Predicting Missing Values with Biclustering:  
a Coherence-Based Approach

Fabrício Olivetti de França\textsuperscript{a}, Guilherme Palermo Coelho\textsuperscript{b,c},  
Fernando J. Von Zuben\textsuperscript{c}

\textsuperscript{a}Center of Mathematics, Computing and Cognition (CMCC), Federal University of ABC  
(UFABC) – Santo André, SP, Brazil.
\textsuperscript{b}Laboratory of Natural Computing (LCoN-FT), School of Technology (FT),  
University of Campinas (Unicamp) – Limeira, SP, Brazil.
\textsuperscript{c}Laboratory of Bioinformatics and Bioinspired Computing (LBiC), Department of  
Computer Engineering and Industrial Automation (DCA), School of Electrical and  
Computer Engineering (FEEC),  
University of Campinas (Unicamp) – Campinas, SP, Brazil.

Abstract

In this work, a novel biclustering-based approach to data imputation is proposed. This approach is based on the Mean Squared Residue metric, used to evaluate the degree of coherence among objects of a dataset, and presents an algebraic development that allows the modeling of the predictor as a quadratic programming problem. The proposed methodology is positioned in the field of missing data, its theoretical aspects are discussed and artificial and real-case scenarios are simulated to evaluate the performance of the technique. Additionally, relevant properties introduced by the biclustering process are also explored in post-imputation analysis, to highlight other advantages of the proposed methodology, more specifically confidence estimation and interpretability of the imputation process.

Keywords: Biclustering, Missing Data Imputation, Knowledge Discovery, Quadratic Programming

Email addresses: olivetti@ieee.org (Fabrício Olivetti de França),  
guilherme.coelho@ieee.org (Guilherme Palermo Coelho),  
vonzuben@dca.fee.unicamp.br (Fernando J. Von Zuben)
1. Introduction

In Data Mining, researchers often have to face the problem of working with missing values, which may be due to a variety of reasons such as an incomplete acquisition of data or the presence of noise. Therefore, the information available in these situations tends to be limited and many applications generally require the prediction of such missing values (i.e., collaborative filtering, bioinformatics).

According to the application and to the amount of missing values in the dataset, different approaches can be adopted [1]: (i) the missing values can be simply ignored, when they correspond to a small percentage of the data or when the amount of data available is large; or (ii) when the missing information is really necessary and corresponds to a high percentage of the dataset, machine learning techniques can be used to predict those values [2, 3, 4]. When missing data is caused by known and predictable events, the techniques employed to impute the missing values are generally based on likelihood estimations, although sometimes alternative methods are also adopted to deal with such situations [1].

Imputation methods try to fill the gaps (missing data) in datasets according to different approaches, being the three most popular in the literature the global imputation on missing attributes, the global imputation on non-missing attributes and the local imputation.

The global imputation on missing attributes generally uses metrics of frequency or central tendency to infer the missing values. For example, the most frequent value of a specific attribute may be used to fill the missing values. However, although these techniques are very simple, they do not perform very well when compared to other more elaborate approaches, as pointed out in [5].

The global imputation on non-missing attributes corresponds to a set of techniques that explore the correlations between the missing and non-missing values, and solve the problem with the adoption of linear and logistic regressions. These techniques assume that the regression function reflects the model of the data being studied, and they may suffer from overfitting [1].

Considering the local imputation techniques, the most famous approach is known as Hot-Deck Imputation [6], which fills the missing values with values from a complete and similar dataset of the same group. This method first divides the whole dataset into classes, using one of the complete attributes, and then it fills the missing values of each subset by taking non-missing values
from subsets of the same class, following a specified rule or applying global
imputation techniques.

Most of the data imputation techniques investigate the whole set of at-
tributes in order to identify a similar profile between incomplete and complete
objects. However, according to the nature of the attributes, the profiles of
complete and incomplete objects may only partially match. This can be ex-
emplified by a dataset of ratings given by a set of users to a set of movies.
Each user has rated only a small subset of movies, and now it is required
that the missing ratings are predicted. It is possible to group those movies
into several subsets (as by genre, for example) instead of considering all of
them as a single large set. By properly clustering the whole set of movies, it
may be possible to obtain more coherent groups of user profiles, since each
user may take part in distinct groups when distinct subsets of movies are
considered. To illustrate, user A may be together with user B, and not with
user C, in a specific group due to the fact that both A and B have similar
judgments for action movies, and the same user A will be together with user
C, and not with user B, when another genre of movies is considered.

A data mining and clustering technique that is capable of considering
these multi-faceted correlations, thus allowing deeper inferences from the
available data, is the so called biclustering technique [7, 8]. Considering
that the datasets are structured as data matrices of objects (rows) and their
corresponding attributes or features (columns), the biclustering technique is
capable of finding several subsets of rows and columns of the data matrix,
so that each subset will be composed of objects that share some similarities
specifically on the selected attributes. In this approach, a single object may
take part in multiple biclusters, with distinct subsets of attributes in each
of them, thus allowing the extraction of additional information from the
dataset.

The local knowledge that is acquired from a bicluster can be interesting
when trying to predict a missing value. However, up to now this charac-
teristic was only exploited in an over-simplified way in most papers, with
the transformation of the data into binary values and the application of a
k-NN (k-nearest neighbor) algorithm [9] or with the use of a memory-based
approach on the resulting biclusters to predict the missing values [10, 11].
Although these methods presented good results, they still do not properly
exploit the knowledge highlighted by the biclusters, as they do not consider
neither the overall structure nor the information used during the construction
of the biclusters.
In this work, the proposed methodology is meant to predict missing values contained in coherent biclusters – more specifically, biclusters constructed according to the mean squared residue (MSR) metric\(^1\) [8] – by extracting the information inherently highlighted by such approach. The proposed method considers the missing values as variables in the MSR equation and explicitly minimizes this metric, which leads to a quadratic programming (QP) problem. The solution of such QP problem can be seen as an attempt to minimize the noise introduced by the missing data, as the imputed values are adjusted to fit the coherent model of the bicluster in the best possible way (in the ideal situation, the resulting bicluster would be perfectly coherent, which can be interpreted as a noise-free mathematical model) with the property of generating a QP problem with unique solution and polynomial time computability.

It is important to notice that the novel approach proposed here for data imputation is independent of both the methodology to solve the resulting QP problem and the biclustering method, as long as it generates coherent biclusters according to Cheng & Church’s MSR metric [8]. However, in order to show the practical results obtained with the proposed method, it was adopted in this work a slightly modified version of the biclustering technique known as SwarmBCluster [12], which was combined with the new imputation technique.

This paper is organized as follows. Section 2 introduces the main concepts of biclustering and the coherence metric that is the basis of the proposed methodology; in Section 3, the technique for predicting the missing values within a bicluster is presented, including all the algebraic manipulation to achieve the QP problem and the properties that the biclusters must have to improve the prediction performance. In Section 4, experiments performed on artificial and real-world datasets will be presented and discussed. After that, the possibility of evaluating the confidence of the imputations made by the proposed technique will be highlighted. Finally, in Section 5, some concluding remarks will be outlined.

\(^1\)It is important to highlight that only the MSR coherence metric was used in this work and not the biclustering algorithm proposed by Cheng & Church [8].
2. Biclusters of datasets

The term biclustering, also known as co-clustering and two-way clustering, is associated with the process of finding subsets of rows and columns of a given data matrix [7, 8, 13], which may represent different kinds of numerical data, such as objects (rows) and their attributes (columns). The biclustering approach can be applied to a wide scope of different applications such as information retrieval and text mining [14, 15], electoral data analysis [7], collaborative filtering [9, 11], and biological data analysis [16, 17].

Cheng & Church [8] were responsible for the popularization of the biclustering paradigm with their CC algorithm, which was the first biclustering algorithm applied to gene expression problems. The CC algorithm searches for subsets of rows and columns of a dataset, called biclusters, so that each row/column of the bicluster presents a profile identical to the one exhibited by other rows/columns, except for a constant bias. By following this pattern, each element from this bicluster can be expressed by:

\[ a_{i,j} = a_{I,j} + a_{i,J} - a_{I,J}, \]  

where \( a_{i,j} \) is the element at row \( i \) and column \( j \) of the bicluster, \( a_{i,J} \) is the mean of elements in row \( i \), \( a_{I,j} \) is the mean of elements in column \( j \) and \( a_{I,J} \) is the mean of all elements in the bicluster.

Since a bicluster that perfectly matches such pattern can be difficult to find in noisy data, Cheng & Church [8] searched for biclusters that minimized the error between the theoretical value (the value obtained when a perfectly coherent bicluster is considered) given in Eq. 1 and the real value of an element of the matrix. This squared error was called mean-squared residue \( H_{I,J} \) of a bicluster:

\[ H_{I,J} = \frac{1}{n'm'} \sum_{i \in I} \sum_{j \in J} r_{i,j}^2, \]  

where

\[ r_{i,j} = a_{i,j} - a_{I,j} - a_{i,J} + a_{I,J}, \]

\[ a_{I,j} = \frac{1}{n'} \sum_{i \in I} a_{i,j}, \quad a_{i,J} = \frac{1}{m'} \sum_{j \in J} a_{i,j}, \quad a_{I,J} = \frac{1}{n'm'} \sum_{i \in I} \sum_{j \in J} a_{i,j}. \]
\[ n' \] is the total number of rows of the bicluster and \( m' \) is the total number of columns of the bicluster. Figure 1 illustrates a perfectly coherent bicluster.

\begin{align*}
\begin{bmatrix}
3 & 1 & 2 & 1 & 5 \\
4 & 1 & 5 & 1 & 5 \\
1 & 1 & 3 & 2 & 5 \\
4 & 2 & 3 & 2 & 6 \\
5 & 3 & 4 & 3 & 7 \\
\end{bmatrix} & \begin{bmatrix}
3 & 2 & 1 & 5 \\
4 & 3 & 2 & 6 \\
5 & 4 & 3 & 7 \\
\end{bmatrix}
\end{align*}

(a) (b)

Figure 1: An example of a bicluster (b) with coherent values (each row is equal to the previous one added by one) extracted from the original matrix (a). Bicluster (b) is constituted by rows \{1, 4, 5\} and columns \{1, 3, 4, 5\}.

In this work, the Mean Squared Residue (MSR), as introduced by Cheng & Church [8] and given in Eq. 2, will be considered as the coherence metric among the elements of the biclusters.

Even though the MSR is still adopted in most of the biclustering techniques described in the literature, there are some reported shortcomings associated with this metric [18, 19, 20]. One of them is related to the mean variance of the row’s residue of the bicluster, which may be maximized during the minimization of the overall mean residue, resulting in biclusters with some non-coherent rows. Other problem is related to the type of coherence among elements that this metric can find: although the MSR metric can lead to biclusters with constant values, constant rows and constant columns (beside the additive coherence), in order to find multiplicative and linear coherence a data transformation is required. Therefore, in order to address some of these problems other metrics were proposed in the literature [18, 21] or adapted from the MSR [19, 20]. Although the mathematical model described in this work was specifically developed for the MSR, other metrics may also benefit from a similar approach, thus extending this proposal. However, such extensions will be left as future perspectives.

Besides the MSR, another important aspect of biclustering is the volume of each bicluster, which is generally denoted in the literature by the number of rows times the number of columns. In order to be functional and to allow a deeper analysis of the dataset, it is usually required that a bicluster presents a large volume (large number of rows and columns). So, together with the minimization of the residue, biclustering techniques must also present mech-
anisms to stimulate the maximization of the volume of the biclusters being built.

As mentioned before, the biclustering-based data imputation technique proposed in this work, which is founded on the algebraic formalism that will be presented in the following section, does not depend on the biclustering method adopted, as long as it minimizes the MSR of each bicluster returned. Besides, it will be assumed here that the chosen biclustering method is capable of building biclusters with missing values, so that the imputation technique can be directly applied.

It is also important to highlight that the proposed technique only provides imputation to those missing values contained in biclusters. Therefore, it will only be able to infer all the missing values of a given dataset if they are all contained in previously obtained biclusters. This aspect of the proposal will be further discussed in Section 4.2.

3. Predicting Missing Values in Biclusters

After the generation of the set of biclusters, which can be accomplished with any approach based on the mean squared residue proposed by Cheng & Church [8], the prediction of the missing values inside each one of the biclusters must be made.

As a bicluster may be seen as a smaller dataset (it is a subset from the original dataset), it is possible to use traditional techniques such as learning-free models or memory-based approaches to predict the missing values. In [10], the missing values in the biclusters were simply replaced by the average of the existing elements in the bicluster. This simple approach presented good results, even better than those obtained by classical approaches, which was expected since the algorithm was dealing with a much smaller space. However, the downside of using simpler techniques to predict missing values in biclusters is that they do not take into account the criterion minimized during the construction of such biclusters, more precisely the mean squared residue.

A more reasonable approach is to convert the missing values into variables that should be obtained in order to minimize the mean squared residue, thus leading to a quadratic programming problem. As an example, consider the bicluster given in Fig. 2.

In this bicluster, there are three missing values that can be rewritten as variables (see Fig. 2(b)). Therefore, the MSR becomes a function of such
variables, as given by Eq. 4:

\[ H_{I,J}(x) = \frac{1}{nm'} \sum_{i \in I} \sum_{j \in J} r_{i,j}(x)^2, \]  

(4)

where \( x = [x_1, x_2, x_3] \)

Since this is clearly a quadratic function, it can be rewritten in the traditional formulation of a quadratic programming (QP) problem:

\[ \text{Min} \ H_{I,J}(x) = \frac{1}{2} x^T Q x + b^T x + c, \]  

(5)

with \( l_b \leq x_k \leq u_b, k = 1, \ldots, vars \),

where \( x = [x_1, x_2, \ldots, x_{\text{vars}}]^T, Q \in \mathbb{R}^{\text{vars} \times \text{vars}}, b \in \mathbb{R}^{\text{vars}}, c \in \mathbb{R} \) and \( l_b \) and \( u_b \) are the lower and upper bounds of the variables, respectively. In Fig. 2(b), \( vars = 3 \). The bounds of the variable domain, when they exist, are determined by the dataset.

The QP problem given in Eq. 5 can be solved by several specialized algorithms. However, in order to solve it, it is necessary to know matrix \( Q \), vector \( b \) and constant \( c \). \( Q \) and \( b \) can be obtained through the calculation of the gradient and Hessian of the mean squared residue (Eq. 2), while the value of \( c \) can be obtained through the calculation of \( H_{I,J}(0) \).

The gradient of the mean squared residue given in Eq. 5 can be expressed as follows:

\[ \nabla H_{I,J}(x) = Q x + b, \]  

(6)

where \( \nabla H_{I,J}(x) \in \mathbb{R}^n \).

Given that \( \nabla_k H_{I,J}(x) \) denotes the \( k \)-th element \((k = 1, \ldots, vars)\) of the gradient and considering now the mean squared residue given in Eq. 2 we have:
\[ \nabla_k H_{I,J}(x) = \frac{2}{n'm'} \cdot \left[ \sum_{i,j \in I,J} r_{i,j}(x) \cdot \frac{\partial r_{i,j}(x)}{\partial x_k} \right], \tag{7} \]

where \( r_{i,j}(x) \) is the residue of the element at row \( i \) and column \( j \) of the bicluster and \( n' \) and \( m' \) are, respectively, the number of rows and columns of the bicluster.

The Hessian of the mean squared residue is given by:

\[ \nabla^2_{k,l} H_{I,J}(x) = \frac{2}{n'm'} \cdot \frac{\partial}{\partial x_l} \left[ \sum_{i,j \in I,J} r_{i,j}(x) \cdot \frac{\partial r_{i,j}(x)}{\partial x_k} \right]. \tag{8} \]

As \( \frac{\partial^2 r_{i,j}(x)}{\partial x_l \partial x_k} = 0 \), then

\[ \nabla^2_{k,l} H_{I,J}(x) = \frac{2}{n'm'} \cdot \sum_{i,j \in I,J} \left[ \frac{\partial r_{i,j}(x)}{\partial x_l} \cdot \frac{\partial r_{i,j}(x)}{\partial x_k} \right]. \tag{9} \]

From the fact that \( H_{I,J}(x) = \frac{1}{2} x^T Q x + b^T x + c \), it is possible to obtain:

\[ Q = \nabla^2 H_{I,J}(x) \tag{10} \]

and

\[ b_k = \nabla_k H_{I,J}(0) = \frac{2}{n'm'} \cdot \left[ \sum_{i,j \in I,J} r_{i,j}(0) \cdot \frac{\partial r_{i,j}(0)}{\partial x_k} \right]. \tag{11} \]

Therefore, to calculate both \( Q \) and \( b \) we just need to calculate the partial derivatives of \( r_{i,j}(x) \) (Eq. 3) w.r.t. \( x_k \) \((k = 1, \ldots, \text{vars})\), which can assume four distinct values according to the relative position between the element subject to derivation and the variable:

\[
\frac{\partial r_{i,j}(x)}{\partial x_k} = \begin{cases} 
  c_1 = 1 - \frac{1}{n'} - \frac{1}{m'} + \frac{1}{n'm'}, & \text{if } i = k_i, j = k_j \\
  c_2 = -\frac{1}{n'} + \frac{1}{n'm'}, & \text{if } i = k_i, j \neq k_j \\
  c_3 = -\frac{1}{m'} + \frac{1}{n'm'}, & \text{if } i \neq k_i, j = k_j \\
  c_4 = \frac{1}{n'm'}, & \text{if } i \neq k_i, j \neq k_j
\end{cases}, \tag{12} \]

where \( k_i \) and \( k_j \) are respectively the row and column indices of variable \( k \) in the bicluster.
It is possible to simplify the calculation of those equations by enumerating the occurrences of each partial derivative in Eqs. 10 and 11. In the calculation of the \( k \)-th element of the gradient vector \( b \), there will be only one occurrence of \( c_1 \), when \((i, j) = (k_i, k_j)\). Values \( c_2 \) and \( c_3 \) will appear, respectively, when \( i = k_i, j \neq k_j \) and \( i \neq k_i, j = k_j \). In all the remaining situations, the derivative will be \( c_4 \). Given that, it is possible to write Eq. 11 as:

\[
\nabla_k H_{I,J}(0) = \frac{2}{n'm'} \left[ (c_1 - c_2 - c_3 + c_4) \cdot r_{k_i,k_j}(0) + (c_2 - c_4) \cdot \sum_{j \in |J|} r_{k_i,j}(0) + (c_3 - c_4) \cdot \sum_{i \in |I|} r_{i,k_j}(0) + c_4 \cdot \sum_{i,j \in I,J} r_{i,j}(0) \right].
\]

As

\[
c_1 - c_2 - c_3 + c_4 = 1, \quad c_2 - c_4 = -\frac{1}{m'}, \quad c_3 - c_4 = -\frac{1}{n'},
\]

then

\[
\nabla_k H_{I,J}(0) = \frac{2}{n'm'} \left[ r_{k_i,k_j}(0) - \frac{1}{m'} \cdot \sum_{j \in |J|} r_{k_i,j}(0) - \frac{1}{n'} \cdot \sum_{i \in |I|} r_{i,k_j}(0) + \sum_{i,j \in I,J} r_{i,j}(0) \right]
\]

= \frac{2}{n'm'} \left[ r_{k_i,k_j}(0) - r_{k_i,J}(0) - r_{I,k_j}(0) + r_{I,J}(0) \right].

Expanding each part of the equation:

\[
r_{k_i,J}(0) = \frac{1}{m'} \cdot \sum_{j \in J} (a_{k_i,j} - a_{k_i,J} - a_{I,j} + a_{I,J}) \tag{13}
\]

= \( a_{k_i,J} - a_{k_i,J} - a_{I,J} + a_{I,J} \)

= 0.
\[\begin{align*}
  r_{I,k_j}(0) &= \frac{1}{n'} \sum_{i \in I} (a_{i,k_j} - a_{i,J} - a_{I,k_j} + a_{I,J}) \\
  &= a_{I,k_j} - a_{I,J} - a_{I,k_j} + a_{I,J} \\
  &= 0,
\end{align*}\]  \hspace{1cm} (14)

\[\begin{align*}
  r_{I,J}(0) &= \frac{1}{n'm'} \sum_{i,j \in I,J} (a_{i,j} - a_{i,J} - a_{I,j} + a_{I,J}) \\
  &= a_{I,J} - a_{I,J} - a_{I,J} + a_{I,J} \\
  &= 0,
\end{align*}\]  \hspace{1cm} (15)

and, as the bicluster element \(a_{k_i,k_j} = x_k = 0\) and

\[\begin{align*}
  r_{k_i,k_j}(0) &= a_{k_i,k_j} - a_{k_i,J} - a_{I,k_j} + a_{I,J} \\
  &= a_{I,J} - a_{k_i,J} - a_{I,k_j},
\end{align*}\]  \hspace{1cm} (16)

we obtain:

\[\begin{align*}
  \nabla_k H_{I,J}(0) &= \frac{2}{n'm'} \left[ a_{I,J} - a_{k_i,J} - a_{I,k_j} \right] .
\end{align*}\]  \hspace{1cm} (17)

Equation 17 shows that it is not necessary to calculate the value of any residue to obtain vector \(b\). Instead, it is only required the replacement of the missing elements by 0 to obtain the average of each line and column of the bicluster.

Similarly for the \(Q\) matrix, it is possible to enumerate the occurrences of each partial derivative given the relative position of the two variables attributed to each element of the matrix. This enumeration leads to a more general equation, which replaces Eq. 10:

\[\begin{align*}
  Q_{k,l} &= \sum_{i,j \in I,J} \left[ \frac{\partial r_{i,j}(x)}{\partial x_k} \cdot \frac{\partial r_{i,j}(x)}{\partial x_l} \right] = \frac{\partial r_{k_i,k_j}}{\partial x_l} = \frac{\partial r_{l_i,l_j}}{\partial x_k},
\end{align*}\]  \hspace{1cm} (18)

where \(k_i, k_j, l_i\) and \(l_j\) are the row and column positions, relative to the bicluster, for the \(k\)-th and \(l\)-th variables respectively.
With \( Q \) and \( b \) calculated, in order to obtain a single global minimum it is required that matrix \( Q \) be positive definite, which can be guaranteed under the conditions defined in Theorem 1.

**Theorem 1.** Given a bicluster \( B \), with \( n' \) rows and \( m' \) columns and a maximum ratio \( \rho \) of missing values in the rows and columns of this bicluster, \( \rho < \frac{n'm'-2}{3n'm'-2n'-2m'} \) is a sufficient condition to guarantee that the Hessian matrix \( Q \) is positive definite.

**Proof.** One way to verify that a given symmetric matrix is positive definite is by means of the following inequality, derived from Geršgorin’s Circle theorem [28]:

\[
q_{i,i} > \sum_{\forall j \in |J|, j \neq i} |q_{i,j}|	ag{19}
\]

In other words, if every diagonal element of the matrix is positive and, for each row \( i \), the sum of the absolute value of every element of this row, except its diagonal, is less than the corresponding diagonal element, then a sufficient condition for the matrix to be positive definite holds [22].

In the proposed formulation, all diagonal elements of matrix \( Q \) (given in Eq. 18) are always equal to \( c_1 \) and, thus, positive and non-zero for \( n', m' > 1 \).

By limiting the number of missing values in the bicluster as \( \rho \cdot m' \) elements in each row and \( \rho \cdot n' \) in each column, with \( \rho \in \mathbb{R}, 0 < \rho < 1 \), and also considering that each missing value is a variable of the quadratic problem, the sum of the absolute values of the elements of each row of matrix \( Q \), except the diagonal element, will have a maximum value of:

\[
\max\left( \sum_{j=1, j \neq i}^{\text{vars}} |q_{i,j}| \right) = (\rho \cdot m' - 1) \cdot |c_2| + (\rho \cdot n' - 1) \cdot |c_3| +
\]

\[
+ (\rho \cdot n'm' - \rho \cdot n' - \rho \cdot m' + 1) \cdot |c_4|,
\]

where \( \text{vars} \) is the number of variables of the quadratic problem. Replacing the values of \( c_2, c_3 \) and \( c_4 \) with those from Eq. 12, this equation becomes:
\[
\max \left( \sum_{j=1, j \neq i}^{\text{vars}} |q_{i,j}| \right) = (\rho \cdot m' - 1) \cdot \left| \frac{1 - n'}{n'm'} \right| + (\rho \cdot n' - 1) \cdot \left| \frac{1 - m'}{n'm'} \right| + \left( \rho \cdot n'm' - \rho \cdot n' - \rho \cdot m' + 1 \right) \frac{1}{n'm'}. \tag{20}
\]

Given that \( n', m' > 1 \), then \( c_2, c_3 < 0 \), which makes the absolute values from the previous equation equal to:

\[
\left| \frac{1 - n'}{n'm'} \right| = \frac{n' - 1}{n'm'} \quad \text{and} \quad \left| \frac{1 - m'}{n'm'} \right| = \frac{m' - 1}{n'm'}. 
\]

Simplifying Eq. 20, we obtain:

\[
\max \left( \sum_{j=1, j \neq i}^{\text{vars}} |q_{i,j}| \right) = \frac{3\rho n'm' - (2\rho + 1)n' - (2\rho + 1)m' + 3}{n'm'}. \tag{21}
\]

Therefore, with the results from Eqs. 12 and 21 we can rewrite Eq. 19 as:

\[
\frac{(n' - 1)(m' - 1)}{n'm'} > \frac{3\rho n'm' - (2\rho + 1)n' - (2\rho + 1)m' + 3}{n'm'}, \tag{22}
\]

which results in:

\[
n'm' - n' - m' + 1 > 3\rho n'm' - (2\rho + 1)n' - (2\rho + 1)m' + 3,
\]

\[
\rho < \frac{n'm' - 2}{3n'm' - 2n' - 2m'}. \tag{23}
\]

Therefore, to guarantee that the Hessian matrix associated with the proposed quadratic programming problem is positive definite, which leads to an optimization problem with a single optimum, it is a sufficient condition that, for any bicluster, the maximum ratio of missing values in each row and column obeys the inequality in Eq. 23. Under these conditions, it is possible to obtain the global minimum of the problem in polynomial time, with advanced quadratic programming algorithms like the one presented in [23].

It is important to notice that the bicluster is just a subspace of the original dataset (which may be seen as a local data model) and, as such, it contains
just a small subset of the missing data (as the allowed ratio of missing data inside a bicluster may be controlled by the biclustering algorithm). Therefore, the QP problem to be solved has just a few variables, which makes it easily tractable by most algorithms. Additionally, there is also the computational effort to generate the biclusters, which is highly dependent on the chosen method. However, as the imputation problem is usually solved “off-line” (the response is not required in real time), the methodology proposed in this work can be suitably applied to most real-world problems.

It is also worth noting that some aspects of the bicluster may influence the quality of predictions. A bicluster with large amount of data, low mean squared residue and high occupancy is most likely to produce better predictions, as the quality of the information present in the data is higher. Besides, the larger the number of missing values covered by the set of biclusters, the larger the amount of predictions that can be made with the proposed technique.

3.1. Summary of the Methodology

From all that was discussed in this section, it is possible to summarize the proposed methodology in the following steps:

1. Select and apply a biclustering algorithm to the dataset. The selected algorithm must be capable of generating biclusters with missing values and of obeying the condition given in Eq. 23.
2. For each bicluster that contains missing values do:
   - Consider the missing values as variables of the QP optimization problem;
   - Obtain matrix $Q$ according to Eq. 18;
   - Obtain vector $b$ according to Eq. 17;
   - Minimize the quadratic expression $x^TQx + b^Tx$ to obtain the missing values $x$.
3. Whenever more than one bicluster contains the same missing element, and thus multiple imputations to the same element are calculated, it should be used some reasoning (i.e., an heuristic) to either decide which imputed value should be used as the prediction or combine the available results in a single prediction (See Section 4.4).
4. Experimental Results

In this section, it will be provided a detailed analysis of the advantages of the proposed method over other well known data imputation approaches from the literature. First, an artificial dataset with perfectly coherent elements will be adopted, and the proposed methodology will be tested against other algorithms on a series of different configurations of noise and missing data. Then, a similar experiment will be performed with two real-world datasets. Finally, further properties unique to the biclustering-based technique, which can be further explored on a post-imputation analysis of the results, will be described.

4.1. Artificial Dataset

In real-world situations, biclusters extracted from datasets are not perfectly coherent, as they are generally subject to different sources of noise. In order to evaluate how the proposed biclustering-based imputation method behaves when applied to noisy environments, in this section a series of controlled experiments was made with an artificial dataset\(^2\) composed of 100 objects (rows of the data matrix), each of which containing 100 real-valued attributes (columns).

The dataset was iteratively created, starting with a single object with 100 randomly generated attributes in the range \([-9, 110]\). The remaining 99 objects of the dataset were created by adding random values to the first object, thus leading to a perfect additive-coherent bicluster.

4.1.1. Methodology

The experiments in this section were performed considering the whole dataset as a single bicluster, since it was originally conceived to be perfectly coherent (in other words, the proposed technique was applied directly, as a global model for data imputation). Gaussian noise with 0 mean and standard-deviation varying from 0 to 50 was introduced in the dataset, and these modified versions were labeled as described in Tab. 1. In this table, it is also given the MSR calculated for each dataset with Gaussian noise, in order to indicate how difficult it should be for the biclustering-based imputation algorithm to deal with each of them.

\(^2\)All variations of the artificial dataset adopted in this section are available at ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/MissingData/.
Table 1: Standard-deviation (Std. Dev.) of the Gaussian distribution and MSR calculated for each dataset.

<table>
<thead>
<tr>
<th>Label</th>
<th>Std. Dev.</th>
<th>MSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO NOISE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMALL NOISE</td>
<td>5</td>
<td>0.97</td>
</tr>
<tr>
<td>MEDIUM NOISE</td>
<td>15</td>
<td>96</td>
</tr>
<tr>
<td>LARGE NOISE</td>
<td>30</td>
<td>380</td>
</tr>
<tr>
<td>HUGE NOISE</td>
<td>50</td>
<td>2450</td>
</tr>
</tbody>
</table>

To evaluate the impact of the number of missing values on the proposed imputation technique, experiments were made with 5%, 15%, 50% and 70%\(^3\) of missing elements in each dataset, and the overall data imputation performance was evaluated with respect to the Mean Absolute Error (MAE).

Two algorithms widely adopted in the literature [2, 3] were also used here (and in the next subsection) as the basis for comparisons. The first one is a variation of the k-Nearest Neighborhood (k-NN) algorithm described in [2], which was adopted with parameter $k$ set to $k = 15$. The second technique is the Regulated Singular Value Decomposition (rSVD), proposed by Funk [3] for the Netflix collaborative filtering challenge [24]. Since all three methods are deterministic, the results reported here were obtained in a single run.

This second approach consists in the estimation of the SVD of the incomplete dataset, using the known values, and in the minimization of the squared residue by means of a conjugate gradient method. In order to avoid overfitting, Funk [3] added a regularization component to the squared residue. For this experimental set, the parameters of the rSVD algorithm were empirically set to: learning rate equal to $10^{-5}$, regularization ratio equal to 0.015, 200 iterations of the conjugate gradient method for each learning feature and 5 features for each decomposed matrix.

\(^3\)Notice that the condition imposed by Theorem 1 is just a sufficient condition so it is still possible to obtain a positive-definite matrix even if this condition is not met. Since this dataset was built from a single coherent bicluster, the positive definiteness was maintained even for higher rates of missing data.
Table 2: MAE obtained by the three algorithms in the experiments performed with different amount of noise.

<table>
<thead>
<tr>
<th></th>
<th>NO NOISE</th>
<th>SMALL NOISE</th>
<th>MEDIUM NOISE</th>
<th>LARGE NOISE</th>
<th>HUGE NOISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Missing</td>
<td>Bicluster</td>
<td>$k$-NN</td>
<td>rSVD</td>
<td>Bicluster</td>
<td>$k$-NN</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.04</td>
<td>2.16</td>
<td>0.75</td>
<td>0.78</td>
</tr>
<tr>
<td>15</td>
<td>0.00</td>
<td>1.66</td>
<td>2.13</td>
<td>0.80</td>
<td>0.83</td>
</tr>
<tr>
<td>50</td>
<td>0.00</td>
<td>10.90</td>
<td>2.15</td>
<td>0.81</td>
<td>2.50</td>
</tr>
<tr>
<td>70</td>
<td>0.00</td>
<td>2.06</td>
<td>2.18</td>
<td>0.82</td>
<td>25.52</td>
</tr>
<tr>
<td></td>
<td>Bicluster</td>
<td>$k$-NN</td>
<td>rSVD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>41.47</td>
<td>42.04</td>
<td>41.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>40.93</td>
<td>42.22</td>
<td>41.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>40.60</td>
<td>41.54</td>
<td>40.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>42.08</td>
<td>42.20</td>
<td>41.38</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.1.2. Results

Regarding the first artificial dataset, without any noise (NO NOISE), it is possible to see in Tab. 2 that the biclustering-based methodology was, as expected, able to find the exact values of each missing element, while the other two approaches presented non-zero MAEs.

In the experiment with a small amount of noise (SMALL NOISE – Tab. 2), the MAEs obtained by the biclustering-based approach for 5% and 15% of missing values were closer but still smaller than those of the $k$-NN technique. With the increase of the number of missing values, the MAEs of the proposed algorithm remained practically stable, while those obtained by the $k$-NN algorithm increased significantly. In this experiment, the MAEs obtained by the rSVD technique did not present much variation either, but they were significantly higher than those achieved by the biclustering-based method.

In the same table, for the dataset with medium amount of noise, the difference between the MAEs obtained by each algorithm was smaller than that
observed in the previous experiments, but the biclustering-based approach was still able to obtain the best results in all the considered scenarios. It was also noted that, again, the \( k \)-NN approach presented some instability when dealing with higher amounts of missing data in the dataset, while the other two algorithms remained stable.

Behaviors similar to that verified for the MEDIUM NOISE dataset were also observed in the LARGE NOISE and HUGE NOISE experiments, but the differences between the MAEs obtained by each algorithm were even smaller in these two situations. The biclustering-based imputation method still presented a slight advantage over the other approaches, except in the HUGE NOISE experiment with 70% of missing values, in which the rSVD technique was able to achieve results slightly better than those of the proposed technique.

In summary, these experiments indicate that the biclustering-based methodology is much better than the \( k \)-NN and rSVD algorithms when the noise level is low. These results were expected since less noise leads to a higher degree of coherence between the elements of the bicluster (which is the basis of the proposed methodology), and illustrate the importance of coherence to the quality of the values imputed to the missing data (evaluated here by the MAE).

It was also observed in the experiments presented in this section that the proposed technique can properly handle noisy data up to a certain limit and, most importantly, that it tends to present a stable behavior when the number of missing values in the dataset is increased. Such a stable behavior was also obtained with rSVD.

Finally, these experiments show that the methodology presented here is sensitive to the degree of additive coherence found in the dataset. The more coherent to an additive model the data is, the better the results. But even if the coherence is compromised, it is still competitive with other approaches on the experiments.

4.2. Real-world Datasets

In the previous section, it was possible to observe that the biclustering-based approach was capable of dealing with noisy datasets and also with datasets with large amounts of missing values. In this section, several experiments were performed with two real-world datasets, so that the overall performance of the proposed technique can be properly evaluated in more
practical situations. When dealing with real-world data, a proper biclustering process must be performed before the data imputation step, as it is not possible to assume that the whole dataset corresponds to a single bicluster.

4.2.1. The biclustering algorithm

As previously stated, in order to use the proposed quadratic programming formulation to impute missing values in biclusters, it is necessary a set of additive coherent biclusters that cover, if possible, all the missing values of an incomplete dataset. In this paper, a modified version of the meta-heuristic named SwarmBCluster [12] was chosen to perform this task, since the approach in [12] was able to find higher quality biclusters when compared to other approaches from the literature.

Basically, SwarmBCluster is constituted by three main heuristics that interact with each other (further details of the algorithm can be found in [12]):

- A constructive heuristic, that starts with an initial bicluster with a single row and $m$ columns of an $n \times m$ dataset, and iteratively inserts new rows to this bicluster and also extract columns to control the residue;

- A dynamically built candidate list that supplies, to the first heuristic, the initial biclusters that are more likely to lead to final biclusters with maximum coverage;

- An Ant Colony Optimization meta-heuristic (ACO) [25], responsible for indicating to the first heuristic the best order in which the rows should be inserted to maximize the bicluster’s volume.

Although SwarmBCluster was reported to perform well on gene expression datasets [12], it cannot properly handle incomplete datasets without some modifications. The original SwarmBCluster algorithm does not guarantee neither a full coverage\(^4\) of the dataset nor that specific elements (positions) of the data matrix will be covered by the obtained set of biclusters. Additionally, it may have problems in calculating the MSR for incomplete biclusters (those with missing values). As the biclustering-based data imputation algorithm proposed here requires that most of the missing values are

\(^4\)In the context of this paper, coverage corresponds to the percentage of missing values of the dataset that are included in at least one bicluster.
covered by the set of biclusters and also that the biclustering method is able to perform MSR calculations with the absence of values in the biclusters, the SwarmBCluster was then adapted to attend these requirements, so that it could be properly adopted here.

First of all, in order to allow the calculation of the MSR in incomplete datasets, the original missing values were pre-imputed with a simple and computationally inexpensive technique, more specifically the Nearest-Neighbor heuristic [26]. This approach certainly introduces noise in the data due to imputation errors but, as the MSR is a robust metric with respect to noise, this leads to better estimates than just ignoring the missing data or filling the gaps with random values.

To guarantee that every missing value is included in at least one bicluster from the set, for each initial bicluster the starting points were defined by rows containing missing values that did not appear in any bicluster yet. Besides, a tabu list was also created so that at least one missing value not yet covered will not be removed from the bicluster during the execution of the constructive heuristic.

Also, if a given bicluster leads to a non-convex QP formulation (which may happen when the condition defined by Theorem 1 is not met), the imputation procedure of the missing data of such bicluster is postponed. After all the missing values that are contained in biclusters that lead to convex QP problems are imputed into the dataset, those “non-convex biclusters” are reconsidered. If they still lead to non-convex formulations, the Nearest-Neighbor algorithm is used to impute those missing values.

As the last modification of the original SwarmBCluster, instead of using ACO to find the best order of rows to be inserted in the constructive heuristic, such order was probabilistically determined. In this new approach, the probability that a given row is selected to be inserted into the bicluster is inversely proportional to the Euclidean distance between the candidate row and all the other rows previously inserted into the bicluster (already calculated in the Nearest-Neighbor step) which reduces the computational complexity of the algorithm. With this last modification the SwarmBCluster loses its swarm-inspiration and becomes a constructive-heuristic with local search. Therefore, in the remaining sections of the paper, this modified biclustering algorithm combined with the data imputation technique proposed in this work will be referenced as QP_Biclustering.
4.2.2. Methodology

To evaluate the performance of the proposed data imputation technique, several experimental tests were performed with two different datasets. The first dataset, known as Yeast [8], is a gene expression dataset with 2,884 rows and 17 columns, frequently used to evaluate the performance of biclustering techniques. The second dataset, known as Jester [27], has 24,983 rows and 101 columns, and is frequently used to evaluate the performance of collaborative filtering algorithms, which can be seen as a type of missing data problem.

To properly determine whether the imputations made by the proposed algorithm actually correspond to the real values that were missing, both Yeast and Jester were considered as complete datasets, and the missing data to be recovered by the algorithms were randomly defined, with the same ratios of missing data used on the previous experiments for each problem: 5%, 15%, 50% and 70%. As Jester is an incomplete dataset, those rows and columns with the original missing values were removed, which resulted in a complete dataset with 7,199 rows and 58 columns.

Since this is not a controlled experiment and the coherence relations are different for each subset of objects and attributes, the variance of the MAE values can be high, considering the whole set of imputed data. This may lead to an incorrect conclusion if, for example, one algorithm has a slightly lower imputation error on most missing values, but a much higher imputation error on just a few missing values.

So, in order to better compare the results obtained by QP-Biclustering, the average relative difference of the absolute errors (Eq. 24) obtained by the proposed technique, k-NN and rSVD will be considered here. According to this metric, a positive result indicates that, on average, the QP-Biclustering algorithm was better than the other technique (presented smaller MAE), while negative results indicate the opposite. Besides, the percentage of situations in which QP-Biclustering presented errors smaller than the other two algorithms ($P_{small}$) will also be provided. For the sake of completeness, the values of the MAEs obtained by each algorithm will also be provided in this section (Tab. 3), but the discussion that follows will be based mainly on the relative differences and $P_{small}$.

\[
rel\_diff = \frac{MAE_{AlgBIC} - MAE_{Alg1}}{\max(MAE_{Alg1}, MAE_{AlgBIC})},
\]

where $MAE_{AlgBIC}$ is the MAE obtained by QP-Biclustering and $MAE_{Alg1}$
Table 3: Mean Absolute Errors for data imputation in the *Yeast* and *Jester* datasets.

<table>
<thead>
<tr>
<th>% Missing</th>
<th>Yeast</th>
<th>Jester</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QP_Bic.</td>
<td>k-NN</td>
</tr>
<tr>
<td>5</td>
<td>17.73</td>
<td>27.71</td>
</tr>
<tr>
<td>15</td>
<td>17.32</td>
<td>27.92</td>
</tr>
<tr>
<td>50</td>
<td>20.03</td>
<td>28.93</td>
</tr>
<tr>
<td>70</td>
<td>31.26</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 4: Average relative difference of absolute errors for data imputation in the *Yeast* dataset.

<table>
<thead>
<tr>
<th>% Missing</th>
<th>QP_Bic. vs. k-NN</th>
<th>QP_Bic. vs. rSVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.35</td>
<td>0.15</td>
</tr>
<tr>
<td>15</td>
<td>0.37</td>
<td>0.22</td>
</tr>
<tr>
<td>50</td>
<td>0.31</td>
<td>0.24</td>
</tr>
<tr>
<td>70</td>
<td>—</td>
<td>−0.02</td>
</tr>
</tbody>
</table>

is the MAE of the algorithm with which QP_Biclustering is being compared.

The parameters of the QP_Biclustering algorithm were empirically adjusted for both datasets, which led to: $\delta = 300$ for *Yeast* and $\delta = 6$ for *Jester*. The size of the biclusters was also limited to a minimum of 7 columns for *Yeast* and 12 columns for *Jester*.

### 4.2.3. Results

Table 4 depicts the average difference of absolute errors for the *Yeast* dataset. As it is possible to see in this table, the biclustering-based approach presented better results for up to 50% of missing values in the dataset, with imputation up to 37% better than $k$-NN and up to 24% better than rSVD.

With 70% of missing data, the QP_Biclustering obtained slightly worse results when compared to rSVD, about 2% worse on average. For this amount of missing values in the dataset, the $k$-NN approach was not able to impute all the missing values, so its results were omitted.

In Tab. 5, the percentage of missing values predicted by QP_Biclustering with errors smaller than those obtained by the other two approaches ($P_{small}$) are presented. By inspecting this table, it is possible to see that the biclustering-based technique is capable of maintaining a $60 - 70\%$ of winning
ratio over the other two algorithms for up to 50% of missing data. Even in the situation with 70% of missing data, the proposed technique was able to present better results in more than 50% of the cases when compared to rSVD, although the average relative error difference shown in Tab. 4 was slightly favorable to the rSVD in this specific case.

The time required by each algorithm to perform the imputations of all the missing values of the Yeast dataset is depicted in Tab. 6. This table shows that QP_Biclustering has a higher computational cost when compared to the other two approaches, which can be explained by the fact that both the rSVD and k-NN algorithms generate a single global model, while the biclustering-based technique creates several local models, each one for a different region of the dataset.

Analyzing such results more deeply (Tab. 7), it is interesting to highlight that QP_Biclustering uses about 25% of the total time reported in Tab. 6 to find biclusters that covers up to 90% of the missing data, while the remaining 10% generally requires the creation of specific biclusters to cover each one of

---

5The experiments were performed on an Intel Core2Quad Q9550 @ 2.83GHz with 2GB of RAM.
Table 7: Total number of biclusters and computational time required by QP_Biclustering to impute a percentage of the missing values in the Yeast dataset.

<table>
<thead>
<tr>
<th>% Missing</th>
<th># of bics.</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>584</td>
<td>38 s.</td>
<td>1.1 m.</td>
<td>2.4 m.</td>
</tr>
<tr>
<td>15</td>
<td>1,131</td>
<td>1.2 m.</td>
<td>2.2 m.</td>
<td>4.5 m.</td>
</tr>
<tr>
<td>50</td>
<td>1,907</td>
<td>9.0 m.</td>
<td>15.3 m.</td>
<td>34.8 m.</td>
</tr>
<tr>
<td>70</td>
<td>2,084</td>
<td>10.4 m.</td>
<td>21.4 m.</td>
<td>43.5 m.</td>
</tr>
</tbody>
</table>

Table 8: Average relative difference of absolute errors for data imputation in the Jester dataset.

<table>
<thead>
<tr>
<th>% Missing</th>
<th>QP_Bic. vs. k-NN</th>
<th>QP_Bic. vs. rSVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.06</td>
<td>0.10</td>
</tr>
<tr>
<td>15</td>
<td>0.06</td>
<td>0.09</td>
</tr>
<tr>
<td>50</td>
<td>0.09</td>
<td>0.10</td>
</tr>
<tr>
<td>70</td>
<td>0.09</td>
<td>0.10</td>
</tr>
</tbody>
</table>

them, with a negative impact in the computational cost. As the main goal of the experiments performed here was to compare the imputation errors obtained by the proposed technique to those obtained by the k-NN and rSVD, the biclustering algorithm adopted here was modified to always obtain a set of biclusters that covers all the missing values of the dataset (see Section 4.2.1), even though such biclusters may contain elements that are not well correlated to each other, which explains the high computational costs observed here.

During the experiments, it was observed that biclusters with a small number of missing values tend to appear during the final steps of the algorithm. Therefore, if it is not required that all the missing values of the dataset are obtained by the biclustering technique, such situation could be properly explored, and the overall computational time required can be significantly reduced.

Table 8 shows the average relative difference of absolute errors obtained for the Jester dataset. This table shows that the QP_Biclustering was able to outperform the other two approaches in all the situations considered, with average improvements ranging from 6% to 10% of the absolute error.

Finally, the $P_{small}$ of the QP_Biclustering algorithm over the other two
Table 9: $P_{small}$ of QP_Biclustering for data imputation in the *Jester* dataset.

<table>
<thead>
<tr>
<th>% Missing</th>
<th>QP_Bic. vs. $k$-NN</th>
<th>QP_Bic. vs. rSVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>54.56%</td>
<td>57.86%</td>
</tr>
<tr>
<td>15</td>
<td>54.09%</td>
<td>57.78%</td>
</tr>
<tr>
<td>50</td>
<td>55.18%</td>
<td>57.91%</td>
</tr>
<tr>
<td>70</td>
<td>54.87%</td>
<td>57.83%</td>
</tr>
</tbody>
</table>

Table 10: Computational time required by each algorithm to impute every missing value in the *Jester* dataset.

<table>
<thead>
<tr>
<th>% Missing</th>
<th>QP_Bic.</th>
<th>k-NN</th>
<th>rSVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3 h.</td>
<td>2.4 m.</td>
<td>1.3 m.</td>
</tr>
<tr>
<td>15</td>
<td>43 h.</td>
<td>4.47 m.</td>
<td>1.08 m.</td>
</tr>
<tr>
<td>50</td>
<td>42 h.</td>
<td>11.00 m.</td>
<td>1.00 m.</td>
</tr>
<tr>
<td>70</td>
<td>61 h.</td>
<td>14.00 m.</td>
<td>50 s.</td>
</tr>
</tbody>
</table>

Table 11: Total number of biclusters and computational time required by QP_Biclustering to impute a percentage of the missing values in the *Jester* dataset.

<table>
<thead>
<tr>
<th>% Missing</th>
<th># of bics.</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5,230</td>
<td>49.8 m.</td>
<td>1.6 h.</td>
<td>3.0 h.</td>
</tr>
<tr>
<td>15</td>
<td>9,152</td>
<td>6.2 h.</td>
<td>12.4 h.</td>
<td>24.6 h.</td>
</tr>
<tr>
<td>50</td>
<td>16,655</td>
<td>5.6 h.</td>
<td>9.9 h.</td>
<td>17.1 h.</td>
</tr>
<tr>
<td>70</td>
<td>28,719</td>
<td>7.2 h.</td>
<td>12.8 h.</td>
<td>23.8 h.</td>
</tr>
</tbody>
</table>

approaches for the *Jester* problem are depicted in Tab. 9. For this problem, the proposed algorithm was still able to overcome the other contenders in most of the imputations, although with lower ratios when compared to the previous dataset. Considering $k$-NN, the biclustering-based technique presented, on average, smaller errors in more than 54% of the imputations, while, with respect to the rSVD, QP_Biclustering achieved a $P_{small}$ of 57%. Concerning the computational time required to impute all the missing values of the *Jester* dataset, Tabs. 10 and 11 show that QP_Biclustering was highly affected by the larger amount of data presented by this dataset, for the same reasons explained before for the *Yeast* dataset.
Finally, it is also worth noticing that, even though the biclustering heuristic applied here cannot guarantee 100% coverage of the missing data, in every experiment performed here it did achieve total coverage.

4.3. Comparison with Other Biclustering Approaches

Other known biclustering-based approaches from the literature that perform some sort of data imputation are the already mentioned Nearest-neighbor Biclustering [9] and BIC-aiNet [10, 11] algorithms. These techniques were initially proposed as recommender system, which means that their goal was to estimate whether an item $f$ (a product in an store, for example) should be recommended to a user $o$. The NN-Biclustering [9] does that by first converting the original dataset into a binary matrix, in which 1 indicates that a given user liked an item (and 0 otherwise), and then by extracting biclusters from the dataset so that other possible 1’s can be found.

The second approach, BIC-aiNet [10, 11], searches for additive coherent biclusters in the original dataset and then estimates the missing values in the obtained biclusters based on the average of the values associated with a given item in each bicluster. So, differently from NN-Biclustering, BIC-aiNet can be seen as an actual data imputation algorithm, since it does not perform any transformation in the original data.

Therefore, in order to compare the performance of this work’s proposal against other biclustering-based data imputation techniques, the same experiments discussed in Section 4.2 were repeated with BIC-aiNet, and the MAEs obtained by such algorithm were directly compared with those obtained by QP_Biclustering. The parameters adopted for BIC-aiNet, after an extensive trial-and-error tuning procedure, were: $\delta = 250$ and 3, respectively for Yeast and Jester datasets, 300 iterations, $\text{rows\_weight} = 3$, $\text{cols\_weight} = 2$ and 300 biclusters.

The obtained results are reported in Table 12. The BIC-aiNet algorithm led to higher values of MAE for both datasets, probably due to the simplicity of its imputation mechanisms.

Therefore, just considering several local models of the data to perform imputation (which is accomplished with the biclustering step) is not enough to obtain better results. It is also required an advanced data imputation mechanism that is capable of extracting as much information as possible from these local models, such as the one proposed in this work.
4.4. Imputation Confidence and Interpretability

Although the local models created by the proposed biclustering-based technique are responsible for raising the computational cost of the imputation procedure, consistently lower imputation errors were obtained. Additionally, these models also bring some unique properties that can be explored in different contexts, depending on the application. One of them is the ability to estimate the quality of imputation.

Each bicluster in the set found by the algorithm has its own values of MSR and volume, which may be used to indicate the quality of the bicluster. The lower the MSR of the bicluster the lower the noise within the data it represents (with respect to the coherence model depicted in Section 2), while the larger the volume the more information is contained within the bicluster. These two quality indicators can be used to evaluate how likely the imputations made from one bicluster are to present lower errors when compared to those made from other biclusters, which can be useful in several applications such as collaborative filtering.

To illustrate that the bicluster technique is endowed with this feature, in this section an analysis of the relation between the MSR and volume of a bicluster and the MAE of its predictions will be made. To accomplish this task, the results obtained in the previous section for the Yeast and Jester datasets with 50% of missing values were considered here. In order to establish the relation between the MSR and the volume of a given bicluster with the MAE of its imputations, first it will be verified whether the MSR and the volume are, for the set of biclusters obtained, independent variables or not.

Figures 3(a) and 3(b) depict the relation between the MSR and the volume of each bicluster generated before the imputation process presented in
Section 4.2. In these figures, each point represents one bicluster returned by the algorithm. It is possible to perceive that, for the Yeast dataset, the combinations of MSR and volume seems to be uniformly distributed, which indicates that it is possible to verify the influence of both parameters on the MAE of the imputations independently. However, for the Jester dataset, Fig. 3(b) indicates a noticeable dependency between both parameters, as higher MSRs are associated with larger volumes. Therefore, in this situation, the quality of the results will depend on the parameter (MSR or volume) with the highest influence on the MAE.

Figure 4(a) shows the relation between the MAE of the imputations made for each bicluster obtained by the algorithm and its associated MSR, while Fig. 4(b) presents the relation between the MAE and the volume of each bicluster, both for the Yeast dataset. From these plots, it is possible to see how the MSR and volume of the biclusters are reflected in the imputation quality. For this problem, it is possible to see that the higher the MSR, the higher the imputation error, whereas the larger the volume, the lower the imputation error. This was the expected behavior as the MSR is directly related to the amount of noise inside a bicluster, and the volume is related to the amount of information available to perform the imputation. Therefore, higher noise is supposed to lead to worse imputations while higher volumes tend to produce better imputations. This was actually observed for this problem.

Figures 4(c) and 4(d) respectively present the same information of Fig-
Figure 4: Relation between (a) MSR and MAE and (b) Volume and MAE for the Yeast dataset, and (c) MSR and MAE and (d) Volume and MAE for the Jester dataset. All the results were obtained with 50% of missing data in the dataset.

ures 4(a) and 4(b), but now for the Jester dataset. From Fig. 4(c), it is possible to observe the same relation between MSR and MAE previously observed for the Yeast dataset but, in Fig. 4(d), the relation between volume and MSR was inverted. For this problem, the larger the volume of the biclusters, the higher their imputation errors. This distinct result occurs, as previously stated, due to the dependency between the MSR and volume observed in Fig. 3(b) for this particular dataset.

In practice, this information can be used to measure the relative degree of confidence of the imputations made by two different biclusters: if one of them has a significantly smaller MSR than the other, it is also expected that it may lead to smaller prediction errors. However, if two biclusters have similar MSRs, their volume can be used to determine which one is more likely to lead to smaller imputation errors.

Other interesting property associated with the proposed biclustering-
based technique is the possibility to interpret the local models directly. The 
$k$-NN, for example, groups the $k$ most similar objects together regarding ev-
every attribute, thus giving no direct information on which set of attributes 
was effectively responsible for defining such group of objects.

Since the models generated by each bicluster explicitly describes which 
objects and features define this local model, it is easier to extract information 
from them. In missing data analysis, this can be useful to identify the reasons 
why a given value is missing, by the presence of related attributes that may 
be used to support the identified explanation. This could be useful to many 
knowledge discovery tasks such as: finding out the mechanism of missing 
data, extracting association rules and finding out meta-attributes to evaluate 
the similarity between objects.

5. Conclusion

In this work, it was introduced and properly formalized the theoretical 
aspects of the adoption of biclusters generated based on the mean squared 
residue (MSR) to predict and impute missing values of a dataset. This 
approach exploits the assumption that each generated bicluster (local model 
that explains a subset of the dataset) should present small MSR to infer 
that the missing values should be those values that minimize the MSR. In 
the proposed technique, the MSR was considered the objective function to be 
minimized in a quadratic programming (QP) problem, and the missing values 
the variables of such optimization problem. By doing that, the prediction 
of missing data becomes dependent upon just a small and coherent part of 
the dataset, which tends to present much more information about the local 
missing values than the whole dataset.

The necessary steps to calculate the Hessian matrix and gradient vector 
of the QP problem for a given bicluster with missing data was thoroughly 
described here and, additionally, it was proved that, given certain sufficient 
conditions, the Hessian matrix is positive definite, which allows the adoption 
of exact optimization methods to solve the quadratic programming problem.

Experimental results indicate that data imputation based on local models 
of real-world datasets leads to higher accuracy, as the overall quality of im-
putation achieved by the proposed biclustering-based approach was superior 
to the one presented by two algorithms based on global models (a modified 
version of the $k$-NN algorithm and the rSVD technique). However, the iden-
tification of the local models introduces an extra computational burden that
depends mostly on the biclustering algorithm chosen. This was particularly relevant in the experiments performed here, in which it was imposed the restriction that all missing values of the dataset must be covered by at least one bicluster, so that the results could be compared to that of global model techniques.

The proposed biclustering-based data imputation technique also presents two interesting additional features: (i) it allows the estimation of the quality of the imputations made according to the internal quality of each bicluster (the MSR), so those missing values that can be predicted with more confidence can be directly identified; and (ii) as the data imputation is based on local models of the dataset, the obtained results can be more easily interpreted, which can be useful to many applications that require deeper data analysis.

Therefore, given these aforementioned characteristics of the proposed methodology, it is possible to say that such approach is more suitable to situations in which the imputation of 100% of the missing values is not a strong requirement, so that only the missing values naturally contained in the obtained biclusters can be imputed. In such cases, all the advantages of the biclustering-based technique can be explored, without significantly increasing the computational time required by the whole data imputation process. If the imputation of all the missing values is required, the proposed technique can be combined with a traditional data imputation method, to predict the missing values not covered by the set of biclusters.

As future steps, we intend to develop a biclustering algorithm that leads to the highest possible performance of the imputation technique proposed here, with respect to prediction quality and computational cost. We aim at finding an algorithm that is capable of handling missing data without requiring any kind of pre-imputation, and also that maximizes the coverage of the whole dataset. Finally, we also intend to investigate the possibility of developing similar formulations for data imputation in biclusters obtained according to different metrics.

References


[28] R.S. Varga, Geršgorin and his circles, Springer Verlag, 2004