Neurocomputing 118 (2013) 65–78

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Locally linear reconstruction based missing value imputation for supervised learning

Pilsung Kang *

Department of Industrial & Information Systems Engineering, College of Business and Technology, Seoul National University of Science & Technology (Seoultech), 139-743, 232 Gongreung ro, Nowon-gu, Seoul, South Korea

ARTICLE INFO

Article history:
Received 1 October 2012
Received in revised form
4 December 2012
Accepted 7 February 2013
Communicated by H. Yu
Available online 14 March 2013

Keywords:
Locally linear reconstruction (LLR)
Missing value imputation
Supervised learning
Classification
Regression

ABSTRACT

Most learning algorithms generally assume that data is complete so each attribute of all instances is filled with a valid value. However, missing values are very common in real datasets for various reasons. In this paper, we propose a new single imputation method based on locally linear reconstruction (LLR) that improves the prediction performance of supervised learning (classification & regression) with missing values. First, we investigate how missing values degrade the prediction performance with various missing ratios. Next, we compare the proposed missing value imputation method (LLR) with six well-known single imputation methods for five different learning algorithms based on 13 classification and nine regression datasets. The experimental results showed that (1) all imputation methods helped to improve the prediction accuracy, although some were very simple; (2) the proposed LLR imputation method enhanced the modeling performance more than all other imputation methods, irrespective of the learning algorithms and the missing ratios; and (3) LLR was outstanding when the missing ratio was relatively high and its prediction accuracy was similar to that of the complete dataset.

1. Introduction

Supervised learning algorithms such as classification and regression in data mining or machine learning generally assume that training and test datasets are complete, i.e., each attribute of all instances is not missing and they are filled with a value. However, real data sets are often incomplete and they contain a proportion of missing values for various reasons, such as the death of a patient, equipment malfunctions, and a lack of responses [2]. The presence of missing values can lead to critical problems during the learning process, such as a loss of efficiency, biased data structure, analytical difficulties, and prediction performance degeneration [3,14,19]. According to Acuna and Rodriguez [1], less than 1% missing instances does not affect the prediction performance in general, while 1–5% is manageable. However, 5–15% missing instances requires sophisticated handling method, while greater than 15% missing data can severely degrade the prediction performance of learning algorithms. In order to handle missing values, several imputation techniques have been proposed in a wide range of data mining and machine learning domains [21,22,27,41,45]. The aim of missing value imputation is to enhance the functionality of learning algorithms and to improve their prediction accuracy, by replacing missing attributes with real values based on information extracted other non-missing data. The treatment of missing values depends on the type of missing values as follows [29,36].

- *Missing completely at random* (*MCAR*): this is the highest level of randomness. The probability of an instance having a missing value for an attribute does not depend on either the observed data or the missing attribute. Any missing value imputation method rarely distort the distribution of original data.
- *Missing at random* (*MAR*): an intermediate level of randomness. The probability of an instance having a missing value for an attribute may depend on the known values, but not on the value of the missing data itself. For example, let us assume that two attributes, *gender* and *pregnancy*, are collected together. If *gender* is recorded as 'male', we can easily deduce that *pregnancy* is 'no' although it is missing [38].
- *Not missing at random* (*NMAR*): this is the lowest level of randomness. The probability of an instance having a missing value for an attribute may depend on the value of that attribute. For example, ex-convicts are likely to leave the *criminal record* attribute missing when they respond to a survey.

If missing values occur that are *MAR* or *NMAR*, imputation can be conducted by domain experts based on their appropriate background knowledge. Therefore, most missing value imputation

* Tel.: +82 2 970 7286; fax: +82 2 979 3377.
E-mail address: pskang@seoultech.ac.kr

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.02.016

techniques are focused on missing values that are classed as *MCAR* [2,13,38].

Depending on the learning algorithm and the number of repetitions, the handling of *MCAR* values can be divided into two main groups. The first group includes learning algorithms that can handle missing values during the learning process. Classification and regression tree (CART) simply ignores missing values when growing a tree [6]. CART iteratively computes the information gain for a large number of split candidates to select the best split point (the attribute and its split value). During CART learning, instances with missing values for an attribute are discarded if a candidate split point uses that attribute, whereas they are used if other non-missing attributes are selected for a split candidate. Naive Bayesian classifier treats missing values in a similar way [26]. When estimating the distribution of each attribute, instances with missing values for that attribute are abandoned and the parameters of the distribution are approximated using the non-missing instances only. When computing the distance between two instances in k-nearest neighbor (k-NN) learning, zero is assigned to an attribute if both instances have missing values. If only one is missing, k-NN assigns the maximum distance of that attribute [40]. The second group includes all imputation techniques that work independently of learning algorithms. This group can be divided into two subgroups: single imputation (SI) and multiple imputation (MI). SI replaces a missing value with a single value, whereas MI replaces a missing value with different values Thus MI transforms a single incomplete dataset into a number of complete datasets. Some representative single imputation techniques are as follows.

- *Mean* (*mode*) *imputation* (*MEI*): this is a simple but fairly effective method in practice. MEI fills the missing values of an attribute with the mean (continuous) or mode (nominal/ordinal) of the non-missing values for the same attribute [13,15].
- *k-nearest neighbor* (*k-NN*): if an instance has a certain attribute missing, k-NN finds k most similar instances using its non-missing attributes. The values of the missing attributes of k neighbors are combined based on a predefined rule or kernel function, such as a simple average or exponential kernel, and it replaces the missing value. The k-NN imputation method is also known as '*Hot Deck*' if $k=1$ is used, while a number of other variations have been proposed based on the modification of the kernel functions [13,30,15,20,41,46].
- *Expectation conditional maximization* (*ECM*): this approach assumes that the entire dataset is derived from a multivariate Gaussian distribution. Initially, the distribution parameters (mean vector and covariance matrix) are estimated for the data without missing values. The expectation-maximization (EM) algorithm is conducted as follows. During the expectation (E) step, the missing values are imputed based on the mean value for its attribute. During the maximization (M) step, the distribution parameters are updated based on the imputed values. After iterating the E–M process, the distribution parameters converge to the optimal values, and the missing values are imputed using values that are consistent with the distribution [10,31,16,35,38].
- *Clustering-based imputation*: when clustering-based imputation methods are applied to an instance with a missing attribute, the entire dataset is grouped into some number of clusters using the non-missing attributes. The attribute values of the members of the cluster nearest to the instance are then used for imputation. Clustering algorithm such as *K*-Means clustering (KMC) or a mixture of Gaussian distributions (MoG) are widely used [32,42,44,28,11].
- *Model-based imputation*: in model-based imputation methods, missing value imputation is reformulated as a supervised learning problem where the missing attribute becomes the dependent (target) variable and the non-missing attributes become independent (explanatory) variables. Thus, the learning task becomes classification if the missing attribute is nominal, whereas it becomes regression if the missing attribute is continuous. For each instance with a missing attribute, a machine learning algorithm is trained based on the instances without missing values and the non-missing values of the instance are used by the model to predict the target missing attribute value. Multiple linear regression, artificial neural network (ANN), Naive Bayesian classification, decision trees, and support vector machines (SVM) are some examples of machine learning algorithms that are commonly used for model-based imputation [18,12,13,45,21,37].

In contrast to single imputation methods, multiple imputation methods impute a set of possible values rather than a single value for the missing attribute of an instance [43,34]. Thus, multiple imputation methods generate a number of different datasets where the complete instances are identical but the incomplete instances have different values for the missing attributes. Some representative multiple imputation methods are as follows.

- *Multivariate imputation by chained equations* (*MICE*) [39]: if missing values occur in more than one attribute for an instance, MICE employs a chained equation to fill the missing value of each attribute. MICE can generate various imputation results by modifying the imputation sequence of the missing attributes or the imputation algorithm for each attribute.
- *Boosting* [14]: This multiple imputation method has three modules, i.e., mean pre-imputation, application of confidence intervals, and boosting. The pre-imputed values in the first module are imputed using a base imputation method that filters the missing values by generating confidence intervals using Student's t-statistics. Based on these confidence intervals, boosting is performed to deliver the high-quality imputed values.

These missing value imputation methods have advantages and limitations. Imputation methods inherent in learning algorithms do not require additional data preprocessing for missing value treatment, but they are usually too simple because most simply discard instances with missing attributes. This may allow learning algorithms to function but their prediction performance cannot be guaranteed. Single imputation methods can be applied before any learning algorithms. However, the prediction performance improvement may be restricted (e.g., mean imputation) or the computational burden might be increased because of the additional parameter optimization process (model-based imputation). Multiple imputation methods may improve the prediction performance better than single imputation methods. However, they significantly increase the computational cost not only by repeating the imputation steps, but also by repeating model learning based on individual imputed datasets. Therefore, multiple imputation methods may have difficulties handling large amount of data during real-time processing.

In this paper, we propose a new efficient single imputation method based on locally linear reconstruction (LLR) [24,25] to improve the prediction performance of supervised learning. LLR is a structured approach that determines two parameters for k-NN learning, i.e., the number of nearest neighbors (k) and the weights given to the neighbors. In LLR, the optimization problem is formulated to minimize the difference between the test instance

and its projection in the neighborhood space. Obtaining the solution of the optimization problem is equivalent to assigning the optimal weights to the neighbor instances, i.e., if a neighbor is necessary to span the neighborhood space, a non-zero weight is assigned, whereas zero is assigned otherwise. Therefore, LLR can determine the neighbors that are critical and assign appropriate weights to them. By adopting LLR, k and the weights of neighbors are not model parameters any more, but they are determined in a structured manner. To verify the effectiveness of the LLR imputation method, we analyzed the extent to which classification and regression algorithms are affected by diverse missing ratios if imputation is not conducted. Next, we compared (1) the accuracy improvement versus the incomplete data and (2) the accuracy recovery versus the complete data (ideal case) when using LLR imputation with other well-known single imputation methods. In order to generalize our conclusions, we conducted extensive experiments using six single imputation methods as benchmark methods and five classification and regression algorithms with 13 different degree of missing ratios, based on 13 classification and nine regression datasets.

The remainder of this paper is organized as follows. In Section 2, we review some representative work associated with the comparative analysis of various imputation methods and we discuss their findings and limitations. In Section 3, we describe the proposed LLR imputation method. In Section 4, we explain the experimental settings, i.e., the data description, missing data ratios, benchmark imputation algorithms, the classification and regression algorithms along with their user-specific parameters, and the performance measures. In Section 5, we compare LLR imputation method with other benchmark imputation methods in various scenarios. In Section 6, we provide some concluding remarks and we discuss areas of future work.

2. Related work

In this section, we review some representative works that have focused on the comparative study of missing value imputation methods for supervised learning in chronological order.

- Grzymala-Busse and Hu [18]: In this study, nine imputation methods for nominal datasets were examined. The imputation methods were grouped into three categories: (1) two mode imputation-based methods where the key distinction was whether class information was used or not; (2) two methods based on the assignment of all possible values (class information was also the key difference); and (3) five miscellaneous methods including C4.5 [33], event-covering method, LEM2 algorithm [17], and treating missing values as a special value designated as '*new*'. Two rule-based classifiers, i.e., naïve LERS [17] and new LERS, were used as base classifiers, and the imputation methods were compared using 10 datasets where the missing ratio varied between 1% and 13%. Based on the Wilcoxon matched-pairs signed rank test, C4.5 was found to be the most effective imputation method. However, the result cannot be accepted in general, not only because only a specific classifier (LERS) was used, but also the missing ratio was different among the datasets.
- Batista and Monard [2]: Three imputation methods, i.e., mean (mode) imputation, Hot-deck, and k-NN imputation, were compared using four datasets with six missing ratios that ranged from 10% to 60%, with 10% increments. Two rule-based classification algorithms, C4.5 and CN2 [8], were used as base classifiers. The experimental results showed that k-NN imputation where $k=10$ delivered good classification performance, even with a large amount of missing data. Some limitations of this study were: (1) only simple imputation methods were examined, (2) only a few datasets were analyzed, and (3) only rule-based classifiers were employed, which prevented the generalization of the experimental results. In addition, this study used an impractical assumption that a maximum of three attributes had missing values.
- Acuna and Rodriguez [1]: Three imputation methods, i.e., mean imputation, median imputation, and k-NN imputation were compared using 12 datasets and two classifiers (linear discriminant analysis and k-NN classification). Three different missing ratios that ranged from 1% to 21% were examined for each dataset. The main conclusions of this study were: (1) any imputation method could be used for very low missing ratios, (2) k-NN imputation was outstanding with high missing ratios but only when it was combined with a k-NN classifier. However, the limitations included: (1) a fair comparison was not possible because different missing ratios were used for different datasets and (2) only a few basic imputation methods were investigated.
- Farhangfar et al. [13]: Six imputation methods for discrete data were compared in various experimental settings. The imputation methods include mean imputation, polytomous regression, Hot-deck, Naïve Bayesian, Boosting with Hot-deck, and Boosting with Naïve Bayesian. They were compared using six missing ratios (5%, 10%, 20%, 30%, 40%, and 50%) with five classifiers (RIPPER, C4.5, k-NN, support vector machine (SVM), and Naïve Bayes) and 15 datasets. The experimental results showed that imputation improved the classifiers, but the effect varied for different classifiers. They also concluded that no imputation method was universally best in all experimental settings. Some limitations of this study were: (1) single and multiple imputations were mixed, (2) the missing ratio was divided roughly, and (3) they assumed that missing values occurred only in the training dataset.
- Su et al. [37]: Then model-based imputation methods were examined for nominal datasets. All attributes were assumed to be nominal, so each missing attribute was set as the target whereas other non-missing attributes were set as the inputs. The ten classification algorithms were compared, which had already been implemented in WEKA, i.e., C4.5, dTable, lazy Bayesian rules, Naïve Bayes, One Rule, decision list, random forests, SVM, radial basis function network (RBF), and multilayer perceptron neural network (MLP). Using four classification algorithms (k-NN, Naïve Bayes, C4.5, and MLP), the imputation methods were tested with five missing ratios (10%, 20%, …, 50%) and 12 datasets. The authors insisted that SVM-based and C4.5-based imputation methods performed well, but their prediction accuracy improvements did not appear to be significantly better than the other methods.
- Ding and Ross [11]: Six imputation methods were analyzed in the context of multi-biometric fusion. Of these imputation methods, four were single imputations (k-NN, maximum likelihood estimation (MLE), mixture of Gaussians (MoG), and predictive mean matching (PMM)) while the other was a multiple imputation method based on MICE. This study conducted pairwise comparisons between two selected imputation methods based on the dataset stored in the Michigan State University database. Based on the experimental results, it was concluded that MoG-based imputation method performed better than other methods. However, this conclusion should be treated with caution because it was based on an inadequate experimental design that lacks a range of missing ratios and different classifiers.

Previous studies have reported some valuable experimental findings, but they all shared two common limitations: (1) the lack

of missing ratio diversity in verifying imputation method and (2) the learning tasks were biased toward classification. Therefore, the goal of the current study was to verify our proposed LLR imputation method by comparing it with other imputation method and to generalize its effective application in different scenarios, such as a wide range of missing ratios, different learning tasks (classification and regression), and different learning algorithms.

3. Missing value imputation based on locally linear reconstruction

Locally linear reconstruction (LLR) was proposed to simultaneously determine the critical neighbors and to assign their optimal weights in k-NN learning [24]. The input–output structure of LLR is shown in the following equation:

$$\hat{y}_{n+1} = LLR(\mathbf{x}_{n+1}, \mathbf{X}^{ref}, \mathbf{y}^{ref}, k), \tag{1}$$

where $\mathbf{x}_{n+1}$ is a $1 \times d$ input vector of a new test instance, $\hat{y}^{n+1}$ is the target value of the test instance, $\mathbf{X}^{ref}$ is an $n \times d$ input matrix where each row corresponds to each reference instance, $\mathbf{y}^{ref}$ is the $n \times 1$ vector of the target values of the reference instances and k is the initial number of nearest neighbors. The LLR procedure is shown in Fig. 1. After a new test instance is given (Fig. 1(a)), its k nearest neighbors are selected in the references set (Fig. 1(b)). The similarity between two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed using an appropriate distance measure. The Minkowski distance, as shown in the following equation, is commonly used for numerical attributes:

$$distance(\mathbf{x}_i, \mathbf{x}_j) = \left|(\mathbf{x}_i - \mathbf{x}_j)^p\right|^{1/p}. \tag{2}$$

In general, the Euclidean distance with $p=2$ is used for continuous variables, while the Manhattan distance with $p=1$ is used for ordinal variables. The second step is to find the convex combination of the selected neighbors that minimizes the reconstruction error, which is the difference between the test pattern and its projection onto the convex hull of the neighbors (Fig. 1(c)). Assuming that the local structure around the test instance is linear, we can formulate the optimization problem as follows:

$$E(\mathbf{w}) = \frac{1}{2}\left|\mathbf{x}_{n+1} - \sum_{j=1}^{k} w_j \mathbf{x}_{n+1}^{j}\right|^2, \tag{3}$$

where $\mathbf{x}_{n+1}^{j}$ is the jth nearest neighbor of $\mathbf{x}_{n+1}$ and w_j is the weight assigned to $\mathbf{x}_{n+1}^{j}$. After taking the derivative with respect to $\mathbf{w}$, the explicit solution can be obtained (Fig. 1(d))

$$\begin{aligned} &\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = -(\mathbf{x}_{n+1} - \mathbf{X}_{n+1}^{NN}\mathbf{w})^{\mathrm{T}}\mathbf{X}_{n+1}^{NN} = 0, \\ &\mathbf{w}^T \mathbf{X}_{n+1}^{NN} T \mathbf{X}_{n+1}^{NN} = \mathbf{x}_{n+1}^{\mathrm{T}} \mathbf{X}_{n+1}^{NN}, \\ &\mathbf{w} = [(\mathbf{x}_{n+1}^{\mathrm{T}} \mathbf{X}_{n+1}^{NN})(\mathbf{X}_{n+1}^{NN} T \mathbf{X}_{n+1}^{NN})^{-1}]^{\mathrm{T}}, \end{aligned} \tag{4}$$

where $\mathbf{X}_{n+1}^{NN}$ is a $k \times d$ matrix where each row corresponds to the kth nearest neighbor of $\mathbf{x}_{n+1}$. Using the obtained optimal $\mathbf{w}$, the prediction is made as follows:

$$\hat{y}_{n+1} = \arg\max_{j} \sum_{i \in k-NN(\mathbf{x}_{n+1}),\ y_i = j} w_i \ \text{(classification)} \tag{5}$$

$$\hat{y}_{n+1} = \sum_{i \in k-NN(\mathbf{x}_{n+1})} w_i \cdot y_i \quad \text{(regression)} \tag{6}$$

The reconstruction error $E(\mathbf{w})$ can be understood as the Euclidean distance between the test instance and its image projected onto the hyperplane created by its neighbors. This image is the best one that describes the test instance, i.e., it is the closest point to the test instance from the hyperplane spanned by the selected neighbors. The weight $\mathbf{w}$ obtained by minimizing $E(\mathbf{w})$ provides the answers to two questions: (1) is this neighbor really important? And (2) if so, how important? If the neighbor $\mathbf{x}_{n+1}^{j}$ is important, i.e., if it is needed for reconstructing the test instance, its weight w_j must be non-zero. If this neighbor is more important

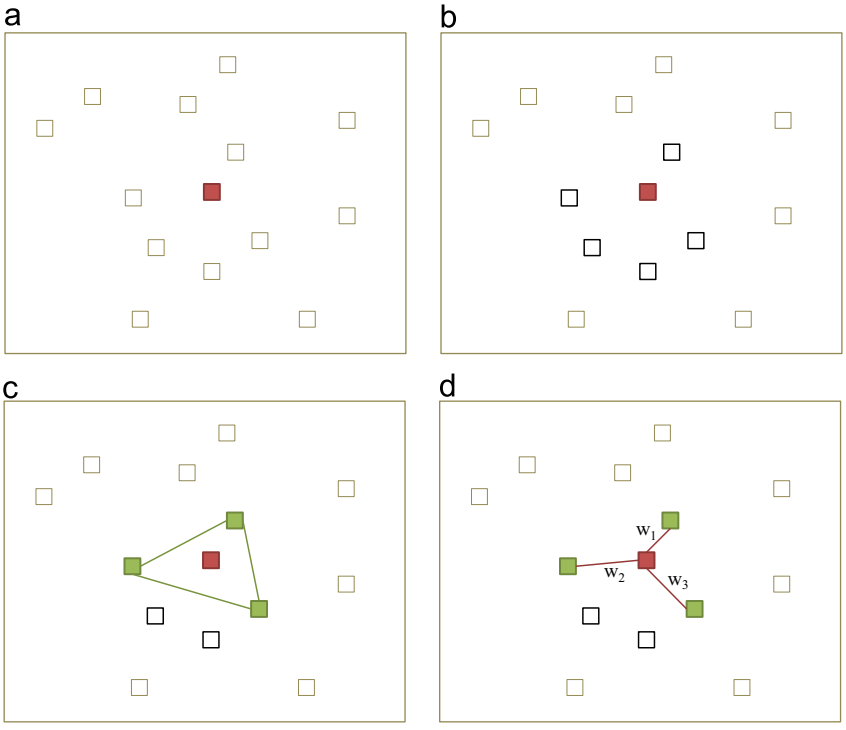

Fig. 1. Locally linear reconstruction (LLR) procedure: (a) a new test instance is given, (b) find k-NN ($k=5$ in this example), (c) find critical neighbors and (d) assign weights.

than the other neighbors, $|w_j|$ should be larger than that of the others. If k and k' denote the number of nearest neighbors that are considered initially and with nonzero weight, respectively, it always holds that $k' \le k$. If the optimal number of nonzero weighted neighbors that minimizes the reconstruction error is k^*, LLR must set k in Eq. (1) to be at least equal to or greater than k^*. The information about k^* is unknown in advance, but it has been reported that a sufficiently large k gave a better prediction accuracy than other conventional weight allocation techniques [24]. Given a sufficiently large k, therefore, LLR can resolve the two main parameter selection problems of k-NN learning, i.e., the number of neighbors and their corresponding weights.

Fig. 2 shows an example of LLR-based missing value imputation. The dataset used in this hypothetical example contained n instances with d attributes (variables). Let us assume that the mth attribute of the ith instance, the mth and pth attributes of the jth instance, and the pth attribute of the kth instance are missing (Fig. 2(a)). The first step of LLR imputation is to divide the entire dataset into the complete and missing datasets where there are no missing values in the complete dataset whereas every instance in the missing dataset has at least one missing attribute. In our example, the missing dataset has three instances while the complete dataset has $(n-3)$ instances (Fig. 2(b)). In the second step, the following imputation process is repeated for each instance with missing attributes. First, divide an instance with missing attributes into input and target vectors so the non-missing attributes belong to the input vector and the missing attributes belong to the target vector. For example, the ith instance has one missing attribute (m) so it is divided into a $1 \times (d-1)$ input vector consisting of $(1,2,\ldots,m-1,m+1,\ldots,d)$th attributes and a 1×1 scalar target value of the mth attribute. We then divide the complete dataset in the same manner. Thus, the complete dataset is divided into an $(n-3) \times (d-1)$ input matrix and an $(n-3) \times 1$ target vector (Fig. 2(c)). Finally, we conduct LLR to fill the missing values. In this example, the input vector of an instance with missing attributes becomes $\mathbf{x}_{n+1}$, the input matrix of the complete dataset becomes $\mathbf{X}^{ref}$, and the target vector of the complete dataset becomes $\mathbf{y}^{ref}$ in Eq. (1). As explained previously, k can be set to a sufficiently large number. As a result, $\hat{y}_{n+1}$ in Eq. (1) is imputed for the mth (missing) attribute of the ith instance. When computing $\hat{y}_{n+1}$, Eq. (6) is used if the missing attribute is numerical, whereas Eq. (5) is used if the missing attribute is nominal. If more than two attributes are missing, all attributes are imputed simultaneously by following the same steps. The LLR-based imputation procedure is summarized in Fig. 3.

We anticipate that our proposed LLR-based imputation method can enhance the prediction accuracy and provide robust performance. First, it can improve the accuracy of the classification and regression algorithms. Most simple imputation methods fill the missing values with the same value or impute them based on a data structure with strict parametric assumptions so the performance improvement is marginal but not impressive. However, LLR does not assume a specific distribution for the data structure but it takes into account the local structure around instances with missing values. Thus, LLR facilitates a locally adopted flexible imputation. Therefore, it is less likely that the data structure is distorted after imputation, which improves the accuracy. Second, it can deliver a

Inputs:
$\mathbf{X}^{original}$: an $n \times d$ original dataset with missing values
k: the initial number of nearest neighbors for LLR

Outputs:
$\mathbf{X}^{imputed}$: an $n \times d$ imputed dataset

Step 1: Divide the data set into complete and missing datasets
$\mathbf{X}^{original} = \mathbf{X}^{complete} \cup \mathbf{X}^{missing}$, $\mathbf{X}^{complete} \cap \mathbf{X}^{missing} = \phi$
$\mathbf{X}^{complete}$: $n_c \times d$ matrix with no missing values
$\mathbf{X}^{missing}$: $n_m \times d$ matrix where each row has at least one missing column.

Step 2: Impute each row of $\mathbf{X}^{missing}$ based on LLR
for i=1 to n_m
$\mathbf{x}_i$ = the i^{th} row of $\mathbf{X}^{missing}$
$I_{missing}$ = a set of missing attribute indexes of $\mathbf{x}_i$
$I_{complete}$ = a set of non-missing attribute indexes of $\mathbf{x}_i$
$\mathbf{x}_i^{input} = \mathbf{x}_i(I_{complete})$
$\mathbf{x}_i^{target} = \mathbf{x}_i(I_{missing})$
$\mathbf{X}_{ref}^{input} = \mathbf{X}^{complete}([1:1:n_c], I_{complete})$
$\mathbf{X}_{ref}^{target} = \mathbf{X}^{complete}([1:1:n_c], I_{missing})$
$\mathbf{x}_i^{target} = LLR(\mathbf{x}_i^{input}, \mathbf{X}_{ref}^{input}, \mathbf{X}_{ref}^{target}, k)$
end

Fig. 3. The LLR-based missing value imputation procedure.

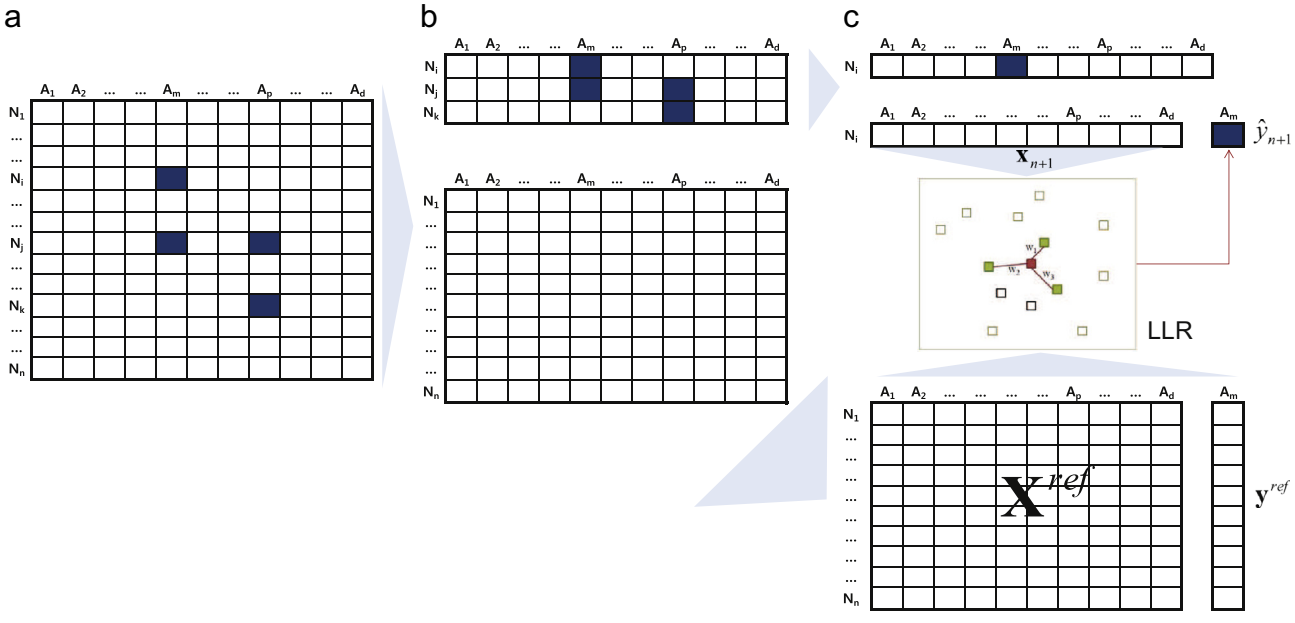

Fig. 2. An illustrative example of missing value imputation based on locally linear reconstruction (LLR).

robust performance by reducing the parameter sensitivity of model-based imputation methods. In general, model-based imputation methods are highly sensitive to the algorithm parameters. For example, the predictive power is enhanced if an appropriate number of hidden nodes is chosen for ANN. However, we also need to consider the risk that inappropriately selected hidden nodes may degrade the learning performance, even after imputation. LLR can find the optimal set of neighbors and their corresponding weights provided that k is set to a sufficiently large number, so its parameter sensitivity is negligible.

4. Experimental settings

To verify the effectiveness of LLR imputation, we designed the following experiments. First, we generate synthetic incomplete datasets from the original complete dataset with various missing instance ratios. Five classification and regression algorithms were trained without any imputation methods to determine the detrimental effects of missing values. The prediction accuracies with the original complete dataset and the missing dataset without imputation, and the prediction power degradation with the missing ratios were compared. Second, we compared the prediction performance of our LLR imputation method with six well-known single imputation methods and that of the original complete dataset.

4.1. Datasets

A total of 13 datasets were used for classification and nine datasets were used for regression. A description of each dataset used for classification and regression is shown in Tables 1 and 2, respectively. The classification datasets were selected from the UCI Machine Learning Repository.[1] The number of instances varied from 150 (Iris) to 14 429 (Shuttle), while the number of attributes varied from 4 (Iris) to 60 (Sonar). Five datasets (Sonar, Liver-disorder, Ionosphere, Pima, and Wdbc) were binary classification problems, while the others were multi-class classification problems. The regression datasets were selected from the repository of Luís Torgo.[2] The number of instances varied from 194 (Wpbc) to 10 000 (Calhousing and House8L), while the number of attributes varied from 8 (Bank8FM, Kin8NM, Puma8NH, Calhousing, and House8L) to 32 (Wpbc).

To quantify how missing data affected the prediction performance, we generated the following synthetic missing values. First, we selected 13 missing instance ratios: 1%, 3%, 5%, 7%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, and 50%. For each missing ratio, we generated 30 different missing datasets from the original complete dataset to ensure that the experimental results were statistically acceptable. For each missing dataset, the instances were selected randomly and their randomly selected attributes were marked as missing. The number of missing attributes for each instance was set randomly between one and half the total number of attributes ($1 \leq$ missing attributes $\leq 2/d$).

4.2. Imputation methods

A total of six imputation methods were used as benchmark methods: mean imputation (MEI), Hot-deck, k-NN, expectation conditional maximization (ECM), K-Means clustering (KMC), and mixture of Gaussians (MoG). The elimination of instances with missing values was used as the lower bound of the prediction performance, whereas the original complete dataset was used as the upper bound of the prediction performance. Therefore, a total of nine methods were compared in this experiment, including LLR imputation. The following is a brief description of each imputation method.

Table 1
Description of the classification datasets used in this study.

Id	Name	# Instances (n)	# Attributes (d)	# Classes (c)
1	Iris	150	4	3
2	Wine	178	13	3
3	Sonar	208	60	2
4	Glass	214	8	7
5	New-thyroid	215	5	3
6	Liver-disorder	345	6	2
7	Ionosphere	351	34	2
8	Wdbc	569	30	2
9	Pima	768	8	2
10	Vehicle	846	18	4
11	Vowel	990	10	11
12	Satellite	6465	36	6
13	Shuttle	14 429	9	3

Table 2
Description of the regression datasets used in this study.

Id	Name	# Instances (n)	# Attributes (d)
1	Wpbc	194	32
2	Stock	950	9
3	Abalone	4177	10
4	Compact	7909	12
5	Bank8FM	8078	8
6	Kin8NM	8192	8
7	Puma8NH	8192	8
8	Calhousing	10 000	8
9	House8L	10 000	8

- Mean imputation (MEI): MEI replaces the missing value with the mean of all known values of that attribute. Let $\mathbf{x}_i^j$ is the jth missing attribute of the ith instance, which is imputed by

$$\mathbf{x}_i^j = \sum_{k \in I(complete)} \frac{\mathbf{x}_k^j}{n_{|I(complete)|}}, \tag{7}$$

  where $I(complete)$ is a set of indices that are not missing in $\mathbf{x}_i$, and $n_{|I(complete)|}$ is the total number of instances where the jth attribute is not missing.
- Hot-deck: Hot-deck replaces the missing value with the value of the attribute with the most similar instance as follows:

$$\mathbf{x}_i^j = \mathbf{x}_k^j, \quad \text{where } k = \arg\min_p (\mathbf{x}_i^{I(complete)} - \mathbf{x}_p^{I(complete)})^2. \tag{8}$$

- k-NN: k-NN fills the missing value based on the values of the attributes of the k most similar instances,

$$\mathbf{x}_i^j = \sum_{p \in k-NN(\mathbf{x}_i)} K(\mathbf{x}_i^{I(complete)}, \mathbf{x}_p^{I(complete)}) \cdot \mathbf{x}_p^j, \tag{9}$$

  where k-NN($\mathbf{x}_i$) is the index set of the k nearest neighbors of $\mathbf{x}_i$ based on the non-missing attribute, and $K(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel

[1] http://mlearn.ics.uci.edu/MLRepository.html.
[2] http://www.liacc.up.pt/~ltorgo.

function that is proportional to the similarity between the two instances $\mathbf{x}_i$ and $\mathbf{x}_j$.

- Expectation conditional maximization (ECM): In the original dataset, $\mathbf{X}^{original}=(\mathbf{X}^{complete},\mathbf{X}^{missing})$, let $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the mean vector and the covariance matrix of the attributes, respectively, and ECM maximizes the log-likelihood in Eq. (10) by iteratively computing the likelihood using the current distribution parameters (*E-step*) and by updating the parameters (*CM-step*) based on the computed likelihood in *E-step*.

$$L_{complete}(\boldsymbol{\mu},\boldsymbol{\Sigma})=\log\int_{\mathbf{X}^{missing}} p(\mathbf{X}^{original};\boldsymbol{\mu},\boldsymbol{\Sigma})\,d\mathbf{X}^{missing}. \tag{10}$$

After determining the distribution parameters, $\mathbf{x}_i^j$ is imputed as follows:

$$\mathbf{x}_i^j=\boldsymbol{\mu}^j. \tag{11}$$

- Mixture of Gaussians (MoG): in MoG, the probability of the non-missing attributes of an instance is regarded as a linear combination of K individual components of a Gaussian distribution, as follows:

$$p(\mathbf{x}_i^{I(complete)})=\sum_{k=1}^{K}P(k)p_k(\mathbf{x}_i^{I(complete)}), \tag{12}$$

where K is the number of components in a mixture model, $P(k)$ is the prior probability of the instance being generated from the kth component, and $p_k(\mathbf{x}_i^{I(complete)})$ is the conditional probability of $\mathbf{x}_i^{I(complete)}$ for the kth component. Each component is assumed to be a Gaussian distribution, so the mean vector and covariance matrix of each component of $\mathbf{X}^{complete}$ and the prior probabilities can be estimated using the expectation-maximization (EM) algorithm. The missing attribute is imputed based on the weighted average of the mean value of the attribute, as follows:

$$\mathbf{x}_i^j=\sum_{k=1}^{K}P(k)\boldsymbol{\mu}_k^j, \tag{13}$$

where $\boldsymbol{\mu}_k^j$ is the mean value of the jth attribute of the kth component.

- K-Means clustering (KMC): the entire dataset is partitioned into K clusters by maximizing the homogeneity inside each cluster and the heterogeneity between clusters, as follows:

$$\arg\min_{\mathbf{C}^{I(complete)}}\sum_{i=1}^{K}\sum_{\mathbf{x}_j^{I(complete)}\in C_i^{I(complete)}}\|\mathbf{x}_j^{I(complete)}-\mathbf{c}_i^{I(complete)}\|^2, \tag{14}$$

where $\mathbf{c}_i^{I(complete)}$ is the centroid of $C_i^{I(complete)}$ and $\mathbf{C}^{I(complete)}$ is the union of all clusters ($\mathbf{C}^{I(complete)}=C_1^{I(complete)}\cup\cdots\cup C_K^{I(complete)}$). For a missing value $\mathbf{x}_i^j$, the mean value of the attribute for the instances in the same cluster with $\mathbf{x}_i^{I(complte)}$ is imputed,

$$\mathbf{x}_i^j=\frac{1}{|C_k^{I(complte)}|}\sum_{\mathbf{x}_p^{I(complete)}\in C_k^{I(complete)}}\mathbf{x}_p^j,$$
$$s.t.\ k=\arg\min_l|\mathbf{x}_j^{I(complete)}-\mathbf{c}_l^{I(complete)}|. \tag{15}$$

4.3. Learning algorithms

A total of five classification and five regression algorithms were used to test the imputation methods. Four algorithms that could handle both categorical and continuous targets were used for both classification and regression, i.e., k-nearest neighbor learning (k-NN), artificial neural networks (ANN), classification and regression tree (CART), and locally linear reconstruction (LLR). In addition, multinomial logistic regression (MNR) was employed for classification and multivariate linear regression (MLR) was used for regression. The following is a brief description of each learning algorithm.

- Multinomial logistic regression (MNR) [5]: MNR is a generalized version of logistic regression for multi-class classification. If there are c classes in a dataset, MNR runs $(c-1)$ independent binary logistic regression models where one class is used as a baseline and the other $(c-1)$ classes are regressed separately against the baseline class:

$$log\frac{P(y_i=j)}{P(y_i=j^*)}=\boldsymbol{\beta}_j^T\cdot\mathbf{x}_i,\quad j\neq j^*. \tag{16}$$

After the regression coefficients $\boldsymbol{\beta}_j$ have been estimated, the prediction for a new instance is made as follows:

$$\hat{y}_{n+1}=\arg\max_j P(y_{n+1}=j|\mathbf{x}_{n+1}). \tag{17}$$

- Multivariate linear regression (MLR) [23]: MLR fits the functional relationship between multiple input variables and single or multiple target variables of the given data in the form of linear equation. For a regression target y_i, the MLR equation that has d predictors with n training instances can be written as

$$y_i=\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\cdots+\beta_dx_{id},\quad \text{for } i=1,\ldots,n. \tag{18}$$

This can be rewritten in a matrix form such that $\mathbf{y}=\mathbf{X}\boldsymbol{\beta}$, and the coefficient $\boldsymbol{\beta}$ can be obtained explicitly by taking a derivative of the squared error function:

$$\min E(\boldsymbol{\beta})=\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}),$$
$$\frac{\partial E(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}=\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}-\mathbf{X}^T\mathbf{y}=0,$$
$$\boldsymbol{\beta}=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{19}$$

- k-Nearest neighbor regression (k-NN) [9]: k-NN finds its most similar k instances in the reference set using a distance metric, as in Eq. (2), and it combines their class memberships (classification) or target values (regression) to make a prediction:

$$\hat{y}_{n+1}=\arg\max_j\sum_{i\in k\text{-NN}(\mathbf{x}_{n+1}),\ y_i=j}w_i\ \text{(classification)}, \tag{20}$$

$$\hat{y}_{n+1}=\sum_{i\in k\text{-NN}(\mathbf{x}_{n+1})}w_i\cdot y_i\ \text{(regression)}. \tag{21}$$

Typically a suitable k is selected by 5-fold cross validation based on the original dataset, while the simple average is commonly used for weight assignment.

- Artificial neural networks (ANN) [4]: ANN is one of the most widely used machine learning algorithms and it is well-known for its high predictive power. A three-layer feed forward neural network was used in our experiments. In ANN, the targets are expressed as a combination of the input attribute values and their weights as follows:

$$y_k=\sum_{q=1}^{h}w_{kq}^{(2)}g\left(\sum_{r=1}^{d}w_{qr}^{(1)}x_r\right),\quad k=1,2,\ldots,d, \tag{22}$$

where $w_{kq}^{(2)}$, $w_{qr}^{(1)}$, and $g(\cdot)$ denote the weight connecting the kth output node and the qth hidden node, the weight connecting the qth hidden node and the rth input node, and the activation function, respectively. The number of output node $|y_k|$ is 1 for regression and c for classification. ANN training is equivalent to selecting the best number of hidden nodes and optimizing

the weights of the selected network structure, which is usually achieved with a family of gradient descent algorithms, such as the Quasi-Newton method [7].

- Classification and regression tree (CART): building a tree involves two main procedures in CART, i.e., recursive partitioning and pruning. During recursive partitioning, a tree grows until the information gain is not statistically significant. After the full tree has been constructed, pruning is performed to avoid over-fitting. The prediction of a new instance is made based on the leaf node to which it belongs. In our experiments, majority voting was used for classification while a simple average was used for regression.
- Locally linear reconstruction (LLR): because LLR was originally proposed for classification and regression tasks, it was also employed as a learning algorithm. The initial k was set to 50 in our experiments.

As performance measures, simple classification accuracy (ACC) was used for classification and the root mean squared error (RMSE) was used for regression. ACC is the fraction of the number of correctly classified instances among all test instances, which is computed as

$$ACC = \frac{1}{n}\sum_{i=1}^{n} I(\hat{y}_i = y_i), \tag{23}$$

where $I(\cdot)$ is an indicator function that returns 1 if the condition is true, but 0 otherwise. RMSE is the root of the average squared differences between the predicted and the original targets, as shown in the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}. \tag{24}$$

5. Experimental results

5.1. Classification results

Fig. 4 shows how the classification accuracy was affected by the existence of missing values. Note that the relative accuracy on the y-axis is a fraction of the two accuracies, i.e., the classification accuracy based on the missing data without imputation (discarding missing instances when training the classifiers) divided by the classification accuracy based on the original complete data. Each point in the graph is the average of five classification algorithms. As expected earlier, missing values generally reduced the classification accuracy, although the magnitude of the reduction varied among datasets. However, higher the missing ratios usually resulted in poorer classifier. If the missing ratio was very low, e.g., less than 3%, the performance degeneration was negligible. As the missing ratio increased, however, it affected the classifiers significantly and the performance degeneration was greater than 30% in over half of the datasets when the missing ratio was 50%. One might argue that we do not need to take missing values into consideration if a sufficient number of training instances are used. For this argument to hold, however, the performance degeneration should be reduced as well as the size of the datasets, whereas is was notable that the performance degeneration caused by missing values did not appear to be related to the dataset size. In Fig. 4, the worst affected dataset was Vowel, which was the third largest dataset in the experiments, whereas the least affected data set (Liver-disorder) was the sixth smallest dataset.

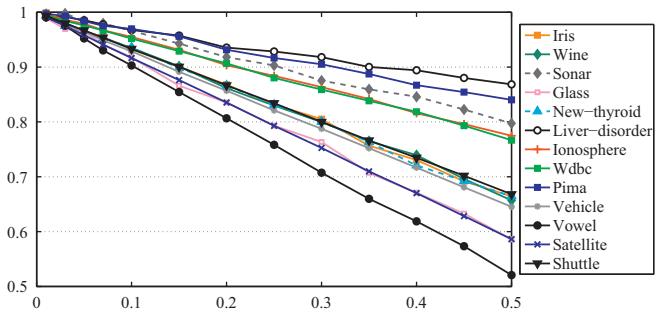

Fig. 4. Effects of missing values on the classification accuracy (the x-axis is the missing instance ratio and the y-axis is the relative prediction accuracy where the accuracy of the original complete dataset is set to 1).

Other large datasets, such as Shuttle and Satellite also appeared to be more affected than smaller datasets with all missing ratios.

Table 3 shows the accuracy improvements (%) with each imputation method versus discarding missing instances for various missing ratios using each classification algorithm. Note that the number in each set is the average accuracy improvement for 13 datasets with the corresponding classifier, imputation method, and missing ratio. Based on these results, we can make the following observations. First, irrespective of the classification algorithm, each imputation method was beneficial and all imputation methods outperformed discarding missing instances for all missing ratios, although the differences in the improvement with classifiers using the same imputation method were not statistically significant. Second, the effect of imputation methods increased with the missing ratio, i.e., the accuracy improvement was less than 1% with a missing ratio of 1%, but it was over 40% when the missing ratio was 50%. Third, LLR generally improved prediction accuracy more than the other methods, especially when the missing ratio was very high. With relatively small missing ratios, such as $1 \sim 10\%$, LLR was the most effective imputation method in 13 out of 25 cases of classifier-missing ratio pairs, while other imputation methods performed better in the other 12 cases. As the missing ratio increased beyond 10%, LLR was better for all classifier-missing ratio pairs.

A further comparison of the imputation methods is summarized in Table 4. This shows the number of dataset-classifier pairs where each imputation method produced the highest accuracy for each missing ratio. Note that there were 65 (13 datasets and five classifiers) possible pairs for each missing ratio. As explained earlier, LLR performed the best of the imputation methods, especially when the missing ratios were relatively high. k-NN was comparable to LLR when the missing ratios were low, i.e., less than 10%, whereas LLR was the most effective solution in most cases as the missing ratio increased. Given that LLR can be considered as a variation of k-NN method, it is worth noting that the classification accuracy in our experiment was proportional to the degree of locality of the imputation method. With the exception of Hot-deck, the most local imputation methods were k-NN and LLR, which were the two best methods for all missing ratios, followed by clustering-based imputation (KMC, MoG), which had an intermediate level of locality. Imputation methods with the lowest locality level, such as MEI and ECM, did not appear to be as effective as the other imputation methods.

The effectiveness of LLR imputation is also shown in Table 5, which compares the classification accuracy with LLR imputation and the original complete dataset using the same classifier.

Table 3
Average accuracy improvements (%) with each imputation method using 13 classification datasets versus discarding missing instances for various missing ratios using each classification algorithm.

Classifier	Imputation	1%	3%	5%	7%	10%	15%	20%	25%	30%	35%	40%	45%	50%
MNR	MEI	0.46	0.91	1.70	2.67	4.12	6.91	9.84	13.25	16.55	20.63	24.59	29.46	34.76
	Hot-deck	0.39	1.24	1.98	3.28	4.66	7.62	10.65	14.31	17.57	21.99	26.10	31.28	36.89
	k-NN	0.56	**1.72**	2.80	4.09	5.59	8.13	11.64	15.56	19.23	24.04	28.39	34.04	39.86
	ECM	0.45	0.98	1.68	2.69	4.06	6.80	9.92	13.33	16.40	20.67	24.58	29.25	34.33
	KMC	0.62	1.34	2.06	3.31	4.78	8.00	10.83	14.80	17.92	22.52	26.57	31.57	37.07
	MoG	0.41	1.32	2.19	3.31	4.72	7.87	10.95	14.70	17.87	22.47	26.55	31.47	37.13
	LLR	**0.71**	1.66	**2.86**	**4.11**	**5.64**	**9.27**	**12.75**	**16.81**	**20.41**	**25.52**	**29.92**	**35.55**	**41.48**
k-NN	MEI	0.54	1.29	2.34	3.36	5.08	7.72	10.70	14.46	17.49	21.94	25.88	30.49	35.93
	Hot-deck	0.44	1.38	2.63	3.70	5.41	8.23	11.52	15.14	18.75	23.57	27.35	32.72	38.22
	k-NN	0.61	**1.64**	2.80	3.97	**6.06**	7.93	11.20	15.55	19.03	24.09	28.46	33.75	39.91
	ECM	0.57	1.33	2.36	3.37	4.91	7.84	10.70	14.22	17.43	21.80	25.80	30.65	35.82
	KMC	0.64	1.55	2.69	3.50	5.47	8.34	11.41	15.32	18.67	23.29	27.24	32.41	38.11
	MoG	**0.69**	1.60	2.59	3.70	5.43	8.47	11.57	15.16	18.55	23.13	27.52	32.24	38.17
	LLR	0.56	1.62	**2.96**	**4.04**	5.79	**8.98**	**12.34**	**16.64**	**20.21**	**25.48**	**30.04**	**35.11**	**41.48**
CART	MEI	0.64	1.48	2.36	3.26	5.00	7.84	11.11	13.98	17.95	22.26	26.67	30.80	37.07
	Hot-deck	0.54	1.56	2.49	3.42	5.10	8.18	11.62	14.85	18.58	22.88	27.31	31.85	37.65
	k-NN	0.53	1.88	2.71	**4.09**	**6.04**	8.45	12.09	16.10	19.53	24.56	29.22	34.14	40.96
	ECM	0.48	1.50	2.42	3.14	4.78	7.98	10.89	14.30	17.55	22.25	26.68	30.80	36.84
	KMC	**0.79**	1.72	2.67	3.60	5.47	8.65	11.82	15.05	18.70	23.05	27.51	32.30	38.14
	MoG	0.58	1.74	2.60	3.35	5.65	8.29	11.89	15.30	18.66	23.06	27.67	32.06	38.22
	LLR	0.60	**1.95**	**2.73**	4.07	5.99	**9.51**	**13.32**	**16.97**	**20.73**	**25.43**	**30.39**	**35.49**	**42.01**
ANN	MEI	**0.86**	1.36	2.29	3.42	5.14	7.93	11.42	13.82	18.03	21.76	25.87	31.72	36.96
	Hot-deck	0.54	1.40	2.40	4.06	5.48	8.60	11.82	15.53	19.17	23.53	27.69	33.74	39.08
	k-NN	0.71	**1.86**	2.27	4.02	**6.59**	8.84	12.68	16.18	20.35	25.16	30.33	36.41	42.49
	ECM	0.39	1.15	2.26	3.33	4.88	8.13	11.04	14.08	18.08	21.35	26.67	31.09	36.77
	KMC	0.59	1.64	2.59	3.68	5.73	9.03	12.65	15.92	19.86	24.33	28.63	34.70	39.82
	MoG	0.63	1.39	2.68	4.14	6.40	8.81	12.55	15.77	19.71	23.86	28.83	34.29	39.98
	LLR	0.41	1.52	**2.90**	**4.25**	**6.59**	**9.63**	**13.64**	**17.19**	**21.83**	**26.70**	**31.71**	**37.96**	**44.09**
LLR	MEI	0.68	1.54	2.51	3.64	5.22	8.22	11.45	15.68	18.78	23.46	27.59	32.76	38.85
	Hot-deck	0.49	1.70	2.79	3.90	5.49	8.68	12.30	16.10	19.64	24.69	28.92	34.42	40.51
	k-NN	0.71	**1.86**	**2.95**	**4.42**	**6.21**	8.73	12.40	16.57	20.28	25.64	30.11	35.83	42.39
	ECM	0.49	1.58	2.35	3.55	5.09	8.11	11.81	15.56	18.72	23.32	27.59	32.86	38.67
	KMC	0.73	1.73	2.69	3.97	5.61	8.89	12.40	16.39	19.98	24.60	29.05	34.46	40.51
	MoG	0.76	1.74	2.76	3.95	5.64	8.86	12.45	16.23	19.92	24.52	29.15	34.51	40.65
	LLR	**0.80**	1.81	**2.95**	4.17	5.92	**9.55**	**13.32**	**17.56**	**21.34**	**26.56**	**31.36**	**37.32**	**43.71**

Table 4
The number of dataset-classifier pairs where each imputation method gave the highest accuracy with each missing ratio.

Imputation	1%	3%	5%	7%	10%	15%	20%	25%	30%	35%	40%	45%	50%
MEI	7	6	3	1	2	2	0	1	5	0	0	0	1
Hot-deck	2	5	6	5	4	3	3	2	1	1	3	4	1
k-NN	**16**	17	**23**	**27**	19	6	5	4	7	6	8	6	6
ECM	7	2	1	2	1	2	1	2	0	1	3	0	0
KMC	6	6	7	7	8	7	5	7	2	4	2	4	0
MoG	15	7	7	8	7	6	9	4	3	6	5	2	4
LLR	12	**22**	18	15	**24**	**39**	**42**	**45**	**47**	**47**	**44**	**49**	**53**

Table 5
Average classification performance decrease (%) of LLR imputation versus the 13 original complete datasets in each missing ratio.

Classifier	1%	3%	5%	7%	10%	15%	20%	25%	30%	35%	40%	45%	50%
MNR	−0.12	0.11	0.16	0.31	0.64	0.79	1.09	1.28	1.65	1.97	2.07	2.43	2.76
k-NN	0.05	0.16	0.13	0.39	0.68	0.94	1.40	1.42	1.85	1.97	2.32	2.83	3.27
CART	0.07	0.08	0.43	**0.33**	0.47	**0.63**	**0.85**	1.22	1.60	2.06	2.28	2.42	2.87
ANN	0.39	0.35	0.19	0.44	**0.16**	0.81	1.02	1.19	**1.20**	**1.54**	**1.74**	**1.90**	**2.21**
LLR	−**0.20**	**0.03**	**0.11**	0.38	0.50	0.70	1.09	**1.18**	1.54	1.88	2.12	2.48	3.02
Average	***0.04***	***0.15***	***0.20***	***0.37***	***0.49***	***0.77***	***1.09***	***1.26***	***1.57***	***1.88***	***2.11***	***2.41***	***2.83***

For example, the value of 0.07 for CART with a 1% missing ratio indicates that the accuracy of LLR was only 0.07% lower than that of the original complete dataset on average for the 13 datasets. Note that LLR imputation was better than the original complete dataset when the missing ratio was very low (see MNR and LLR with a 1% missing value). There was a possibility that the original value might be problematic if it was an outlier or a value that was inconsistent with other instances in the

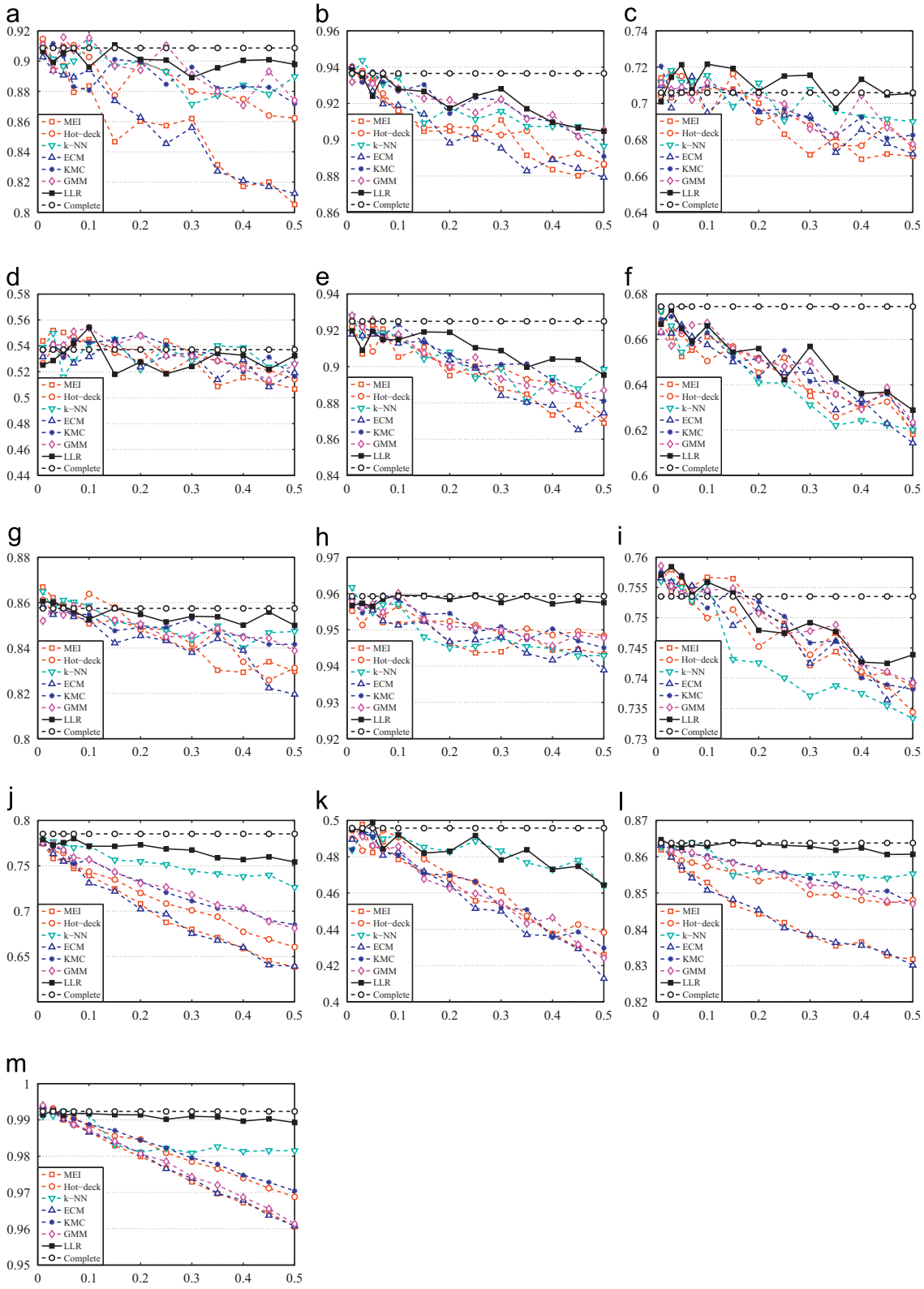

Fig. 5. Classification accuracy of each imputation method for various missing ratios with each dataset. (a) Iris, (b) Wine, (c) Sonar, (d) Glass, (e) New-thyroid, (f) Liver-disorder, (g) Ionosphere, (h) Wdbc, (i) Pima, (j) Vehicle, (k) Vowel, (l) Satellite, (m) Shuttle.

same class, but it was removed and replaced with a better value by LLR. It was remarkable that LLR imputation almost matched the accuracy of the original complete dataset not only when missing ratio was low, but also when missing ratio was high. When the missing ratio exceeded 10%, the classification accuracy only decreased by about 0.3% on average each time the missing ratio increased by 5%. Therefore, although half of the instances had at least one missing attribute, LLR was capable of handling those missing values so its accuracy decreased by at most 3.27% compared with the original complete dataset. In addition, ANN was enhanced by LLR imputation more than the other classifiers, i.e., its decrease in accuracy was only about 2% even when 50% of the instances had missing values.

Fig. 5 shows the classification accuracy of each imputation method with ANN for each dataset using various missing ratios, which illustrate the differences in the imputation methods in more detail than the summarized results. Note that the classification accuracy of MEI and ECM decreased almost linearly with the missing ratio for most datasets. However, the decrease in accuracy with LLR imputation appeared to be much less than that of the other methods. Surprisingly, it was still comparable with the accuracy of the original complete dataset even when the missing ratio was 50% in many datasets, e.g., Iris, Sonar, Glass, Ionosphere, Wdbc, Satellite, and Shuttle.

5.2. Regression results

Fig. 6 shows how the regression performance was affected by the existence of missing values. Note that the *y*-axis shows the RMSE increase as a percentage (%) of the original complete dataset, while each point in the graph is the average of five regression algorithms. Similar to the classification results, the missing values usually degraded the regression performance, although the degree of degradation varied among datasets. With regression, a higher missing ratio correlated with inferior performance. Unlike classification, the performance degeneration was noticeably different among the datasets. For some datasets, such as Wpbc and Puma8NH, it appeared that the performance was affected little by the missing values and their RMSEs were stable with various missing ratios. For other data sets, however, such as Bank8FM, Stock, and House8L, the missing values significantly degraded the model performance and their RMSEs were over three times higher than the complete datasets. The Compact dataset was an extreme case because its RMSE was seven times higher than that of the complete dataset.

Table 6 shows the RMSE improvements (%) with each imputation method versus discarding missing instances for various missing ratios using each regression algorithm. Note that the number in each cell is the average RMSE improvement (%) for nine datasets with the corresponding regression algorithm, imputation method and missing ratio, as shown in the following equation:

$$\text{RMSE improvement } (\%) = 100 \times \left(1 - \frac{\text{RMSE with imputation}}{\text{RMSE without imputation}}\right). \quad (25)$$

Unlike the classification results, LLR imputation clearly outperformed the other imputation methods for all regression algorithm-missing ratio pairs with less than 1% missing ratio, with the single exception of ANN. Some experimental

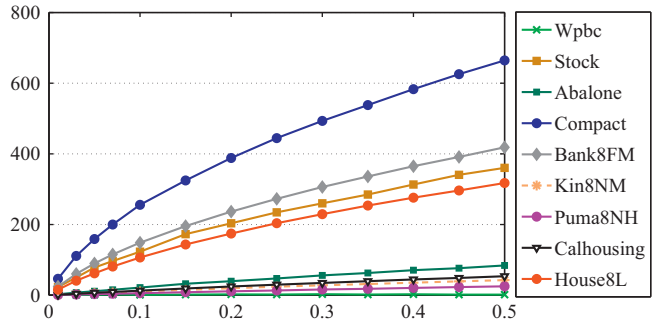

Fig. 6. The average RMSE increase (%) for five regression algorithms with each dataset using various missing ratios.

results were the same as the classification experiment. First, the RMSE improvement increased with the missing ratio, i.e., the RMSE improvement was around 10% with a missing ratio of 1% but it increased up to about 45% when the missing ratio was 50%. Second, the difference in the RMSE improvement with LLR compared with other benchmark methods increased with the size of the missing ratio for all regression algorithms.

Table 7 shows the number of dataset-regression algorithm pairs with each imputation method that produced the lowest RMSE for each missing ratio. Note that there were 45 (i.e., nine data sets and five classifiers) possible pairs for each missing ratio. This also confirmed the superiority of LLR imputation over the other benchmark methods. LLR was the most effective imputation method for all missing ratios and it produced the lowest RMSE of all the imputation methods, i.e., between 36% (16/45 with a 1% missing ratio) and 62% (28/45 with a 30% missing ratio) with all possible pairs. It was interesting that unlike classification, the simplest imputation method, MEI, was found to be slightly better than other simple imputation methods in some cases, i.e., it was the second most effective imputation method when the missing ratio was at its highest level (50%).

A further verification of LLR imputation is shown in Table 8, which summarizes the average RMSE increase with LLR imputation for the complete dataset with each regression algorithm-missing ratio pair. The RMSEs increased with the missing ratio, but the magnitude of the increases were noticeably reduced as the missing ratio increased, i.e., the RMSE difference was greater than half the missing ratio when the missing ratio was below 10%, whereas it reduced to less than one-third when the missing ratio approached 50%. This showed that LLR imputation was particularly effective in recovering the performance of MLR compared with the regression algorithms. The RMSE with LLR imputation was even lower than the complete dataset with a 1% missing ratio and its performance degeneration remained less than 10% up to a missing ratio of 50%.

Fig. 7 shows the RMSEs for each imputation method with MLR for each dataset using various missing ratios, which shows the differences among imputation methods in more detail compared with the summarized results. Based on the results, the datasets were divided into three groups. The first group contained most of the data sets, i.e., Stock, Abalone, Compact, Calhousing, and House8L where the RMSE with LLR imputation remained similar to that of the complete dataset or it increased slightly with the missing ratio, whereas the RMSEs with other imputation methods increased significantly. As a consequence, the RMSE gap between LLR imputation and other methods was very large with high missing ratios such as 50%. The second group included Bank8FM, Kim8NM, and Puma8NH where LLR imputation was still effective but other methods were just as effective as LLR imputation. Wpbc

Table 6
The average RMSE improvements (%) with each imputation method for nine datasets versus discarding missing instances for various missing ratios using each regression algorithm.

Regression	Imputation	1%	3%	5%	7%	10%	15%	20%	25%	30%	35%	40%	45%	50%
MLR	MEI	6.67	13.17	16.89	19.99	22.98	26.62	28.97	31.22	32.93	34.26	35.61	36.75	37.79
	Hot-deck	6.60	13.23	17.21	20.02	23.31	26.90	29.54	31.69	33.24	34.76	36.18	37.30	38.38
	k-NN	6.79	13.44	17.26	20.09	23.31	26.94	29.59	31.72	33.29	34.74	36.23	37.34	38.27
	ECM	6.63	13.06	17.10	19.86	23.00	26.64	29.02	31.24	32.83	34.29	35.72	36.78	37.78
	KMC	6.92	13.77	17.91	20.88	24.25	28.31	30.94	33.27	34.96	36.53	37.94	39.31	40.19
	MoG	7.10	13.68	17.83	20.67	24.11	28.34	30.74	33.22	34.93	36.44	37.80	39.19	40.09
	LLR	**7.14**	**14.22**	**18.35**	**21.56**	**25.01**	**29.36**	**32.11**	**34.63**	**36.46**	**38.11**	**39.69**	**41.03**	**41.95**
k-NN	MEI	7.27	13.19	16.41	18.26	20.46	23.22	24.79	25.72	26.81	27.23	28.11	28.56	28.99
	Hot-deck	7.57	13.79	17.03	19.03	21.23	23.80	25.33	26.39	27.33	27.71	28.55	28.77	29.48
	k-NN	7.68	13.82	17.10	19.18	21.40	23.69	25.32	26.39	27.16	27.94	28.78	28.88	29.65
	ECM	7.27	13.31	16.17	18.15	20.57	23.23	24.78	25.61	26.79	27.34	28.16	28.37	29.03
	KMC	7.74	14.19	17.66	19.98	22.34	25.04	26.44	27.74	28.47	29.07	29.89	30.15	30.70
	MoG	7.74	14.28	17.64	19.98	22.39	24.97	26.64	27.71	28.87	29.22	29.99	30.20	30.88
	LLR	**8.17**	**15.08**	**18.92**	**21.43**	**24.01**	**26.85**	**28.84**	**29.99**	**31.05**	**31.66**	**32.68**	**32.69**	**33.28**
CART	MEI	6.43	12.42	16.51	18.73	21.52	24.74	26.80	28.41	29.95	30.69	31.78	32.64	33.08
	Hot-deck	6.66	12.28	16.54	18.06	20.53	23.33	25.26	26.52	27.33	28.37	29.01	29.57	30.01
	k-NN	6.33	11.95	16.13	18.40	20.40	23.37	25.19	26.36	27.65	28.43	29.25	29.77	30.03
	ECM	6.51	12.44	16.61	18.95	21.61	24.55	26.82	28.61	29.94	30.77	31.85	32.55	33.18
	KMC	7.34	13.67	18.07	20.57	22.92	26.15	27.74	29.28	30.43	31.43	32.28	32.84	33.47
	MoG	7.49	13.79	17.83	20.41	22.82	26.06	27.83	29.08	30.43	31.49	32.48	32.95	33.27
	LLR	**7.85**	**14.76**	**18.88**	**21.85**	**24.47**	**27.96**	**30.16**	**31.61**	**32.98**	**33.69**	**34.66**	**35.43**	**35.83**
ANN	MEI	9.14	16.36	21.18	23.67	26.86	29.78	32.94	35.03	36.11	37.35	39.36	40.14	41.12
	Hot-deck	9.31	16.96	21.08	23.29	25.99	29.20	32.96	34.57	36.83	37.27	38.61	38.66	40.83
	k-NN	9.08	16.81	20.96	22.65	26.05	30.43	32.77	34.75	36.35	37.55	39.34	39.63	39.17
	ECM	9.21	15.63	21.06	23.56	25.84	29.95	32.87	34.13	36.40	37.35	38.82	39.59	41.18
	KMC	**10.19**	17.88	23.12	25.50	28.95	33.05	35.19	37.74	39.24	40.02	41.65	42.47	43.83
	MoG	8.95	17.61	23.01	25.99	29.01	33.05	35.59	37.67	39.23	39.91	41.89	42.28	43.70
	LLR	10.03	**19.00**	**23.67**	**27.27**	**30.71**	**35.62**	**37.70**	**39.88**	**42.00**	**42.73**	**44.52**	**45.68**	**46.49**
LLR	MEI	9.44	17.24	20.98	23.38	26.65	30.59	32.96	34.81	36.45	37.50	38.73	39.60	40.29
	Hot-deck	9.61	17.39	21.67	23.62	26.50	30.17	33.02	34.28	36.23	37.37	38.16	38.97	39.45
	k-NN	9.81	17.66	21.43	23.66	26.54	30.21	32.97	34.37	36.34	37.32	38.36	39.01	39.61
	ECM	9.37	17.06	20.94	22.78	26.55	30.41	32.63	35.00	36.41	37.62	38.77	39.29	40.04
	KMC	11.10	19.42	23.56	26.24	29.27	32.67	34.86	36.66	38.38	39.23	40.40	41.17	41.98
	MoG	11.30	19.31	23.56	26.11	28.98	32.93	34.91	36.69	38.40	39.23	40.63	41.23	41.78
	LLR	**11.82**	**21.08**	**25.87**	**28.96**	**32.08**	**35.94**	**38.59**	**40.18**	**41.93**	**43.03**	**43.93**	**44.61**	**44.97**

Table 7
The number of dataset-regression algorithm pairs with each imputation method that produced the lowest RMSE for each missing ratio.

Imputation	1%	3%	5%	7%	10%	15%	20%	25%	30%	35%	40%	45%	50%
MEI	6	7	8	7	2	3	8	5	3	2	7	6	9
Hot-deck	5	2	1	1	0	1	0	0	1	1	0	0	1
k-NN	3	0	1	1	0	0	2	0	0	1	3	0	1
ECM	6	3	4	3	0	5	5	4	4	9	5	4	3
KMC	6	6	6	2	10	6	2	4	5	1	2	2	2
MoG	3	3	2	8	6	5	2	5	4	5	4	6	4
LLR	**16**	**24**	**23**	**23**	**27**	**25**	**26**	**27**	**28**	**26**	**24**	**27**	**25**

formed the final group where the RMSEs with any imputation method were even lower than those of the complete dataset. As explained in the previous section, it is possible that the original dataset was too noisy so the generation of synthetic missing values and filling other values may have improved the prediction performance.

The experimental results of missing value imputation in classification and regression are summarized as follows. First, the proportion of missing data directly affected the prediction accuracy regardless of the learning task; the higher the missing ratio, the lower the accuracy. Second, all imputation methods recovered the model performance to a certain degree, so it was better to use even a simple imputation method rather than discarding instances with missing attributes. Third, LLR imputation outperformed the other benchmark methods in many respects. In classification, the average accuracy improvement with an incomplete dataset was the highest of all classifiers when the missing ratio was relatively high ($\geq 15\%$), whereas it was comparable with *k*-NN imputation and outperformed the others when the missing ratio was relatively low ($\leq 15\%$). In regression, the average RMSEs improvement for the incomplete dataset were the highest with LLR for all regression-missing ratio pairs with the exception of only one case. Finally, the prediction performance with LLR imputation was similar to the complete dataset. In classification, the average accuracy was only 3% lower than that of the complete dataset even with a 50% missing ratio. In regression, the performance recovery was not as dramatic as that in classification but it was still impressive because the average RMSE with LLR imputation was less than 16% even with a 50% missing ratio.

Table 8
Average RMSE increase (%) with LLR imputation versus nine complete datasets with each missing ratio.

Regression	1%	3%	5%	7%	10%	15%	20%	25%	30%	35%	40%	45%	50%
MLR	−0.04	0.34	1.08	1.37	2.08	2.97	4.04	4.65	5.35	5.97	6.60	7.05	7.92
k-NN	1.10	2.28	3.25	4.43	5.91	8.23	9.73	11.67	13.38	14.61	15.78	17.40	18.65
CART	0.68	1.95	3.51	4.16	5.99	7.87	9.48	11.18	12.48	13.82	15.17	16.18	17.37
ANN	1.18	1.93	3.57	4.07	5.43	7.05	9.67	10.67	12.03	13.18	14.44	14.98	16.57
LLR	1.35	2.34	3.76	4.88	6.61	9.06	11.00	13.04	14.36	15.81	17.38	18.75	20.45
Average	***0.85***	***1.77***	***3.03***	***3.78***	***5.20***	***7.04***	***8.78***	***10.24***	***11.52***	***12.68***	***13.87***	***14.87***	***16.19***

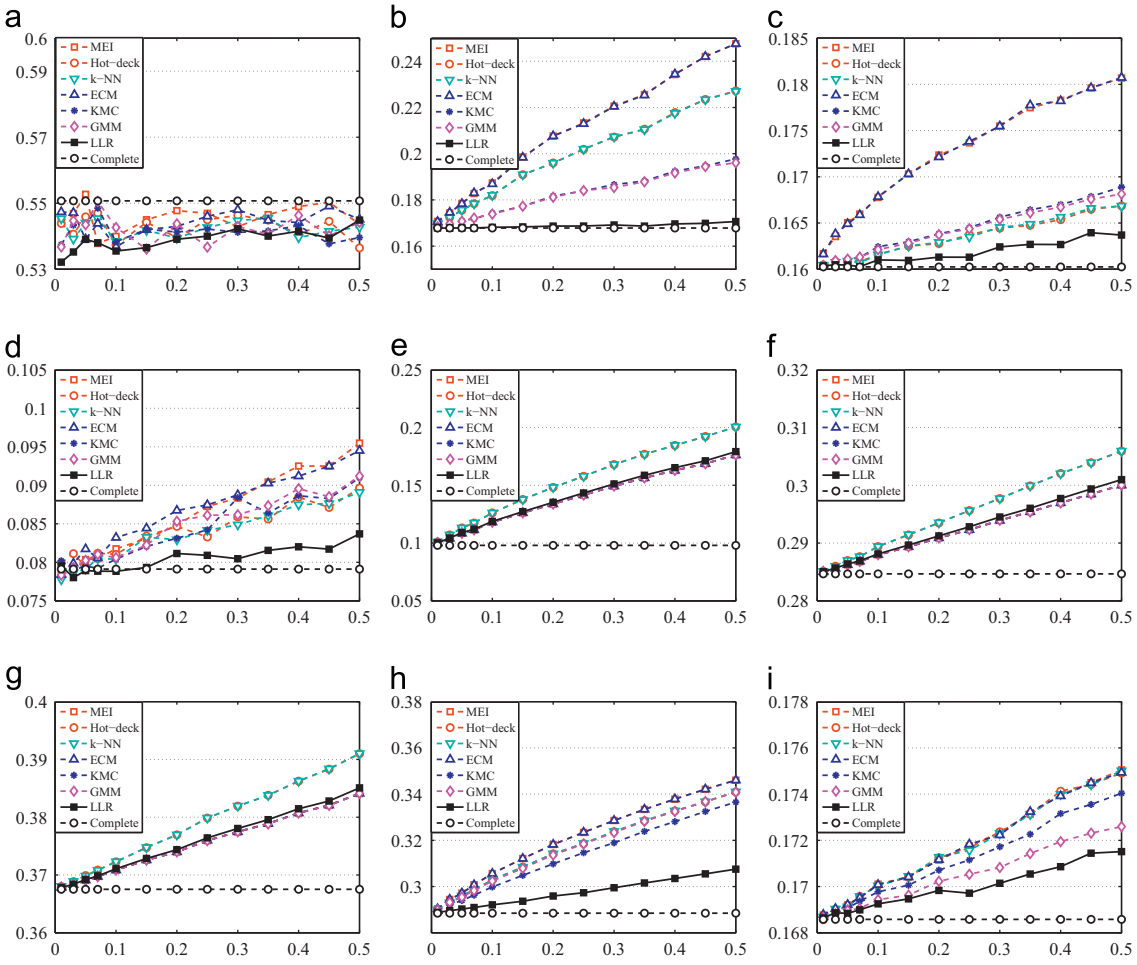

Fig. 7. MLR RMSEs for each imputation method with various missing ratios for each dataset. (a) Wpbc, (b) Stock, (c) Abalone, (d) Compact, (e) Bank8FM, (f) Kin8NM, (g) Puma8NH, (h) Calhousing, (i) House8L.

6. Conclusion

In this paper, we propose a new single imputation method based on locally linear reconstruction (LLR). Experiments verified that increasing the missing ratio severely degraded the performance of all learning algorithms, which confirmed that missing values should be treated with caution. After taking the local structure into account, the proposed LLR imputation could handle missing values more effectively than other parametric or global structure-based imputation methods. Carefully designed extensive experiments showed that LLR imputation usually outperformed well-known benchmark imputation methods. In classification, it was most effective when the missing ratio was relatively high because its prediction accuracy was similar to that of the complete dataset. In regression, LLR performed the best with all learning algorithm-missing ratio pairs, with only one exception.

Apart from some noticeable experimental results mentioned above, there were a few limitations in the current study that suggest further directions for research. First, only datasets containing numerical attributes were selected in our experiment to ensure a fair comparison because some benchmark imputation methods could not handle categorical attributes. Since LLR can deal with both numerical and categorical attributes, further comparative study should be conducted that is based on categorical or mixed attributed datasets. In addition, we generated synthetic missing values to systematically evaluate the effect of missing values and we compared various imputation methods in a wide range of possible environments in this study. A natural next step is to verify LLR imputation using real datasets that inherently contain missing values.

Acknowledgments

The work was supported by the research program funded by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST) (No. 2011-0021893).

References

[1] E. Acuna, C. Rodriguez, The treatment of missing values and its effect in the classifier accuracy, in: D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds.), Classification, Clustering and Data Mining ApplicationsSpringer, 2004, pp. 639–648.
[2] G. Batista, M. Monard, An analysis of four missing data treatment methods for supervosed learning, Appl. Artif. Intell. 17 (5–6) (2003) 519–533.
[3] J. Bernard, X. Meng, Applications of multiple imputation in medical studies: From AIDS to NHANES, Stat. Methods Med. Res. 8 (1) (1999) 17–36.
[4] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2002.
[5] D. Böhning, Multinomial logistic regression algorithm, Ann. Inst. Stat. Math. 44 (1) (1992) 197–200.
[6] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, CRC Press, Boca Raton, FL, USA, 1984.
[7] C. Broyden, Quasi-Newton methods and their application to functional minimisation, Math. Comput. 21 (99) (1967) 368–381.
[8] P. Clark, T. Niblett, The CN2 induction algorithm, Mach. Learn. 3 (4) (1988) 261–283.
[9] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[10] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incompleter data via the EM algorithm, J. R. Stat. Soc.: Ser. B 39 (1) (1977) 1–37.
[11] Y. Ding, A. Ross, A comparison of imputation methods for handling missing scores in biometric fusion, Pattern Recognition 45 (3) (2012) 919–933.
[12] C. Ennett, M. Frize, C. Walker, Imputation of missing values by integrating neural networks and case-based reasoning, in: Proceedings of the 30th Annual International IEEE EMBS Conference (EMBS'08), Vancouver, BC, Canada, 2008, pp. 4337–4341.
[13] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognition 41 (12) (2008) 3692–3705.
[14] A. Farhangfar, L. Kurgan, W. Pedrycz, A novel framework for imputation of missing values in database, IEEE Trans. Syst. Man Cybern. A: Syst. Hum. 37 (5) (2007) 692–709.
[15] P. Garcia-Laencina, J.-L. Sancho-Gomez, A. Rigueiras-Vidal, M. Verleysen, K-nearest neighbours with mutual information for simultaneous classification and missing data imputation, Neurocomputing 72 (7–9) (2009) 1483–1493.
[16] Z. Ghahramani, M. Jordan, Supervised learning from incomplete data via an EM approach, in: Advances in NIPS 6, Morgan Kaufmann, Los Altos, CA, USA, 1994, pp. 120–127.
[17] W. Grzymala-Busse, A new version of the rule induction system LERS, Fundam. Inf. 31 (1) (1997) 27–39.
[18] W. Grzymala-Busse, M. Hu, A comparison of several approaches to missing attribute values in data mining, in: Lecture Note in Artificial Intelligence 2005, 2001, pp. 378–385.
[19] Y. He, M. YousuffHussaini, J. Ma, BehrangShafei, G. Steidl, A new fuzzy c-means method with total variation regularization for segmentation of images with noisy and incomplete data, Pattern Recognition 45 (9) (2012) 3463–3471.
[20] K. Hron, M. Templ, P. Filzmoser, Imputation of missing values for compositional data using classical and robust methods, Comput. Stat. Data Anal. 54 (12) (2010) 3095–3107.
[21] J. Jerez, I. Molina, G. Garcia-Laencina, E. Alba, N.R.M. Martin, L. Franco, Missing data imputation using statistical and machine learning methods in a real breast cancer problem, Artif. Intell. Med. 50 (2) (2010) 105–115.
[22] W.-B. Jerzy, M. Hu, A comparison of several approaches to missing attribute values in data mining, in: Proceedings of the 2nd International Conference on Rough Sets and Current Trends in Computing (RSCTC00), Banff, Alberta, Canada, 2000, pp. 378–385.
[23] R. Johnson, D. Wichern, Applied Multivariate Statistical Analysis, Prentice Hall, Upper Saddle River, NJ, USA, 2002.
[24] P. Kang, S. Cho, Locally linear reconstruction for instance-based learning, Pattern Recognition 41 (11) (2008) 3507–3518.
[25] S.-W. Kim, B.J. Oommen, On using prototypereduction schemes to optimize locally linear reconstruction methods, Pattern Recognition 45 (1) (2012) 498–511.
[26] R. Kohavi, B. Becker, D. Sommerfield, Improving simple Bayes, in: Proceedings of the European Conference on Machine Learning (ECML97), 1997, Prague, Czech Republic.
[27] H. Li, X. Zhou, Y. Yao, Missing values imputation hypothesis: An experimental evaluation, in: Proceedings of the 8th IEEE International Conference on Cognitive Informatics (ICCI09), Hong Kong, China, 2009, pp. 275–280.
[28] Z. Liao, X. Lu, T. Yang, H. Wang, Missing data imputation: A fuzzy K-means clustering algorithm over sliding window, in: Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'09). Tianjin, China, 2009, pp. 133–137.
[29] R. Little, D. Rubin, Statistical Analysis with Missing Data, Wiley and Sons, New York, NY, USA, 1987.
[30] P. Meesad, K. Hengpraprohm, Combining of KNN-based feature selection and KNN-based missing value imputation of microarray data, in: Proceedings of the 3rd IEEE International Conference on Innovative Computing Information and Control (ICICIC08), Hong Kong, China, Dalian, China, 2008, pp.
[31] X.-L. Meng, D. Rubin, Maximum likelihood estimation via the ECM algorithm, Biometrika 80 (2) (1993) 267–278.
[32] M. Ouyang, W. Welsh, P. Georgopoulos, Gaussain mixture clustering and imputation of microarray data, Bioinformatics 20 (6) (2004) 917–923.
[33] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann San Francisco, CA, USA, 1993.
[34] P. Royston, Multiple imputation of missing values, Stata J. 4 (3) (2004) 227–241.
[35] J. Sexton, A. Swensen, ECM algorithms that converge at the rate of EM, Biometrika 87 (3) (2000) 651–662.
[36] J. Shafer, Analysis of Incomplete Multivariate Data, Chapman and Hall, London, UK, 1997.
[37] X. Su, R. Greiner, T. Khoshgoftaar, A. Napolitano, Using classifier-based nominal imputation to improve machine learning, in: Lecture Note in Computer Science, vol. 6634, 2011, pp. 124–135.
[38] X. Su, T. Khoshgoftaar, R. Greiner. Using imputation techniques to help learn accurate classifiers, in: Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI08), Dayton, OH, USA, 2008, pp. 437–444.
[39] S. vanBuuren, K. Groothuis-Oudshoorn, MICE: Multivariate imputation by chained equations in R, J. Stat. Software 45 (3) (2011).
[40] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Faufmann, Burlington, MA, USA, 2005.
[41] T. Yu, H. Peng, W. Sun, Incorporating nonlinear relationships in microarray missing value imputation, IEEE ACM Trans. Comput. Biol. Bioinformatics 8 (3) (2011) 723–731.
[42] C. Zhang, Y. Qin, X. Zhu, J. Zhang, S. Zhang, Clustering-based missing value imputation for data preprocessing, in: Proceedings of the IEEE International Conference on Industrial Informatics (INDIN'06), Singapore, 2006, pp. 1081–1086.
[43] P. Zhang, Multiple imputation: theory and method, Int. Stat. Rev. 71 (3) (2003) 581–592.
[44] S. Zhang, J. Zhang, X. Zhu, Y. Qin, C. Zhang, Missing value imputation based on data clustering, in: Lecture Notes in Computer Science, vol. 4750, 2008, pp. 128–138.
[45] Y. Zhang, Y. Liu, Data imputation using least squares support vector machines in urban arterial street, IEEE Signal Process. Lett. 15 (5) (2009) 414–417.
[46] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, Z. Xu, Missing value estimation for mixed-attribute data sets, IEEE Trans. Knowl. Data Eng. 23 (1) (2011) 110–121.

Pilsung Kang is an assistant professor in Department of Industrial and Information Systems Engineering, College of Business and Technology, Seoul National University of Science and Technology (SeoultTECH), Seoul, South Korea. He received B.S. and Ph.D. in Industrial Engineering, College of Engineering, Seoul National University, Seoul, South Korea. His research interests include instance-based learning, learning kernel machines, novelty detection, learning algorithms in class imbalance, and non-linear dimensionality reduction. He is also interested in a wide range of applications such as keystroke dynamics-based authentication, fault detection in manufacturing process, and customer relationship management. He has published a number of papers on related topics in international journals and conferences.