Information Sciences 240 (2013) 115–128

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Distance estimation in numerical data sets with missing values

Emil Eirola [a,*], Gauthier Doquire [b], Michel Verleysen [b], Amaury Lendasse [a,c,d]

[a] Department of Information and Computer Science, Aalto University, FI-00076 Aalto, Finland
[b] Machine Learning Group—ICTEAM, Université catholique de Louvain, Place du Levant 3, 1348 Louvain-la-Neuve, Belgium
[c] IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain
[d] Computational Intelligence Group, Computer Science Faculty, University of the Basque Country, Paseo Manuel Lardizabal 1, Donostia/San Sebastián, Spain

ARTICLE INFO

Article history:
Received 11 May 2012
Received in revised form 13 February 2013
Accepted 25 March 2013
Available online 3 April 2013

Keywords:
Missing data
Distance estimation
Imputation
Nearest neighbour

ABSTRACT

The possibility of missing or incomplete data is often ignored when describing statistical or machine learning methods, but as it is a common problem in practice, it is relevant to consider. A popular strategy is to fill in the missing values by imputation as a pre-processing step, but for many methods this is not necessary, and can yield sub-optimal results. Instead, appropriately estimating pairwise distances in a data set directly enables the use of any machine learning methods using nearest neighbours or otherwise based on distances between samples. In this paper, it is shown how directly estimating distances tends to result in more accurate results than calculating distances from an imputed data set, and an algorithm to calculate the estimated distances is presented. The theoretical framework operates under the assumption of a multivariate normal distribution, but the algorithm is shown to be robust to violations of this assumption. The focus is on numerical data with a considerable proportion of missing values, and simulated experiments are provided to show accurate performance on several data sets.

1. Introduction

In many real world machine learning tasks, data sets with missing values (also referred to as incomplete data) are all too common to be easily ignored. 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.

If the fraction of missing data is sufficiently small, a common pre-processing step is to perform imputation to fill in the missing values and proceed with conventional methods for further processing. Any errors introduced by inaccurate imputation may be considered insignificant in terms of the entire processing chain. With a larger proportion of measurements being missing, errors caused by the imputation are increasingly relevant, and the missing value imputation cannot be considered a separate step. Instead, the task should be seen from a holistic perspective, and the statistical properties of the missing data should be considered more carefully. In the current scenario, the interest lies in modelling data sets with a

* Corresponding author. Tel.: +358504302878.
E-mail addresses: emil.eirola@aalto.fi (E. Eirola), gauthier.doquire@uclouvain.be (G. Doquire), michel.verleysen@uclouvain.be (M. Verleysen), amaury.lendasse@aalto.fi (A. Lendasse).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.03.043

considerable fraction of missing data, and one cannot afford to discard the incomplete samples. The focus is on cases where there is such a significant amount of data missing compared to the data available, that it is not conceivable to fully estimate the distribution of the data. Instead, the only statistics we can hope to accurately estimate are the mean and covariance.

In this paper, the particular problem of estimating distances between samples in a numerical data set is studied. Assuming that data is Missing-at-Random (MAR) [20] – i.e., the probability of a particular measurement being missing is independent of the value it would have taken – and that the samples originate from some probability distribution, statistical techniques are applied to find an expression for the expectation of the squared Euclidean distance between samples. A specific algorithm is presented to calculate such estimates of all the pairwise distances in a data set with missing values. The theoretical framework operates under the assumption of a multivariate normal distribution, but the algorithm is shown to be robust to violations of the assumptions concerning the distribution of data.

Being able to appropriately estimate distances between samples, or between samples and prototypes, in a data set with missing values has numerous applications. It directly enables the use of several standard statistical and machine learning methods which are based only on distances and not the direct values, e.g.: nearest neighbours (k-NN) [28], multidimensional scaling (MDS) [9], support vector machines (SVM) [6], or radial basis function (RBF) neural networks [3]. The algorithm presented in this paper, when used as a pre-processing step, directly allows the user to apply such methods to their data, without having to separately tweak the methods to explicitly handle cases with missing data. While there are methods to fill in missing values, directly estimating the distance matrix from the data is more reliable than calculating distances from the imputed data since the uncertainty of the missing data can be accounted for, as evidenced by the derivation and experiments in this paper.

The sequel of this paper is structured so that Section 2 reviews related approaches for dealing with missing data. The justification and description of the proposed algorithm is presented in Section 3, while Section 4 contains the experimental results of simulations comparing the algorithm to alternative methods.

2. Related work

Data sets with missing data have been extensively studied from the perspectives of machine learning and statistics – see, e.g., [20] for an overview, or [13] for an analysis on the effect of imputation on classification accuracy. Generally learning with numerical data and missing attribute values has focused on filling in the missing values. A simple method of imputation by searching for the nearest neighbor among only the fully known patterns can be effective when only a few values are missing [16,17], but is ineffective when a majority of the data samples have missing components. An improved approach is incomplete-case k-NN imputation (ICkNNI) [29], which searches for neighbours among all patterns for which a superset of the known components of the query point are known. This still fails in high-dimensional cases, or with a sufficiently large proportion of missing data. A more intricate method where multiple nearest neighbours are considered, and a model is separately learned for each incomplete sample, is presented in [30]. Another variation is to restrict the search to certain samples or attributes according to specified rules, as in the "concept closest fit" [14] and "rough sets fit" [19] methods. One possibility for integrating the imputation of missing values with learning a prediction model is presented in the MLEM2 rule induction algorithm [14].

The problem of directly estimating pairwise distances is, however, less studied. Previous approaches for estimating distances in data sets with missing values generally involve imputing the missing data with some estimates, and calculating distances from the imputed data. This technique severely underestimates the uncertainty of the imputed values. Estimating the distances directly leads to more reliable estimates as the uncertainty can also be considered.

A simple and widely used method for estimating distance with missing values is the Partial Distance Strategy (PDS) [10,15]. In the PDS, an estimate for the squared distance is found by calculating the sum of squared differences of the mutually known components, and scaling the value proportionally to account for the missing values. This approach has a tendency to underestimate distances, as it ignores the general variability of the data, and only takes into account the locally known components. Also, if two samples have no common components, the output of this strategy is undefined. The PDS has been used for nearest neighbour search in order to estimate mutual information [11].

The multiple imputation [24] paradigm has been proposed as another solution to naturally account for the uncertainty. It still requires that some model is fit to the data, so that the imputation can be generated from the posterior distribution and is thus non-trivial to conduct appropriately in practice [12,25].

In a specific case of an entropy-based distance measure [5], the authors propose that the distance to an incomplete sample can be estimated as the mean distance after the missing value is replaced by random draws. However, the missing value is successively replaced by the corresponding attribute from every specified sample, ignoring any dependence to the observed attributes of the incomplete sample.

Finding distances from each sample to some prototype patterns (where the prototypes have no missing values) has been conducted by ignoring those components which are missing for the query pattern. Such distances from the same query point to different prototypes are comparable, and this strategy has, for instance, been used successfully with self-organising maps (SOM) [7]. However, if a prototype has a very extreme value for a component which the query point is missing, the distance will be underestimated.

3. Theory and description

An important consideration when dealing with missing data is the missingness mechanism. We will assume that a missing value represents a value which is defined and exists, but for an unspecified reason is not known. Following the conventions of Little and Rubin [20], the assumption here is that data is Missing-at-Random (MAR):

$$P(M|x_{\text{obs}}, x_{\text{mis}}) = P(M|x_{\text{obs}}) \tag{1}$$

i.e., the event of a measurement being missing is independent from the value it would take, conditional on the observed data. The stronger assumption of Missing-Completely-at-Random (MCAR) is not necessary, as MAR is an ignorable missingness mechanism in the sense that, for instance, maximum likelihood estimation still provides a consistent estimator [20].

3.1. The expected squared distance

In the following, we consider data vectors $\boldsymbol{x}_i,\ \boldsymbol{x}_j \in \mathbb{R}^d$ with components denoted by $x_{i,l}$, $x_{j,l}$ for $1 \leqslant l \leqslant d$, and focus on calculating the expectation of the *squared* Euclidean (ℓ^2) distance between them:

$$d(\boldsymbol{x}_i,\ \boldsymbol{x}_j)^2_{\ell^2} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \sum_{l=1}^{d} (x_{i,l} - x_{j,l})^2 \tag{2}$$

Estimating the ℓ^2-norm itself could be feasible, but due to the square-root, the expressions do not simplify and separate as cleanly, meaning that any computer implementation would be considerably more computationally demanding. 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 samples $\boldsymbol{x}_i \in \mathbb{R}^d$ which may contain missing values, denote by $M_i \subseteq \{1, \ldots, d\}$ the set of indices of the missing 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 according to which attributes are missing for the two samples:

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \sum_{l \notin M_i \cup M_j} (x_{i,l} - x_{j,l})^2 + \sum_{l \in M_j \setminus M_i} (x_{i,l} - x_{j,l})^2 + \sum_{l \in M_i \setminus M_j} (x_{i,l} - x_{j,l})^2 + \sum_{l \in M_i \cap M_j} (x_{i,l} - x_{j,l})^2 \tag{3}$$

The first term here ($l \notin M_i \cup M_j$) is a sum over those components which are known for both samples, and hence this part of the sum can be calculated directly. The second term includes those components which are missing in $\boldsymbol{x}_j$, but *not* in $\boldsymbol{x}_i$. Correspondingly, the third term covers those attributes which are missing for $\boldsymbol{x}_i$ but not $\boldsymbol{x}_j$. Any components missing in *both* $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are in the final summation. Note that any of the sums may be empty, depending on the pattern of missing values. Now the missing values can be modelled as random variables, $X_{i,l}$, $l \in M_i$. Taking the expected value of the above expression with respect to these random variables, and using the linearity of expectation, the expression can be separated further:

$$\begin{aligned}
\mathrm{E}\left[\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right] &= \sum_{l \notin M_i \cup M_j} (x_{i,l} - x_{j,l})^2 + \sum_{l \in M_j \setminus M_i} \mathrm{E}\left[(x_{i,l} - X_{j,l})^2\right] + \sum_{l \in M_i \setminus M_j} \mathrm{E}\left[(X_{i,l} - x_{j,l})^2\right] + \sum_{l \in M_i \cap M_j} \mathrm{E}\left[(X_{i,l} - X_{j,l})^2\right] \\
&= \sum_{l \notin M_i \cup M_j} (x_{i,l} - x_{j,l})^2 + \sum_{l \in M_j \setminus M_i} \left((x_{i,l} - \mathrm{E}[X_{j,l}])^2 + \mathrm{Var}[X_{j,l}]\right) + \sum_{l \in M_i \setminus M_j} \left((\mathrm{E}[X_{i,l}] - x_{j,l})^2 + \mathrm{Var}[X_{i,l}]\right) \\
&\quad + \sum_{l \in M_i \cap M_j} \left((\mathrm{E}[X_{i,l}] - \mathrm{E}[X_{j,l}])^2 + \mathrm{Var}[X_{i,l}] + \mathrm{Var}[X_{j,l}]\right)
\end{aligned}$$

To illustrate, we show the expansion of the second summation ($l \in M_j \setminus M_i$):

$$\begin{aligned}
\mathrm{E}[(x_{i,l} - X_{j,l})^2] &= \mathrm{E}\left[x_{i,l}^2 - 2x_{i,l}X_{j,l} + X_{j,l}^2\right] = x_{i,l}^2 - 2x_{i,l}\mathrm{E}[X_{j,l}] + \mathrm{E}[X_{j,l}^2] = x_{i,l}^2 - 2x_{i,l}\mathrm{E}[X_{j,l}] + \mathrm{E}[X_{j,l}]^2 - \mathrm{E}[X_{j,l}]^2 + \mathrm{E}[X_{j,l}^2] \\
&= (x_{i,l} - \mathrm{E}[X_{j,l}])^2 + \mathrm{E}\left[X_{j,l}^2 - \mathrm{E}[X_{j,l}]^2\right] = (x_{i,l} - \mathrm{E}[X_{j,l}])^2 + \mathrm{Var}[X_{j,l}]
\end{aligned}$$

The remaining cases are analogous, while in the final case, it is necessary to consider $X_{i,l}$ and $X_{j,l}$ to be uncorrelated, given the known values of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

It thus suffices to find the expectation and variance of each random variable separately, and it is not necessary to determine the full probability density.

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. By this argument, 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 $\boldsymbol{x}'_i$ to be an imputed version of $\boldsymbol{x}_i$ where each missing value has been replaced by its conditional mean.

$$x'_{i,l} = \begin{cases} \mathrm{E}[X_{i,l}|x_{i,\text{obs}}] & \text{if } l \in M_i, \\ x_{i,l} & \text{otherwise} \end{cases} \tag{4}$$

Define $\sigma_{i,l}^2$ correspondingly as the conditional variance

$$\sigma_{i,l}^2 = \begin{cases} \mathrm{Var}[X_{i,l}|x_{i,\mathrm{obs}}] & \text{if } l \in M_i, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

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

$$\mathrm{E}\left[\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right] = \sum_{l=1}^{d}\left(\left(x'_{i,l} - x'_{j,l}\right)^2 + \sigma_{i,l}^2 + \sigma_{j,l}^2\right) \tag{6}$$

or, equivalently,

$$\mathrm{E}\left[\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right] = \|\boldsymbol{x}'_i - \boldsymbol{x}'_j\|^2 + s_i^2 + s_j^2, \quad \text{where} \quad s_i^2 = \sum_{l\in M_i}\sigma_{i,l}^2 \tag{7}$$

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.

3.2. The conditional mean and variance

Lacking specific knowledge about the distributions of the random variables denoting the missing components, the reasonable approach is to derive statistical estimates based on the available data. Assuming the samples are part of a collection of data, it is often useful to proceed from the assumption that the samples are drawn – i.i.d. – from some multivariate probability distribution. Estimating the distribution enables the calculation of the required conditional means and variances.

For a general distribution, finding the conditional mean and variance requires estimating the joint probability distribution, which is not feasible to conduct with any kind of accuracy in a high-dimensional space when the number of samples is limited, particularly when there is missing data.

To enable the following calculation, we consider the data to originate from a parametric distribution: the multivariate normal distribution. The normal distribution is ubiquitous, and can be used as a crude approximation for nearly any continuous distribution. By matching first and second moments, it can appreciably fit most distributions that are encountered in practice. The multivariate normal distribution maximises the differential entropy for a given variance–covariance structure [8, Thm. 8.6.5], and is hence a natural choice to model an unknown distribution in accordance with the maximum entropy principle, as it maximises the uncertainty about the missing data. In other words, it minimises any assumptions of additional structure in the distribution. The distribution is fully defined by the mean and covariance matrix, so estimating these quantities is sufficient.

The conditional means and variances are straightforward to calculate in the case of the multivariate Gaussian. If the components of a multivariate normal random variable X are partitioned to $X^{(1)}$ (the missing values) and $X^{(2)}$ (the known values), and the mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ are similarly divided into $\boldsymbol{\mu}^{(1)}$, $\boldsymbol{\mu}^{(2)}$, $\boldsymbol{\Sigma}_{11}$, $\boldsymbol{\Sigma}_{12}$, $\boldsymbol{\Sigma}_{21}$, and $\boldsymbol{\Sigma}_{22}$:

$$X = \begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix}, \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}^{(1)} \\ \boldsymbol{\mu}^{(2)} \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}$$

then the conditional distribution of $X^{(1)}$ given $X^{(2)} = \boldsymbol{x}^{(2)}$ is normally distributed with mean

$$\boldsymbol{\mu}'_1 = \boldsymbol{\mu}^{(1)} + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}^{(2)} - \boldsymbol{\mu}^{(2)}) \tag{8}$$

and covariance matrix

$$\boldsymbol{\Sigma}'_{11} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \tag{9}$$

as shown in [1, Thm. 2.5.1]. The conditional means and variances of each missing value are then found by extracting the appropriate element from $\boldsymbol{\mu}'_1$ or the diagonal of $\boldsymbol{\Sigma}'_{11}$.

In the current scenario, it is not necessary to consider the full conditional joint distribution of the missing values, as only the mean and covariance are required. The theorem above specifically deals with Gaussian distributions. However, the conditional mean and covariance matrix given by Eqs. (8) and (9) are accurate for a somewhat larger class of distributions. In particular, the equations clearly hold exactly whenever all the variables are all mutually independent since the covariance matrix is diagonal – regardless of the distribution, as long as the variance of each variable is positive and finite. By extension, the equations also hold exactly if one set of variables conform to a multivariate Gaussian distribution, and the remaining variables are mutually independent, and independent of the variables in the Gaussian distribution. Furthermore, the equations appear to provide good approximations of the conditional mean and covariance for an even greater class of distributions. Hence, using these expressions as estimates of the conditional mean and covariance is justified and effective even if the data at hand is decidedly non-Gaussian. The accuracy achieved with this strategy in the simulations in Section 4 further supports the use of this procedure.

If the data does not follow a Gaussian distribution, the estimated expectations may not be accurate. In such cases, matching the mean and covariance will tend to lead to a Gaussian distribution which covers too large areas of the input space. The effect of this is that the conditional variance terms for any missing vales will tend to be too large, and that distances with high uncertainty (between samples with many missing values) will be overestimated rather than underestimated. In the context of a nearest neighbour search, this leads to a situation where errors are skewed towards minimising the number of false positives. This can be a desired effect of the estimation procedure, as small distances tend to be more important in practical situations, particularly in pattern recognition where the focus is on samples with high similarity. Often we are interested in finding samples which are most similar to other samples, and then false positives are a bigger problem than false negatives. Hence we feel that it is safer to overestimate distances, if we cannot be accurate. Overall, the method is naturally the most accurate for data which resemble a multivariate normal distribution, but is still reasonably safe for any continuous distribution (in the sense of low false positives when identifying small distances).

3.3. Estimating the covariance matrix

Estimating the covariance matrix from a data set with missing values is non-trivial. The two basic approaches [20] are generally insufficient for the current purpose:

Available-case analysis (pairwise estimation of the covariances between variables) is not appropriate because even if the individual covariances are rather accurate, the covariance matrix as a whole is not estimated consistently. In particular, the resulting matrix is often not positive definite. When solving a linear system using such a matrix, errors are amplified and the behaviour is not as expected. Estimating pairwise correlations instead of covariances, and rescaling by the individual variance of each variable, can in some cases lead to a better estimate for the covariance matrix, but does not completely avoid the problem.

Complete-case analysis (ignoring all samples with missing values) does provide a consistent estimate of the matrix as long as there are enough samples with no missing data. However, the quality of the estimate deteriorates rapidly with an increasing proportion of missing data.

On the other hand, *maximum likelihood (ML)*-estimation can provide accurate estimates of the covariance matrix, usable even for more than 50% of missing data. In [20], it is described how the EM algorithm can be used to obtain the estimate. For the experiments in the next section, we choose to use a standard referenced implementation. The ECM (expectation conditional maximisation) method is applied as provided in the MATLAB Financial Toolbox [21], which implements the method of Meng and Rubin [22] with some improvements by Sexton and Swensen [27]. Although the maximum likelihood framework is based on a model of normally distributed data, non-normal data has been found to have a negligible impact on the accuracy of the estimated parameters [12].

3.4. Proposed algorithm

Based on the previous arguments, we propose an algorithm to calculate the Expected Squared Distances (ESD) pairwise between samples.

Input: A data set $\{\boldsymbol{x}_i\}_{i=1}^N$ of N samples in $\mathbb{R}^d$, of which M samples contain missing values.

1. Estimate the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of the data set with the ECM algorithm.
2. For each sample $\boldsymbol{x}_i$ with missing values, do
3. Find the conditional mean by $\boldsymbol{\mu}_1' = \boldsymbol{\mu}^{(1)} + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\boldsymbol{x}_i^{(2)} - \boldsymbol{\mu}^{(2)}\right)$
4. Create $\boldsymbol{x}_i'$ from $\boldsymbol{x}_i$ by replacing the missing values by values from $\boldsymbol{\mu}_1'$
5. Find the conditional covariance matrix by $\boldsymbol{\Sigma}_{11}' = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$
6. Calculate the sum of the diagonal of $\boldsymbol{\Sigma}_{11}'$ and set $s_i^2 = \sum_{l=1}^{|M_i|}(\boldsymbol{\Sigma}_{11}')_{ll}$
7. For each pair of samples $\boldsymbol{x}_i$, $\boldsymbol{x}_j$, do
8. Find the squared distance between $\boldsymbol{x}_i'$ and $\boldsymbol{x}_j'$ as $P_{ij} = \sum_{l=1}^{d}\left(x_{i,l}' - x_{j,l}'\right)^2$
9. Add the sum of the conditional variances of the missing values, $P_{ij} \leftarrow P_{ij} + s_i^2 + s_j^2$

Output: The matrix $\mathbf{P}$ with elements P_{ij} of estimates of the pairwise expected squared distances.

The computational complexity of the first loop is at most $\mathcal{O}(Md^3)$, depending on the particular way of handling the inverse of $\boldsymbol{\Sigma}_{22}$. The complexity of the second loop is $\mathcal{O}(N^2d)$, equivalent to finding the pairwise distances in a data set with no missing values.

As the algorithm calculates the expected mean and variance of each missing value, as a side-effect the user obtains an imputed version of the data set, which can be useful in some applications, for instance for initialising prototype patterns in certain machine learning methods.

The algorithm trivially extends to some other scenarios, such as finding the distances from a set of samples to a set of prototypes.

3.5. Special case: independent variables

If the components are known to be independent (or can be assumed to be), the situation becomes much simpler as the conditional means and variances of the missing data do not depend on the observed components of the samples. Hence it is enough to separately estimate the mean and variance of each variable, and form $x'_{i,l}$ and $\sigma^2_{i,l}$ using those:

$$x'_{i,l} = \begin{cases} \mathrm{E}[X_l] & \text{if } l \in M_i, \\ x_{i,l} & \text{otherwise} \end{cases}$$

and

$$\sigma^2_{i,l} = \begin{cases} \mathrm{Var}[X_l] & \text{if } l \in M_i, \\ 0 & \text{otherwise} \end{cases}$$

and apply Eq. (7). Any other assumptions about the distributions of the variables are not necessary. No matrix inversion is required, and as a result, this alternative method is computationally lighter and significantly faster, although generally inaccurate in any interesting cases.

3.6. Extension to weighted distances

The procedure can be extended to any weighted Euclidean distance, such as the Mahalonobis distance, or any such distance weighted by a positive definite matrix S^{-1}. First, find the Cholesky decomposition $S^{-1} = LL^T$. Then:

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_S^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T S^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j) = \|L^T \boldsymbol{x}_i - L^T \boldsymbol{x}_j\|^2 = \sum_{l=1}^{d} \left(L_l^T \boldsymbol{x}_i - L_l^T \boldsymbol{x}_j \right)^2$$

where L_l is the lth column of L. Then

$$\mathrm{E}\left[\|L^T \boldsymbol{x}_i - L^T \boldsymbol{x}_j\|^2 \right] = \|L^T \boldsymbol{x}'_i - L^T \boldsymbol{x}'_j\|^2 + \sum_{l=1}^{d} \mathrm{Var}\left(L_l^T \boldsymbol{x}_i \right) + \sum_{l=1}^{d} \mathrm{Var}\left(L_l^T \boldsymbol{x}_j \right)$$

Now, using the conditional covariance matrix $\boldsymbol{\Sigma}'_{11}$ corresponding to the sample $\boldsymbol{x}_i$,

$$\mathrm{Var}(L_l^T \boldsymbol{x}_i) = \mathrm{Var}\left[\sum_{j=1}^{d} L_{jl} x_{i,j} \right] = \sum_{j \in M_i} \sum_{k \in M_i} L_{jl} L_{kl} \mathrm{Cov}(X_{i,j}, X_{i,k}) = L_l'^T \boldsymbol{\Sigma}'_{11} L'_l = \left(L'^T \boldsymbol{\Sigma}'_{11} L' \right)_{ll}$$

Here the conditional covariance matrix from Eq. (9) is used, and L' is a matrix formed from L by retaining only those rows corresponding to indices in M_i and L'_l is the lth column of L'.

Hence for the Mahalonobis case, the expected squared distance can be written as

$$\mathrm{E}\left[\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_S^2 \right] = \|\boldsymbol{x}'_i - \boldsymbol{x}'_j\|_S^2 + s_i^2 + s_j^2 \quad \text{where} \quad s_i^2 = \sum_{l=1}^{d} \left(L'^T \boldsymbol{\Sigma}'_{11} L' \right)_{ll} \tag{10}$$

3.7. Using the estimated distances to form a kernel matrix

A common use of pairwise distances is for kernel methods, which can be formulated in terms of a suitable kernel matrix representing the inner products between the samples in an unspecified higher-dimensional space. This is known as the *kernel trick*, as the projection to the higher space does not need to be formulated explicitly. Several of these kernel matrices can be formulated in terms of only the distances between samples, in particular the Gaussian radial basis function $K(x,y) = \exp(-\|x-y\|^2)$, which is one the most popular choices for a kernel function. These kernel methods include many well known algorithms, such as support vector machines [6], Gaussian process [4], radial basis function neural networks [3], kernel principal component analysis [26], kernel Fisher discriminant analysis [23], and kernel canonical correlation analysis [18]. A critical requirement is that the kernel matrix is positive definite. Hence it is of interest to show that using the matrix of estimated pairwise distances indeed results in a positive definite kernel matrix.

The distances estimated by the algorithm can be seen as conventional Euclidean distances after embedding the data to a higher-dimensional ($d + N$-dimensional) space in a specific way:

- The first d components as per $x'_{i,l}$ in Eq. (4).
- Each point $\boldsymbol{x}_i$ is offset by s_i in a direction orthogonal to everything else.

Calculating the squared Euclidean distance between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in this space exactly leads to Eq. (7). As the matrix of estimated pairwise distances is equal to a matrix of pairwise distances (in another space), the kernel matrix will be positive definite for any appropriate kernel function.

This interpretation of the estimated distances also ensures that most other properties of distances, such as the triangle inequality, also apply to the estimated distances.

4. Experimental results

To study the performance of the algorithm, some simulated experiments are conducted to compare the proposed algorithm to alternate methods on several data sets with three different performance criteria. 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. The probability of values being missing is gradually increased from 1% to 70%.

4.1. Data

Nine different data sets are selected from the UCI Machine Learning Repository [2]. To make distances meaningful, the variables in each data set are standardised to zero mean and unit variance. As the problem of pairwise distance estimation is unsupervised, the labels for the samples are ignored. The data sets in order of increasing dimensionality:

Iris Iris Data Set. $N = 150$ (samples), $d = 4$ (variables).
Ecoli Ecoli Data Set (ignoring accession number). $N = 336$, $d = 7$.
Breast tissue Breast Tissue Data Set. $N = 106$, $d = 9$.
Glass Glass Identification Data Set (ignoring id). $N = 214$, $d = 9$.
Wine Wine Data Set. $N = 178$, $d = 13$.
Parkinsons Parkinsons Data Set. $N = 195$, $d = 22$.
Ionosphere Ionosphere Data Set (ignoring the second column, which is constant). $N = 351$, $d = 33$.
SPECTF SPECTF Heart Data Set. $N = 267$, $d = 44$.
Sonar Connectionist Bench (Sonar, Mines vs. Rocks) Data Set. $N = 208$, $d = 60$.

These data sets are chosen to be representative of common machine learning tasks, while being varied in terms of dimensionality and structure. The samples in the data sets are distributed in different ways, but decidedly do not correspond to multivariate Gaussian distributions.

4.2. Methods

A total of four different methods are compared:

PDS The Partial Distance Strategy [10,15]. Calculate the sum of squared differences of the mutually known components and scale to the missing components:

$$\hat{d}_{ij}^2 = \frac{d}{d - |M_i \cup M_j|} \sum_{l \notin M_i \cup M_j} (x_{i,l} - x_{j,l})^2 \tag{11}$$

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

ESD The Expected Squared Distances as calculated by the proposed algorithm presented in Section 3.4. The square root of the result is used to get an estimate of the distance. In the notation of Eq. (7)

$$\hat{d}_{ij}^2 = \|\boldsymbol{x}_i' - \boldsymbol{x}_j'\|^2 + s_i^2 + s_j^2 \tag{12}$$

Regression imputation Imputation by the conditional expectation of Eq. (8). This is equivalent to least-squares linear regression, if the covariance matrix and mean were exactly known.

$$\hat{d}_{ij}^2 = \|\boldsymbol{x}_i' - \boldsymbol{x}_j'\|^2 \tag{13}$$

ICkNNI Incomplete-case k-NN imputation [29]. An improvement of complete-case k-NN imputation, here any sample with a valid missingness pattern is viable nearest neighbour. In accordance to the suggestions in [29], up to $k = 5$ neighbours are considered (if there are enough). The imputation fails whenever there are no samples with valid missingness patterns. For such cases, the missing value is imputed by the sample mean for that variable.

The algorithm introduced in this paper outputs an estimate of the distance for any possible case of missing patterns, so no fall-back is needed. The ECM algorithm in Matlab is run with its default parameters, meaning it does not always converge to the strict tolerances in the maximum number of iterations, but the final result is still used.

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:

$$C_1 = \left(\frac{1}{\lambda} \sum_{i>j} (\hat{d}_{ij} - d_{ij})^2 \right)^{\frac{1}{2}} \tag{14}$$

Here, 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 square root of the ESD result is used as an estimate of the distance. 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: $\lambda = MN - M(M+1)/2$.

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:

$$C_2 = \frac{1}{N} \sum_{i=1}^{N} d_{i,\widehat{\mathrm{NN}}(i)} \quad \text{where} \quad \widehat{\mathrm{NN}}(i) = \arg \min_{j \neq i} \hat{d}_{ij} \tag{15}$$

Here, $\widehat{\mathrm{NN}}(i)$ is the nearest neighbour of the ith sample as estimated by the method, and $d_{i,\widehat{\mathrm{NN}}(i)}$ 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 particular, it answers the question of how close the estimated neighbours in fact are, on average. The average distance to the nearest neighbour also represents how well the method is able to estimate small distances, which are more important than large ones in several machine learning applications.

In order to evaluate the accuracy of each method for identifying nearest neighbours, we measure the average size of the intersection between the estimated k nearest neighbours and the true k nearest neighbours for $k = 10$:

$$C_3 = \frac{1}{N} \sum_{i=1}^{N} |\widehat{\mathrm{NN}}(i, 10) \cap \mathrm{NN}(i, 10)|, \tag{16}$$

where $\widehat{\mathrm{NN}}(i, 10)$ is the estimated set of the 10 nearest neighbours, and $\mathrm{NN}(i, 10)$ is the set of the 10 true nearest neighbours.

4.4. Procedure

Before values are removed, the data is standardised to zero mean and unit variance. This is conducted beforehand, to make the scaling consistent for each repetition of the experiment, so that the mean performances can be reasonably estimated as the averages of several randomised experiments. If the scaling for each realisation was slightly different, this would introduce unnecessary variability in the average distances and errors. In terms of the accuracy of the methods, there is no practical difference between standardising before or after the removal of data, as none of the methods assume standardised data. Instead, the ESD method estimates the means and covariances by the ECM algorithm separately for each realisation.

Values are removed from the data set independently at a fixed probably p. For each value of p, 250 repetitions are conducted for the Monte Carlo simulation, and the value of p is gradually increased from 0.01 to 0.70 in increments of 0.01.

Having 250 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 $\alpha = 0.01$. Comparing the performance of the best method to that of every other method results in a multiple hypothesis scenario, and thus the Bonferroni correction is used to control the error rate.

4.5. Results

The average RMSE values for each method are presented in Table 1 for four missingness levels (5%, 15%, 30%, and 60%) for each data set. The most obviously visible trend is that the accuracy decreases with an increasing proportion of missing data for all methods and data sets, as expected. However, it can be seen that in the majority of the cases, the proposed algorithm (ESD) performs the best. The difference compared to regression imputation is not always large, but it is in most cases nevertheless statistically significant. Only for the Ionosphere data are the roles notably reversed. For a high proportion of missing data (60%), ESD obtains the lowest error in every data set tested. The PDS and ICkNNI provide clearly less accurate estimates through most the experiments; only for the most low-dimensional data sets (Iris and Ecoli) with low levels of missing data is the ICkNNI accuracy on par with ESD.

Table 1
Average RMSE of estimated pairwise distances, and standard deviations in parenthesis. The best result for each row is underlined, and any results which are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.01$) from the best result are bolded.

		ESD	Regression imput.	ICkNNI	PDS
Iris. $N = 150$, $d = 4$	5%	**0.244** (0.048)	**0.246** (0.061)	**0.242** (0.059)	0.435 (0.055)
	15%	**0.326** (0.040)	0.338 (0.052)	**0.330** (0.059)	0.591 (0.043)
	30%	**0.495** (0.048)	0.536 (0.071)	0.525 (0.072)	0.840 (0.047)
	60%	**0.919** (0.046)	1.147 (0.093)	1.170 (0.091)	1.202 (0.031)
Ecoli. $N = 336$, $d = 7$	5%	0.482 (0.294)	0.480 (0.303)	**0.464** (0.296)	0.750 (0.220)
	15%	0.634 (0.210)	0.649 (0.220)	**0.626** (0.221)	1.046 (0.126)
	30%	**0.964** (0.237)	1.028 (0.256)	0.989 (0.259)	1.637 (0.119)
	60%	**1.529** (0.203)	1.841 (0.245)	1.731 (0.241)	2.428 (0.082)
Breast Tissue. $N = 106$, $d = 9$	5%	**0.216** (0.105)	**0.215** (0.110)	0.246 (0.107)	0.425 (0.074)
	15%	**0.346** (0.110)	0.349 (0.114)	0.446 (0.133)	0.654 (0.065)
	30%	**0.574** (0.139)	0.580 (0.149)	0.904 (0.156)	1.086 (0.089)
	60%	**1.171** (0.155)	1.250 (0.187)	1.812 (0.247)	2.059 (0.117)
Glass. $N = 214$, $d = 9$	5%	**0.223** (0.063)	**0.221** (0.073)	0.335 (0.124)	0.521 (0.080)
	15%	**0.427** (0.087)	**0.423** (0.101)	0.584 (0.131)	0.818 (0.070)
	30%	**0.761** (0.109)	**0.760** (0.137)	0.972 (0.134)	1.326 (0.073)
	60%	**1.423** (0.076)	1.630 (0.146)	1.811 (0.133)	2.368 (0.076)
Wine. $N = 178$, $d = 13$	5%	**0.249** (0.029)	0.264 (0.033)	0.268 (0.034)	0.364 (0.025)
	15%	**0.399** (0.030)	0.448 (0.040)	0.481 (0.043)	0.606 (0.029)
	30%	**0.607** (0.034)	0.736 (0.051)	0.980 (0.059)	1.024 (0.040)
	60%	**1.015** (0.033)	1.463 (0.078)	1.731 (0.075)	2.134 (0.043)
Parkinsons. $N = 195$, $d = 22$	5%	**0.174** (0.041)	0.178 (0.045)	0.248 (0.050)	0.332 (0.032)
	15%	**0.306** (0.040)	0.322 (0.045)	0.660 (0.078)	0.597 (0.032)
	30%	**0.487** (0.035)	0.524 (0.044)	1.412 (0.404)	0.990 (0.038)
	60%	**0.904** (0.072)	1.090 (0.084)	2.633 (0.114)	2.328 (0.088)
Ionosphere. $N = 351$, $d = 33$	5%	0.243 (0.014)	**0.223** (0.016)	0.255 (0.020)	0.275 (0.010)
	15%	0.478 (0.017)	**0.424** (0.023)	0.843 (0.083)	0.515 (0.013)
	30%	0.772 (0.028)	**0.680** (0.031)	1.529 (0.052)	0.860 (0.017)
	60%	**1.332** (0.035)	1.405 (0.063)	3.201 (0.055)	2.207 (0.053)
SPECTF. $N = 267$, $d = 44$	5%	**0.201** (0.022)	0.203 (0.023)	0.326 (0.040)	0.332 (0.022)
	15%	**0.394** (0.036)	0.400 (0.039)	0.978 (0.259)	0.627 (0.026)
	30%	**0.680** (0.046)	**0.680** (0.051)	1.735 (0.068)	1.047 (0.034)
	60%	**1.326** (0.092)	1.510 (0.109)	3.681 (0.104)	2.498 (0.069)
Sonar. $N = 208$, $d = 60$	5%	**0.193** (0.030)	**0.193** (0.031)	0.373 (0.039)	0.344 (0.025)
	15%	0.433 (0.040)	**0.420** (0.043)	1.025 (0.070)	0.653 (0.032)
	30%	0.719 (0.048)	**0.702** (0.051)	2.005 (0.070)	1.079 (0.034)
	60%	**1.331** (0.087)	1.498 (0.110)	4.271 (0.097)	2.501 (0.063)

Table 2 shows the corresponding performances in terms of the true distance to the predicted nearest neighbour. While this measure of quality emphasises the estimation of small distances, the relative performances between the methods are nearly identical compared to Table 1. In particular, the ESD is for most data sets able to consistently identify nearest neighbours which are, on average, closer to the query point than the other methods.

As the data sets are ordered according to dimensionality, looking at both Tables 1 and 2, one trend seems apparent. Comparing ESD to regression imputation, it appears that for low-dimensional (d up to around 10) problems, ESD consistently tends to be more accurate, whereas for high-dimensional data the difference appears less pronounced (even if it is still statistically significant). For the lowest-dimensional data sets, ICkNNI is also competitive in terms or RMSE.

Comparing PDS to ICkNNI, it seems that for low-dimensional data, ICkNNI tends to be more accurate. However, as the dimension is increased, along with missingness levels, eventually finding compatible missingness patterns for ICkNNI becomes exceedingly improbable, and the accuracy of the method suffers greatly. The weakness of PDS can to some extent be attributed to discarding part of the data when estimating distances: any values for variables known for only one of the samples are not used. Based on these experiments, the use of PDS cannot be recommended for nearly any task where the accuracy of estimating distances from data with missing values is important. ICkNNI can provide effective results, but only in cases where there are enough samples with suitable missingness patterns.

The average set intersection of the 10 nearest neighbours is presented in Table 3. The relative accuracies between methods are mostly in line with the previous tables, but interestingly, regression imputation now appears more accurate than the

Table 2
Average of the mean distance to the estimated nearest neighbour, and standard deviations in parenthesis. The best result for each row is underlined, and any results which are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.01$) from the best result are bolded.

		ESD	Regression imput.	ICkNNI	PDS
Iris. $N = 150$, $d = 4$	5%	**0.344** (0.013)	0.360 (0.021)	0.362 (0.023)	0.478 (0.064)
	15%	**0.429** (0.024)	0.476 (0.041)	0.482 (0.040)	0.848 (0.103)
	30%	**0.590** (0.038)	0.693 (0.061)	0.693 (0.061)	1.276 (0.125)
	60%	**1.040** (0.063)	1.281 (0.103)	1.315 (0.106)	1.588 (0.108)
Ecoli. $N = 336$, $d = 7$	5%	**0.741** (0.019)	0.770 (0.033)	0.773 (0.035)	1.098 (0.088)
	15%	**0.939** (0.031)	1.013 (0.052)	1.024 (0.053)	1.983 (0.233)
	30%	**1.240** (0.057)	1.388 (0.077)	1.406 (0.077)	2.750 (0.193)
	60%	**1.858** (0.092)	2.111 (0.088)	2.229 (0.092)	2.771 (0.093)
Breast Tissue. $N = 106$, $d = 9$	5%	**0.776** (0.017)	**0.776** (0.020)	0.791 (0.025)	0.860 (0.052)
	15%	**0.854** (0.028)	0.861 (0.034)	0.932 (0.050)	1.086 (0.073)
	30%	**1.007** (0.048)	1.032 (0.064)	1.310 (0.108)	1.562 (0.128)
	60%	**1.521** (0.100)	1.640 (0.123)	2.209 (0.164)	2.307 (0.142)
Glass. $N = 214$, $d = 9$	5%	**0.882** (0.014)	0.889 (0.019)	0.919 (0.032)	1.068 (0.054)
	15%	**1.019** (0.031)	1.040 (0.043)	1.129 (0.057)	1.468 (0.119)
	30%	**1.300** (0.062)	1.352 (0.079)	1.511 (0.088)	2.316 (0.215)
	60%	**2.001** (0.093)	2.199 (0.110)	2.475 (0.121)	2.877 (0.109)
Wine. $N = 178$, $d = 13$	5%	**1.918** (0.016)	1.922 (0.019)	1.927 (0.021)	1.984 (0.033)
	15%	**2.080** (0.034)	2.104 (0.042)	2.142 (0.045)	2.297 (0.059)
	30%	**2.355** (0.049)	2.420 (0.062)	2.683 (0.097)	2.932 (0.115)
	60%	**3.037** (0.087)	3.196 (0.098)	3.640 (0.115)	4.188 (0.125)
Parkinsons. $N = 195$, $d = 22$	5%	**1.640** (0.021)	**1.638** (0.023)	1.661 (0.028)	1.682 (0.030)
	15%	**1.741** (0.032)	**1.742** (0.037)	1.976 (0.075)	1.862 (0.046)
	30%	**1.941** (0.046)	1.951 (0.053)	2.684 (0.161)	2.240 (0.083)
	60%	**2.570** (0.080)	2.653 (0.090)	3.553 (0.117)	4.141 (0.155)
Ionosphere. $N = 351$, $d = 33$	5%	2.775 (0.010)	**2.738** (0.008)	2.744 (0.012)	2.776 (0.013)
	15%	2.899 (0.020)	**2.830** (0.018)	3.148 (0.074)	2.914 (0.023)
	30%	3.088 (0.031)	**3.014** (0.031)	3.405 (0.067)	3.151 (0.036)
	60%	3.596 (0.063)	**3.571** (0.058)	4.148 (0.076)	5.501 (0.197)
SPECTF. $N = 267$, $d = 44$	5%	**4.566** (0.007)	**4.567** (0.007)	4.597 (0.012)	4.620 (0.015)
	15%	**4.660** (0.017)	4.663 (0.017)	4.912 (0.100)	4.820 (0.027)
	30%	**4.871** (0.031)	4.885 (0.034)	5.133 (0.041)	5.222 (0.050)
	60%	**5.485** (0.064)	5.551 (0.071)	5.943 (0.083)	6.825 (0.135)
Sonar. $N = 208$, $d = 60$	5%	**5.293** (0.007)	**5.294** (0.007)	5.340 (0.017)	5.332 (0.013)
	15%	**5.378** (0.020)	**5.377** (0.019)	5.544 (0.067)	5.466 (0.029)
	30%	**5.565** (0.037)	**5.567** (0.040)	5.922 (0.065)	5.749 (0.053)
	60%	**6.336** (0.091)	6.366 (0.096)	7.342 (0.120)	7.400 (0.155)

ESD approach, which is the opposite of the situation Table 2. This apparent contradiction can be resolved by recalling that for non-gaussian data, distances between samples with many missing values will tend to be overestimated by ESD and underestimated by regression imputation. The conclusion is that regression imputation is more likely to correctly identify the specific nearest neighbour – but when it is wrong, the wrong neighbours are further than with ESD (as evidenced by Table 2). This criterion has a subtle bias in this regard, in that it only considers distances which are known to be small in the complete data. Consequently it indirectly favours a method which would tend to report all uncertain distances as small.

Figs. 1–9 compare the errors for all percentages 1–70% for the various methods on all nine data sets, and the proposed algorithm consistently provides competitive performance in terms of both of the considered criteria. The ESD appears to outperform the regression imputation version in most cases, suggesting that adding the variance terms accounting for the uncertainties in the estimated distances provides notable additional value.

5. Conclusions

The algorithm presented in this paper enables the direct estimation of pairwise distances in a data set with missing data. Estimating the distances is useful since there are many well-known and efficient methods to do further processing of the

Table 3
Average size of the mean set intersection of the 10 nearest neighbours, and standard deviations in parenthesis. The best result for each row is underlined, and any results which are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.01$) from the best result are bolded.

		ESD	Regression imput.	ICkNNI	PDS
Iris. $N = 150$, $d = 4$	5%	8.795 (0.252)	**8.896** (0.265)	**8.883** (0.279)	8.481 (0.306)
	15%	7.080 (0.337)	**7.304** (0.349)	**7.304** (0.347)	6.274 (0.367)
	30%	5.147 (0.310)	**5.341** (0.344)	**5.385** (0.338)	4.168 (0.247)
	60%	**2.604** (0.231)	**2.631** (0.235)	2.240 (0.264)	2.542 (0.172)
Ecoli. $N = 336$, $d = 7$	5%	7.389 (0.223)	8.186 (0.214)	**8.228** (0.221)	7.362 (0.268)
	15%	4.836 (0.220)	**5.930** (0.249)	**5.956** (0.247)	4.211 (0.287)
	30%	3.118 (0.183)	**3.755** (0.212)	3.691 (0.234)	1.449 (0.201)
	60%	1.291 (0.134)	**1.462** (0.106)	1.084 (0.108)	0.584 (0.046)
Breast Tissue. $N = 106$, $d = 9$	5%	9.180 (0.210)	**9.322** (0.215)	9.180 (0.227)	8.778 (0.272)
	15%	8.073 (0.246)	**8.321** (0.275)	7.855 (0.335)	7.118 (0.342)
	30%	6.749 (0.273)	**7.015** (0.315)	5.260 (0.589)	5.156 (0.280)
	60%	4.142 (0.288)	**4.247** (0.294)	2.265 (0.252)	2.915 (0.167)
Glass. $N = 214$, $d = 9$	5%	8.876 (0.196)	**8.980** (0.206)	8.709 (0.222)	7.833 (0.258)
	15%	6.815 (0.248)	**7.236** (0.256)	6.771 (0.282)	5.557 (0.231)
	30%	4.658 (0.226)	**5.172** (0.232)	4.420 (0.359)	3.348 (0.203)
	60%	2.082 (0.195)	**2.210** (0.189)	1.434 (0.147)	1.147 (0.089)
Wine. $N = 178$, $d = 13$	5%	8.561 (0.146)	**8.612** (0.157)	8.573 (0.170)	8.247 (0.164)
	15%	**7.070** (0.164)	**7.069** (0.182)	6.843 (0.212)	6.388 (0.166)
	30%	**5.433** (0.194)	5.391 (0.206)	4.020 (0.324)	4.478 (0.175)
	60%	**2.833** (0.178)	2.705 (0.186)	1.703 (0.144)	1.693 (0.115)
Parkinsons. $N = 195$, $d = 22$	5%	9.137 (0.112)	**9.176** (0.122)	8.940 (0.150)	8.597 (0.144)
	15%	8.149 (0.133)	**8.170** (0.144)	6.868 (0.359)	7.139 (0.158)
	30%	**6.945** (0.157)	**6.955** (0.163)	4.114 (0.404)	5.586 (0.148)
	60%	**4.469** (0.189)	4.424 (0.196)	2.165 (0.151)	2.553 (0.125)
Ionosphere. $N = 351$, $d = 33$	5%	8.176 (0.096)	**9.014** (0.064)	8.970 (0.085)	8.384 (0.071)
	15%	7.054 (0.110)	**8.068** (0.084)	5.158 (0.502)	7.221 (0.066)
	30%	5.890 (0.123)	**6.927** (0.109)	4.231 (0.219)	6.106 (0.074)
	60%	3.875 (0.151)	**4.725** (0.132)	2.065 (0.087)	3.114 (0.120)
SPECTF. $N = 267$, $d = 44$	5%	8.756 (0.069)	**8.772** (0.072)	8.426 (0.094)	8.190 (0.083)
	15%	7.613 (0.089)	**7.632** (0.092)	5.860 (0.569)	6.602 (0.095)
	30%	6.071 (0.112)	**6.085** (0.110)	4.802 (0.118)	4.818 (0.104)
	60%	**3.503** (0.143)	3.471 (0.140)	2.134 (0.115)	1.772 (0.089)
Sonar. $N = 208$, $d = 60$	5%	9.179 (0.060)	**9.185** (0.060)	8.670 (0.111)	8.629 (0.064)
	15%	8.254 (0.086)	**8.260** (0.087)	7.309 (0.281)	7.511 (0.081)
	30%	**7.175** (0.105)	**7.174** (0.105)	5.876 (0.117)	6.198 (0.099)
	60%	**4.985** (0.149)	4.972 (0.149)	2.861 (0.135)	3.365 (0.122)

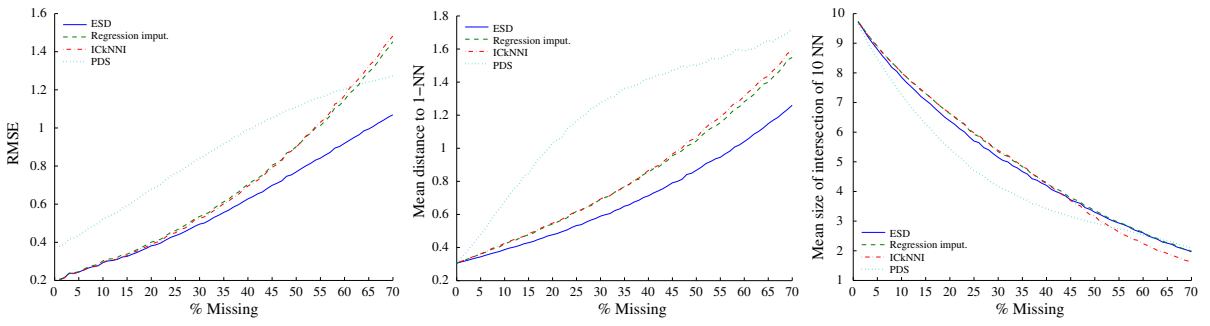

Fig. 1. Iris Data Set. $N = 150$, $d = 4$.

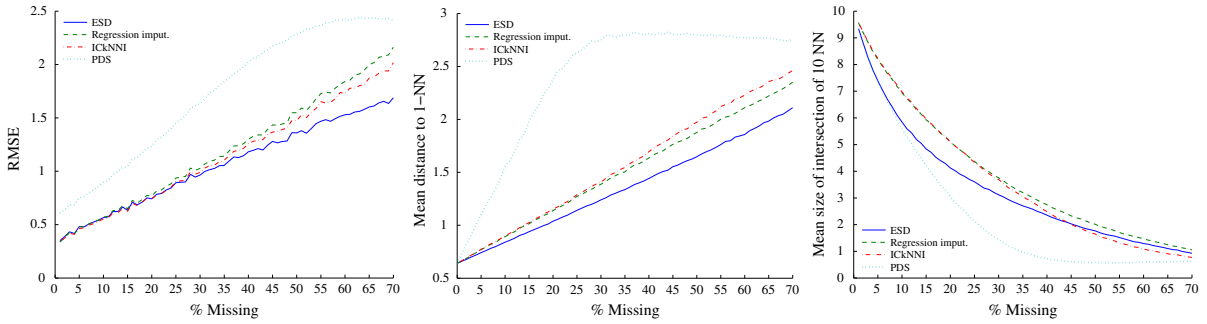

Fig. 2. Ecoli Data Set. $N = 336$, $d = 7$.

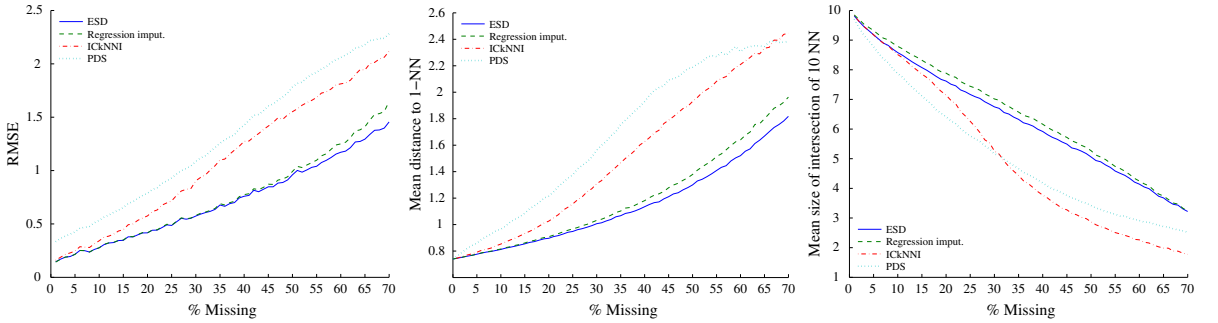

Fig. 3. Breast Tissue Data Set. $N = 106$, $d = 9$.

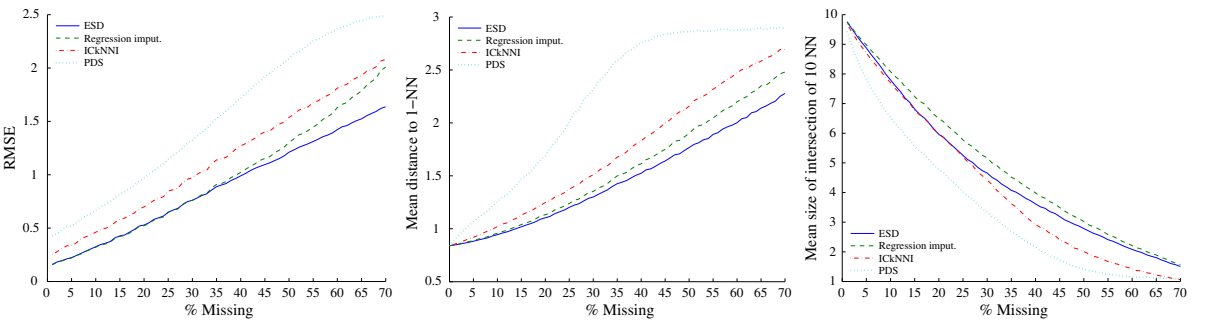

Fig. 4. Glass Identification Data Set. $N = 214$, $d = 9$.

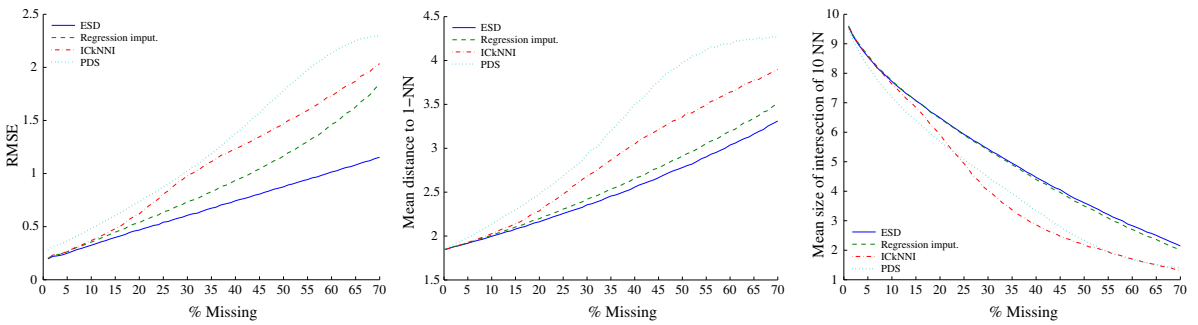

Fig. 5. Wine Data Set. $N = 178$, $d = 13$.

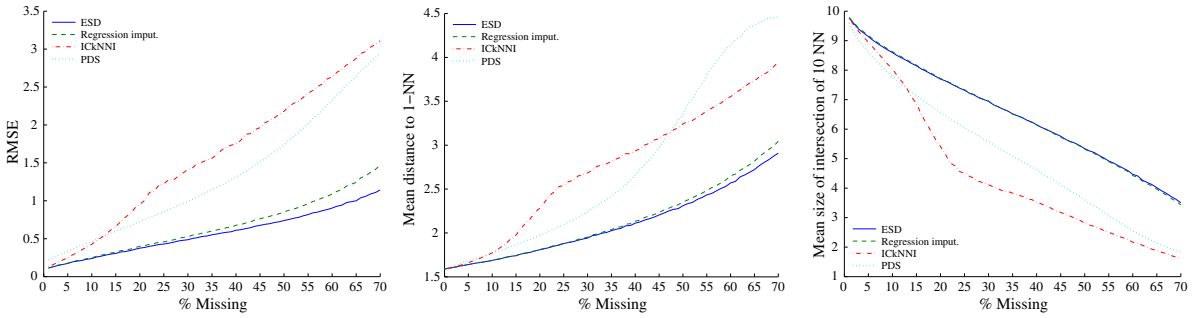

Fig. 6. Parkinsons Data Set. $N = 195$, $d = 22$.

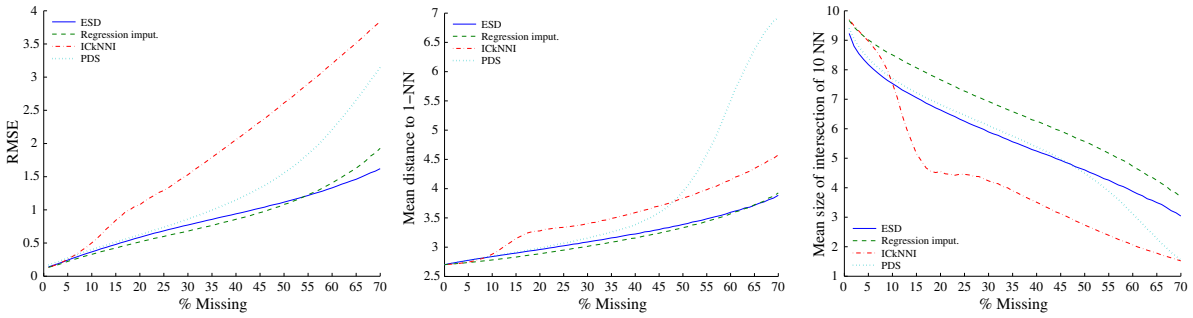

Fig. 7. Ionosphere Data Set. $N = 351$, $d = 33$.

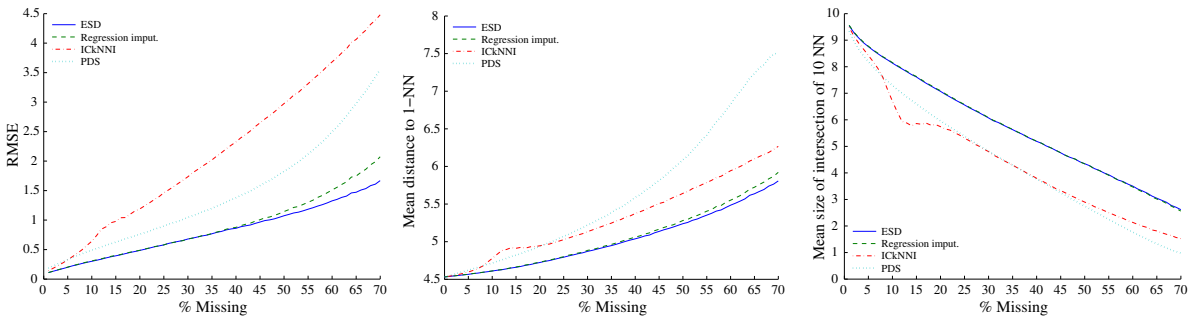

Fig. 8. SPECTF Heart Data Set. $N = 267$, $d = 44$.

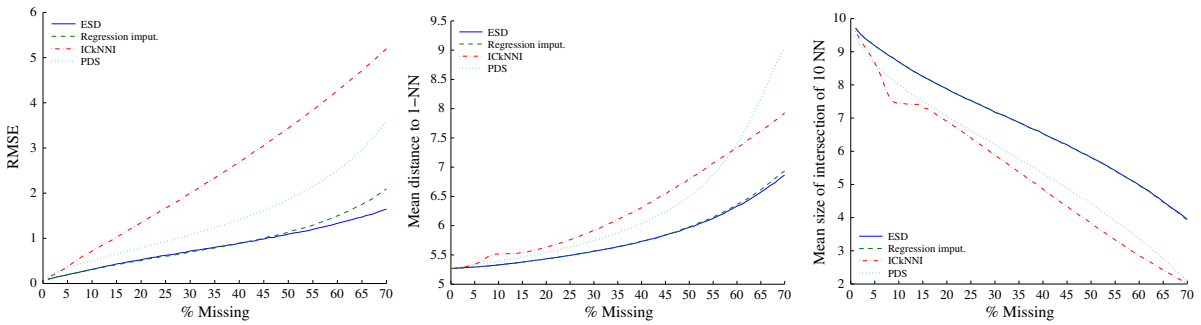

Fig. 9. Connectionist Bench (Sonar, Mines vs. Rocks) Data Set. $N = 208$, $d = 60$.

data based only on the distance matrix. As the algorithm can accurately estimate the pairwise distances in a data set, the distance matrix can be used to apply k-NN or other methods (MDS, SVM, RBF) which rely only on the distances between samples or points, rather than considering the particular coordinates.

Given a data set with missing values, the method is based on using the EM algorithm for maximum likelihood estimation of the mean and covariance. This enables the calculation of the expected squared distance between any two samples by assuming a multivariate Gaussian distribution. The Gaussian distribution maximises the differential entropy, and thus corresponds to maximal uncertainty in the missing values. Hence, this scheme inherently accounts for the uncertainty, and has a tendency to output large distances between samples with a large proportion of missing values. This can be a desirable effect in some processing chains, as it reduces the rate of false positives when searching for nearest neighbours.

Compared to standard methods for estimating distances (PDS or imputation), the proposed algorithm provides more accurate results across the entire range of missingness of data, as evidenced by the experiments in Section 4. In all the tested cases, the algorithm is the most accurate when more than 50% of the data missing, while other methods appear to reach serious difficulties at lower missingness levels. These experiments support the conclusion that accounting for the uncertainty in imputation when estimating distances leads to a significant improvement in accuracy.

As this paper has clearly shown the efficiency of the proposed algorithm for distance estimation in the presence of missing data, future work should investigate its interest for machine learning and pattern recognition problems; these problems could include clustering, regression, classification, projection, and feature selection.

References

[1] T.W. Anderson, An Introduction to Multivariate Statistical Analysis, third ed., Wiley-Interscience, New York, 2003.
[2] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, University of California, School of Information and Computer Sciences, Irvine, 2011.
[3] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, USA, 1995.
[4] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[5] J.G. Cleary, L.E. Trigg, K*: an instance-based learner using and entropic distance measure, in: A. Prieditis, S.J. Russell (Eds.), 12th International Conference on Machine Learning, Morgan Kaufman, 1995, pp. 108–114.
[6] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297.
[7] M. Cottrell, P. Letrémy, Missing values: processing with the Kohonen algorithm, in: J. Janssen, P. Lenca (Eds.), Proc. International Symposium on Applied Stochastic Models and Data Analysis (ASMDA), pp. 489–496.
[8] T.M. Cover, J.A. Thomas, Elements of Information Theory, second ed., Wiley, 2006.
[9] M.A.A. Cox, T.F. Cox, Multidimensional scaling, in: Ch. Chen, W. Härdle, A. Unwin (Eds.), Handbook of Data Visualization, Springer Handbooks of Computational Statistics, Springer, Berlin Heidelberg, 2008, pp. 315–347.
[10] J.K. Dixon, Pattern recognition with partly missing data, IEEE Transactions on Systems, Man and Cybernetics 9 (1979) 617–621.
[11] G. Doquire, M. Verleysen, Feature selection with missing data using mutual information estimators, Neurocomputing 90 (2012) 3–11.
[12] C.K. Enders, Applied missing data analysis, methodology, in: The Social Sciences, Guilford Press, 2010.
[13] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognition 41 (2008) 3692–3705.
[14] J.W. Grzymala-Busse, W.J. Grzymala-Busse, Handling missing attribute values, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, second ed., Springer, US, 2010, pp. 33–51.
[15] L. Himmelspach, S. Conrad, Clustering approaches for data with missing values: comparison and evaluation, in: 2010 Fifth International Conference on Digital Information Management (ICDIM), pp. 19–28.
[16] E.R. Hruschka, E.R. Hruschka Jr., N.F.F. Ebecken, Evaluating a nearest-neighbor method to substitute continuous missing values, AI 2003: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 2903, Springer, Berlin Heidelberg, 2003, pp. 723–734.
[17] P. Jönsson, C. Wohlin, An evaluation of k-nearest neighbour imputation using Likert data, in: Proc. International Symposium on Software Metrics, IEEE Computer Society, 2004, pp. 108–118.
[18] P.L. Lai, C. Fyfe, Kernel and nonlinear canonical correlation analysis, International Journal of Neural Systems 10 (2000) 365–377.
[19] J. Li, N. Cercone, Assigning missing attribute values based on rough sets theory, in: 2006 IEEE International Conference on Granular Computing, IEEE Computer Society, 2006, pp. 607–610.
[20] R.J.A. Little, D.B. Rubin, Statistical Analysis with Missing Data, second ed., Wiley Interscience, 2002.
[21] MathWorks, Matlab Financial Toolbox R2012a Documentation: ecmnmle, 2012.
[22] X.L. Meng, D.B. Rubin, Maximum likelihood estimation via the ECM algorithm, Biometrika 80 (1993) 267–278.
[23] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, K.R. Müller, Fisher discriminant analysis with kernels, in: Proceedings of the 1999 IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing IX, 1999, pp. 41–48.
[24] D.B. Rubin, Multiple Imputation for Nonresponse in Surveys, Wiley Series in Probability and Statistics, Wiley, 1987.
[25] J.L. Schafer, J.W. Graham, Missing data: our view of the state of the art, Psychological Methods 7 (2002) 147–177.
[26] B. Schölkopf, A. Smola, K.R. Müller, Kernel principal component analysis, in: W. Gerstner, A. Germond, M. Hasler, J.D. Nicoud (Eds.), Artificial Neural Networks – ICANN'97, Lecture Notes in Computer Science, vol. 1327, Springer, Berlin Heidelberg, 1997, pp. 583–588.
[27] J. Sexton, A.R. Swensen, ECM algorithms that converge at the rate of EM, Biometrika 87 (2000) 651–662.
[28] G. Shakhnarovich, Nearest-Neighbor Methods in Learning and Vision, MIT Press, Cambridge, 2005.
[29] J. Van Hulse, T.M. Khoshgoftaar, Incomplete-case nearest neighbor imputation in software measurement data, Information Sciences (2011), http://dx.doi.org/10.1016/j.ins.2010.12.017.
[30] I. Wasito, B. Mirkin, Nearest neighbour approach in the least-squares data imputation algorithms, Information Sciences 169 (2005) 1–25.