Least Squares Algorithms with Nearest Neighbour Techniques for Imputing Missing Data Values

Ito Wasito

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy

University of London

April 2003

School of Computer Science and Information Systems
Birkbeck College

To Nining and Denny.

Table of Contents

Abstract

The subject of imputation of missing data entries has attracted considerable efforts in such areas as editing of survey data, maintenance of medical documentation and modeling of DNA microarray data.

There are several popular approaches to this of which we concentrate on the least squares approach extending the singular value decomposition (SVD) of matrices. We consider two generic least squares imputation algorithms: (a) ILS, which interpolates missing values by using only the non-missing entries for an SVD-type approximation and (b) IMLS, which recursively applies SVD to the data completed initially with ad-hoc values (zero, in our case).

We propose nearest neighbour versions of these algorithms, N-ILS and N-IMLS, as well as a combined algorithm INI that applies the nearest neighbour approach to the data initially completed with IMLS. Altogether, a set of ten least squares imputation algorithms including the method of imputing mean values as the bottom-line are considered.

An experimental study of these algorithms has been performed on artificial data generated according to Gaussian mixture data models. The data have been combined with four different mechanisms for generating missing entries: (1) Complete random pattern; (2) Inherited random pattern; (3) Sensitive issue pattern and (4) Merged databases pattern. The mechanisms (2), (3) and (4) have been introduced in this study.

Since data and missings are generated independently, the performance of an algorithm is evaluated based on the difference of the imputed values and those originally generated.

The major result of these experiments is that the nearest neighbour versions of the least squares algorithms almost always surpass the global least squares algorithms; both the mean and nearest neighbour mean imputation are always worse.

In the case of the most popular Complete random missing pattern, our global-local algorithm INI appears to outperform the other algorithms.

We also considered two different data models: (1) Rank one and (2) Sampling from a real-world data base. At the latter, INI results are comparable to and, at greater proportions of missings, surpass results of EM (expectation-maximization) and MI (multiple imputation) algorithms based on another popular approach, the maximum likelihood.

Acknowledgements

I would like to thank Professor Boris Mirkin, my principal supervisor, for his many suggestions, guidance and constant support during this research. He also taught me many useful aspects how to do experimental research properly. I am also thankful to Dr. Trevor Fenner, my second supervisor, for his valuable advice.

I should also mention very kindly persons who contributed greatly in my study: Professor George Loizou, Head of School of Computer Science and Information Systems, provided me resources to make this thesis completed; Professor Mark Levene gave me an opportunity to present my work at a research report seminar in the School of Computer Science and Information Systems, Birkbeck College; also, Dr. Igor Mandel of Axciom-Sigma Marketing Group, Rochester, New York USA, supplied me with a large scale marketing database which has been very useful to demonstrate that the proposed method can be applicable to real world missing data problems.

I have been given full technical support by friendly members of the Systems Group of School of Computer Science and Information Systems: Phil Gregg, Phil Docking, Andrew Watkins and Graham Sadler. Also I am grateful to Rachel Hamill for proof reading of my thesis.

My graduate studies in the School of Computer Science and Information Systems, Birkbeck, University of London were fully supported by Development of Undergraduate Education Project Batch II, Universitas Jenderal Soedirman, Indonesia.

Finally, I wish to thank the following: my family, my parents, DUE-UNSOED staff, administrative staff of School of Computer Science and Information Systems, Birkbeck College, Dr. Steve Counsell, Dr. Stephen Swift and Dr. Alan Tucker (for their help in my earlier life in Birkbeck College), Rajaa (my room mate), and Jaseem (for his hospitality).

London, UK I. Wasito
25 March, 2003.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Any real-world data set is prone to have a number of missing entries. There are two major approaches to dealing with missing data: (1) impute missing entries before processing and analysing the data; (2) develop such modifications of statistical/data mining techniques that can be applied to data with missing entries (handling missings within a method).

The latter approach has attracted considerable efforts in such areas of data analysis as multivariate regression, classification and pattern recognition [Dybowski, 1998, Little, 1992, Morris et al., 1998]. However, the former approach cannot be ignored at all because there are considerable applications in which missings are to be imputed before (or without) any follow-up processing.

In particular, the problem of imputation of missing data emerges in many areas such as editing of survey data [Tirri and Silander, 1998, Tjostheim et al., 1999, Laaksonen, 2001, Little and Smith, 1987, Tsai, 2000], maintenance of medical documentation [Gryzbowski, 2000, Kenney and Macfarlane, 1999], modelling of DNA microarray data [Alizadeh et al., 2000, Troyanskaya et al., 2001] and morphometric studies [Strauss et al., 2002].

In the last few decades a number of approaches have been proposed and utilised for filling in missing values. The most straightforward idea of using average values for imputation into missing entries, known as the Mean substitution [Little and Rubin, 1987], is probably the most popular approach. It has been supplemented recently with more refined versions such as hot/cold deck imputation and multidimensional techniques such as regression [Laaksonen, 2000], decision trees [Kamakashi et al., 1996, Quinlan, 1989], etc. Two other approaches, the maximum likelihood and least squares approximation, take into account all available data to fill in all missings in parallel.

In the traditional statistics framework, any data set is considered as generated from a probability distribution, which immediately leads to applying the maximum likelihood approach for modelling and imputation of incomplete data. This approach has led to the introduction of the so called expectation-maximization (EM) method for handling incomplete data [Dempster et al., 1977, Little and Schluchter, 1985, Schafer, 1997a]. The EM algorithm provides a good framework both in theory and in practice. Another method within this approach, multiple imputation (MI), has also proven to be a powerful tool [Rubin, 1987, 1996, Schafer, 1997a]. However, the methods within this approach have two features that may become of issue in some situations. First, they may involve unsubstantiated hypotheses of the underlying distribution. Second, sometimes the rate of convergence of EM can be very slow. Furthermore, the computational cost of the method heavily depends on the absolute number of missing entries, and this can prevent its scalability to large databases.

Another multidimensional approach to imputation of missing data, the so-called least squares approximation, extends the well-known matrix singular value decomposition (SVD) and, therefore, relies on the geometric structure of the data rather than on probabilistic properties. This approach is computationally effective and has attracted the attention of a considerable number of researchers [Gabriel and Zamir, 1979, Kiers, 1997, Mirkin, 1996]. However, this approach is not sensitive to the shape of the underlying distribution, which can become an issue in imputing missing data from a complex distribution.

A computationally viable approach to overcome this drawback of the least squares imputation is to combine it with the nearest neighbours methodology which is widely used in the machine learning research. A combined method would treat the problem of imputation as a machine learning problem: for each of the missing entries only the entity's neighbours are utilised to predict and impute it. Such an NN based upgrade of the algorithm Mean has been suggested recently and showed good performances in imputing gene expression data [Hastie et al., 1999, Troyanskaya et al., 2001].

However, developing NN based versions for the least squares imputation methods is only a part of the problem. Another problem immediately emerges: how to prove that modified methods outperform the original ones? There is no generally recognised technology for experimental comparisons: existing literature is scarce and confined with very limited experiments involving mainly just a few real data sets [Hastie et al., 1999, Myrtveit et al., 2001, Troyanskaya et al., 2001]. Thus one needs to develop a strategy for computationally testing different data imputation methods. Such a strategy should involve independent generation of data sets and patterns of missing entries so that the quality of imputation can be evaluated by comparing imputed values with those generated originally.

Now one encounters further problems: what mechanisms of data generation should be utilised? What data models of missings should be considered? These questions have never been answered in computational data imputation. There is a generally accepted view in the machine learning community with regard to data generation: the data should be generated from a mixture of Gaussian data structures. Still, it is not clear how much this type of distribution covers the set of potential distributions and, moreover, what is its relevance to the real data. As for the latter item, models of missings have been treated in by far too general terms, and moreover in a somewhat biased way, by referring to survey practices only, with no references to other experimental settings or databases [Hastie et al., 1999, Myrtveit et al., 2001, Troyanskaya et al., 2001].

The goal of this project thesis is to advance in addressing the issues listed and those related, though unlisted.

1.2 Objectives

1.2.1 Least Squares Data Imputation Combined with the Nearest Neighbour Framework

This work experimentally explores the computational properties of the least squares approach and combines it with a machine learning approach, the so-called nearest neighbour (NN) method, which should balance the insensitivity of the least squares to the data structure as mentioned above. Indeed, the NN approach suggests that, to impute a missing entry, only information of the nearest neighbours should be utilised, leaving other observations aside. This approach was recently successfully applied in the context of bioinformatics to the Mean substitution method at various level of missing [Hastie et al., 1999, Troyanskaya et al., 2001]. We would to extend this to the core least squares methods.

However, the value of the NN based approach, has so far only been demonstrated on specific real data sets, namely DNA microarray data, which have completely random missing pattern. Thus, more comprehensive frameworks of the experimental investigation of the missing data problem need to be developed which will be the objective of this work.

1.2.2 The Development of Experimental Setting

The technology of data generation is quite well developed (see for instance [Everrit and Hand, 1981, Roweis, 1998, Tipping and Bishop, 1999a, Nabney, 1999, 2002]). This is not so for the generation of missing patterns. The only concept considered so far is that of randomly distributed missing entries. All of the causes of missing data considered in the literature fit into three classes, which are based on the relationship between the missing data mechanism and the missing and observed values [Little and Rubin, 1987]:

1. Missing Completely at Random (MCAR).

   MCAR means that the missing data mechanism is unrelated to the values of any variables, whether missing or observed. Unfortunately, most missing data are not always MCAR.

2. Missing at Random (MAR).

   This class requires that the cause of the missing data be unrelated to the missing values, but may be related to the observed values of other variables. Thus, MAR means that the missing values are related to either observed covariates or response variables.

3. Non-Ignorable (NI).

   NI means that the missing data mechanism is related to the missing values. It commonly occurs when people do not want to reveal something very personal or unpopular about themselves. For example, if individuals with higher incomes are less likely to reveal their income in a survey than are individuals with lower incomes, the missing data mechanism for income is non-ignorable. If proportionally more low and moderate income individuals are left in the sample because high income people are missing, an estimate of the mean income will be lower than the actual population mean.

However, in a real world problem, the unobservable entry could occur under specific circumstances which cannot be explained according to above mechanisms. Here follows some examples.

1. In situation of experiment where the process of data collection is organized within time series process, for instance some of the missing entries can be further investigated or measured to be collected and imputed as part of raw data.

2. There is a set of questions related to an issue which is sensitive for a random group of respondents. These respondents tend to leave the sensitive questions with no answer, this way generating incomplete data in a survey.

3. The data set under consideration may have been obtained by merging two or more databases of the same type of records. This is frequent in medical

informatics. It may happen that records in either of the original databases lack some features that have been recorded for the other data base. This way, any part of the data may miss a submatrix of entries corresponding to the records of a corresponding database and the features that have been missed in it.

In our simulation study, the above scenarios of missing entries will be taken into account. However, the non-ignorable errors remain beyond the scope of this research project.

1.2.3 The Experimental Comparison of Various Least Squares Data Imputation Algorithms

In order to examine whether the NN versions of least squares data imputation methods always surpass the global least squares approaches for imputing missing entries, a simulation study based on processing generated data within the developed frameworks need to be performed.

To do this, a complete data set and a set of related missing patterns are generated separately. Then, for every data set and missing pattern, the imputed values can be compared with those originally generated; the smaller the difference, the better the method. The well-known average scoring method Mean and its NN version, N-Mean [Hastie et al., 1999, Myrtveit et al., 2001, Troyanskaya et al., 2001], will be used as the bottom-line.

Main attention will be given to the commonly used Gaussian mixture data generation mechanism. However, some other data structures should be utilised as well, to see how much the experimental results depend on the data structure.

1.3 The Structure of the Thesis

This thesis will be organized as follows. Chapter 2 provides a review of existing techniques for handling missing data by categorizing them in three groups: (a) prediction rule based, (b) the maximum likelihood based and (c) the least squares approximation based ones. Chapter 3 gives a brief description of two global least squares imputation methods that can be considered as standing behind various algorithms published method. The nearest neighbour versions of least squares imputation methods including combined global-local framework will be proposed in Chapter 4. The setting and results of the experimental study of least squares and their nearest neighbour versions will be described in Chapter 5. The experiments with different data generation mechanism are considered in Chapter 6. Chapter 7 concludes the thesis and describes directions for future research.

Chapter 2

A Review of Imputation Techniques

This chapter overviews the techniques of imputation of incomplete data which could be categorized in the following three approaches:

1. Prediction rules [Buck, 1960, Laaksonen, 2000, Little and Rubin, 1987, Mesa et al., 2000, Quinlan, 1989, Tsai, 2000];

2. Maximum likelihood [Dempster et al., 1977, Liu and Rubin, 1994, Little and Rubin, 1987, Rubin, 1996, Schafer, 1997a];

3. Least squares approximation [Gabriel and Zamir, 1979, Grung and Manne, 1998, Kiers, 1997, Mirkin, 1996, Shum et al., 1995, Wold, 1966].

Here follows a review of the techniques for each category.

2.1 Prediction Rules

The common feature of the prediction rule based approaches is that they rely on a limited number of variables. The other two approaches take advantage of using the entire available data entries to handle missings. Depending on the characteristic of rule to be utilized, imputation techniques in this category could be differentiated into two classes:

1. Simple rule prediction.

   This approach uses a relationship rule to fill in the missing value for each entity within one variable. The most popular approaches are Mean and Hot/Cold deck imputation. The imputation methods within this class are characterized by the conceptual simplicity, easy to implement and computationally simple.

2. Multivariate rule prediction.

   Basically, this is an extension of the simple rule prediction which utilizes more than one variable to impute the missing entries. The regression, tree-based and neural network are examples of imputation techniques using multivariate rule prediction.

Brief description of each imputation techniques can be summarized as follows.

2.1.1 Mean Imputation

The most popular method of imputation is substitution of a missing entry by the corresponding variable's mean, which will be referred to as the Mean algorithm. The popularity of Mean imputation is probably caused by its simplicity. However, an important drawback of the mean imputation is that the variance of the imputed data systematically underestimates the real variance which can be described as follows [Little and Rubin, 1987]. Suppose the missing values x_{ik} are substituted by their mean of the observed values, $\bar{x_k}$. The variance of the "completed data" is:

$$\hat{\sigma}^2 = \frac{(n_k - 1)S_k}{(n - 1)} \tag{2.1.1}$$

where n_k, n and S_k denote number of observed entities, number of overall entities and the estimated variance from the observed values, respectively. Thus, the variance of "completed data" underestimates the 'true' variance by a factor of $(n_k - 1)/(n - 1)$. As a result, the ordinary statistical analysis will give biased results on the "completed data".

Recently, a nearest-neighbour based modification of Mean imputation techniques has been proposed (see [Hastie et al., 1999, Myrtveit et al., 2001, Troyanskaya et al., 2001]).

2.1.2 Hot/Cold Deck Imputation

The Mean algorithm has been recently extended with other prediction models such as hot deck imputation in which the nearest neighbour's value is imputed [Laaksonen, 2000, Little and Rubin, 1987]. This approach has the following advantages in: (1) Conceptual and technical simplicity; (2) Sustenance of the proper measurement level of variables such that categorical variables remain categorical and continuous variables remain continuous; (3) The result of imputed data that can be analyzed like any complete data matrix. However, this approach might heavily depends on the criterion to select the neighbour. If the missing value is substituted by the modal value rather than by that from its most similar entity, it is referred to as cold deck imputation.

2.1.3 Regression Imputation

The improvement of Mean algorithm was realized in regression imputation, also well-known as conditional Mean imputation, introduced by Buck [Buck, 1960]. The method proceeds as follows. First, compute the estimates of mean vector and covariance matrix of data, μ and Σ, based on sub-matrix which consists of entities with no missing variables. If $\bar{x}$ and S denote the estimates of mean vector and covariance matrix, respectively then calculate the linear regression of the missing variables on the observed variables, entity by entity, using those $\bar{x}$ and S. Finally, fill in the missing values by the predicted-values, i.e, those found in linear regression computation. This approach still underestimates the variance and covariance, although the factor of underestimation is less than when the Mean imputation is used [Laaksonen, 2000, Little and Rubin, 1987].

More recently, the combination of the multivariate regression model and a nearest neighbour hot decking method, which is referred to as regression-based nearest neighbour hot decking (RBNNHT) algorithm, was introduced in [Laaksonen, 2000]. This approach consists of the following steps:

1. Input the data set, $\mathbf{X}$.
2. Form a multivariate regression model so that $\mathbf{Y}$ is the dependent variable which is to be imputed, and the variables without missing entries are chosen as the independent variables.
3. Compute the predicted values $\hat{\mathbf{Y}}$ of $\mathbf{Y}$.
4. Order the data set by the predicted values $\hat{\mathbf{Y}}$.
5. For each of the missing entries of $\mathbf{Y}$ input that observed value of $\mathbf{Y}$ which is the closest to it in the order.

As described in [Laaksonen, 2000], unlike the pure regression imputation method, RBNNHT method does not cause to underestimate the variance. However, this approach may have a problem due to the poor balance of the proportion between missing and observed entries within specified range values in a data set.

2.1.4 Tree Based Imputation

Basically, there are two types of tree-based models which could be characterized in terms of the scale of measurement of the response variable [Breiman et al., 1984, Mesa et al., 2000, Kamakashi et al., 1996, Quinlan, 1989]:

1. Classification Tree Model

   In this tree model, the response variable is assumed to be categorical. The measures of homogeneity to determsine the splits of the tree can be accomplished according to: (a) F-test; (b) Chi-squared test; (c) Likelihood ratio test; (d) Gini index; (e) Twoing rule.

2. Regression Tree Model

   The type of response variable under this tree model is numerical. Two splitting rules are frequently used: (a) Least squares and (b) Least absolute deviations.

There are two well-known methods to construct the tree: (1) CHAID (Chi-squared Automatic Interaction Detector) which builds non-binary trees and (2) CART (Classification and Regression Trees) which constructs binary trees only [Tsai, 2000, Steinberg and Colla, 1995].

Handling missing values using tree-based methods is very straightforward. First, take a response variable and the independent variables without missing entries. Then build a classification/regression tree which represents the distribution of the response variable in terms of the values of independent variables. Fill in the missing values in the response variable using an appropriate approach, for instance Mean imputation, based on the "complete data" entities in the same node [Mesa et al., 2000].

2.1.5 Neural Network Imputation

Recently, more sophisticated version of the regression based imputation techniques using neural network based approaches were implemented in [Nordbotten, 1996]. This method uses the feed-forward neural network with a single layer of hidden units. The generic imputation model can be formulated as follows:

$$y_i = f(b_i + \sum_{j=1}^{N_h} d_{ij} * f(a_k + \sum_{k=1}^{N_x} c_{jk} * x_k)) \quad i = 1, \ldots, N_y \tag{2.1.2}$$

where x_k, y_i, N_x, N_y and N_h denote the independent variables, dependent variables, number of independent variables, number of dependent variables, and number of units in the hidden layer, respectively. The parameters a, b, c and d are estimated using the following sigmoid function f:

$$f(t) = 1/(1 + e^{-t}) \tag{2.1.3}$$

In order to train the above model, the back-propagation (BP) approach is utilized on the variables without missings. The training is evaluated according to the mean square error (MSE) of the differences between the predicted-values of y and observed values. Then, the optimal weights of BP can be determined at which the MSE converges within a pre-specified threshold value. This method has been successfully applied in the data editing application (see for instance in [Nordbotten, 1995, 1996]).

In another development, the imputation method using recurrent neural network

Figure 2.1: The architecture of recurrent networks with 90-3-4 architecture data with missing values [Bengio and Gingras, 1996]

is proposed in [Bengio and Gingras, 1996]. Under this approach, the missing values in the input variables are initialized to their unconditional means, then their values are updated within the architectures of the recurrent networks as shown in figure 2.1.5.

The advantage of the use of neural network imputation is that the generic imputation model can be regarded as a set of non-parametric, non-linear multivariate regression. However, this approach is computational expensive. Furthermore, it is not always easy to determine the goodness of the training model.

2.2 Maximum Likelihood

The maximum likelihood approach is very popular since it is based on a precise statistical model. This approach relies on a parametric model of data generation, typically, multivariate Gaussian mixture model. Then a maximum likelihood method is applied for both fitting the model and imputation of the missing data. However, methods within this approach may involve insubstantiated hypotheses and have a slow rate of convergence. Either of these may prevent their scalability to large databases. According to the number of imputations needed to fill in missing entries, there are two broad categories of approaches which are referred to as single imputation and multiple imputation. In the former category, to fill in missings, the imputation is accomplished once only, for instance in the EM algorithm [Dempster et al., 1977, Little and Rubin, 1987, Liu and Rubin, 1994, Schafer, 1997a]. On the other hand, in the multiple imputation (MI), the missing entries are imputed more than once, usually 3-10 times, see for instance Multiple Imputation (MI) algorithm [Rubin, 1987, 1996, Schafer, 1997a, Schafer and Olsen, 1998]. Further details of each approach will be described in the following subsections.

2.2.1 EM Algorithm

Maximum-likelihood estimates can often be calculated directly from the incomplete data by specialized numerical methods such as the expectation-maximization (EM) algorithm which was introduced in [Dempster et al., 1977]. Further development of the implementation of EM algorithm for handling missing data was explored in [Little and Rubin, 1987, Schafer, 1997a]. Indeed, the EM algorithm is derived from the old-fashioned idea of handling missing values through iterative steps:

1. Impute the missings values using ad-hoc values.
2. Estimate the parameters of distribution.
3. Re-impute the missing values using the parameters from step 2.
4. Repeat steps 2 and 3 until the iteration converges for pre-specified threshold values.

Formally, the EM algorithm can be illustrated mathematically as follows: suppose the variables and the current estimate of parameter denoted by X and $\theta(t)$ respectively, then the completed-data likelihood, which is composed from missing and observed values, is written as $\ell(\theta|X)$. The E-step of t-th iteration of EM algorithm can be computed as: $\mathcal{Q}(\theta|\theta^t) = \int \ell(\theta|X)f(X_{mis}|X_{obs}, \theta = \theta^t)\mathrm{d}X_{mis}$ where X_{mis}, X_{obs} and f denote the missing values, observed values and probability density function respectively. The $f(..)$ usually represents multivariate normal distribution. Then θ^{t+1} is chosen as the value of θ which maximize $\mathcal{Q}$. A brief description of EM algorithm for multivariate incomplete data will be given further. This algorithm has been implemented in [Schafer, 1997a, Strauss et al., 2002].

2.2.1.1 EM Method for Imputation of Incomplete Multivariate Normal Data

To find maximum likelihood estimates in close form when portions of the data matrix $\mathbf{X}$ are missing, the EM algorithm for a multivariate normal data matrices with an arbitrary pattern of missing is carried out in two steps: E-step and M-step [Schafer, 1997a]. Further details of E and M computation will be given in next section. However, firstly, some useful computation detail for EM approach will be introduced.

Sweep Operator

The importance of the sweep operator in the maximum likelihood computation for incomplete multivariate normal data with general pattern of missingness is demonstrated in [Little and Rubin, 1987, Schafer, 1997a]. In literature, this procedure is used for linear model approximation, stepwise regression and orthogonalization procedure [Schafer, 1997a].

Suppose that $\mathbf{G}$ is a $n \times n$ symmetric matrix with elements g_{ik}. Then sweep operator, $SWP[k]$, operates on $\mathbf{G}$ by replacing it with $n \times n$ symmetric matrix $\mathbf{H}$, this operation formally can be formulated as:

$$H = SWP[k]G \tag{2.2.1}$$

with the elements defined as follows:

$h_{kk} = -1/g_{kk}$

$h_{jk} = h_{kj} = g_{jk}/g_{kk}; \quad for \ \ j \neq k$

$h_{jl} = h_{lj} = g_{jl} - g_{jk}g_{kl}/g_{kk}; \quad for \ \ j \neq k, \ \ l \neq k$

Practically, the “sweeping” operation is accomplished in the following steps:

1. Replacing g_{kk} by $h_{kk} = -1/g_{kk}$.
2. Replacing the remaining elements g_{jk} and g_{kj} in row and column k by $h_{jk} = h_{kj} = -g_{jk}h_{kk}$.
3. Replacing elements g_{jl} that beyond row k or column k by $h_{jl} = g_{jl} - h_{jk}g_{kl}$.

If we define $SWP[k_1, k_2, \ldots, k_t] = SWP[k_1]SWP[k_2]\ldots SWP[k_t]$, it is not difficult to show that $SWP[j,k] = SWP[k,j]$ and the reverse-sweep operator (RSW), denoted by $\mathbf{H} = RSW[k]G$, returns a swept matrix to its original matrix, for instance $RSW[k]SWP[k]G = G$.

Patterns of Missing

Before performing E-step and M-step computation, first, a procedure to predict missing entries in each column $\mathbf{X}_k$ will be introduced. Suppose the are patterns of missing in $\mathbf{X}$ and let $\mathbf{M}$ be $N \times n$ matrix of binary indicators whose elements are defined as:

Patterns	Variables			
	$\mathbf{X}_1$	$\mathbf{X}_2$	$\dots$	$\mathbf{X}_n$
1	1	1	$\dots$	1
2	1	1	$\dots$	0
.	1	0	$\dots$	1
.	1	0	$\dots$	0
.	0	1	$\dots$	1
.	0	1	$\dots$	0
.	0	0	$\dots$	1
N	0	0	$\dots$	0

Table 2.1: An example of a patterns of matrix **M**

$$m_{ik} = \begin{cases} 1 & \text{if } X_k \text{ is observed in row } i. \\ 0 & \text{if } X_k \text{ is missing in pattern in row } i. \end{cases} \tag{2.2.2}$$

Table 2.1 shows that for each missingness pattern, the variables $\{\mathbf{X}_1, \mathbf{X}_2, \dots \mathbf{X}_n\}$ consist of subsets which point to observed and missing values, denoted as $\mathbf{O}bs(i)$ and $\mathbf{M}is(i)$ respectively, which are defined as follows:

$$\mathbf{O}bs(i) = \{k : m_{ik} = 1\}$$
$$\mathbf{M}is(i) = \{k : m_{ik} = 0\}$$

For $i = 1, 2, \dots, N$.

E-step

There are well-known results of the maximum likelihood estimates for parameters of multivariate normal distribution $\theta = \{\mu, \mathbf{\Sigma}\}$ which consist of the sample mean vector:

$$\bar{x} = (1/N) \sum_{i=1}^{N} x_i, \tag{2.2.3}$$

and the sample covariance matrix:

$$S = (1/N)\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})' \tag{2.2.4}$$

respectively. Both values also well known as sufficient statistics of μ and $\mathbf{\Sigma}$ which are derived from data sample.

Unfortunately, when there are missing entries in data matrix, the traditional statistical approaches to compute maximum likelihood estimates can not be utilized. Based on this rationale, the Expectation step as part of EM algorithm, referred to as E-step, will be applied. This step is accomplished as follows [Little and Rubin, 1987, Schafer, 1997a].

Suppose $\mathbf{X}_{mis}$ and $\mathbf{X}_{obs}$ are the missing and observed entries of the matrix, respectively. Thus, the E-step is implemented as calculates the expectation of the complete-data sufficient statistics, in terms of $\sum_i x_{ik}$ and $\sum_i x_{ik}x_{ij}, j \neq k$, over $P(\mathbf{X}_{mis}|\mathbf{X}_{obs}, \theta)$ for assumed value of θ. By assuming the rows $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ of $\mathbf{X}$ independent given θ, their probability can be formulated as:

$$P(\mathbf{X}_{mis}|\mathbf{X}_{obs}, \theta) = \Pi_{i=1}^{n} P(\mathbf{x}_{i(mis)}|\mathbf{x}_{i(obs)}, \theta) \tag{2.2.5}$$

where $\mathbf{x}_{i(obs)}$ and $\mathbf{x}_{i(mis)}$ denote the observed and missing subvectors of $\mathbf{x}_i$, respectively [Schafer, 1997a].

Furthermore, $\mathbf{x}_{i(mis)}$ can be computed from a multivariate normal linear regression altogether with their parameters by sweeping the matrix θ on the positions corresponding to the observed variables in $\mathbf{x}_{i(obs)}$. As a result, the location of parameters of $P(\mathbf{x}_{i(mis)}|\mathbf{x}_{i(obs)}, \theta)$ is in the rows and columns labeled $\mathbf{M}is(i)$ of the matrix $\mathbf{Z}$ which is defined as:

$$\mathbf{Z} = \mathbf{S}WP[\mathbf{O}bs(i)]\theta \tag{2.2.6}$$

This swept parameter matrix is operated on the row i which is in the missingness pattern s from Table 2.1. Suppose that the (k, l)-th element of $\mathbf{Z}$ is denoted as z_{kl}, $(k, l = 0, 1, \dots n)$, then having made simple manipulation, E-step gives [Schafer, 1997a]:

$$E(x_{ik}|X_{Obs}, \theta) = \begin{cases} x_{ik} & \text{for } k \in Obs(i) \\ x^*_{ik} & \text{for } k \in Mis(i) \end{cases} \tag{2.2.7}$$

$(E(x_{ik}x_{il}|X_{Obs}, \theta)) =$

$$\begin{cases} x_{ik}x_{il} & \text{for } k, l \in Obs(i) \\ x^*_{ik}x_{il} & \text{for } k \in Mis(i), l \in Obs(i) \\ z_{kl} + x^*_{ik}x^*_{il} & \text{for } k, l \in Mis(i) \end{cases} \tag{2.2.8}$$

where

$$x^*_{ik} = z_{0k} + \sum_{l \in Obs(i)} z_{lk}x_{ik} \tag{2.2.9}$$

In another formulation, the E-step can be written as $E(\mathbf{U}|X_{obs}, \theta)$, where $\mathbf{U}$ is the matrix of the second-order moments, a complete-data sufficient statistics:

$$\mathbf{U} = \sum_{i=1}^{N} \begin{bmatrix} N & x_{i1} & x_{i2} \dots & x_{in} \\ & x_{i1}^2 & x_{i1}x_{i2} \dots & x_{i1}x_{in} \\ & & x_{i2}^2 \dots & x_{i2}x_{in} \\ & & \ddots & \vdots \\ & & & x_{in}^2 \end{bmatrix} \tag{2.2.10}$$

M-step

Given a complete-data log likelihood from E-step, M-step finds the parameter estimates to maximize the complete-data log likelihood as:

$$\widehat{\theta} = SWP[0]N^{-1}E(\mathbf{U}|\mathbf{X}_{obs}, \theta) \tag{2.2.11}$$

The formal approach of EM algorithm can be summarized as follows.

EM Imputation Algorithm

1. *Impute the missings values using ad-hoc values.*
2. *E-Step: Compute the conditional expectation of complete-data log likelihood, U, which is operated as $E(\mathbf{U}|X_{obs}, \theta)$.*
3. *M-Step: Given complete-data log likelihood from step 2, calculate the parameter estimates $\widehat{\theta}$ from (2.2.11).*
4. *Set $\theta = \widehat{\theta}$, then repeat steps 2 and 3 until the iteration converges for pre-specified threshold value.*
5. *Impute missing values using an appropriate approach based on the found parameters from step 4.*

EM with Different Mechanisms

There are two popular approaches to fill in missing values as shown in step 5 of EM imputation algorithm. In the first approach, the missings are imputed with random values generated from parameters those to be found in the EM computation. This approach is implemented in "Norm" software developed by Schafer which is freely available in [Schafer, 1997b]. Indeed, this approach mainly to be implemented within multiple imputation method. In this framework, the missings are imputed more than once using specific simulation. Then, several imputed data sets are analyzed using ordinary statistical techniques (see for instance [Rubin, 1987, 1996, Schafer, 1997a]).

In either approach, the imputation of missing entries are accomplished under multiple regression scheme using parameters those to be found in the EM computation. This technique demonstrated by Strauss in [Strauss et al., 2002].

2.2.2 Multiple Imputation with Markov Chain Monte-Carlo

Multiple imputation method was first implemented in an editing of data survey to create widely public-use data sets to be shared by many end-users. Under this framework, the imputation of missing values is carried out more than once, typically 3-10 times, in order to provide valid inferences from imputed values. Thus, MI method is designed mainly for statistical analysis purposes and much attention has been paid to it in the statistical literature. As described in [Rubin, 1996, Horton and Lipsitz, 2001], MI method consists of the following three-step process:

1. Imputation: Generate m sets of reasonable values for missing entries. Each of these sets of values can be used to impute the unobserved values. Thus, there are m "completed" data sets. This is the most critical step since it is designed to account for the relationships between unobserved and observed variables. Thus the MAR (Missing at Random) assumption is the central issue to the validity of the of multiple imputation approach. There are a number of imputation models that can be applied. Probably the imputation model via the Markov Chain Monte-Carlo (MCMC) is the most popular approach. This simulation approach is demonstrated within the following IP (Imputation-Parameter steps) algorithm [Schafer, 1997a]:

   I-step: Generate $\mathbf{X}^{mis,t+1}$ from $f(\mathbf{X}|\mathbf{X}^{obs}, \theta^t)$.

   P-step: Generate θ^{t+1} from $f(\theta|\mathbf{X}^{obs}, \mathbf{X}^{mis,t+1})$.

   The above steps produce Markov chain $(\{\mathbf{X}^1, \theta^1\}, \{\mathbf{X}^2, \theta^2\}, \ldots, \{\mathbf{X}^{t+1}, \theta^{t+1}\}, \ldots)$ which converge to the posterior distribution.

2. Analysis: Apply the ordinary statistical method to analyze each "completed"

data sets. From each analysis, one must first calculate and save the estimates and standard errors. Suppose that $\hat{\theta_j}$ is an estimate of a scalar quantity of interest (e.g. a regression coefficient) obtained from data set j ($j = 1, 2, \ldots, m$) and ${\sigma_{\hat{\theta,j}}}^2$ is the variance associated with $\hat{\theta_j}$.

3. Combine the results of analysis.

   In this step, the results are combined to compute the estimates of the within imputation and between imputation variability [Rubin, 1987]. The overall estimate is the average of the individual estimates:

$$\bar{\theta} = 1/m \sum_{j=1}^{m} \theta_j \tag{2.2.12}$$

   For the overall variance, one must first calculate the within-imputation variance:

$$\bar{\sigma_\theta}^2 = 1/m \sum_{j=1}^{m} \sigma^2_{\hat{\theta,j}} \tag{2.2.13}$$

   and the between-imputation variance:

$$B = 1/(m-1) \sum_{j=1}^{m} (\hat{\theta}_j - \bar{\theta})^2 \tag{2.2.14}$$

   then the total variance is:

$$\sigma^2_{pool} = \bar{\sigma_\theta}^2 + (1 + 1/m)B \tag{2.2.15}$$

Thus, the overall standard error is the square root of σ^2_{pool}. Confidence intervals are found as: $\bar{\theta} \pm \sigma_{pool}$ with degrees of freedom:

$$df = (m-1)(1 + \frac{m\bar{\theta}}{(m+1)B}) \qquad (2.2.16)$$

This method is powerful since the uncertainty of the imputation is taken into account [Rubin, 1987, 1996, Schafer, 1997a, Schafer and Olsen, 1998]. However, as a computational tool MCMC based approach has drawbacks: (1) Complicated and computationally expensive; (2) Unclear convergence of computation; (3) Multivariate normal distribution assumption requirement.

Obviously, if the predictive accuracy of imputed values is the only main criterion for choosing existing imputation technique, then MI seems to be an inefficient technique compared to EM algorithm. MI has been implemented in a program called as NORM written by Schafer which is freely available on his website [Schafer, 1997b]. In the context of data imputation, in our view, MI can be applied to estime missing data as average, estimates of the multiple imputations.

2.2.3 Full Information Maximum Likelihood

The full information maximum likelihood (FIML) is a model-based imputation algorithm which is implemented as part of a fitted statistical model. This method utilize the observed values in data to construct mean vector and covariance matrix. Indeed, FIML method is implemented based on assumption that the data come from multivariate normal distribution. The FIML method can be presented as follows [Little and Rubin, 1987, Myrtveit et al., 2001]:

1. Suppose X_{ik}, $i = 1, \ldots, N$, $k = 1, \ldots, n$ is a data matrix which has a multivariate normal distribution with mean vector, μ, and covariance matrix, $\mathbf{\Sigma}$.

2. For each entity i, remove parts of the mean vector and covariance matrix of variables corresponding to missing values. Set the corresponding mean and covariance matrix as μ_i and $\mathbf{\Sigma}_i$.

3. Define the log likelihood of entity i as:
   $\log l_i = C_i - 1/2 * log|\mathbf{\Sigma}_i| - 1/2 * (x_{i.} - \mu_i)'\mathbf{\Sigma}_i^{-1}(x_{i.} - \mu_i)$ where C_i is a contant.

4. The overall log-likelihood of data matrix can be calculated as: $\log L = \sum_{i=1}^{N} \log l_i$.

5. Given that $\log L$ is a function of parameters $\theta = (\mu, \mathbf{\Sigma})$, then maximum likelihood estimates θ are computed through the first-order optimality conditions: $grad(\log L(\theta)) = 0$.

As described in the above procedure, the FIML method produces a mean vector and covariance matrix which can be utilized for further analysis.

FIML has advantage of easy of use and well-defined statistical properties. On the other hand, a disadvantage of this approach is that it requires large data set.

2.3 Least Squares Approximation

This is a nonparametric approach based on approximation of the available data with a low-rank bilinear model akin to the singular value decomposition (SVD) of a data matrix.

Methods within this approach, typically, work sequentially by producing one factor at a time to minimize the sum of squared differences between the available data entries and those reconstructed via bilinear modelling. The rate of convergence of the least squares approximation is very fast and it might suggest scalability to large databases. There are two ways to implement this approach which are described as follows.

2.3.1 Non-missing Data Model Approximation

Under this approach, an approximate data model is found using nonmissing data only and then missing values are interpolated using values found with the model. Formerly, this approach was developed for the purpose of handling the principal component analysis (PCA) with missings introduced in [Wold, 1966]. In [Wold, 1966] the unidimensional subspace was utilized to find an approximate data model with a rather complex procedure of two-way regression analysis, the so-called criss-cross regression. However, in many cases, this approach incurs a significant error of approximation. Independently, [Gabriel and Zamir, 1979] and [Mirkin, 1996] described a similar approaches in which the data is approximated by a bilinear model that assumes a subspace of higher than one dimensionality. Similar developments within chemometrics and object modelling applications were explored in [Grung and Manne, 1998] and [Shum et al., 1995], respectively.

2.3.2 Completed Data Model Approximation

Unlike the previous approach, the methods within this framework are initialized by filling in all the missing values using ad-hoc values, then iteratively approximating the completed data and updating the imputed values with those implied by the approximation. Basically, this technique has been described differently in [Grung and Manne, 1998] and [Kiers, 1997]. The former built on the criss-cross regression by Wold [Wold, 1966] while the latter on the so-called majorization method by Heiser [Heiser, 1995]. The rate of convergence of the methods within this approach is slower than those of the non-missing data model approximation. However, it converges in many situations in which the non-missing approximation fails (see further page 34).

Chapter 3

Two Global Least Squares Imputation Techniques

This chapter describes generic methods within each of the two least squares approximation approaches referred to in the previous chapter: (1) The iterative least squares algorithm [Gabriel and Zamir, 1979, Grung and Manne, 1998, Mirkin, 1996, Shum et al., 1995], (2) The iterative majorization least squares algorithm [Grung and Manne, 1998, Kiers, 1997].

3.1 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 a matrix $\mathbf{X}$ are denoted by x_{ik} ($i = 1, ..., N$, $k = 1, ..., n$). The situation in which some entries (i, k) in $\mathbf{X}$ may be missed is modeled with an additional matrix $\mathbf{M} = (m_{ik})$ where $m_{ik} = 0$ if the 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 we show the operation of matrix multiplication with symbol $*$.

3.2 Least-Squares Approximation with Iterative SVD

This section describes the concept of singular value decomposition of a matrix (SVD) as a bilinear model for factor analysis of data. This model assumes the existence of a number $p \geq 1$ of hidden factors that underlie the observed data as follows:

$$x_{ik} = \sum_{t=1}^{p} c_{tk} z_{it} + e_{ik}, \quad i = 1, \ldots N, \quad k = 1, \ldots, n. \tag{3.2.1}$$

The vectors $\mathbf{z}_t = (z_{it})$ and $\mathbf{c}_t = (c_{tk})$ are referred to as factor scores for entities $i = 1, \ldots, N$ and factor loadings for variables $k = 1, \ldots, n$, respectively [Jollife, 1986, Mirkin, 1996]. 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 $\mathbf{z}_t = (z_{it})$, we minimize the least squares criterion:

$$L_2 = \sum_{i=1}^{N} \sum_{k=1}^{n} (x_{ik} - \sum_{t=1}^{p} c_{tk} z_{it})^2 \tag{3.2.2}$$

It is proven that minimizing criterion (3.2.2) can be done with the following one-by-one strategy, which is, basically, the contents of the method of principal component analysis, one of the major data mining techniques [Jollife, 1986, Mirkin, 1996] as well as the so-called power method for SVD [Golub and Loan, 1986].

According to this strategy, computations are carried out iteratively. At each iteration t, $t = 1, ..., p$, only one factor is sought for. The criterion to be minimized at iteration t is:

$$l_2(\mathbf{c}, \mathbf{z}) = \sum_{i=1}^{N} \sum_{k=1}^{n} (x_{ik} - c_k z_i)^2 \tag{3.2.3}$$

with respect to condition $\sum_{k=1}^{n} c_k^2 = 1$. It is well-known that the solution to this problem is the singular triple $(\mu, \mathbf{z}, \mathbf{c})$ such that $\mathbf{Xc} = \mathbf{z}$ and $\mathbf{X}^T\mathbf{z} = \mu\mathbf{c}$ with $\mu = \sqrt{\sum_{i=1}^{N} z_i^2}$, the maximum singular value of $\mathbf{X}$. The found vectors $\mathbf{c}$ and $\mathbf{z}$ are stored as $\mathbf{c}_t$ and $\mathbf{z}_t$ and the next iteration $t+1$ is performed. The matrix $\mathbf{X} = (x_{ik})$ changes from iteration t to iteration $t+1$ by subtracting the found solution according to formula $x_{ik} \leftarrow x_{ik} - c_{tk} z_{ti}$.

To minimize (3.2.3), the method of alternating minimization can be utilized. This method also works iteratively. Each iteration proceeds in two steps: (1) given (c_k), find optimal (z_i); (2) given (z_i), find optimal (c_k). Finding the optimal score and loading vectors can be done according to equations:

$$z_i = \frac{\sum_{k=1}^{n} x_{ik} c_k}{\sum_{k=1}^{n} c_k^2} \tag{3.2.4}$$

and

$$c_k = \frac{\sum_{i=1}^{N} x_{ik} z_i}{\sum_{i=1}^{N} z_i^2} \tag{3.2.5}$$

that follow from the first-order optimality conditions.

This can be wrapped up as the following algorithm for finding a pre-specified number p of singular values and vectors.

Iterative SVD Algorithm

```
0. Set number of factors $p$ and specify $\epsilon > 0$, a precision threshold.
1. Set iteration number t=1.
2. Initialize $\mathbf{c}^*$ arbitrarily and normalize it. (Typically, we take $\mathbf{c}^{*\prime} = (1 \ldots, 1)$.)
3. Given $\mathbf{c}^*$, calculate $\mathbf{z}$ according to (3.2.4).
4. Given $\mathbf{z}$ from step 3, calculate $\mathbf{c}$ according to (3.2.5) and normalize it.
5. If $||c - c^*|| < \epsilon$, go to 6; otherwise put $\mathbf{c}^* = c$ and go to 3.
6. Set $\mu = ||\mathbf{z}||$, $\mathbf{z}_t = z$, and $\mathbf{c}_t = c$.
7. If $t == p$, end; otherwise, update $x_{ik} = x_{ik} - c_{tk} z_{tk}$, set $t = t + 1$ and go to step 2.
```

Note that $\mathbf{z}$ is not normalised in the version of the algorithm described, which implies that its norm converges to the singular value μ. This method always converges if the initial $\mathbf{c}$ does not belong to the subspace already taken into account in the previous singular vectors.

3.2.1 Iterative Least-Squares (ILS) Algorithm

The ILS algorithm is based on the SVD method described above. However, this time equation (3.2.1) applies only to those entries that are not missed.

The idea of the method is to find the score and loading vectors in decomposition (3.2.1) by using only those entries that are available and then use (3.2.1) for imputation of missing entries (with the residuals ignored).

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

$$l_2 = \sum_{i=1}^{N}\sum_{k=1}^{n} e_{ik}^2 m_{ik} = \sum_{i=1}^{N}\sum_{k=1}^{n}(x_{ik} - \sum_{t=1}^{p} c_{tk}z_{it})^2 m_{ik} \tag{3.2.6}$$

where $m_{ik} = 0$ at missings and $m_{ik} = 1$, otherwise.

To minimize criterion (3.2.6), the one-by-one strategy of the principal component analysis is utilized. According to this strategy, computations are carried out iteratively. At each iteration t, $t = 1, ..., p$, only one factor is sought for to minimize criterion:

$$l_2 = \sum_{i=1}^{N}\sum_{k=1}^{n}(x_{ik} - c_k z_i)^2 m_{ik} \tag{3.2.7}$$

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.

To minimize (3.2.7), the same method of alternating minimization is utilized. Each iteration proceeds in two steps: (1) given a vector (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 extending (3.2.4) and (3.2.5) to:

$$z_i = \frac{\sum_{k=1}^{n} x_{ik} m_{ik} c_k}{\sum_{k=1}^{n} c_k^2 m_{ik}} \tag{3.2.8}$$

and

$$c_k = \frac{\sum_{i=1}^{N} x_{ik} m_{ik} z_i}{\sum_{i=1}^{N} z_i^2 m_{ik}} \tag{3.2.9}$$

Basically, it is this procedure that was variously described in Gabriel and Zamir [1979], Grung and Manne [1998], Mirkin [1996]. The following is a more formal presentation of the algorithm.

```
ILS Algorithm

0. Set number of factors $p$ and $\epsilon > 0$, a pre-specified precision threshold.

1. Set iteration number t=1.

2. Initialize n-dimensional $\mathbf{c}^{*\prime} = (1, \ldots, 1)$ and normalize it.

3. Given $\mathbf{c}^*$, calculate $\mathbf{z}$ according to (3.2.8).

4. Given $\mathbf{z}$ from step 3, calculate $\mathbf{c}$ according to (3.2.9) and normalize it
   afterwards.

5. If $||c - c^*|| > \epsilon$, put $\mathbf{c}^* = c$ and go to 3.

6. If $t < p$ set $\mathbf{c}_t = c$, $\mathbf{z}_t = z$, then update $x_{ik} = x_{ik} - c_{tk} z_{tk}$ for $(i, k)$ such that
   $m_{ik} = 1$ and $t = t + 1$ and go to 2, otherwise end.

7. Impute the missing values $x_{ik}$ at $m_{ik} = 0$ according to (3.2.1) with $e_{ik} = 0$.
```

There are two issues which should be taken into account when implementing ILS:

1. Convergence.

   The method may fail to converge depending on configuration of missings and starting point. Some other causes of non-convergence such as those described in [Grung and Manne, 1998] have been taken care of in the formulation of the algorithm. In the present approach, somewhat simplistically, the normed vector of ones was used as the starting point (step 2 in the algorithm above). However, sometimes a more sophisticated choice is required as the iterations may come to a "wrong convergence" or no converge at all. To this end, Gabriel and Zamir [Gabriel and Zamir, 1979] 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

$$\omega_{ik} = \sum_b m_{bk} x_{bk}^2 + \sum_d m_{id} x_{id}^2 \tag{3.2.10}$$

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

2. With these i and k, compute

$$\beta = \frac{\sum_{b \neq i} \sum_{d \neq k} m_{bd} x_{bk}^2 x_{id}^2}{\sum_{b \neq i} \sum_{d \neq k} m_{bd} x_{bk} x_{id} x_{bd}} \tag{3.2.11}$$

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

$$\mathbf{c}^{*\prime} = (x_{i1} \ldots x_{ik-1}, \beta, x_{ik+1} \ldots, x_{in}) \tag{3.2.12}$$

The method is rather computationally intensive and may cause to slow down the speed of computation (up to 60 times in our experiments). However, it can be very 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 [Wold, 1966] and his student Christoffersson under the name of "nonlinear iterative partial least squares" (NIPALS). In most cases the one-factor technique leads to significant errors, which implies the need for more factors. Selection of p may be driven by the same scoring function as selection of the number of principal components: the proportion of the data variance taken into account by the factors. This logic is well justified in the case of the principal component analysis in which model (3.2.1) fits the data exactly when p is equal to the rank of $\mathbf{X}$. When missings are present in the data, the number of factors sequentially found by ILS may be infinite. However, the logic is still justified since we can prove that the residual data matrix converges to the zero matrix as follows (See also Statement 2.2 in [Mirkin, 1996]).

Define $\Gamma = \{(i,k)|\mathbf{x}_{ik} \text{ is not missed}\}$ and $\mathbf{X}^* = \{x_{ik}|(i,k) \in \Gamma\}$. For simplicity purpose, for any sets $\mathbf{A} = (A_{ik}|(i,k) \in \Gamma)$ and $\mathbf{B} = (B_{ik}|(i,k) \in \Gamma)$ the following notation to be used:

$$(\mathbf{A},\mathbf{B}) = \sum_{(i,k)\in\Gamma} \mathbf{A}_{ik} * \mathbf{B}_{ik} \tag{3.2.13}$$

The bilinear model described in (3.2.1), can be represented in the following way:

$$x_{ik} = \sum_{t=1}^{p} \mathbf{c}_{tk}\mathbf{z}_{it} + \mathbf{E}_{ik}, \ \ (i,k) \in \Gamma \tag{3.2.14}$$

where $\mathbf{z}_{it}$ and $\mathbf{c}_{tk}$ are N-dimensional and n-dimensional vectors and the residual $\mathbf{E}$ is least squares minimized. By denoting the residual $\mathbf{E}_{Obs}$ by $\mathbf{X}_{t+1}$, ILS method can be presented as:

$$\mathbf{X}_t = \mathbf{z}_t\mathbf{c}_t^T + \mathbf{X}_{t+1} \tag{3.2.15}$$

with equations (3.2.1) holding at $(i,k) \in \Gamma$. By mutliplying (3.2.15) by itself we obtain:

$$(\mathbf{X}_t, \mathbf{X}_t) = (\mathbf{z}_t\mathbf{c}_t^T, \mathbf{z}_t\mathbf{c}_t^T) + (\mathbf{X}_{t+1}, \mathbf{X}_{t+1}) \tag{3.2.16}$$

According to equation (3.2.16) the value $g_t = (\mathbf{z}_t\mathbf{c}_t^T, \mathbf{z}_t\mathbf{c}_t^T)$ is contribution of t-th factor to the squared norm of the residual data observed $(\mathbf{X}_t, \mathbf{X}_t)$. To derive a lower boundary to g_t, let us consider an admissible solution to minimization of least squares $(\mathbf{X}_{t+1}, \mathbf{X}_{t+1})$ according to (3.2.15). This admissible solution is defined by vector $z = v(i^*)$ all components of which are zero except for i^*-th component equal to 1. Then according to (3.2.9) the optimal $\mathbf{c}_k$ which is at k^* such that $(i^*, k^*) \in \Gamma$, will be equal to $x_{ti^*k^*}$, the (i^*, k^*)-th element of matrix $\mathbf{X}_t$. Then obviously the combination of $(\mathbf{zc}^T)$ to $(\mathbf{X}_t, \mathbf{X}_t)$ according to (3.2.15) will be $(\mathbf{zc}^T, \mathbf{zc}^T) = \sum_{k^* \in \Gamma} x_{ti^*k^*}^2$. Thus the optimal contribution g_t must be not less than $x_{ti^*k}^2$ for arbitrary $(i^*, k) \in \Gamma$.

Let $|x_{tik}| = max_{i'=1,\dots,N;k'=1,\dots,n}|x_{ti'k'}|$ then obviously $(\mathbf{X}_t, \mathbf{X}_t)/Nn \leq x_{tik}^2$. Thus $g_t \geq x_{tik}^2 \geq (\mathbf{X}_t, \mathbf{X}_t)/Nn$. From this the following inequality can be easily derived by induction over t:

$$(\mathbf{X}_{t+1}, \mathbf{X}_{t+1}) = (\mathbf{X}_t, \mathbf{X}_t) - g_t \leq (\mathbf{X}, \mathbf{X})(1 - \frac{1}{Nn})^t \;\; t = 1, 2, \dots \qquad (3.2.17)$$

However, $(1 - \frac{1}{Nn})^t$ converges to 0 as t increases, therefore, $(\mathbf{X}_t, \mathbf{X}_t) \to 0$ □.

3.2.2 Iterative Majorization Least-Squares (IMLS) Algorithm

This method is an example of an application of the general idea that the weighted least squares minimimization 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 [Heiser, 1995]. In this framework, Kiers

[Kiers, 1997] developed the following algorithm, that in its final form can be formulated without 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, that is, pair $\mathbf{z}$ and $\mathbf{c}$, which will be referred to as Kiers algorithm and described first. The Kiers algorithm operates with a completed version of matrix $\mathbf{X}$ denoted by $\mathbf{X}^s$ where $s = 0, 1, ..$ is the iteration's number. At each iteration s, the algorithm finds one best factor of the SVD decomposition of $\mathbf{X}^s$ and imputs the results into the missing entries, after which the next iteration starts.

Kiers Algorithm

1. *Set* $\mathbf{c}' = (1, ..., 1)$ *and normalize it.*

2. *Set* $s = 0$ *and define matrix* $\mathbf{X}^s$ *by putting zeros into missing entries of* $\mathbf{X}$*. Set a measure of quality* $h_s = \sum_{i=1}^{N} \sum_{k=1}^{n} {x_{ik}^{s}}^2$*.*

3. *Find the first singular triple* $\mathbf{z}_1$*,* $\mathbf{c}_1$*,* μ *for matrix* $\mathbf{X}^s$ *by applying the iterative SVD algorithm with* $p = 1$ *and denote the resulting value of criterion (3.2.6) by* h_{s+1}*. (Vectors* $\mathbf{z}_1$*,* $\mathbf{c}_1$ *are assumed normalised here.)*

4. *If* $|h_s - h_{s+1}| > \epsilon * h_s$ *for a small* $\epsilon > 0$*, set* $s = s + 1$*, put* $\mu z_{i1} c_{1k}$ *for any missing entry* (i, k) *in* $\mathbf{X}$ *and go back to step 3.*

5. *Set* $\mu \mathbf{z}_1$ *and* $\mathbf{c}_1$ *as the output.*

Now the IMLS algorithm [Kiers, 1997] can be formulated with its properties yet to be explored.

```
IMLS Algorithm

0. Set the number of factors $p$.

1. Set iteration number t=1.

2. Apply Kiers algorithm to matrix $\mathbf{X}$ with the missing structure $\mathbf{M}$.
   Denote results by $\mathbf{z}_t$ and $\mathbf{c}_t$.

3. If $t = p$, go to step 5.

4. For $(i, k)$ such that $m_{ik} = 1$, update $x_{ik} = x_{ik} - c_{tk}z_{tk}$, put t=t+1 and go
   to step 2.

5. Impute the missing values $x_{ik}$ at $m_{ik} = 0$ according to (3.2.1) with $e_{ik} = 0$.
```

Chapter 4

Combining Nearest Neighbour Approach with the Least Squares Imputation

4.1 A Review of Lazy Learning

The term “lazy learning”, also known as instance-based learning, applies to a class of local learning algorithms which could be characterized by the following three key properties [Aha et al., 1991, Aha, 1997, Mitchel, 1997]:

1. The computations are postponed until they receive a request for prediction.
2. Then, a request for prediction is responded by combining information from training samples.
3. Finally, the constructed answer and intermediate results are discarded.

These properties would distinguish the term lazy learning from the other type learning which is referred to as “eager learning”. In the latter type, the learning is accomplished before a request for prediction is received. The advantages of lazy learning approach are summarized as follows:

1. During the training session, the lazy algorithms have fewer computations than the eager algorithms. Thus, lazy learning is very suitable for incremental learning tasks, i.e. if the data stream is continually updated [Aha, 1998].

2. The lazy algorithms provide highly adaptive behaviour under local approaches which are implemented in subsequent problems [Bottou and Vapnik, 1992].

3. Lazy algorithms can inspire abstractions of complex tasks, for instance in developing the lazy version of backpropagation that learns a different neural network for each new query [Bottou and Vapnik, 1992, Mitchel, 1997].

4. Lazy learning proposes conceptually straightforward approaches to approximating real-valued or discrete-valued target functions [Atkeson et al., 1997].

There are three well-known broad approaches within this framework to which the machine learning community has paid much attention:

1. k-Nearest Neighbour (k-NN).

   This is the most basic lazy learning technique which involves three main characteristics as explored in [Aha et al., 1991, Mitchel, 1997, Wettschereck and Dietterich, 1994]:

   - The implementation of the nearest neighbour algorithm is based on the assumption that all points in the n-dimensional space represent all entities.

   - The decision of how to generalize beyond the training data is postponed until a request for prediction is received.

- The prediction is accomplished based on “similar entities” only. Thus, according to this criterion, the k-nearest neighbours of a target entity $\mathbf{X}_i$ and its neighbour entity $\mathbf{X}_j$ are defined in terms of the standard Euclidean distance which can be computed as follows:

$$D_2(\mathbf{X}_i, \mathbf{X}_j) = \sum_{k=1}^{n}[x_{ik} - x_{jk}]^2; i, j = 1, 2, \dots N \qquad (4.1.1)$$

2. Locally weighted regression.

   In locally weighted regression, values are weighted by proximity to the current query using a kernel function which is defined as a function of distance that is used to determine the weight of each training data value. A regression is then computed using the weighted values [Aha, 1997, Atkeson et al., 1997, Bottou and Vapnik, 1992].

3. Case-based reasoning.

   In this approach, case-based reasoning expertise is expressed by a collection of past instances (cases) which consist of enrichment of symbolic descriptions. Each entity typically contains a description of the problem including a solution. The knowledge and reasoning process used by an expert to the solve the problem is implicit in the solution [Aha, 1991, Kolodner, 1993, Mitchel, 1997].

For simplicity purpose, this work implements the k-NN algorithm approach for constructing local versions of the least squares imputation. To do this, some aspects of the traditional k-NN discussed widely in the literature can be extended in the following ways:

1. The distance measurement for incomplete data.

Since the traditional k-NN algorithm can only possible be applied to the complete data case, it is necessary to extend the conventional distance measurement (see for instance in [Hastie et al., 1999, Troyanskaya et al., 2001]).

2. Selection of neighbours criterion.

   As some attributes contain missing entries for some entities, there are two possibilities to select the neighbours: (1) select the neighbour as is; (2) select the neighbours which contain no missing entries in corresponding attributes to the target entity.

The details of the extension of the k-NN algorithm for incomplete data imputation will be explored in the following section.

4.2 Nearest Neighbour Imputation Algorithms for Incomplete Data

As explained in the previous section, in order to determine and select the neighbour in case of some entities containing missing entries, the adaptation of nearest neighbour algorithm for incomplete data need to be developed. The ultimate objective is to extend the two global least squares imputations as described in Section 3.2.1 and 3.2.2 into their local versions by using techniques from the nearest neighbour framework.

In this approach, the imputations are carried out sequentially, by analyzing 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. The imputation model such as (3.2.1), 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.

Briefly, the k-NN based 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 matrix $\mathbf{X}$ consisting of the target entity and the neighbours. Then apply an imputation algorithm to the matrix $\mathbf{X}$, imputing missing entries at the target entity only. Repeat this until all missing entries are filled in. Then output the completed data matrix.

To apply the k-NN approach to incomplete data, the following two issues should be addressed.

4.2.1 Measuring Distance.

There can be a multitude of distance measures considered. Euclidean distance squared was chosen as this measure is compatible with the least squares framework. Thus, the equation (4.1.1) is extended in the following form:

$$D_2(\mathbf{X}_i, \mathbf{X}_j, \mathbf{M}) = \sum_{k=1}^{n} [x_{ik} - x_{jk}]^2 m_{ik} m_{jk}; i, j = 1, 2, \dots N \qquad (4.2.1)$$

where m_{ik} and m_{jk} are missingness values for x_{ik} and x_{jk}, respectively. This distance was also used in [Hastie et al., 1999, Myrtveit et al., 2001, Troyanskaya et al., 2001].

4.2.2 Selection of the Neighbourhood.

The principle of selecting the closest entities can be realized, 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 [Hastie et al., 1999, Myrtveit et al., 2001, Troyanskaya et al., 2001] for data imputation with the Mean method. The proposed methods apply the same approach when using this method. However, for ILS and IMLS, the presence of missing entries in the neighbouring entities creates no problems, therefore, for these methods, all entities were selected.

4.3 Least Squares Imputation with Nearest Neighbour

Basically, this method involves three main procedures which can be accomplished in the following steps: first, search the entity that contains missing entries, referred to as the target entity; then find its neighbours based on the distance measure in 4.2.1 regardless of the missingness in the corresponding attributes of the target entity; finally impute the missings in the target entity on the subset of the data matrix which consists of the target entity and its closest entities using ILS and IMLS algorithms. Repeat the procedures until all entities contain no missing entries. More formally, the algorithms can be described as follows:

```
NN Version of Imputation Algorithm A($\mathbf{X}, \mathbf{M}$)

0. Observe the data. If there are no missing entries, end.

1. Take the first row that contains a missing entry as the target entity $\mathbf{X}_i$.

2. Find K neighbours of $\mathbf{X}_i$.

3. Create a data matrix $\mathbf{X}$ consisting of $\mathbf{X}_i$ and K selected neighbours.

4. Apply imputation algorithm $A(\mathbf{X}, M)$, and impute missing values in $\mathbf{X}_i$ and
   go to 0.
```

To make the NN-based imputation algorithms work fast, let K be of the order of 5 to 10 entities to be chosen. Then, to apply the least squares imputation techniques, the number of factors is restricted to guarantee that the subspace approximation processes converge. Thus, $p = 1$ taken alongside the Gabriel-Zamir's initialization in ILS was implemented. The ILS algorithm may still lead to nonconvergent results because of the small NN data sizes.

4.4 Global-Local Least Squares Imputation Algorithm

One more NN based approach can be suggested to combine nearest neighbour approach with the global imputation algorithms described in Section 3. In this approach, a global imputation technique was used to fill in all the missings in matrix **X**. Suppose the resulting matrix is denoted as $\mathbf{X}^*$. Then nearest neighbour technique is applied to fill in the missings in **X** again, but, this time, based on distances computed with the completed data $\mathbf{X}^*$.

The same distance formula (4.2.1) can be utilized in this case as well, by assuming that all values m_{ik} are unities, which is the case when matrix $\mathbf{X}^*$ is utilised. This distance will be referred to as the *prime distance*.

This proposed technique is an application of this global-local approach involving IMLS at both stages. This technique, referred to as INI from this point on, will include four main steps. Firstly, impute missing values in the data matrix **X** by using IMLS with p factors. Then compute the prime distance metric with the found $\mathbf{X}^*$. Take a target entity according to **X** and apply the NN algorithm to find its neighbours according to $\mathbf{X}^*$. Finally, impute all the missing values in the target entity with NN-based IMLS technique (this time, with $p = 1$).

INI Algorithm

1. *Apply IMLS algorithm to* $\mathbf{X}$ *with* $p > 1$ *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* $\mathbf{X}_i$ *of* $\mathbf{X}$*.*

6. *If no missing entries remain, stop; otherwise go back to step 2.*

4.5 Related Work

4.5.1 Nearest Neighbour Mean Imputation (N-Mean)

The Mean imputation within nearest neighbour framework has been successfully proposed in the context of Bioinformatics problem (see for instance in [Hastie et al., 1999, Troyanskaya et al., 2001]). As described in previous section, in the N-Mean, the neighbours are selected by considering only entities with non-missing entries in the attribute corresponding to that of the target's missing entry. Then, the missing values are imputed with weighted average of the neighbours .

The results show that this approach provide very fast, robust and accurate ways of imputing missing values for microarray data and far surpasses the currently accepted solutions such as: filling missing values by zeros or Mean imputation. However, the performance of the N-Mean imputation deteriorates as the proportion of missings grows.

4.5.2 Similar Response Pattern Imputation (SRPI)

The similar response pattern method for handling missing data had been paid attention to in the Software Engineering community as described in [Myrtveit et al., 2001, SSI, 1995]. Basically, the SRPI method is a general form of the N-Mean imputation. This approach utilizes a mechanism which is similar to with N-Mean in terms of selecting the neighbours such that the entities should have no missing at the column values corresponding to that of the target's missing entry. Then the entities are selected according to the Euclidean distance squared:

$$Q = \sum_{k=1}^{n} [x_{\ell k} - x_{\jmath k}]^2 \tag{4.5.1}$$

where index ℓ and $\jmath = 1, 2, \ldots, N$, $\jmath \neq \ell$, denote the target entity and the neighbours candidate to be selected, respectively.

The distance (4.5.1) is minimized over $j \neq l$ such that two possibilities to impute the missing values in $\mathbf{x}_{\ell.}$ may occur:

1. There is only one $\mathbf{x}_{\jmath.}$ found. In this case, the missing value in $\mathbf{x}_{\ell.}$ is imputed with $\mathbf{x}_{\jmath.}$.

2. More than one $\mathbf{x}_{\jmath}$ are found by minimizing (4.5.1). In this case, use the average of them to fill in missing value in $\mathbf{x}_{\ell.}$.

The use of the squared Euclidean distance (4.5.1) suggests that the SRPI method does not involve the outlier values to impute the missing values. However, this approach requires a through knowledge of data with regard to selection of the neighbours of the target entity.

This approach provides promising performance according to the results in [Myrtveit et al., 2001]. In a commercial application, the SRPI method has been implemented in the software package PRELIS 2.3 [SSI, 1995].

Chapter 5

Experimental Study of Least Squares Imputation

The experimental study of the least squares imputation and their extensions to be carried out through several massive experiments within simulation framework. The main goal of the experimental study is twofold:

1. To compare the performance of various least squares data imputation on Gaussian mixture data models with Complete Random missing pattern.

2. To study the performance of least squares data imputation with different missing patterns.

5.1 Selection of Algorithms

The goals specification above lead us to consider the following eight least squares data imputation algorithms as a representative selection:

1. ILS-NIPALS 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 ILS with $p = 1$.
7. N-IMLS: NN based IMLS-1.
8. INI: NN based IMLS-1 imputation based on distances from an IMLS-4 imputation.

Of these, the first five are versions of the global ILS and IMLS methods, the next two are nearest neighbour versions of the same approaches, and the last algorithm INI combines local and global versions of IMLS. Similar combined algorithms involving ILS have been omitted here since they do not always converge. 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.

5.2 Gaussian Mixture Data Models

This experimental study applies two types of data model generation. The mechanism to generate each data model will be described in turn.

5.2.1 NetLab Gaussian Mixture Data Model

Gaussian mixture data model is described in many monographs (see, for instance, [Everrit and Hand, 1981]). In this model, a data matrix $\mathbf{D}_{N\times n}$ is generated randomly from the Gaussian mixture distribution with a probabilistic principal component analysis (PCA) covariance matrix [Roweis, 1998, Tipping and Bishop, 1999a]. For now on the term Gaussian p-mixture is referred to a mixture of p Gaussian distributions (classes). The following three-step procedure, Neural Network NETLAB, is applied as implemented in a MATLAB Toolbox freely available on the web [Nabney, 1999]:

1. Architecture: set the dimension of data equal to n, number of classes (Gaussian distributions) to p and the type of covariance matrix based on the probabilistic PCA in a q dimension subspace. In our experiments, p is 5, n between 15 and 25, and q typically is $n-3$.

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

   $$\mathbf{W}_q = \begin{pmatrix} \mathbf{I}_{q\times q} \\ \mathbf{1}_{(n-q)\times q} \end{pmatrix} \tag{5.2.1}$$

   where $\mathbf{I}_{q\times q}$ and $\mathbf{1}_{(n-q)\times q}$ are the identity matrix and matrix of ones, respectively.

   In the experiments, the general variance σ^2 is set to be equal to 0.1. The probabilistic PCA (PPCA) covariance matrix is computed as follows:

$$\mathbf{C}ov = \mathbf{W}_q * \mathbf{W}_q' + \sigma^2 \mathbf{I}_{n\times n} \tag{5.2.2}$$

3. Data: generate randomly data matrix $\mathbf{D}_{N\times n}$ from the Gaussian mixture distribution as follows:

```
Compute eigenvectors and corresponding eigenvalues of $\mathbf{C}ov$ and denote
the matrix of eigenvectors by $\mathbf{e}vec$ and vector of the square roots of
eigenvalues by $\sqrt{\mathbf{e}igen}$.
For $i = 1, \dots, p$:
    Set $N_i = N/p$, the number of rows in i-th class.
    Generate randomly $\mathbf{R}_{(N_i\times n)}$ based on Gaussian distribution N(0,1).
    Compute $\mathbf{D}_i = \mathbf{1}_{N_i\times 1} * \mathbf{a}vg_i' + \mathbf{R} * diag(\sqrt{\mathbf{e}igen}) * \mathbf{e}vec'$.
end
Define $\mathbf{D}$ as $N \times n$ matrix combining all generated matrices $\mathbf{D}_i$, $i = 1, ..., p$.
```

5.2.2 Exploration of NetLab Gaussian Mixture Data Model

The structure of (5.2.1) is rather simple and produces a simple structure of covariance (5.2.2) as well. Indeed, it is not difficult to show that

$$\mathbf{C}ov(0) = \begin{pmatrix} \mathbf{I}_{q\times q} & \mathbf{1}_{q\times(n-q)} \\ \mathbf{1}_{(n-q)\times q} & q\mathbf{1}_{(n-q)\times(n-q)} \end{pmatrix} \tag{5.2.3}$$

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. Let us denote the sum of $\mathbf{x}_q$ by a and the sum of $\mathbf{x}_{n-q}$ by b. Obviously, to be an eigenvector of $\mathbf{C}ov(0)$ corresponding to its eigenvalue λ, $\mathbf{x}$ must satisfy the following equations:

$$\mathbf{x}_q + b\mathbf{1}_q = \lambda \mathbf{x}_q \text{ and } (a+qb)\mathbf{1}_{n-q} = \lambda \mathbf{x}_{n-q}.$$

With little arithmetics, these imply that $\mathbf{C}ov(0)$ has only two nonzero eigenvalues, the maximum $\lambda = 1 + (n-q)q$ and second-best $\lambda = 1$. 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 $\lambda = 1 + (n - q)q$ 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 Gaussian distributions, still will tend to be approximately unidimensionally distributed along the first eigenvector.

Changing σ in $\mathbf{C}ov(\sigma)$ 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 very high. Thus the model as is would yield very small deviations of generated data sets from the unidimensional case.

5.2.3 Scaled NetLab Gaussian Mixture Data Model

To better control the complexity of generated data sets, then the modification of the Gaussian mixture data model above is called for. The improvement is carried out by differently scaling the covariance matrix $\mathbf{C}ov(\sigma)$ and the mean vector $\mathbf{a}vg$ for each class. To do this, for each Gaussian class $i = 1, ..., p$, the random scaling factor, b_i, to be utilized in order to move $\mathbf{a}vg_i$ away from the origin. Also scaling the covariance matrix by factor a_i to be taken as proportional to i. The dimension of the PPCA model is taken as $q = [n/2]$. In brief, by using the architecture and data structure described above, the data generator can be summarized as follows:

```
For $i = 1, \dots, p$, given $\mathbf{a}vg_i$ and $\mathbf{C}ov_i(\sigma)$ from NetLab:
    Randomly generate the scaling factor $b_i$ in the range range between 5 and 15.
    Compute scaled $\mathbf{C}ov_i$ as $\mathbf{C}ov_i = 0.8 * i * b_i * \mathbf{Cov}(\sigma)$.
    Compute eigenvalues and corresponding eigenvectors of $\mathbf{C}ov_i$ and denote
    the matrix of eigenvectors by $\mathbf{e}vec_i$ and vector of the square roots of
    eigenvalues by $\sqrt{\mathbf{e}igen}_i$ .
    Set $N_i = N/p$, the number of rows in i-th class.
    Generate randomly $\mathbf{R}_{(N_i * n)}$ according to Gaussian distribution N(0,1).
    Compute $\mathbf{D}_i = b_i * \mathbf{1}_{N_i \times 1} * \mathbf{a}vg_i' + \mathbf{R} * diag(\sqrt{\mathbf{e}igen}_i) * \mathbf{e}vec_i'$.
end
Define $\mathbf{D}$ as $N \times n$ matrix combining all generated matrices $\mathbf{D}_i$, $i = 1, ..., p$.
```

5.3 Mechanisms for Missing Data

5.3.1 Complete Random Pattern

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. The proportion of missing entries may vary. The random uniform distribution is used for generating missing positions and the proportion's range at 1%, 5%, 10%, 15%, 20% and 25% of the total number of entries.

5.3.2 Inherited Pattern

In this scheme, the same range of proportions of missing entries as above is specified. Then, given the size $N \times n$ of data matrix, a 25% set P of missing entries (i, j), $i = 1, ..., N; j = 1, ..., n$, is generated from the uniform distribution. The next 20% set of missing entries is created to be part of this P by randomly selecting 80% of the entries in P. These 80% form a 20% missing set to be taken as P for the next step. The next inherited sets of missing entries are created similarly, by randomly selecting 75%, 66.7%, 50%, 20% of elements in the previous set P, respectively. This way, a nested set of six sets of missing entries is created, representing an Inherited pattern.

5.3.3 Sensitive Issue Pattern

According to the model accepted, missings may occur only at a subset of entities with regard to a subset of issues. In the experiments, additional constraints on selection of the "sensitive" rows and columns to be maintained, to avoid trivial patterns. The missings under this scenario are carried out as follows:

Sensitive Issue Pattern Generation

Given proportion p of missings entries, randomly select proportions c of sensitive issues (columns) and r of sensitive respondents (rows) such that $p < cr$.
Then, in the data submatrix formed by the selected columns and rows randomly generate proportion of p/cr missings.
Accept the following additional constraints on the values generated:
1. $10\% < c < 50\%$ *and* $25\% < r < 50\%$ *for* $p = 1\%$.
2. $20\% < c < 50\%$ *and* $25\% < r < 50\%$ *for* $p = 5\%$.
3. $25\% < c < 50\%$ *and* $40\% < r < 80\%$ *for* $p = 10\%$.

5.3.4 Merged Database Pattern

In this pattern, two scenarios for merging two databases to be implemented which can be categorized as:

1. Missings come from only one database.

2. Missings come from both of the databases.

5.3.4.1 Missings from One Database

Under this scenario, the missings are generated as follows. First, specify the proportion p% of missing entries on the merged database. Then generate q% of columns consist of missings entries in the merged database. These are assumed to come from

	$\mathbf{X}_1$	$\mathbf{X}_2$	...	$\mathbf{X}_{n-1}$	$\mathbf{X}_n$
1	O	O	...	U	U
2	O	O	...	U	U
3	O	O	...	U	U
...	...	...	...	...	...
N_1	O	O	...	U	U
$N_1 + 1$	O	O	...	O	O
$N_1 + 2$	O	O	...	O	O
$N_1 + 3$	O	O	...	O	O
...	...	...	...	...	...
N	O	O	...	O	O

Table 5.1: A pattern of data at which missings come from one database, where U and O denote missing and not-missing entries, respectively.

database at which the corresponding variables are missed. Finally, the proportion of respondents (rows) is computed as $t = p/q$. (see Table 5.1).

In the experiments, $q = 20\%, 30\%$ are selected for generating 1% and 5% missings.

5.3.4.2 Missings from Two Databases

Suppose each of the two databases to be merged such that each contain variables that are absent in the other variables. The merged database will have a pattern presented in Table 5.2 at which the variables which are present only in the first database are placed on the left while variables that are present only in the second database are placed on the right. A procedure to generate the missings of this type will be introduced as follows.

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

	$\mathbf{X}_1$	$\mathbf{X}_2$	$\dots$	$\mathbf{X}_{n-1}$	$\mathbf{X}_n$
1	O	O	$\dots$	U	U
2	O	O	$\dots$	U	U
3	O	O	$\dots$	U	U
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
N_1	O	O	$\dots$	U	U
N_1+1	U	U	$\dots$	O	O
N_1+2	U	U	$\dots$	O	O
N_1+3	U	U	$\dots$	O	O
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
N	U	U	$\dots$	O	O

Table 5.2: A pattern of data at which missings come from two databases, where U and O denote missing and not-missing entries, respectively.

$$p = \frac{k_1N_1 + k_2N_2}{nN} \tag{5.3.1}$$

where $N = N_1 + N_2$. This implies that, given p and k_1, k_2 can be determined from equation

$$k_2 = \frac{pnN - k_1N_1}{N_2} \tag{5.3.2}$$

where $N = N_1 + N_2$. A procedure to generate missings of this type will be introduced as follows:

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 of 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 satisfying the constraint $k_1 < \frac{npN-N_2}{N_1}$.*
4. *Compute k_2 according to equation (5.3.2).*
5. *Finally, put $m_{ik} = 0$ for all $i = 1, \dots, N_1$, $k = 1, \dots, k_1$ and for all $i = N_1 + 1, \dots, N$, $k = n, n-1, \dots, n - k_2 + 1$.*

5.4 Evaluation of Results

Since the data and missings are generated separately, the quality of imputation is evaluated by comparing the imputed values with those generated at the stage of data generating. The following measurements are frequently used for assessing the quality of imputation in many literature (see for instance [Chambers, 2000, Hoogland and Pannekoek, 2000, Strauss et al., 2002]):

1. Mean absolute deviation (MAD) of imputed value is defined as:

   $$\frac{1}{d}\sum_{i=1}^{N}\sum_{k=1}^{n}(1-m_{ik})|x^*_{ik}-x_{ik}| \tag{5.4.1}$$

   where d, m_{ik} and x^*_{ik} denote number of missing values, the missingness matrix entry and x^*_{ik} the data matrix $\mathbf{X}^*$ with imputed values, respectively.

2. Mean absolute relative deviation (MARD) of x^*_{ik} is defined as

   $$\frac{1}{d}\sum_{i=1}^{N}\sum_{k=1}^{n}(1-m_{ik})|(x^*_{ik}-x_{ik})/x_{ik}| \tag{5.4.2}$$

3. Distribution of imputed values.

   In this criterion, the empirical distribution of x^*_{ik} is compared to that of x_{ik} correspondingly. This is carried out by computing their mean and standard deviation or higher order statistics such as skewness and kurtosis.

However, the above measurements mainly determine the variation of the imputed values or parameters of data distribution which could be incompatible to the least squares framework of the algorithms considered in this project. Thus, rather using the above criteria, this experiment utilizes the squared imputation error, *IE*, to measure the performance of an algorithm. The measure is defined as follows:

$$IE = \frac{\sum_{i=1}^{N}\sum_{k=1}^{n}(1-m_{ik})(x_{ik}-x_{ik}^{*})^2}{\sum_{i=1}^{N}\sum_{k=1}^{n}(1-m_{ik})x_{ik}^2} \tag{5.4.3}$$

5.5 Results of the Experimental Study of Imputations with the Complete Random Missing Pattern

5.5.1 Experiments with NetLab Gaussian Mixture Data Model

The experiments are carried out with data generated as Gaussian 5-mixture with the dimension of the PPCA subspace equal to n-3. Ten data sets of random sizes (200 - 250 rows and 15-25 columns) were generated for each of this data type. Also ten sets of missings patterns were generated for each of the following levels of data missing: 1%, 5%, 10%, 15%, 20% and 25%. Altogether, there are 60 various patterns of missings for each of the data sets. The results of running of all the ten imputation methods over each of the 10 data sets with the 60 missing patterns will be presented.

The results of the experiments are presented in Table 5.3. The performance of N-ILS reflect somewhat poor convergence of the method at the levels of missing greater than 1% which is labeled as “N/A”. As expected, for each of the algorithms, except the Mean, the error increases as the number of missings grows. At the Mean method, the error is constant, of the order of 95%.

The obvious winner, at each level of missings, is INI, the global-local version of least squares imputation. In a close range, it is followed by IMLS-4 and GZ.

N-IMLS method is the second best when missings are sparse, but it falls out of the range when the proportion of missings grows to 25 %. These conclusions may

Methods	*Proportion of Missings*					
	1%	5%	10%	15%	20%	25%
ILS	41.25 (15.00)	39.66 (7.91)	38.68 (4.64)	40.42 (6.00)	45.33 (7.82)	48.56 (9.58)
GZ	41.26 (15.01)	39.66 (7.91)	38.67 (4.67)	40.43 (6.02)	45.27 (7.83)	48.25 (8.99)
NIPALS	56.49 (20.12)	50.64 (10.27)	49.54 (5.46)	49.75 (6.76)	54.14 (8.88)	55.25 (9.51)
IMLS-1	56.59 (20.19)	50.59 (10.35)	49.38 (5.52)	48.93 (4.82)	51.75 (5.49)	52.35 (5.80)
IMLS-4	41.25 (14.99)	39.66 (8.11)	38.68 (4.56)	40.42 (4.37)	45.33 (4.50)	48.56 (6.41)
Mean	97.13 (8.50)	95.82 (2.94)	96.19 (3.01)	96.11 (1.89)	95.72 (1.90)	95.62 (1.51)
N-ILS	35.14 (14.02)	N/A	N/A	N/A	N/A	N/A
N-IMLS	35.04 (13.96)	34.71 (7.83)	36.80 (7.64)	41.82 (8.67)	53.58 (17.38)	66.75 (19.06)
INI	35.29 (13.03)	33.19 (6.81)	33.27 (4.83)	34.89 (4.75)	38.93 (5.47)	43.01 (8.05)
N-Mean	37.66 (13.19)	41.98 (7.32)	50.96 (5.67)	59.32 (5.91)	69.70 (7.80)	80.98 (8.58)

Table 5.3: The average squared error of imputation and its standard deviation (%) at NetLab Gaussian 5-mixture data model with different levels of missing entries.

be blurred by the overlapping standard deviations of the methods' average errors. Therefore, the results of direct pairwise comparisons between the methods should be shown as well. At this perspective, the results appear to depend on the level of missings: there are somewhat different situations at 1% missings and at the other, greater, missing levels, which are similar to each other (see Tables 5.4 and 5.5).

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

Table 5.4: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on NetLab Gaussian 5-mixture data model with $[n-3]$ PPCA factors for 1% random missing data.

The results show that, overall, there are three different patterns in the pairwise comparison: (1) at 1% missings (Table 5.4), (2) at 5 % missings, 10% and more missings (Table 5.5).

At 1% missings, according to Table 5.4, there are four winners, all the nearest neighbour based methods, N-Mean included. Although N-Mean loses to INI by 30% to 70%, it outperforms the others in winning over one-dimensional NIPALS and IMLS-1.

Methods of Imputation	5%				15%			
	1	2	3	4	1	2	3	4
ILS	80	80	100	10	40	50	100	0
GZ	80	80	100	10	40	60	100	0
NIPALS	100	100	100	90	60	100	100	10
IMLS-1	100	100	100	90	70	100	100	10
IMLS-4	80	90	100	20	40	60	90	0
Mean	100	100	100	100	80	100	100	100
N-ILS	-	70	70	20	-	100	100	0
N-IMLS	30	-	70	20	0	-	100	0
INI	30	30	-	0	0	0	-	0
N-Mean	80	80	100	-	80	100	100	-

Table 5.5: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on Gaussian 5-mixture with $[n-3]$ PPCA factors for 5% and 15% random missing data where 1,2,3 and 4 denote N-ILS, N-IMLS, INI, N-Mean, respectively (the other methods are not shown because of poor performance).

Unfortunately, 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.

This time, there are only three winners, INI, N-IMLS and N-ILS, that are ordered in such a way that the previous one wins over the next one(s) in 70% of the cases. Thus, INI leads the contest and Mean loses it on almost every count, at the 5% proportion of missings.

When the proportion of missings grows further on, INI becomes the only winner again, as can be seen from Table 5.5 presenting a typical pattern. Another feature of the pattern is that ILS, GZ and IMLS-4 perform similarly to the local versions, N-ILS and N-IMLS. As expected, the Mean imputation is the worst method.

5.5.2 Experiments with Scaled NetLab Gaussian Mixture Data Model

The 5-mixture data sets, with the dimension of the PPCA subspace equal to [n/2], were generated ten times with random sizes (from 200-250 rows and 15-25 columns) altogether with ten missing patterns using similar level of missings as implemented in previous experiments. Thus, there are 60 missing complete random patterns participated in the experiments.

Methods	*Proportion of Missings*					
	1%	5%	10%	15%	20%	25%
ILS	16.75 (7.52)	17.80 (4.86)	17.57 (3.48)	18.92 (4.38)	20.17 (5.03)	21.65 (5.26)
GZ	16.75 (7.52)	17.80 (4.86)	17.57 (3.48)	18.92 (4.37)	20.17 (5.02)	21.67 (5.27)
NIPALS	62.37 (17.41)	63.99 (12.42)	62.52 (10.26)	63.32 (10.64)	63.38 (10.46)	64.19 (11.12)
IMLS-1	62.30 (17.47)	63.95 (12.31)	62.76 (10.72)	63.41 (10.74)	63.43 (10.49)	64.15 (10.97)
IMLS-4	16.79 (7.49)	17.83 (5.00)	17.84 (4.05)	18.94 (4.36)	20.34 (5.25)	21.65 (5.17)
Mean	90.46 (11.36)	91.26 (5.87)	89.77 (6.51)	89.98 (6.22)	89.92 (6.06)	89.61 (6.08)
N-ILS	7.31 (3.39)	7.68 (1.90)	7.49 (1.41)	7.64 (1.36)	N/A	N/A
N-IMLS	7.30 (3.37)	7.67 (1.89)	7.48 (1.40)	7.63 (1.35)	7.95 (1.39)	8.73 (1.55)
INI	7.47 (3.19)	7.82 (2.08)	7.67 (1.38)	8.05 (1.21)	8.85 (1.80)	9.74 (1.90)
N-Mean	14.50 (6.77)	35.75 (8.71)	63.49 (13.02)	83.67 (18.30)	91.11 (14.17)	97.54 (18.12)

Table 5.6: The average squared error of imputation and its standard deviation (%) at scaled NetLab Gaussian 5-mixture data model with different levels of random missing entries.

Rather unexpectedly, the errors of imputation of all methods, except Mean, appear to be much lower than those found with the original NetLab Gaussian mixture model as can be seen from Table 5.6 versus Table 5.3. Furthermore, the increase of the error of imputation along the growth of the proportion of missings here, though can be observed as a trend, is not that dramatic as in the case of the original Netlab data model.

The two NN-based local least squares methods, N-ILS and N-IMLS, are the best according to Table 5.6 with INI a very close runner up. For N-ILS algorithm, the label “N/A” is applied to show that it does not always converge. However, N-ILS outperforms the others when it converges.

Methods of Imputation	1%				5%				10%			
	1	2	3	4	1	2	3	4	1	2	3	4
ILS	90	90	100	60	100	100	100	0	100	100	100	0
GZ	90	90	100	60	100	100	100	0	100	100	100	0
NIPALS	100	100	100	100	100	100	100	100	100	100	100	50
IMLS-1	100	100	100	100	100	100	100	100	100	100	100	50
IMLS-4	90	90	100	100	100	100	100	0	100	100	100	0
Mean	100	100	100	100	100	100	100	100	100	100	100	90
N-ILS	-	60	50	0	-	60	60	0	-	70	80	0
N-IMLS	40	-	50	0	40	-	60	0	30	-	80	0
INI	50	50	-	10	40	40	-	0	20	20	-	0
N-Mean	100	100	90	-	100	100	100	-	100	100	100	-

Table 5.7: The pair-wise comparison of 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 1%, 5% and 10% random missing data where 1,2,3 and 4 denote N-ILS, N-IMLS, INI and N-Mean, respectively.

In the perspective of pair-wise comparison, the results can be divided in two broad categories: (1) at level 1%-10% shown in Table 5.7 and (2) at level 15%-25% shown in Table 5.8. In the former category INI can be claimed the winner, especially at higher levels of missings, followed by N-IMLS and N-ILS.

At the level of 15% missings and higher, N-IMLS turns out the only one winner (see Table 5.8). INI and N-ILS follow it closely. N-Mean loses here; at higher missings, it loses even to its global version, Mean.

Methods of Imputation	15%				20%				25%			
	1	2	3	4	1	2	3	4	1	2	3	4
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	-

Table 5.8: The pair-wise comparison of 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 where where 1,2,3 and 4 denote N-ILS, N-IMLS, INI and N-Mean, respectively.

5.5.3 Summary of the Results

According to the results of the experiment on the NetLab Gaussian 5-mixure, INI is consistently the best method. It is followed by N-IMLS, the nearest-neighbour version of IMLS, as the second winner. Thus, under this simple Gaussian mixture structure, the combination of a general form of IMLS, IMLS-4, and its nearest neighbour version, is the best method.

Finally, for more complex structure of data sets generated with the scaled NetLab Gaussian 5-mixture, the results are varied according to the level of missings. At levels 1%-10%, INI surpasses the other methods. In the close range, N-ILS and N-IMLS, appear as second best methods. As the level of missings increases to 15%-25%, N-IMLS comes up as the best method. It is followed by INI and N-ILS as the second winners. Also, at this level of missings, Mean imputation beats its nearest neighbour versions, N-Mean.

Also, the scaled NetLab Data Model leads to much smaller errors in the least-squares methods, which probably can be attributed to the fact that the data are spread differently at different directions with the scaled model which conforms to the one-by-one factor extraction procedure underlying the methods.

5.6 Results of the Experimental Study on Different Missing Patterns

In this experiment three missing patterns as described in Section 5.3 will be employed on both NetLab Gaussian mixture and its scaled versions. Both data model generations use 5-mixture in this experiment. Also, as described in previous experiment (see Section 5.5.2), the statistical values of error of imputation of both data model generations to be represented in different way.

If some methods occasionally do not converge, they will be labeled as "N/A". On occasion one or more methods cannot be proceed they are denoted as "NN".

5.6.1 Inherited Pattern

The performances of ten algorithms on two types of Gaussian mixture data model with Inherited missings pattern are studied. The results will be presented according to the data model generation.

Netlab Gaussian Mixture Data Model

For each of the ten data sets generated according to the 5-mixture original Netlab Data Model, ten Inherited missing patterns have been generated according to the algorithm described in section 5.3.2. All Inherited missing patterns were based on the six levels of missings from 25% to 1%. The average errors of the ten selected algorithms are shown in Table 5.9 and pair-wise comparison in Tables 5.10 and 5.11.

According to Table 5.9 the errors of all methods, except Mean which is the worst

Methods	*Proportion of Missings*					
	1%	5%	10%	15%	20%	25%
ILS	31.38 (11.30)	34.39 (5.91)	36.90 (4.64)	41.14 (7.35)	43.93 (7.98)	48.82 (10.47)
GZ	31.38 (11.30)	34.40 (5.91)	36.90 (4.64)	41.13 (7.35)	43.97 (8.06)	48.80 (10.47)
NIPALS	42.33 (13.77)	44.94 (7.91)	47.06 (6.13)	50.58 (8.01)	52.45 (8.57)	55.91 (10.82)
IMLS-1	42.38 (13.80)	44.84 (7.80)	46.84 (6.05)	49.08 (5.81)	50.60 (5.77)	51.93 (5.47)
IMLS-4	31.36 (11.54)	33.90 (5.83)	36.72 (4.92)	39.05 (4.16)	41.20 (4.19)	43.87 (6.40)
Mean	96.53 (6.38)	96.48 (2.94)	96.27 (2.28)	96.14 (1.93)	96.06 (1.77)	96.04 (1.62)
N-ILS	36.64 (18.62)	150 (1100)	N/A	N/A	N/A	N/A
N-IMLS	26.82 (9.89)	30.40 (8.64)	35.67 (11.12)	42.54 (10.45)	52.93 (13.79)	66.48 (20.35)
INI	25.95 (9.22)	28.68 (5.10)	31.86 (4.72)	35.85 (6.41)	39.01 (7.17)	44.29 (10.55)
N-Mean	29.04 (8.80)	37.38 (5.58)	48.17 (5.88)	59.59 (7.26)	69.39 (7.50)	79.95 (8.52)

Table 5.9: The average squared error of imputation and its standard deviation (%) at NetLab Gaussian 5-mixture data with different levels of Inherited missing entries.

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

Table 5.10: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on NetLab Gaussian 5-mixtures with $[n-3]$ PPCA factors for 1% Inherited missing data.

anyway, grow as the percentage of missings grows. Once again INI wins except at the level of 25% missings at which the slow increase of errors in IMLS-4 wins over a faster increase in INI's errors. Moreover, with the Inherited missing pattern, global least squares outperform their local versions, N-IMLS and N-ILS.

Looking at the pair-wise comparison results, we see that at 1% missings, INI is the only winner (see Table 5.10). It is followed by the local least squares methods N-ILS and N-IMLS. The local version of Mean, N-Mean, is the fourth winner. In general, at 1% missings, the local versions of imputation techniques surpass their

Methods of Imputation	5%					15%					25%				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
ILS	70	80	80	90	30	70	30	50	100	0	80	0	0	90	0
GZ	70	80	80	90	30	70	30	50	100	0	80	0	0	90	0
NIPALS	90	90	100	100	80	100	50	80	100	10	90	0	50	90	0
IMLS-1	90	80	100	100	80	100	50	80	100	10	90	0	30	90	0
IMLS-4	-	80	80	90	30	-	30	50	100	0	-	0	10	80	100
Mean	100	100	100	100	100	100	60	100	100	100	100	10	100	100	100
N-ILS	20	-	80	90	20	70	-	90	100	40	100	-	100	100	90
N-IMLS	20	20	-	100	20	50	10	-	100	0	90	0	-	90	10
INI	10	10	0	-	0	0	0	0	-	0	20	0	10	-	0
N-Mean	70	80	80	100	-	100	60	100	100	-	100	10	90	100	-

Table 5.11: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on NetLab Gaussian 5-mixtures with $[n-3]$ PPCA factors for 5%, 15% and 25% Inherited missing data where 1, 2, 3, 4 and 5 denote IMLS-4, N-ILS, N-IMLS, INI, and N-Mean, respectively.

global versions.

INI remains the only winner at higher levels of missings according to Table 5.11. However this time N-Mean loses to the least squares global techniques IMLS-4, ILS and GZ. Moreover, IMLS-4 becomes the second best when the percentage of missings grows to 15% and higher. N-ILS totally drops off at 25 % of missings because of a poor convergence rate.

Scaled Netlab Gaussian Mixture Data Model

Table 5.12 shows the average square errors of imputations in the experiments with the Inherited missings pattern with the data generated according to the scaled NetLab Gaussian 5-mixture data model with the dimension of PPCA space equal to $[n/2]$.

The average errors of all methods except for the one-dimensional NIPALS, IMLS-1 and Mean are much smaller than with data generated according to the original Netlab model. Table 5.12 shows three obvious winners, the NN based least squares

Methods	*Proportion of Missings*					
	1%	5%	10%	15%	20%	25%
ILS	16.74 (6.98)	16.50 (4.25)	17.53 (4.23)	18.70 (4.58)	20.03 (5.06)	21.81 (5.83)
GZ	16.74 (6.98)	16.50 (4.25)	17.53 (4.23)	18.70 (4.58)	20.04 (5.06)	21.78 (5.80)
NIPALS	62.82 (17.61)	61.94 (12.04)	62.52 (11.55)	62.62 (10.96)	62.93 (10.85)	63.78 (11.05)
IMLS-1	62.93 (17.81)	62.07 (12.17)	62.53 (11.53)	62.71 (11.12)	62.93 (10.82)	63.69 (10.81)
IMLS-4	16.79 (7.08)	16.58 (4.33)	17.54 (4.20)	18.70 (4.56)	20.22 (4.99)	21.67 (5.53)
Mean	90.46 (9.94)	89.93 (7.37)	89.95 (6.83)	89.93 (6.44)	89.99 (6.10)	89.94 (6.00)
N-ILS	7.79 (3.05)	7.29 (1.64)	7.39 (1.25)	7.55 (1.23)	N/A	N/A
N-IMLS	7.78 (3.05)	7.29 (1.64)	7.38 (1.25)	7.54 (1.23)	7.93 (1.31)	8.74 (1.66)
INI	7.84 (3.10)	7.33 (1.58)	7.59 (1.23)	8.01 (1.26)	8.81 (1.49)	9.85 (2.15)
N-Mean	15.33 (6.18)	36.77 (9.96)	62.35 (13.43)	82.26 (16.87)	91.03 (16.47)	97.63 (16.57)

Table 5.12: The average squared error of imputation and its standard deviation (%) at scaled NetLab Gaussian 5-mixture data with different levels of Inherited missing entries.

methods, with N-IMLS leading and N-ILS and INI following; at higher missing proportions of 20% and 25%, N-ILS does not always converge, though it performs quite well when converges.

These conclusions can be detailed with Table 5.13 presenting the results of pair-wise comparison between the methods. N-ILS is the best at 1% giving way to N-IMLS at 10% and 20%. INI loses only to these two methods.

Methods of Imputation	1%				10%				20%			
	1	2	3	4	1	2	3	4	1	2	3	4
ILS	100	100	100	70	100	100	100	0	100	100	100	0
GZ	100	100	100	70	100	100	100	0	100	100	100	0
NIPALS	100	100	100	100	100	100	100	50	100	100	100	0
IMLS-1	100	100	100	100	100	100	100	50	100	100	100	0
IMLS-4	100	100	100	70	100	100	100	0	100	100	100	0
Mean	100	100	100	100	100	100	100	100	100	100	100	30
N-ILS	-	40	40	20	-	70	20	0	-	60	30	0
N-IMLS	60	-	40	20	30	-	20	0	40	-	30	0
INI	60	60	-	20	80	80	-	0	70	70	-	0
N-Mean	80	80	80	-	100	100	100	-	100	100	100	-

Table 5.13: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on scaled NetLab Gaussian 5-mixtures with [n/2] PPCA factors for 1%, 10% and 20% Inherited missing data where 1, 2, 3 and 4 denote N-ILS, N-IMLS, INI and N-Mean, respectively.

Overall, at Inherited random missings, the three NN-based least squares techniques remain winners. The global-local INI dominates the imputation contest with the original Netlab Data Model and it loses to N-IMLS and N-ILS with the scaled Netlab Data Model. Once again the scaled Netlab Data Model leads to much smaller errors in the least squares methods, probably because of the same factor that the data are spread differently at different directions with the scaled model rather than with the original Netlab model, which conforms to the iterative extracting of factors underlying the methods.

5.6.2 Sensitive Issue Pattern

The experiments were conducted according to the scenario introduced in Section 5.3.3. As in the previous experiments, the results will be exposed sequentially according to the NetLab Gaussian mixture and its scaled versions in turn.

NetLab Gaussian Mixture Data Model

The results of experiments on the original NetLab Gaussian mixture data model with the Sensitive issue pattern are summarized in Table 5.14. We limited the span of missings to 10% here because the missing entries are now confined within a relatively small submatrix of the data matrix.

Amazingly, with this missing pattern the error of imputation does not monotonely follow the growth of the numbers of missing entries. All least squares based methods perform at 5% missings better than at 1%.

On the level of average errors, NN based least squares methods N-IMLS, N-ILS and INI surpass the other methods, and no obvious winner can be chosen among them.

Methods	*Proportion of Missings*		
	1%	5%	10%
ILS	44.97 (19.94)	31.55 (3.33)	N/A (N/A)
GZ	44.97 (19.94)	31.55 (3.33)	35.59 (6.40)
NIPALS	54.15(20.04)	39.22 (4.57)	42.76 (9.80)
IMLS-1	54.19 (20.11)	39.33 (4.55)	42.86 (9.73)
IMLS-4	44.60 (21.04)	31.79 (3.21)	57.58 (69.91)
Mean	92.78 (7.08)	96.31 (4.14)	96.67 (3.66)
N-ILS	37.57 (19.42)	27.31 (3.44)	38.11 (12.44)
N-IMLS	37.45 (19.31)	27.36 (3.35)	33.17(5.25)
INI	39.27 (20.03)	26.57 (3.63)	37.02(15.18)
N-Mean	42.25 (20.23)	57.34 (12.05)	93.26(27.13)

Table 5.14: The average squared error of imputation and its standard deviation (%) NetLab Gaussian 5-mixture data with different levels of missings from sensitive issue pattern.

Methods of Imputation	1%				5%				10%			
	1	2	3	4	1	2	3	4	1	2	3	4
ILS	80	80	70	50	100	100	90	0	70	80	90	10
GZ	80	80	70	50	100	100	90	0	70	80	80	0
NIPALS	100	100	100	100	100	100	100	10	80	100	90	0
IMLS-1	100	100	100	100	100	100	100	10	80	100	90	0
IMLS-4	80	80	70	50	100	100	90	10	70	80	80	10
Mean	100	100	100	100	100	100	100	100	100	100	100	90
N-ILS	-	80	30	30	-	70	50	0	-	100	80	0
N-IMLS	20	-	30	30	30	-	50	0	0	-	60	0
INI	70	70	-	20	50	50	-	0	20	40	-	0
N-Mean	70	70	80	-	100	100	100	-	100	100	100	-

Table 5.15: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on NetLab Gaussian 5-mixtures with [n-3] PPCA factors for 1%, 5% and 10% missings from sensitive issue pattern where 1, 2, 3 and 4 denote N-ILS, N-IMLS, INI and N-Mean, respectively.

With the pair-wise comparison presented in Table 5.15, N-IMLS being the winner at little missings is substituted by INI when the proportion of missings increases to 5% and, especially, 10%.

Scaled Netlab Gaussian Mixture Data Model

The performance of the ten algorithms on the scaled Netlab Gaussian 5-mixture Data Model with Sensitive issue pattern missings are shown in Table 5.16.

Here the errors grow indeed when the proportion of missing entries increases.

The errors of all methods are smaller at this data type again, except for those of Mean and unidimensional NIPALS and IMLS-1.

Methods	*Proportion of Missings*		
	1%	5%	10%
ILS	15.87 (6.13)	17.47 (5.98)	25.64 (10.87)
GZ	15.86 (6.13)	17.47 (5.98)	25.63 (10.86)
NIPALS	68.47(16.16)	60.26 (14.76)	64.71 (13.41)
IMLS-1	68.70 (16.24)	60.36 (14.54)	64.60 (13.30)
IMLS-4	16.08 (6.36)	17.52 (5.65)	25.65 (10.31)
Mean	95.43 (9.04)	91.77 (7.49)	91.98 (8.25)
N-ILS	7.15 (2.94)	6.71 (1.75)	7.28 (1.87)
N-IMLS	7.15 (2.92)	6.70 (1.75)	7.28(1.85)
INI	6.83 (3.07)	7.59 (2.89)	11.39(7.96)
N-Mean	28.58 (12.99)	100.37 (28.97)	246.13(84.93)

Table 5.16: The average squared error of imputation and its standard deviation (%) scaled NetLab Gaussian 5-mixture data with different levels of sensitive issue pattern.

The local versions of the least squares imputation always surpass their global counterparts. Two local least squares techniques, N-ILS and N-IMLS, show quite low levels of errors, about 7% only, which is surpassed only once by INI's performance at 1% missings.

Method Mean outperforms N-Mean here at higher levels of missings, probably because it relies on more data with no missings at all at the Sensitive issue missing pattern.

On the level of pair-wise comparison presented in Table 5.17 method INI appears to be better than the others not only at 1% missings but also at 5%. It only loses to N-IMLS at 10% missings. Also, Mean beats N-Mean indeed at 5% and 10% missings.

As was the case with the other missing patterns, the three NN-based least squares techniques are obvious winners at the Sensitive issue random missings. The global-local INI dominates the imputation contest at little missing proportions with the

Methods of Imputation	1%				5%				10%			
	1	2	3	4	1	2	3	4	1	2	3	4
ILS	100	100	100	20	100	100	100	0	100	100	100	0
GZ	100	100	100	20	100	100	100	0	100	100	100	0
NIPALS	100	100	100	100	100	100	100	10	100	100	100	0
IMLS-1	100	100	100	100	100	100	100	10	100	100	100	0
IMLS-4	100	100	100	20	100	100	100	0	100	100	100	0
Mean	100	100	100	100	100	100	100	30	100	100	100	0
N-ILS	-	60	60	0	-	70	60	0	-	80	20	0
N-IMLS	40	-	60	0	30	-	60	0	20	-	20	0
INI	40	40	-	0	40	40	-	0	80	80	-	0
N-Mean	100	100	100	-	100	100	100	-	100	100	100	-

Table 5.17: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on scaled NetLab Gaussian 5-mixtures data with [n/2] PPCA factors for 1%, 5% and 10% missings from sensitive issue pattern where 1,2,3 and 4 denote N-ILS, N-IMLS, INI and N-Mean, respectively.

original Netlab Data Model and at higher levels of missings with the scaled Netlab Data Model. We cannot see explanation for such a rather strange behaviour. Once again the scaled Netlab Data Model leads to much smaller errors in the least squares methods except for unidimensional ones. Also, Mean outperforms N-Mean here at higher missing levels with the scaled Netlab Data Model.

5.6.3 Merged Database Pattern

At this section, two types of merged database pattern, missings from one database and two databases, will be explored. As usual, two types of the NetLab Gaussian mixture model were applied at each type of missings generation.

5.6.3.1 Missings from One Database

Netlab Gaussian Mixture Data Model

The average error results of experiments on the original NetLab Gaussian 5-mixture Data Model with the Merged database missing pattern at which missings come from only one database are presented in Table 5.18.

Imputation Methods	$q = 20\%$		$q = 30\%$	
	1%	5%	1%	5%
ILS	58.62 (32.56)	94.74 (358.51)	48.82(28.10)	57.19(27.68)
GZ	56.82 (32.56)	94.74 (358.72)	48.82(28.10)	57.36(27.76)
NIPALS	70.55 (32.62)	105.01 (358.17)	61.84(29.51)	67.83(29.05)
IMLS-1	70.55 (32.62)	71.76 (42.06)	62.00(29.43)	67.86(29.05)
IMLS-4	58.00 (33.47)	203.95 (1451.30)	49.03(27.71)	56.77(26.90)
Mean	93.55 (9.74)	95.59 (6.72)	94.38(9.72)	93.53(6.20)
N-ILS	49.84 (29.16)	68.05 (160.54)	43.17(27.57)	94.59(13.12)
N-IMLS	49.69 (29.10)	53.77 (34.10)	42.87(27.23)	49.06(24.27)
INI	48.45 (28.14)	55.39 (41.85)	41.54(25.07)	49.11(23.91)
N-Mean	74.57 (40.00)	75.93 (38.70)	90.66(48.33)	99.88(52.87)

Table 5.18: The average squared error of imputation and its standard deviation (%) NetLab Gaussian 5-mixture data with different levels missings entries from one database where q denotes the proportion of column number which contains missings.

The denotation q refers to the proportion of columns that are absent from the 'incomplete' database as explained in section 5.3.4.1. In general, two winning methods here are INI and N-IMLS. The errors are somewhat less at q=30%, probably because there are relatively less rows containing missings entries in this case than at q=20%.

Methods of Imputation	1%					5%				
	1	2	3	4	5	1	2	3	4	5
ILS	60	70	70	80	20	40	90	90	90	0
GZ	60	70	70	80	20	40	90	90	90	0
NIPALS	80	80	80	90	30	100	100	100	100	30
IMLS-1	80	80	80	90	30	100	100	100	100	30
IMLS-4	-	70	70	80	20	-	90	90	90	10
Mean	80	70	70	80	40	100	100	100	100	70
N-ILS	30	-	60	70	10	10	-	100	50	0
N-IMLS	30	40	-	70	0	10	0	-	50	0
INI	20	30	30	-	0	10	50	50	-	0
N-Mean	80	90	100	100	-	90	100	100	100	-

Table 5.19: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on NetLab Gaussian 5-mixtures with [n-3] PPCA factors at 1% and 5% missings and 20% proportion of column number missings where 1,2,3,4 and 5 denote IMLS-4, N-ILS, N-IMLS, INI and N-Mean, respectively.

All methods, N-ILS and ILS included, converge here, probably because of smaller proportions of the overall missings. Considering pair-wise comparison of the methods presented in Tables 5.19 and 5.20 lead us to see that INI is the best. Altogether, NN based least squares methods beat their global counterparts while N-Mean loses to Mean at 5% missings.

Methods of Imputation	1%					5%				
	1	2	3	4	5	1	2	3	4	5
ILS	20	70	70	100	0	20	80	80	90	10
GZ	20	70	70	100	0	20	80	80	90	10
NIPALS	100	100	100	100	10	90	100	100	100	20
IMLS-1	100	100	100	100	10	90	100	100	100	30
IMLS-4	-	80	80	100	10	-	80	80	90	10
Mean	100	100	100	100	60	80	100	100	100	30
N-ILS	20	-	90	60	0	20	-	90	80	0
N-IMLS	20	10	-	50	0	20	10	-	80	0
INI	0	40	50	-	0	10	20	20	-	0
N-Mean	90	100	100	100	-	90	100	100	100	-

Table 5.20: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on NetLab Gaussian 5-mixtures with [n-3] PPCA factors at 1% and 5% missings and 30% proportion of column number missings where 1,2,3,4 and 5 denote IMLS-4, N-ILS, N-IMLS, INI and N-Mean, respectively.

Scaled Netlab Gaussian Mixture Data Model

The summary of the average error results on the scaled NetLab Gaussian 5-mixture data model with missings from one of the databases is shown in Table 5.21. Here, the difference in q values bears no influence on the errors, in contrast to the case of the original Netlab data model probably because the set of entities is much more diversified in this case.

This time, the three NN based least squares techniques are the best, with N-IMLS showing slightly better results and INI trailing behind very closely.

On the level of pair-wise comparison for q=20%, however, the results are more in favour of INI (see Table 5.22): INI obviously outperforms the others at 1% missings

Methods	$q = 20\%$		$q = 30\%$	
	1%	5%	1%	5%
ILS	19.19 (11.64)	18.74 (10.59)	21.48 (13.41)	20.44 (8.46)
GZ	19.19 (11.65)	18.74 (10.59)	21.48 (13.41)	20.33 (8.09)
NIPALS	64.45 (24.15)	64.20 (19.84)	69.70 (28.11)	63.70 (19.84)
IMLS-1	64.35 (24.38)	64.11 (19.77)	69.64 (27.84)	63.73 (20.02)
IMLS-4	19.01 (11.15)	18.50 (9.95)	21.42 (13.40)	20.56 (8.92)
Mean	89.97 (12.80)	88.88 (11.77)	90.48 (11.35)	90.67 (8.08)
N-ILS	8.33 (6.55)	7.56 (4.13)	8.92 (5.51)	7.75 (3.47)
N-IMLS	8.32 (6.54)	7.54 (4.12)	8.90 (5.50)	7.72 (3.47)
INI	8.95 (6.40)	8.33 (5.67)	10.31 (9.59)	8.75 (3.75)
N-Mean	88.82 (62.93)	88.38 (51.30)	162.37 (94.23)	152.03 (72.96)

Table 5.21: The average squared error of imputation and its standard deviation (%) scaled NetLab Gaussian 5-mixture data with different levels missings entries from one database where q denotes the proportion of column number which contains missings.

and ties up with N-IMLS at 5%. Once again, N-Mean is beaten by its global counterpart, Mean.

Methods of Imputation	1%					5%				
	1	2	3	4	5	1	2	3	4	5
ILS	50	100	100	100	0	80	90	90	90	0
GZ	50	100	100	100	0	80	90	90	90	0
NIPALS	100	100	100	100	40	100	100	100	100	30
IMLS-1	100	100	100	100	40	100	100	100	100	30
IMLS-4	-	100	100	100	0	-	90	90	90	0
Mean	100	100	100	100	80	100	100	100	100	70
N-ILS	0	-	60	60	0	10	-	80	50	0
N-IMLS	0	40	-	60	0	10	20	-	50	0
INI	0	40	40	-	0	10	50	50	-	0
N-Mean	100	100	100	100	-	100	60	100	100	-

Table 5.22: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on scaled NetLab Gaussian 5-mixtures with [n/2] PPCA factors at 1% and 5% missings and 20% proportion of column number missings where 1,2,3,4 and 5 denote IMLS-4, N-ILS, N-IMLS, INI and N-Mean, respectively.

The results quite differ though at q=30% (see Table 5.23). This time, N-ILS takes the lead at 1% missings giving way to N-IMLS at 5%. Also, N-Mean beats Mean here. In general, the NN based least squares techniques appear the best at the Merged database with missings coming from one of the databases. INI performs better at 1% missings while N-IMLS is better at 5%.

Methods of Imputation	1%					5%				
	1	2	3	4	5	1	2	3	4	5
ILS	60	100	100	100	0	30	100	100	100	0
GZ	60	100	100	100	0	30	100	100	100	0
NIPALS	100	100	100	100	10	100	100	100	100	0
IMLS-1	100	100	100	100	10	100	100	100	100	0
IMLS-4	-	100	100	100	0	-	100	100	100	0
Mean	100	100	100	100	20	100	100	100	100	0
N-ILS	0	-	30	30	0	0	-	60	30	0
N-IMLS	0	70	-	30	0	0	40	-	30	0
INI	0	70	70	-	0	0	70	70	-	0
N-Mean	100	80	100	100	-	100	100	100	100	-

Table 5.23: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on scaled NetLab Gaussian 5-mixtures with [n/2] PPCA factors at 1% and 5% missings and 30% proportion of column number missings where 1, 2, 3, 4 and 5 denote IMLS-4, N-ILS, N-IMLS, INI and N-Mean, respectively.

5.6.3.2 Missings from Two Databases

Netlab Gaussian Mixture Data Model

The performance of algorithms on the NetLab Gaussian 5-mixture data model with missings from two databases is shown in Table 5.24 and Table 5.25. Interestingly, the other NN based methods, N-Mean included, are the best here. At higher missings the error drastically increases. This conclusion is supported by the pair-wise comparison presented in Table 5.25.

With this pattern of missings all ILS-like methods may not converge at 5% missings. Moreover, N-ILS cannot proceed at all because of missing values occurring in a whole column of the NN matrix, which is denoted by NN in Table 5.25.

Methods	*Proportion of Missings*	
	1%	5%
ILS	65.48 (100.00)	55.60 (29.34)($*$)
GZ	65.44 (103.36)	55.53 (29.87)($*$)
NIPALS	78.51(55.67)	94.83 (114.85)($*$)
IMLS-1	77.81 (55.22)	80.20 (61.56)
IMLS-4	68.24 (104.96)	84.00 (64.96)
Mean	143.31 (80.25)	128.36 (37.14)
N-ILS	NN	NN
N-IMLS	26.00 (28.45)	69.62 (12.85)
INI	32.45 (30.77)	71.10 (11.84)
N-Mean	31.29 (35.50)	70.43 (39.81)

Table 5.24: The average squared error of imputation and its standard deviation (%) NetLab Gaussian 5-mixture data model with different levels of missings from two databases where ($*$) and NN denote taken only of the converged entries and cannot proceed, respectively.

Methods of Imputation	1%					5%				
	1	2	3	4	5	1	2	3	4	5
ILS	40	30	80	70	70	60	80	70	60	80
GZ	40	30	80	70	70	60	80	60	60	80
NIPALS	60	60	80	80	80	80	40	60	50	50
IMLS-1	-	60	80	80	80	-	40	60	50	50
IMLS-4	40	-	80	70	70	60	-	40	40	80
Mean	60	90	100	100	90	90	90	100	100	90
N-IMLS	20	20	-	20	0	40	60	-	20	60
INI	20	30	80	-	20	50	60	80	-	60
N-Mean	20	30	100	80	-	50	20	40	40	-

Table 5.25: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on NetLab Gaussian 5-mixture with [n-3] PPCA factors at 1% and 5% missings from two databases where 1, 2, 3, 4 and 5 denote IMLS-1, IMLS-4, N-IMLS, INI and N-Mean, respectively.

Scaled Netlab Gaussian Mixture Data Model

The results of experiments are shown in Table 5.26. The scaled data model with the Merged database missings coming from both databases leads to the global least squares techniques, except for the frequently nonconvergent ILS, to win.

This probably can be explained by the greater spread of data at this model to cover the fact that entire subtables are missing in this model. Especially intriguing is the fact that one-dimesional methods NIPALS and IMLS-1 win at 5% missings over their four-dimensional analogues. These findings are supported by the results of pair-wise comparison in Table 5.27.

Methods	*Proportion of Missings*	
	1%	5%
ILS	12.47 (8.38)	N/A
GZ	12.42 (8.33)	18.84 (8.13)
NIPALS	12.96(6.66)	16.07 (9.52)
IMLS-1	12.87 (6.63)	15.67 (9.27)
IMLS-4	12.45 (8.60)	24.83 (21.13)
Mean	100.25 (3.80)	103.88 (3.75)
N-ILS	N/A	NN (∗)
N-IMLS	20.68 (13.86)	49.45 (11.96)
INI	17.85 (11.18)	45.06 (13.14)
N-Mean	14.58 (8.00)	30.67 (7.93)

Table 5.26: The average squared error of imputation and its standard deviation (%) scaled NetLab Gaussian 5-mixture data with different levels missings entries from two databases where ∗ denotes cannot proceed.

The NN imputation techniques, including INI, show rather poor performances (see Table 5.27).

Methods of Imputation	1%					5%				
	1	2	3	4	5	1	2	3	4	5
ILS	-	30	50	30	30	-	100	70	0	10
GZ	80	30	50	30	30	60	100	70	0	10
NIPALS	70	60	70	20	60	10	80	30	0	10
IMLS-1	70	-	70	20	60	0	-	30	0	10
IMLS-4	50	30	-	20	40	30	70	-	10	20
Mean	100	100	100	100	100	100	100	100	100	100
N-IMLS	80	60	90	70	70	100	100	90	90	100
INI	70	80	80	-	60	100	100	90	-	100
N-Mean	70	40	60	40	-	90	90	80	0	-

Table 5.27: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on scaled NetLab Gaussian 5-mixtures with [n/2] PPCA factors for 1% and 5% missings from two databases where 1,2,3,4 and 5 denote ILS, IMLS-1, IMLS-4, INI and N-Mean.

5.6.4 Summary of the Results

The results of experiments show that the performances of ten algorithms are varied according to the type of data model and level of missings which can be summarized as follows.

In Inherited missings, with the NetLab Gaussian mixture data model, at all levels of missings, INI is consistently the best method. In contrast, at the scaled NetLab Gaussian mixture data model, N-IMLS surpasses INI and comes up as the winner.

In the experiments with the sensitive, the results show that Issue pattern mechanism are accomplished. The results show that N-IMLS surpasses the other methods at the level 1% missings with NetLab Gaussian data model. However, as the level of missings increases to 5%, N-IMLS and INI provide almost similar performance. Finally, at the level 10%, INI appears as the only one winner. In contrast, with the scaled NetLab Gaussian mixture data model, at the level 1% missings, INI surpasses other methods. However, as the level of missings increases to 5% and 10%, N-IMLS consistently to be the best method.

The results of experiments with two types of Merged database pattern are summarized as follows. In the case of missings coming from one data base on the NetLab Gaussian mixture data model, overall, INI is the best method. In the other case, with the scaled NetLab Gaussian mixture, the results are varied according to the proportion of columns which contain missing values. At the proportion 20%, INI and N-IMLS, provide almost equal performances. As the proportion grows to 30%, N-IMLS comes up as the only one winner.

With the missings from two databases on NetLab Gaussian mixture, all nearest neighbour versions of least squares, including N-Mean, surpass the other methods. In contrast, the results of experiments with the scaled NetLab Gaussian mixture data model show that the ordinary least squares, ILS and IMLS, come up as the best methods. In either case, the nearest neighbours of the least squares imputation show very poor performance, which can be probably be explained by the fact that, at this missing model, the nearest neighbours that have no missings are rather distant indeed.

Chapter 6

Other Data Models

According to the experimental study on Gaussian mixture distributions with Complete random missing pattern, the local versions of LS always outperform their global approaches. However, different results might be produced if the data sets are generated with different data model that may less conform to the nature of the local versions of the least squares imputation approaches. The experimental study also shows that the global-local LS imputation, INI, on the Complete random missing pattern, almost always outperforms the other methods. This results lead us to consider INI as a good to be tried on real-world missing data problems and compared with the maximum likelihood based approaches: EM and MI. Based on the this considerations, this chapter explores experimentally performances of the least squares imputation for handling incomplete entries on different data models: rank one and a real-world marketing data set. For benchmarking purposes, two versions of EM imputation and multiple imputation (MI) are participated in the experiments.

The goal of this experimental study is twofold:

1. To see if there is a data model at which the global LS imputation techniques are better than their nearest neighbour versions.

2. To compare the performance of the global-local LS imputation with available as machine codes.

6.1 Least Squares Imputation Experiments with Rank One Data Model

6.1.1 Selection of Algorithms

1. ILS-NIPALS 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 ILS with $p = 1$.

7. N-IMLS: NN based IMLS-1.

8. INI: NN based IMLS-1 imputation based on distances from an IMLS-4 imputation.

9. Mean imputation.

10. N-Mean: NN based Mean imputation.

In the follow-up experiments, the NN based techniques will operate with K=10.

6.1.2 Generation of Data

Under this data model generation some vectors, say $\mathbf{c}_{(15\times 1)}$, $\mathbf{z}_{(200\times 1)}$ with their components in the range between -1 and 1 to be specified, and generate a uniformly random matrix $\mathbf{E}_{(200\times 15)}$ within the same range. Then the data model can be formulated as follows:

$$\mathbf{X}_{\epsilon} = \mathbf{z} * \mathbf{c}' + \epsilon E \tag{6.1.1}$$

where coefficient ϵ scales the random noise $\mathbf{E}$ added to the onedimensional matrix $\mathbf{z} * \mathbf{c}'$. The coefficient ϵ will be referred to as the noise level; it has been taken at 6 levels from $\epsilon = 0.1$ to $\epsilon = 0.6$.

Model (6.1.1) can be applied as many times as a data set is needed.

6.1.3 Mechanisms for Missing Data

The complete random missing pattern matrix is generated with the proportion's range at 1%, 5%, 10%, 15%, 20% and 25% of the total number of entries.

6.1.4 Evaluation of Results

The results of experiments with regard the error of imputation is computed according to the (5.4.3).

6.1.5 Results

Table 6.1 shows the average results of 30 experiments (five data sets times six missings patterns) with the selected algorithms for each noise level. Both ILS and GZ frequently do not converge, probably because of too many factors, four, required in them. This is labeled by the symbol 'N/A' put in the corresponding cells of the Table 1. Table 6.1 shows that, in each of the methods, except the Mean, the error increases with the noise level growing. At the Mean, the error stands constant on the level of about 100 %. Two obvious winners, according to the Table, are NIPALS (that is, ILS-1) and IMLS-1. The other methods' performances are much worse. This, probably, can be explained by the very nature of the data generated: unidimensionality. The other methods seem just overspecified and thus wrong in this situation.

Methods	*Noise Level*					
	0.1	0.2	0.3	0.4	0.5	0.6
ILS	33.45 (179.42)	110.38 (478.79)	235.06 (928.57)	N/A	N/A	N/A
GZ	N/A	N/A	282.97 (1042)	359.19 (1416)	303.12 (1413)	260.17 (1180)
NIPALS	3.44 (0.58)	12.67 (2.17)	25.09 (4.44)	38.12 (6.98)	50.12 (9.33)	60.41 (11.14)
IMLS-1	3.44 (0.58)	12.67 (2.17)	25.07 (4.44)	38.09 (6.98)	50.08 (9.32)	60.35 (11.13)
IMLS-4	34.37 (82.54)	17.69 (3.02)	34.92 (6.26)	53.18 (9.76)	69.70 (12.84)	84.10 (15.40)
Mean	100.77 (1.21)	100.73 (1.21)	100.69 (1.23)	100.65 (1.25)	100.61 (1.29)	100.59 (1.32)
N-ILS	N/A	N/A	N/A	N/A	N/A	N/A
N-IMLS	13.38 (23.20)	20.61 (13.21)	43.76 (36.70)	61.29 (31.05)	84.89 (36.84)	103.56 (43.02)
INI	33.84 (83.45)	28.08 (42.19)	42.26 (55.89)	53.55 (34.84)	63.86 (16.39)	78.75 (21.27)
N-Mean	74.30 (45.43)	77.55 (41.32)	82.08 (35.89)	87.53 (30.95)	92.57 (27.36)	97.83 (24.53)

Table 6.1: The average squared errors of imputation (in %) for different methods (in rows) at different noise levels, columns; the values in parentheses are corresponding standard deviations, per cent as well.

To remove the effect of failing convergences and, moreover, the effect of overlapping dispersions in performances of the methods, the pairwise comparison is called for. That is, for each pair of methods, the number of times at which one of them outperformed the other will be counted. These data are shown in Table 6.2 an (i, j) entry in which shows how many times, per cent, the method in j-th column outperformed the method in i-th row.

The data in Table 6.2 confirm the results in Table 6.1. Moreover, the IMLS-1 was the winner more often than NIPALS (76% to 24%). Also, method INI gets noted as the third ranking winner, which should be attributed to the fact that it heavily relies on IMLS-1.

	ILS	GZ	NIPALS	IMLS-1	IMLS-4	Mean	N-ILS	N-IMLS	INI	N-Mean
ILS	-	44.00	100.00	100.00	74.00	1.50	32.00	80.00	98.00	20.00
GZ	56.00	-	100.00	100.00	77.00	1.50	33.00	88.00	98.00	17.00
NIPALS	0.00	0.00	-	76.00	0.00	0.00	0.00	0.00	0.00	0.00
IMLS-1	0.00	0.00	24.00	-	0.00	0.00	0.00	0.00	0.00	0.00
IMLS-4	26.00	23.00	100.00	100.00	-	0.00	21.00	67.00	97.50	0.00
Mean	98.50	98.50	100.00	100.00	100.00	-	52.00	99.50	100.00	77.00
N-ILS	68.00	67.00	100.00	100.00	79.00	48.00	-	94.00	100.00	67.00
N-IMLS	20.00	12.00	100.00	100.00	33.00	0.50	6.00	-	92.00	3.00
INI	2.00	2.00	100.00	100.00	2.50	0.00	0.00	8.00	-	0.00
N-Mean	80.00	83.00	100.00	100.00	100.00	23.00	33.00	97.00	100.00	-

Table 6.2: The pair-wise comparison between 10 methods; an entry (i, j) shows how many times, per cent, method j outperformed method i with the rank one data model for all noise level.

6.1.6 Structural Complexity of Data Sets

As seen from the results of the experimental study on the Complete random missing pattern using Gaussian mixture data models, the performance of NN versions of the least squares imputation methods always surpasses the global least squares approaches. In contrast, the ordinary least squares imputation methods perform better than the local versions when the data is generated from a unidimensional source.

Thus, a measure of structural complexity of the data should be taken into account to give the user a guidance in prior selection of appropriate imputation methods. In this project, two different approaches, one based on the principal component analysis, a.k.a. SVD, and the other on te single-linkage clustering are implemented.

As is well known, a singular value squared shows the part of the total data scatter taken into account by the corresponding principal component. The relative contribution of h-th factor to the data scatter, thus, is equal to

$$Contribution_h = \frac{\mu_h^2}{\sum_{i=1}^{N} \sum_{k=1}^{n} x_{ik}^2} \qquad (6.1.2)$$

where μ_h is h-th singular value of matrix $\mathbf{X}$. This measure can be extended to the case of missing data, as well.

The proportion of the first greatest component (and, sometimes, the second greatest component) shows how much the rank of the data matrix is close to 1: the larger $Contribution_1$ the closer. The data matrix is simplest when it has rank 1, that is, $Contribution_1 = 1$.

The single-linkage clustering shows how many connected components are formed by the data entities [Jain and Dubes, 1988, Mirkin, 1996]. When the data consists of several well separated clusters, the single-linkage components tend to reflect them so that a partition of the single-linkage clustering hierarchy produces a number of clusters with many entities in each. In contrast, when the data set has no visible clustering structure, the single linkage clusters appear much uniform: there is only one a big cluster and the rest are just singletons. This is why the distribution of entities in single-linkage clusters can be used as an indicator of visibility of a cluster structure in data.

The Table 6.3 presents a summary of the two measures described at each of the ten data sets generated from NetLab Gaussian 5-mixture: the left part of the table shows contributions of the first and second greatest factors to the data scatter, and the right part distributions of entities over three or five single linkage clusters.

The data in Table 6.3 show that all the data sets generated under the original Netlab Data Model are rather tight clouds of points with no visible structure in them.

Table 6.4 shows the individual and summary contributions of the first two factors to the data scatter at scaled NetLab Gaussian mixture data generators. In contrast

Data Sets	*Contribution (%)*		*Single linkage classes*				
	First	*Second*	*First*	*Second*	*Third*	*Fourth*	*Fifth*
Data-1	57.93	8.26	207	1	1	3	1
Data-2	62.73	8.37	195	1	2	1	1
Data-3	60.91	7.15	228	2	1	1	1
Data-4	58.88	7.81	197	1	4	1	1
Data-5	58.41	10.81	203	2	1	1	1
Data-6	61.50	8.79	221	1	1	1	2
Data-7	59.69	11.27	241	1	1	1	2
Data-8	53.64	9.79	241	1	1	1	1
Data-9	58.33	8.99	225	2	1	1	2
Data-10	58.42	9.60	1	1	224	1	1

Table 6.3: The Contribution of singular values and distribution of single lingkage clusters in NetLab Gaussian 5-mixture data model.

to the previous tables, data sets generated under the scaled Netlab Data Model have a visible cluster structure (see Table 6.4).

Data Sets	*Contribution (%)*		*Single linkage classes*				
	First	*Second*	*First*	*Second*	*Third*	*Fourth*	*Fifth*
Data-1	50.38	26.95	190	2	1	53	2
Data-2	46.83	24.56	164	1	40	1	1
Data-3	37.83	29.62	37	1	5	114	76
Data-4	38.42	22.23	144	1	42	6	47
Data-5	48.08	23.02	43	200	1	1	1
Data-6	49.69	20.51	129	33	1	43	1
Data-7	34.33	31.36	39	44	38	39	47
Data-8	55.77	16.02	122	38	44	1	1
Data-9	47.00	19.38	105	1	44	46	51
Data-10	49.21	19.09	144	3	38	1	45

Table 6.4: The Contribution of singular values and distribution of single linkage clusters in Scaled NetLab Gaussian 5-mixture data model.

Table 6.5 shows contributions of the first factor to data sets generated according to the unidimensional model at the noise levels from $\epsilon = 0.1$ through $\epsilon = 0.6$. The contribution falls from 97% to 52%, on average, almost proportionally to the noise level. Yet the unidimensional global least squares methods NIPALS and IMLS-1 outperform all the other methods at the latter's data generation model (as shown in Table 6.1).

Data Sets	*Noise Level*					
	0.1	*0.2*	*0.3*	*0.4*	*0.5*	*0.6*
Data-1	97.39	90.28	80.54	70.11	60.31	51.78
Data-2	96.95	88.84	78.10	67.03	56.99	48.48
Data-3	96.92	88.89	78.45	67.77	58.10	49.89
Data-4	97.61	91.14	82.18	72.42	63.05	54.69
Data-5	97.15	89.61	79.57	69.09	59.41	51.06

Table 6.5: Contribution of the first factor to the data scatter (%) for rank one data generator.

6.1.7 Summary of the Results

In the rank one data model experiments, the simple least squares imputation with one factor approximation, IMLS-1 and NIPALS (ILS-1) algorithms, surpass the other methods. Indeed, these methods work on unidimensional subspace only. Thus, more general approaches of least squares imputation, ILS and IMLS-4, have just produced larger error. In ether case, the global-local version of least squares imputation, INI, comes up as second best. In order to provide user-guidance the structural complexity of data sets is introduced. This complexity is measured according to contribution of singular values or/and the linkage-single cluster of entities.

6.2 Global-Local Least Squares Imputation Experiments with Real-World Marketing Research Data

6.2.1 Selection of Algorithms

1. INI: the global-local versions of least squares imputation [Wasito and Mirkin, 2002].

2. EM-Strauss: EM algorithm with multiple regression imputation [Strauss et al., 2002].

3. EM-Schafer: EM algorithm with random imputation [Schafer, 1997b].

4. MI: Multiple imputation with Markov-Chain Monte Carlo simulation using 10 imputations [Schafer, 1997b].

The details of the maximum likelihood based method can be seen in Section (2.2).

6.2.2 Description of the Data Set

The data set is produced from a typical database of the large manufacturer and is devoted to the problem of retention of existing customers. There are many variables describing customers behavior and service features, and there is a target binary variable (”refused the service or not”). The problem is to create a satisfactory recognition rule(-s) to predict those who will cancel the service agreement. The data set and the problem formulated are quite typical for many applications, and in that sense the reconstruction of missing values for such a data set is of a practical interest. This original data set consists of 5001 entities and 65 attributes which are mostly numeric (60), categorical (2), and binary attributes (3).

6.2.3 Sampling

This experiment utilizes 50 samples (size: 250×20) which are generated randomly from the original database at each level of missings. Thus, there are 150 samples which were generated for the experiments.

6.2.4 Generation of Missings

The procedure for creating missings similar to the previous experiment. However, the missings are generated randomly on the original real data set (size 5001×65) at the level of missings 1%, 5% and 10%.

6.2.5 Data Pre-processing

Within the imputation techniques while the experiment running, the data pre-processing, especially for real data sets, is calculated in following procedures:

$$x_{ik} = \frac{(x_{ik} - \mu_k)}{range_k} \tag{6.2.1}$$

where μ_k and $range_k$ defined as mean and range of attribute respectively those calculated as:

$$\mu_k = \frac{\sum_{i=1}^{N} x_{ik} * m_{ik}}{N} \tag{6.2.2}$$

$$range_k = max_i(x_k) - min_i(x_k) \tag{6.2.3}$$

6.2.6 Evaluation of Results

The evaluation of results concerning the exactness of imputation quality is measured according to (5.4.3). To evaluate the performance of the imputation methods, the elapsed CPU-time for running the program at Pentium III 733 MHz are recorded.

6.2.7 Results

The experiments are carried out in the two settings: (1) The experiments involving INI and two EM imputation versions: EM-Strauss and EM-Schafer. In this experiments, 50 samples are used for each level of missings. Thus there are 150 samples in the experiments; (2) The experiments involving INI, EM-Strauss, EM-Schafer and multiple imputation with 10 times imputation for each data sample. In this experiments, 20 samples are used for two level of missings: 5% and 10%. The results of each experiment will be shown in turn.

6.2.7.1 The Experiments with INI and Two EM Imputation Versions

The experiments are carried out using 50 samples from three "population" with level of missings: 1%, 5% and 10% of all data entry. The result of experiment is summarized according to the pair-wise comparison of performance of imputation techniques: INI, EM-Strauss and EM-Schafer.

In our experiments, all imputation techniques, on some occasions, cannot be proceed. The failing of computation is caused by the nature of algorithm. It is can be described as follows. INI cannot be implemented in case the subset of data matrix those to be found by k-NN algorithm contains all zeros elements in one column or more. Thus, the Equation (3.2.8) and (3.2.9) cannot be computed. Finally, the imputed values cannot be found. At other hand, for both versions of EM algorithm, the full covariance matrix that to be found from EM computation should be positive definite, otherwise, imputed values cannot be calculated.

The Comparison of Error of Imputation

Table 6.6 shows that, overall, except at level 5% missings, the INI to be the best method followed by EM algorithm with multiple regression (EM-Strauss) and the EM algorithm with random imputation (EM-Schafer) to be the worst.

Methods of Imputation	1%			5%			10%		
	1	2	3	1	2	3	1	2	3
INI	-	36	14	-	50	16	-	36	18
EM-Strauss	64	-	22	50	-	14	64	-	20
EM-Schafer	86	78	-	84	86	-	82	80	-

Table 6.6: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on 50 samples generated from database with 1%, 5% and 10% random missing data where 1,2 and 3 denote INI, EM-Strauss and EM-Schafer, respectively.

The Performance

With regarding CPU time performance, EM-Schafer algorithm provides the most fastest of rate of convergence and INI to be the second fastest. On average, EM-Schafer, is 1-10 times faster than INI and 10-1000 times faster then EM-Strauss.

6.2.7.2 The Experiments with INI, Two EM Imputation Versions and MI

This time, the experiments are carried out using 20 samples out 50 samples which are used in the previous experiments. The samples are chosen from "population" with level of missings: 5% and 10%. The error of imputation for each method is presented in Table 6.7.

Samples	*Methods*			
	INI	EM-Strauss	EM-Schafer	MI
1	73.78	91.35	NN	28.32
2	95.98	835.21	575.87	24.45
3	57.78	53.89	58.21	545.72
4	43.68	45.10	73.88	129.34
5	NN	NN	NN	40.99
6	48.35	59.94	58.20	144.32
7	61.28	51.40	89.86	99.91
8	142.80	307.59	1048.37	95.52
9	97.29	86.93	128.11	126.62
10	53.79	56.70	109.95	50.52
11	73.56	92.00	235.75	NN
12	75.86	293.90	184.65	389.28
13	134.05	840.37	5429.77	57.07
14	62.17	41.53	136.28	49.51
15	78.97	360.20	NN	NN
16	67.80	113.21	723.93	57.63
17	44.93	63.34	62.96	50.76
18	74.37	71.53	NN	333.34
19	72.44	78.21	150.24	87.83
20	78.68	115.89	542.38	51.86

Table 6.7: The squared error of imputation (in %) of INI, EM-Strauss, EM-Schafer and MI on 20 samples at 10% missings entry where NN denotes the methods fail to proceed.

Once again, the result of experiment is summarized according to the pair-wise comparison of imputation methods: INI, EM-Strauss, EM-Schafer and MI with 10 times imputation for each sample. The comparison is shown in Table 6.8.

Table 6.8 shows that at level 5%, three methods, INI, EM-Strauss and MI, provide almost the similar results. However, in the close range, EM-Strauss appears as the best method. Then MI appears as the second best. However, as the level of missings increase to 10%. INI surpasses the other methods. Then it is followed by EM-Strauss. As shown in the previous experiments, the EM-Schafer consistently to be the worst method.

Methods of Imputation	5%				10%			
	1	2	3	4	1	2	3	4
INI	-	50	30	55	-	26	0	47
EM-Strauss	50	-	25	45	74	-	25	47
EM-Schafer	70	75	-	80	100	75	-	67
MI	45	55	20	-	53	53	33	-

Table 6.8: The pair-wise comparison of methods; an entry (i, j) shows how many times in % method j outperformed method i on 20 samples generated from database with 5% and 10% random missing data where 1,2,3 and 4 denote INI, EM-Strauss, EM-Schafer and MI, respectively.

6.2.8 Summary of the Results

With regard the error of imputation, overall, INI surpasses EM-Strauss and EM-Schafer. Also, INI surpasses MI at level of missings 10%.

In either case, in terms of the rate of convergence, EM-Schafer which calculates the complete-data sufficient statistics matrix on its upper-triangular portions only, consistently to be the fastest method.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

7.1.1 Global and Local Least Squares Imputation

This work experimentally explores a number of least squares data imputation techniques that extend the singular value decomposition of complete data matrices to the case of incomplete data. There appears to be two principal approaches to this: (1) by iteratively fitting the available data only, as in ILS, and (2) by iteratively updating a completed data matrix, as in IMLS.

Then local versions of the ILS and IMLS methods based on utilising the nearest neighbour (NN) approach were proposed. Also, a combined method INI has been developed by using the NN approach on a globally imputed matrix.

7.1.2 The Development of Experimental Setting

A scheme for experiments has been adopted based on independently generating data (with all entries present) and missing entries so that the imputation results could be tested against those entries originally generated.

7.1.2.1 Generation of Missing Patterns

Three different patterns of missings have been proposed to supplement the conventional Complete Random pattern:

a. Inherited Random pattern reflecting step-by-step measurements in experimental data.

b. Sensitive Issue pattern modelling a concentration of missings within some issues that are sensitive to some respondents.

c. Merged Database to model the situation where features that are present in one database can be absent in the other.

7.1.2.2 Data Sets

The main type of data generation, Mixture of Gaussian distribution, is considered. Other data types of interest include: (1) Noised rank one data and (2) Samples from real-world marketing research database.

7.1.3 The Experimental Comparison of Various Least Squares Imputation

A set of eight least squares based methods have been tested on simulated data to compare their performances. The well-known average scoring method Mean and its NN version, N-Mean, recently described in the literature, have been used as the bottom-line. The results show that the relative performances of the methods depend on the characteristics of the data, missing patterns and the proportion of missings.

7.1.3.1 The Performance of Least Squares Imputation on Complete Random Pattern

NetLab Gaussian Mixture Data Model

Based on this data model, overall, the local versions perform better than their global versions. Even N-Mean may do relatively well when missings are rare. However, the only method to consistently outperform the others, especially when proportion of missings is in the range of 5 to 25 per cent, is the combined method INI.

Scaled NetLab Gaussian Mixture Data Model

Overall, local versions of least squares imputation perform well. Furthermore, the performance of the methods vary according to level of missings. At level 1% to 10%, the global-local least squares, INI, is the best. However, at level 15% or more, the local version of IMLS, N-IMLS, surpasses the other methods.

7.1.3.2 The Performance of Least Squares Imputation on Different Missing Patterns

Some of the findings resulting from the experiments with two Gaussian data models and five missing patterns (Complete random, Inherited random, Sensitive issue, Merged database with missings from one database, Merged database with missings from two databases):

1. The three NN-based least squares techniques provide the best results for the first four missing patterns under either of the two Netlab Data Models. Global least squares methods win only with the Merged database with missings from two databases under the scaled Netlab Data Model. N-Mean joins in the winning methods when there are few Complete random missings when the missings patterns is caused by Merged database with missings from two databases.

2. The global-local method, INI, outperforms the others under the original Netlab Data Model, and N-IMLS frequently wins under the scaled Netlab Data Model.

3. The ILS approach frequently does not converge, which makes the IMLS technique and its NN version more preferable.

4. The scaled Netlab Gaussian Data Model leads to smaller errors of least squares imputation techniques for all five missing patterns considered. For some missing patterns, the scaled model leads to different results.

7.1.4 Other Data Models

7.1.4.1 Rank One Data Model

Under this data model, simple global least squares imputation methods with one factor approximation, IMLS-1 and NIPALS (that is, ILS-1), are best. This probably can be explained by the very nature of the data generated: unidimensionality. The combined method, INI, could be considered as second best. Indeed, INI heavily relies on IMLS-1.

7.1.4.2 Experiments on a Marketing Research Database

The combined global-local least squares imputation, INI, in terms of the error of imputation, on average, outperformed two versions of the EM algorithm: EM-Strauss and EM-Schafer and multiple imputation approach. The only exception to this is in the case when the samples came from the database with 5% missing entries. However, the version EM-Schafer performs rather fast, typically faster than INI, which probably explains the huge error of the working of this program.

7.2 Future Work

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

1. It should be a theoretical investigation carried out on the properties of convergence for both major iterative techniques, ILS and IMLS. In our computations, ILS does not always converge, even when the Gabriel-Zamir setting is applied to initialise the process.

2. The performances of the least squares based techniques should more comprehensively compared with those of another set of popular imputation techniques based on a maximum likelihood principle such as the multiple imputation (MI). This requires some additional work since the latter methods can be computationally expensive, and also inapplicable when the proportion of missings is comparatively large (10% or more). Also, the evaluation criteria for a method's performance should be extended to ordinary statistical analysis such as distribution and estimation of parameter accuracy.

3. Modelling of missing pattern should be extended to the most challenging missing pattern, the non-ignorable (NI).

Appendix A

A Set of MATLAB Tools

A.1 Methods

A.1.1 Iterative Least Squares (ILS)

```
function [xout,exils]=ils(x,mt,p,maxstep)

% This program implements  ILS algorithm on MatLab Version 6.
% Input:
% x=data set with r rows and n columns.
% mt=arrays of missingness matrix at level t% with p various patterns
% p=number of missings patterns.
% maxstep=number of factors to be approximated.
% Output:
% xout= data reconstruction including the imputed missing values.
% exils= the error of imputation.


% Copy right 2001 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.

tol=1e-4;               % tolerance convergence.
ns=0;                   % counting number for nonconvergences.
ItMax=800;              % maximimum iterations.
[r,n]=size(x);
Z=x;
co=ones(r,1);

for sm=1:p
    mis=mt{sm};         % Use the p-th missing pattern at level t% missings.
    X=x;
    Xm=X.*mis;          % Construct the missing values in data set.
    cr=ones(1,n);
    czt=zeros(r,n);
    rec=0;

for t=1:maxstep
 ci(t,:)=cr;             % Initial value for vector c. For ILS-GZ, utilize
 cn=1/sqrt(r)*ci(t,:);  % the Gabriel-Zamir initial computation as shown in
 Y=X.*mis;              % init program.
 delta=1;
 cycle=0;
 so=0;

% Execute ILS algorithm.

   while (delta>tol)
   ss=so;
   c=cn;
   cc=c.*c;
   cycle=cycle+1;
   num1=Y*c';
   den1=mis*(cc)';
   zn=num1./den1;
   znsq=zn.*zn;
   num2=zn'*Y;
   den2=znsq'*mis;
   cn=num2./den2;
   cnm=norm(cn);         % Normalize the c values
   cn=cn/cnm;
   delta=norm(cn-c);
   rec=zn*cn;
   if cycle > ItMax
   disp('ILS Algorithm is not converged'); % record the nonconvergences.
          disp('in th-component');
          t
          ns=ns+1;
          break,end
   end;
   cp(t,:)=cn;         % store the found c-value on the t-th factor.
   zp(:,t)=zn;         % store the found z-value on the t-th factor.
   X=X-rec;            % Approximate the next subspace.
   czt=czt+rec;        % data reconstruction.
end;


% Evaluate  the imputed data values according to Equation 5.4.3.


exact(sm)=ie(Z,czt,mis);
end
xout=czt;
exils=exact;
return
```

A.1.2 Iterative Majorization Least Squares (IMLS)

```
function [xout,exmls]=imls(x,mt,p,maxstep)

% This program implements  IMLS algorithm on MatLab Version 6.
% Input:
% x=data set with r rows and n columns.
% mt=arrays of missingness matrix at level t% with p various patterns
% p=number of missings patterns.
% maxstep=number of factors to be approximated.
% Output:
% xout= data reconstruction including the imputed missing values.
% exmls= the error of imputation.

%Copy right 2001 I. Wasito
% School of Computer Science, Birkbeck College, University of London
```

```
tol=1e-4; % tolerance convergence.
ns=0; % counting values for nonconvergences.
ItMax=800; % maximum iteration
[r,n]=size(x);
Z=x;
co=ones(r,1);

for sm=1:p
     mis=mt{sm};
     X=x;
     Xm=X.*mis;                % Construct missing values
     xsum=sum(sum(Xm.*Xm));    % Convergence criteria alongside tol value.
     m=1-mis;
     cp=ones(maxstep,n);
     zp=ones(r,maxstep);
     ic=ones(r,n);

czt=zeros(r,n);
rec=zeros(r,n);                % initialize the missing values to zero

% Execute the IMLS algorithm

      for t=1:maxstep
         delta=xsum;
         counter=0;
         ci(t,:)=ones(1,n);
         cn=1/sqrt(r)*ci(t,:);
         so=xsum;

% Execute the Kiers algorithm

while (delta>tol*so)

   reco=rec;
   Y=X.*mis+reco.*m;
   ss=so;
   c=cn;
   cc=c.*c;
   counter=counter+1;
   num1=Y*c';
   den1=ic*(cc)';
   zn=num1./den1;
   mzn=norm(zn);
   znsq=zn.*zn;
   num2=zn'*Y;
   den2=znsq'*ic;
   cn=num2./den2;
   cnm=norm(cn);
   cn=cn/cnm;                 % normalize c values
   rec=zn*cn;                 % data reconstruction
   dif=Xm-rec;
   difmis=dif.*mis;
   so=sum(sum(difmis.*difmis));
   delta=abs(so-ss);

   if counter > ItMax
   disp('G-WLS Algorithm is not converged');  % record the nonconvergence
          disp('in th-component');
          t
          ns=ns+1;
          break,end
  end
cp(t,:)=cn;   % store the found c-value on the t-th factor.
zp(:,t)=zn;   % store the found z-value on the t-th factor.
X=X-rec;      % approximate the next subspace.
czt=czt+rec;  % data reconstruction.
end

% Evaluate  the imputed data values according to Equation 5.4.3.

exact(sm)=ie(Z,czt,mis);
end
xout=czt;
exmls=exact;
return
```

A.1.3 Nearest Neighbour Least Squares Imputation

```
function [xout,exnils]=nnils(x,mt,p,k,meth)

% This program implements  Least Squares algorithm with Nearest Neighbour
% on MatLab Version 6.
% Input:
% x=data set with r rows and n columns.
% mt=arrays of missingness matrix at level t% with p various patterns
% p=number of missings patterns.
% k=number of neighbours.
% meth=1 for ILS and  meth=2 for IMLS, otherwise compute mean imputation.
% Output:
% xout= data reconstruction including the imputed missing values.
% exnils= the error of imputation.

%Copy right 2001-2002 I. Wasito
%School of Computer Science, Birkbeck College, University of London

Z=x;
xp=x;
[r,n]=size(x);

for sm=1:p
     m=mt{sm};
 distm=eucmiss(xp,m);       % Calculate the Euclidean distance
 [sdm,idxm]=sort(distm);    % Sort the distance.

for kn=1:r
 xaug=(xp(idxm(1:r,kn),:)); % Set the ordered entity.
 maug=(m(idxm(1:r,kn),:));  % Set the corresponding ordered missing matrix.
 dist=sdm(1:r,kn);
    v=find(maug(1,:)==0);   %  Find the target entity.
  if length(v)~=0
  [xk,mk]=negils(xaug,maug,k); % Select the neighbours of target entity.

switch meth    % select one of the least squares imputation technique.
   case 1
    xl=ils(xk,mk,1);           %  N-ILS algorithm.
   case 2
    xl=imls(xk,mk,1);          %  N-IMLS algorithm.
   otherwise
    xl=mean(xk,mk);  %         %  N-Mean algorithm.
   end
     y(kn,:)=xl(1,:);
 else
     y(kn,:)=xaug(1,:);
    end
end

% Evaluate  the imputed data values according to Equation 5.4.3.
exact(sm)=ie(Z,y,m);
end
xout=y;
exnils=exact;
return
```

A.1.4 IMLS N-IMLS (INI)

```
function exiki=nnikip(x,mt,p,k)

% This program implements combination  of ordinary IMLS algorithm
% and its Nearest Neighbour version on MatLab Version 6.
% Input:
% x=data set with r rows and n columns.
% mt=arrays of missingness matrix at level t% with p various patterns
% p=number of missings patterns.
% k=number of neighbours.
% Output:
% xout= data reconstruction including the imputed missing values.
% exnils= the error of imputation.

%Copy right 2001-2002 I. Wasito
%School of Computer Science, Birkbeck College, University of London

Z=x;
[r,n]=size(x);

for s=1:p
mis=mt{s};
xt=imls(x,mis,4);       % Compute the  IMLS with 4 factors.
xp=xp.*mis+xt.*(1-mis); % Fill in the missing value.

%Compute the distance with ‘‘completed’’ data

x2=sum(xp.^2,2);
distance=repmat(x2,1,r)+repmat(x2',r,1)-2*xp*xp';
[sd,idx]=sort(distance);



for kn=1:r
  xaug=(xp(idx(1:r,kn),:));      % Set  the ordered entity
  maug=(mis(idx(1:r,kn),:));     % Set the corresponding  ordered missing matrix.
  v=find(maug(1,:)==0);
  if length(v)~=0
   [xk,mk]=negils(xaug,maug,k); % Select the neighbours.
   xl=ngwls(xk,mk,1);           % Compute IMLS with 1 factor only.
   y(kn,:)=xl(1,:);             % store the data reconstruction.
else
  y(kn,:)=xaug(1,:);
  %end
end
end
```

```
% Evaluate the imputed values using Equation 4.5.3.

exact(s)=ie(Z,y,m);
end
exiki=exact;
xout=y;
return
```

A.1.5 Evaluation of the Quality of Imputation

```
function ex=ie(xt,xm,m);

% This program implements evaluation of quality of imputation
% according to Equation 5.4.3  on MatLab Version 6.
% Input:
% xt= data set without missings;
% xm= data set with missings;
% m= matrix of missingness.
% Output:
% ex= the error of imputation (in %).

%Copy right 2001-2002 I. Wasito
%School of Computer Science, Birkbeck College, University of London


dif=xt-xm;
difmis=dif.*(1-m);
dd=sum(sum(difmis.*difmis));
xtmis=xt.*(1-mis);
bb=sum(sum(xtmis.*xtmis));
ex=dd/bb*100;
return
```

A.1.6 Euclidean Distance with Incomplete Data

```
function d=eucmiss(x,m);

% This program compute the distance of entities with some missing values.
% Input:
% x=data set with r rows and n columns;
% m=matrix of missingness;
% Output:
% d= Euclidean distance;

% Copy right 2001-2002 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.

[r,n]=size(x);

for i=1:r
   for ii=i:r
       dx=0;
       md=ones(r,n);  % matrix of binary  which have value 1 if the pair of attributes
```

```
          for k=1:n   % are non-missing.
              if (m(i,k)==0) |  (m(ii,k)==0)
                  md(i,k)=0;  % if at least one of pair of attributes are missing
              end             % then set the matrix of binary to zero.
             dx=dx+((x(i,k)-x(ii,k))^2)*md(i,k); % Calculate the Euclidean distance.
          end
        d(i,ii)=dx;          % The distance of entity i and entity ii.
        d(ii,i)=d(i,ii);     % symmetric properties of distance metric.
        end
       end
 d=sqrt(d);                  % square-root of the distance metric.
 return
```

A.1.7 Selection of Neighbours

```
function [xout,mout]=negils(x,m,k);

% This program determines  the neighbours
% Input:
% x=data set with r rows and n columns;
% m=matrix of missingness;
% k=number of neighbours;
% Output:
% xout= data reconstruction including imputed values;
% mout= corresponding missingness matrix;

% Copy right 2001-2002 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.

[r,n]=size(x);
xout=x(1:k,:);
mout=m(1:k,:);
return
```

A.1.8 Gabriel-Zamir Initialization

```
function c=init(x,m);

% This program calculate the Gabriel-Zamir initialization for c value.
% Input:
% x=data set;
% m=matrix of missingness;
% Output:
% c= the initiall value of c;

% Copy right 2001-2002 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.



[r,n]=size(x); y=x; cq=0; mxp=0;
[v,w]=find(m==0);         % find the rows and columns which contain missing values.

% Proceed the Gabriel-Zamir initialization
```

```
 for vt=1:length(v)
   for wt=1:length(w)
     qq=sum(m(v(vt),:).*(x(v(vt),:).^2));
     pq=sum(m(:,w(wt)).*(x(:,w(wt)).^2));
     cq=pq+qq;
   if cq>=mxp
        mxp=cq;
        it=v(vt);
        pk=w(wt);
   end
 end
end

wyyy=0; wyy=0;

for yk=1:r
  for xk=1:n
   if (yk~=it) | (xk~=pk)
   wyyy=wyyy+(mt(yk,xk)*x(yk,pk).^2*(x(it,xk).^2));
     wyy=wyy+mt(yk,xk)*x(yk,pk)*x(it,xk)*x(yk,xk);
  end
  end
  end
  be=(wyy/wyyy);
  [tf,tg]=find(m(it,:)==0);
  x(it,tg)=be;
  [rt,nt]=size(it);

  if rt>1
    it(2:rt,:)=[];
  end
  c=x(it,:);
  c=c/norm(c); % normalize the c value.
return
```

A.1.9 Data-Preprocessing

```
function xt=prep(x,mis);

% This program implements data pre-processing.
% Input:
% x=data set with r rows and n columns;
% mis=matrix of missingness;
% Output:
% xt= standardized data;

% Copy right 2001-2002 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.
[r,n]=size(x); co=ones(r,1);
 for km=1:n
   v=find(mis(:,km)==1);
   mx(km)=mean(x(v,km));  % Compute mean of observed values in the variables.
   jkx(km)=max(x(v,km))-min(x(v,km)); % Compute range of variables.
end
xt=(x-co*mx)./(co*jkx);
return
```

A.2 Generation of Missing Patterns

A.2.1 Inherited Pattern

```
function mp=mrandsys(r,n);
% This program generates inherited missing pattern.
% Input:
% r= number of rows;
% n= number of columns;
% Output:
% mp= matrix of missingness;

% Copy right 2001-2002 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.

% The proportion of missing to be implemented

p(1)=0.25;
p(2)=0.20;
p(3)=0.15;
p(4)=0.10;
p(5)=0.05;
p(6)=0.01;


% Generate the matrix of missingness for p=0.25.

numsim=100;
m=ones(r,n);
    q=p(1);
     nl=round(q*r*n);
     t=0;
    while t<nl
      for i=1:numsim
      u=round(rand*(r-1)+1);
      w=round(rand*(n-1)+1);
      nmr=length(find(m(u,:)>0));
      nmc=length(find(m(:,w)>0));
      if (m(u,w) ~= 0),break,end
      end
   if (nmr-1)>0
    m(u,w)=0;
    t=t+1;
   end
   end
   mp{6}=m;
 m2=m;      % store the found matrix  of missingness for next simulation.
 nl0=nl;

 % Generate the inherited missings starting from the previous matrix.

 for ss=2:6
     q=p(ss);
     nl=round(q*r*n);
     t=nl0;
    while t>nl
      for i=1:numsim
      u=round(rand*(r-1)+1);
      w=round(rand*(n-1)+1);
      nmr=length(find(m2(u,:)>0));
      nmc=length(find(m2(:,w)>0));
      if (m2(u,w) ~= 1),break,end
      end
   if (nmr-1)>0
    m2(u,w)=1;
    t=t-1;
   end
   end
   mp{7-ss}=m2;
   nl0=nl;
end
return
```

A.2.2 Sensitive Issue Pattern

```
function mcom=comsen(mm,n);
% This program generates sensitive issue pattern.
% Input:
% mm= number of rows;
% n= number of columns;
% Output:
% mcom= matrix of missingness;

% Copy right 2001-2002 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.

% Proportion of missings

ip(1)=0.01;
ip(2)=0.05;
ip(3)=0.10;

m=ones(mm,n);
r=0;
q=0;

% Generate for each proportion of missings

for ll=1:3
 switch ll
   case 1
    p=round(ip(ll)*mm*n)
    while r*q<p
     q=round((0.10+0.40*rand)*n);
     r=round((0.25+0.25*rand)*mm);
   end

   for kk=1:q
     qc(kk)=round(1+rand*(n-1));
   end
   for jj=1:r
    qr(jj)=round(1+rand*(mm-1));
   end

mcom{ll}=msen(m,p,q,r,qc,qr);

rand('state',0) case 2
```

```
   p=round(ip(ll)*mm*n)
    while r*q<p
     q=round((0.20+0.30*rand)*n);
     r=round((0.25+0.25*rand)*mm);
 end
for kk=1:q
   qc(kk)=round(1+rand*(n-1));
 end

 for jj=1:r
    qr(jj)=round(1+rand*(mm-1));
 end

 mcom{ll}=msen(m,p,q,r,qc,qr);

 rand('state',0)
 case 3
    p=round(ip(ll)*mm*n)
    while r*q<p
     q=round((0.25+0.25*rand)*n);
     r=round((0.40+0.40*rand)*mm);
    end
 for kk=1:q
   qc(kk)=round(1+rand*(n-1));
 end
 for jj=1:r
    qr(jj)=round(1+rand*(mm-1));
 end
 m=ones(mm,n);
mcom{ll}=msen(m,p,q,r,qc,qr);
rand('state',0)
otherwise
  disp('no missing');
end
end
return

%The following subprogram determines the missings for each issue pattern

function mc=msen(m,t,nps,nts,qc,qr);

tn=0;
   while tn<t
      mm=round(1+(nts-1)*rand);
      nn=round(1+(nps-1)*rand);
      rr=round(rand);
         if (m(qr(mm),qc(nn))==1) & (rr==1)
           m(qr(mm),qc(nn))=0;
           tn=tn+1;
          end
    rand('state');
  end
  mc=m;
  return
```

A.2.3 Missings from One Database

```
function mcom=commis(r,n,p);
% This program generates missings from one database.
% Input:
```

```
% r= number of rows;
% n= number of columns;
% p= proportion of column which contains missing values
% Output:
% mcom= matrix of missingness;

% Copy right 2001-2002 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.

% Proportion of missings
ip(1)=1;
ip(2)=5;

m=ones(r,n);

% Generate missings from one database for each level of missings (1% and 5%).

for tt=1:2

q=ip(tt); t=(q/p)*100; % proportion of respondents which would no response
nps=round((p/100)*n);  % number of columns which contain missing values
nts=round((t/100)*r);  % number of respondents which would no response

% Generate randomly the columns which contain missing values

for i=1:nts
    ncs(i)=round(1+rand*(r-1));
end

%Generate randomly the rwos which contain no response respondents

for k=1:nps
    nrs(k)=round(1+rand*(n-1));
end

for ii=1:nts
    for kk=1:nps
     m(ncs(ii),nrs(kk))=0;
  end
end

mcom{tt}=m;
end
return
```

A.2.4 Missings from Two Databases

```
function mcom=merged(r,n);
% This program generates missings from two databases.
% Input:
% r= number of rows;
% n= number of columns;
% Output:
% mcom= matrix of missingness;

% Copy right 2001-2002 I.Wasito
% School of Computer Science and Information Systems, Birkbeck,
% University of London.
```

```
%Proportion of missings

p(1)=1;
p(2)=5;

%The proportion of number of entities for two databases
m1=0.6+0.2*rand;
m2=1-m1;

%Initialize matrix of missingness
m=ones(r,n);

%Generate missing from two databases for each level of missings.

for ss=1:2
r1=round(r*m1);
r2=round(r-r1);

%Determine the left-most of missings in second database

if p(ss)==1
  k2=rand*((p(ss)*n*r/(100*r1)));
else
    k2=((rand*((p(ss)*n*r/(100*r1))-r2/r1)));
end

%Determine the right-most of missings in the first database.

k1=((((p(ss)*n*r)-100*r2*k2))/(100*r1));

%Configure the missings pattern for two databases.

f1=fix(k1);
f1r=k1-f1;
f2=fix(k2);
f2r=k2-f2;

  if f1==0
     rk=round(r1*k1);
else
     rk=round(r1*f1r);
  end

ck=f1;

for ii=n-ck+1:n
  for i=1:r1
    m(i,ii)=0;
  end
end

kk=n-ck;
 for k=1:rk
  m(k,kk)=0;
end

 if f2==0
    rs2=round(r2*k2);
   else
    rs2=round(r2*f2r);
 end
```

```
ck2=f2;

for j=r:r-r2+1
   for jj=1:ck2
    m(j,jj)=0;
   end
end

dd=ck2+1;

for l=r-rs2+1:r
    m(l,dd)=0;
end
mcom{ss}=m;
end
return
```

Appendix B

Data Generators Illustrated

B.1 NetLab Gaussian 5-Mixture Data Model

(a) The first example of NetLab Gaussian 5-mixture

(b) The second example of NetLab Gaussian 5-mixture

(c) The third example of NetLab Gaussian 5-mixture

B.2 Scaled NetLab Gaussian 5-Mixture Data Model

(d) The first example of scaled NetLab Gaussian 5-mixture

(e) The second example of scaled NetLab Gaussian 5-mixture

(f) The third example of scaled NetLab Gaussian 5-mixture

B.3 Rank One Data Model

(g) Rank one data model with noise level=0.1

(h) Rank one data model with noise level=0.3

(i) Rank one data model with noise level=0.6

B.4 Standarized Data Samples of Marketing Database

(j) The First Sample

(k) The Second Sample

(l) The Third Sample

Appendix C

Exemplary Results

C.1 The Results of Experiments with 10 Scaled NetLab Data Sets times 10 Complete Random Missing Patterns

C.1.1 The Error of Imputation of ILS Method

The Proportion of Missing					
1%	5%	10%	15%	20%	25%
18.33	14.05	17.13	16.97	16.78	17.85
14.83	11.27	12.60	16.18	13.70	17.14
8.90	20.37	15.21	18.28	17.92	17.42
9.76	11.94	16.13	15.16	14.61	19.26
10.43	13.88	13.39	15.85	17.69	18.22
18.86	19.33	15.34	16.16	15.02	20.78
30.26	16.14	15.56	15.24	16.50	16.52
16.07	13.06	15.15	15.32	19.98	16.64
26.62	14.13	16.79	14.62	17.46	18.33
11.40	17.49	15.67	16.67	16.18	16.50
6.69	10.64	12.62	12.51	13.72	15.46
14.83	12.04	16.68	12.38	13.68	18.14
8.42	8.11	11.68	16.17	15.20	15.00
15.95	14.13	12.58	13.72	13.73	15.59
17.76	12.70	13.94	15.00	14.21	17.36
9.55	15.80	13.52	11.99	16.59	15.25
15.77	10.82	12.95	17.03	14.27	16.11
8.83	14.68	14.23	15.82	12.51	16.29
17.74	13.59	12.28	15.09	14.25	14.88
4.92	11.62	13.30	14.40	12.70	16.32
18.16	16.69	19.50	23.19	21.06	23.14
15.65	32.41	17.00	20.18	23.82	24.35
17.51	18.80	17.19	23.01	21.86	23.22
8.34	15.45	16.91	22.14	20.77	25.63
12.47	19.68	21.89	21.48	23.33	30.77
28.85	22.74	18.55	19.32	22.41	25.17
23.34	16.46	21.21	21.57	23.23	25.48
12.87	18.46	18.24	18.60	26.11	32.13

16.15	15.35	19.80	20.95	24.26	24.34
13.11	24.52	19.98	22.37	27.44	19.69
9.57	15.41	18.59	18.55	21.17	21.87
6.22	17.97	18.73	17.75	23.88	22.95
11.82	18.62	15.78	20.30	21.28	22.88
14.96	13.14	20.87	18.77	18.30	23.10
13.20	24.96	18.30	19.29	20.81	24.08
12.35	17.30	18.95	16.64	21.50	24.48
11.59	19.76	16.91	17.51	22.26	22.11
29.84	17.90	19.69	17.99	20.00	23.52
11.61	15.71	18.83	18.37	23.34	23.98
18.58	19.98	17.31	23.50	24.04	25.01
37.76	17.27	17.65	19.00	21.89	20.19
15.27	18.14	17.35	17.06	18.60	18.78
27.39	19.76	18.52	17.01	21.06	20.93
11.87	25.83	18.83	16.01	18.11	21.74
19.31	14.97	20.20	20.00	18.16	19.23
32.08	16.39	16.45	17.33	18.77	21.89
10.13	18.85	14.12	19.79	20.97	21.08
9.76	13.87	17.58	21.62	20.15	21.32
16.56	18.91	17.71	17.23	19.55	18.71
18.64	16.94	15.24	18.03	20.72	20.13
10.56	18.55	12.76	13.05	14.82	15.36
10.07	15.92	11.48	13.61	15.12	15.93
10.24	16.64	13.84	14.94	13.53	14.99
13.57	9.96	14.13	15.26	14.39	13.84
6.29	15.83	15.19	11.85	12.59	16.01
15.37	10.56	14.34	16.27	14.10	14.38
19.51	15.26	12.94	13.10	16.75	17.14
14.97	13.74	12.21	14.64	12.66	15.70
12.24	13.16	16.59	13.39	12.81	13.25
12.14	13.54	14.08	12.07	14.32	13.64
20.02	24.49	25.01	29.01	30.49	36.36
40.19	26.06	19.42	25.65	28.19	31.91
36.72	23.15	24.74	29.06	30.48	35.77
17.56	18.56	25.34	28.94	37.77	33.59
30.32	27.49	20.60	24.56	29.56	31.70
29.17	34.16	26.05	30.59	32.03	29.33
13.34	19.28	26.89	30.78	27.93	34.88
18.78	22.59	19.99	28.61	32.75	33.72
22.64	23.85	27.72	32.36	27.56	34.93
23.05	24.15	22.13	27.85	30.35	28.55
23.67	18.45	21.29	27.18	21.68	23.47
12.30	17.39	19.14	21.43	26.13	25.18
33.70	19.45	19.14	22.09	25.87	26.83
13.36	22.02	21.54	20.70	23.92	26.74
9.14	18.09	23.06	22.37	24.36	23.94
13.73	22.49	24.62	19.02	21.65	20.17
27.22	16.26	18.09	22.60	23.83	24.70
16.73	15.82	20.74	19.31	20.98	22.87
11.35	37.50	20.51	20.17	23.66	22.33
11.12	21.11	20.67	18.61	23.05	22.81
14.84	14.56	15.79	15.99	20.63	22.15
12.14	15.31	14.67	20.16	17.12	20.52
15.04	22.55	18.78	19.17	19.07	21.44
13.92	15.06	18.59	20.23	18.69	22.11
13.40	14.16	17.14	17.24	18.70	23.03
13.34	17.34	14.70	18.60	17.33	20.74
10.28	21.87	20.79	19.55	22.22	21.75
16.01	21.24	18.29	18.52	20.42	18.88
10.78	21.36	18.18	17.87	17.96	22.61

19.58	16.19	17.91	19.92	18.69	21.26
11.41	15.10	17.83	17.10	18.93	19.13
9.99	18.87	15.19	17.80	18.97	20.83
28.35	16.81	14.24	18.01	20.04	18.78
18.60	18.13	14.66	19.52	19.12	20.39
28.38	14.95	16.07	15.55	18.71	20.94
16.87	17.80	16.75	17.50	17.35	18.69
23.07	17.45	18.43	17.73	19.03	20.02
13.75	14.32	15.95	18.20	18.61	20.45
31.77	16.12	17.70	17.87	17.75	21.02
13.44	16.77	15.99	16.92	19.14	19.64

C.1.2 The Error of Imputation of GZ Method

The Proportion of Missing					
1%	5%	10%	15%	20%	25%
18.33	14.05	17.13	16.97	16.78	17.85
14.83	11.27	12.60	16.18	13.70	17.14
8.90	20.36	15.21	18.28	17.92	17.41
9.76	11.94	16.13	15.16	14.61	19.26
10.43	13.88	13.39	15.85	17.69	18.22
18.86	19.33	15.33	16.16	15.01	20.79
30.26	16.14	15.56	15.24	16.50	16.52
16.07	13.06	15.15	15.32	19.98	16.64
26.62	14.13	16.79	14.62	17.46	18.33
11.40	17.49	15.67	16.67	16.18	16.50
6.69	10.64	12.62	12.51	13.72	15.46
14.83	12.03	16.68	12.38	13.68	18.14
8.42	8.11	11.68	16.17	15.20	15.00
15.95	14.13	12.58	13.73	13.73	15.59
17.76	12.70	13.94	15.00	14.21	17.36
9.55	15.80	13.52	11.99	16.59	15.24
15.77	10.82	12.95	17.03	14.27	16.11
8.83	14.68	14.23	15.82	12.51	16.29
17.75	13.59	12.28	15.10	14.25	14.88
4.92	11.62	13.29	14.40	12.70	16.32
18.16	16.69	19.51	23.19	20.71	23.14
15.64	32.41	17.00	20.18	24.29	24.35
17.51	18.81	17.20	23.01	21.85	23.22
8.34	15.45	17.49	22.14	20.77	26.06
12.46	19.68	21.89	21.48	23.34	30.76
28.86	22.75	18.56	19.32	22.41	25.82
23.33	16.45	21.21	21.57	24.25	25.48
12.87	18.46	18.24	18.60	26.13	32.13
16.15	15.34	19.80	20.95	24.25	24.34
13.11	24.54	19.98	22.37	27.44	19.69
9.57	15.41	18.59	18.55	21.17	21.87
6.22	17.97	18.73	17.75	23.88	22.95
11.82	18.62	15.78	20.30	21.28	22.88
14.96	13.13	20.87	18.77	18.31	23.10
13.20	24.96	18.30	19.29	20.81	24.08
12.35	17.30	18.95	16.64	21.50	24.47
11.59	19.76	16.91	17.51	22.26	22.11
29.84	17.90	19.69	17.99	20.00	23.52
11.61	15.71	18.83	18.37	23.34	23.98
18.58	19.98	17.31	23.50	24.04	25.01
37.76	17.27	17.65	19.00	21.89	20.19

15.27	18.14	17.35	17.06	18.60	18.78
27.39	19.76	18.52	17.01	21.06	20.93
11.87	25.83	18.83	16.01	18.11	21.74
19.31	14.97	20.20	20.00	18.16	19.23
32.08	16.39	16.45	17.33	18.77	21.89
10.13	18.85	14.12	19.79	20.97	21.08
9.76	13.87	17.58	21.62	20.14	21.32
16.56	18.91	17.71	17.23	19.55	18.71
18.64	16.94	15.24	18.03	20.72	20.13
10.56	18.55	12.76	13.05	14.82	15.36
10.07	15.92	11.48	13.61	15.12	15.93
10.24	16.64	13.84	14.94	13.53	14.99
13.57	9.96	14.13	15.26	14.39	13.84
6.29	15.83	15.19	11.85	12.59	16.01
15.37	10.56	14.34	16.27	14.10	14.38
19.51	15.26	12.94	13.10	16.75	17.14
14.97	13.74	12.22	14.64	12.66	15.70
12.24	13.16	16.59	13.40	12.81	13.25
12.14	13.54	14.08	12.07	14.32	13.64
20.03	24.49	25.01	29.01	30.50	36.36
40.19	26.06	19.42	25.65	26.97	31.91
36.72	23.15	24.74	29.06	30.48	35.77
17.56	18.56	25.34	28.94	37.77	33.58
30.32	27.49	20.60	24.56	29.56	31.70
29.17	34.16	26.05	30.59	32.03	29.33
13.34	19.28	26.89	30.78	27.93	34.88
18.78	22.59	19.99	28.61	32.75	33.90
22.63	23.85	27.72	32.36	27.56	34.93
23.06	24.15	22.13	27.85	30.34	28.56
23.67	18.45	21.29	27.18	21.68	23.47
12.30	17.39	19.14	21.43	26.13	25.18
33.71	19.45	19.14	22.09	25.86	26.83
13.36	22.02	21.54	20.70	23.92	26.74
9.14	18.09	23.07	22.37	24.36	23.94
13.73	22.49	24.62	19.03	21.65	20.17
27.22	16.26	18.09	22.59	23.83	24.69
16.73	15.82	20.74	19.31	20.98	22.87
11.35	37.51	20.51	20.17	23.66	22.33
11.12	21.11	20.67	18.62	23.05	22.81
14.84	14.56	15.79	15.99	20.63	22.15
12.14	15.31	14.67	20.16	17.12	20.52
15.03	22.55	18.78	19.17	19.07	21.43
13.92	15.06	18.59	20.23	18.68	22.11
13.40	14.16	17.14	17.24	18.70	23.03
13.35	17.34	14.70	18.60	17.33	20.74
10.28	21.87	20.79	19.55	22.22	21.74
16.01	21.24	18.29	18.52	20.42	18.88
10.78	21.37	18.18	17.87	17.96	22.61
19.58	16.19	17.90	19.92	18.69	21.26
11.40	15.10	17.84	17.10	18.93	19.14
9.99	18.88	15.19	17.80	18.97	20.83
28.35	16.81	14.24	18.01	20.05	18.78
18.60	18.13	14.66	19.52	19.12	20.39
28.39	14.95	16.07	15.55	18.71	20.94
16.87	17.80	16.75	17.50	17.35	18.69
23.06	17.45	18.43	17.73	19.03	20.02
13.75	14.33	15.95	18.20	18.61	20.45
31.77	16.11	17.70	17.88	17.75	21.02
13.43	16.77	15.99	16.92	19.14	19.65

C.1.3 The Error of Imputation of NIPALS Method

The Proportion of Missing					
1%	5%	10%	15%	20%	25%
62.78	58.39	63.42	58.73	62.26	61.73
70.61	49.83	59.30	58.42	54.37	58.39
47.15	75.29	57.86	71.28	56.76	58.55
63.54	47.91	56.85	67.17	56.66	57.32
58.16	57.52	55.28	55.87	62.53	55.27
50.31	55.53	60.00	60.91	55.35	67.11
56.68	57.70	66.71	57.25	59.75	60.31
60.31	55.81	57.54	55.76	57.88	55.27
85.19	63.89	67.50	55.63	56.36	59.66
57.32	62.04	63.43	61.12	57.08	56.36
55.52	51.54	56.62	58.40	58.76	59.99
84.04	59.80	61.34	58.84	61.48	67.44
44.77	45.01	61.38	63.55	63.48	60.06
57.72	64.65	62.00	60.74	54.93	63.22
88.28	60.71	61.63	64.72	61.38	66.01
86.93	67.36	56.70	60.06	68.07	59.58
67.75	50.01	64.55	68.22	61.27	58.69
51.67	64.64	58.50	66.36	57.76	60.62
60.41	70.96	57.38	74.21	62.64	57.01
33.07	66.43	62.89	61.52	54.78	64.80
68.64	87.80	83.77	77.62	71.67	70.78
76.51	97.79	71.71	79.74	80.94	77.41
70.86	74.47	69.21	74.72	78.48	75.21
61.28	61.52	73.21	78.00	73.88	73.32
54.33	70.14	76.21	70.57	73.83	82.71
78.20	81.44	72.87	74.39	78.89	73.47
91.96	60.87	80.39	77.37	81.19	80.12
52.92	79.93	69.19	64.32	76.63	83.84
42.31	78.38	71.38	76.08	74.66	76.15
67.01	78.98	72.94	72.59	79.75	77.45
69.47	67.39	65.21	65.55	65.77	66.77
37.84	65.90	77.34	63.95	73.71	66.80
62.53	75.45	63.15	69.53	69.69	70.57
59.45	64.81	78.85	72.97	65.23	69.16
63.87	78.98	59.83	69.12	68.70	69.61
55.82	66.90	70.37	63.39	69.31	71.93
63.58	75.03	72.36	64.97	67.66	69.24
72.61	70.04	73.50	75.29	62.24	68.78
92.89	74.88	66.67	64.06	66.94	68.56
82.52	68.91	71.96	69.64	73.21	66.42
77.19	55.96	63.01	55.71	59.84	57.42
72.37	57.75	56.78	57.93	55.08	53.79
51.41	62.87	58.27	56.53	65.70	60.40
50.97	69.52	61.34	52.45	58.23	64.94
64.00	48.28	63.63	60.99	57.26	59.06
115.41	49.11	54.00	54.94	55.49	58.67
37.38	55.40	51.26	58.03	63.15	64.58
39.70	54.07	56.47	68.04	63.47	62.75
54.74	62.57	53.85	51.56	63.47	60.09
58.37	51.48	55.61	55.67	61.20	58.06
57.88	66.60	48.15	59.31	55.97	56.05
31.72	65.07	50.86	60.11	56.96	58.69
51.64	55.93	61.47	61.25	54.21	55.39
49.17	42.26	60.23	60.71	59.32	58.22

57.87	66.12	54.35	55.01	53.14	59.14
64.09	50.66	61.65	59.47	53.13	53.43
65.25	57.77	52.98	54.24	57.38	60.69
37.95	60.01	58.71	54.48	57.07	53.12
65.88	52.41	58.72	53.99	51.37	52.81
61.36	52.91	62.34	52.64	58.74	54.72
61.60	76.64	74.50	93.56	82.85	90.08
105.29	90.05	75.40	79.83	84.51	88.67
84.95	80.06	86.10	83.89	88.19	96.45
75.09	76.09	84.87	85.29	90.19	92.72
102.98	85.38	79.87	77.55	81.69	86.93
97.60	107.17	87.55	85.06	93.59	88.09
83.30	78.18	86.15	85.62	80.38	92.85
58.55	86.10	72.86	90.92	85.91	90.25
89.37	89.29	82.81	95.44	81.05	85.64
55.48	90.50	87.02	81.07	89.91	79.17
59.48	48.54	56.33	61.93	49.36	54.43
39.88	52.05	50.49	48.35	57.69	51.76
52.17	54.23	51.01	51.09	55.25	51.46
33.45	59.96	49.83	48.67	57.55	54.03
30.04	56.92	49.23	54.28	52.15	49.76
42.63	52.87	52.64	49.47	52.06	43.95
76.75	43.59	47.81	55.50	54.45	52.58
63.60	51.87	49.43	49.73	49.79	49.76
43.50	75.32	43.80	47.47	52.14	50.74
34.39	63.96	48.38	46.89	49.52	49.07
69.40	61.25	58.90	54.59	59.19	61.91
59.87	61.53	53.41	62.60	60.77	57.80
79.48	70.28	54.40	62.34	59.52	59.89
60.95	61.79	61.13	59.05	56.73	60.37
69.38	48.33	60.42	52.48	60.11	60.83
59.68	57.16	53.41	62.61	53.54	60.67
27.50	70.45	67.18	61.89	60.22	61.85
68.37	56.11	59.65	61.15	61.84	58.32
30.26	64.36	62.51	61.56	57.41	64.87
61.45	64.23	65.55	55.15	57.54	59.57
49.45	46.81	59.46	59.53	60.39	57.89
41.24	50.40	50.43	53.85	57.13	58.83
86.83	58.55	46.20	62.79	60.67	56.38
61.52	60.14	51.24	54.43	58.24	59.86
73.18	52.79	58.57	54.92	58.01	60.74
56.68	58.04	62.16	53.44	53.60	56.31
78.84	64.65	55.66	52.97	55.65	58.93
59.35	53.59	48.14	55.03	56.03	58.23
73.16	58.81	55.81	58.73	51.49	60.22
49.34	59.61	57.29	55.57	55.79	58.97

C.1.4 The Error of Imputation of IMLS-1 Method

The Proportion of Missing					
1%	5%	10%	15%	20%	25%
62.45	58.57	63.38	59.04	62.26	61.78
70.55	49.89	59.45	58.38	54.27	58.59
47.06	75.19	57.79	71.42	56.81	58.71
63.62	47.96	57.00	67.19	56.78	57.78
58.15	57.69	55.38	55.96	62.70	55.60
50.34	55.69	60.09	61.10	55.52	67.12
56.45	58.14	66.85	57.43	59.93	60.52

60.53	55.98	57.41	55.87	57.78	55.16
84.90	63.81	67.71	55.85	56.42	59.93
57.37	62.21	63.41	61.28	56.97	56.58
55.57	51.62	56.69	58.44	58.61	60.12
83.94	59.92	61.26	58.81	61.56	67.38
44.60	44.96	61.47	63.56	63.57	60.10
57.48	64.72	62.06	60.83	54.92	63.18
88.30	60.67	61.62	64.68	61.32	66.04
87.17	67.47	56.82	60.15	68.08	59.63
67.64	50.06	64.59	68.23	61.23	58.72
51.70	64.72	58.52	66.39	57.69	60.68
60.54	70.95	57.44	74.21	62.58	57.03
33.23	66.52	62.89	61.48	54.87	64.90
66.89	87.50	84.64	78.37	72.16	70.54
79.29	97.59	72.06	79.85	80.73	77.68
69.81	73.76	68.86	75.12	78.69	74.27
60.61	61.95	73.18	78.26	74.31	73.73
56.52	69.54	76.46	70.82	74.49	83.94
77.67	82.63	73.25	74.44	79.49	74.74
93.30	60.96	80.19	77.42	81.98	80.74
54.56	79.94	70.60	64.85	75.20	84.24
42.68	79.66	71.90	76.36	75.69	75.89
70.19	79.82	73.20	73.21	81.69	77.67
69.67	67.36	65.22	65.51	65.21	66.59
37.92	65.85	77.33	63.93	73.44	66.82
62.71	75.64	63.24	69.75	69.78	70.11
59.38	64.84	78.89	72.94	65.28	69.44
63.82	79.07	59.89	69.16	68.88	69.75
56.02	67.06	70.52	63.34	69.00	71.89
63.75	75.07	72.47	64.97	67.63	69.54
72.78	70.10	73.58	75.18	62.28	68.56
92.90	75.13	66.79	64.18	67.12	68.56
82.72	69.02	71.98	69.07	73.36	66.29
76.88	55.93	62.99	55.67	59.66	57.32
72.42	57.66	56.78	57.75	54.96	53.75
51.23	62.74	58.26	56.56	65.46	60.32
50.80	69.41	61.33	52.25	58.62	64.77
63.52	48.32	63.53	60.98	57.25	59.16
115.05	49.08	53.90	54.76	55.36	58.77
37.30	55.33	51.22	57.92	63.32	64.45
39.56	54.03	56.54	68.05	63.25	62.66
54.54	62.53	53.84	51.35	63.45	59.95
58.34	51.42	55.55	55.59	61.20	58.01
57.99	66.62	48.04	59.28	55.97	56.05
31.55	65.16	50.77	60.06	57.03	58.66
51.79	55.96	61.48	61.20	54.20	55.36
49.15	42.27	60.19	60.68	59.32	58.18
57.97	66.19	54.42	55.03	53.08	59.11
64.11	50.64	61.68	59.47	53.11	53.37
65.20	57.81	52.92	54.25	57.36	60.59
38.00	59.97	58.71	54.36	57.22	53.06
65.75	52.38	58.75	53.96	51.29	52.84
61.43	52.89	62.40	52.64	58.71	54.70
58.34	73.78	73.09	93.58	80.12	87.34
107.46	91.83	75.08	78.29	84.35	85.36
84.29	80.80	84.77	91.12	87.91	96.24
75.93	74.60	97.05	84.34	89.74	92.34
101.03	82.87	80.23	80.60	82.75	86.15
95.89	106.85	87.18	84.92	93.20	88.24
82.40	78.80	85.33	84.93	81.60	92.61
49.58	82.10	77.85	89.87	84.78	86.07

91.40	90.47	88.12	94.07	84.19	89.51
54.63	88.19	86.36	80.67	89.01	78.88
59.39	48.56	56.28	61.92	49.43	54.44
39.87	52.03	50.52	48.36	57.67	51.85
52.17	54.25	51.03	51.24	55.29	51.35
33.48	59.93	49.83	48.65	57.61	53.98
30.04	56.94	49.24	54.31	52.17	49.79
42.61	52.95	52.59	49.55	51.99	43.99
76.95	43.59	47.83	55.41	54.42	52.67
63.70	51.96	49.47	49.72	49.81	49.86
43.53	75.39	43.82	47.52	52.24	50.67
34.55	64.05	48.43	46.97	49.42	49.08
69.54	61.28	58.90	54.71	59.14	61.94
59.81	61.59	54.16	62.59	60.77	57.97
79.49	70.40	54.41	62.54	59.53	60.01
61.00	61.77	61.17	59.09	56.80	60.39
69.41	48.26	60.35	52.54	60.10	60.60
59.72	57.18	53.50	62.68	53.56	60.79
27.55	70.46	67.24	61.98	60.21	61.86
68.31	56.07	59.67	61.24	61.86	58.48
30.23	64.40	62.53	61.71	57.46	64.86
61.43	64.28	65.64	55.24	57.54	59.58
49.37	46.80	59.54	59.56	60.49	57.99
41.26	50.39	50.48	53.83	57.19	58.60
86.63	58.46	46.25	62.93	60.80	56.32
61.51	60.16	51.15	54.57	58.50	60.06
73.00	52.85	58.69	54.98	58.07	60.95
56.53	58.03	62.22	53.59	53.66	56.34
78.95	64.65	55.75	52.99	55.62	58.95
59.32	53.68	48.11	55.07	56.07	58.30
73.12	58.93	55.82	58.83	51.55	60.19
49.47	59.53	57.43	55.66	55.85	58.85

C.1.5 The Error of Imputation of IMLS-4 Method

The Proportion of Missing

1%	5%	10%	15%	20%	25%
18.29	14.07	17.07	17.01	16.62	17.95
14.74	11.15	12.61	16.34	13.58	17.27
8.88	20.37	15.35	18.56	18.09	17.71
10.19	11.88	16.33	15.20	14.45	19.54
10.51	13.95	13.32	15.88	17.76	18.02
19.11	19.51	15.49	16.23	15.04	20.75
30.24	16.32	15.58	15.43	16.45	16.58
16.05	13.20	15.27	15.39	19.94	16.74
27.05	14.19	16.78	14.63	17.56	19.05
11.51	17.49	15.68	16.74	16.18	16.75
6.67	10.66	12.63	12.54	13.89	15.17
14.77	12.03	16.72	12.40	13.79	18.30
8.51	8.09	11.77	16.05	15.42	15.17
15.99	14.12	12.72	13.69	13.96	15.32
17.64	12.75	13.97	14.86	14.50	17.13
9.51	15.67	13.45	11.98	16.25	15.38
15.73	10.79	12.96	17.10	14.11	15.67
8.81	14.58	14.22	15.71	13.21	16.38
17.71	13.94	12.23	15.09	14.36	14.90
4.96	11.71	13.57	14.63	12.41	15.69
18.13	16.63	18.68	25.24	21.44	22.18

15.88	32.59	16.88	20.15	23.38	24.39
17.35	18.42	17.04	23.43	21.87	22.85
8.05	15.09	16.91	22.50	21.26	25.55
12.68	19.55	21.67	21.11	23.20	31.30
28.33	21.66	17.27	19.33	21.84	25.56
23.85	16.47	21.03	21.63	23.28	24.78
13.21	18.28	18.85	18.84	24.47	31.43
15.60	15.49	19.87	20.53	24.71	23.63
13.66	23.73	19.84	21.54	28.60	19.74
9.61	15.37	18.05	18.52	20.95	21.48
6.12	17.90	19.06	17.62	23.60	23.11
12.00	18.74	15.62	19.81	20.89	23.18
14.89	13.26	20.76	19.61	17.20	27.55
13.18	24.91	18.42	18.91	21.19	23.70
12.48	17.49	18.84	16.53	21.93	24.13
11.51	20.05	16.70	17.45	22.48	22.25
29.75	17.82	19.54	17.63	19.71	23.93
11.85	15.41	18.91	18.02	23.59	23.45
18.39	19.79	17.13	22.99	23.76	24.29
37.62	17.36	17.77	19.11	21.86	20.29
15.35	18.22	17.41	17.04	18.58	18.86
27.84	19.72	18.56	17.06	21.04	20.86
12.02	25.94	18.61	16.00	18.48	21.84
19.25	14.89	20.27	19.97	18.08	19.37
31.90	16.46	16.37	17.41	18.78	21.88
10.21	18.86	14.05	19.92	21.18	21.04
9.71	13.80	17.63	21.69	20.27	21.34
16.78	18.89	17.81	17.25	19.62	18.74
18.85	16.85	15.29	18.03	20.69	20.17
10.59	18.75	19.97	13.05	14.96	15.46
10.08	16.02	11.43	13.74	15.12	15.88
10.40	16.65	13.80	15.04	13.46	15.09
13.40	9.92	13.98	15.20	14.45	13.71
6.29	15.74	15.22	11.93	12.61	16.26
15.57	10.53	14.45	16.38	14.30	14.48
19.31	15.29	12.93	13.16	16.88	16.98
15.13	13.78	12.21	14.67	13.00	15.73
12.31	13.47	16.56	13.36	12.89	13.34
12.21	13.62	14.23	12.11	14.29	13.78
18.98	24.12	24.43	29.61	26.90	34.55
41.01	26.07	18.80	24.68	27.85	31.23
35.52	23.58	23.65	30.74	30.22	35.84
17.89	17.79	33.30	28.18	37.47	33.48
30.37	33.01	20.58	23.70	29.71	30.95
29.04	34.63	25.97	30.68	32.00	29.27
13.25	18.72	27.21	30.67	27.83	34.56
17.42	22.35	19.06	28.30	32.85	31.99
22.84	24.54	33.14	31.37	38.94	36.80
23.40	24.48	30.92	27.68	29.96	28.45
24.00	18.47	21.87	26.82	21.86	23.40
12.26	17.44	19.70	21.59	26.25	25.12
33.08	19.51	19.38	22.17	26.06	25.28
13.43	22.39	21.52	20.92	23.84	26.11
9.39	18.14	22.97	22.30	24.25	24.01
13.82	22.59	24.88	19.00	21.73	20.26
27.42	16.43	18.59	22.43	23.67	25.08
16.81	15.84	20.25	19.26	21.24	22.80
11.23	37.24	20.42	20.36	23.73	22.27
11.20	21.18	20.57	18.20	23.49	22.61
14.02	15.11	16.13	16.19	20.69	22.25
12.50	15.15	15.26	20.76	17.09	20.97

17.00	22.46	19.04	20.37	18.88	20.92
13.40	15.25	18.90	20.35	18.77	22.01
12.99	14.06	17.33	17.27	19.07	22.74
12.79	17.61	14.71	18.61	17.37	21.57
10.52	21.85	20.56	19.03	22.13	21.91
16.28	20.99	18.01	19.49	20.57	20.10
10.64	20.80	18.22	18.57	17.43	22.52
19.46	15.45	17.94	19.81	21.06	21.92
11.88	14.99	17.62	17.39	19.13	19.02
9.98	18.72	15.15	17.76	18.98	20.44
29.27	16.88	14.45	18.08	25.73	18.95
18.91	18.10	14.76	19.65	19.53	22.28
27.29	14.79	16.02	15.19	18.93	21.35
16.82	18.00	16.52	17.35	17.27	18.59
24.63	17.52	19.11	17.68	18.83	19.75
14.14	13.78	16.12	18.15	18.69	20.26
31.45	16.57	17.60	17.64	17.49	21.00
14.34	16.60	15.76	16.80	19.13	19.50

C.1.6 The Error of Imputation of Mean Method

The Proportion of Missing					
1%	5%	10%	15%	20%	25%
98.42	94.34	91.59	93.73	92.20	91.54
87.71	94.64	88.36	92.22	91.35	91.13
92.37	85.23	90.84	90.50	88.04	91.30
85.86	90.65	96.85	89.10	95.92	91.99
80.50	92.47	92.28	95.23	92.43	94.16
95.27	85.30	90.52	92.41	94.45	92.08
93.16	98.51	94.14	92.78	92.00	90.87
104.19	92.26	92.07	93.83	94.52	92.16
85.51	93.52	91.45	90.93	90.25	92.07
88.94	89.60	93.44	92.27	92.48	88.46
86.42	96.27	91.22	94.87	90.27	93.16
96.62	83.08	91.16	90.52	90.28	89.14
90.92	94.81	90.42	87.47	89.03	91.52
98.56	90.70	91.18	96.89	90.08	92.78
102.86	91.52	91.51	90.15	91.70	89.61
77.78	92.44	94.82	89.33	89.72	90.79
115.56	95.64	87.07	91.19	90.55	88.98
85.53	91.33	93.70	92.24	90.07	91.02
108.35	88.65	88.77	88.61	91.45	91.63
96.60	92.05	90.89	95.46	94.49	90.51
71.00	79.50	74.85	77.88	74.06	77.02
68.59	95.35	78.22	79.67	80.78	79.51
93.36	87.58	83.73	78.00	78.18	79.24
76.96	80.25	77.44	75.85	77.76	75.61
73.86	87.85	80.06	76.04	77.67	76.72
61.89	83.28	78.79	76.10	76.85	78.71
57.92	85.52	79.96	80.20	78.93	75.97
80.03	77.23	76.51	76.49	78.70	76.31
79.49	80.25	72.11	78.58	77.81	77.02
73.65	78.00	80.58	78.43	80.86	76.71
85.35	90.15	87.76	88.97	88.52	91.38
104.71	91.41	84.83	87.47	84.91	87.07
96.17	85.86	85.64	84.20	85.12	84.29
89.90	87.30	85.22	83.54	83.38	87.16
76.18	90.24	90.05	87.40	84.52	85.47

79.74	95.59	87.49	91.41	84.94	84.01
74.42	92.28	82.84	88.70	86.47	87.29
82.11	82.44	86.41	87.12	86.30	86.57
68.44	88.27	81.46	88.80	86.24	87.93
82.73	84.07	86.72	85.74	84.83	87.29
89.51	94.52	98.33	98.84	97.15	98.30
97.82	95.47	97.36	97.46	98.80	96.65
110.10	99.20	98.48	98.03	99.44	98.47
95.16	100.64	96.91	98.70	97.49	97.65
88.03	96.77	97.04	96.80	99.54	96.03
94.15	98.86	101.05	95.95	97.40	97.01
96.38	96.94	98.54	96.72	98.82	97.88
94.30	97.42	100.03	96.93	97.45	97.86
105.37	95.26	97.83	96.89	97.19	98.87
107.43	97.41	99.58	99.77	96.29	98.40
82.12	88.86	91.97	90.39	89.62	88.92
106.81	92.15	90.04	88.71	90.14	86.88
79.58	91.36	86.00	87.53	92.12	91.20
99.11	94.39	89.41	87.56	88.77	87.73
73.50	95.73	90.17	91.86	90.00	84.75
82.47	84.36	87.87	89.60	89.31	88.57
90.91	94.38	92.04	90.95	91.56	89.46
96.54	89.94	86.11	89.78	84.84	91.91
99.53	93.30	93.40	89.60	90.74	88.99
83.84	86.47	86.06	86.17	87.64	88.35
95.05	87.33	84.17	84.36	84.62	79.77
79.69	74.81	82.50	80.79	78.31	81.17
83.68	77.74	83.61	78.28	80.31	82.88
73.11	86.94	74.77	80.37	84.10	84.17
94.38	88.86	79.85	77.57	82.00	83.02
82.03	81.09	74.19	81.09	85.56	83.89
88.88	77.26	86.27	84.90	82.20	80.05
99.70	90.04	81.57	85.19	82.84	80.61
79.46	82.37	82.97	86.39	82.71	79.92
104.09	89.96	79.67	78.88	83.67	83.93
100.74	97.56	96.76	96.66	96.19	95.81
106.86	91.77	93.40	93.79	97.32	95.68
119.67	98.56	95.32	93.45	94.03	97.50
99.09	92.36	96.07	97.67	95.19	98.26
104.93	94.39	94.20	95.34	95.52	95.07
98.76	93.88	98.77	95.43	95.11	93.82
93.73	99.86	96.32	96.54	96.49	95.37
105.82	95.94	98.26	93.30	95.01	96.25
89.97	92.92	92.82	96.96	97.17	96.27
93.35	100.57	96.12	98.95	96.52	96.18
83.20	93.33	90.06	92.02	92.42	91.39
97.88	93.44	91.70	95.28	89.90	91.91
87.35	93.76	96.09	89.25	93.87	92.09
86.75	94.85	90.43	94.65	94.97	91.74
95.72	91.75	92.08	92.51	93.38	93.68
91.73	96.17	94.97	94.19	93.01	91.23
87.50	98.89	90.53	94.36	91.76	92.51
86.25	95.86	92.81	93.40	91.91	93.68
94.59	96.09	97.01	92.36	91.27	90.91
101.06	95.15	90.38	92.05	94.54	91.98
86.91	99.89	91.59	94.24	93.68	91.48
92.64	95.77	92.84	93.43	96.31	91.33
99.30	94.86	92.33	93.89	92.35	95.49
92.60	93.36	95.58	93.48	91.53	93.88
102.35	96.42	92.50	95.80	92.40	92.38
92.89	91.37	93.95	90.77	92.87	92.32

100.51	89.11	93.87	92.27	94.10	92.80
75.66	97.82	95.35	91.65	96.89	92.66
82.43	94.73	93.31	93.13	92.34	93.28
95.11	90.49	90.66	95.26	94.79	94.44

C.1.7 The Error of Imputation of N-ILS Method

The Proportion of Missing					
1%	5%	10%	15%	20%	25%
14.86	8.94	10.13	8.20	9.46	11.21
6.00	6.21	8.81	7.50	7.38	3598.15
8.55	11.80	10.55	10.15	9.35	9.61
7.33	7.04	8.51	10.91	8.77	10.83
7.26	9.15	8.26	9.05	10.39	10.05
17.53	12.28	8.43	9.08	8.57	11.62
12.79	7.45	8.80	9.44	9.52	9.62
7.15	7.53	8.98	8.50	11.01	2037.53
16.19	8.19	9.61	7.85	8.89	63266.52
9.22	10.84	9.21	9.41	8.54	9.41
4.98	6.06	7.56	6.41	7.28	7.27
11.96	5.61	8.86	6.92	6.19	8.02
3.70	5.91	5.59	9.25	7.87	7.19
14.26	7.77	8.46	6.99	6.87	7.64
8.90	6.92	6.34	7.29	7.23	7.69
7.94	8.84	7.07	7.75	7.14	7.42
6.78	4.78	5.83	9.28	7.27	4854.48
8.36	7.10	7.18	8.23	7.11	8.31
8.51	6.35	5.94	6.59	8.93	8.11
6.58	7.62	6.29	9.01	7.25	8.65
7.14	7.19	7.54	7.35	8.57	7390.58
4.95	8.59	7.72	6.96	7.48	8.06
6.92	5.95	5.83	7.72	9.58	9.55
5.18	4.59	6.38	7.88	7.31	21411.86
5.22	8.06	8.46	6.74	7.06	9.21
8.81	8.98	6.59	6.79	6.93	6852.42
10.36	7.58	7.78	7.41	7.94	8.37
4.12	9.63	6.09	5.77	7.94	9.59
2.91	6.42	8.91	6.91	7.29	9.99
3.62	11.34	7.05	8.15	7.62	7.79
10.03	8.04	8.30	9.14	9.42	13.11
6.20	8.63	9.12	9.63	8.37	7306.50
3.93	9.45	7.75	7.39	8.90	672.81
6.34	7.56	9.29	7.89	7.42	17310.95
8.51	10.22	9.94	8.98	9.56	9.84
5.94	10.45	8.30	8.13	8.42	12.73
9.28	11.81	8.53	7.58	9.07	9.00
9.31	9.91	9.53	8.11	9.15	10.36
5.25	9.44	8.57	8.18	9.19	9.14
9.87	11.11	8.00	9.85	9.84	10.07
8.97	8.65	7.91	10.04	8.46	10.04
6.34	8.26	7.14	9.69	8.90	8.89
11.76	7.79	7.07	8.12	9.44	9.89
9.19	10.78	7.43	7.65	10.10	9.72
9.36	7.45	9.96	8.92	8.50	8.74
8.10	7.88	8.20	7.98	9.81	11.30
2.92	8.53	7.76	8.88	8.86	10.20
4.73	5.66	7.77	11.24	9.92	2746.32

```
 9.62      9.84      9.62      8.14      8.27     1438.52
 9.18       7.14       7.26       8.68       7.82       8.53
 7.41       10.02      6.66       8.79       6.86       8.46
 5.85      9.48      7.33      7.00     7.81     11691.34
 4.29       9.02       7.61       9.03       7.16       7.65
 5.63       6.12       8.88       8.31       8.27       8.36
 6.43       8.63       8.62       7.68       7.47       7.87
 6.67       6.11       8.38       8.41       7.93       7.92
 3.43       5.41       7.78       7.07       8.09       8.99
 8.32       8.10       6.82       7.45       7.36       8.15
 4.76       5.94       9.26       7.83       6.96       7.31
 7.27       7.83       7.21       7.00       7.29       7.50
 6.44       6.17       6.69       6.00       7.50       9.19
10.35       6.35       5.37       6.47       6.42       6.71
 7.99       7.45       4.88       6.15       6.17       7.48
 5.34       3.79       6.61       5.00       7.14       6.95
11.73      6.17      6.29      4.99      6.36     1309.46
 8.17       10.60      6.21       6.09       6.62       7.23
 3.99       4.49       8.34       5.45       5.01       7.38
 3.00       4.27       5.51       7.61       8.13       6.41
 3.50       6.84       7.23       6.64       5.78       7.81
 3.29      5.94      6.26      5.20      7.39     2374.65
 3.59      7.63      8.65      10.70      8.02      10.41
 5.80      9.10      7.35      8.53      13.49      462.32
21.62       9.38       7.53       7.69       9.33       12.43
 4.37       9.49       7.54       7.82       9.11       10.71
 3.73      8.75      11.42      8.46     9.52     10148.28
 4.91       8.93       8.86       9.05       9.79       10.17
12.90       6.54       8.02       8.91       10.21       8.72
 6.08       8.03       9.65       7.31       7.92       10.20
 5.87      11.54      6.18     7.17     29912.83      10.23
 4.12      8.58      9.74     7.91     10.57     23598.24
 8.22       6.08       6.81       4.76       6.89       6.74
 4.17       6.52       5.57       8.12       6.18       7.01
 7.42       8.13       5.85       5.38       7.16       7.39
 3.19       5.22       5.53       6.33       5.82       7.17
 6.94       6.17       6.23       6.72       5.79       7.05
 7.79       7.01       6.26       6.96       6.77       7.35
 2.52       7.27       6.48       6.98       6.97       7.88
 8.00       5.70       5.12       5.87       5.81       6.79
 2.71       10.27      7.40      5.79     10216.99      8.32
 4.65       6.50       6.56       6.76       6.26       6.97
 5.23       4.71       6.82       6.28       7.14       6.47
 3.39       5.24       5.45       6.91       7.77       6.05
12.36       6.89       6.08       6.76       7.27       8.16
 7.88       6.51       5.47       7.56       7.36       7.57
11.46       5.26       6.69       6.34       6.20       7.02
 6.06       5.80       6.29       5.99       6.52       6.82
 6.60       6.19       7.00       6.53       6.60       7.49
 4.50       5.95       6.17       5.57       6.99       6.66
 8.04       6.91       5.50       6.57       6.45       7.58
 9.75       5.48       4.55       6.42       7.43       7.30
```

C.1.8 The Error of Imputation of N-IMLS Method

```
                   The Proportion of Missing
---------------------------------------------------------
   1%         5%         10%        15%        20%        25%
---------------------------------------------------------
14.84      8.95       10.13      8.20       9.46       11.04
```

5.99	6.21	8.78	7.47	7.39	10.26
8.56	11.79	10.55	10.14	9.35	9.59
7.34	7.02	8.50	10.85	8.77	10.79
7.27	9.14	8.25	9.01	10.37	10.02
17.52	12.27	8.43	9.06	8.54	10.73
12.76	7.45	8.78	9.41	9.46	9.60
7.13	7.52	8.96	8.48	11.00	9.91
16.18	8.19	9.58	7.83	8.90	11.00
9.16	10.83	9.21	9.40	8.52	9.36
4.98	6.03	7.55	6.41	7.27	7.24
11.94	5.59	8.87	6.90	6.18	8.04
3.70	5.91	5.59	9.25	7.84	7.18
14.21	7.76	8.45	7.00	6.86	7.62
8.91	6.93	6.34	7.30	7.25	7.67
7.92	8.84	7.07	7.74	7.14	7.43
6.76	4.78	5.83	9.27	7.28	6.98
8.33	7.12	7.18	8.22	7.14	8.25
8.52	6.39	5.94	6.55	8.92	8.08
6.58	7.61	6.30	9.02	7.27	8.61
7.14	7.21	7.52	7.37	8.55	8.39
4.97	8.60	7.73	6.97	7.44	8.03
6.87	5.93	5.87	7.74	9.57	9.55
5.18	4.59	6.37	7.86	7.29	10.93
5.25	8.06	8.44	6.73	7.04	9.20
8.82	8.98	6.58	6.79	6.91	8.88
10.35	7.55	7.77	7.41	7.92	8.36
4.18	9.62	6.08	5.78	7.93	9.55
2.92	6.42	8.91	6.92	7.29	10.14
3.63	11.32	7.05	8.15	7.76	7.77
9.98	8.04	8.29	9.13	9.42	12.62
6.20	8.62	9.12	9.61	8.40	9.70
3.94	9.43	7.75	7.38	8.87	8.99
6.32	7.57	9.27	7.89	7.41	11.11
8.50	10.20	9.92	8.96	9.64	9.81
5.94	10.43	8.29	8.13	8.40	13.26
9.26	11.80	8.51	7.58	9.07	8.96
9.31	9.90	9.48	8.08	9.16	10.31
5.27	9.43	8.57	8.18	9.18	9.18
9.88	11.11	7.98	9.85	9.72	10.00
8.96	8.64	7.89	10.03	8.45	10.01
6.32	8.25	7.10	9.69	8.89	8.88
11.73	7.81	7.08	8.12	9.36	9.89
9.17	10.78	7.42	7.65	10.08	9.69
9.38	7.43	9.95	8.93	8.47	8.72
8.12	7.88	8.17	7.98	9.83	11.24
2.91	8.50	7.79	8.86	8.86	10.03
4.74	5.65	7.76	11.17	9.92	9.76
9.64	9.85	9.59	8.14	8.27	8.88
9.19	7.14	7.25	8.69	7.79	8.57
7.43	10.03	6.67	8.78	6.85	8.39
5.86	9.48	7.34	6.98	7.79	8.10
4.29	9.00	7.60	9.02	7.17	7.63
5.63	6.11	8.88	8.30	8.26	8.31
6.43	8.61	8.62	7.69	7.47	7.87
6.66	6.10	8.36	8.40	7.94	7.89
3.45	5.42	7.81	7.07	8.07	9.02
8.31	8.11	6.82	7.44	7.33	8.13
4.76	5.95	9.26	7.84	6.95	7.31
7.27	7.84	7.17	6.98	7.27	7.49
6.43	6.17	6.69	5.97	7.51	8.30
10.35	6.35	5.35	6.45	6.41	6.73

7.99	7.42	4.90	6.17	6.20	7.47
5.32	3.78	6.61	5.00	7.13	6.94
11.72	6.17	6.30	4.97	6.34	7.80
8.15	10.63	6.20	6.10	6.64	7.25
3.98	4.49	8.32	5.45	4.97	7.35
3.00	4.26	5.51	7.58	8.03	6.38
3.50	6.82	7.08	6.62	5.78	7.79
3.29	5.93	6.25	5.20	7.38	11.98
3.58	7.62	8.68	10.67	8.03	10.35
5.78	9.08	7.34	8.50	13.46	11.99
21.68	9.39	7.54	7.65	9.32	12.41
4.40	9.49	7.55	7.79	9.07	10.61
3.71	8.76	11.37	8.45	9.58	8.57
4.89	8.89	8.87	9.06	9.73	10.10
12.86	6.53	8.03	8.90	10.18	8.74
6.06	8.01	9.65	7.29	7.91	9.75
5.82	11.55	6.17	7.15	10.26	10.17
4.11	8.59	9.73	7.89	10.54	9.17
8.18	6.08	6.79	4.76	6.90	6.73
4.16	6.52	5.57	8.11	6.16	6.98
7.40	8.12	5.85	5.36	7.16	7.36
3.18	5.22	5.52	6.32	5.81	7.17
6.96	6.17	6.22	6.70	5.78	7.01
7.82	6.99	6.27	6.94	6.81	7.32
2.52	7.25	6.48	6.98	6.94	7.89
8.02	5.68	5.13	5.86	5.81	6.77
2.71	10.24	7.39	5.78	6.33	7.96
4.64	6.49	6.54	6.75	6.25	6.87
5.24	4.71	6.81	6.25	7.10	6.45
3.37	5.23	5.44	6.89	7.77	6.05
12.35	6.89	6.07	6.76	7.26	8.15
7.86	6.50	5.47	7.55	7.33	7.58
11.46	5.25	6.67	6.34	6.17	7.00
6.06	5.80	6.27	5.98	6.53	6.80
6.58	6.19	7.00	6.52	6.59	7.46
4.51	5.96	6.17	5.56	6.99	6.64
8.01	6.90	5.50	6.56	6.45	7.58
9.75	5.48	4.55	6.42	7.43	7.28

C.1.9 The Error of Imputation of INI Method

The Proportion of Missing					
1%	5%	10%	15%	20%	25%
14.66	8.04	10.42	8.91	9.41	10.87
5.88	6.07	9.01	10.99	7.14	10.68
8.99	12.28	9.87	9.45	9.89	9.00
6.47	7.03	8.95	9.01	9.10	11.02
7.61	8.73	7.86	9.53	10.48	10.63
16.39	12.34	8.39	9.58	8.63	12.91
13.14	6.78	8.71	9.51	9.84	9.79
6.94	7.30	8.88	8.54	11.22	9.80
16.48	8.49	10.46	8.83	9.74	10.78
9.42	11.95	9.30	9.67	8.84	9.99
4.96	5.03	7.24	6.32	7.36	7.69
11.89	5.74	8.85	6.33	5.84	8.11
3.89	6.01	5.52	8.24	7.29	7.06
12.99	7.15	8.14	7.50	6.55	8.29
8.90	6.97	6.19	7.35	6.59	8.57

7.87	9.35	6.38	7.33	7.99	8.34
7.12	4.74	6.02	8.65	7.20	7.47
8.15	6.73	6.78	8.16	6.49	8.48
8.77	6.07	5.84	6.95	8.18	7.92
5.58	7.67	6.54	8.17	7.09	9.47
7.23	7.30	7.46	9.64	9.28	9.41
6.34	10.21	8.23	8.21	8.43	10.00
6.97	5.76	7.08	8.72	10.18	10.97
4.68	4.70	6.73	8.96	10.44	11.48
5.02	8.43	8.89	7.08	8.46	10.58
9.99	10.08	7.54	7.93	7.80	10.08
10.57	8.61	8.18	8.78	8.87	9.23
4.51	9.50	7.02	7.00	10.21	12.10
3.26	6.42	8.93	7.61	12.54	9.45
3.58	11.20	7.49	8.75	11.38	7.55
8.88	7.58	7.62	9.62	8.73	11.76
6.00	8.56	9.94	9.94	9.20	12.73
4.35	9.45	7.71	7.65	9.59	11.95
7.71	7.16	9.53	9.16	8.68	11.04
8.20	9.98	9.83	9.32	9.24	15.11
5.92	10.85	8.33	7.93	9.00	11.92
8.97	11.74	8.81	8.10	11.80	9.81
10.08	9.18	9.57	8.11	9.19	13.17
5.89	9.58	8.66	8.30	11.83	12.51
10.06	11.25	8.42	10.02	11.22	11.23
10.35	8.60	7.35	8.99	9.19	9.89
5.49	8.46	7.35	9.01	9.91	9.55
10.04	7.78	7.79	8.58	9.92	10.36
8.21	9.97	8.05	7.23	10.37	9.96
8.66	7.05	9.42	8.81	8.73	8.67
9.11	7.75	7.98	7.87	9.01	10.58
2.75	8.25	7.17	9.07	8.52	11.13
5.05	6.03	7.50	11.61	10.44	10.35
8.96	8.77	9.86	7.84	7.97	8.52
8.72	6.28	6.83	8.33	8.33	8.35
7.08	9.78	6.63	8.34	7.07	8.09
5.59	9.88	6.58	7.45	7.88	9.32
4.56	9.09	8.17	8.74	7.38	7.52
5.88	6.05	8.83	8.34	8.36	8.15
6.14	8.91	8.03	7.25	7.76	9.61
7.47	6.04	8.45	9.23	7.60	7.48
4.67	6.10	7.95	7.30	8.61	10.03
8.52	8.23	7.17	7.08	7.16	7.93
5.12	5.82	9.51	7.84	6.85	6.90
7.71	7.72	7.76	6.90	8.24	7.68
7.36	6.24	6.41	7.32	12.98	14.71
11.06	6.21	5.80	6.72	7.95	8.93
7.88	7.82	6.00	8.85	8.55	12.88
6.52	3.76	6.27	8.43	14.59	16.03
12.89	6.64	6.96	6.12	8.22	12.91
7.54	13.75	7.13	8.41	8.29	8.24
4.04	4.99	8.68	6.15	8.63	10.11
3.86	4.35	6.16	7.99	12.60	9.42
3.96	7.38	7.71	7.18	15.09	9.69
4.06	5.71	6.73	7.09	9.08	12.19
4.31	7.76	9.07	10.49	8.74	11.19
5.96	9.20	7.70	7.99	12.63	11.54
21.15	9.00	8.31	9.27	9.33	12.00
4.63	9.80	6.98	9.03	10.05	11.28
3.92	9.00	10.43	9.53	9.30	9.40
4.93	9.31	12.61	10.05	9.93	8.87

12.04	7.18	7.41	8.66	10.12	10.03
6.47	7.38	8.88	7.00	10.96	9.66
5.78	13.60	7.36	7.13	11.52	9.45
4.52	9.27	8.34	7.10	10.68	10.30
9.39	6.07	6.79	5.38	7.55	8.48
6.13	6.20	6.07	8.94	7.34	8.37
7.35	8.92	6.72	6.25	9.35	7.94
3.44	7.46	6.50	7.84	7.56	9.43
7.16	6.00	6.97	7.40	7.14	11.83
8.21	7.36	6.90	7.26	6.98	8.99
2.82	8.19	8.92	8.71	8.25	8.91
7.67	5.69	5.23	6.97	7.74	8.51
3.16	12.67	7.83	6.86	6.60	11.05
9.74	6.66	6.71	8.94	7.75	9.77
5.11	4.96	6.57	6.33	7.46	6.24
3.47	5.24	5.49	6.86	7.26	7.20
11.49	6.50	6.05	6.52	7.57	7.90
6.94	6.43	5.19	7.00	7.16	7.70
11.00	6.03	6.20	6.51	6.90	6.97
5.71	5.74	6.36	5.83	6.62	7.67
9.19	6.10	6.50	6.51	6.84	7.15
3.90	5.89	6.41	5.90	6.44	7.16
8.01	7.04	5.56	6.42	6.49	7.32
9.43	5.64	4.89	6.75	6.88	7.58

C.1.10 The Error of Imputation of N-Mean Method

The Proportion of Missing

1%	5%	10%	15%	20%	25%
16.10	29.28	53.68	66.12	72.19	78.39
10.35	26.92	52.71	67.93	73.71	69.32
13.34	34.75	59.31	61.51	73.28	78.05
29.66	29.45	41.00	65.86	76.07	78.38
12.50	34.07	50.13	59.03	80.25	85.15
20.57	35.75	62.70	75.65	77.56	72.38
16.50	35.76	54.20	61.49	84.36	76.40
12.07	28.34	53.97	67.43	86.22	83.61
17.17	33.06	49.62	79.76	71.14	78.86
19.71	36.63	67.54	81.44	79.62	66.50
28.00	41.23	74.19	89.74	118.75	86.87
21.06	47.46	65.27	102.24	95.46	112.21
12.03	28.01	88.94	91.79	108.26	108.26
22.45	48.49	62.03	79.52	92.01	107.70
12.42	28.71	84.98	99.17	96.18	105.01
17.67	56.15	72.20	81.80	108.95	80.97
10.30	32.72	66.88	101.05	89.24	99.56
12.41	38.76	80.11	102.27	84.39	98.80
10.46	35.37	81.83	110.59	88.80	112.54
21.03	39.12	73.88	84.87	79.93	120.79
7.86	32.49	59.57	63.81	89.31	106.77
5.07	51.92	49.75	51.15	68.58	79.80
9.11	23.47	51.10	67.40	76.16	98.22
9.70	24.63	48.37	83.59	90.80	85.22
5.27	24.69	53.14	66.74	90.10	92.51
6.96	26.14	43.33	75.81	68.22	81.40
21.65	19.56	45.54	59.02	74.80	99.39
3.64	27.08	46.44	72.06	82.38	84.82
15.07	26.53	57.55	61.29	72.14	82.89

5.54	26.64	57.58	72.51	83.36	85.59
13.40	35.67	62.24	91.89	78.21	91.08
10.42	33.77	83.78	74.85	85.86	90.84
5.08	35.33	52.70	86.66	94.18	93.44
7.46	29.13	67.48	67.77	89.07	88.33
29.09	37.57	44.92	75.41	86.90	85.04
9.75	33.80	70.71	73.62	80.56	98.06
7.10	52.87	57.12	84.36	98.88	90.45
21.83	38.96	69.71	88.83	91.59	85.96
15.80	39.15	74.82	68.35	85.16	113.52
10.78	40.74	68.91	72.40	98.58	93.21
11.78	25.98	76.15	77.79	86.22	82.98
7.76	34.88	57.38	73.27	74.38	87.74
28.07	29.38	58.89	70.04	97.22	82.51
22.72	31.90	73.51	72.84	93.85	88.11
12.91	39.42	71.11	81.78	84.36	95.37
10.10	33.91	48.53	72.65	77.87	101.84
15.74	37.94	49.54	72.16	83.31	85.60
8.69	31.47	65.46	80.54	89.08	86.65
21.51	35.98	55.28	73.39	93.39	84.79
18.44	30.63	53.05	74.90	88.71	95.22
16.05	41.88	67.53	93.41	99.20	122.92
9.79	48.67	65.05	91.08	96.79	104.05
9.68	38.47	79.39	95.91	95.04	159.45
17.27	35.45	83.38	124.24	108.01	109.93
16.20	45.04	63.04	86.71	93.93	113.72
10.72	40.39	54.21	104.30	84.13	105.77
17.45	46.21	68.12	118.74	122.20	127.07
21.86	28.78	78.50	102.38	110.44	131.45
17.35	64.08	73.17	82.91	118.14	91.11
21.75	40.73	71.54	99.28	102.92	124.55
6.95	27.57	55.29	77.23	91.51	93.13
10.39	45.69	63.81	78.87	76.50	78.95
16.66	22.05	47.47	59.79	85.70	90.00
5.23	30.86	61.55	87.55	95.40	95.20
12.90	30.72	63.10	71.57	90.17	90.55
8.38	29.02	50.19	64.80	89.43	96.66
6.21	55.22	57.47	66.42	80.60	84.50
5.42	31.90	52.64	87.17	78.68	110.76
19.04	18.32	65.73	83.52	80.96	77.20
12.91	44.50	74.90	77.13	73.63	89.67
6.91	32.13	49.43	66.45	63.65	75.75
8.56	24.64	57.27	58.25	80.06	75.65
25.66	30.75	35.91	64.64	71.10	75.02
19.71	51.34	51.97	54.68	84.95	84.85
11.55	36.43	41.83	82.32	83.26	76.23
18.00	28.17	56.61	59.83	66.59	69.79
15.04	28.40	44.97	83.01	92.24	79.02
18.47	29.12	46.93	73.28	74.60	82.68
7.41	35.62	50.36	58.75	87.10	79.36
6.92	29.02	42.65	62.69	78.94	94.96
10.25	30.17	70.13	81.12	106.40	107.50
19.39	32.84	72.26	107.39	107.12	115.14
7.26	25.08	59.24	118.23	107.43	110.01
3.70	29.15	66.68	100.52	108.55	110.72
26.11	41.61	70.79	74.81	108.75	123.74
13.65	34.01	78.68	110.21	94.02	106.36
7.42	51.19	86.16	101.77	95.19	109.00
12.52	37.14	77.21	119.19	115.00	104.79
35.31	37.85	75.96	98.75	108.80	122.42
12.48	27.23	54.58	100.69	94.49	105.17

18.62	36.98	94.28	96.31	103.88	104.59
25.10	29.31	62.98	121.25	115.50	146.77
14.07	48.99	87.74	112.57	117.49	129.24
23.28	37.35	75.98	125.79	105.61	121.74
26.92	43.99	70.41	105.38	126.73	122.66
12.75	50.62	77.48	112.14	103.16	112.03
13.03	42.03	78.76	95.79	106.35	134.86
9.55	39.20	72.26	86.31	111.47	118.29
18.01	51.17	94.00	130.07	108.79	132.46
26.16	46.30	80.95	115.48	120.69	109.29

C.2 The Results of Experiments with Marketing Database

C.2.1 Errors for Different Data Samples with 1% Missing

Error of Imputation (in %)

INI	EM-Strauss	EM-Schafer
35.06	245.42	889.56
57.94	32.84	35.14
34.71	128.90	1002.45
39.33	183.30	542.80
18.34	204.06	1235.70
241.97	1005.38	1905.87
43.37	202.68	370.86
22.58	737.89	235.80
165.25	1740.86	5656.27
18.31	8.57	121.01
65.26	51.89	21323767000.00
97.77	302.13	106.55
432.95	226.58	102.41
870.39	2027.37	100.00
87.17	74.41	1139386100.00
87.27	101.42	14722486.00
358.48	2151.72	15253037.00
65.50	201.41	100.00
76.23	52.13	1441.58
46.90	17.11	127.59
50.23	706.56	4912.40
16.39	371.91	3448.14
56.61	229.73	404.24
35.22	27.81	9.04
163.24	153.27	256.93
38.64	1.63	40.61
71.15	121.77	599.85
307.36	242.52	110.47
27.30	307.36	2440.13
4.90	18.02	185.28
54.04	127.86	1171.20
100.02	70.28	NaN
54.43	353.28	5046.96
57252.87	55878.16	374457.91
44.82	16.80	111.82
124.52	381.68	864.16

34.90	119.91	70.29
4398.05	3573.77	4241.88
81.74	249.72	313.01
62.66	42.27	118.28
31.49	35.19	34.34
39.57	51.02	51.66
72.11	132.76	1078.70
1805.45	4203.04	61075.00
66.82	32.29	74.68
58.91	64.46	721.21
43.29	36.47	84.89
304.36	399.30	74.68
38.03	54.32	77.68
59.01	149.39	100.27

Bibliography

C.C. Aggarwal and S. Parthasarathy. Mining massively incomplete data sets by conceptual reconstruction. In *Proc. KDD Conference '01*, San Francisco, USA, 2001.

D. Aha. Case-based learning algorithms. In *Proc. of the DARPA Case-Based Reasoning Workshop*, Massachusetts, USA, 1991.

D. Aha. Editorial. *Artificial Intelligence Review*, 11:1–6, 1997.

D. Aha. Feature weighting for lazy learning algorithms. In L. Huan and M. Hiroshi (eds)., editors, *Feature Extraction, Construction, and Selection. A Data Mining Perspectives.*, pages 13–32. Kluwer Academic, 1998.

D. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

S. Ahmad and V. Tresp. Some solutions to the missing feature problem in vision. In S.J Hanson, J.D Cowan, and C.L Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 393–400. MIT Press, 1993.

A. A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore,

J. Hudson Jr, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Griener, D. D. Weisenberger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O Brown, and L.M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

Y. Bengio and B. Gingras. Recurrent neural networks for missing or asynchronous data. In M. E. Hasselmo (eds.) D. S. Touretzky, M. C. Mozer, editor, *Advances in Neural Information Processing Systems 8*, pages 395–401. MIT Press., 1996.

L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4:888–900, 1992.

L. Breiman, J. H. Friedman, R.A Olshen, and C. J. Stone. *Classification and Regression Trees*. Belmont: Wadsworth, 1984.

S.F. Buck. A method of estimation of missing values in multivariate data suitable for use with an electronic computer. *Journal of the Royal Statistical Society, series B.*, 22:302–306, 1960.

R. Caruana. A non-parametric EM style algorithm for imputing missing values. In *Eight International Workshop on AI and Statistics*, Florida, USA, 2001.

R. Chambers. Evaluation criteria for statistical editing and imputation. Research note, Department of Social Statistics, University of Southampton, 2000. National Statistics Methodological Series No.28.

A. de Falguerolles and B. Francis. Algorithmic approaches for fitting bilinear models. In Y.Dodge and J.Whittaker, editors, *Computational Statistics*, pages 77–82. Springer-Verlag, 1992.

A.P Dempster, N.M. Laird, and D.B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Society*, 39:1–38, 1977.

R. Dybowski. Classification of incomplete feature vectors by radial basis function networks. *Pattern Recognition Letters*, 19:1257–1264, 1998.

B. Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78:316–331, 1983.

B. Efron. Missing data, imputation, and the bootstrap. *Journal of the American Statistical Association*, 89:463–475, 1994.

B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap.* Chapman and Hall, 1998.

EU. Euroedit project, 2000. URL `http://www.cs.york.ac.uk/euredit/`.

B.S. Everrit and D.J. Hand. *Finite Mixture Distributions.* Chapman and Hall, 1981.

I.P. Fellegi and D. Holt. A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association*, 71:17–35, 1976.

J. Fridlyand and S. Dudoit. Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method. Technical report, University of California, Berkeley, 2001.

K.R. Gabriel. Least squares approximation of matrices by additive and multiplicative models. *Journal of the Royal Statistical Society, series B.*, 40:186–196, 1978.

K.R. Gabriel and S. Zamir. Lower rank approximation of matrices by least square with any choices of weights. *Technometrics*, 21:489–298, 1979.

G.H. Golub and C.F. Loan. *Matrix Computation.* John Hopkins University Press, second edition, 1986.

G.H. Golub and C. Reinsch. Singular value decomposition and least square solutions. In *Handbook for Automatic Computation*, pages 134–151. Springer-Verlag, 1971.

I.J. Good. Some applications of the singular decomposition of a matrix. *Technometrics*, 2:823–831, 1969.

J.W. Graham and S.M. Hofer. EMCOV: EM based software, 1997. URL `http://methodology.psu.edu/EMCOV.html`.

B. Grung and R. Manne. Missing values in principal component analysis. *Chemometrics and Intelligent Lab. System*, 42:125–139, 1998.

D. Gryzbowski. Simplified deficiency processing brings hospital-wide benefits. *Journal of AHIMA*, 71:58–61, 2000.

T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botsein. Imputing missing values for gene expression arrays. Technical report, Divison of Biostatistics, Stanford University, 1999.

W.J. Heiser. Convergent computation by iterative majorization: theory and applications in multidimensional analysis. In *Recent Advances in Descriptive Multivariate Analysis*, pages 157–189. Oxford University Press, 1995.

J. Hoogland and J. Pannekoek. Evaluation of SPSS missing value analysis 7.5. Technical report, Statistics Netherlands, 2000.

N.J. Horton and S.R. Lipsitz. Multiple imputation in practice: comparison of software packages for regression models with missing variables. *The American Statistician*, 55:244–254, 2001.

J. J. Hox. A review of current software for handling missing data. *Kwantitatieve Methoden*, 62:123–138, 1999.

L. Hunt and M. Jorgensen. Mixture model clustering for mixed data with missing information. *Computational Statistics and Data Analysis*, 41:429–440, 2003.

A.K Jain and R.C Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

I.T. Jollife. *Principal Component Analysis.* Springer-Verlag, 1986.

L. Kamakashi, S. A. Harp, T. Samad, and R. P. Goldman. Imputation of missing data using machine learning techniques. In *Second International Conference on Knowledge Discovery and Data Mining*, Oregon, USA, 1996.

N. Kenney and A. Macfarlane. Identifying problems with data collection at local level: survey of nhs maternity units in england. *British Medical Journal*, 319: 619–622, 1999.

H.A.L. Kiers. Weighted least squares fitting using ordinary least squares algorithms. *Psychometrika*, 62:251–266, 1997.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, Canada, 1995.

J. L. Kolodner. *Case-Based Reasoning.* Morgan-Kaufmann, 1993.

P. Kontkanen, P. Myllymaki, T. Silander, and H. Tirri. Comparing stochastic complexity minimization algorithms in estimating missing data. In *Proc. Workshop on Uncertainty Processing '97*, Prague, Czech, 1997.

S. Laaksonen. Regression-based nearest neighbour hot decking. *Computational Statistics*, 15:65–71, 2000.

S. Laaksonen. Imputation strategies and their evaluation. Presentation for the Intermediate Chintex Wokshop, Statitics Findland, Helsinki, 2001.

R.J.A Little. Regression with missing x's: A review. *Journal of the American Statistical Association*, 87:1227–1237, 1992.

R.J.A. Little. Modeling the drop-out mechanism in repeated-measures studies. *Journal of the American Statistical Association*, 90:1112–1121, 1995.

R.J.A. Little and D.B. Rubin. *Statistical Analysis with Missing Data.* John Wiley and Sons, 1987.

R.J.A. Little and M.D. Schluchter. Maximum likelihood estimation for mixed continous and categorical data with missing values. *Biometrika*, 72:497–512, 1985.

R.J.A. Little and P.J. Smith. Editing and imputation for quantitative survey data. *Journal of the American Statistical Association*, 82:58–68, 1987.

C. Liu and D.B. Rubin. The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. *Biometrika*, 81:633–648, 1994.

E. Lyttkens. On the fix-point property of Wold's iteration iterative estimation method for principal components. In Krisnaiah, editor, *Multivariate Analysis*, Proceeding of International Symposium, pages 335–350, 1966.

MATLAB 6. Mathworks Inc., 2001.

E.C. Matlhouse. *Nonlinear Partial Least Squares*. PhD thesis, Northwestern University, 1995.

X.L. Meng. Missing data: Dial M for ??? *Journal of the American Statistical Association*, 95:1325–1330, 2000.

D. Mesa, P. Tsai, and R.L. Chambers. Using tree-based models for missing data imputation: an evaluation using UK census data. Research note, Department of Social Statistics, University of Southampton, 2000.

B. Mirkin. A sequential fitting procedure for linear data analysis model. *Journal of Classification*, 7:167–195, 1990.

B. Mirkin. *Mathematical Classification and Clustering*. Kluwer Academics, 1996.

B. Mirkin. Least-squares structuring, clustering, and data processing issues. *The Computer Journal*, 41:519–536, 1998.

T.M. Mitchel. *Machine Learning*. McGraw-Hill, 1997.

A.C. Morris, M.P. Cooke, and P.D. Green. Some solutions to the missing feature problem in data classification, with application to noise robust ASR. In *Proc. ICASSP'98*, Washington, USA, 1998.

I. Myrtveit, E. Stensrud, and U.H. Olsson. Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods. *IEEE Transaction on Software Engineering*, 27:999–1013, 2001.

I.T. Nabney. NETLAB, 1999. URL `http://www.ncrg.aston.ac.uk/netlab`.

I.T. Nabney. *NETLAB Algorithms for Pattern Recognition.* Springer, 2002.

S. Nordbotten. Editing statistical records by neural networks. *Journal of Official Statistics*, 11:391–411, 1995.

S. Nordbotten. Neural network imputation applied to the norwegian 1990 population census data. *Journal of Official Statistics*, 12:385–401, 1996.

V. Nrhi, S. Laaksonen, R. Hietala, T. Ahonen, and H. Lyyti. COMPUTERIZING THE CLINICIAN treating missing data in a clinical neuropsychological dataset data imputation. *The Clinical Neuropsychologist*, 15:380–392, 2001.

J.R. Quinlan. Unknown attribute values in induction. In *Proc. of the 6th International Machine Learning Workshop*, New York, USA, 1989.

J.N.K. Rao and R.R. Sitter. Variance estimation under two-phase sampling with application to imputation for missing data. *Biometrika*, 82:453–460, 1995.

S. Roweis. EM algorithms for PCA and SPCA. In M. Jordan, M. Kerans, and

S. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 626–632. MIT Press, 1998.

D. Rubin. *Multiple Imputation for Nonresponse in Surveys.* John-Wiley and Sons, 1987.

D.B. Rubin. Inference and missing data. *Biometrika*, 63:581–592, 1976.

D.B. Rubin. Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91:473–489, 1996.

W. Sarle. Prediction with missing inputs. In *Proc. of Joint Conference on Information Sciences*, North Carolina, USA, 1998.

J.L. Schafer. *Analysis of Incomplete Multivariate.* Chapman and Hall, 1997a.

J.L. Schafer. NORM, 1997b. URL `http://www.stst.psu.edu/jls/misoftwa.html`.

J.L. Schafer and M.K. Olsen. Multiple imputation for multivariate missing-data problems: a data analyst's perspective. *Multivariate Behavioral Research*, 33: 545–571, 1998.

H.Y. Shum, K. Ikeuchi, and R. Reddy. PCA with missing data and its application to polyhedral object modelling. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 17:854–867, 1995.

SSI. *PRELIS 2.3*, 1995.

D. Steinberg and P.L. Colla. *CART: Tree-Structred Nonparametric Data Analysis.* Salford Systems, 1995.

R. E. Strauss, M. N. Atanassov, and J. A. De Oliveira. Evaluation of the principal-component and expectation-maximization methods for estimating missing data in morphometric studies. To appear in Journal of Vertebrate Paleontology, 2002.

H. Timm and F. Klawonn. Different approaches for fuzzy cluster analysis with missing values. In *Proc. of 7th Europen Congress on Intelligent Techniques and Soft Computing CD-ROM*, Aachen, Germany, 1999.

M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society series B*, 61:611–622, 1999a.

M.T. Tipping and C.M. Bishop. Mixtures of principal component analysis. *Neural Computation*, 11:443–482, 1999b.

H. Tirri and T. Silander. Stochastic complexity based estimation of missing elements in questionnaire data, 1998. http://cosco.hiit.fi/publications.html.

I. Tjostheim, I. Solheim, and M. Alrdin. Combining information from a web survey and telephone survey. In A.K. Manrai and H.L Meadow, editors, *Proc. of Ninth Biennial World Marketing Congress*, 1999.

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Hastie, R. Tibshirani, D. Botsein, , and R.B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17:520–525, 2001.

P. Tsai. Using CART for missing data imputation: An analysis based on Dutch housing demand survey. Research report, Department of Social Statistics, University of Southampton, 2000.

F. van de Pol and J. Betlehem. Data editing perspectives. *Statistical Journal of the United Nations Economic Commision for Europe*, 14:153–177, 1997.

I. Wasito. Lazy learning in the least-squares imputation of incomplete data. Research Report Seminar, School of Computer Science and Information System, Birkbeck Collge, 2002.

I. Wasito. Least squares data imputation within nearest neighbour framework. In *International Workshop on Computational Management Science, Economics, Finance and Engineering*, Limassol, Cyprus, 2003.

I. Wasito, I. Mandel, and B. Mirkin. Experimental comparison of a global-local least-squares imputation techniques with EM algorithm. in preparation, 2003.

I. Wasito and B. Mirkin. Nearest neighbour approach in the least-squares imputation algorithms. Submitted to *Information Sciences*, 2002.

I. Wasito and B. Mirkin. Nearest neighbour approach in the least squares data imputation algorithms with different missing patterns. Submitted to *Computational Statistics and Data Analysis*, 2003.

D. Wettschereck and T. G. Dietterich. Locally adaptive nearest neighbour algorithms. In J. Cowan, G. Tesauro, and J. Alspector (eds), editors, *Advances in Neural Information Processing Systems 1993*, pages 184–191. Morgan-Kaufmann, 1994.

T. Wiberg. Computation of principal components when data are missing. In *Proc. Second Symposium Computational Statistics*, pages 229–236, 1976.

H. Wold. Estimation of principal components and related models by iterative least square. In Krisnaiah, editor, *Multivariate Analysis*, Proceeding of International Symposium, pages 391–420, 1966.

S. Wold. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20:397–405, 1978.

P. Wright. The significance of the missing data problem in knowledge discovery. In *Proc. First Southern Symposium on Computing*, Misissipi, USA, 1998.