMeansWojtekOrig<-apply(WojtekOrig, 1, mean, trim=0.2, na.rm=T)
  trimmedmean<-mean(WojtekOrig, trim=0.2, na.rm=T)
  TrimmedRowMeansWojtekOrig[is.na(TrimmedRowMeansWojtekOrig)] <- trimmedmean
  if(outliers(TrimmedRowMeansWojtekOrig)$numOutliers>0){
    trimmedmean.out2<-mean(TrimmedRowMeansWojtekOrig, trim=0.2, na.rm=T)
    num.outliers2<-outliers(TrimmedRowMeansWojtekOrig)$numOutliers
    pos.outliers2<-outliers(TrimmedRowMeansWojtekOrig)$idxOutliers
    for(o in 1:num.outliers2){
      TrimmedRowMeansWojtekOrig[pos.outliers2[o]]<-trimmedmean.out2
    }
    #print(o)
  }### End of for(o in 1:num.outliers){
}### End of if(outliers(TrimmedColMeansWojtekOrig)$numOutliers>0){
###Inicial vector a SVD84
epsilon<-1*10**(-5)
stabilitycrit<-1
iter<-0
RSS_A0<-0
RSS_B0<-0
oldB<-TrimmedColMeansWojtekOrig
oldA<-TrimmedRowMeansWojtekOrig
  while (stabilitycrit>epsilon & iter<=100){
a_s1_matrix<-matrix(1,nrow(WojtekOrig),ncol(WojtekOrig))*TrimmedRowMeansWojtekOrig
as_1_inverso<-1/a_s1_matrix
Y<-abs(WojtekOrig*as_1_inverso)
b_sj<-apply(Y, 2, quantile, probs=0.5, na.rm=T)
b_sj_matrix<-t(b_sj*matrix(1,ncol(WojtekOrig),nrow(WojtekOrig)))
b_sj_inverso<-1/b_sj_matrix
Z<-abs(WojtekOrig*b_sj_inverso)
RSS_B<-sum((b_sj-oldB)**2)
stabilitycritB<-abs(RSS_B-RSS_B0)
TrimmedRowMeansWojtekOrig<-apply(Z, 1, quantile, probs=0.5, na.rm=T)
trimmedmean2<-mean(Z, trim=0.2, na.rm=T)
TrimmedRowMeansWojtekOrig[is.na(TrimmedRowMeansWojtekOrig)] <- trimmedmean2
RSS_A<-sum((oldA-TrimmedRowMeansWojtekOrig)**2)
stabilitycritA<-abs(RSS_A-RSS_A0)
stabilitycrit<-max(stabilitycritA,stabilitycritB)
if(outliers(TrimmedRowMeansWojtekOrig)$numOutliers>0){
  trimmedmean.out2<-mean(TrimmedRowMeansWojtekOrig, trim=0.2, na.rm=T)
```

```

num.outliers2<-outliers(TrimmedRowMeansWojtekOrig)$numOutliers
pos.outliers2<-outliers(TrimmedRowMeansWojtekOrig)$idxOutliers
for(o in 1:num.outliers2){
  TrimmedRowMeansWojtekOrig[pos.outliers2[o]]<-trimmedmean.out2
  #print(o)
}### End of for(o in 1:num.outliers){
}### End of if(outliers(TrimmedColMeansWojtekOrig)$numOutliers>0){
oldA<-TrimmedRowMeansWojtekOrig
oldB<-b_sj
RSS_B0<-RSS_B
RSS_A0<-RSS_A
iter<-iter+1
} ### End of while
lower.rank.approx<-oldA%*%t(oldB)
for (k in 1:totalobservados){
  if(WojtekOrig[posicobservados[k,1],posicobservados[k,2]]<0
    & lower.rank.approx[posicobservados[k,1],posicobservados[k,2]]>0){
    lower.rank.approx[posicobservados[k,1],posicobservados[k,2]]<
-lower.rank.approx[posicobservados[k,1],posicobservados[k,2]] * (-1)}
  if(WojtekOrig[posicobservados[k,1],posicobservados[k,2]]>0
    & lower.rank.approx[posicobservados[k,1],posicobservados[k,2]]<0){
    lower.rank.approx[posicobservados[k,1],posicobservados[k,2]]<
-lower.rank.approx[posicobservados[k,1],posicobservados[k,2]] * (-1)}
} ### End of for (k in 1:totalobservados){
list(rSVD84.iter=iter,
     rSVD84.converA=stabilitycritA,
     rSVD84.converB=stabilitycritB,
     rSVD84.approx=lower.rank.approx,
     rSVD84.A=oldA,
     rSVD84.B=oldB
)
} ### End of rSVD84SAM<-function(WojtekOrig){
robustSvdGO<-function(X){
svdu <- matrix(NA, nrow=nrow(X), ncol=ncol(X))
svdv <- matrix(NA, nrow=ncol(X), ncol=ncol(X))
svdd <- rep(NA, ncol(X))
  for(k in 1:ncol(X)) {
    rob.approx<-rSVD84SAM(X)
    rsvd<-svd(rob.approx$rSVD84.approx)
    ## Deflate the X matrix
    X <- (X - rob.approx$rSVD84.approx)
    svdu[,k] <- rsvd$u[,1]
    svdv[,k] <- rsvd$v[,1]
    svdd[k] <- rsvd$d[1 ]
    #print(k)
    #print(rsvd$iter)
    #print(rsvd$stabilization)
  }###End of for(k in 1:ncol(X))
total.variability<-sum(svdd^2)
cumuled.variability<-cumsum((svdd^2)/total.variability)
Calinski.crit1<-which(cumuled.variability > 0.70)[1]
cumuled.variability2<-cumsum((sort(svdd^2, decreasing=T))/total.variability)
Calinski.crit2<-which(cumuled.variability2 > 0.70)[1]
Calinski.crit<-min(Calinski.crit1,Calinski.crit2)

```

```

## Create the result object
ret2 <- list()
ret2$d <- svdd
ret2$u <- svdu
ret2$v <- svdv
ret2$Calinski99 <- Calinski.crit
return(ret2)
}###End of robustSvdGO<-function(X){

```

References

- Greenacre, M. *Biplots in Practice*; Fundación BBVA: Bilbao, Spain, 2010.
- Gauch, H.G. A Simple Protocol for AMMI Analysis of Yield Trials. *Crop. Sci.* **2013**, *53*, 1860–1869. [CrossRef]
- Greenacre, M.J. Biplots: The joy of singular value decomposition. *WIREs Comput. Stat.* **2012**, *4*, 399–406. [CrossRef]
- Yan, W. Biplot Analysis of Incomplete Two-Way Data. *Crop. Sci.* **2013**, *53*, 48–57. [CrossRef]
- Van Ginkel, J.R.; Linting, M.; Rippe, R.C.A.; Van Der Voort, A. Rebutting existing misconceptions about multiple imputation as a method for handling missing data. *J. Personal. Assess.* **2019**, *102*, 297–308. [CrossRef]
- Paderewski, J.; Rodrigues, P.C. The usefulness of EM-AMMI to study the influence of missing data pattern and application to Polish post-registration winter wheat data. *Aust. J. Crop. Sci.* **2014**, *8*, 640–645.
- Leke, C.A.; Marwala, T. *Deep Learning and Missing Data in Engineering Systems*; Springer International Publishing: Berlin, Germany, 2019; p. 179.
- Stekhoven, D.J.; Bühlmann, P. MissForest—Non-parametric missing value imputation for mixed-type data. *Bioinformatics* **2012**, *28*, 112–118. [CrossRef]
- Arciniegas-Alarcón, S.; Dias, C.T.S.; García-Peña, M. Distribution-free multiple imputation in incomplete two-way tables. *Pesqui. Agropecuária Bras.* **2014**, *49*, 683–691. [CrossRef]
- Krzanowski, W.J. Missing value imputation in multivariate data using the singular value decomposition of a matrix. *Biom. Lett.* **1988**, *25*, 31–39.
- Arciniegas-Alarcón, S.; García-Peña, M.; Krzanowski, W.J. Missing value imputation in multi-environment trials: Reconsidering the Krzanowski method. *Crop. Breed. Appl. Biotechnol.* **2016**, *16*, 77–85. [CrossRef]
- Alkan, B.B.; Atakan, C.; Alkan, N. A comparison of different procedures for principal component analysis in the presence of outliers. *J. Appl. Stat.* **2015**, *42*, 1716–1722. [CrossRef]
- R Core Team. R: A Language and Environment for Statistical Computing. Available online: <http://www.R-project.org/> (accessed on 28 May 2020).
- Feng, X.; He, X. Robust low-rank data matrix approximations. *Sci. China Ser. A Math.* **2017**, *60*, 189–200. [CrossRef]
- Maronna, R.A.; Martin, R.D.; Yohai, V.J.; Salibián-Barrera, M. *Robust Statistics: Theory and Methods (with R)*; Wiley: Hoboken, NJ, USA, 2019; p. 464.
- Feng, X.; He, X. Statistical inference based on robust low-rank data matrix approximation. *Ann. Stat.* **2014**, *42*, 190–210. [CrossRef]
- Zhang, L.; Shen, H.; Huang, J.Z. Robust regularized singular value decomposition with application to mortality data. *Ann. Appl. Stat.* **2013**, *7*, 1540–1561. [CrossRef]
- Liu, L.; Hawkins, D.M.; Ghosh, S.; Young, S.S. Robust singular value decomposition analysis of microarray data. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 13167–13172. [CrossRef]
- Maronna, R.A.; Yohai, V.J. Robust Low-Rank Approximation of Data Matrices with Elementwise Contamination. *Technometrics* **2008**, *50*, 295–304. [CrossRef]
- Rodrigues, P.C.; Monteiro, A.; Lourenço, V.M. A robust AMMI model for the analysis of genotype-by-environment data. *Bioinformatics* **2016**, *32*, 58–66. [CrossRef]
- Rodrigues, P.; Lourenço, V.; Mahmoudvand, R. A robust approach to singular spectrum analysis. *Qual. Reliab. Eng. Int.* **2018**, *34*, 1437–1447. [CrossRef]
- Hawkins, D.M.; Liu, L.; Young, S.S. Robust Singular Value Decomposition. *Natl. Inst. Stat. Sci.* **2001**, *122*, 1–13.
- Jung, K.M. Robust singular value decomposition based on weighted least absolute deviation regression. *Commun. Korean Stat. Soc.* **2010**, *17*, 803–810.
- Gabriel, K.R.; Odoroff, L. Resistant lower rank approximation of matrices. In *Data Analysis and Statistics III*; Diday, E., Jambu, M., Lebart, L., Thomassone, Eds.; North-Holland: Amsterdam, The Netherlands, 1984; pp. 23–30.
- García-Peña, M.; Arciniegas-Alarcón, S.; Krzanowski, W.J.; Duarte, D. Missing-value imputation using the robust singular-value decomposition: Proposals and numerical evaluation. *Crop. Sci.* **2021**, 1–13. [CrossRef]
- Arciniegas-Alarcón, S.; García-Peña, M.; Rodrigues, P.C. New multiple imputation methods for genotype-by-environment data that combine singular value decomposition and Jackknife resampling or weighting schemes. *Comput. Electron. Agric.* **2020**, *176*, 105617. [CrossRef]
- Yan, W.; Kang, M.S.; Ma, B.; Woods, S.; Cornelius, P.L. GGE biplot vs. AMMI analysis of genotype-by-environment data. *Crop. Sci.* **2007**, *47*, 641–653. [CrossRef]

-
28. Forkman, J.; Piepho, H.P. Robustness of the simple parametric bootstrap method for the additive main effects and multi-plicative interaction (AMMI) model. *Biul. Oceny Odmian* **2015**, *34*, 11–18.
 29. Gabriel, K.R. Goodness of fit of biplots and correspondence analysis. *Biometrika* **2002**, *89*, 423–436. [[CrossRef](#)]
 30. Stekhoven, D.J. MissForest: Nonparametric Missing Value Imputation Using Random Forest; R Package, Version 1.4. 2013. Available online: <https://cran.r-project.org/web/packages/missForest/missForest.pdf> (accessed on 25 June 2021).
 31. Tang, F.; Ishwaran, H. Random forest missing data algorithms. *Stat. Anal. Data Min. ASA Data Sci. J.* **2017**, *10*, 363–377. [[CrossRef](#)] [[PubMed](#)]
 32. Rousseeuw, P.; Bossche, W.V.D. Detecting Deviating Data Cells. *Technometrics* **2018**, *60*, 135–145. [[CrossRef](#)]
 33. Serneels, S.; Verdonck, T. Principal component analysis for data containing outliers and missing elements. *Comput. Stat. Data Anal.* **2008**, *52*, 1712–1727. [[CrossRef](#)]