Mixture of Gaussians for Distance Estimation with Missing Data

Emil Eirola, Amaury Lendasse, Vincent Vandewalle, Christophe Biernacki

Mixture of Gaussians for distance estimation with missing data

Emil Eirola[a,*], Amaury Lendasse[a,b,c,d], Vincent Vandewalle[e,f], Christophe Biernacki[d,f]

[a] *Department of Information and Computer Science, Aalto University, FI–00076 Aalto, Finland*
[b] *IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain*
[c] *Computational Intelligence Group, Computer Science Faculty, University of the Basque Country, Paseo Manuel Lardizabal 1, Donostia/San Sebastián, Spain*
[d] *Laboratoire P. Painlevé, UMR 8524 CNRS Université Lille I, Bât M3, Cité Scientifique, F–59655 Villeneuve d'Ascq Cedex, France*
[e] *EA 2694, Université Lille 2, 1 place de Verdun, 59045 Lille Cedex, France*
[f] *MODAL team, INRIA Lille - Nord Europe, 40 avenue Halley, 59650, Villeneuve d'Ascq, France*

Abstract

Many data sets have missing values in practical application contexts, but the majority of commonly studied machine learning methods cannot be applied directly when there are incomplete samples. However, most such methods only depend on the relative differences between samples instead of their particular values, and thus one useful approach is to directly estimate the pairwise distances between all samples in the data set. This is accomplished by fitting a Gaussian mixture model to the data, and using it to derive estimates for the distances. A variant of the model for high-dimensional data with missing values is also studied. Experimental simulations confirm that the proposed method provides accurate estimates compared to alternative methods for estimating distances. In particular, using the mixture model for estimating distances is on average more accurate than using the same model to impute any missing values and then calculating distances. The experimental evaluation additionally shows that more accurately estimating distances leads to improved prediction performance for classification and regression tasks when used as inputs for a neural network.

Keywords: missing data, distance estimation, mixture model

*Corresponding author
Email address: emil.eirola@aalto.fi (Emil Eirola)

1. Introduction

Missing values are a common phenomenon when dealing with real world data sets. The effects of incomplete data cannot be ignored when designing machine learning procedures. Values could be missing for a variety of reasons depending on the source of the data, including measurement error, device malfunction, operator failure, etc. However, many modelling approaches start with the assumption of a data set with a certain number of samples, and a fixed set of measurements for each sample. Such methods cannot be applied directly if some measurements are missing. Simply discarding the samples or variables which have missing components often means throwing out a large part of data that could be useful for the model. It is relevant to look for better ways of dealing with missing values in such scenarios.

In this paper, the particular problem of estimating distances between samples in a data set with missing values is studied. Being able to appropriately estimate distances between samples, or between samples and prototypes, has numerous applications. It directly enables the use of several powerful statistical and machine learning methods which are based only on distances and not the direct values, including nearest neighbours (k-NN), support vector machines (SVM) with Gaussian kernels, or radial basis function (RBF) neural networks [1].

There are several paradigms for dealing with missing data used in conjunction with machine learning methods [2]. *Conditional mean imputation*, which is optimal in terms of minimising the mean squared error of the imputed values, suffers from leading to biased derived statistics of the data. For instance, estimates of variance or distances are negatively biased. *Random draw imputation* is more appropriate for generating a representative example of a fully imputed data set, but has too much variability in estimates of any single values, or distances between particular samples, to be accurate. *Multiple imputation* (drawing several representative imputations of the data, analysing each set separately, and combining the results) [3] can result in unbiased and accurate estimates after a sufficiently high number of draws, but it is not always straightforward to determine the posterior distribution to draw from [4]. In the context of machine learning, repeating the analysis several times is however impractical as training and analysing a sophisticated model tends to be computationally expensive.

The conceptually simplest approach to dealing with incomplete data is to fill in the missing values before commencing any further analysis. Many methods have been suggested for imputation with the intent to appropriately conform to the distribution of the data. These include imputation by nearest neighbours [5], or the improved incomplete-case k-NN imputation [6]. An alternative approach is to study the input density indirectly through conditional distributions, by fully conditional specification [7]. However, the uncertainty of the imputed values is often not explicitly modelled in most imputation methods, and hence ignored in the further analysis, potentially leading to biased results.

One option for automatic preprocessing of data for machine learning is feature selection, which has been studied in the context of incomplete data in

various ways [8, 9]. A possibility for integrating the imputation of missing values with learning a prediction model is presented in the MLEM2 rule induction algorithm [10]. Another suggested alternative is to use nearest neighbours to simultaneously conduct classification and imputation [11].

Finite mixture models have proven to be a versatile and powerful modelling tool in a wide variety of applications. Particularly mixture models of Gaussians have been studied extensively to describe the distributions of data sets. The general approach to estimate the model parameters from data is maximum likelihood (ML) estimation by the EM algorithm [12]. This has been extended to estimating Gaussian mixture models for data sets with missing values [13, 14]. Recently, Gaussian mixture models as applied to missing data problems have been studied extensively [15, 16, 17].

A way to use mixtures of Gaussians for training neural networks on data with missing values has previously been proposed in [18], involving finding the average gradient of the relevant parameters by integrating over the conditional distribution of missing values. However, the authors only specify widths for the Gaussian components separately for each dimension in their implementation. This simplifies the analysis greatly, effectively ignoring correlations by restricting the covariance matrices to be diagonal. The suggested procedure specifically applies to training the network by back-propagation, and cannot directly be used for other machine learning methods. Another, more limited, approach to directly allow incomplete samples to be used in back-propagation is to flag input neurons corresponding to unknown attributes as protected, temporarily restricting them from being modified [19]. Further suggested approaches to using a mixture of Gaussians to model the input density for machine learning include forming hidden Markov models for speech recognition by integrating over the density [20]. Another analysis accounting for the uncertainty of missing values using a single multivariate Gaussian in clinical trials is to be found in [21].

A variant of Gaussian mixture models for high-dimensional data has been proposed as high-dimensional data clustering (HDDC) [22]. Its use for data with missing values is studied in this paper.

In spite of the large body of work to approach the problem of incomplete data from different angles, very few authors explicitly consider the distances between the samples. The only specific method to be found to directly estimate these distances is the partial distance strategy [23], which is popularly used and efficient but too simple to adequately account for more sophisticated distributions of the data.

In this paper, we propose to apply mixtures of Gaussians to the task of estimating pairwise distance between samples. Using a Gaussian mixture model is appropriate for this problem, as it 1) can be optimised efficiently even in the presence of missing values, 2) allows one to derive estimates of pairwise distances, 3) is flexible enough to cover any distribution of samples, and 4) is sufficiently sophisticated to provide non-linear imputation.

The sequel of this paper is organised as follows. Section 2 reviews the EM algorithm for mixtures of Gaussians, and introduces the extension to missing

data, including how to efficiently conduct the computations and how to extend the method to high-dimensional data. Section 3 presents the estimation of pairwise distances. Section 4 contains comparison experiments on simulations of data with missing values, including results of neural networks for supervised learning tasks using the estimated distances. Section 5 concludes the paper.

2. EM for mixture of Gaussians with missing data

2.1. The standard EM algorithm

Before studying the case with missing data, we present the conventional EM algorithm for fitting a mixture of Gaussians [1, Section 9.2]. Given data $\mathbf{X}$ consisting of a set of N observations $\{\boldsymbol{x}_i\}_{i=1}^N$, we wish to model the data using a mixture of K Gaussian distributions. The log-likelihood is given by

$$\log \mathcal{L}(\theta) = \log p(\mathbf{X} \,|\, \theta) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right), \tag{1}$$

where $\mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the probability density function of the multivariate normal distribution, and $\theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ is the set of parameters to be determined. The log-likelihood can be maximised by applying the EM-algorithm. Initialisation consists in choosing values for the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$, and mixing coefficients π_k for each component k ($0 < \pi_k < 1$, $\sum_{k=1}^K \pi_k = 1$). The E-step is to evaluate the probabilities t_{ik} that $\boldsymbol{x}_i$ is generated by the kth Gaussian using the current parameter values:

$$t_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{2}$$

In the M-step, the parameters are re-estimated with the updated probabilities:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N} t_{ik} \boldsymbol{x}_i, \quad \boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N} t_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T, \quad \pi_k = \frac{N_k}{N}, \tag{3}$$

where $N_k = \sum_{i=1}^N t_{ik}$. The E and M-steps are alternated repeatedly until convergence is observed in the log-likelihood or parameter values.

2.2. Missing data extension of the EM algorithm

An important consideration when dealing with missing data is the missing-data mechanism. We assume that a missing value represents a value which is defined and exists, but for an unspecified reason is unknown. Following the conventions of [2], the assumption here is that data are Missing-at-Random (MAR): $P(M|x_{\text{obs}}, x_{\text{mis}}) = P(M|x_{\text{obs}})$, i.e., the event M of a measurement being missing is independent from the value it would take (x_{mis}), conditional on the observed data (x_{obs}). The stronger assumption of Missing-Completely-at-Random (MCAR) is not necessary, as MAR is an ignorable missing-data

mechanism in the sense that maximum likelihood estimation still provides a consistent estimator [2].

The standard EM algorithm for fitting Gaussian mixture models has been extended to handle data with missing values [13, 14]. The input data $\mathbf{X}$ is now a set of observations $\{\boldsymbol{x}_i\}_{i=1}^N$ such that for each sample there is an index set $O_i \subseteq \{1, \ldots, d\}$ enumerating the observed variables. The indices in the complement set M_i correspond to missing values in the data sample $\boldsymbol{x}_i$. In the case with missing values, the observed log-likelihood can be written as

$$\log \mathcal{L}(\theta) = \log p(\mathbf{X}^O \mid \theta) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{x}_i^{O_i} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right) \tag{4}$$

where $\mathbf{X}^O = \{\boldsymbol{x}_i^O i\}_{i=1}^N$, and as a shorthand of notation, $\mathcal{N}(\boldsymbol{x}_i^{O_i} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is also used for the *marginal* multivariate normal distribution probability density of the observed values of $\boldsymbol{x}_i$.

In the EM algorithm, in order to account for the missing data, some additional expectations need to be computed in the E-step. These include the conditional expectations of the missing components of a sample with respect to each Gaussian component k, and their conditional covariance matrices, i.e., $\tilde{\boldsymbol{\mu}}_{ik}^{M_i} = \mathrm{E}[\boldsymbol{x}_i^{M_i} \mid \boldsymbol{x}_i^{O_i}]$, and $\tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i} = \mathrm{Var}[\boldsymbol{x}_i^{M_i} \mid \boldsymbol{x}_i^{O_i}]$, where the mean and covariance are calculated under the assumption that $\boldsymbol{x}_i$ originates from the kth Gaussian. For convenience, we also define corresponding imputed data vectors $\tilde{\boldsymbol{x}}_{ik}$ and full covariance matrices $\tilde{\boldsymbol{\Sigma}}_{ik}$ which are padded with zeros for the known components. Then the E-step is:

$$t_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i^{O_i} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i^{O_i} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}, \tag{5}$$

$$\tilde{\boldsymbol{\mu}}_{ik}^{M_i} = \boldsymbol{\mu}_k^{M_i} + \boldsymbol{\Sigma}_k^{MO_i} (\boldsymbol{\Sigma}_k^{OO_i})^{-1} (\boldsymbol{x}_i^{O_i} - \boldsymbol{\mu}_k^{O_i}), \quad \tilde{\boldsymbol{x}}_{ik} = \begin{pmatrix} \boldsymbol{x}_i^{O_i} \\ \tilde{\boldsymbol{\mu}}_{ik}^{M_i} \end{pmatrix}, \tag{6}$$

$$\tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i} = \boldsymbol{\Sigma}_k^{MM_i} - \boldsymbol{\Sigma}_k^{MO_i} (\boldsymbol{\Sigma}_k^{OO_i})^{-1} \boldsymbol{\Sigma}_k^{OM_i}, \qquad \tilde{\boldsymbol{\Sigma}}_{ik} = \begin{pmatrix} \mathbf{0}^{OO_i} & \mathbf{0}^{OM_i} \\ \mathbf{0}^{MO_i} & \tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i} \end{pmatrix}. \tag{7}$$

The notation $\boldsymbol{\mu}_k^{M_i}$ refers to using only the elements from the vector $\boldsymbol{\mu}_k$ specified by the index set M_i, and similarly for $\boldsymbol{x}_i^{O_i}$, etc. For matrices, $\boldsymbol{\Sigma}_k^{MO_i}$ refers to elements in the rows specified by M_i and columns by O_i, and so on. The expressions for the parameters in equations (6) and (7) originate from the observation that the conditional distribution of the missing components also follows a multivariate normal distribution, with these parameters [24, Thm. 2.5.1].

The M-step remains functionally equivalent, the only changes being that the component means are estimated from the imputed data vectors and the covariance matrix estimate requires an additional term to include the covariances concerning the imputed values.

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N} t_{ik} \tilde{\boldsymbol{x}}_{ik}, \quad \boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N} t_{ik} \left[(\tilde{\boldsymbol{x}}_{ik} - \boldsymbol{\mu}_k)(\tilde{\boldsymbol{x}}_{ik} - \boldsymbol{\mu}_k)^T + \tilde{\boldsymbol{\Sigma}}_{ik} \right], \quad \pi_k = \frac{N_k}{N}. \tag{8}$$

2.3. High-dimensional data

As the number of free parameters grows with the square of the data dimension, in high-dimensional cases it is often not possible to fit a conventional Gaussian mixture model, or even a model with a single Gaussian component. A version of Gaussian mixture models, *high-dimensional data clustering (HDDC)*, has been proposed for this scenario [22] where in the M-step, the covariances matrices are replaced by a reduced representation. Applying this idea to the case of missing data is possible by modifying the covariance matrices of each component after calculating them in the M-step.

From the different variants of HDDC, the experiments in this paper allow all the parameters of the reduced representation to be determined freely for each Gaussian (the model $[a_{ij}b_iQ_id_i]$ in the notation of [22]). Following [22], the number of significant eigenvalues is selected by the scree test [25], where the dimension is selected when the subsequent eigenvalues have a difference smaller than a specified threshold (0.001 of the trace of the covariance matrix).

In brief (see [22] for details and derivations), the reduced representation entails taking the eigenvalue decomposition of the covariance matrix

$$\boldsymbol{\Sigma}_k = \mathbf{Q}_k \boldsymbol{\Lambda}_k \mathbf{Q}_k^T$$

and modifying all the eigenvalues λ_{kj} apart from some of the largest ones. The scree test is used to find the number of eigenvalues d_k to be kept exactly. The remaining eigenvalues are replaced by their arithmetic mean. In other words, the covariance matrix $\boldsymbol{\Sigma}_k$ is replaced by a matrix $\boldsymbol{\Sigma}'_k = \mathbf{Q}_k \boldsymbol{\Lambda}'_k \mathbf{Q}_k^T$ where $\boldsymbol{\Lambda}'_k$ is a diagonal matrix with the elements

$$\lambda'_{kj} = \begin{cases} \lambda_{kj} & j \leq d_k \\ b_k & j > d_k \end{cases}$$

where

$$b_k = \frac{1}{d - d_k} \sum_{l=d_k+1}^{d} \lambda_{kl} = \frac{1}{d - d_k} \left(\operatorname{trace}(\boldsymbol{\Sigma}_k) - \sum_{l=1}^{d_k} \lambda_{kl} \right).$$

assuming the eigenvalues λ_{kj} are in decreasing order. This representation implies that only the first d_k eigenvalues and eigenvectors need to be calculated and stored, efficiently reducing the number of free parameters required to specify each Gaussian component.

2.4. Initialisation

In our implementation of the EM algorithm with missing data, the means $\boldsymbol{\mu}_k$ are initialised by a random selection from the observed data, preferring samples without missing values if available. When there are insufficient complete samples, some of the Gaussian means are initialised by incomplete samples where the missing values are imputed by the sample mean. The covariances $\boldsymbol{\Sigma}_k$ are initialised with the sample covariance of the data (ignoring samples with missing values). Alternatively, the covariances can be initialised as diagonal matrices, using only the sample variance of each variable.

2.5. Efficient EM iterations with the sweep operator

The most computationally consuming part of the EM algorithm is to deal with the inverses of the covariance matrices in equations (6) and (7), as the specific matrix to be inverted depends on the pattern of missing values for each data sample. A naive implementation would require inverting a matrix for each sample and Gaussian component every iteration, which is time consuming. A more efficient procedure for the same end result is to use the sweep operator [2]. A symmetric matrix G is said to be swept on row and column m when it results in a matrix $H = \mathrm{SWP}[m]G$ with elements:

$$\begin{aligned} h_{mm} &= -1/g_{mm} & \\ h_{jm} = h_{mj} &= g_{jm}/g_{mm} & j \neq m \\ h_{jl} &= g_{jl} - g_{jm}g_{ml}/g_{mm} & j, l \neq m \end{aligned}$$

Sweeping over several rows and columns can be conducted sequentially, denoted as $\mathrm{SWP}[m_1, m_2, \ldots, m_t] = \mathrm{SWP}[m_1]\,\mathrm{SWP}[m_2]\ldots\mathrm{SWP}[m_t]$. Furthermore, it can be shown that the sweep operator is commutative, and thus when sweeping over a set of indices it is not necessary to specify the order in which the sweeps are done. Hence it is possible to simplify and consider the process of sweeping over a set of indices, meaning sequentially sweeping over each index in the set, in any order. Now to apply the sweep operator to the present situation in order to find the conditional expectation and covariance distribution of a sample $\boldsymbol{x}_i$ with respect to the Gaussian component k, form the $(d+1) \times (d+1)$ matrix:

$$G = \begin{bmatrix} \boldsymbol{\Sigma}_k^{MM_i} & \boldsymbol{\Sigma}_k^{MO_i} & \boldsymbol{\mu}_k^{M_i} \\ \boldsymbol{\Sigma}_k^{OM_i} & \boldsymbol{\Sigma}_k^{OO_i} & \boldsymbol{\mu}_k^{O_i} - \boldsymbol{x}_i^{O_i} \\ (\boldsymbol{\mu}_k^{M_i})^T & (\boldsymbol{\mu}_k^{O_i} - \boldsymbol{x}_i^{O_i})^T & 0 \end{bmatrix}$$

Then sweeping over all observed variables O_i results in

$$\mathrm{SWP}[O_i]G = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i} & \boldsymbol{\Sigma}_k^{MO_i}(\boldsymbol{\Sigma}_k^{OO_i})^{-1} & \tilde{\boldsymbol{\mu}}_{ik}^{M_i} \\ (\boldsymbol{\Sigma}_k^{OO_i})^{-1}\boldsymbol{\Sigma}_k^{OM_i} & (\boldsymbol{\Sigma}_k^{OO_i})^{-1} & (\boldsymbol{\Sigma}_k^{OO_i})^{-1}(\boldsymbol{x}_i^{O_i} - \boldsymbol{\mu}_k^{O_i}) \\ (\tilde{\boldsymbol{\mu}}_{ik}^{M_i})^T & (\boldsymbol{x}_i^{O_i} - \boldsymbol{\mu}_k^{O_i})^T(\boldsymbol{\Sigma}_k^{OO_i})^{-1} & a \end{bmatrix}$$

where

$$\begin{aligned} \tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i} &= \boldsymbol{\Sigma}_k^{MM_i} - \boldsymbol{\Sigma}_k^{MO_i}(\boldsymbol{\Sigma}_k^{OO_i})^{-1}\boldsymbol{\Sigma}_k^{OM_i} \\ \tilde{\boldsymbol{\mu}}_{ik}^{M_i} &= \boldsymbol{\mu}_k^{M_i} - \boldsymbol{\Sigma}_k^{MO_i}(\boldsymbol{\Sigma}_k^{OO_i})^{-1}(\boldsymbol{\mu}_k^{O_i} - \boldsymbol{x}_i^{O_i}) \\ a &= -(\boldsymbol{\mu}_k^{O_i} - \boldsymbol{x}_i^{O_i})^T(\boldsymbol{\Sigma}_k^{OO_i})^{-1}(\boldsymbol{\mu}_k^{O_i} - \boldsymbol{x}_i^{O_i}) \end{aligned}$$

Here the top left part of the matrix is the conditional covariance of the missing values ($\tilde{\boldsymbol{\Sigma}}_{ik}^{MM_i}$), the vector at the top right is the conditional mean ($\tilde{\boldsymbol{\mu}}_{ik}^{M_i}$), and the scalar value in the bottom right is necessary for finding the value of the marginal probability density for evaluating the log-likelihood.

However, in most data sets there are more observed data points than missing values, and hence it is more efficient to sweep over the missing values rather

than the observed ones. This can be accomplished by applying the inverse of the sweep operator – the reverse sweep operator – denoted $H = \mathrm{RSWP}[m]G$ and defined by its elements:

$$\begin{aligned} h_{mm} &= -1/g_{mm} \\ h_{jm} = h_{mj} &= -g_{jm}/g_{mm} && j \neq m \\ h_{jl} &= g_{jl} - g_{jm}g_{ml}/g_{mm} && j, l \neq m \end{aligned}$$

Instead of calculating $H = \mathrm{SWP}[O_i]G$ directly, we can find the same result by computing $H = \mathrm{RSWP}[M_i]G'$ instead, where

$$G' = \mathrm{SWP}[1, \ldots, d]G = \begin{bmatrix} \boldsymbol{\Sigma}_k^{-1} & \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \boldsymbol{x}_i) \\ (\boldsymbol{\mu}_k - \boldsymbol{x}_i)^T\boldsymbol{\Sigma}_k^{-1} & -(\boldsymbol{\mu}_k - \boldsymbol{x}_i)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k - \boldsymbol{x}_i) \end{bmatrix}$$

where for this calculation, zeros are inserted for the missing values of $\boldsymbol{x}_i$. Note that forming G' does not require the matrix $\boldsymbol{\Sigma}_k$ to be inverted repeatedly for each sample $\boldsymbol{x}_i$.

For samples with no missing values, it is only necessary to evaluate $(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)$ for calculating the value of the probability density. The covariance matrices $\boldsymbol{\Sigma}_k$ can be inverted in $O(Kd^3)$ operations, and the vector-matrix-vector product requires $O(d^2)$ operations for each sample and component. Sweeping over one index in G' requires $O(d^2)$ operations. Every missing value in the data set requires one sweep operation for each Gaussian component. The M-step is computationally less intensive, but estimating the updated covariance matrices still has a complexity of $O(NKd^2)$. Adding it all up, the complexity of one iteration of the EM algorithm is $O(Kd^3 + NKd^2 + Kd^2L)$, where $L = \sum_{i=1}^{N} |M_i|$ is the total number of missing values in the data set.

2.6. Model selection

The number of components is selected according to the Akaike information criterion [26] with the small sample (second-order) bias adjustment [27]. Using the corrected version is crucial, as the number of parameters grows relatively large when increasing the number of components. The corrected Akaike information criterion is a function of the log-likelihood:

$$\mathrm{AIC}_C = -2\log\mathcal{L}(\theta) + 2P + \frac{2P(P+1)}{N-P-1} \tag{9}$$

where P is the number of free parameters. $P = Kd + K - 1 + \frac{1}{2}Kd(d+1)$ in the case of full, separate, covariance matrices for each of the K components. With high-dimensional data sets, the number of parameters quickly tends to become larger than the number of available samples when increasing the number of components, and the criterion would not be valid anymore. This effect can be mitigated by imposing restrictions on the structure of the covariance matrices, but this would also make the model less powerful.

3. Distance estimation with missing data

The intended application of the mixture of Gaussians model is to use it for distance estimation. The problem of estimating distances between samples with missing data is non-trivial, since even perfect imputation (by the conditional expectation) results in biased estimates for the distance. Using additional knowledge about the distribution of the data leads to more accurate estimates.

In the following, we focus on calculating the expectation of the *squared* Euclidean (ℓ^2) distance. Estimating the ℓ^2-norm itself could be feasible, but due to the square-root, the expressions do not simplify and separate as cleanly. Another motivation for directly estimating the squared distance is that many methods for further processing of the distance matrix actually only make use of the squared distances (e.g., RBF and SVM), while others only consider the ranking of the distances (nearest neighbours).

Given two samples $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$ with components x_i^l, x_j^l $(1 \leq l \leq d)$, which may contain missing values, denote again by M_i and O_i the set of indices of the missing and, respectively, observed components for each sample $\boldsymbol{x}_i$. Partition the index set into four parts based on the missing components, and the expression for the squared distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ can be split accordingly:

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \sum_{l=1}^{d} (x_i^l - x_j^l)^2 = \sum_{l \in O_i \cap O_j} (x_i^l - x_j^l)^2 + \sum_{l \in O_i \cap M_j} (x_i^l - x_j^l)^2 + \sum_{l \in M_i \cap O_j} (x_i^l - x_j^l)^2 + \sum_{l \in M_i \cap M_j} (x_i^l - x_j^l)^2.$$

The missing values can be modelled as random variables, X_i^l, $l \in M_i$. Taking the expectation of the above expression, by the linearity of expectation:

$$\begin{aligned} \mathrm{E}\left[\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right] = &\sum_{l \in O_i \cap O_j} (x_i^l - x_j^l)^2 + \sum_{l \in O_i \cap M_j} \left((x_i^l - \mathrm{E}[X_j^l])^2 + \mathrm{Var}[X_j^l]\right) \\ &+ \sum_{l \in M_i \cap O_j} \left((\mathrm{E}[X_i^l] - x_j^l)^2 + \mathrm{Var}[X_i^l]\right) \\ &+ \sum_{l \in M_i \cap M_j} \left((\mathrm{E}[X_i^l] - \mathrm{E}[X_j^l])^2 + \mathrm{Var}[X_i^l] + \mathrm{Var}[X_j^l]\right). \end{aligned}$$

In the final summation, it is necessary to consider X_i^l and X_j^l to be uncorrelated, given the known values of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. This assumption is not restrictive, and follows directly from the common approach that samples are independent draws from an unknown multivariate distribution.

It thus suffices to find the expectation and variance of each random variable separately. If the original samples $\boldsymbol{x}_i$ are thought to originate as independent draws from a multivariate distribution, the distributions of the random variables X_i^l can be found as the conditional distribution when conditioning their joint distribution on the observed values. Then finding the expected squared distance

between two samples reduces to finding the (conditional on the observed values) expectation and variance of each missing component separately. Define $\tilde{\boldsymbol{x}}_i$ to be an imputed version of $\boldsymbol{x}_i$ where each missing value has been replaced by its conditional mean, and define $\tilde{\sigma}_i^l$ as the corresponding conditional variance:

$$\tilde{x}_i^l = \begin{cases} \mathrm{E}[X_i^l \,|\, \boldsymbol{x}_i^{O_i}] & \text{if } l \in M_i, \\ x_i^l & \text{otherwise} \end{cases} \qquad \tilde{\sigma}_i^l = \begin{cases} \mathrm{Var}[X_i^l \,|\, \boldsymbol{x}_i^{O_i}] & \text{if } l \in M_i, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

With these notations, the expectation of the squared distance can conveniently be expressed as:

$$\mathrm{E}\left[\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right] = \|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|^2 + s_i + s_j, \quad \text{where} \quad s_i = \sum_{l \in M_i} \tilde{\sigma}_i^l. \tag{11}$$

This form of the expression particularly emphasises how the uncertainty of the missing values is accounted for. The first term – the distance between imputed samples – already provides an estimate of the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, but including the variances of each imputed component is the deciding factor.

The conditional means and covariances can be calculated using the Gaussian mixture model. These are calculated separately for each component in the M-step, and it only remains to determine the overall conditional mean and covariance matrix. These are found weighted by the memberships as follows:

$$\tilde{\boldsymbol{x}}_i = \sum_{k=1}^{K} t_{ik} \tilde{\boldsymbol{x}}_{ik}, \qquad \tilde{\boldsymbol{\Sigma}}_i = \sum_{k=1}^{K} t_{ik} \left(\tilde{\boldsymbol{\Sigma}}_{ik} + \tilde{\boldsymbol{x}}_{ik} \tilde{\boldsymbol{x}}_{ik}^T \right) - \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^T. \tag{12}$$

The expression for the covariance is found by direct calculation of the second moments. In order to estimate pairwise distances, the conditional variances $\tilde{\sigma}_i^l = \tilde{\boldsymbol{\Sigma}}_i^{ll}$ can be extracted from the diagonal of the conditional covariance matrix, or s_i calculated directly as the trace of $\tilde{\boldsymbol{\Sigma}}_i$.

In summary, the complete procedure for estimating distances consists of the following steps:

1. Fit a Gaussian mixture model of K components by the EM algorithm, for values of K from 1 up to a chosen maximum.
2. Calculate the log-likelihood and AIC_C for each model.
3. Choose the model which minimises the AIC_C, and apply it calculate conditional means and variances for each missing value in the data set.
4. Estimate distances between samples using the conditional means and variances.

4. Experiments

To study the effectiveness of the proposed approach, some simulated experiments are conducted to compare the algorithm to alternative methods on several data sets. The accuracy of the estimated distances is directly evaluated by different criteria, and also through the performance of neural network classifiers

Table 1: Data sets used for the experiments, with the number of samples (N), number of variables (d), supervised learning task, and source.

Name	N	d	Task	Source
Computer Hardware	209	66	Regression	[28]
Glass Identification	214	9	6-class	[28]
Housing	506	9	Regression	[28]
Image Segmentation [1]	2310	14	7-class	[28]
Iris	150	4	3-class	[28]
Pima Indians Diabetes	768	8	2-class	[28]
Servo	167	4	Regression	[28]
Stocks	950	9	Regression	[29]
Statlog (Vehicle Silhouettes)	846	18	4-class	[28]
Auto-price	159	15	Regression	[29]
Breast Cancer Wisconsin (Diagnostic)	569	30	2-class	[28]
Breast Cancer Wisconsin (Prognostic)	194	32	Regression	[28]
Breast Tissue	106	9	6-class	[28]
Ecoli	336	7	4-class	[28]
Financial Ratios [2]	500	40	2-class	[5]
Ionosphere	351	33	2-class	[28]
Parkinsons [3]	195	20	2-class	[28]
Spambase [4]	4601	57	2-class	[28]
SPECTF Heart	267	44	2-class	[28]
Wine	178	13	3-class	[28]
BCI	400	117	2-class	[6]
COIL	1500	241	6-class	[6]
Tecator	240	100	Regression	[7]

[1] Discarding the attributes 3, 10, 14, 15, and 16, as they are averages of other variables
[2] Discarding variable SF7 which is merely the sum of SF10 and LI8
[3] Discarding Jitter:DDP and Shimmer:DD, being duplicates of MDVP:RAP and Shimmer:APQ3
[4] Preprocessed by taking the logarithm of each value
[5] Available at http://research.ics.aalto.fi/eiml/datasets.shtml
[6] Available at http://olivier.chapelle.cc/ssl-book/benchmarks.html; see also [30]
[7] Available at http://www.dm.unibo.it/~simoncin/tecator

and regression models built using the estimated distances. Starting with a complete data set, values are removed at random with a fixed probability. As the true distances between samples are known, the methods can then be compared on how well they estimate the distances after values have been removed.

4.1. Data

Several different data sets are used for the experiments, and they are listed in Table 1. Some data sets contain variables which are direct linear combinations of other variables, and as this induces unnecessary computational difficulties, some variables are discarded as detailed in the table. The data sets contain binary classification, multiclass classification, and regression tasks. As the problem of pairwise distance estimation is unsupervised, the outputs for the samples are ignored while estimating distances.

To make distances meaningful, the variables in each data set are standardised to zero mean and unit variance before values are removed. This standardisation is conducted only in order to have comparable error rates between repeated

experiments, and the methods used do not depend on the variables being standardised.

4.2. Methods

The Gaussian mixture model approach is compared to two other methods for estimating distances:

PDS The Partial Distance Strategy [23]. Calculate the sum of squared differences of the mutually known components and scale to the missing components:

$$\hat{d}_{ij}^2 = \frac{d}{|O_i \cap O_j|} \sum_{l \in O_i \cap O_j} \left(x_i^l - x_j^l\right)^2. \tag{13}$$

For samples which have no known components in common, the method is not defined. For such pairs, the average of the pairwise distances which were possible to estimate is returned instead.

ICkNNI Incomplete-case k-NN imputation [6]. An improvement of complete-case k-NN imputation, here any sample with a valid pattern of missing values is viable nearest neighbour. In accordance to the suggestions in [6], up to $k = 5$ neighbours are considered. The imputation fails whenever there are no samples with such valid patterns. For these cases, the missing value is imputed by the sample mean for that variable.

In addition, the mixture model is compared to estimating the distribution by a single Gaussian. As another alternative, the distances are estimated after using the mixture model for imputation by the conditional mean – equivalent to discarding the variance terms of Eq. (11). Using a single Gaussian for imputation by the conditional mean is an interesting special case, as it is equivalent to a least-squares linear regression.

4.3. Performance criteria

The methods are evaluated by three different performance criteria. First, the methods are compared by the root mean squared error (RMSE) of all the estimated pairwise distances in the data set,

$$C_1 = \left(\frac{1}{\lambda} \sum_{i>j} (\hat{d}_{ij} - d_{ij})^2\right)^{1/2}$$

where d_{ij} is the true Euclidean distance between samples i and j calculated without any missing data, and $\hat{d}_{ij}$ is the estimate of the distance provided by each method after removing data. The scaling factor λ is determined so that the average is calculated only over those distances which are estimates, discarding all the cases where the distance can be calculated exactly because neither sample has any missing components: $\lambda = MN - M(M+1)/2$, where M is the number of samples having missing values.

A common application for pairwise distances is a nearest neighbour search, and thus we also consider the average (true) distance to the predicted nearest neighbour,

$$C_2 = \frac{1}{N}\sum_{i=1}^{N} d_{i,\mathrm{NN}(i)}, \quad \text{where} \quad \mathrm{NN}(i) = \arg\min_{j \neq i} \hat{d}_{ij}$$

Here, $\mathrm{NN}(i)$ is the nearest neighbour of the ith sample as estimated by the method, and $d_{i,\mathrm{NN}(i)}$ is the true Euclidean distance between the samples as calculated without any missing data. The criterion measures how well the method can identify samples which actually are close in the real data.

In addition, the accuracy is evaluated by the mean relative error of all pairwise distances:

$$C_3 = \frac{1}{\lambda}\sum_{i>j} \frac{\left|\hat{d}_{ij} - d_{ij}\right|}{d_{ij}}$$

This criterion gives more weight to small distances, and is also an ℓ^1-type error. (As some data sets contain duplicate samples, sample pairs i, j where $d_{ij} = 0$ are ignored when calculating this criterion.)

4.4. Procedure

Values are removed from the data set independently at a fixed probability p. For each value of p, 100 repetitions are conducted for the Monte Carlo simulation, and simulations are run for value of p of 5% (low ratio of missing values), 20%, and 50% (high ratio of missing values). The EM algorithm is run for 200 iterations, and repeated for a total of 5 times for each number of components. Runs are aborted if a covariance matrix becomes too poorly conditioned (condition number over 10^{12}). The best solution in terms of log-likelihood is selected, and the number of components is selected by the AIC_C criterion.

Having 100 repetitions of the same set-up enables the use of statistical significance testing to assess the difference between the mean errors of different methods. The testing is conducted as a two-tailed paired t-test, with a significance level of $\alpha = 0.05$. Comparing the performance of the best method to that of every other method results in a multiple hypothesis scenario, and thus the Bonferroni correction [31] is used to control the error rate.

4.5. Distance estimation results

The average RMSE values for the methods are presented in Table 2. The data sets are grouped into three categories as follows, depending on which mixture model is applicable. The first category includes data for which the EM algorithm converges appropriately with at least two components, and here the number of components is selected by the AIC_C criterion. For data in the second group, the EM algorithm either did not converge with two components, or the AIC_C indicated that a single component is clearly sufficient. Finally there are

the high-dimensional data sets (BCI, COIL, Tecator), for which the EM algorithm would not converge even with one component, and we apply the HDDC instead.

The most immediate observation is that using a single Gaussian or the mixture model tends to give the most accurate results for most data sets. For certain data (Computer Hardware, Ecoli, Image Segmentation), it appears that ICkNNI leads to better estimates.

For many data sets in the first category (Stocks, Iris, Statlog), the mixture model provides a clear advantage over using a single Gaussian. In other cases the differences are not significant, meaning that it may be sufficient to model the data with one Gaussian for this purpose, and indeed several runs resulted in mixture models of consisting of a single Gaussian, as evidenced by the mean K values of less than two.

For most data sets, it can also be seen that including the variance terms of Equation (11) tends to lead to an improvement in the accuracy compared to only conducting imputation.

Table 3 shows the corresponding performances in terms of the true distance to the predicted nearest neighbour. This criterion is more emphasised on the accuracy of small distances, and reveals similar behaviour between the methods on most data sets. There are some notable differences, such as for the Computer Hardware data, where ICkNNI is the most accurate in terms of RMSE, but the mixture model approach is better for estimating nearest neighbours. For most data sets in the first group, the difference between using the mixture model and a single Gaussian is more pronounced than in Table 2.

Table 4 shows the mean relative error of each method. The relative performances are otherwise similar, except that in nearly all cases, using the mixture model for imputation is more accurate in terms of average relative error than directly estimating distances.

In terms of computational resources, the mixture model is more expensive than PDS or ICkNNI (which do not require multiple iterations over the data), but correspondingly delivers more accurate results. As such, the mixture model can be recommended in any situation where the computational task is feasible. Comparing imputation by the mixture model to using the same model for directly estimating distances, there is no difference in computational cost, but the directly estimated distances tend to be more accurate.

4.6. Regression and classification results

In order to study the effect of estimated distances on the performance of supervised learning tasks we assume a semi-supervised learning scenario. In particular, we have a full set of input samples, but labels only for a subset of them, and the task is to label the unlabelled samples. In the current situation this implies that the distances are estimated using the full set of input samples (ignoring labels), and a machine learning model is built using the distances between the labelled samples only (i.e., the "training set").

The model used is an Extreme Learning Machine model with RBF kernels [32, 33]. The kernel centres $\boldsymbol{\mu}_j$ are randomly selected samples from the training

set, and widths σ_j are assigned randomly. The number of neurons is fixed at $N/10$, where N is the number of samples. The model is not optimised at all so that the comparison would be as fair as possible, i.e., the models are exactly the same, the only change is the estimated distances (and the final weights of the output layer after training the ELM). The training phase only consists in finding the least-squares solution to the linear system

$$\mathbf{H}\beta = \mathbf{T}$$

where $\mathbf{T}$ is the target output of the labelled data and the hidden layer output matrix $\mathbf{H}$ has the elements

$$H_{ij} = \exp\left(-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\right\|^2}{\sigma_j^2}\right)$$

as well as a constant column of ones to account for the bias term. In the current scenario, $\mathbf{T}$, $\boldsymbol{\mu}_j$, and σ_j are equal for each method. The only difference is how the distance $\left\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\right\|$ is estimated considering the missing values.

In the simulations, the data samples were randomly partitioned into 75% for training and 25% for testing.

The results in terms of average prediction MSE (normalised by the variance of the target variable) for regression tasks are shown in Table 5, and prediction accuracies for classification tasks in Table 6. In a majority of cases, using a mixture model or single Gaussian to estimate distances results in more accurate predictions than PDS or ICkNNI. Comparing the relative performances to the relative accuracies in Table 2, it seems evident that, in general, more accurately estimated distances lead to a better performance in prediction tasks.

5. Conclusions

In this paper, we present how mixtures of Gaussians can be used to effectively estimate pairwise distances in incomplete data. The problem of estimating distances in a data set with missing values can be reduced to finding the conditional means and variances separately for each missing value. Having a Gaussian mixture model of the distribution of the data enables these quantities to be estimated. In order to fit the mixture model, certain extensions to the standard EM algorithm are presented in Section 2, including a variant specifically for high-dimensional data.

The combination of these ideas provides for a method to estimate distances, and the simulations in Section 4 show that the method is competitive, if not better, than alternative methods in terms of accuracy. The increased accuracy in estimated distances is reflected in an improved prediction performance of neural networks built using the distances.

For future work, it remains to investigate the most effective ways to extend the method to high-dimensional cases where the number of dimensions would exceed the number of samples. Furthermore, it will be interesting to study the influence of the distance estimation when used with other machine learning methods, such as SVM.

Table 2: Average RMSE of estimated pairwise distances. The best result for each row is underlined, and any results which are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.05$) from the best result are bolded. The values in parenthesis represent the accuracy when the distances are calculated using the particular model for imputation only. The final column shows the mean number of Gaussian components K as selected by the AIC_C criterion, and the mean number of distinct eigenvalues d_k for HDDC.

		PDS	ICkNNI	Single Gaussian		Mixture model		Mean K
Computer Hardware	5%	0.676	**0.408**	0.459	(0.443)	0.451	(**0.441**)	3.82
$N = 209$, $d = 6$	20%	1.102	**0.710**	0.736	(0.737)	**0.732**	(**0.730**)	3.70
	50%	1.938	1.340	**1.273**	(1.327)	1.331	(1.341)	3.52
Glass Identification	5%	0.526	0.337	0.226	(**0.221**)	**0.231**	(**0.228**)	2.31
$N = 214$, $d = 9$	20%	0.971	0.706	**0.524**	(**0.522**)	**0.519**	(0.532)	2.51
	50%	2.098	1.540	**1.197**	(1.281)	**1.197**	(1.265)	2.45
Housing	5%	0.514	**0.329**	0.338	(0.348)	**0.331**	(0.338)	3.34
$N = 506$, $d = 13$	20%	1.001	0.672	0.597	(0.650)	**0.587**	(0.619)	3.26
	50%	2.269	1.593	**1.066**	(1.330)	1.104	(1.245)	3.21
Image Segmentation	5%	0.632	**0.376**	0.389	(0.397)	**0.399**	(**0.395**)	4.24
$N = 2310$, $d = 14$	20%	1.254	0.774	**0.720**	(0.759)	**0.778**	(**0.778**)	4.12
	50%	2.801	1.788	**1.364**	(1.560)	1.439	(1.490)	3.99
Iris	5%	0.440	0.246	0.253	(0.257)	**0.219**	(0.225)	3.00
$N = 150$, $d = 4$	20%	0.676	0.391	0.379	(0.401)	**0.335**	(0.358)	2.49
	50%	1.106	0.898	0.766	(0.897)	**0.738**	(0.845)	2.79
Pima Indians	5%	0.540	0.413	**0.388**	(0.418)	0.395	(0.416)	4.74
$N = 768$, $d = 8$	20%	0.957	0.692	**0.600**	(0.714)	0.610	(0.691)	4.47
	50%	1.910	1.329	**0.973**	(1.441)	0.998	(1.335)	4.27
Servo	5%	0.575	0.463	**0.383**	(0.425)	0.399	(0.437)	3.41
$N = 167$, $d = 4$	20%	0.864	0.629	**0.509**	(0.610)	0.536	(0.620)	3.86
	50%	1.217	1.089	**0.761**	(1.122)	0.835	(1.110)	4.14
Stocks	5%	0.350	**0.062**	0.183	(0.182)	0.076	(0.077)	9.44
$N = 950$, $d = 9$	20%	0.649	0.179	0.343	(0.352)	**0.147**	(0.150)	8.72
	50%	1.516	0.864	0.780	(0.897)	**0.479**	(0.533)	6.69
Statlog	5%	0.343	0.199	0.156	(0.161)	**0.142**	(0.147)	2.62
$N = 846$, $d = 18$	20%	0.728	0.562	0.308	(0.331)	**0.287**	(0.305)	2.60
	50%	1.779	1.831	0.708	(0.813)	**0.645**	(0.731)	2.11

		PDS	ICkNNI	Single Gaussian	
Auto-price	5%	0.368	0.263	**0.236**	(0.239)
$N = 159$, $d = 15$	20%	0.750	0.733	**0.455**	(0.485)
	50%	1.820	1.687	**0.871**	(1.015)
Breast Cancer (Diag.)	5%	0.358	0.260	**0.141**	(0.142)
$N = 569$, $d = 30$	20%	0.806	1.086	**0.344**	(0.352)
	50%	1.913	2.551	**0.802**	(0.871)
Breast Cancer (Prog.)	5%	0.349	0.298	**0.166**	(0.169)
$N = 194$, $d = 32$	20%	0.790	1.071	**0.383**	(0.395)
	50%	1.872	2.612	**0.843**	(0.911)
Breast Tissue	5%	0.424	0.243	**0.221**	(**0.222**)
$N = 106$, $d = 9$	20%	0.784	0.586	**0.415**	(0.420)
	50%	1.783	1.594	**0.959**	(0.996)
Ecoli	5%	0.717	**0.432**	0.441	(0.440)
$N = 336$, $d = 7$	20%	1.239	**0.737**	**0.745**	(0.767)
	50%	2.300	1.525	**1.387**	(1.591)
Financial Ratios	5%	0.682	0.483	0.391	(**0.387**)
$N = 500$, $d = 40$	20%	1.516	1.447	**0.910**	(**0.905**)
	50%	3.207	3.152	**1.741**	(1.794)
Ionosphere	5%	0.276	0.257	0.243	(**0.223**)
$N = 351$, $d = 33$	20%	0.624	1.099	0.584	(**0.514**)
	50%	1.550	2.603	1.112	(**1.073**)
Parkinsons	5%	0.346	0.264	**0.188**	(0.193)
$N = 195$, $d = 20$	20%	0.742	0.901	**0.385**	(0.409)
	50%	1.780	2.066	**0.772**	(0.892)
Spambase	5%	0.988	0.695	**0.657**	(0.688)
$N = 4601$, $d = 57$	20%	2.205	1.859	**1.364**	(1.609)
	50%	4.664	3.940	**2.236**	(3.280)
SPECTF Heart	5%	0.334	0.331	**0.203**	(0.206)
$N = 267$, $d = 44$	20%	0.763	1.190	**0.491**	(0.495)
	50%	1.815	2.970	**1.066**	(1.141)
Wine	5%	0.361	0.267	**0.248**	(0.262)
$N = 178$, $d = 13$	20%	0.734	0.623	**0.469**	(0.540)
	50%	1.783	1.469	**0.881**	(1.173)

		PDS	ICkNNI	HDDC mixture		Mean d_k
BCI	5%	0.284	0.548	**0.108**	(0.113)	23.31
$N = 400$, $d = 117$	20%	0.649	1.815	**0.249**	(0.273)	23.96
	50%	1.518	4.768	**0.610**	(0.692)	18.19
COIL	5%	0.404	0.689	**0.202**	(0.224)	26.23
$N = 1500$, $d = 241$	20%	0.926	2.538	**0.488**	(0.630)	26.10
	50%	2.123	6.709	**1.067**	(1.625)	21.23
Tecator	5%	0.052	1.163	0.004	(**0.004**)	4.00
$N = 240$, $d = 100$	20%	0.120	2.366	0.008	(**0.008**)	4.00
	50%	0.282	5.218	0.017	(**0.016**)	4.00

Table 3: Average of the mean distance to the estimated nearest neighbour. The best result for each row is underlined, and any results which are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.05$) from the best result are bolded. The values in parenthesis represent the accuracy when the distances are calculated using the particular model for imputation only. The final column shows the mean number of Gaussian components K as selected by the AIC_C criterion, and the mean number of distinct eigenvalues d_k for HDDC.

		PDS	ICkNNI	Single Gaussian		Mixture model		Mean K
Computer Hardware	5%	0.680	0.516	0.511	(0.521)	**0.496**	(0.516)	3.82
$N = 209$, $d = 6$	20%	1.205	0.834	0.801	(0.837)	**0.763**	(0.832)	3.70
	50%	1.822	1.507	**1.321**	(1.464)	**1.334**	(1.486)	3.52
Glass Identification	5%	1.070	0.919	0.882	(0.887)	**0.879**	(0.886)	2.31
$N = 214$, $d = 9$	20%	1.718	1.245	1.100	(1.133)	**1.088**	(1.129)	2.51
	50%	2.877	2.153	**1.751**	(1.882)	1.797	(1.933)	2.45
Housing	5%	1.047	0.901	0.911	(0.907)	**0.886**	(0.894)	3.34
$N = 506$, $d = 13$	20%	1.790	1.376	1.299	(1.309)	**1.237**	(1.277)	3.26
	50%	3.692	2.744	**2.086**	(2.228)	**2.073**	(2.225)	3.21
Image Segmentation	5%	0.772	**0.501**	0.520	(0.518)	**0.501**	(0.511)	4.24
$N = 2310$, $d = 14$	20%	1.468	0.980	0.928	(0.929)	**0.889**	(0.917)	4.12
	50%	4.029	2.275	**1.684**	(1.791)	1.727	(1.776)	3.99
Iris	5%	0.469	0.360	0.344	(0.361)	**0.339**	(0.354)	3.00
$N = 150$, $d = 4$	20%	1.027	0.550	0.478	(0.546)	**0.459**	(0.530)	2.49
	50%	1.492	1.058	0.864	(1.037)	**0.840**	(1.028)	2.79
Pima Indians	5%	1.530	1.225	**1.176**	(1.212)	**1.175**	(1.215)	4.74
$N = 768$, $d = 8$	20%	2.680	1.728	**1.542**	(1.675)	**1.546**	(1.691)	4.47
	50%	3.292	2.612	**2.215**	(2.510)	2.263	(2.550)	4.27
Servo	5%	1.090	0.849	**0.754**	(0.821)	**0.759**	(0.830)	3.41
$N = 167$, $d = 4$	20%	1.772	1.191	**0.961**	(1.138)	0.985	(1.157)	3.86
	50%	2.048	1.788	**1.456**	(1.745)	1.565	(1.775)	4.14
Stocks	5%	**0.241**	**0.242**	0.293	(0.285)	0.248	(0.246)	9.44
$N = 950$, $d = 9$	20%	0.446	0.345	0.485	(0.462)	**0.321**	(**0.320**)	8.72
	50%	2.439	1.164	1.057	(1.049)	**0.693**	(0.745)	6.69
Statlog	5%	1.313	1.243	1.218	(1.224)	**1.211**	(1.219)	2.62
$N = 846$, $d = 18$	20%	1.751	1.582	1.379	(1.405)	**1.356**	(1.387)	2.60
	50%	3.711	2.960	1.859	(1.962)	**1.816**	(1.920)	2.11

		PDS	ICkNNI	Single Gaussian	
Auto-price	5%	0.978	0.981	0.985	(**0.970**)
$N = 159$, $d = 15$	20%	**1.182**	1.509	1.253	(1.216)
	50%	2.295	3.017	**2.048**	(2.082)
Breast Cancer (Diag.)	5%	2.483	2.439	**2.406**	(**2.406**)
$N = 569$, $d = 30$	20%	2.832	3.066	**2.485**	(2.489)
	50%	4.342	3.744	**2.837**	(2.884)
Breast Cancer (Prog.)	5%	3.453	3.408	**3.371**	(**3.372**)
$N = 194$, $d = 32$	20%	3.806	3.935	**3.491**	(3.500)
	50%	5.036	4.574	**3.903**	(3.947)
Breast Tissue	5%	0.865	0.792	**0.777**	(**0.776**)
$N = 106$, $d = 9$	20%	1.226	1.031	**0.897**	(0.912)
	50%	2.218	1.959	**1.309**	(1.401)
Ecoli	5%	1.088	0.773	**0.741**	(0.766)
$N = 336$, $d = 7$	20%	2.361	1.147	**1.038**	(1.129)
	50%	2.808	1.980	**1.669**	(1.886)
Financial Ratios	5%	3.263	3.108	3.074	(**3.070**)
$N = 500$, $d = 40$	20%	3.697	3.727	3.259	(**3.241**)
	50%	5.364	4.520	**3.880**	(**3.886**)
Ionosphere	5%	2.778	2.745	2.776	(**2.738**)
$N = 351$, $d = 33$	20%	2.985	3.291	2.954	(**2.885**)
	50%	3.938	3.827	3.383	(**3.332**)
Parkinsons	5%	1.637	1.615	**1.594**	(**1.594**)
$N = 195$, $d = 20$	20%	1.941	2.194	**1.773**	(**1.775**)
	50%	3.411	3.200	**2.301**	(2.353)
Spambase	5%	3.304	2.679	2.651	(**2.631**)
$N = 4601$, $d = 57$	20%	4.704	3.892	3.510	(**3.470**)
	50%	8.305	5.912	**5.547**	(**5.520**)
SPECTF Heart	5%	4.622	4.599	**4.567**	(**4.568**)
$N = 267$, $d = 44$	20%	4.940	4.953	**4.720**	(4.725)
	50%	6.103	5.640	**5.239**	(5.276)
Wine	5%	1.986	1.928	**1.919**	(1.925)
$N = 178$, $d = 13$	20%	2.472	2.284	**2.164**	(2.195)
	50%	3.990	3.364	**2.779**	(2.906)

		PDS	ICkNNI	HDDC mixture		Mean d_k
BCI	5%	6.118	6.258	**6.105**	(**6.105**)	23.31
$N = 400$, $d = 117$	20%	6.210	6.579	**6.144**	(**6.143**)	23.96
	50%	6.751	8.098	**6.365**	(**6.366**)	18.19
COIL	5%	5.098	5.224	5.101	(**5.095**)	26.23
$N = 1500$, $d = 241$	20%	**5.213**	5.713	5.229	(**5.210**)	26.10
	50%	**5.591**	7.990	5.738	(5.700)	21.23
Tecator	5%	0.592	1.167	**0.592**	(**0.592**)	4.00
$N = 240$, $d = 100$	20%	0.594	1.873	**0.592**	(**0.592**)	4.00
	50%	0.606	3.627	**0.592**	(**0.592**)	4.00

Table 4: Average relative error of estimated pairwise distances. The best result for each row is underlined, and any results which are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.05$) from the best result are bolded. The values in parenthesis represent the accuracy when the distances are calculated using the particular model for imputation only. The final column shows the mean number of Gaussian components K as selected by the AIC_C criterion, and the mean number of distinct eigenvalues d_k for HDDC.

		PDS	ICkNNI	Single Gaussian		Mixture model		Mean K
Computer Hardware	5%	0.141	**0.096**	0.227	(0.105)	0.142	(0.103)	3.82
$N = 209$, $d = 6$	20%	0.248	**0.180**	0.381	(**0.176**)	0.260	(0.188)	3.70
	50%	0.554	0.404	0.737	(**0.353**)	0.613	(0.413)	3.52
Glass Identification	5%	0.091	0.053	0.055	(**0.038**)	0.047	(**0.037**)	2.31
$N = 214$, $d = 9$	20%	0.180	0.130	0.170	(**0.102**)	0.129	(**0.101**)	2.51
	50%	0.439	0.369	0.458	(**0.261**)	0.367	(0.270)	2.45
Housing	5%	0.071	**0.036**	0.055	(0.040)	0.045	(0.037)	3.34
$N = 506$, $d = 13$	20%	0.158	0.103	0.122	(0.095)	0.107	(**0.090**)	3.26
	50%	0.387	0.298	0.245	(0.219)	0.237	(**0.213**)	3.21
Image Segmentation	5%	0.077	**0.030**	0.060	(0.036)	0.046	(0.034)	4.24
$N = 2310$, $d = 14$	20%	0.181	**0.095**	0.144	(**0.095**)	0.124	(**0.094**)	4.12
	50%	0.436	0.309	0.300	(**0.226**)	0.284	(**0.227**)	3.99
Iris	5%	0.141	0.087	0.114	(0.092)	0.093	(**0.080**)	3.00
$N = 150$, $d = 4$	20%	0.217	0.136	0.182	(0.140)	0.148	(**0.125**)	2.49
	50%	0.471	0.342	0.435	(0.322)	0.391	(**0.308**)	2.79
Pima Indians	5%	0.092	0.066	0.078	(0.065)	0.073	(**0.064**)	4.74
$N = 768$, $d = 8$	20%	0.174	0.123	0.136	(0.124)	0.129	(**0.120**)	4.47
	50%	0.383	0.258	0.249	(0.283)	**0.243**	(0.259)	4.27
Servo	5%	0.167	0.140	0.141	(**0.127**)	0.141	(**0.127**)	3.41
$N = 167$, $d = 4$	20%	0.250	0.190	0.193	(**0.180**)	0.197	(0.185)	3.86
	50%	0.392	0.327	**0.310**	(0.333)	0.330	(0.339)	4.14
Stocks	5%	0.066	**0.012**	0.054	(0.043)	0.017	(0.016)	9.44
$N = 950$, $d = 9$	20%	0.123	0.042	0.110	(0.087)	0.037	(**0.034**)	8.72
	50%	0.318	0.234	0.278	(0.220)	0.133	(**0.119**)	6.69
Statlog	5%	0.040	0.021	0.018	(0.017)	0.016	(**0.015**)	2.62
$N = 846$, $d = 18$	20%	0.093	0.074	0.046	(0.045)	0.041	(**0.041**)	2.60
	50%	0.231	0.273	0.109	(0.111)	**0.099**	(0.103)	2.11

		PDS	ICkNNI	Single Gaussian	
Auto-price	5%	**0.052**	0.054	0.089	(0.059)
$N = 159$, $d = 15$	20%	**0.117**	0.205	0.183	(0.130)
	50%	0.292	0.462	0.340	(**0.259**)
Breast Cancer (Diag.)	5%	0.033	0.021	0.011	(**0.010**)
$N = 569$, $d = 30$	20%	0.080	0.130	0.033	(**0.031**)
	50%	0.194	0.273	0.090	(**0.087**)
Breast Cancer (Prog.)	5%	0.032	0.025	0.013	(**0.012**)
$N = 194$, $d = 32$	20%	0.079	0.113	0.038	(**0.037**)
	50%	0.187	0.281	**0.090**	(0.091)
Breast Tissue	5%	0.085	0.045	0.046	(**0.036**)
$N = 106$, $d = 9$	20%	0.169	0.120	0.111	(**0.083**)
	50%	0.419	0.480	0.317	(**0.222**)
Ecoli	5%	0.119	**0.069**	0.114	(**0.068**)
$N = 336$, $d = 7$	20%	0.216	**0.132**	0.206	(**0.131**)
	50%	0.479	0.311	0.403	(**0.301**)
Financial Ratios	5%	0.039	0.025	0.018	(**0.013**)
$N = 500$, $d = 40$	20%	0.101	0.121	0.061	(**0.042**)
	50%	0.236	0.289	0.149	(**0.112**)
Ionosphere	5%	0.027	0.021	0.037	(**0.018**)
$N = 351$, $d = 33$	20%	0.064	0.181	0.103	(**0.048**)
	50%	0.156	0.321	0.204	(**0.109**)
Parkinsons	5%	0.044	0.028	0.023	(**0.021**)
$N = 195$, $d = 20$	20%	0.105	0.130	0.062	(**0.058**)
	50%	0.257	0.287	**0.135**	(0.138)
Spambase	5%	0.052	0.035	0.048	(**0.032**)
$N = 4601$, $d = 57$	20%	0.144	0.139	0.124	(**0.105**)
	50%	0.338	0.332	**0.226**	(0.259)
SPECTF Heart	5%	0.028	0.025	0.018	(**0.017**)
$N = 267$, $d = 44$	20%	0.066	0.103	0.048	(**0.044**)
	50%	0.158	0.275	0.107	(**0.101**)
Wine	5%	0.053	**0.037**	0.038	(**0.036**)
$N = 178$, $d = 13$	20%	0.114	0.096	**0.080**	(0.082)
	50%	0.283	0.249	**0.162**	(0.188)

		PDS	ICkNNI	HDDC mixture		Mean d_k
BCI	5%	0.015	0.030	**0.006**	(0.006)	23.31
$N = 400$, $d = 117$	20%	0.036	0.102	**0.015**	(0.016)	23.96
	50%	0.084	0.277	**0.037**	(0.042)	18.19
COIL	5%	0.014	0.028	**0.007**	(0.007)	26.23
$N = 1500$, $d = 241$	20%	0.033	0.108	**0.019**	(0.024)	26.10
	50%	0.076	0.288	**0.044**	(0.067)	21.23
Tecator	5%	0.005	0.236	0.0005	(**0.0004**)	4.00
$N = 240$, $d = 100$	20%	0.012	0.369	0.001	(**0.001**)	4.00
	50%	0.028	0.591	0.002	(**0.002**)	4.00

Table 5: Average normalised MSE of ELM predictions for regression tasks. The best result for each row is underlined, and any results which are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.05$) from the best result are bolded. The values in parenthesis represent the accuracy when the distances are calculated using the particular model for imputation only. The final column shows the mean number of Gaussian components K as selected by the AIC_C criterion, and the mean number of distinct eigenvalues d_k for HDDC.

		PDS	ICkNNI	Single Gaussian		Mixture model		Mean K
Computer Hardware	5%	0.382	**0.351**	0.354	(**0.351**)	0.353	(0.353)	3.82
$N = 209$, $d = 6$	20%	0.427	0.383	0.385	(0.381)	**0.375**	(0.382)	3.70
	50%	0.558	0.540	0.491	(0.531)	**0.486**	(0.532)	3.52
Housing	5%	0.242	0.199	0.199	(0.199)	**0.198**	(**0.198**)	3.34
$N = 506$, $d = 13$	20%	0.342	0.279	**0.255**	(0.256)	**0.255**	(0.258)	3.26
	50%	0.567	0.593	**0.419**	(0.446)	0.433	(0.461)	3.21
Servo	5%	0.444	0.359	0.358	(**0.350**)	0.360	(0.353)	3.41
$N = 167$, $d = 4$	20%	0.644	0.532	0.533	(**0.522**)	0.540	(0.539)	3.86
	50%	0.798	0.816	**0.772**	(0.799)	0.793	(0.837)	4.14
Stocks	5%	0.039	**0.019**	0.028	(0.027)	0.020	(0.020)	9.44
$N = 950$, $d = 9$	20%	0.085	0.028	0.062	(0.059)	**0.026**	(**0.026**)	8.72
	50%	0.302	0.226	0.230	(0.218)	**0.101**	(0.107)	6.69

		PDS	ICkNNI	Single Gaussian	
Auto-price	5%	0.253	0.223	**0.221**	(0.221)
$N = 159$, $d = 15$	20%	0.315	0.245	**0.224**	(**0.224**)
	50%	0.438	0.483	0.265	(**0.262**)
Breast Cancer (Prog.)	5%	0.949	0.940	**0.939**	(0.940)
$N = 194$, $d = 32$	20%	0.974	0.943	**0.939**	(0.944)
	50%	1.028	0.975	**0.944**	(0.961)

		PDS	ICkNNI	HDDC mixture		Mean d_k
Tecator	5%	**0.285**	0.364	**0.290**	(**0.290**)	4.00
$N = 240$, $d = 100$	20%	**0.348**	0.471	**0.331**	(0.333)	4.00
	50%	0.519	0.788	**0.427**	(0.520)	4.00

Table 6: Average ELM correct classification rate for classification tasks. The best result for each row is underlined, and any results which are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.05$) from the best result are bolded. The values in parenthesis represent the accuracy when the distances are calculated using the particular model for imputation only. The final column shows the mean number of Gaussian components K as selected by the AIC_C criterion, and the mean number of distinct eigenvalues d_k for HDDC.

		PDS	ICkNNI	Single Gaussian		Mixture model		Mean K
Glass Identification $N = 214$, $d = 9$	5%	0.640	0.655	**0.659**	(**0.659**)	**0.659**	(**0.659**)	2.31
	20%	0.589	0.630	0.641	(0.645)	0.644	(**0.646**)	2.51
	50%	0.494	0.493	0.551	(0.562)	0.561	(**0.569**)	2.45
Image Segmentation $N = 2310$, $d = 14$	5%	0.912	**0.942**	0.926	(0.933)	0.935	(0.938)	4.24
	20%	0.856	0.890	0.858	(0.883)	0.881	(**0.897**)	4.12
	50%	0.715	0.673	0.731	(0.761)	0.758	(**0.784**)	3.99
Iris $N = 150$, $d = 4$	5%	0.946	0.951	0.953	(0.953)	**0.953**	(**0.953**)	3.00
	20%	0.913	0.933	0.938	(0.936)	**0.941**	(0.940)	2.49
	50%	0.806	0.817	0.839	(0.832)	**0.846**	(0.837)	2.79
Pima Indians $N = 768$, $d = 8$	5%	0.747	0.747	**0.752**	(0.748)	0.751	(0.748)	4.74
	20%	0.729	0.731	**0.743**	(0.733)	0.742	(0.733)	4.47
	50%	0.686	0.686	**0.710**	(0.693)	0.706	(0.690)	4.27
Statlog $N = 846$, $d = 18$	5%	0.750	0.779	0.780	(0.783)	0.783	(**0.784**)	2.62
	20%	0.696	0.722	0.750	(0.758)	0.759	(**0.766**)	2.60
	50%	0.575	0.504	0.666	(0.676)	0.687	(**0.694**)	2.11

		PDS	ICkNNI	Single Gaussian	
Breast Cancer (Diag.) $N = 569$, $d = 30$	5%	0.963	0.967	**0.968**	(0.968)
	20%	0.954	0.946	**0.967**	(0.967)
	50%	0.934	0.926	0.960	(**0.960**)
Breast Tissue $N = 106$, $d = 9$	5%	0.579	**0.586**	0.585	(**0.586**)
	20%	0.548	0.566	0.578	(**0.584**)
	50%	0.447	0.420	0.522	(**0.534**)
Ecoli $N = 336$, $d = 7$	5%	0.921	**0.927**	0.926	(0.926)
	20%	0.888	**0.896**	**0.896**	(0.895)
	50%	0.763	0.758	**0.781**	(0.776)
Financial Ratios $N = 500$, $d = 40$	5%	0.901	0.910	**0.911**	(0.911)
	20%	0.888	0.887	**0.910**	(**0.910**)
	50%	0.859	0.880	**0.903**	(0.902)
Ionosphere $N = 351$, $d = 33$	5%	0.944	0.949	**0.949**	(0.949)
	20%	0.930	0.906	**0.944**	(**0.944**)
	50%	0.877	0.865	0.920	(**0.922**)
Parkinsons $N = 195$, $d = 20$	5%	0.845	0.852	**0.853**	(**0.853**)
	20%	0.828	0.831	**0.852**	(**0.852**)
	50%	0.800	0.797	**0.835**	(**0.835**)
Spambase $N = 4601$, $d = 57$	5%	0.936	0.940	**0.941**	(**0.941**)
	20%	0.919	0.922	0.929	(**0.931**)
	50%	0.880	0.891	0.896	(**0.897**)
SPECTF Heart $N = 267$, $d = 44$	5%	0.799	0.801	**0.802**	(0.802)
	20%	0.792	0.794	**0.799**	(0.799)
	50%	0.782	0.789	**0.796**	(0.793)
Wine $N = 178$, $d = 13$	5%	0.971	0.974	**0.974**	(0.974)
	20%	0.949	0.957	**0.965**	(0.963)
	50%	0.850	0.791	**0.908**	(0.905)

		PDS	ICkNNI	HDDC mixture		Mean d_k
BCI $N = 400$, $d = 117$	5%	0.641	0.641	0.651	(**0.651**)	23.31
	20%	0.608	0.621	0.646	(**0.648**)	23.96
	50%	0.558	0.573	0.632	(**0.636**)	18.19
COIL $N = 1500$, $d = 241$	5%	0.926	0.927	0.934	(**0.935**)	26.23
	20%	0.898	0.891	0.929	(**0.934**)	26.10
	50%	0.799	0.748	0.912	(**0.926**)	21.23

[1] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[2] R. J. A. Little, D. B. Rubin, Statistical Analysis with Missing Data, Wiley-Interscience, second edition, 2002.

[3] D. B. Rubin, Multiple Imputation for Nonresponse in Surveys, Wiley Series in Probability and Statistics, Wiley, 1987.

[4] C. K. Enders, Applied Missing Data Analysis, Methodology In The Social Sciences, Guilford Press, 2010.

[5] E. R. Hruschka, E. R. Hruschka Jr., N. F. F. Ebecken, Evaluating a nearest-neighbor method to substitute continuous missing values, in: AI 2003: Advances in Artificial Intelligence, volume 2903 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2003, pp. 723–734.

[6] J. Van Hulse, T. M. Khoshgoftaar, Incomplete-case nearest neighbor imputation in software measurement data, Information Sciences (2011). In press, available online 9 January 2011, DOI: 10.1016/j.ins.2010.12.017.

[7] S. Van Buuren, J. P. Brand, C. G. Groothuis-Oudshoorn, D. B. Rubin, Fully conditional specification in multivariate imputation, Journal of Statistical Computation and Simulation 76 (2006) 1049–1064.

[8] A. Aussem, S. R. de Morais, A conservative feature subset selection algorithm with missing data, Neurocomputing 73 (2010) 585–590.

[9] G. Doquire, M. Verleysen, Feature selection with missing data using mutual information estimators, Neurocomputing 90 (2012) 3–11.

[10] J. W. Grzymala-Busse, W. J. Grzymala-Busse, Handling missing attribute values, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer US, second edition, 2010, pp. 33–51.

[11] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, M. Verleysen, K nearest neighbours with mutual information for simultaneous classification and missing data imputation, Neurocomputing 72 (2009) 1483–1493.

[12] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1977) pp. 1–38.

[13] Z. Ghahramani, M. Jordan, Learning From Incomplete Data, Technical Report, Lab Memo No. 1509, CBCL Paper No. 108, MIT AI Lab, 1995.

[14] L. Hunt, M. Jorgensen, Mixture model clustering for mixed data with missing information, Computational Statistics & Data Analysis 41 (2003) 429–440.

[15] T. I. Lin, J. C. Lee, H. J. Ho, On fast supervised learning for normal mixture models with missing information, Pattern Recognition 39 (2006) 1177–1187.

[16] R. J. Steele, N. Wang, A. E. Raftery, Inference from multiple imputation for missing data using mixtures of normals, Statistical Methodology 7 (2010) 351–365.

[17] O. Delalleau, A. C. Courville, Y. Bengio, Efficient EM training of Gaussian mixtures with missing data, CoRR abs/1209.0521 (2012).

[18] V. Tresp, S. Ahmad, R. Neuneier, Training neural networks with deficient data, in: Advances in Neural Information Processing Systems 6, Morgan Kaufmann, 1994, pp. 128–135.

[19] Z. Viharos, L. Monostori, T. Vincze, Training and application of artificial neural networks with incomplete data, in: T. Hendtlass, M. Ali (Eds.), Developments in Applied Artificial Intelligence, volume 2358 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2002, pp. 649–659.

[20] A. Morris, M. Cooke, P. Green, Some solutions to the missing feature problem in data classification, with application to noise robust ASR, in: Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, volume 2, pp. 737–740.

[21] C. Beunckens, G. Molenberghs, M. G. Kenward, Direct likelihood analysis versus simple forms of imputation for missing data in randomized clinical trials, Clinical Trials 2 (2005) 379–386.

[22] C. Bouveyron, S. Girard, C. Schmid, High-dimensional data clustering, Computational Statistics & Data Analysis 52 (2007) 502–519.

[23] J. K. Dixon, Pattern recognition with partly missing data, Systems, Man and Cybernetics, IEEE Transactions on 9 (1979) 617–621.

[24] T. W. Anderson, An Introduction to Multivariate Statistical Analysis, Wiley-Interscience, New York, third edition, 2003.

[25] R. B. Cattell, The scree test for the number of factors, Multivariate Behavioral Research 1 (1966) 245–276.

[26] H. Akaike, A new look at the statistical model identification, Automatic Control, IEEE Transactions on 19 (1974) 716–723.

[27] C. M. Hurvich, C.-L. Tsai, Regression and time series model selection in small samples, Biometrika 76 (1989) 297–307.

[28] A. Asuncion, D. J. Newman, UCI machine learning repository, 2012. University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml/.

[29] L. Torgo, Regression datasets, 2012. University of Porto, http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html.

[30] O. Chapelle, B. Schölkopf, A. Zien (Eds.), Semi-Supervised Learning, MIT Press, Cambridge, MA, 2006.

[31] J. Shaffer, Multiple hypothesis testing, Annual review of psychology 46 (1995) 561–584.

[32] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (2006) 489–501.

[33] G.-B. Huang, C.-K. Siew, Extreme learning machine with randomly assigned RBF kernels, International Journal of Information Technology 11 (2005) 16–24.