



BIROn - Birkbeck Institutional Research Online

Wasito, I. and Mirkin, Boris (2006) Nearest neighbours in least-squares data imputation algorithms with different missing patterns. *Computational Statistics & Data Analysis* 50 (4), pp. 926-949. ISSN 0167-9473.

Downloaded from: <https://eprints.bbk.ac.uk/id/eprint/313/>

Usage Guidelines:

Please refer to usage guidelines at <https://eprints.bbk.ac.uk/policies.html>
contact lib-eprints@bbk.ac.uk.

or alternatively

Birkbeck ePrints: an open access repository of the research output of Birkbeck College

<http://eprints.bbk.ac.uk>

Wasito, I. and Mirkin, B. (2006). Nearest neighbours in least-squares data imputation algorithms with different missing patterns. *Computational Statistics & Data Analysis* 50 (4) 926-949.

This is an author-produced version of a paper published in *Computational Statistics & Data Analysis* (ISSN 0167-9473). This version has been peer-reviewed but does not include the final publisher proof corrections, published layout or pagination.

All articles available through Birkbeck ePrints are protected by intellectual property law, including copyright law. Any use made of the contents should comply with the relevant law.

Citation for this version:

Wasito, I. and Mirkin, B. (2006). Nearest neighbours in least-squares data imputation algorithms with different missing patterns. *London: Birkbeck ePrints*. Available at: <http://eprints.bbk.ac.uk/archive/00000313>

Citation for the publisher's version:

Wasito, I. and Mirkin, B. (2006). Nearest neighbours in least-squares data imputation algorithms with different missing patterns. *Computational Statistics & Data Analysis* 50 (4) 926-949.

<http://eprints.bbk.ac.uk>

Contact Birkbeck ePrints at lib-eprints@bbk.ac.uk

Nearest Neighbours in Least-Squares Data Imputation Algorithms with Different Missing Patterns

Ito Wasito^a and Boris Mirkin^{b,*}

^a*Department of Electrical and Computer Engineering, Faculty of Engineering, IIUM
Jl. Gombak, 53100 Kuala-Lumpur, Malaysia*

^b*School of Computer Science and Information Systems, Birkbeck College, University of
London, Malet Street, London, WC1E 7HX, UK*

Abstract

Methods for imputation of missing data in the so-called least-squares approximation approach, a non parametric computationally efficient multidimensional technique, are experimentally compared. Contributions are made to each of the three components of the experiment setting: (a) algorithms to be compared, (b) data generation, and (c) patterns of missing data. Specifically, “global” methods for least-squares data imputation are reviewed and extensions to them are proposed based on the nearest neighbours (NN) approach. A conventional generator of mixtures of Gaussian distributions is theoretically analysed and, then, modified to scale clusters differently. Patterns of missing data are defined in terms of rows and columns according to three different mechanisms that are referred to as Random missings, Restricted random missings, and Merged database. It appears that NN based versions almost always outperform their global counterparts. With the Random missings pattern, the winner is always the authors’ two-stage method INI, which combines global and local imputation algorithms.

Key words: Least squares, Nearest neighbours, Singular Value Decomposition, Missing data, Random missing, Restricted random missing, Merged database missing

1 Introduction

The problem of imputation of missing data emerges in such areas as editing of survey data[6], maintaining of medical documentation [21] and modelling DNA microarray

* Tel: +44-207-631-6746; Fax: +44-207-631-6727

Email address: mirkin@dcs.bbk.ac.uk (Boris Mirkin).

data [36]. In the last decades a number of approaches have been proposed for solving it. The most popular approach of the past, imputing the feature's mean, has been supplemented by the nearest neighbour mean [14,36] as well as other conditional means [24,25,28]. Also, two more global and computationally intensive approaches to imputation of missing entries based on the least-squares and maximum-likelihood principles have been developed [10,23,26,22,7,25,31,18].

In [37] the authors reviewed least-squares based algorithms and proposed a number of their modifications involving the nearest neighbourhood methods, then carried out a series of computational experiments involving uniformly random missing entries across the data generated with either of two mechanisms: (a) a unidimensional generator with an additive uniformly distributed noise, (b) a Gaussian mixture as presented in the popular NetLab package [11]. In our experiments, a complete data matrix and a set of entries that are considered missing in it are generated separately. This design enable us, for any data set and pattern of missing data, to compare the imputed values with those originally generated: the smaller the difference, the better the method. According to experiments reported in [37], the two different data models lead to different results. With the unidimensional data generator, the best imputation methods are those using just one factor. Nearest neighbour based modifications do not improve results in this case, even at high levels of noise. In contrast, with data generated according to the Gaussian mixture distribution, methods involving the nearest neighbours are the best.

In this paper, the study reported in [37] is extended in both aspects, the data generation and modelling patterns of missing data.

In the aspect of data generation, we concentrate on mixtures of Gaussian distributions. In particular, we mathematically investigate the conventional utility provided by NetLab [11] to see that it produces rather a blot of inseparable clusters; we modify it to get a controllable structure of Gaussian clusters with regard to both their location and orientation.

With respect to modelling patterns of missing data, we are going to take a line differing from the basic types of missing data introduced in [25]. The types defined in [25] are associated with the extent of dependence of missings on feature values, from the no dependence case referred to as MCAR (Missing Completely At Random) to the case of dependence referred to as NI (Non-Ignorable). The latter commonly occurs when respondents do not want to reveal something personal or unpopular about themselves [25].

Since we are not going to restrict ourselves with surveys only, we prefer defining patterns of missing data in terms of rows and columns rather than in terms of data entries as in [25]. In particular, we will be interested in the following three mechanisms for emergence of patterns of missing data:

- (1) **Random missing (RM)**. RM means that the missing data mechanism is the same over all columns and rows, which may happen when a missing entry is due to an

occasional lack of accuracy or measurement facilities. To define such a mechanism, one needs to fix the probability of missing in any entry and then generate a pattern of independent missings according to this probability.

- (2) **Restricted random missing (RRM)**. RRM means that there is a submatrix in the data specified by a subset of features and a subset of cases such that missings may occur only within the specified subsets. This type of missing patterns models a situation occurring in surveys or polls at which there is a set of questions related to an issue which is sensitive for a group of respondents. These respondents tend to leave the sensitive questions with no answer, this way generating incomplete data in a survey. A similar pattern may occur in a marketing research or medical documentation at which a block of questions is considered less important than the rest.
- (3) **Merged database (MD)**. This type of patterns of missing data is defined to model a situation at which the data set under consideration has been obtained by merging two or more databases of the same type of records which is frequent in medical informatics. It may happen that records in either of the original databases lack some features that have been recorded in the other database. This way, a submatrix of entries corresponding to the records in the incomplete database and the features that have been missed in it will be missed in the data entirely. MD can be considered an ultimate RRM pattern at which the probability of missing is unity.

With regard to imputation algorithms, we focus on the set of eight algorithms specified in [37] as a representative sample from the realm of least-squares iterative approximation approaches and add the mean imputation algorithm in two versions, as is and a nearest neighbour modified version, as a bottom-line. The variety of the least-squares approximation algorithms in our study comes from two major approaches: one approach, IMLS, is based on iterative reimputation of missing entries according to a few singular vectors based on the redefined completed data set; the other, ILS, iteratively extracts factors approximating only those data entries that have been actually observed.

The paper is organized as follows. Section 2 gives a brief description of the global least-squares imputation methods. Our NN versions of the imputation methods will be described in section 3. Section 4 provides for the setting of experiments and section 5 reports of our findings. Section 6 concludes the paper.

2 Least-Squares (LS) Imputation Techniques

In the published literature, one can distinguish between the following three approaches to imputation of missing data:

- (1) Conditional mean, at which each missing entry is substituted by a predicted value such as the variable mean or the value predicted by a regression function or decision

tree [20,24,25,28]. A common feature of the conditional mean approaches is that missing entries are dealt with sequentially, one-by-one, and most of the methods rely on a limited number of variables.

- (2) Maximum likelihood based methods, at which missings are imputed according to an estimated data density function [7,8,25,30–32]. Methods within this approach rely on a well established framework of the probabilistic modelling. However, they may involve unsubstantiated hypotheses and be computationally intensive. Either of these may prevent their scalability to large databases.
- (3) Least-squares approximation, at which the data are approximated with a low-rank model data matrix [10,13,22,23,26,38]. This approach is computationally effective. However, it is not sensitive to the shape of the underlying distribution, which can become an issue in imputing missing data from a complex distribution.

The low-rank approximation methods have proven to be a useful tool in psychology, at which they originated as the so-called principal component analysis [17], in data visualization [3], in information retrieval [4], and bioinformatics [16].

Methods within this approach minimize the sum of squared differences between the observed data entries and those reconstructed via bilinear modelling which is akin to the singular value decomposition (SVD) of a data matrix. Two ways to implement this approach can be distinguished:

- (1) Fit a low-rank data model by using non-missing entries only and then interpolate the missing values with values found according to the model [10,13,26,33,38]. This approach will be referred to as the Iterative least-squares algorithm ILS.
- (2) Start by filling in all missing entries with some values such as zero, then iteratively approximate thus completed data by updating the imputed values with those implied by a low-rank approximation [13,22,23]. This approach will be referred to as the Iterative majorization least-squares algorithm IMLS.

To describe these in more detail, let us start with an iterative procedure for finding the singular-value decomposition (SVD) of a data matrix, which is basic to each of them.

2.1 Notation

The data is considered in the format of a matrix \mathbf{X} with N rows and n columns. The rows are assumed to correspond to entities (observations) and columns to variables (features). The elements of \mathbf{X} are denoted by x_{ik} ($i = 1, \dots, N$, $k = 1, \dots, n$). The situation in which some entries (i, k) in \mathbf{X} are missed is modeled with an additional matrix $\mathbf{M} = (m_{ik})$ where $m_{ik} = 0$ if the (i, k) -th entry is missed and $m_{ik} = 1$, otherwise.

The matrices and vectors are denoted with boldface letters. A vector is always considered as a column; thus, the row vectors are denoted as transposes of the column vectors. Sometimes, the operation of matrix or inner product will be shown with symbol $*$.

2.2 Iterative Computation of a Singular Value Decomposition

Let us describe the concept of singular value decomposition of a matrix (SVD) in terms of a bilinear model for factor analysis of data matrices with no missing entries. This model assumes the existence of a number $p \geq 1$ of hidden factors that underlie the observed data as follows:

$$x_{ik} = \sum_{t=1}^p c_{tk} z_{it} + e_{ik}, \quad i = 1, \dots, N, \quad k = 1, \dots, n. \quad (1)$$

The (unknown) vectors $\mathbf{z}_t = (z_{it})$ and $\mathbf{c}_t = (c_{tk})$ are referred to as factor t scores for entities $i = 1, \dots, N$ and factor loadings for variables $k = 1, \dots, n$, respectively ($t = 1, \dots, p$) [17,26]. Values e_{ik} are residuals that are not explained by the model and should be made as small as possible.

To find approximating vectors $\mathbf{c}_t = (c_{tk})$ and $\mathbf{z}_t = (z_{it})$, one can minimize the least-squares criterion:

$$L_2 = \sum_{i=1}^N \sum_{k=1}^n (x_{ik} - \sum_{t=1}^p c_{tk} z_{it})^2. \quad (2)$$

In the follow-up we will rely on a strategy for minimizing criterion (2) over all p unknown factors by extracting them one-by-one, which is, basically, the contents of the method of principal component analysis, one of the major data mining techniques [19]. This strategy sometimes is referred to as the power method for SVD [12].

According to this method, computations are carried out iteratively. At each iteration $t = 1, \dots, p$ only one factor is sought by minimizing criterion

$$l_2(\mathbf{c}, \mathbf{z}) = \sum_{i=1}^N \sum_{k=1}^n (x_{ik} - c_k z_i)^2, \quad (3)$$

over real c_k and z_i with respect to the condition that \mathbf{c} is normalised, that is, $\sum_{k=1}^n c_k^2 = 1$. The condition is added to avoid multiple solutions as c_k and z_i are multiplied in (3). It is well-known that the singular triple $(\mu, \mathbf{z}, \mathbf{c})$ such that $\mathbf{X}\mathbf{c} = \mathbf{z}$ and $\mathbf{X}^T \mathbf{z} = \mu^2 \mathbf{c}$ with $\mu = \sqrt{\sum_{i=1}^N z_i^2}$, the norm of \mathbf{z} which is the maximum singular value of \mathbf{X} , solves the problem. Note an asymmetry in the described solution: \mathbf{c} is normalised while \mathbf{z} is not. The found vectors \mathbf{c} and \mathbf{z} are stored as \mathbf{c}_t and \mathbf{z}_t and next iteration $t + 1$ is performed. The matrix $\mathbf{X} = (x_{ik})$ is changed from iteration t to iteration $t + 1$ by subtracting the found solution according to the rule $x_{ik} \leftarrow x_{ik} - c_{tk} z_{it}$.

Since \mathbf{z}_t is not normalised in the described version of the algorithm, its norm is equal

to the singular value μ_t . This method always converges if the initial \mathbf{c} does not belong to the subspace already taken into account by the previously found singular vectors.

2.3 Iterative Least-Squares (ILS) Algorithm for Data Imputation

The ILS algorithm is based on the SVD method described above. However, this time equation (1) applies only to those entries that are not missed.

The idea of the method is to find the score and loading vectors in decomposition (1) by using only those entries that are available and then use (1) at missing entries to impute them (with the residuals ignored).

To find approximating vectors $\mathbf{c}_t = (c_{tk})$ and $\mathbf{z}_t = (z_{it})$, the least-squares criterion on the nonmissing entries can be written in the following form:

$$L_2 = \sum_{i=1}^N \sum_{k=1}^n e_{ik}^2 m_{ik} = \sum_{i=1}^N \sum_{k=1}^n (x_{ik} - \sum_{t=1}^p c_{tk} z_{it})^2 m_{ik}, \quad (4)$$

where $m_{ik} = 0$ at missing entries (i, k) and $m_{ik} = 1$, otherwise.

As is well known, the singular value decomposition cannot be applied to a matrix with missings. Thus the problem of minimization of L_2 in (4) becomes a difficult nonlinear optimisation problem. Note that the factors in (4) are not necessarily mutually orthogonal. To minimize criterion (4), the one-by-one strategy above is utilized. According to this strategy, computations are carried out iteratively. At each iteration t , $t = 1, \dots, p$, only one factor is sought by minimizing criterion:

$$l_2 = \sum_{i=1}^N \sum_{k=1}^n (x_{ik} - c_k z_i)^2 m_{ik}, \quad (5)$$

with respect to condition $\sum_{i=1}^N c_k^2 = 1$. The found vectors \mathbf{c} and \mathbf{z} are stored as \mathbf{c}_t and \mathbf{z}_t , non-missing data entries x_{ik} are substituted by $x_{ik} - c_k z_i$, and next iteration $t + 1$ is performed.

Note that, with missing entries, the one-by-one strategy does not guarantee that the found solution does minimise the least squares criterion (4) anymore.

To minimize (5), the method of alternating minimization is utilised. Each iteration proceeds in two steps: (1) given (c_k) , find optimal (z_i) ; (2) given (z_i) , find optimal (c_k) . Finding optimal score and loading vectors can be done according to equations following from the first-order optimality conditions:

$$z_i = \frac{\sum_{k=1}^n x_{ik} m_{ik} c_k}{\sum_{k=1}^n c_k^2 m_{ik}}. \quad (6)$$

and

$$c_k = \frac{\sum_{i=1}^N x_{ik} m_{ik} z_i}{\sum_{i=1}^N z_i^2 m_{ik}}. \quad (7)$$

Given \mathbf{z} , a \mathbf{c} is found with (7) and the follow-up normalization of the result. Then, given \mathbf{c} , next \mathbf{z} is found with (6). Basically, it is this procedure that was differently described in [10,13,26].

Two issues, which never occur at the SVD decomposition of complete data matrices, may emerge at ILS:

(1) Convergence.

The method may fail to converge depending on the configuration of missing entries and the initial setting. Some other causes of non-convergence as those described by Grung and Manne [13] have been taken care of in our formulation of the algorithm.

In the present approach we, somewhat simplistically, use the normalised vector of ones as the starting point. Sometimes a more sophisticated choice may be required because the iterations may come to a “wrong convergence” or not converge at all. To this end, Gabriel and Zamir [10] developed a method to use a row of \mathbf{X} to build an initial \mathbf{c}^* , as follows:

1. Find (i, k) with the maximum

$$\omega_{ik} = \sum_b m_{bk} x_{bk}^2 + \sum_d m_{id} x_{id}^2, \quad (8)$$

over those (i, k) for which $m_{ik} = 0$.

2. With these i and k , compute

$$\beta = \frac{\sum_{b \neq i} \sum_{d \neq k} m_{bd} x_{bk}^2 x_{id}^2}{\sum_{b \neq i} \sum_{d \neq k} m_{bd} x_{bk} x_{id} x_{bd}}. \quad (9)$$

3. Set the following vector as the initial at ILS step 2:

$$\mathbf{c}^{*l} = (x_{i1} \dots x_{ik-1}, \beta, x_{ik+1} \dots, x_{in}). \quad (10)$$

This method is computationally intensive and may cause to slow down the speed of computation (up to 60 times in our experiments). However, it can be useful indeed when the size of the data is small.

(2) Number of factors.

When the number of factors is equal to one, $p = 1$, ILS is equivalent to the method introduced by Wold [38] and his student Christofferson [5] under the name “nonlinear iterative partial least squares” (NIPALS). In most cases the one-factor technique leads to significant errors so that more factors may be needed.

Selection of p may be driven by the same scoring function as selection of the number of principal components: by the proportion of the data variance taken into account by factors. This logic is well justified in the case of the principal component analysis at which model (1) fits the data exactly when p is equal to the rank of \mathbf{X} . With missings present in data, the number of ILS iterations can be infinite; however, the logic is still justified since it can be shown that the residual data matrix converges to zero when the number of iterations grows. Actually, the convergence follows from Statement 2.2 in [26] that covers a wide class of iterative data extraction techniques.

2.4 Iterative Majorization Least-Squares (IMLS) Algorithm

This method is a specification of the general idea that any weighted least-squares minimization problem can be addressed as a series of non-weighted least-squares minimization problems with iteratively adjusting found solutions according to a so-called majorization function [15]. In this framework, Kiers [22] developed an approach to the problem of imputation, a version of which can be formulated in the present context without using any concept beyond those previously specified. The algorithm starts with a complete data matrix and updates it by relying on both non-missing entries and estimates of missing entries.

The algorithm is similar to ILS except for the fact that it employs a different iterative procedure for finding a factor, which will be referred to as Factor algorithm and described first. Factor algorithm operates with a completed version of matrix \mathbf{X} that will be denoted by \mathbf{X}^s where $s = 0, 1, \dots$ is the iteration's number. At each iteration s , the algorithm finds the best SVD factor for \mathbf{X}^s and imputes the results into missing entries, after which the next iteration starts.

The general setting in [22] refers to any weighted least-squares problem and does not require restricting \mathbf{X}^s to one factor only. Computationally, finding several SVD factors simultaneously is as easy as just one. We restrict ourselves with one factor only because more factors would better approximate the arbitrary values put instead of missing entries to complete \mathbf{X} and, thus, be more biased towards these ad hoc values.

Factor Algorithm

1. Set $\mathbf{c}' = (1, \dots, 1)$ and normalise it.
2. Set $s = 0$; define matrix \mathbf{X}^s by putting zeros into missing entries of \mathbf{X} .
Set measure of quality $h_s = \sum_{i=1}^N \sum_{k=1}^n (x_{ik}^s)^2$.
3. Find the first singular triple $\mathbf{z}_1, \mathbf{c}_1, \mu$ for matrix \mathbf{X}^s by applying the iterative SVD algorithm with $p = 1$ and take the resulting value of criterion (3) as h_{s+1} .
4. If $|h_s - h_{s+1}| > \epsilon * h_s$ for a small $\epsilon > 0$, set $s = s + 1$, put $z_{i1}c_{1k}$ for each missing entry (i, k) in \mathbf{X} and go back to step 3.
5. Set \mathbf{z}_1 and \mathbf{c}_1 as the output.

Now a version of IMLS algorithm [22] can be formulated as follows.

IMLS Algorithm

0. Set the number of factors p .
1. Set iteration number $t=1$.
2. Apply Factor algorithm to matrix \mathbf{X} with the missing structure \mathbf{M} .
Denote results by \mathbf{z}_t and \mathbf{c}_t .
3. If $t < p$, for each (i, k) such that $m_{ik} = 1$, update $x_{ik} = x_{ik} - c_{tk}z_{it}$, put $t=t+1$ and go to step 2.
4. Impute missing values x_{ik} at $m_{ik} = 0$ according to (1) with $e_{ik} = 0$.

Theoretical properties of the IMLS method remain to be explored.

A similar algorithm, but with an additional tuning, was developed in [23]. In the version of [23], missing entries are imputed one-by-one, so that to fill in entry (i, k) , vectors \mathbf{c} and \mathbf{z} are found for the matrix \mathbf{X} with the either i -th row or k -th column removed. This version is not used in our experiments, because its results may depend on the order of imputation of missing entries and, moreover, at large data sizes and many missings, we see no advantages in the one entry based tuning.

The number of factors p in ILS is limited by poor convergence of the method at matrices with small elements; in IMLS, this number also cannot be large: otherwise, the initial arbitrarily imputed values will affect the final result. In the follow-up experiments, two options will be selected: $p = 1$, the minimum number, and $p = 4$, to have a deeper coverage of the data, but still rather small in comparison with the ranks of generated data matrices (from 15 to 20).

3 Nearest Neighbour Based Data Imputation

3.1 Nearest Neighbour Approach

Let us apply the machine learning framework [1,2] at which imputations are carried out by analysing entities with missing entries one-by-one. An entity containing one or more missing entries which are to be imputed is referred to as a target entity. According to the nearest neighbour (NN) approach, a distance measure is computed between the target entity and each of the other entities and then K nearest to the target entities are selected. An imputation model such as that based on (1) with $e_{ik} = 0$ for the target entity is found by using a shortened version of \mathbf{X} to contain only K+1 elements: the target and K selected neighbours. The intuition behind using only nearest neighbours is that least-squares approximation methods may become more sensitive to the data structure this way.

To apply the NN approach, the following two issues should be addressed.

- (1) **Measuring distance.** There can be a multitude of distance measures considered. We choose the Euclidean distance squared because this measure is compatible with the least-squares framework. The distance between a target entity \mathbf{X}_i and an entity \mathbf{X}_j is defined as:

$$D_2(\mathbf{X}_i, \mathbf{X}_j, \mathbf{M}) = \sum_{k=1}^n [x_{ik} - x_{jk}]^2 m_{ik} m_{jk}; i, j = 1, 2, \dots, N, \quad (11)$$

where m_{ik} and m_{jk} are missingness values for x_{ik} and x_{jk} , respectively. This distance was also used in [14,27,36].

- (2) **Selection of the neighbourhood.** The principle of selecting the nearest entities can be implemented, first, as is, on the set of all entities, and, second, by considering only entities with non-missing entries in the attribute corresponding to that of the target's missing entry. The second approach was applied in [14,36] for data imputation with the method Mean. We apply the same approach as well when using this method. However, for ILS and IMLS, the presence of missing entries in the neighbours typically creates no problems, and, with these methods, we select neighbours among all entities.

3.2 Nearest Neighbour Imputation Algorithms

Briefly, the NN based imputation techniques can be formulated as follows: take the first row that contains a missing entry as the target entity \mathbf{X}_i , find its K nearest neighbours and form a shortened matrix \mathbf{X}_c consisting of the target entity and the neighbours. Then apply an imputation algorithm $A(\mathbf{X}, \mathbf{M})$ to the shortened matrix \mathbf{X}_c with imputing

missing entries at the target entity only. Repeat this until all missing entries are filled in. Then output the completed data matrix \mathbf{X} [37].

To make the NN based imputation algorithms work fast, we choose K of the order from 5 to 10. Then, in applying the least-squares imputation techniques, we have to restrict ourselves to small numbers of factors in order to guarantee that the subspace approximation processes converge. Thus, with ILS, take $p = 1$ and use the Gabriel-Zamir's initialisation. Still, ILS algorithm may not converge sometimes because of the small NN data sizes.

3.3 Global-Local Imputation Algorithm: INI

A two-stage approach from [37] combines the NN and global imputation approaches. First stage: Use a global imputation technique to fill in all the missings in matrix \mathbf{X} . Let us denote the resulting matrix \mathbf{X}^* . Second stage: Apply a NN based imputation technique to fill in the missings in \mathbf{X} , but this time based on distances computed with the completed data \mathbf{X}^* . These distances will be referred to as *prime distances*.

We specify this global-local approach by using IMLS at both of the stages, which is referred to as the INI algorithm (a shortened triple denotation IMLS-NN-IMLS [37]). INI consists of four steps. First, impute missing values in the data matrix \mathbf{X} by using IMLS with $p = 4$. Then compute the prime distance metric with thus found \mathbf{X}^* . Third, take a target entity according to \mathbf{X} and find its neighbours according to the prime distance. Finally, impute all missing entries in the target entity with a NN version of IMLS algorithm (this time with $p = 1$).

INI Algorithm

1. Apply IMLS algorithm to \mathbf{X} with $p = 4$ to impute all missing entries in matrix \mathbf{X} ; denote resulting matrix by \mathbf{X}^* .
2. Take the first row in \mathbf{X} that contains a missing entry as the target entity \mathbf{X}_i .
3. Find K neighbours of \mathbf{X}_i on matrix \mathbf{X}^* .
4. Create a data matrix \mathbf{X}_c consisting of \mathbf{X}_i and rows of \mathbf{X} corresponding to the selected K neighbours.
5. Apply IMLS algorithm with $p = 1$ to \mathbf{X}_c and impute missing values in \mathbf{X}_i .
6. If no missing entries remain, stop; otherwise go back to step 2.

4 Experimental Study

The goal of the experimental study is to compare different methods of imputation on various data sets and patterns of missing data.

4.1 Selection of Algorithms

Based on the considerations above, the following eight least-squares data imputation algorithms will be considered as a representative selection:

- (1) ILS-1 or NIPALS: ILS with $p = 1$.
- (2) ILS: ILS with $p = 4$.
- (3) ILS-GZ or GZ: ILS with the Gabriel-Zamir procedure for initial settings.
- (4) IMLS-1: IMLS with $p = 1$.
- (5) IMLS-4: IMLS with $p = 4$.
- (6) N-ILS: NN based NIPALS.
- (7) N-IMLS: NN based IMLS-1.
- (8) INI: NN based IMLS-1 imputation based on distances from IMLS-4 imputation.

For the purposes of comparison, two mean scoring algorithms have been added:

- (9) Mean: Imputing the average column value.
- (10) N-Mean: NN based Mean.

In the follow-up experiments, the NN based techniques will operate with $K=10$ which is about 5% of the numbers of entities in generated datasets.

4.2 Data Generation

4.2.1 NetLab Gaussian Mixture Data Model

To generate data, we employ the Gaussian mixture data model which is described in many sources (see, for instance, [9]). We refer to a mixture of m Gaussian distributions (classes) as a Gaussian m -mixture. Within this model, we select an option at which a data matrix $\mathbf{D}_{N \times n}$ is generated randomly from the Gaussian mixture with a probabilistic principal component analysis (PPCA) covariance matrix [29,35].

The following three-step procedure, Neural Network NetLab, is applied as implemented in a MATLAB Toolbox freely available on the web [11]:

- (1) *Architecture*: set the dimension of data equal to n , number of classes (Gaussian distributions) to m and the type of covariance matrix as based on PPCA in a q

dimension subspace. In our experiments, m is 3 or 5, n between 15 and 25, and q is either $n - 3$ or $\lceil n/2 \rceil$.

- (2) *Data Structure*: create a Gaussian mixture model with the mixing coefficient equal to $1/m$ for each class. A Gaussian distribution for each i -th class ($i = 1, \dots, m$) is defined as follows: components of a random n -dimensional vector \mathbf{avg}_i are generated according to Gaussian distribution $N(0, 1)$. The $n \times q$ matrix of the first q loading n -dimensional vectors is defined as:

$$\mathbf{W}_q = \begin{pmatrix} \mathbf{I}_{q \times q} \\ \mathbf{1}_{(n-q) \times q} \end{pmatrix}, \quad (12)$$

where $\mathbf{I}_{q \times q}$ and $\mathbf{1}_{(n-q) \times q}$ are the identity matrix and matrix of ones, respectively. The covariance matrix is then defined as:

$$\mathbf{Cov}(\sigma) = \mathbf{W}_q * \mathbf{W}_q' + \sigma^2 \mathbf{I}_{n \times n}. \quad (13)$$

In our experiments, the general variance σ^2 of PPCA is set to be equal to 0.1.

- (3) *Data*: generate randomly data matrix $\mathbf{D}_{N \times n}$ from a Gaussian m -mixture distribution as follows:

NetLab Gaussian m -mixture

Compute eigenvalues and corresponding eigenvectors of $\mathbf{Cov}(\sigma)$ and denote the matrix of eigenvectors by \mathbf{evec} and vector of the square roots of eigenvalues by $\sqrt{\mathbf{eigen}}$.

For $i = 1, \dots, m$:

Set $N_i = N/m$, the number of rows in i -th class.

Generate randomly $\mathbf{R}_{(N_i \times n)}$ based on the Gaussian distribution $N(0,1)$.

*Compute $\mathbf{D}_i = \mathbf{avg}_i + \mathbf{R} * \mathbf{diag}(\sqrt{\mathbf{eigen}}) * \mathbf{evec}'$.*

end

Define \mathbf{D} as $N \times n$ matrix combining all generated matrices \mathbf{D}_i , $i = 1, \dots, m$.

4.2.2 Exploration of NetLab Gaussian Mixture Data Model

The structure of (12) is rather simple and so is the structure of covariance matrix (13) as well. It is not difficult to show that

$$\mathbf{Cov}(0) = \begin{pmatrix} \mathbf{I}_{q \times q} & \mathbf{1}_{q \times (n-q)} \\ \mathbf{1}_{(n-q) \times q} & q\mathbf{1}_{(n-q) \times (n-q)} \end{pmatrix}, \quad (14)$$

where \mathbf{I} and $\mathbf{1}$ are the identity and all-ones matrices, respectively.

This matrix's rank is q as follows from the definition (12) and the structure in (14) as well.

To explore the eigenvalues of $\mathbf{Cov}(0)$, let us consider an n -dimensional vector \mathbf{x} in the format $\mathbf{x} = (\mathbf{x}_q, \mathbf{x}_{n-q})$ where \mathbf{x}_q and \mathbf{x}_{n-q} denote subvectors with q and $n - q$ components, respectively. Also denote the sum of elements of \mathbf{x}_q by a and the sum of elements of \mathbf{x}_{n-q} by b . Obviously, to be an eigenvector of $\mathbf{Cov}(0)$ corresponding to its eigenvalue λ , \mathbf{x} must satisfy the following equations: $\mathbf{x}_q + b\mathbf{1}_q = \lambda\mathbf{x}_q$ and $(a + qb)\mathbf{1}_{n-q} = \lambda\mathbf{x}_{n-q}$. Summing up components of these vector equations leads to (i) $a + bq = \lambda a$ and (ii) $(a + bq)(n - q) = \lambda b$, respectively. With little arithmetics, these imply that $\mathbf{Cov}(0)$ has only two nonzero eigenvalues, the maximum $\lambda = 1 + (n - q)q$ and second-best $\lambda = 1$.

Indeed, let us see first that $a = 0$ implies $b = 0$ and $\lambda = 1$. Having put $a = 0$ into (i) one obviously gets $b = 0$ as well. This implies that $a + bq = 0$ so that $(a + bq)\mathbf{1}_{n-q} = \lambda\mathbf{x}_{n-q}$ can hold only at $\mathbf{x}_{n-q} = 0$, provided that $\lambda \neq 0$. Similarly, $\mathbf{x}_q + b\mathbf{1}_q = \lambda\mathbf{x}_q$ can hold only if $\mathbf{x}_q = \lambda\mathbf{x}_q$, that is, if $\lambda = 1$, which proves that $\lambda = 1$ is an eigenvalue. Moreover, the rank of the subspace of eigenvectors corresponding to $\lambda = 1$ is equal to $q - 1$, because they all are defined by the condition that the sum of their components $a = 0$.

Let us now assume that a is not zero. Equation (i) implies that λa can be put for $a + qb$ in (ii), leading to $\lambda a(n - q) = \lambda b$. Thus, with $\lambda \neq 0$, $a(n - q) = b$ and $b/a = n - q$. But $\lambda = 1 + qb/a$ according to (i), which leads to $\lambda = 1 + q(n - q)$ and proves the statement.

In the eigenvector corresponding to the maximum eigenvalue, part \mathbf{x}_q consists of the same components and, similarly, elements of \mathbf{x}_{n-q} are the same. Part \mathbf{x}_{n-q} of the eigenvector corresponding to $\lambda = 1$ is zero. Also, part \mathbf{x}_q of eigenvectors corresponding to $\lambda = 0$ consists of the same values.

Obviously, having n and q of the order of 20 and 3, respectively, makes the maximum eigenvalue $\lambda = 1 + (n - q)q$ equal to 52, which leads to an overwhelming presence of the maximum eigenvalue and corresponding eigenvector in the data generated according to the model above. That is, the data formally generated from a mixture of multidimensional Gaussian distributions are approximately distributed along the first eigenvector, thus tending to form a blot. Changing σ in $\mathbf{Cov}(\sigma)$ to an arbitrary value does not change eigenvectors but adds σ^2 to the eigenvalues. Even with σ approaching unity, the contribution of the first factor remains high.

4.2.3 Scaled NetLab Gaussian Mixture Data Model

To increase the complexity of generated data sets, the Gaussian mixture data model above will be modified by differently scaling the covariance matrix $\mathbf{Cov}(\sigma)$ and the mean vector \mathbf{avg} for each class. To do this, for each Gaussian class $i = 1, \dots, m$, a random scaling factor, b_i , is generated to move \mathbf{avg}_i away from the origin. Then the covariance matrix is scaled by a factor, a_i , to be taken proportional to i . The dimension of the probabilistic principal component analysis model is taken as $q = n/2$ (at an even n). The modified data generator can be summarized as follows:

Scaled NetLab Gaussian m -mixture

For $i = 1, \dots, m$, given \mathbf{avg}_i and $\mathbf{Cov}_i(\sigma)$:

Randomly generate the scaling factor b_i in the range between 5 and 15.

Compute scaled \mathbf{Cov}_i as $\mathbf{Cov}_i = 0.8 * i * b_i * \mathbf{Cov}(\sigma)$.

Compute eigenvalues and corresponding eigenvectors of \mathbf{Cov}_i ; denote the matrix of eigenvectors by \mathbf{evec}_i and vector of the square roots of eigenvalues by $\sqrt{\mathbf{eigen}_i}$.

Set $N_i = N/m$, the number of rows in i -th class.

Generate randomly $\mathbf{R}_{(N_i \times n)}$ according to Gaussian distribution $N(0,1)$.

Compute $\mathbf{D}_i = b_i * \mathbf{avg}_i + \mathbf{R} * \text{diag}(\sqrt{\mathbf{eigen}_i}) * \mathbf{evec}_i'$.

end

Define \mathbf{D} as $N \times n$ matrix combining all generated matrices \mathbf{D}_i .

Thus generated data set is obviously spread over rather distant cluster centres with differently scaled cluster covariance matrices.

4.3 Generation of patterns of missing data

As described above, three types of patterns of missing data are considered: Random missing RM, Restricted random missing RRM, and Merged database MD. The further text details how each of the patterns is generated.

4.3.1 Random missing

Given a data table generated, a pattern of missing entries is produced randomly on a matrix of the size of the data table with a pre-specified proportion of the missings. We use the random uniform distribution for generating missing positions and the propor-

tion's range at six different values accounting for 1%, 5%, 10%, 15%, 20%, or 25% of the total number of entries.

4.3.2 Restricted random missing

According to this model, missings may occur only in a subset of entities with respect to a subset of issues. In our experiments, additional constraints on the sizes of sets of rows and columns at which missing entries may occur have been maintained to avoid trivial patterns of missing data. The missings under this scenario are carried out as follows:

Generation of a Restricted Missing Pattern

Given proportion p of missings entries, randomly select proportions s of related issues (columns) and r of related respondents (rows) such that $p < sr$.

In the data submatrix formed by the selected s columns and r rows randomly generate proportion p/sr missings.

Accept the following additional constraints on the values generated:

1. $10\% < s < 50\%$ and $25\% < r < 50\%$ for $p = 1\%$.
2. $20\% < s < 50\%$ and $25\% < r < 50\%$ for $p = 5\%$.
3. $25\% < s < 50\%$ and $40\% < r < 80\%$ for $p = 10\%$.

4.3.3 Merged database pattern

In this pattern, a data base is supposed to have been obtained by merging two databases so that missings may result only from the absence of certain features in either of the merged databases or both of them. Accordingly, two types of scenarios for merging databases are to be considered:

- (1) MD-1: Missings come from only one database.
- (2) MD-2: Missings come from both of the databases.

4.3.3.1 Missings from one database MD-1

Under this scenario, the missings are generated as follows. First, specify the proportion p of missing entries in the merged database. Then generate proportion s of columns to have missing entries in the merged database. These are assumed to come from the database at which the corresponding variables are missed. Finally, the proportion of respondents (rows) in the "damaged" database is computed as $t = p/s$.

In the experiments, $s=20\%$ or 30% are selected for generating $p = 1\%$ or 5% missings.

We consider that greater proportions of missings are not realistic for filling in with this pattern.

4.3.3.2 Missings from two databases MD-2

Suppose each of the two databases to be merged contains variables that are absent in the other database. The merged database will have a pattern which can be presented as follows: the variables which are absent only from the first database are placed on the left while variables that are absent only from the second database are placed on the right.

Obviously, if N_1 and N_2 are the numbers of rows and k_1 and k_2 are the numbers of missing columns in the respective databases then the total proportion of missings can be calculated as

$$p = \frac{k_1 N_1 + k_2 N_2}{nN}, \quad (15)$$

where $N = N_1 + N_2$ and $k_1 + k_2 < n$. This implies that, given p , k_1 and k_2 are not independent so that k_2 can be determined by k_1 with the following equation:

$$k_2 = \frac{pnN - k_1 N_1}{N_2}. \quad (16)$$

This allows us to introduce a procedure for generation of missings of this type by putting proportion p of missings first, as we did with the other patterns of missing data. Furthermore, it is assumed that one of the databases is somewhat but not overwhelmingly greater than the other, say N_1 may be greater than N_2 1.5 to 4 times. This assumption corresponds to N_1/N being between 0.6 and 0.8.

Generation of missings from two databases

1. Specify the proportion p of missings entries.
2. Specify the number of rows N and columns n in the merged database.
Then randomly generate the number of rows in the first database, N_1 , subject to constraint $0.6 < N_1/N < 0.8$ and define the number of entities in the second database, $N_2 = N - N_1$.
3. Randomly generate integer k_1 such that $k_1 < \frac{npN - N_2}{N_1}$.
4. Compute k_2 according to equation (16).
5. Finally, put $m_{ik} = 0$ for all $i = 1, \dots, N_1$, $k = 1, \dots, k_1$ and for all $i = N_1 + 1, \dots, N$, $k = n, n - 1, \dots, n - k_2 + 1$.

4.4 Scoring Results

Since the data and missings are generated separately, the quality of imputation can be scored by comparing the imputed values with those generated at the stage of data generation. The squared imputation error, IE , will be used to measure the performance of an algorithm. The measure is defined as follows:

$$IE = \frac{\sum_{i=1}^N \sum_{k=1}^n (1 - m_{ik})(x_{ik} - x_{ik}^*)^2}{\sum_{i=1}^N \sum_{k=1}^n (1 - m_{ik})x_{ik}^2}, \quad (17)$$

where m_{ik} is the missingness matrix entry and x_{ik}^* an entry in the data matrix \mathbf{X}^* with imputed values. This measure is compatible with the least squares framework which is considered in this paper.

5 Experimental Results

The results will be presented separately for each of the three patterns of missing data specified above: Random missings RM, Restricted random Missings RRM and Merged database MD.

5.1 General Results

The experiments have been carried out with data sets generated according to each original and scaled NetLab 5-mixture data model with the dimension of the principal component subspace equal to $q = n - 3$ and $q = n/2$, respectively. Ten data sets have been generated according to each of the two models; sizes of the data sets were taken randomly from intervals of $N=200 - 250$ rows and $n=15 - 25$ columns. The data size has no meaning in our experiments; by varying it, we decrease the number of arbitrarily specified parameter values unavoidable in any experiment.

For each of the data sets and each of the considered missing types and levels of missings p , six patterns of missing data have been generated, and each of the ten algorithms has been run at each data/missing pattern. The results have been scored according to criterion IE defined in equation (17).

Tables 1 and 2 present errors averaged within corresponding categories; standard deviations are in the parentheses. For the sake of space, errors at intermediate levels of missing are omitted, but they naturally follow regularities seen in the tables.

Table 1

The average squared error of imputation and its standard deviation (%) at NetLab Gaussian 5-mixture data model with different patterns of missing data.

<i>Methods</i>	<i>Random</i>		<i>Restricted random</i>		<i>MD-1 with q = 30%</i>		<i>MD-2</i>
	1%	25%	1%	10%	1%	5%	5%
ILS	41.25 (15.00)	48.56 (9.58)	44.97 (19.94)	37.02 (8.00) (*)	48.82(28.10)	57.19(27.68)	26.12(13.17)(*)
GZ	41.26 (15.01)	48.25 (8.99)	44.97 (19.94)	35.59 (6.40)	48.82(28.10)	57.36(27.76)	25.91(13.09)(*)
NIPALS	56.49 (20.12)	55.25 (9.51)	54.15 (20.04)	42.76 (9.80)	61.84(29.51)	67.83(29.05)	16.76(10.40)(*)
IMLS-1	56.59 (20.19)	52.35 (5.80)	54.15 (21.04)	42.86 (9.73)	62.00(29.43)	67.86(29.05)	80.20(61.56)
IMLS-4	41.25 (14.99)	48.56 (6.41)	44.60 (21.04)	57.58 (69.91)	49.03(27.71)	56.77(26.90)	84.00(64.96)
Mean	97.13 (8.50)	95.62 (1.51)	92.78 (7.08)	96.67 (3.66)	94.38(9.72)	93.53(6.20)	128.36(37.14)
N-ILS	35.14 (14.02)	69.35(23.20)(*)	37.57(19.42)	38.11 (12.44)	43.17(27.57)	94.59(13.12)	NA
N-IMLS	35.04 (13.96)	66.75 (19.06)	37.45 (19.31)	33.17 (5.25)	42.87(27.23)	49.06(24.27)	69.62(12.85)
INI	35.29 (13.03)	43.01 (8.05)	39.27 (20.03)	37.02 (15.18)	41.54(25.07)	49.11(23.91)	71.10(11.84)
N-Mean	37.66 (13.19)	80.98 (8.58)	42.25(20.33)	93.26 (27.13)	90.66(48.33)	99.88(52.87)	70.43(39.41)

* Does not converge sometimes. The figure is calculated over converged cases only.

NA - Not applicable.

First, some expected regularities:

- (1) Method Mean appears the worst at about constant rate of error, 100%.
- (2) Errors typically grow with the growth of the level of missings.
- (3) Least-squares based methods form three groups of similarly performing methods: NN based methods, one rank based global methods, and other global methods (ILS, GZ, IMLS-4).
- (4) NN based methods outperform global methods always except with MD-2 pattern at which global methods prevail, at least in situations at which they converge.
- (5) Of NN based methods N-IMLS and INI are the best; they show similar results in most situations.

Table 2

The average squared error of imputation and its standard deviation (%) at Scaled NetLab Gaussian 5-mixture data model with different patterns of missing data.

<i>Methods</i>	<i>Random</i>		<i>Restricted random</i>		<i>MD-1 with q = 30%</i>		<i>MD-2</i>
	1%	25%	1%	10%	1%	5%	5%
ILS	16.75 (7.52)	21.65 (5.26)	15.87 (6.13)	25.64 (10.87) (*)	21.48(13.41)	20.44(8.46)	56.41(30.80)(*)
GZ	16.75 (7.52)	21.67 (5.27)	15.86 (6.13)	25.63 (10.86)	21.48(13.41)	20.33(8.09)	18.84(8.13)
NIPALS	62.37 (17.41)	64.19(11.12)	68.47 (16.16)	64.71 (13.41)	69.70(28.11)	63.70(19.84)	16.07(9.52)
IMLS-1	62.30 (17.47)	64.15(10.97)	68.70 (16.24)	64.60 (13.30)	69.64(27.84)	63.73(20.02)	15.67(9.27)
IMLS-4	16.79 (7.49)	21.65 (5.17)	16.08 (6.36)	25.65 (10.31)	21.42(13.40)	20.56(8.92)	24.83(21.13)
Mean	90.46 (11.36)	89.61 (6.08)	95.43 (9.04)	91.98 (8.25)	88.88(11.77)	90.48(11.35)	103.88(3.75)
N-ILS	7.31 (3.39)	7.90(1.28)(*)	7.15(2.94)	7.28 (1.87)	8.92(5.51)	7.75(3.47)	NA
N-IMLS	7.30 (3.37)	8.73 (1.55)	7.15 (2.92)	7.28 (1.85)	8.90(5.50)	7.72(3.47)	49.45(11.96)
INI	7.47 (3.19)	9.74(1.90)	6.83 (3.07)	11.39 (7.96)	10.31(9.59)	8.75(3.75)	45.06(13.14)
N-Mean	14.50 (6.77)	97.54 (18.12)	28.58(12.99)	246.13 (84.93)	162.37(94.23)	152.03(72.96)	30.67(7.93)

* Does not converge sometimes. The figure is calculated over converged cases only.

NA - Not applicable.

Second, unexpected results:

- (1) The level of errors at each of the eight least-square methods is typically much smaller for the scaled mixture model than for the original one. This probably can be attributed to the fact that data are spread differently at different directions with the scaled model, which conforms to the one-by-one factor extraction procedure underlying the methods.

- (2) The level of errors of N-Mean method approaches that of the NN based least-squares techniques when the level of missing is small (except for MD-1 pattern), and it drastically increases with the increase of the level of missings.
- (3) Errors do not grow with the growth of the level of missings at RRM patterns with data generated according to the unscaled NetLab model.
- (4) At the MD-2 missing type, unidimensional methods NIPALS and IMLS-1 may win over their four-dimensional analogues.

These conclusions may be blurred by the overlapping standard deviations of the methods' average errors. Therefore, further on we present results of direct pairwise comparisons between the methods at different missing models.

5.2 Random missings

With random missings, there are three different patterns in the pairwise comparisons: (1) at 1% missings, (2) at 5 % missings, and (3) at 10% and more missings. These patterns are shown in Table 3. They follow the features seen in the average error Table 1. However, there appears a regularity, which has not been seen at that table.

At 1% missings, all four NN based methods perform better than the rest. Although N-Mean loses to INI by 30% to 70%, it outperforms the others in winning over unidimensional methods, NIPALS and IMLS-1. This, to an extent, supports the results of experiments with N-Mean in [37] when the proportion of missings is small. However, when the proportion of missings increases to 5% and more, the N-Mean method loses to all the least-squares imputation methods except for the unidimensional ones, NIPALS and IMLS-1 (see Table 3). It should be pointed out that the least squares imputation methods have not been considered in [37]. When the proportion of missings grows further on, INI becomes the only winner (see the right-hand part of Table 3 presenting a typical pattern at 10%-25% missings).

Table 3

The pair-wise comparison of the methods; entry (i, j) shows how many times in % method j outperformed method i on Gaussian 5-mixture with $q = n - 3$ factors for 1%, 5%, and 15% random missing data.

Method	1%				5%				15%			
	N-ILS	N-IMLS	INI	N-Mean	N-ILS	N-IMLS	INI	N-Mean	N-ILS	N-IMLS	INI	N-Mean
ILS	60	60	60	60	80	80	100	10	40	50	100	0
GZ	60	60	60	60	80	80	100	10	40	60	100	0
NIPALS	90	90	100	100	100	100	100	90	60	100	100	10
IMLS-1	90	90	100	100	100	100	100	90	70	100	100	10
IMLS-4	70	70	80	60	80	90	100	20	40	60	90	0
Mean	100	100	100	100	100	100	100	100	80	100	100	100
N-ILS	-	100	50	50	-	70	70	20	-	100	100	0
N-IMLS	0	-	50	50	30	-	70	20	0	-	100	0
INI	50	50	-	30	30	30	-	0	0	0	-	0
N-Mean	50	50	70	-	80	80	100	-	80	100	100	-

A similar pattern emerges with data generated according to the scaled NetLab model when the level of missings is small, 10% or less. However, at greater levels of missings with this data model, INI is not the winner anymore (see Table 4).

Table 4

The pair-wise comparison of the methods; an entry (i, j) shows how many times in % method j outperformed method i on scaled NetLab Gaussian 5-mixture with $n/2$ PPCA factors for 15%, 20% and 25% Random missing data.

<i>Method</i>	15%				20%				25%			
	N-ILS	N-IMLS	INI	N-Mean	N-ILS	N-IMLS	INI	N-Mean	N-ILS	N-IMLS	INI	N-Mean
ILS	100	100	100	0	100	100	100	0	90	100	100	0
GZ	100	100	100	0	100	100	100	0	90	100	100	0
NIPALS	100	100	100	20	100	100	100	0	90	100	100	0
IMLS-1	100	100	100	20	100	100	100	0	90	100	100	0
IMLS-4	100	100	100	0	100	100	100	0	90	100	100	0
Mean	100	100	100	70	100	100	100	40	90	100	100	50
N-ILS	-	90	40	0	-	60	20	0	-	100	60	10
N-IMLS	10	-	40	0	40	-	20	0	0	-	50	0
INI	60	60	-	10	80	80	-	0	40	50	-	0
N-Mean	100	100	100	-	100	100	100	-	90	100	100	-

5.3 Restricted Missing Pattern

At the Restricted missing pattern, we limited the level of missings to be not more than 10% because missing entries are now confined within a relatively small submatrix of the data matrix.

As indicated earlier, with this pattern of missing data the error of imputation does not monotonely follow the growth of the number of missing entries.

Judging by the average errors, NN based least squares methods N-IMLS, N-ILS and INI surpass the other methods, and no obvious winner can be indicated among them. The first of these statements is confirmed at the level of one-to-one contests at both of the data models: the NN based methods win unanimously with the scaled NetLab model and almost unanimously with the unscaled model. There is one exception, though: at both data models, N-Mean loses to all other methods except for Mean, NIPALS and IMLS-1 at 1% of missings and it loses to all other methods when the proportion of missings grows further. At the scaled data model, Mean outperforms N-Mean at higher levels of missings, probably because Mean relies on more data with no missings at all at the RRM pattern with this data model.

NN based least squares methods perform differently at different data models. At the unscaled data model, with 1% and 5% missings, N-IMLS is the winner, and INI wins at 10% missings. The situation is opposite at the scaled model, which is reflected in Table 5.

Table 5

The pair-wise comparisons between three NN based least-squares imputation methods at unscaled and scaled NetLab Gaussian 5-mixture models for 1%, 5% and 10% missings from Restricted random pattern of missing data. (i, j) entry a/b shows that method j outperformed method i a% of the time on the unscaled model and b% of the time on the scaled model.

<i>Method</i>	1%			5%			10%		
	N-ILS	N-IMLS	INI	N-ILS	N-IMLS	INI	N-ILS	N-IMLS	INI
N-ILS	-	80/60	30/60	-	70/70	50/60	-	100/80	80/20
N-IMLS	20/40	-	30/60	30/30	-	50/60	0/20	-	60/20
INI	70/40	70/40	-	50/40	50/40	-	20/80	40/80	-

5.4 Merged Database Patterns

Missings from one database MD-1

With this pattern, all global least-squares imputation techniques show poor convergence at 10% missings or more.

At $p = 1\%$ or 5% , we consider either of two values of proportion s of columns absent from the incomplete database, 20% or 30%, which appears not affecting levels of errors at the scaled NetLab data model. At the unscaled data model, however, the levels of errors are somewhat less at the greater proportion, $s=30\%$. This may be attributed to the fact that, in our setting, the number of rows N_1 containing missings entries decreases when the number of absent columns grows. This decrease may be the factor improving the performance. Furthermore, all methods, N-ILS and ILS included, converge here probably because of smaller proportions of the overall missings.

According to pair-wise comparisons of the methods, INI is the best at the unscaled data model and the scaled data model at $s = 20\%$. However, INI gives way to N-IMLS at $s = 30\%$ with the scaled data model.

Missings from two databases MD-2

Only 5% missings level is considered at MD-2 missing model because 1% missing level is unfeasible at the data sizes generated. The results of pair-wise comparisons of methods in Table 6, in general, confirm results reported in Tables 1 and 2. However, they also correct them in some cases.

Table 6 shows that N-Mean beats the other methods at the unscaled data generation model. This seems to contradict the last column in Table 1 at which unidimensional least-squares methods on average perform much better. The difference is due to the way of calculation of figures in Table 1. The errors are calculated for converged cases only while Table 6 shows real contest results: when a method does not converge, those convergent beat it.

IMLS-1 appears to be the only winner at the scaled data model. The difference proba-

Table 6

The pair-wise comparison of methods at MD-2 pattern of missing data with 5% missings. Entry (i, j) shows how many times, %, method j outperformed method i .

<i>Method</i>	Unscaled data model						Scaled data model			
	IMLS-1	IMLS-4	N-IMLS	INI	N-Mean	ILS	IMLS-1	IMLS-4	INI	N-Mean
ILS	60	80	70	60	80	-	100	70	0	10
GZ	60	80	60	60	80	60	100	70	0	10
NIPALS	80	40	60	50	50	10	80	30	0	10
IMLS-1	-	40	60	50	50	0	-	30	0	10
IMLS-4	60	-	40	40	80	30	70	-	10	20
Mean	90	90	100	100	90	100	100	100	100	100
N-ILS	70	-	90	100	40	70	-	90	100	40
N-IMLS	40	60	-	20	60	100	100	90	90	100
INI	50	60	80	-	60	100	100	90	-	100
N-Mean	50	20	40	40	-	90	90	80	0	-

bly comes from the phenomenon mentioned above: NN based least-squares imputation methods may fail to converge at the unscaled data model because small fragments of data it produces are almost unidimensional thus unstable.

5.5 Performance

As was mentioned already, the data table sizes in our experiments have been about 200 to 250 times 15 to 25. With this type of data, the ordinary global least-squares methods such as ILS require about half a second to run on the platform MatLab-6 installed at a Pentium III 733 MHz. The nearest neighbour versions of least squares imputation are about 30 times slower than that. ILS-GZ appears to be the slowest of the ten methods.

5.6 Summary of the Experimental Results

Our experiments with the ten imputation methods over two Gaussian mixture data models and four patterns of missing data (Random RM, Restricted random RRM, Merged database with missings from one database MD-1 and from two databases MD-2) can be summarized as follows:

- (1) The scaled NetLab Gaussian data model leads to smaller errors of least squares imputation techniques at all missing patterns considered. At some patterns of missing data the scaled model leads to different results.
- (2) The three NN-based least squares techniques are obvious winners at the first three patterns of missing data under each of the two data models. Global least squares methods win only at the Merged database with missings from two databases under the scaled NetLab Gaussian mixture data model. Moreover, in this case the unidimensional versions are the best at the level of 5% missings.

- (3) Method IMLS and its NN versions have better convergence properties than ILS and its NN versions.
- (4) Our global-local method INI outperforms the others under the original NetLab data model, and N-IMLS frequently wins under the scaled NetLab data model.
- (5) N-Means joins in the winning methods when there are few Random missings and at the Merged database with missings from two databases Md-2.

6 Conclusion

We described a number of least-squares data imputation techniques. These methods extend the one-by-one extraction strategy of the principal component analysis to the case of incomplete data and combine it with the nearest neighbour approach as proposed in [37]. A representative set of eight least-squares based methods have been tested at simulated data to compare their performances. The well-known average scoring method Mean and its nearest-neighbour version, N-Mean, recently described in the literature, have been used as the bottom-line.

We also proposed a number of mechanisms for missing entries. In contrast to conventional definitions based on data values [25], our patterns are formulated in terms of columns and rows of the data matrix. These are: Random missings RM, Restricted random missings RRM and Merged database MD-1 and MD-2 types.

We carried out experiments with data generated according to two NetLab based versions of the mixture of Gaussian distributions, one giving a blot of overlapping distributions, the other well separated clusters.

It appears, the global-local two-stage NN based method INI overwhelmingly outperforms the others under the original NetLab data model while N-IMLS takes the lead under the scaled NetLab data model with INI trailing behind very closely. Some exceptions to this rule, mostly related to the mechanism MD-2 of massive block-wise missing, have been described above. This allows us to recommend the two NN based methods, INI and N-IMLS, for imputing missing data in most situations.

With regard to the issues raised, directions for future work should include the following:

- (1) More flexible versions of the least squares imputation techniques should be developed.
- (2) A theoretical investigation should be carried out on the properties of convergence for both major iterative techniques, ILS and IMLS. In our computations, ILS may fail to converge, even when the Gabriel-Zamir setting is applied to initialise the process. We also failed to see how the majorization principle [15,22] can be applied in the IMLS framework.
- (3) The performances of the least-squares based techniques should be compared with

those of another set of popular imputation techniques based on the maximum likelihood principle. This requires rearranging the setting of experiments since the latter methods are computationally expensive and may fail to converge. In our preliminary experiments, though, the errors were similar between the global least-squares imputation techniques and standard expectation-maximisation techniques implementing the maximum likelihood principle. This concurs with empirical findings reported in [34].

Acknowledgements

The authors gratefully acknowledge many helpful comments by reviewers that have been very helpful in improving the presentation.

References

- [1] Aha, D., Editorial, 1997. Artificial Intelligence Review 11, 1-6.
- [2] Atkeson, C.G., Moore, A.W. and Schaal, S., 1997. Locally weighted learning, Artificial Intelligence Review 11, 11-73.
- [3] Benzecri, J.P., 1973. Analyse des Donnees, Paris, Dunod.
- [4] Berry, M., Dumais, S., Landauer, T. and O'Brien, G., 1995. Using linear algebra for intelligent information retrieval, SIAM Review 37, 573-595.
- [5] Christoffersson, A., 1970. The one component model with incomplete data, PhD Thesis, Uppsala University.
- [6] Davies, P. and Smith, P., 1999. Model quality reports in business statistics, ONS, UK.
- [7] Dempster, A.P., Laird, N.M. and Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39, 1-38.
- [8] EM based imputation software: <http://www.stat.psu.edu/jls/misoftwa.html>, <http://methcenter.psu.edu/EMCOV.html>.
- [9] Everitt, B.S. and Hand, D.J., 1981. Finite mixture distributions, Chapman and Hall, London.
- [10] Gabriel, K.R. and Zamir, S., 1979. Lower rank approximation of matrices by least squares with any choices of weights, Technometrics 21, 489-298.
- [11] Generation of Gaussian mixture distributed data, NETLAB neural network software, <http://www.ncrg.aston.ac.uk/netlab>.
- [12] Golub, G.H. and Loan, C.F., 1986. Matrix Computation, John Hopkins University Press, 2nd Edition.

- [13] Grung, B. and Manne, R., 1998. Missing values in principal component analysis, *Chemometrics and Intelligent Lab. System* 42, 125-139.
- [14] Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P. and Botstein, D., 1999. Imputing missing data for gene expression arrays, Technical Report, Division of Biostatistics, Stanford University.
- [15] Heiser, W.J., 1995. Convergent computation by iterative majorization: theory and applications in multidimensional analysis, In: W. J. Krzanowski (Ed.), *Recent advances in descriptive multivariate analysis*, Oxford Univ. Press, pp. 157-189.
- [16] Holter, N.S., Maritan, A., Cieplak, M., Fedoroff, N.V., Banavar, J.R., 2001. Dynamic modeling of gene expression data, *PNAS* 98, 1693-1698.
- [17] Holzinger, K.J. and Harman, H.H., 1941. *Factor Analysis*. University of Chicago Press, Chicago.
- [18] Hunt, L. and Jorgensen, M., 2003. Mixture model clustering for mixed data with missing information, *Computational Statistics and Data Analysis* 41, 193-210.
- [19] Jolliffe, I.T., 1986. *Principal component analysis*, Springer-Verlag, New-York.
- [20] Kamakashi, L., Harp, S.A., Samad, T., and Goldman, R.P., 1996. Imputation of missing data using machine learning techniques, In: Simoudis, E., Han, J., and Fayyad, U. (Eds.), *Second International Conference on Knowledge Discovery and Data Mining*, Oregon, pp. 140-145.
- [21] Kenney, N. and Macfarlane, A., 1999. Identifying problems with data collection at a local level: survey of NHS maternity units in England, *British Medical Journal* 319, 619-622.
- [22] Kiers, H.A.L., 1997. Weighted least squares fitting using ordinary least squares algorithms, *Psychometrika* 62, 251-266.
- [23] Krzanowski, W.J., 1988. Missing value imputation in multivariate data using the singular value decomposition of a matrix, *Biometrical Letters* 25, 31-39.
- [24] Laaksonen, S., 2000. Regression-based nearest neighbour hot decking, *Computational Statistics* 15, 65-71.
- [25] Little, R.J.A. and Rubin, D.B., 1987. *Statistical analysis with missing data*, John Wiley and Sons, New York.
- [26] Mirkin, B., 1996. *Mathematical classification and clustering*, Kluwer Academic Publishers, Dordrecht.
- [27] Myrtveit, I., Stensrud, E., and Olsson, U.H., 2001. Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods, *IEEE Transaction on Software Engineering* 27, 999-1013.
- [28] Quinlan, J.R., 1989. Unknown attribute values in induction, *Sixth International Machine Learning Workshop*, New-York.
- [29] Roweis, S., 1998. EM algorithms for PCA and SPCA, In: Jordan, M., Kearns, M. and Solla, S. (Eds), *Advances in Neural Information Processing Systems* 10, MIT Press, pp. 626-632.

- [30] Rubin, D.B., 1987. Multiple imputation for nonresponse in surveys, John Wiley & Sons, Inc., New York.
- [31] Rubin, D.B., 1996. Multiple imputation after 18+ years, *Journal of the American Statistical Association* 91, 473-489.
- [32] Schafer, J.L., 1997. Analysis of incomplete multivariate data, Chapman and Hall, London.
- [33] Shum, H.Y., Ikeuchi, K. and Reddy, R., 1995. PCA with missing data and its application to polyhedral object modelling, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17, 854-867.
- [34] Strauss, R.E., Atanassov, M.N. and De Oliveira, J.A., 2003. Evaluation of the principal-component and expectation-maximization methods for estimating missing data in morphometric studies, *Journal of Vertebrate Paleontology* 23, 284-296.
- [35] Tipping, M.E. and Bishop, C.M., 1999. Probabilistic principal component analysis, *Journal of the Royal Statistical Society Series B* 61, 611-622.
- [36] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Hastie, R., Tibshirani, R., Botstein, D., and Altman, R.B., 2001. Missing value estimation methods for DNA microarrays, *Bioinformatics* 17, 520-525.
- [37] Wasito, I. and Mirkin, B., 2005. Nearest neighbour approach in the least-squares data imputation algorithms, *Information Sciences*, 169, n. 1.
- [38] Wold, H., 1966. Estimation of principal components and related models by iterative least square, In: Krishnaiah, P.R. (Ed.), *Multivariate Analysis, Proceedings of International Symposium in Dayton*, Academic Press, pp. 391-402.