UNIVERSITY OF GRANADA
Dept. of Computer Science and Artificial Intelligence
C.P. 18071, Granada, Spain

Imputation of Missing Values

Methods' Description

Julián Luengo
Salvador García
Francisco Herrera

1 Introduction

Many existing, industrial and research data sets contain Missing Values. They are introduced due to various reasons, such as manual data entry procedures, equipment errors and incorrect measurements. The simplest way of dealing with missing values is to discard the examples that contain the missing values. However, this method is practical only when the data contains relatively small number of examples with missing values and when analysis of the complete examples will not lead to serious bias during the inference.

Missing data treatment should be carefully thought, otherwise bias might be introduced into the knowledge induced. Depending on the way MVs have been produced, our approach to handle them will be different. Several methods have been proposed in the literature to treat missing data [1, 2]. The treatment of missing data can be handled in three different ways [4]:

- The first approach is to discard the examples with missing data in their attributes. Therefore deleting attributes with elevated levels of missing data are enclosed in this category too.

- Other approach is the use of maximum likelihood procedures, where the parameters of a model for the complete data are estimated, and used later for impute by means of sampling.

- Finally, the imputation of MVs is a class of procedures that aims to fill in the MVs with estimated ones. In most cases, data sets attributes are not independent from each other. Thus, through the identification of relationships among attributes, MVs can be determined. This is the most used approach.

In this document we focus our attention on the imputation methods. A fundamental advantage of this approach is that the missing data treatment is independent of the learning algorithm used. For this reason, the user can select the most appropriate method for each situation he faces. There is a wide family of imputation methods, from mean imputation to those which analyze the relationships between attributes.

The extended descriptions of the methods follow in the next Sections. Please refers to the reference papers to obtain the full citations and references which come along such descriptions. Each method's notations have been maintained as close to the original as possible.

2 Global Most Common Attribute Value for Symbolic Attributes, and Global Average Value for Numerical Attributes (MC)

This method [7] is very simple: given an instance y_i, if the attribute h of such example, that is y_{ih}, contains a MV, we use two possible methods. If the attribute is numerical, then y_{ih} is estimated by the average of all observed values of such attribute:

$$y_{\hat{ih}} = \frac{\sum_{j \in I_{ih}} y_{jh}}{|I_{ih}|}, \tag{2.1}$$

where I_{ih} is the index set of all examples with attribute h observed, and if y_{jh} is missing the j-th attribute is excluded from I_{ih}.

If h refers to a nominal attribute, then we substitute by the mode of such attribute:

$$y_{\hat{ih}} = \max_{j \in I_{ih}} \{count(y_{jh})\} \tag{2.2}$$

3 Concept Most Common Attribute Value for Symbolic Attributes, and Concept Average Value for Numerical Attributes (CMC)

As stated in *MC*, in this method [7] we replace the MV by the most repeated one if nominal or the mean value if numerical, but considering only the instances with same class as the reference instance. So if the attribute is numerical, then y_{ih} is estimated by:

$$y_{\hat{ih}} = \frac{\sum_{j \in I_{ih}} y_{jh}}{|I_{ih}|}, \tag{3.1}$$

where I_{ih} is now the index set of all examples with attribute h observed *and* same class than y_i, and if y_{jh} is missing the j-th attribute is excluded from I_{ih}. If h refers to a nominal attribute, then we substitute by the mode of such attribute:

$$y_{\hat{ih}} = \max_{j \in I_{ih}} \{count(y_{jh})\} \tag{3.2}$$

4 Imputation with K-Nearest Neighbour (KNNI)

In order to estimate a missing value y_{ih} in the i-th example vector y_i by KNN imputation[2], we first select K examples whose attribute values are similar to y_i . Next, the missing value is estimated as the average of the

corresponding entries in the selected K expression vectors. When there are other missing values in y_i and/or y_j , their treatment requires some heuristics. The missing entry y_{ih} is estimated as average:

$$\hat{y_{ih}} = \frac{\sum_{j \in I_{Kih}} y_{jh}}{|I_{Kih}|}, \tag{4.1}$$

where I_{Kih} is now the index set of K-nearest neighbor examples of the i-th example, and if y_{jh} is missing the j-th attribute is excluded from I_{Kih}. Note that KNNI has no theoretical criteria for selecting the best K-value and the K-value has to be determined empirically.

5 Weighted imputation with K-Nearest Neighbour (WKNNI)

The Weighted K-Nearest Neighbour method[11] selects the instances with similar values (in terms of distance) to a considered one, so it can impute as *KNNI* does. However, the estimated value now takes into account the different distances to the neighbours, using a weighted mean or the most repeated value according to a similarity measure. The similarity measure $s_i(y_j)$ between two examples y_i and y_j is defined by the reciprocal of the Euclidian distance calculated over observed attributes in y_i. Following we define the measure as follows:

$$1/s_i = \sum_{h_i \in O_i \bigcap O_j} (y_{ih} - y_{jh})^2, \tag{5.1}$$

where $O_i = \{h|$ the h-th component of y_i is observed$\}$.

The missing entry y_{ih} is estimated as average weighted by the similarity measure:

$$\hat{y_{ih}} = \frac{\sum_{j \in I_{Kih}} s_i(y_j) y_{jh}}{\sum_{j \in I_{Kih}} s_i(y_j)}, \tag{5.2}$$

where I_{Kih} is the index set of K-nearest neighbor examples of the i-th example, and if y_{jh} is missing the j-th attribute is excluded from I_{Kih}. Note that KNNI has no theoretical criteria for selecting the best K-value and the K-value has to be determined empirically.

6 K-means Clustering Imputation (KMI)

In K-means clustering[4], the intra-cluster dissimilarity is measured by the summation of distances between the objects and the centroid of the cluster

they are assigned to. A cluster centroid represents the mean value of the objects in the cluster.

Given a set of N objects $X = x_1, x_2, ..., x_N$ where each object has S attributes, we use $x_{ij}(1 \leq i \leq N \text{ and } 1 \leq j \leq S)$ to denote the value of attribute j in object x_i. Object x_i is called a *complete* object, if $\{x_{ij} \neq \phi | \forall 1 \leq j \leq S\}$, and and an incomplete object, if $\{x_{ij} = \phi | \exists 1 \leq j \leq S\}$, and we say object x_i has a missing value on attribute j. For any incomplete object x_i, we use $R = \{j | x_{ij} \neq \phi, 1 \leq j \leq S\}$ to denote the set of attributes whose values are available, and these attributes are called *reference* attributes. Our objective is to obtain the values of non-reference attributes for the incomplete objects. By K-means clustering method, we divide data set X into K clusters, and each cluster is represented by the centroid of the set of objects in the cluster. Let $V = v_1, \ldots, v_k$ be the set of K clusters, where $v_k(1 \leq k \leq K)$ represents the centroid of cluster k. Note that v_k is also a vector in a S-dimensiona space. We use $d(v_k, x_i)$ to denote the distance between centroid v_k and object x_i.

The algorithm for missing data imputation with K-means clustering method can be divided into three processes. First, randomly select K complete data objects as K centroids. Second, iteratively modify the partition to reduce the sum of the distances for each object from the centroid of the cluster to which the object belongs. The process terminates once the summation of distances is less than a user-specified threshold $\varepsilon = 100$, or no change on the centroids were made in last iteration. The last process is to fill in all the non-reference attributes for each incomplete object based on the cluster information. Data objects that belong to the same cluster are taken as nearest neighbors of each other, and we apply a nearest neighbor algorithm to replace missing data. We take as distance measure the Euclidean distance.

7 Imputation with Fuzzy K-means Clustering (FKMI)

Now we want to extend the original K-means clustering method to a fuzzy version to impute missing data[1, 4]. The reason for applying fuzzy approach is that fuzzy clustering provides a better description tool when the clusters are not well-separated, as is the case in missing data imputation. Moreover, the original K-means clustering may be trapped in a local minimum status if the initial points are not selected properly. However, continuous membership values in fuzzy clustering make the resulting algorithms less susceptible to get stuck in local minimum situation.

In fuzzy clustering, each data object xi has a membership function which describes the degree that this data object belongs to certain cluster vk. The

membership function is defined in the next equation

$$U(v_k, x_i) = \frac{d(v_k, x_i)^{-27(m-1)}}{\sum_{j=1}^{K} d(v_j, x_i)^{-2/(m-1)}} \tag{7.1}$$

where $m > 1$ is the fuzzifier, and $\sum_{j=1}^{K} U(v_j, x_i) = 1$ for any data object $x_i (1 \leq i \leq N)$. Now we can not simply compute the cluster centroids by the mean values. Instead, we need to consider the membership degree of each data object. Equation (7.2) provides the formula for cluster centroid computation:

$$v_k = \frac{\sum_{i=1}^{N} U(v_k, x_i) \times x_i}{\sum_{i=1}^{N} U(v_k, x_i)} \tag{7.2}$$

Since there are unavailable data in incomplete objects, we use only reference attributes to compute the cluster centroids.

The algorithm for missing data imputation with fuzzy K-means clustering method also has three processes. Note that in the initialization process, we pick K centroids which are evenly distributed to avoid local minimum situation. In the second process, we iteratively update membership functions and centroids until the overall distance meets the user-specified distance threshold ε. In this process, we cannot assign the data object to a concrete cluster represented by a cluster centroid (as did in the basic K-mean clustering algorithm), because each data object belongs to all K clusters with different membership degrees. Finally, we impute non-reference attributes for each incomplete object. We replace non-reference attributes for each incomplete data object x_i based on the information about membership degrees and the values of cluster centroids, as shown in next equation:

$$x_{i,j} = \sum_{k=1}^{K} U(x_i, v_k) \times v_{k,j}, \text{ for any non-reference attribute } j \notin R \tag{7.3}$$

8 Support Vector Machines Imputation (SVMI)

Support Vector (SV) machines comprise a new class of learning algorithms, motivated by the results of the statistical learning theory. SV regression estimation seeks to estimate functions

$$f(x) = (wx) + b, \qquad w, x \in \mathbb{R}^n, b \in \mathbb{R} \tag{8.1}$$

based on data

$$(x_1, y_1), \ldots, (x_l, y_l) \in \mathbb{R} \times \mathbb{R} \tag{8.2}$$

by minimizing the regularized risk functional

$$\| W \|^2 /2 + C \bullet R^{\varepsilon}_{emp} \tag{8.3}$$

where C is a constant determining the trade-off between minimizing the training error, or empirical risk

$$R^{\varepsilon}_{emp} = \frac{1}{l} \sum_{i=1}^{l} |y_i - f(x_i)|_{\varepsilon} \tag{8.4}$$

and the model complexity term $\| W \|^2$. Here, we use the so-called ε-insensitive loss function

$$|y - f(x)|_{\varepsilon} = \max\{0, |y - f(x)| - \varepsilon\} \tag{8.5}$$

The main insight of the statistical learning theory is that in order to obtain a small risk, one needs to control both training error and model complexity, i.e. explain the data with a simple model. The minimization of Eq. 8.5 is equivalent to the following constrained optimization problem[12]: minimize

$$\tau(w, \xi^{(*)}) = \frac{1}{2} \| w \|^2 + C \frac{1}{l} \sum_{i=1}^{l} (\xi_i + \xi_i^*) \tag{8.6}$$

subject to the following constraints

$$((w \bullet x_i) + b) - y_i \leq \varepsilon + \xi_i \tag{8.7}$$

$$y_i - ((w \bullet x_i) + b) \leq \varepsilon + \xi_i^* \tag{8.8}$$

$$\xi_i^{(*)} \geq 0, \qquad \varepsilon \geq 0 \tag{8.9}$$

As mentioned above, at each point x_i we allow an error of magnitude ε. Errors above ε are captured by the slack variables ξ^* (see constraints 8.7 and 8.8). They are penalized in the objective function via the regularization parameter C chosen a priori.

In the ν-SVM the size of ε is not defined a priori but is itself a variable. Its value is traded off against model complexity and slack variables via a constant $\nu \in (0, 1]$ minimize

$$\tau(W, \xi^{(*)}, \varepsilon) = \frac{1}{2} \| W \|^2 + C \bullet (\nu\varepsilon + \frac{1}{l} \sum_{i=1}^{l} (\xi_i + \xi_i^*)) \tag{8.10}$$

subject to the constraints 8.7 to 8.9. Using Lagrange multipliers techniques, one can show [12] that the minimization of Eq. 8.6 under the constraints 8.7

to 8.9 results in a convex optimization problem with a global minimum. The same is true for the optimization problem 8.10 under the constraints 8.7 to 8.9. At the optimum, the regression estimate can be shown to take the form

$$f(x) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i)(x_i \bullet x) + b \tag{8.11}$$

In most cases, only a subset of the coefficients $(\alpha_i^* - \alpha_i)$ will be nonzero. The corresponding examples x_i are termed support vectors (SVs). The coefficients and the SVs, as well as the offset b; are computed by the ν-SVM algorithm. In order to move from linear (as in eq. 8.11) to nonlinear functions the following generalization can be done: we map the input vectors x_i into a high-dimensional feature space Z through some nonlinear mapping $\Phi : X_i \rightarrow Z_i$ chosen a priori. We then solve the optimization problem 8.10 in the feature space Z. In that case, the inner product of the input vectors $(x_i \bullet x)$ in Eq. 8.11 is replaced by the inner product of their icons in feature space Z, $(\Phi(x_i) \bullet \Phi(x))$. The calculation of the inner product in a high-dimensional space is computationally very expensive. Nevertheless, under general conditions (see [12] and references therein) these expensive calculations can be reduced significantly by using a suitable function k such that

$$(\Phi(x_i) \bullet \Phi(x)) = k(x_i \bullet x), \tag{8.12}$$

leading to nonlinear regressions functions of the form:

$$f(x) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) k(x_i, x) + b \tag{8.13}$$

The nonlinear function k is called a kernel[12]. In our work we use a Gaussian kernel

$$k(x, y) \vDash \exp(- \parallel x - y \parallel^2 / (2\sigma_{kernel}^2)) \tag{8.14}$$

We can use SVM regression[6] to predict the missing condition attribute values. In order to do that, first we select the examples in which there are no missing attribute values. In the next step we set one of the condition attributes (input attribute), some of those values are missing, as the decision attribute (output attribute), and the decision attributes as the condition attributes by contraries. Finally, we use SVM regression to predict the decision attribute values.

9 Event Covering (EC)

Based on the work of Wong et al.[13], a mixed-mode probability model is approximated by a discrete one. First, we discretize the continuous components using a minimum loss of information criterion. Treating a mixed-mode feature n-tuple as a discrete-valued one, the authors propose a new statistical approach for synthesis of knowledge based on cluster analysis. As main advantage, this method does not require neither scale normalization nor ordering of discrete values. By synthesis of the data into statistical knowledge, they refer to the following processes: 1) synthesize and detect from data inherent patterns which indicate statistical interdependency; 2) group the given data into inherent clusters based on these detected interdependency; and 3) interpret the underlying patterns for each clusters identified. The method of synthesis is based on author's *event–covering* approach. With the developed inference method, we are able to estimate the MVs in the data.

In order to discretize the continuous values, we have used the Fayyad algorithm [5]. After discretization, we can apply the cluster analysis algorithm on incomplete mixed-mode data. In the next processes we only take into account the complete examples from the data set.

The cluster initiation process involver the analysis of the nearest neighbour distance distribution on a subset of samples, the selection of which is based on a mean probability criterion. Let $X = (X_1, X_2, \ldots, X_n)$ be a random n-tuple of related variables and $x = (x_1, x_2, \ldots, x_n)$ be its realization. Then a sample can be represented as x. Let S be an ensemble of observed samples represented as n-tuples. The nearest-neighbour distance of a sample x_i to a set of examples S is defined as:

$$D(x_i, S) = min_{x_j \in S_{x_i \neq x_j}} d(x_i, x_j) \tag{9.1}$$

where $d(x_i, x_j)$ is a distance measure. Since we are using discrete values, we have adopted the Hamming distance. Let C be a set of examples forming a simple cluster. We define the maximum within-cluster nearest-neighbour distance as

$$D_c^* = max_{x_i \in C} D(x_i, C) \tag{9.2}$$

D_c^* reflects an interesting characteristic of the cluster configuration: that is, the smaller the D_c^*, the denser the cluster. If the cluster in S are unknown, we do not know the value of D_c^*. However, we can estimate D_c^* with the following analysis. The estimation will depend on our conception of a cluster, which is as follows:

- If all the clusters C_i in an ensemble S have the same degree of denseness,

then D_c^* is the same for all C_i in S and also the same as the maximum of all the $D(x, S)$ values.

- If the clusters in S have different degrees of denseness, then when all $D(x, S)$ values are projected to a real axis, distinct groups will result. An isolated sample x which does not belong to any cluster (an outlier) will have a relatively large $D(x, S)$ value. Thus one way to characterize the denseness of all distinct clusters is by the maximum value of $D(x, S)$ for all x in S after the large values associated with isolated examples are removed. We represent this value as D^*.

Using a mean probability criterion to select a similar subset of examples, the isolated samples can be easily detected by observing the wide gaps in the nearest-neighbour distance space. The probability distribution from which the criterion is derived for the samples can be estimated using a second-order probability estimation. An estimation of $P(x)$ known as the *dependence tree product approximation* can be expressed as:

$$\widehat{P}(x) = \prod_{j=1}^{n} P(x_{mj}|x_{m_{k(j)}}), 0 < k(j) < 1 \tag{9.3}$$

where (1) the index set $m_1, m_2, \ldots, m_n$ is a permutation of the integer set $1, 2, \ldots, n$, (2) the ordered pairs $x_{mj}, x_{m_{k(j)}}$ are so chosen that they represent the set of branches of a spanning tree defined on X with their summed mutual information maximized, and (3) $P(x_{m1}|x_{m0}) = P(x_{m1})$. The probability defined above is known to be the best second-order aproximation of the high-order probability distribution. Then corresponding to each x in the ensemble, a probability $P(x)$ can be estimated.

In general, it is more likely for samples of relatively high probability to form clusters. By introducing the mean probability as below, samples can be divided into two subsets: those above the mean and those below. Samples above the mean will be considered first for cluster initiation.

Let $S = x$. The mean probability is defined as

$$\mu_s = \sum_{x \in S} P(x)/|S| \tag{9.4}$$

where $|S|$ is the number of samples in S. For more details in the probability estimation with *dependence tree product approximation*, please refer to [3].

When distance is considered for cluster initiation, we can use the following criteria in assigning a sample x to a cluster.

1. If there exists more than one cluster, say $C_k|k = 1, 2, \ldots$, such that $D(x, C_k) \leq D^*$ for all k, then all these clusters together can be merged.

2. If there exists exactly one cluster C_k, such that $D(x, C_k) \leq D^*$, then x can be grouped into C_k.
3. If $D(x, C_K) > D^*$ for all clusters C_k, then x may not belong to any cluster.

We use the mean probability to control this merging process at each iteration in the algorithm outlined below:

1. Calculate $P(x)$ for all x in S.
2. Set $K = 0, t = 0$
3. Let C_0 be a dummy subgroup representing samples of unknown cluster. Initially C_0 is empty.
4. If $|S| > T$ then $P' = \mu_s$ else $P' = 0$. (T is a size threshold indicating the smallest size of a cluster).
5. List all $x \in S$ in a table L, if $P(x) > P'$.
6. Calculate $D(x, L)$ for all x in L.
7. $D^* = \max_{x \in L} D(x, L)$ and assume that x is not isolated.
8. For all $x \in L$ do the following
   (a) Find x such that $P(x)$ is the highest.
   (b) If $D(x, C_k) \leq D^*$ for more than one cluster, say $C_k,\ i = 1, 2, \ldots$ then do
      i. if one of the cluster, say C_{ki}, is found at a previous iteration, i.e. $k_i < K$, then $C_0 = C_0 \cup \{x\}$;
      ii. else all the clusters $C_{ki},\ i = 1, 2, \ldots$ are merged.
   (c) If $D(x, C_k) \leq D^*$ for exactly one cluster C_k, then $C_k = \{x\} \cup C_k$.
   (d) If $D(x, C_k) > D^*$ for all clusters $C_k,\ k = 1, 2, \ldots, t$ then $t = t + 1$ and $C_t = \{x\}$.
   (e) Remove x from L and S.
9. $K = t$
10. Go to (4) until $S = 0$
11. For $k = 1$ to t do the following.
    If $|C_k| < T$, then $C_0 = C_0 \cup C_k$.

To avoid including distance calculation of outlier, we use a simple method suggested in [13] which assigns D^* the maximum value of all nearest-neighbour distances in L provided there is a sample in L having a nearest-neighbour distance value of $D^* - 1$ (with the distance values rounded to the nearest integer value). The value of T could be assumed to be

$$T = A \times \max_{j=1,\ldots,n} L_j^2,$$

but as stated in [13], the sample size allows to choose a smaller value of T based on some initial trials of the experiments.

After finding the initial clusters along with their membership, the re-grouping process is thus essentially an inference process for estimating the cluster label of a sample. The event-covering method can be conceptualized as a mapping which maps events onto a binary decision state which indicates whether or not they are relevant for clustering. Let $C = a_{c1}, a_{c2}, \ldots, a_{cq}$ be the set of labels for all possible clusters to which x can be assigned. Initially, C is the set of cluster labels found after the initiation process. Since each x in S is a realization of $X = (X_1, \ldots, X_n)$ and also associates with a value in C, C can be considered as an additional variable associated with X. The information of significant events associated with the cluster configuration is obtained by analyzing the frequency of events observed in the ensemble through the use of a contingency table. For X_k in X, we can form a contingency table between X_k and C. Let a_{ks} and a_{cj} be possible outcomes of X_k and C respectively, and let $obs(a_{ks}$ and $obsa_{cj}$ be the respectively marginal frequencies of their observed occurrences. The expected relative frequency of (a_{ks}, a_{cj}) is expressed as:

$$exp(a_{ks}, a_{cj}) = \frac{obs(a_{ks}) \times obs(a_{cj})}{|S|} \tag{9.5}$$

Let $obs(a_{ks}, a_{cj})$ represent the actual observed frequency of (a_{ks}, a_{cj}) in S. The expression

$$D = \sum_{j=1}^{q} \frac{(obs_{ks} - exp(a_{ks}, a_{cj}))^2}{exp(a_{ks}, a_{cj})} \tag{9.6}$$

summing over the outcomes of C in the contingency table, possesses an asymptotic chi-square property with $(q-1)$ degrees of freedom. D can then be used in a criterion for testing the statistical dependency between a_{ks}, and C at a presumed significant level as described below. For this purpose, we define a mapping

$$h_k^c(a_{ks}, C) = \begin{cases} 1, & \text{if } D > \chi^2(q-1); \\ 0, & \text{otherwise.} \end{cases} \tag{9.7}$$

where $\chi^2(q-1)$ is the tabulated chi-square value. The subset of selected events of X_k, which has statistical interdependency with C, is defined as

$$E_k^c = \{a_{ks} | h_k^c(a_{ks}, C) = 1\} \tag{9.8}$$

We call E_k^c the covered event subset of X_k with respect to C. Likewise, the covered event subset E_c^k of C with respect to X_k can be defined.

After finding the covered event subsets of E_c^k and E_k^c between a variable pair (C, X_k), information measures can be used to detect the statistical pattern of these subsets. These information measures are based on an incomplete probability scheme defined over the subset of significant events in the outcome space of the variables. Let X_k^c and C^k represent the restricted variables of the covered event subsets E_c^k and E_k^c respectively. An interdependence redundancy measure between X_k^c and C^k can be defined as

$$R(X_k^c, C^k) = \frac{I(X_k^c, C^k)}{H(X_k^c, C^k)} \tag{9.9}$$

where $I(X_k^c, C^k)$ is the expected mutual information and $H(X_k^c, C^k)$ is the Shannon's entropy defined respectively on X_k^c and C^k. Mathematically, they are expressed as

$$I(X_k^c, C^k) = \sum_{a_{cu} \in E_c^k} \sum_{a_{ks} \in E_k^c} P(a_{cu}, a_{ks}) \log \frac{P(a_{cu}, a_{ks})}{P(a_{cu}) P(a_{ks})} \tag{9.10}$$

and

$$H(X_k^c, C^k) = -\sum_{a_{cu} \in E_c^k} \sum_{a_{ks} \in E_k^c} P(a_{cu}, a_{ks}) \log P(a_{cu}, a_{ks}) \tag{9.11}$$

The interdependence redundancy measure has a chi-square distribution:

$$I(X_k^c, C^k) \; \frac{\chi^2_{df}}{2|S| H(x_k^c, C^k)} \tag{9.12}$$

where *df* is the corresponding degree of freedom having the value $(|E_c^k| - 1)(|E_k^c| - 1)$. A chi-square test is then used to select interdependent variables in X at a presumed significant level.

For a data set with low-noise leve, analysis based on the marginal probability distribution of the first-order events (events of a single variable) may be adequate. However, for data with higher noise level, the second order probability distribution, defined on the joint events corresponding to a variable pair, may be needed. We call these joint events of a variable pair the *second-order events*. The second-order events are of particular importance because

1) reliable probability estimates can be obtained in an ensemble of a reasonable size and 2) random noise which may affect the outcome of one variable is less likely to simultaneously affect the joint outcome of two variables. Thus, during the clustering process, it is desirable that only second-order events are included.

When selecting joint events for clustering purposes, those reflecting interdependency usually contain more information. In other words, their observed frequency should deviate significantly from the expected marginal relative frequency derived from its first-order event. Thus the second-order event corresponding to (X_k, X_i) must be in $E_k^i \times E_i^k$, if they contain additional information as compared to the marginal events. Hence, we accept only these second-order events for further testing while the others are disregarded. Since only a subset of second-order events is now involved, the number of events for analysis during regrouping phase is substantially reduced.

Now, a new variable corresponding to a variable-pair (X_k, X_i) in X can be used to associate with the second order events in the outcome space of $E_k^i \times E_i^k$. For samples represented as $X = (X_1, \ldots, X_n)$, we can construct a new representation $X_e = (X_1, \ldots, X_N)$. X_e consists of all the variables in X as well as those representing all the possible combination of the variable-pairs. Thus, N has the value $n + n(n-1)/2$. We call X_e the extended tuple of X. We can then extend the selection of significant events and variables for clustering as described before to X_e.

Since not all the components in a sample are statistically relevant for clustering purposes, components (first- and second-order events) of a sample x are chosen based on the subset of events selected in the event-covering process. The component os a sample is selected if it has significant interdependency with the hypothesized cluster label. Let $x'(a_{cj}) = \{x'_1, \ldots, x'_m\}(m > 0)$ be the set of selected components of x_e in estimating the cluster label as a_{cj}. The event x_k in the set $x'(a_{cj})$ is chosen if the following conditions are satisfied.

1. The value of x_k is not a second-order event that is disregarded.
2. The value of x_k is in E_k^c and a_{cj} is in E_c^k.
3. $R(X_k^c, C^k)$ is significant.

The cluster regrouping process uses an information measure to regroup data iteratively. Wong et al. have proposed an information measure called *normalized surprisal* (NS) to indicate significant of joint information. Using this measure, the information conditioned by an observed event x_k is weighted according to $R(X_k^c, C^K)$, their measure of interdependency with the cluster label variable. Therefore, the higher the interdependency of a conditioning

event, the more relevant the event is. NS measures the joint information of a hypothesized value baed on the selected set of significant components. It is defined as

$$NS(a_{cj}|x'(a_{cj})) = \frac{I(a_{cj}|x'(a_{cj}))}{m\left(\sum_{k=1}^{m} R(X_k^c, C^k)\right)} \tag{9.13}$$

where $I(a_{cj}|x'(a_{cj}))$ is the summation of the weighted conditional information defined on the incomplete probability distribution scheme as

$$\begin{aligned} I(a_{cj}|x'(a_{cj})) &= \sum_{k=1}^{m} R(X_k^c, C^k) I(a_{cj}|x_k)) \\ &= \sum_{k=1}^{m} R(X_k^c, C^k) \left(-log \frac{P(a_{cj}|x_k)}{\sum_{a_{cu} \in E_c^k} P(a_{cu}|x_k)} \right) \end{aligned} \tag{9.14}$$

In rendering a meaningful calculation in the incomplete probability scheme formulation, x_k is selected if

$$\sum_{a_{cu} \in E_c^k} P(a_{cu}|x_k) > T \tag{9.15}$$

where $T \geq 0$ is a size threshold for meaningful estimation. NS can be used in a decision rule in the regrouping process. Let $C = \{a_{c1}, \ldots, a_{cq}\}$ be the set of possible cluster labels. We assign a_{cj} to x_e if

$$NS(a_{cj}|x'(a_{cj})) = \min_{a_{cu} \in C} NS(a_{cu}|x'(a_{cu})).$$

If no component is selected with respect to all hypothesized cluster labels, or if there are more than one label associated with the same minimum NS, then the sample is assigned a dummy label, indicating that the estimated cluster label is still uncertain. Also, zero probability may be encountered in the probability estimation, an unbiased probability based on *Entropy minimax*. In the regrouping algorithm, the cluster label for each sample is estimated iteratively until a stable set of label assignments is attained. The cluster regrouping algorithm is outlined as follows.

1. Construct x_e from x in the ensemble
2. Identify $\{E_k^c\}$, $\{E_c^k\}$ and compute the finite probability schemes based on the current cluster labels C.
3. Set $number\_of\_change = 0$
4. For each x_e in the ensemble do the following.

(a) If estimation is uncertain, then assign the dummy label a_{cj}.

(b) Otherwise assign x_e to cluster a_{cj} if

$$NS(a_{cj}|x'(a_{cj}))) \min_{a_{cu} \in C} NS(a_{cu}|x'(a_{cu})).$$

(c) if $a_{cj} \neq previous\_cluster\_label$ then do the following.

i. Set $number\_of\_change = number\_of\_change + 1$.

ii. Update cluster label for x_e.

5. If $number\_of\_change > 0$ then goto (2), else stop.

Once the clusters are stable, we take the examples with MVs. Now we use the distance $D(x_i, S) = \min_{x_j \in S_{x_i \neq x_j}} d(x_i, x_j)$ to find the nearest cluster C_i to such instance. From this cluster we compute the centroid x‘ such that

$$D(x', C_i) < D(x_i, C_i) \tag{9.16}$$

for all instances x_i of the cluster C_i. Once attained the centroid, the MV of the example is imputed with the value of the proper attribute of x_i.

10 Regularized Expectation-Maximization (EM)

The EM algorithm and the methods that will be derived from it in subsequent sections are only applicable to data sets in which the missing values are missing at random (MAR). The probability distribution of multivariate Gaussian data can be parameterized by the mean and the covariance matrix (i.e., the mean and the covariance matrix are sufficient statistics of the Gaussian distribution). In an iteration of the EM algorithm for Gaussian data, estimates of the mean and of the covariance matrix are revised in three steps. First, for each record with missing values, the regression parameters of the variables with missing values on the variables with available values are computed from the estimates of the mean and of the covariance matrix. Second, the missing values in a record are filled in with their conditional expectation values given the available values and the estimates of the mean and of the covariance matrix, the conditional expectation values being the product of the available values and the estimated regression coefficients. Third, the mean and the covariance matrix are re-estimated, the mean as the sample mean of the completed dataset and the covariance matrix as the sum of the sample covariance matrix of the completed dataset and the contributions of the conditional covariance matrices of the imputation errors in the records with imputed values. The EM algorithm starts with initial estimates of the

mean and of the covariance matrix and cycles through these steps until the imputed values and the estimates of the mean and of the covariance matrix stop changing appreciably from one iteration to the next.

For the following formal description of the EM algorithm [10], let $X \in \mathbb{R}^{n \times p}$ be a data matrix with n records consisting of p variables, with the values of some of the variables missing in some records. In the conventional EM algorithm, the number n of records is assumed to be much greater than the number p of variables, so that the sample covariance matrix of the data set completed with imputed values is positive definite.

From the incomplete data set, the mean $\mu \in \mathbb{R}^{1 \times p}$ of the records and the covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$ of the variables are to be estimated. For a given record $x = X_i$ with missing values, let the vector $x_a \in \mathbb{R}^{1 \times p_a}$ consist of the p_a variables for which, in the given record, the values are available, and let the vector $x_m \in \mathbb{R}^{1 \times p_m}$ consist of the remaining p_m variables for which, in the given record, the values are missing. Let the mean be partitioned correspondingly into a part $\mu_a \in \mathbb{R}^{1 \times p_a}$ with the mean values of the variables for which, in the given record, the values are available, and a part $\mu_m \in \mathbb{R}^{1 \times p_m}$ with the mean values of the variables for which, in the given record, the values are missing. For each record $x = X_i, (i = 1, \ldots, n)$ with missing values, the relationship between the variables with missing values and the variables with available values is modeled by a linear regression model

$$x_m = \mu_m + (x_a - \mu_a)B + e \tag{10.1}$$

The matrix $B \in \mathbb{R}^{p_a \times p_m}$ is a matrix of regression coefficients,and the residual $e \in \mathbb{R}^{1 \times p_m}$ is assumed to be a random vector with mean zero and unknown covariance matrix $C \in \mathbb{R}^{p_m \times p_m}$. In each iteration of the EM algorithm, estimates of the mean μ and of the covariance matrix Σ are taken as given, and from these estimates, the conditional maximum likelihood estimates of the matrix of regression coefficients B and of the covariance matrix C of the residual are computed for each record with missing values. With the estimated regression model for each record, the missing values are then filled in with imputed values, and new estimates of the mean μ and of the covariance matrix Σ are computed from the completed data set and from the estimates of the residual covariance matrices C.

Let $\widehat{\mu}^{(t)}$ and $\widehat{\Sigma}^{(t)}$ denote the estimates of the mean and of the covariance matrix in the tth iteration of the EM algorithm. (The hat accent $\widehat{A}$ designates an estimate of a quantity A.) The estimates of the mean and of the covariance matrix are either the result of the preceding EM iteration or, in the first EM iteration, they may be the sample mean and the sample covariance matrix of the dataset with initial guesses filled in for the missing

values. For a given record $x = X_i$ with missing values, let the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ be partitioned corresponding to the partitioning of the given record into variables with available values and variables with missing values: let the submatrix $\widehat{\Sigma}_{aa}$ of the estimated covariance matrix $\widehat{\Sigma}^{(t)}$ consist of the estimated variances and covariances of the variables for which, in the given record, the values are available; let the submatrix $\widehat{\Sigma}_{mm}$ consist of the estimated variances and covariances of the variables for which, in the given record, the values are missing; and let the two submatrices $\widehat{\Sigma}_{am}$ and $\widehat{\Sigma}_{ma}$ with $\widehat{\Sigma}_{am} = \widehat{\Sigma}_{ma}^T$ consist of the estimated cross-covariances of the variables for which, in the given record, the values are available with the variables for which, in the given record, the values are missing. Given the partitioned estimate of the covariance matrix $\widehat{\Sigma}^{(t)}$, the conditional maximum likelihood estimate of the regression coefficients can be written as

$$\widehat{B} = \widehat{\Sigma}_{aa}^{-1}\widehat{\Sigma}_{am} \tag{10.2}$$

From the structure of the regression model (10.1) follows that, given an estimate $\widehat{B}$ of the regression coefficients and the partitioned estimate of the covariance matrix $\widehat{\Sigma}^{(t)}$, an estimate of the residual covariance matrix takes the generic form

$$\widehat{C} = \widehat{\Sigma}_{mm} + \widehat{B}^T\widehat{\Sigma}_{aa}\widehat{B} - \widehat{B}^T\widehat{\Sigma}_{am} - \widehat{\Sigma}_{ma}\widehat{B} \tag{10.3}$$

Upon substitution of the conditional maximum likelihood estimate (10.2) of the regression coefficients, the conditional maximum likelihood estimate of the residual covariance matrix turns out to be the Schur complement

$$\widehat{C} = \widehat{\Sigma}_{mm} - \widehat{\Sigma}_{ma}\widehat{\Sigma}_{aa}^{-1}\widehat{\Sigma}_{am} \tag{10.4}$$

of the submatrix $\widehat{\Sigma}_{aa}$ in the covariance matrix estimate $\widehat{\Sigma}^{(t)}$. As a Schur complement of a positive definite matrix $\widehat{\Sigma}^{(t)}$, the residual covariance matrix $\widehat{C}$ is assured to be positive definite. The conditional expectation value $\widehat{x}_m \equiv E(x_m|x_a;\widehat{\mu}^{(t)})$ of the missing values in a given record follows from the estimated regression coefficients $\widehat{B}$ and the available values x_a as

$$\widehat{x}_m = \widehat{\mu}_m + (x_a - \widehat{\mu}_a)\widehat{B}, \tag{10.5}$$

where the vector $\widehat{\mu}_a$ is that part of the mean estimate $\widehat{\mu}^{(t)}$ that belongs to the variables for which, in the given record, the values are available, and the vector $\widehat{\mu}_m$ is that part of the mean estimate $\widehat{\mu}^{(t)}$ that belongs to the variables for which, in the given record, the values are missing.

After the missing values in all records $x = X_i (i = 1, \ldots, n)$ have thus been filled in with imputed values x_m, the sample mean

$$\widehat{\mu}^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{10.6}$$

of the completed data set is a new estimate of the mean of the records. A new estimate of the covariance matrix follows from the conditional expectation of the cross-products $\widehat{S}_i^{(t)} \equiv E(X_i^T X_i | x_a; \widehat{\mu}^{(t)}, \widehat{\Sigma}^{(t)})$ as

$$\widehat{\Sigma}^{(t+1)} = \frac{1}{\tilde{n}} \sum_{i=1}^{n} \left[\widehat{S}_i^{(t)} - (\widehat{\mu}^{(t+1)}) \widehat{\mu}^{(t+1)} \right], \tag{10.7}$$

where, for each record $x = X_i$, the conditional expectation $\widehat{S}_i^{(t)}$ of the cross-products is composed of three parts. The two parts that involve the available values in the record,

$$E(x_a^T x_a | x_a; \widehat{\mu}^{(t)}, \widehat{\Sigma}^{(t)}) = x_a^T x_a \tag{10.8}$$

and

$$E(x_m^T x_m | x_a; \widehat{\mu}^{(t)}, \widehat{\Sigma}^{(t)}) = \widehat{x}_m^T \widehat{x}_m + \widehat{C}, \tag{10.9}$$

is the sum of the cross-product of the imputed values and the residual covariance matrix $\widehat{C} = Cov(x_m, x_m | x_a; \widehat{\mu}^{(t)}, \widehat{\Sigma}^{(t)})$, the conditional covariance matrix of the imputation error. The normalization constant $\tilde{n}$ of the covariance matrix estimate (10.7) is the number of degrees of freedom of the sample covariance matrix of the completed data set. If, as above, one mean vector μ is estimated, the number of degrees of freedom is $\tilde{n} = n - 1$. The covariance matrix (10.7) is computed with the factor $\frac{1}{\tilde{n}}$ in place of the factor $\frac{1}{n}$ with which a maximum likelihood estimate would be computed, in order to correct the bias of the maximum likelihood estimate in a manner that parallels the bias-correction in the case of a complete data set. Thus, the new estimate (10.7) of the covariance matrix is computed in the same way as the sample covariance matrix of the completed data set, except that, for each record with missing values, the estimated residual covariance matrix $\widehat{C}$ is added to the cross-products $\widehat{x}_m^T \widehat{x}_m$ of the imputed values.

The next iteration of the EM algorithm is carried out with the updated estimates $\widehat{\mu}^{(t+1)}$ and $\widehat{\Sigma}^{(t+1)}$ of the mean and of the covariance matrix. The iterations are stopped when the algorithm has converged, that is, when the estimates $\widehat{\mu}^{(t)}$ and $\widehat{\Sigma}^{(t)}$ and the imputed values $\widehat{x}_m$ stop changing appreciably. The change is measured by the value $rd_{X_{mis}}$ defined as:

$$rd_{X_{mis}} = \frac{d_{X_{mis}}}{n_{X_{misPre}}} \tag{10.10}$$

where

$$\begin{aligned} d_{X_{mis}} &= \frac{\| \widehat{x}_m^{(t)} - \widehat{x}_m^{(t-1)} \|}{|p_m \in X|} \\ n_{X_{misPre}} &= \frac{\| \widehat{x}_m^{(t-1)} + \mu^{(t-1)} \|}{|p_m \in X|}. \end{aligned}$$

The value $rd_{X_{mis}}$ is used as *stagnation control* for the algorithm.

The EM algorithm converges monotonically in that the likelihood of the available data increases monotonically from iteration to iteration. However, the EM algorithm converges only linearly, with a rate of convergence that depends on the fraction of values that are missing in the data set, and so it may need many iterations to converge.

If, for any record, the number p_a of variables with available values is greater than the number $\tilde{n}$ of degrees of freedom available for the estimation of the covariance matrix, the submatrix $\widehat{\Sigma}_{aa}$ of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ is singular and the conditional maximum likelihood estimate (10.2) of the matrix of regression coefficients B is not defined. The submatrix $\widehat{\Sigma}_{aa}$ of the covariance matrix estimate may already be poorly conditioned if the number $\tilde{n}$ of degrees of freedom only marginally exceeds the number p_a of available values in a record. In such ill-posed or ill-conditioned cases, it is necessary to regularize the estimate (10.2) of the regression coefficients.

The *regularized EM* algorithm[10] consists of the same steps as the EM algorithm, with the exception that, in each iteration and for each record with missing values, the inverse matrix $\widehat{\Sigma}_{aa}^{-1}$ in the estimate (10.2) of the regression coefficients is replaced with a regularized inverse

$$\widehat{\Sigma}_{aa}^{-1} \leftarrow (\widehat{\Sigma}_{aa} + h^2 \widehat{D})^{-1}, \tag{10.11}$$

where $\widehat{D} = Diag(\widehat{\Sigma}_{aa})$ is the diagonal matrix consisting of the diagonal elements of the covariance matrix and the scalar h is a regularization parameter. That is, the ill-defined or ill-conditioned inverse $\widehat{\Sigma}_{aa}^{-1}$ is replaced with the inverse of the matrix that results from the covariance matrix $\widehat{\Sigma}_{aa}$ when the diagonal elements are inflated by the factor $1 + h^2$. This method of regularizing the inverse of a matrix, in which a regularized inverse is formed as the inverse of the sum of the matrix and a multiple of a positive definite matrix, is called ridge regression in the statistics literature and Tikhonov regularization in the literature on numerical linear algebra.

First, we will develop a representation of the regularized estimates of the regression parameters that makes some properties of ridge regression manifest and leads to a procedure for computing the regression parameters

in the regularized EM algorithm. Second, we will describe a criterion for the choice of the regularization parameter h. Third, we will juxtapose two variants of ridge regression, both of which can be used in the regularized EM algorithm.

a. Ridge Regression

In terms of the correlation matrix

$$\widehat{\Sigma}'_{aa} \equiv \widehat{D}^{-1/2}\widehat{\Sigma}_{aa}\widehat{D}^{1/2}$$

and the scaled cross-covariance matrix

$$\widehat{\Sigma}'_{am} \equiv \widehat{D}^{-1/2}\widehat{\Sigma}_{am},$$

the regularized estimate of the regression coefficients can be written as

$$\widehat{B}_h = \widehat{D}^{-1/2}\widehat{B}'_h \tag{10.12}$$

where

$$\widehat{B}'_h \equiv (\widehat{\Sigma}'_{aa} + h^2 I)^{-1}\widehat{\Sigma}'_{am} \tag{10.13}$$

is termed the standard form of the estimate. The fact that the correlation matrix $\widehat{\Sigma}'_{aa}$ and the scaled cross-covariance matrix $\widehat{\Sigma}'_{am}$ can be factored in similar ways can be exploited to cast the problem of estimating the regression coefficients from scaled submatrices $\widehat{\Sigma}'_{aa}$ and $\widehat{\Sigma}'_{am}$ of a given covariance matrix estimate $\widehat{\Sigma}^{(t)}$ into the more conventional form of estimating regression coefficients from given data matrices. This recasting of the estimation problem will lead to a representation of the regularized regression coefficients that makes some properties of ridge regression manifest and translates into a procedure for computing the regression coefficients in the regularized EM algorithm.

The correlation matrix $\widehat{\Sigma}'_{aa}$, the scaled cross-covariance matrix $\widehat{\Sigma}'_{am}$, and the submatrix $\widehat{\Sigma}_{mm}$ of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ can be decomposed into factors $X_a \in \mathbb{R}^{\tilde{n}\times p_a}$ and $X_m \in \mathbb{R}^{\tilde{n}\times p_m}$, such that

$$\widehat{\Sigma}'_{aa} = X_a^T X_a/\tilde{n}, \widehat{\Sigma}'_{am} = X_a^T X_m/\tilde{n} \tag{10.14}$$

and

$$\widehat{\Sigma}_{mm} = X_m^T X_m/\tilde{n}. \tag{10.15}$$

The factors X_a and X_m can be viewed as analogues of data matrices whose second moment matrices $X_a^T X_a/\tilde{n}$, $X_a^T X_m/\tilde{n}$, and $X_m^T X_m/\tilde{n}$ are the scaled submatrices (10.14,10.15) of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$.

The sampling error of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ contributes to the error of the imputed values and hence will play a role in determining

the regularization parameter h. Let us assume that the sampling error of the covariance matrix estimate is equal to the sampling error that would be expected if the data set were complete and if the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ were the sample covariance matrix. The distribution of the sampling error of a sample covariance matrix is a function of the number $\tilde{n}$ of degrees of freedom available for the estimation of the covariance matrix, and so, in order for the assumed sampling error of the scaled submatrices (10.14 and 10.15) to be equal to the sampling error that would be expected for second moment matrices of actual data matrices X_a and X_m, it is necessary that the number of rows of the factors X_a and X_m be equal to the number $\tilde{n}$ of degrees of freedom. That is, the factors X_a and X_m must have $\tilde{n} = n - 1$ rows if one mean vector is estimated from the data set X and $\tilde{n} = n - S$ rows if mean vectors of S groups of records are estimated.

The factorization (10.14 and 10.15) of the scaled submatrices can, for instance, be obtained from an eigendecomposition $\widehat{\Sigma}^{(t)} = T\Phi^2T^T$ of the covariance matrix estimate, with a matrix $T \in \mathbb{R}^{p\times\tilde{n}}$ containing as its columns the mutually orthogonal eigenvectors of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ and with a diagonal matrix $\Phi^2 = Diag(\phi_j^2)$of eigenvalues $\phi_j^2(j = 1, \ldots, \tilde{n})$. Let the submatrix $T_a \in \mathbb{R}^{p_a\times\tilde{n}}$ consist of those rows of the eigenvector matrix T that belong to the variables for which, in the record under consideration, the values are available, and let the submatrix $T_m \in \mathbb{R}^{p_m\times\tilde{n}}$ consist of the remaining rows of the eigenvector matrix T that belong to the variables for which, in the record under consideration, the values are missing. In terms of the partitioned eigendecomposition of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$, the factors X_a and X_m can be written as

$$X_a = \sqrt{\tilde{n}}\Phi T_a^T \widehat{D}^{-1/2} and X_m = \sqrt{\tilde{n}}\Phi T_m^T, \tag{10.16}$$

which shows that a factorization of the form (13) exists. If the number p of variables is greater than or equal to the number $\tilde{n}$ of degrees of freedom available for the estimation of the covariance matrix, the number $\tilde{n}$ of degrees of freedom is just the number of nonzero eigenvalues ϕ_j^2 of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$. If the number p of variables is less than the number $\tilde{n}$ of degrees of freedom, the number of nonzero eigenvalues ϕ_j^2 is less than the number $\tilde{n}$ of degrees of freedom. In this latter case, the factors X_a and X_m could be a product of the above form (10.16), provided that the matrix Φ with the square roots of the eigenvalues ϕ_j^2 is completed with zeros to have $\tilde{n}$ rows. However, the form of the factors is irrelevant for the present argument. What is relevant is that a factorization (10.14 and 10.15) of the scaled submatrices of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ exists.

The factors X_a and X_m can be interpreted as the data matrices in the

linear regression model

$$X_m = X_a B' + E, \tag{10.17}$$

where $E \in \mathbb{R}^{\tilde{n} \times p_m}$ is a matrix of residuals. From the factorization (10.14 and 10.15) of the scaled submatrices $\widehat{\Sigma}'_{aa}$ and $\widehat{\Sigma}'_{am}$ of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ follows that estimating the regression coefficients B' of the regression model (10.17) from given data matrices X_a and X_m is equivalent to estimating the standard form $B' = \widehat{D}^{1/2} B$ of the regression coefficients of the model (10.1) from a given covariance matrix estimate $\widehat{\Sigma}^{(t)}$. The standard form $\widehat{B}'_h = (\widehat{\Sigma}'_{aa} + h^2 I)^{-1} \widehat{\Sigma}'_{am}$ of the regularized regression coefficients expressed in terms of the submatrices of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$ is identical to the standard form $\widehat{B}'_h = (X_a^T X_a + \tilde{n} h^2 I)^{-1} X_a^T X_m$ of the regularized regression coefficients expressed in terms of the data matrices X_a and X_m. Moreover, for any estimate $\widehat{B}'$ of the standard form regression coefficients B', the second moment matrix $\widehat{E}^T \widehat{E}/\tilde{n}$ of the estimated residuals

$$\widehat{E} = X_m - X_a \widehat{B}'$$

is identical to the generic estimate (10.3) of the residual covariance matrix C of the regression model (10.1). Hence, estimating the regression coefficients and the residual second moment matrix of the regression model (10.17) from given data matrices X_a and X_m is equivalent to estimating the regression coefficients and the residual covariance matrix of the regression model (10.1) from a given covariance matrix estimate $\widehat{\Sigma}^{(t)}$. Since, under the above assumptions on the sampling error of the covariance matrix estimate $\widehat{\Sigma}^{(t)}$, the expected sampling errors of the estimated parameters also coincide, estimating the parameters of the regression model (10.1) from a given estimate $\widehat{\Sigma}^{(t)}$ of the covariance matrix is equivalent to estimating the parameters of the regression model (10.17) from given data matrices X_a and X_m. This equivalence makes it possible to apply standard methods for the regularization of conventional regression models (10.17) to the regression model (10.1) figuring in the EM algorithm.

A revealing representation of the ridge regression coefficients results from a singular value decomposition of the matrix X_a. Let us rescale the factors $\overline{X}_a = X_a/\sqrt{\tilde{n}}$ and $\overline{X}_m = X_m/\sqrt{\tilde{n}}$ such that in the factorization of the correlation matrix $\widehat{\Sigma}'_{aa} = \overline{X}_a^T \overline{X}_a$ and of the scaled cross-covariance matrix $\widehat{\Sigma}'_{am} = \overline{X}_a^T \overline{X}_m$ the number $\tilde{n}$ of degrees of freedom no longer appears explicitly. Whatever form is ascribed to the rescaled factor $\overline{X}_a$, it has a singular value decomposition $\overline{X}_a = \bigcup \wedge \vee^t$, where $\bigcup$ and $\vee$ are orthogonal matrices and $\wedge = Diag(\lambda_j)$ is the diagonal matrix of singular values λ_j. In the basis of the singular value decomposition, the correlation matrix becomes

$\widehat{\Sigma}'_{aa} = \vee \wedge^2 \vee^T$, which implies that the squared singular values λ_j^2 are the eigenvalues of the correlation matrix $\widehat{\Sigma}'_a a$ and that the right singular vectors $\vee_{:j}$, the columns of the matrix $\vee$, are the corresponding eigenvectors. Substituting the factorization (10.14 and 10.15) and the singular value decomposition of the rescaled factor $\overline{X}_a$ into the standard form estimate (10.13) yields the representation

$$\widehat{B}'_h = \vee Diag\left(\frac{\lambda_j}{\lambda_j^2 + h^2}\right) F \tag{10.18}$$

of the regression coefficients. The elements of the matrix $F \equiv \bigcup^T \overline{X}_m$ are called Fourier coefficients, in analogy to inverse problems in which the counterpart of the matrix $\overline{X}_a$ is a convolution operator whose singular value decomposition is equivalent to a Fourier expansion.

The representation (10.18) of the regression coefficients shows that, in the standard form, the columns of the regression coefficient matrix $\widehat{B}'_h$ are linear combinations of the eigenvectors $\vee_{:j}$ of the correlation matrix $\widehat{\Sigma}'_{aa}$. Only the eigenvectors $\vee_{:j}$ belonging to nonzero eigenvalues λ_j^2 contribute to the regression coefficients. The weights of the eigenvectors $\vee_{:j}$ are given by the products of the scalars $\lambda_j/(\lambda_j^2 + h^2)$ and the Fourier coefficients $F_{j:}$, which implies that only those rows $F_{j:}$ of the Fourier coefficient matrix that belong to nonzero eigenvalues λ_j^2 contribute to the regression coefficients.

The Fourier coefficients can be expressed in terms of the scaled cross-covariance matrix $\widehat{\Sigma}'_{am}$ and of the nonzero eigenvalues and corresponding eigenvectors of the correlation matrix $\widehat{\Sigma}'_{aa}$. Since, in terms of the singular value decomposition of the rescaled factor $\overline{X}_a$, the scaled cross-covariance matrix $\widehat{\Sigma}'_{am} = \overline{X}_a^T \overline{X}_m$ can be written as $\widehat{\Sigma}'_{am} = (\vee \wedge \bigcup^T)\overline{X}_m = \vee \wedge F$, we can take

$$F = \wedge^+ \vee^T \widehat{\Sigma}'_{am} \tag{10.19}$$

as the matrix of Fourier coefficients, the diagonal matrix $\wedge^+ = Diag(\lambda_j^+)$ being the pseudoinverse of the singular value matrix $\wedge$; that is, the diagonal elements of the pseudoinverse $\wedge^+$ are $\lambda_j^+ = 1/\lambda_j$ if $\lambda_j > 0$ and $\lambda_j^+ = 0$ if $\lambda_j = 0$. [In actual computations, an element λ_j^+ of the pseudoinverse should be set to zero if the singular value λ_j is smaller than a threshold value ε that depends on the machine precision.] If the jth eigenvalue λ_j^2 of the correlation matrix $\widehat{\Sigma}'_{aa}$ is zero, the jth row $F_{j:}$ of the Fourier coefficient matrix (10.19) consists of zeros and might thus differ from the jth row of the matrix $\bigcup^T \overline{X}_m$ that was originally defined to be the matrix of Fourier coefficients. But since all other rows of these matrices-the rows belonging to nonzero eigenvalues

λ_j^2- agree, the differences in the rows belonging to zero eigenvalues do not affect the estimate (10.18) of the regression coefficients.

Thus, we can compute the regression coefficients $\widehat{B}'_h$ from the partitioned covariance matrix estimate $\widehat{\Sigma}^{(t)}$ as a product (10.18) that involves the nonzero eigenvalues and corresponding eigenvectors of the correlation matrix $\widehat{\Sigma}'_{aa}$ and the Fourier coefficients (10.19). If there are $\tilde{n}$ degrees of freedom for the estimation of the covariance matrix Σ and p_a available values in the record for which the regression parameters are estimated, the number r of nonzero eigenvalues of the correlation matrix is at most $\tilde{n}$ or p_a , whichever is smaller. Henceforth, we let the eigenvalue matrix $\wedge^2 \in \mathbb{R}^{r\times r}$ and the eigenvector matrix $\vee \in \mathbb{R}^{p_a\times r}$ contain only the r nonzero eigenvalues and corresponding eigenvectors, and we similarly restrict the Fourier coefficient matrix $F \in \mathbb{R}^{r\times p_m}$ to the r relevant rows. The expression (10.18) for the standard form estimate of the regression coefficients remains valid with these restricted matrices.

The covariance matrix of the residuals, which, in updating the covariance matrix estimate at the end of each EM iteration, is added to the cross-products (9) of the completed data matrix, can also be represented in a factored form. Substituting the estimate $\widehat{B}_h = \widehat{D}^{-1/2}\widehat{B}'_h$ of the regression coefficients into the generic expression (10.3) for the residual covariance matrix yields the estimate

$$\widehat{C}_h = \widehat{C}_0 + F^T Diag\left(\frac{h^4}{(\lambda_j^2 + h^2)^2}\right) F. \tag{10.20}$$

The term

$$\widehat{C}_0 \equiv \widehat{\Sigma}_{mm} - F^T F,$$

which is independent of the regularization parameter h, vanishes if the regression coefficients are not overdetermined, which is the case if the number $\tilde{n}$ of degrees of freedom for the estimation of the covariance matrix Σ is less than or equal to the number p_a of variables with available values. Since the residual covariance matrix depends on the regularization method and on the regularization parameter, both of which cannot usually be chosen a priori, without reference to the data set under consideration, the residual covariance matrix is not, as in the well-posed case, the conditional covariance matrix of the imputation error. The uncertainties about the adequacy of the regularization method and the regularization parameter contribute to the conditional imputation error given the estimates of the mean and of the covariance matrix, but the residual covariance matrix does not account for these uncertainties. Nevertheless, substituting the residual covariance matrix (10.20) for the conditional covariance matrix of the imputation error in

updating the covariance matrix estimate at the end of each EM iteration seems a plausible heuristic.

The representation (10.18) makes manifest the way in which ridge regression regularizes the regression coefficients, that is, the way in which the noise, the high-frequency or small-scale components of the data, is filtered out. Both the regression coefficients regularized by a truncated principal component analysis of the correlation matrix $\widehat{\Sigma}'_{aa}$ and the regression coefficients regularized by ridge regression can be written as

$$\widehat{B}'_h = \vee Diag(f_j) \wedge^+ F$$

where what are called the filter factors f_j depend on the regularization method. For principal component regression, the filter factors of the retained principal component vectors (EOFs) $\vee_{:j}$ are unity, and the filter factors of the discarded principal component vectors are zero. Thus, regularization by a truncated principal component analysis of the correlation matrix $\widehat{\Sigma}'_{aa}$, which is what applied mathematicians call regularization by truncated singular value decomposition, corresponds to filtering with a step function filter. For ridge regression, the filter factors are

$$f_j = \frac{\lambda_j^2}{\lambda_j^2 + h^2}. \tag{10.21}$$

This filter function is structurally identical to the Wiener filter. The eigenvalues λ_j^2 are the correlate of the spectral density of what is called the signal in Wiener filtering, and the squared regularization parameter h^2 is the correlate of the spectral density of what is called the noise in Wiener filtering. The filter function of ridge regression decays more slowly with decreasing eigenvalues λ_j^2 than the step function filter of principal component regression. Principal component vectors with eigenvalues λ_j^2 much greater than the squared regularization parameter h^2 are unaffected by the filtering. Principal component vectors with eigenvalues λ_j^2 much smaller than the squared regularization parameter h^2 are effectively filtered out.

For typical data, which do not have an evident gap in the eigenvalue spectrum and whose samples are so small that only a few principal components can be retained in a truncated principal component analysis, leaving only a small choice of possible truncation parameters, the smoother filtering afforded by ridge regression and the greater flexibility of a continuous regularization parameter could offer advantages over principal component regression. The structural parallels between the ridge regression filter and the optimal Wiener filter moreover suggest that ridge regression might suppress noise in the data in a more robust way and with less loss of relevant

information than principal component regression. Indeed, ridge regression also arises as a regularization method when the observational error in the available data, which is ignored in the regression model (10.1), is explicitly taken into account. In the regression model (10.1), the available values x_a in a record are taken as known and observational errors are neglected, but ridge regression in the form presented here is still an adequate regularization method if the available values are affected by a non-negligible observational error whose relative variance- the variance of the observational error relative to the variance of the observed variable- is homogeneous throughout the data set. By choosing a regularized inverse (10.11) with a different matrix $\widehat{D}'$, one that, in contrast to the matrix $\widehat{D}$ above, does not consist of the diagonal elements of the covariance matrix $\widehat{\Sigma}_{aa}$, other variance structures of the observational error can be accommodated in ridge regression. To be sure, observational errors are also taken into account in a regularization method known as truncated total least squares, in which regression coefficients are computed in a truncated basis of principal components of the overall covariance matrix $\widehat{\Sigma}^{(t)}$ instead of the scaled submatrix $\widehat{\Sigma}'_{aa}$. But the continuous regularization parameter of ridge regression might still offer advantages over a truncated principal component analysis when there is only a small choice of possible truncation parameters.

b. Generalized cross-validation

In the regularized EM algorithm, the estimated regression coefficients are not of interest in their own right but only as intermediaries in the imputation of missing values. As a criterion for the choice of the regularization parameter h, it is therefore suitable to require that the error of the imputed values be as small as possible. As the regularization parameter tends to zero, the imputed values are increasingly affected by noise, implying an increasing sampling error. Conversely, as the regularization parameter tends to infinity, the ridge regression coefficients tend to zero and the imputed values (10.5) tend to the estimated mean values, implying an increasing regularization error. A good choice of regularization parameter, in between the limiting cases of zero and infinity, should minimize the total imputation error, the sum of the sampling error and the regularization error.

The author argued that the regularization parameter h that minimizes the expected mean squared error of predictions with an estimated linear regression model (10.17) is approximately equal to the minimizer of the generalized cross-validation (GCV) function

$$\mathcal{G}(h) \equiv \tilde{n} \frac{\| X_a \widehat{B}'_h - X_m \|_F^2}{tr(I - X_a X_a^{\dagger})^2} \tag{10.22}$$

an object function that resembles the object function of ordinary cross-

validation but is, in contrast to the latter, invariant under orthogonal transformations of the data. The notations $\| A \|_F$ and trA indicate the Frobenius norm and the trace of a matrix R, and the matrix

$$X_a^\dagger \equiv (X_a^T X_a + \tilde{n}h^2 I)^{-1} X_a^T \tag{10.23}$$

in the denominator of the GCV function is the regularized pseudoinverse of the data matrix X_a. With the regularized pseudoinverse $X_a^\dagger$ of the data matrix X_a, the regularized regression coefficients of the model (10.17) can be written as $\widehat{B}'_h = X_a^\dagger X_m$, which, if the data matrices X_a and X_m are again regarded as the factors in the decomposition (10.14 and 10.15) of the correlation matrix $\widehat{\Sigma}'_{aa}$ and of the scaled cross-covariance matrix $\widehat{\Sigma}'_{am}$, is identical to the standard form (10.13) of the regularized regression coefficients. Under the assumptions of section (a) on the sampling error of the correlation matrix $\widehat{\Sigma}'_{aa}$ and of the scaled cross-covariance matrix $\widehat{\Sigma}'_{am}$, the sampling error of the regularized regression coefficients is equal to the sampling error that would be expected if the regularized regression coefficients were estimated from actual data matrices X_a and X_m with $\tilde{n}$ records, so that the regularization parameter h that minimizes the expected mean squared error of the imputed values is likewise approximately equal to the minimizer of the GCV function (10.22). Therefore, in each iteration of the regularized EM algorithm, the regularization parameter h for each record with missing values is chosen as the minimizer of the GCV function (10.22).

An alternative form of the GCV function follows from the eigendecomposition of the correlation matrix $\widehat{\Sigma}'_{aa}$ and the derived representations of the regression coefficients and of the residual covariance matrix. Since the squared Frobenius norm of a matrix is equal to the trace of the product of the matrix and its transpose, $\| A \|_F^2 = tr(A^T A)$, the squared Frobenius norm $\| X_a \widehat{B}'_h - X_m \|_F^2$ in the numerator of the GCV function is proportional to the trace of the residual covariance matrix $\widehat{C}_h = \widehat{E}^T \widehat{E}/\tilde{n}$. Hence, the GCV function can be written as

$$\mathcal{G}(h) = \frac{\tilde{n}^2}{\mathcal{T}^2(h)} tr\widehat{C}_h$$

where

$$\mathcal{T}(h) = tr(I - X_a X_a^\dagger),$$

an effective number of degrees of freedom for the estimation of the residual covariance matrix $\widehat{C}_h$, can be expressed in terms of the filter factors (10.21) as

$$\mathcal{T}(h) = \tilde{n} - \sum_{j=1}^{r} f_j. \tag{10.24}$$

For a given regularization parameter h, evaluating both the trace $tr\widehat{C}_h$ of the residual covariance matrix from the factored representation (10.20) and the effective number of degrees of freedom $\mathcal{T}(h)$ from the filter factors (10.21) requires $O(r)$ operations, where r is the number of nonzero eigenvalues of the correlation matrix $\widehat{\Sigma}'_{aa}$. That is, if the ridge regression is computed via an eigendecomposition of the correlation matrix $\widehat{\Sigma}'_{aa}$, only a small additional effort is required to find, with one of the common scalar optimization methods, the regularization parameter h that minimizes the GCV function.

With the regularization parameter determined by generalized cross-validation, the regularized estimates of the imputed values are usually reliable, even when the noise in the data, which might be a result of observational errors, is not Gaussian and has an inhomogeneous variance structure. Since with small but nonzero probability the GCV function has a minimum near zero, generalized cross-validation occasionally leads to a regularization parameter near zero when, in fact, a greater regularization parameter would be more appropriate. Choosing too small a regularization parameter in such cases can be avoided by constructing a lower bound for the regularization parameter from a priori guesses of the magnitude of the imputation error.

c. Multiple and individual ridge regressions

If ridge regression with generalized cross-validation is used in the regularized EM algorithm as described above, the regularization of the regression coefficients is controlled by one regularization parameter per record with missing values. For each record, the regression coefficients of all variables with missing values are estimated jointly by multiple ridge regression. With generalized cross-validation, the regularization parameter is chosen such as to minimize, approximately, the expected mean squared error of the imputed values.

However, with the above methods, it is also possible to estimate individually regularized regression coefficients for each missing value. The matrix of regression coefficients (10.12) can be computed columnwise with an individual regularization parameter for each column. Instead of only one regularization parameter per record in multiple ridge regressions, in individual ridge regressions we can, for a record with p_m missing values, adjust p_m regularization parameters. Choosing the regularization parameter for each column of the regression coefficient matrix by generalized cross-validation approximately minimizes not only the expected average error of the imputed values in the record, but also the expected error of each individual imputed value.

The computation of individual ridge regressions is similar to the computation of a multiple ridge regression. If the ridge regression is computed via an eigendecomposition of the correlation matrix $\widehat{\Sigma}'_{aa}$, one obtains the stan-

dard form estimate (10.18) of the regression coefficients columnwise from the columns of the Fourier coefficient matrix (10.19), with an individual regularization parameter $h_j (j = 1, \ldots, p_m)$ for each column. The regularization parameters h_j are determined as the minimizers of the GCV function (10.22), where the numerator of the GCV function reduces from the squared Frobenius norm of a residual matrix to the squared Euclidean norm of a residual vector. Generalizing the factored representation (10.20) of the residual covariance matrix from the case of a multiple ridge regression to that of individual ridge regressions, one finds that the residual covariance matrix of individual ridge regressions consists of the elements

$$(\widehat{C}_h)_{kl} = (\widehat{C}_0)_{kl} + (F_{:k})^T \Gamma^{(k)} \Gamma^{(l)} F_{:l} \tag{10.25}$$

where $\Gamma^{(j)} \equiv h_j^2 (\wedge^2 + h_j^2 I)^{-1}$ is a diagonal matrix and h_j is the regularization parameter for the jth column of the matrix of regression coefficients. In a regularized EM algorithm with individual ridge regressions, this residual covariance matrix is added to the cross-products $\widehat{x}_m^T \widehat{x}_m$ of the imputed values when a new estimate of the covariance matrix (10.7) is assembled.

Thus, the additional computational expense of individual ridge regressions in place of a multiple ridge regression is merely that which is required to minimize the GCV function p_m times for p_m residual vectors, compared with minimizing it once for one residual matrix with p_m column vectors. As long as the greater number of regularization parameters to be estimated does not lead to the estimated regularization parameters becoming unreliable, the greater accuracy of the imputed values that can be expected with individual ridge regressions suggests the use of individual ridge regressions in the regularized EM algorithm whenever computationally feasible.

In our case, we have used multiple ridge regression due the computational cost for some data sets. The regularization parameter h is such that minimizes the GCV function. We have set an iteration limit besides the stagnation tolerance. If the algorithm reach the maximum iterations or either the value $rd_{X_{mis}}$ falls below of the stagnation tolerance the procedure stops with the current imputation used as solution. The inflation factor for the Covariance matrix is set to 1, that is, we do not inflate the Covariance matrix at all.

11 Singular Value Decomposition Imputation (SVDI)

In this method, we employ singular value decomposition (11.1) to obtain a set of mutually orthogonal expression patterns that can be linearly combined to approximate the values of all attributes in the data set[11]. These patterns,

which in this case are identical to the principle components of the data values' matrix, are further referred to as eigenvalues.

$$A_{m\times m} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n}. \tag{11.1}$$

Matrix V^T now contains eigenvalues, whose contribution to the expression in the eigenspace is quantified by corresponding eigenvalues on the diagonal of matrix Σ. We then identify the most significant eigenvalues by sorting the eigenvalues based on their corresponding eigenvalue. Although it has been shown that several significant eigenvalues are sufficient to describe most of the expression data, the exact fraction of eigenvalues best for estimation needs to be determined empirically.

Once k most significant eigenvalues from V^T are selected, we estimate a missing value j in example i by first regressing this attribute value against the k eigenvalues and then use the coefficients of the regression to reconstruct j from a linear combination of the k eigenvalues. The jth value of example i and the jth values of the k eigenvalues are not used in determining these regression coefficients. It should be noted that SVD can only be performed on complete matrices; therefore we originally substitute row average for all missing values in matrix A, obtaining A'. We then utilize an Regularized Expectation-Maximization method to arrive at the final estimate, as follows. Each missing value in A' is estimated using the above algorithm, and then the procedure is repeated on the newly obtained matrix, until the total change in the matrix falls below the empirically determined (by the authors [11]) threshold of 0.01 (noted as *stagnation tolerance* in the *EM* algorithm). The other parameters of the *EM* algorithm are the same for both algorithms.

12 Bayesian Principal Component Analysis (BPCA)

The missing value estimation method based on BPCA[9] consists of three elementary processes. They are (1) principal component (PC) regression, (2) Bayesian estimation, and (3) an expectation-maximization (EM)-like repetitive algorithm. Below, we describe each of these processes.

a. PC regression

For the time being, we consider a situation where there is no missing value. PCA represents the variation of D-dimensional example vectors y as a linear combination of principal axis vectors $w_l (1 \leq l \leq K)$ whose number is relatively small ($K < D$):

$$y = \sum_{l=1}^{K} x_l w_l + \epsilon \tag{12.1}$$

The linear coefficients $x_l (1 \leq l \leq K)$ are called factor scores. ϵ denotes the residual error. Using a specifically determined number K, PCA obtains x_l and w_l such that the sum of squared error $\parallel \epsilon \parallel^2$ over the whole data set Y is minimized.

When there is no missing value, x_l and w_l are calculated as follows. A covariance matrix S for the example vectors $y_i (1 \leq i \leq N)$ is given by

$$S = \frac{1}{N} \sum_{i=1}^{N} (y_i - \mu)(y_i - \mu)^T, \tag{12.2}$$

where μ is the mean vector of y: $\mu = (1/N) \sum_{i=1}^{N} y_i$. T denotes the transpose of a vector or a matrix. For description convenience, Y is assumed to be row-wisely normalized by a preprocess, so that $\mu = 0$ holds. With this normalization, the result by PCA is identical to that by SVD.

Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D$ and $u_1, u_2, \ldots, u_D$ denote the eigenvalues and the corresponding eigenvectors, respectively, of S. We also define the l-th principal axis vector by $w_l = \sqrt{\lambda_l} u_l$ .With these notations, the l-th factor score for an example vector y is given by $x_l = (w_l / \lambda_l)^T y$. Now we assume the existence of missing values. In PC regression, the missing part y^{miss} in the expression vector y is estimated from the observed part y^{obs} by using the PCA result. Let w_l^{obs} and w_l^{miss} be parts of each principal axis w_l, corresponding to the observed and missing parts, respectively, in y. Similarly, let $W = (W^{obs}, W^{miss})$ where W^{obs} or W^{miss} denotes a matrix whose column vectors are $w_1^{obs}, \ldots, w_K^{obs}$ or $w_1^{miss}, \ldots, w_K^{miss}$, respectively.

Factor scores $x = (x_1, \ldots, x_K)$ for the example vector y are obtained by minimization of the residual error

$$err = \parallel y^{obs} - W^{obs} x \parallel^2 .$$

This is a well-known regression problem, and the least square solution is given by

$$x = (W^{obsT} W^{obs})^{-1} W^{obsT} y^{obs}.$$

Using x, the missing part is estimated as

$$y^{miss} = W^{miss} x \tag{12.3}$$

In the PC regression above, W should be known beforehand. Later, we will discuss the way to determine the parameter.

b. Bayesian estimation

A parametric probabilistic model, which is called probabilistic PCA (PPCA), has been proposed recently. The probabilistic model is based on the assumption that the residual error ϵ and the factor scores $x_l (1 \leq l \leq K)$ in Equation

(reflinearcomb) obey normal distributions:

$$p(x) = \mathcal{N}_K(x|0, I_K),$$

$$p(\epsilon) = \mathcal{N}_D(\epsilon|0, (1/\tau)I_D),$$

where $\mathcal{N}_K(x|\mu, \Sigma)$ denotes a K-dimensional normal distribution for x, whose mean and covariance are μ and Σ, respectively. I_K is a $(K \times K)$ identity matrix and τ is a scalar inverse variance of ϵ. In this PPCA model, a complete log-likelihood function is written as:

$$\begin{aligned} ln\ p(y, x|\theta) &\equiv ln\ p(y, x|W, \mu, \tau) \\ &= -\frac{\tau}{2} \parallel y - Wx - \tau \parallel^2 - \frac{1}{2} \parallel x \parallel^2 + \frac{D}{2} ln\ \tau - \frac{K+D}{2} ln 2\Pi, \end{aligned}$$

where $\theta \equiv W, \mu, \tau$ is the parameter set. Since the maximum likelihood (ML) estimation of the PPCA is identical to PCA, PPCA is a natural extension of PCA to a probabilistic model.

We present here a Bayesian estimation method for PPCA from the authors. Bayesian estimation obtains the posterior distribution of θ and X, according to the Bayes theorem:

$$p(\theta, X|Y) \propto p(Y, X|\theta)p(\theta). \tag{12.4}$$

$p(\theta)$ is called a prior distribution, which denotes a priori preference for parameter θ. The prior distribution is a part of the model and must be defined before estimation. We assume conjugate priors for τ and μ, and a hierarchical prior for W, namely, the prior for $W, p(W|\tau, \alpha)$, is parameterized by a hyperparameter $\alpha \in \mathbb{R}^K$.

$$\begin{aligned} p(\theta|\alpha) &\equiv p(\mu, W, \tau|\alpha) = p(\mu|\tau)p(\tau) \prod_{j=1}^{K} p(w_j|\tau, \alpha_j), \\ p(\mu|tau) &= \mathcal{N}(\mu|\overline{\mu}_0, (\gamma_{\mu_0}^{\tau})^{-1} I_m), \\ p(w_j|\tau, \alpha_j) &= \mathcal{N}(w_j|0, (\alpha_j \tau)^{-1} I_m), \\ p(\tau) &= \mathcal{G}(\tau|\overline{\tau}_0, \gamma_{\tau_0}) \end{aligned}$$

$\mathcal{G}(\tau|\overline{\tau}, \gamma_\tau)$ denotes a Gamma distribution with hyperparameters $\overline{\tau}$ and γ_τ:

$$\mathcal{G}(\tau|\overline{\tau}, \gamma_\tau) \equiv \frac{(\gamma_\tau \overline{\tau}^{-1})^{\gamma_\tau}}{\Gamma(\gamma_\tau)} \exp\left[-\gamma_\tau \overline{\tau}^{-1} \tau + (\gamma_\tau - 1) ln \tau\right]$$

where $\Gamma(\cdot)$ is a Gamma function.

The variables used in the above priors, $\gamma_{\mu 0}$, $\overline{\mu}_0$, $\gamma_{\tau 0}$ and $\overline{\tau}_0$ are deterministic hyperparameters that define the prior. Their actual values should be given before the estimation. We set $\gamma_{\mu 0} = \gamma_{\tau 0} = 10^{-10}$, $\overline{\mu}_0 = 0$ and $\overline{\tau}_0 = 1$, which corresponds to an almost non-informative prior.

Assuming the priors and given a whole data set $Y = y$, the type-II ML hyperparameter α_{ML-II} and the posterior distribution of the parameter, $q(\theta) = p(\theta|Y, \alpha_{ML-II})$, are obtained by Bayesian estimation.

The hierarchical prior $p(W|\alpha, \tau)$, which is called an automatic relevance determination (ARD) prior, has an important role in BPCA. The j-th principal axis w_j has a Gaussian prior, and its variance $1/(\alpha_j \tau)$ is controlled by a hyperparameter α_j which is determined by type-II ML estimation from the data. When the Euclidian norm of the principal axis, $\parallel w_j \parallel$, is small relatively to the noise variance $1/\tau$, the hyperparameter α_j gets large and the principal axis w_j shrinks nearly to be 0. Thus, redundant principal axes are automatically suppressed.

c. EM-like repetitive algorithm

If we know the true parameter true, the posterior of the missing values is given by

$$q(Y^{miss}) = p(Y^{miss}|Y^{obs}, \theta_{true}),$$

which produces equivalent estimation to the PC regression. Here, $p(Y^{miss}|Y^{obs}, \theta_{true})$ is obtained by marginalizing the likelihood (12.4) with respect to the observed variables Y^{obs}. If we have the parameter posterior $q(\theta)$ instead of the true parameter, the posterior of the missing values is given by

$$q(Y^{miss}) = \int d\theta q(\theta) p(Y^{miss}|Y^{obs}, \theta),$$

which corresponds to the Bayesian PC regression. Since we do not know the true parameter naturally, we conduct the BPCA. Although the parameter posterior $q(\theta)$ can be easily obtained by the Bayesian estimation when a complete data set Y is available, we assume that only a part of Y, Y^{obs}, is observed and the rest Y^{miss} is missing. In that situation, it is required to obtain $q(\theta)$ and $q(Y^{miss})$ simultaneously.

We use a variational Bayes (VB) algorithm, in order to execute Bayesian estimation for both model parameter θ and missing values Y^{miss}. Although the VB algorithm resembles the EM algorithm that obtains ML estimators for θ and Y^{miss}, it obtains the posterior distributions for θ and Y^{miss}, $q(\theta)$ and $q(Y^{miss})$, by a repetitive algorithm.

The VB algorithm is implemented as follows: (a) the posterior distribution of missing values, $q(Y^{miss})$, is initialized by imputing each of the missing values to instance-wise average; (b) the posterior distribution of the

parameter θ, $q(\theta)$, is estimated using the observed data Y^{obs} and the current posterior distribution of missing values, $q(Y^{miss})$; (c) the posterior distribution of the missing values, $q(Y^{miss})$, is estimated using the current $q(\theta)$; (d) the hyperparameter α is updated using both of the current $q(\theta)$ and the current $q(Y^{miss})$; (e) repeat (b)-(d) until convergence.

The VB algorithm has been proved to converge to a locally optimal solution. Although the convergence to the global optimum is not guaranteed, the VB algorithm for BPCA almost always converges to a single solution practically. This is probably because the objective function of BPCA has a simple landscape. As a consequence of the VB algorithm, therefore, $q(\theta)$ and $q(Y^{miss})$ are expected to approach the global optimal posteriors.

Then, the missing values in the expression matrix are imputed to the expectation with respect to the estimated posterior distribution:

$$\widehat{Y}^{miss} = \int y^{miss} q(Y^{miss}) dY^{miss}. \tag{12.5}$$

In the implementation from the authors, no parameters are requested from the user, since all of them are computed by the algorithm itself.

13 Local Least Squares Imputation (LLSI)

Throughout the paper, we will use $G \in \mathbb{R}^{m \times n}$ to denote a gene expression data matrix with m genes and n experiments, and assume $m \gg n$. In the matrix G, a row $g_i^T \in \mathbb{R}^{1 \times n}$ represents expressions of the i-th gene in n experiments:

$$G = \begin{pmatrix} g_1^T \\ \vdots \\ g_m^T \end{pmatrix} \in \mathbb{R}^{m \times n}$$

A missing value in the l-th location of the i-th gene is denoted as α, i.e.

$$G(i, l) = \mathbf{g}_i(l) = \alpha$$

For simplicity of algorithm description, all missing value estimation algorithms mentioned in this paper are described first assuming there is a missing value in the first position of the first gene, i.e.

$$G(1, 1) = \mathbf{g}_1(1) = \alpha$$

then the general algorithms for the proposed missing value estimation methods for DNA microarray expression data are introduced.

This formulation can be adapted to a data set's format, by identifying the m gene with the m instance, and the n experiment will represent the n attribute.

A target gene that has missing values is represented as a linear combination of similar genes. Rather than using all available genes in the data, since only similar genes based on a similarity measure are used, the method is referred to as local least squares imputation (LLSimpute). As similarity measures, L_2-norm is used.

There are two steps in the local least squares imputation. The first step is to select k genes by the L_2-norm. The second step is regression and estimation, regardless of how the k genes are selected. A heuristic k parameter selection method is described.

13.1. Selecting genes

To recover a missing value α in the first location $\mathbf{g}_1(1)$ of $\mathbf{g}_1$ in $G \in \mathbb{R}^{m\times n}$, the k-nearest neighbor gene vectors for $\mathbf{g}_1$,

$$\mathbf{g}_{S_i}^T \in \mathbb{R}^{1\times n}, \quad 1 \leq i \leq k,$$

are found for LLSimpute based on the L_2-norm (LLSimpute). In this process of finding the similar genes, the first component of each gene is ignored following the fact that $\mathbf{g}_1(1)$ is missing.

The LLSimpute based on the Pearson correlation coefficient to select the k genes can be consulted in [8].

13.2. Gene-wise formulation of local least squares imputation

As imputation can be performed regardless of how the k-genes are selected, we present only the imputation based on L_2-norm for simplicity. Based on these k-neighboring gene vectors, the matrix $A \in \mathbb{R}^{k\times(n-1)}$ and the two vectors $\mathbf{b} \in \mathbb{R}^{k\times 1}$ and $\mathbf{w} \in \mathbb{R}^{(n-1)\times 1}$ are formed. The k rows of the matrix A consist of the k-nearest neighbor genes $\mathbf{g}_{S_i}^T \in \mathbb{R}^{1\times n}$, $1 \leq i \leq k$, with their first values deleted, the elements of the vector b consists of the first components of the k vectors $\mathbf{g}_{S_i}^T$, and the elements of the vector $\mathbf{w}$ are the $n-1$ elements of the gene vector $\mathbf{g}_1$ whose missing first item is deleted. After the matrix A, and the vectors $\mathbf{b}$ and $\mathbf{w}$ are formed, the least squares problem is formulated as

$$\min_x ||A^T\mathbf{x} - \mathbf{w}||_2 \tag{13.1}$$

Then, the missing value α is estimated as a linear combination of first values of genes

$$\alpha = \mathbf{b}^T\mathbf{x} = \mathbf{b}^T(A^T)^\dagger\mathbf{w}, \tag{13.2}$$

where $(A^T)^\dagger$ is the pseudoinverse of A^T.

For example, assume that the target gene $\mathbf{g}_1$ has a missing value in the first position among the total of six experiments. If the missing value is to be estimated by the k similar genes, the matrix A, and vectors $\mathbf{b}$ and $\mathbf{w}$ are constructed as

$$\begin{pmatrix} g_1^T \\ \vdots \\ g_m^T \end{pmatrix} = \begin{pmatrix} \alpha & \mathbf{w}^T \\ \mathbf{b} & A \end{pmatrix}$$

$$= \begin{pmatrix} \alpha & \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 & \mathbf{w}_4 & \mathbf{w}_5 \\ \mathbf{b}_1 & A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{b}_k & A_{k,1} & A_{k,2} & A_{k,3} & A_{k,4} & A_{k,5} \end{pmatrix}$$

where α is the missing value and $\mathbf{g}_{S_1}^T, \ldots, \mathbf{g}_{S_k}^T$ are genes similar to $\mathbf{g}_1^T$. From the second to the last components of the neighbor genes, $a_i^T,\ 1 \le i \le k$, form the i-th row vector of the matrix A. The vector w of the known elements of target gene $\mathbf{g}_1$ can be represented as a linear combination

$$\mathbf{w} \simeq \mathbf{x}_1\mathbf{a}_1 + \mathbf{x}_2\mathbf{a}_2 + \ldots + \mathbf{x}_k\mathbf{a}_k$$

where $\mathbf{x}_i$ are the coefficients of the linear combination, found from the least squares formulation (13.1). Accordingly, the missing value α in $\mathbf{g}_1$ can be estimated by

$$\alpha = \mathbf{b}^T x = \mathbf{b}_1\mathbf{x}_1 + \mathbf{b}_2\mathbf{x}_2 + \ldots + \mathbf{b}_k\mathbf{x}_k$$

Now, we deal with the case in which there are more than one missing values in a gene vector. In this case, to recover the total of q missing values in any locations of the gene $\mathbf{g}_1$, first, the k-nearest neighbor gene vectors for g_1,

$$\mathbf{g}_{S_i}^T \in \mathbb{R}^{1\times n}, \quad 1 \le i \le k,$$

are found. In this process of finding the similar genes, the q components of each gene at the q locations of missing values in $\mathbf{g}_1$ are ignored. Then, based on these k neighboring gene vectors, a matrix $A \in \mathbb{R}^{k\times(n-q)}$ a matrix $B \in \mathbb{R}^{k\times q}$ and a vector $\mathbf{w} \in \mathbb{R}^{(n-q)\times 1}$ are formed. The i-th row vector $\mathbf{a}_i^T$ of the matrix A consists of the i-th nearest neighbor genes $\mathbf{g}_{S_i}^T \in \mathbb{R}^{1\times n}$, $1 \le i \le k$,

with its elements at the q missing locations of missing values of $\mathbf{g}_1$ excluded. Each column vector of the matrix B consists of the values of the j-th location of the missing values ($1 \leq j \leq q$) of the k vectors $\mathbf{g}_{S_i}^T$.The elements of the vector $\mathbf{w}$ are the $n-q$ elements of the gene vector $\mathbf{g}$ whose missing items are deleted. After the matrices A and B and a vector $\mathbf{w}$ are formed, the least squares problem is formulated as

$$\min_x ||A^T\mathbf{x} - \mathbf{w}||_2 \tag{13.3}$$

Then, the vector $\mathbf{u} = (\alpha_1, \alpha_2, \ldots, \alpha_q)^T$ of q missing values can be estimated as

$$\mathbf{u} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_q \end{pmatrix} = B^T\mathbf{x} = B^T(A^T)^\dagger\mathbf{w}, \tag{13.4}$$

where $(A^T)^\dagger$ is the pseudoinverse of A^T.

Bibliography

[1] E. Acuna and C. Rodriguez, The treatment of Missing Values and its effect in the classifier accuracy, In: D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds). Classification, Clustering and Data Mining Applications, Springer-Verlag Berlin-Heidelberg, 2004, 639–648.

[2] G.E.A.P.A. Batista and M.C. Monard, An analysis of four Missing Data treatment methods for supervised learning, Applied Artificial Intelligence 17 (2003) 519–533.

[3] C. Chow and C. Liu, Approximating discrete probability distributions with dependence trees, IEEE Transactions on Information Theory 14 (1968) 462–467.

[4] J. Deogun, W. Spaulding, B. Shuart and D. Li, Towards Missing Data Imputation: A study of fuzzy K-means Clustering Method, In Rough Sets and Current Trends in Computing (RSCTC 2004), Lecture Notes in Computer Science 3066, Springer-Verlag, 2004, 573–579.

[5] U.M. Fayyad and K.B. Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, 13th International Joint Conference on Uncertainly in Artificial Intelligence (IJCAI93), 1993, 1022–1029.

[6] H.A.B. Feng, G.C. Chen, C.D. Yin, B.B. Yang and Y.E. Chen, A SVM regression based approach to filling in Missing Values, In Knowledge-Based Intelligent Information and Engineering Systems (KES 2005), Lecture Notes in Artificial Intelligence 3683, 2005, 581–587.

[7] J.W. Grzymala-Busse and L.K. Goodwin, Handling Missing Attribute Values in preterm birth data sets, In Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2005), Lecture Notes in Computer Science 3642, 2005, 342–351.

[8] H. Kim, G.H. Golub, H. Park, Missing value estimation for DNA microarray gene expression data: local least squares imputation, Bioinformatics 21 (2005) 187–198.

[9] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara and S. Ishii, A Bayesian missing value estimation method for gene expression profile data, Bioinformatics 19 (2003) 2088–2096.

[10] T. Schneider, Analysis of incomplete climate data: Estimation of Mean Values and covariance matrices and imputation of Missing values, Journal of Climate 14 (2001) 853–871.

[11] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown , T. Hastie, R. Tibshirani, D. Botstein and R.B. Altman, Missing value estimation methods for DNA microarrays, Bioinformatics 17 (2001) 520–525.

[12] V.N. Vapnik, The Nature of Statistical Learning Theory, NY: Springer-Verlag, 1995.

[13] A.K.C. Wong and D.K.Y. Chiu, Synthesizing statistical knowledge from incomplete mixed-mode data, IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (1987) 796–805.