Knowl Inf Syst (2016) 46:389–422
DOI 10.1007/s10115-015-0822-y

REGULAR PAPER

Missing value imputation using a fuzzy clustering-based EM approach

Md. Geaur Rahman · Md Zahidul Islam

Received: 12 July 2013 / Revised: 17 October 2014 / Accepted: 31 January 2015 /
Published online: 25 February 2015

Abstract Data preprocessing and cleansing play a vital role in data mining by ensuring good quality of data. Data-cleansing tasks include imputation of missing values, identification of outliers, and identification and correction of noisy data. In this paper, we present a novel technique called *A Fuzzy Expectation Maximization and Fuzzy Clustering-based Missing Value Imputation Framework for Data Pre-processing* (FEMI). It imputes numerical and categorical missing values by making an educated guess based on records that are similar to the record having a missing value. While identifying a group of similar records and making a guess based on the group, it applies a fuzzy clustering approach and our novel fuzzy expectation maximization algorithm. We evaluate FEMI on eight publicly available natural data sets by comparing its performance with the performance of five high-quality existing techniques, namely EMI, GkNN, FKMI, SVR and IBLLS. We use thirty-two types (patterns) of missing values for each data set. Two evaluation criteria namely root mean squared error and mean absolute error are used. Our experimental results indicate (according to a confidence interval and t test analysis) that FEMI performs significantly better than EMI, GkNN, FKMI, SVR, and IBLLS.

Keywords Data preprocessing · Data cleansing · Data quality · Missing value imputation · Fuzzy clustering

Md. G. Rahman · M. Z. Islam (✉)
Center for Research in Complex Systems (CRiCS), School of Computing and
Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia
e-mail: zislam@csu.edu.au
URL: http://csusap.csu.edu.au/∼zislam/

Md. G. Rahman
e-mail: grahman@csu.edu.au
URL: http://www.gea.bau.edu.bd

1 Introduction

Data collection, storage, and analysis have become crucial nowadays for various decision-making processes of modern organizations. Typically various sources and techniques (including surveys, interviews, and sensors) are used for data collection, and the collected data are then generally integrated into a single data set for data mining purposes [23]. For example, different types of sensors such as weather stations are typically used to collect temperature, humidity, and wind speed data in a habitat monitoring system (HMS). Various factors including human error and misunderstanding, equipment malfunctioning, and faulty data transmission can cause data corruption and missing during the whole process of data collection, storage, and preparation. Approximately 5 % or more data values can often be lost (missing) unless extreme care is taken by the organizations [32,35,40,48].

In this study, we consider that a data set D_F is a two-dimensional table, where rows represent the records ($R = \{R_1, R_2, \ldots, R_n\}$) and columns represent the attributes ($A = \{A_1, A_2, \ldots A_M\}$). A record $R_x \in D_F$ has a set of M attributes $A = \{A_1, A_2, \ldots, A_M\}$ (i.e., $|A| = M$). It represents an individual such as a patient in the patient data set of a hospital. An attribute A_i represents an information such as the age of the records. Each attribute A_i has a domain. If A_i is a categorical attribute, then the domain of A_i is $A_i = \{A_{i1}, A_{i2}, \ldots, A_{ij}\}$, and if A_i is a numerical attribute, then $A_i = [low, up]$, where the lowest limit of A_i is *low* and the highest limit of A_i is *up*. By R_{xi}, we mean the value of the ith attribute of the xth record, and by a missing value of R_{xi}, we mean that the value R_{xi} is missing. A data set D_F may have some missing values in it. Note that in this study, we do not consider time series data.

Use of poor-quality data, having missing and incorrect values, can result in an inaccurate and non-sensible conclusion, making the whole process of data collection and analysis useless for the users [16,36]. Therefore, in order to deal with the inaccurate and missing values, it is extremely important to have an effective data preprocessing framework [8,17,28,29]. One important data preprocessing task is the imputation/estimation of missing values as accurately as possible. A number of imputation methods have been proposed recently [12,14,22,31,37–39,43,52,54,55].

The imputation performance generally depends on the selection of a suitable technique [51]. Different techniques take different approaches for imputation of a missing value. While imputing a missing value, some techniques including EMI [43] and Mean Imputation [22] use the whole data set D_F. On the other hand, some techniques including DMI [35], KMI [31], LLSI [25], ILLSI [9], and IBLLS [12] use only a portion (i.e., a horizontal segment) $D_a \subset D_F$ where the records $R_x \in D_a; \forall x$ are similar to the record R_y having a missing value.

In this paper, we propose a novel imputation technique called *A Fuzzy Expectation Maximization and Fuzzy Clustering-based Missing Value Imputation Framework for Data Preprocessing* (FEMI). The basic idea of the technique is to impute/estimate a missing value R_{xi} of a record R_x using the records that are similar to R_x. Our technique considers the fuzzy nature of a data set, where a record R_x has an association (i.e., a membership degree) with each cluster instead of one and only one cluster. A cluster is a portion D_a containing a group of records. Similar records are grouped together in a cluster, and dissimilar records are placed in different clusters.

FEMI uses two levels of fuzziness. In the first level, it considers that the record R_x (having a missing value R_{xi}) has a fuzzy nature in the sense that it has membership degrees U_{xl} with all clusters $C_l; \forall l$, instead of R_x having a complete (100 %) association with one cluster C_l and zero association with all other clusters $C_j; \forall j \neq l$. The missing value of R_x is estimated by considering each cluster C_l separately. Therefore, if there are k numbers of clusters, we get k numbers of imputed values for R_{xi}. A final imputation is then computed through the weighted average of all k imputed values, by using the imputed values $(v_1, v_2, \ldots, v_k)$ and the membership degrees $(U_{x1}, U_{x2}, \ldots, U_{xk})$.

In the second level of fuzziness, FEMI considers that all records $R_y \in D_F; \forall y$ (not just R_x) have a fuzzy association with all the k clusters. Therefore, while imputing the missing value R_{xi} according to a cluster C_l, FEMI uses a novel fuzzy imputation technique (as explained in Eqs. (6), (7) and (8)) that considers this second level of fuzziness. The fuzzy imputation technique considers all records $R_y; \forall y$ and their membership degrees $U_{yl}; \forall y$ with C_l. Note that all records $R_y; \forall y$ are used for all clusters $C_l; \forall l$, but each cluster produces an imputed value that is different to the imputed values produced by other clusters due to the difference in membership degrees of the records with the clusters.

The main contributions of the technique are as follows: 1. The overall framework for imputing the missing values, 2. The use of multiple clusters $C_l; \forall l$ for imputing a missing value R_{xi}, 3. Combining the imputed values $(v_1, v_2, \ldots, v_k)$ considering the membership degrees $(U_{x1}, U_{x2}, \ldots, U_{xk})$ of the record R_x (that has the missing value R_{xi}), and 4. Imputing the missing value R_{xi} using a fuzzy imputation technique [see Eqs. (6), (7), and (8)] applied on each cluster.

We evaluate FEMI on eight (8) natural data sets (available from UCI Machine Learning Repository [15]) by comparing its performance with the performance of five high-quality existing techniques, namely EMI [22,43], GkNN [53], FKMI [27,31], SVR [44,49] and IBLLS [12], which have been argued to be better than many other existing techniques including Bayesian principal component analysis (BPCA) [33], LLSI [25], and ILLSI [9]. Two evaluation criteria such as root mean squared error (RMSE) and mean absolute error (MAE) are used. Our experimental results indicate (based on some statistical analysis namely t test and confidence interval) that FEMI performs significantly better than EMI, GkNN, FKMI, SVR, and IBLLS.

The organization of the paper is as follows. Section 2 presents a literature review. Our technique (FEMI) is presented in Sect. 3. Section 4 presents experimental results, and Sect. 5 gives concluding remarks.

2 Background study

A number of missing value imputation techniques have recently been proposed [12,14,22,30,31,43,52,54,55]. A few of the existing techniques are nearest neighbor (NN), linear interpolation (LIN), cubic spline interpolation, regression-based expectation maximization (REGEM) imputation, self-organizing map (SOM), and multilayer perceptron (MLP) [22]. However, many existing techniques cannot handle a data set having both numerical and categorical attributes [47].

The mean of all values of an attribute is used to impute a missing value of the attribute by a relatively simple technique [43]. However, a more advanced technique called k-nearest neighbor imputation (kNNI) [4] first finds the k-most similar records from the data set by using Euclidean distance measure. It then imputes a missing categorical value belonging to an attribute by using the most frequent value, of the same attribute, within the k-nearest neighbor (k-NN) records. For imputing a numerical value, the technique utilizes the attribute mean value for the k-NN records. The experimental results show that the performance of kNNI technique is higher than the performance of a technique using mean/mode imputation on a whole data set, instead of the horizontal segment having k-NN records. It is a simple technique in a way since it does not require to create explicit models such as decision trees and forests. However, it needs to search the whole data set as many times as the number of records having missing values in order to find the nearest neighbors of each record having missing value/s. Therefore, the technique can be found expensive for a large data set [4,50].

Instead of repeatedly finding k-NN for each record having a missing value, another existing technique called "K-means Clustering-based Imputation (KMI)" [27,31] uses the clusters of records for imputation. It first divides a given data set D_F into two data sets namely D_C and D_I, where D_C contains complete (no missing) records and D_I contains records having missing values. KMI then partitions the data set D_C into k (user-defined) clusters using a well-known K-Means clustering approach. It then assigns a record belonging to D_I into a cluster, which it has the minimum distance with. The missing value is then imputed following an educated guess based on the records of the cluster. The clustering is only done once for imputing all records belonging to D_I.

In order to impute numerical missing values of a record R_i, a recent technique called "Iterative Bi-Cluster based Local Least Square Imputation" (IBLLS) [12] divides a data set in both horizontal and vertical segments. It first finds the k-most similar records for R_i. Within the k-most similar records, it then calculates the correlation information, Q, between the attributes, the values of which are available in R_i, and the attributes the values of which are missing in R_i. IBLLS then uses the correlation information, Q, in order to re-calculate the k-most similar records for R_i.

IBLLS then detects the attributes (within the k-most similar records) that have high correlations with the attribute having a missing value in R_i. It thus partitions a data set horizontally (using the k-most similar records) and vertically using the most correlated attributes. IBLLS then uses Local Least Square Framework [25] into the partition in order to impute a missing value of R_i. IBLLS repeats this procedure for imputing other missing values (if any) of R_i. Similarly, IBLLS imputes other records of the data set having missing values.

IBLLS is an iterative method. In each iteration, it checks how well the imputed value agrees with the correlation matrix of the attributes within the data segment that is partitioned both horizontally and vertically. If the imputed value of the current iteration agrees better than the imputed value of the previous iteration, then it replaces the previous imputed value by the new imputed value; otherwise, it keeps the previous imputed value. The process of imputation and updating missing values continues recursively until the change of degrees of agreement (between an imputed value and the correlation matrix) for two consecutive iterations goes under a user-defined threshold.

Another iterative method called Expectation Maximization Imputation (EMI) [22,43] relies on the basic concept of the well-known expectation maximization (EM) algorithm [6, 14]. The EM algorithm uses two main steps known as the expectation step (E-Step) and the maximization step (M-Step) [6,14]. For the imputation of missing values, the EM algorithm in the E-Step first computes the mean and covariance values of a data set based on the available (that is, non-missing) values. It then imputes the missing values based on the estimated mean and covariance values. The imputed values are the best possible results according to the maximum likelihood approach based on the available information, which in this case is the mean and covariance values. The EM algorithm then goes to the M-Step in order to update the mean and covariance values by taking the imputed values into consideration. It then again uses the E-Step to make a better imputation of the values using the updated mean and covariance values. The steps continue iteratively as long as the imputation quality keeps improving.

Expectation maximization imputation (EMI) uses the mean and covariance matrix of a whole data set (not a segment) in order to impute a numerical missing value. Let x_a be the vector containing the available values of R_i. If R_i has four available values, then the vector x_a has four elements, where the first element contains the first available value of R_i. Let x_m be the vector that will contain the imputed values of the missing values of R_i. If R_i has three missing values, then x_m has three elements, where in the first element, the imputed value of the first missing value of R_i will be stored. Let μ_m be the mean vector of the attributes having missing values for a record R_i. For example, if R_i has three missing values, then μ_m is a vector having three elements. The first element contains the average value (over all records of D_F) of the first attribute for which R_i has a missing value. Similarly, the second and third elements contain the average of the second and third attributes for which R_i has missing values. Let μ_a be the mean vector of the attributes without missing values for a record R_i. Let $\mu(= \mu_a \cup \mu_m)$ be the mean vector of attributes having available values and of attributes with missing values. Let B be a regression coefficient matrix. $B = \theta_{aa}^{-1}\theta_{am}$, where θ_{aa} is the covariance matrix for the attributes having available values for R_i. For example, if R_i has four available values, then θ_{aa} is a 4×4 matrix, where the element σ_{pq} represents the covariance between the pth attribute and qth attribute for which R_i has available values.

In the E-Step, EMI imputes the missing values x_m of R_i based on the available values x_a of R_i, mean vectors μ_a and μ_m, and coefficient matrix B as shown in Eq. (1). During the imputation, EMI considers the correlations among the attributes.

In Eq. (1), EMI considers that the deviation of a missing value $R_{ij} \in x_m$ from the mean value of the jth attribute $\mu_j \in \mu_m$ is proportional to the deviation of an available value $R_{il} \in x_a$ from the mean value of the lth attribute $\mu_l \in \mu_a$, when the correlation between the jth and the lth attribute is high. The missing values x_m of R_i are imputed using Eq. (1) as follows [43].

$$x_m = \mu_m + (x_a - \mu_a)B + e \tag{1}$$

where e is a residual error with mean zero and unknown covariance matrix [43]. The e value is obtained by randomizing the covariances of the attributes as explained in Step 4 of Sect. 3.3. The e value is used only in the first iteration (i.e., in the first execution of the E-step) of the EMI algorithms.

In the M-Step, EMI again estimates the mean vector μ and the coefficient matrix B considering the imputed data set. The objective of this step was to maximize the imputation quality of the E-Step.

EMI repeats the E-Step and M-Step until the difference between the mean (and covariance matrix) of the current iteration and the mean (and covariance matrix) of the previous iteration is less than user defined thresholds.

3 A novel missing value imputation framework

3.1 The basic contributions

We propose a technique called *A Fuzzy Expectation Maximization and Fuzzy Clustering-based Missing Value Imputation Framework for Data Pre-processing* (FEMI) that makes use of a fuzzy clustering technique and a fuzzy expectation maximization algorithm for imputation. Before we discuss the technique in details, we first introduce the basic concepts and contributions of the proposed framework as follows.

If a numerical attribute value of a record is missing, then it can be imputed based on the available attribute values of the record and the correlations for the attributes in the data set D_F as shown for the EMI technique [43] in Sect. 2. We argue that the imputation accuracy is likely

Table 1 Correlation analysis on the Yeast data set

Cluster ID	Number of Better correlations within a cluster (out of 28 correlations)	Number of records
A	B	C
1	12	427
2	23	50
3	23	44
4	18	20
5	19	461
6	24	10
7	26	30
8	17	163
9	23	35
10	17	244

to be high when all records $R_x \in D_F$; $\forall x$ are very similar to each other, and the correlations for the attributes are high. By similar records, we mean records having similar attribute values resulting in low distances among them. When the records are very similar to each other, then the total number of possible values/options for an attribute are low. Low number of possible values helps to achieve high imputation accuracy. Similarly, when the correlations between two attributes are high, then with the increase of the value of one attribute, the value of the other attribute also increases (or decreases in case of negative high correlation). Therefore, by knowing the value of one attribute, it is possible to impute the value of the other attribute more accurately. Based on this understanding, we aim to first find a group of similar records (from D_F) with high correlations for the attributes and then apply an imputation technique within the group. In order to find similar records, we perform a clustering on the records. Note that existing EMI [14,22,43] uses all records of the whole data set for imputing a missing value R_{xi} and therefore does not use any clusters.

If we perform a clustering on the records of D_F, then we can expect to get similar records in a cluster and dissimilar records in different clusters [11,26]. We argue that typically the records grouped together in a cluster (i.e., similar records) should also have a high correlation between two attributes of the data set. An initial experimentation is carried out on the Yeast data set [15] in order to empirically assess the validity of the argument. We first compute the correlation $C_{i,j}$ between two attributes A_i and A_j based on all records of the Yeast data set. We then group the records of the Yeast data set into ten clusters. The correlation $C^l_{i,j}$ between the attributes A_i and A_j is then computed for the records belonging to a cluster C_l. If $C^l_{i,j} > C_{i,j}$, then we have a stronger/higher correlation between A_i and A_j within a cluster C_l than the correlation between A_i and A_j over the whole data set D_F. Since Yeast has 8 attributes, there are 28 possible pairs of attributes, and therefore, 28 correlations $C_{i,j}$. Table 1 shows the number of pairs of attributes having higher correlations within a cluster. Column A shows the Cluster ID, and Column B presents the number of attribute-pairs (out of 28 pairs) having stronger correlation within the cluster than for the whole data set. It clearly shows that attributes have higher correlations within the clusters than in the whole data set. Column C presents the number of records in each cluster.

We also calculate the similarity of a record R_x with another record R_y for all R_x and R_y. All these similarities are then used to calculate the average similarity S_D among the records

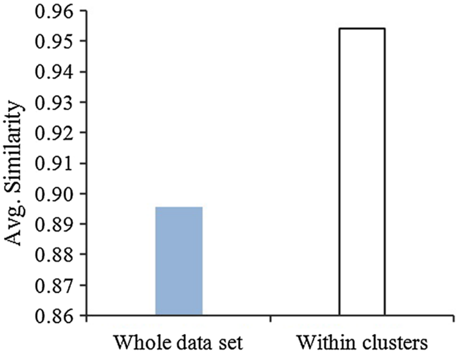

Fig. 1 Similarity analysis on Yeast data set

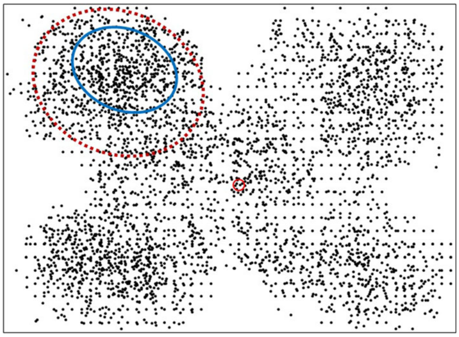

Fig. 2 Basic concepts of FEMI

for the whole data set, as shown by the left bar graph in Fig. 1. The average similarity S_l for the records within a cluster C_l is also calculated. We then compute the average similarity $S_c = \frac{\sum_{l=1}^{k} S_l}{k}$, where k is the total number of clusters. Figure 1 shows that the similarity among the records within the clusters is higher than the similarity among the records for the whole data set.

Sometimes, the records in a data set do not have a clear separation among them, and therefore, obvious boundaries do not exist among the clusters. For example, Fig. 2 shows an example data set (two-dimensional), where each dot represents a record R_x. Each record R_x has two attribute values: one along the X-axis and the other one along the Y-axis. There are perhaps four clusters in the four corners of the example data set (see Fig. 2). However, it can be extremely difficult to draw the exact boundaries of the clusters. For example, the records just outside the outer circle (in the top left corner of Fig. 2) are almost equally attached to the cluster (shown in the top left corner) as the records just inside the circle. Additionally, the records also have some association with other clusters. Therefore, in the fuzzy clustering approach [11,26], a record is assigned to all clusters with different membership degrees based on the argument that a record may not just belong to one or the other cluster, rather it can have different degrees of association with all clusters.

In this study, we propose two levels of fuzziness. For the first level of fuzziness, we consider that a record R_x having a missing value has a membership degree (i.e., a fuzzy association) with each cluster. The fuzzy association of a record with all clusters can be computed using a fuzzy clustering technique such as GFCM [26] as briefly explained in Sect. 3.2. Therefore, instead of imputing a missing value of R_x based on just one cluster, we use all k clusters, and thereby produce the k number of imputed values ($v_1, v_2, \ldots, v_k$). We then compute the final imputation using $v_1, v_2, \ldots, v_k$ and the membership degrees (U_{x1},$U_{x2}, \ldots, U_{xk}$) of R_x with the k clusters. The cluster with which the record R_x has a higher membership degree has more influence in the imputation than the cluster with a lower membership degree.

The second level of fuzziness is that while imputing a missing value ($R_{xi} = v_l$) based on a cluster C_l, we consider that each record of the data set ($R_x \in D_F; \forall x$) has a fuzzy association with C_l. Therefore, while calculating the mean vector for the missing values μ_m, mean vector for the available values μ_a and regression coefficient matrix B (see Eq. (1)) for C_l, we take all records of a data set into consideration according to their membership degrees with C_l. Hence, we modify Eq. (1) as shown in Eqs. (6), (7) and (8).

Let us give a logical justification of the second level of fuzziness as follows using Fig. 2. Let us assume that a record R_x has a missing value R_{xi}, and it is located somewhere within the smaller/inner circle in the top left corner of Fig. 2. Let us also assume that R_{xi} is the Y-axis value of R_x in the two-dimensional example data set. According to our framework, we need to find the records similar to R_x, i.e., we need to find the cluster where R_x belongs to. If we consider the records within the inner circle as the cluster, then we get a different set of μ_a, μ_m and B values than the values we get if we consider the records within the outer circle as the cluster. Since the Y-axis value of R_x is missing, μ_a is the average of the X-axis values for all records within a cluster. It is not trivial to determine which set of values (for μ_a, μ_m and B) are the most useful or sensible since for different cluster boundaries, we get different values. It is not possible to determine the exact boundaries of the clusters. That is, it is not possible to identify the exact set of records belonging to a cluster for the example data set. Therefore, while calculating μ_a, μ_m, and B for C_l, we consider all records of the data set and their membership degrees with C_l as shown in Eq. (6).

Hence, when a data set has the fuzzy nature (as shown in Fig. 2), we expect our novel technique to perform better than even the existing techniques that use similar records either by K-nearest neighbors or hard clustering. Besides, our technique should not be disadvantaged even for a data set that does not have the fuzzy nature since our technique can adjust its behavior accordingly through the use of membership degrees. If a data set is completely non-fuzzy, then the records belonging to a cluster will have a membership degree equal to 1 for the cluster and zero for all other clusters.

The novel concept of the two level fuzziness, the necessary modifications of Eq. (1) [see Eqs. (6), (7) and (8)] in order to support the Fuzzy EMI, and the overall framework are the basic contributions of the study. Note that the existing EM algorithms [14,22,43] do not consider the fuzzy clustering approach and fuzzy EMI imputation while imputing missing values. In the following section, we briefly introduce an existing fuzzy clustering technique called GFCM [26] for the clear understanding of an interest reader on Fuzzy Clustering.

3.2 A general fuzzy C-means (GFCM) [26] clustering algorithm

Clustering algorithms can be grouped into two categories, namely hard clustering and fuzzy (soft) clustering [3]. In hard clustering, a record R_i of a data set D_F belongs to one and only one cluster to which R_i is the most similar. However, in fuzzy clustering, R_i has certain probability (called membership degree) of belonging to each of the clusters. The membership degree U_{ik} for the record R_i with the cluster C_k can vary between 0 and 1. The value $U_{ik} = 1$ indicates a complete association between R_i and C_k, and $U_{ik} = 0$ indicates a complete absence of any association between R_i and C_k. Moreover, the total association of R_i with k clusters (i.e., $C_1, C_2, \ldots C_k$) is equal to 1 (i.e., $\sum_{j=1}^{k} U_{ij} = 1$) [5,26].

We now discuss the general fuzzy C-means (GFCM) [26] clustering algorithm as follows. Let a data set D_F has N records and M attributes. GFCM first requires a user-defined value for k, based on which it groups the records of D_F into k clusters (i.e., $C_1, C_2, \ldots C_k$). It then randomly assigns a membership degree to each record R_i for each cluster in such a way so that $\sum_{j=1}^{k} U_{ij} = 1$. Therefore, for k number of clusters a record R_i has k number of membership degrees (i.e., $\{U_{i1}, U_{i2}, \ldots, U_{ik}\}$), where $\sum_{j=1}^{k} U_{ij} = 1$. The membership degrees of all records with all clusters can be stored in two-dimensional matrix U having N rows and k columns, where U_{ij} contains the membership degree of the ith record with the jth cluster.

Based on U, GFCM then calculates the center of each cluster. Let $V = \{V_1, V_2, \ldots, V_k\}$ be a set of k centers for k-clusters. The center $V_j \in V$ of the jth cluster contains M values $\{v_{j1}, v_{j2}, \ldots, v_{jM}\}$ for the M attributes of the data set, where v_{jl} is the lth attribute value of the jth cluster center.

While computing the attribute values of a cluster center, GFCM applies different approaches for numerical and categorical attributes. For a numerical attribute $A_l \in A$, GFCM calculates v_{jl} by taking the weighted average of the lth attribute values for all records of D_F. The weighted average is computed considering the membership degrees of the records with the jth cluster as shown in Eq. (2).

$$v_{jl} = \frac{\sum_{i=1}^{N} U_{ij}^{m} R_{il}}{\sum_{i=1}^{N} U_{ij}^{m}} \tag{2}$$

where m is a fuzzification coefficient, which is greater than 1.0 and the default value of which is 1.3 [26].

For a categorical attribute $A_l \in A$, GFCM considers that the lth attribute value of a cluster center actually contains all domain values of A_l instead of just one of the domain values. Each of the domain values of A_l has a probability of being the actual value of the lth attribute for the center. For example, let us assume that the domain size of A_l is three meaning that A_l has three domain values say x, y and z. GFCM calculates the probability v_{jl}^{x} of the domain value x being the actual value of v_{jl} as shown in Eq. (3). Similarly, it also calculates the probabilities v_{jl}^{y} and v_{jl}^{z}.

$$v_{jl}^{x} = \frac{\sum_{i=1}^{N} \left(U_{ij}^{m} | R_{il} = x \right)}{\sum_{i=1}^{N} U_{ij}^{m}} \tag{3}$$

The random assignment of the membership degrees $U_{ij}; \forall i, j$ and the calculation of the cluster centers $V = \{V_1, V_2, \ldots, V_k\}$ are considered to be the first iteration of GFCM. After the completion of the first iteration, GFCM then enters into the second iteration, where it recalculates the membership degrees [see Eq. (4)] and the cluster centers as explained above. The membership degree U_{ij} of a record R_i with a cluster C_j is calculated using the similarity of R_i with C_j. The record having high similarity with a cluster will have high membership degree with the cluster [26].

$$U_{ij} = \frac{1}{\sum_{z=1}^{k} \left(\frac{\sum_{l=1}^{M} \delta(R_{il}, v_{jl})^2}{\sum_{l=1}^{M} \delta(R_{il}, v_{zl})^2} \right)^{\frac{1}{(m-1)}}} \tag{4}$$

where $\delta(R_{il}, v_{jl})$ is the dissimilarity between R_{il} and v_{jl}.

Based on the updated (recalculated) membership degrees, a set of cluster centers are calculated using Eqs. (2) and (3), as explained before. The process of recalculating the

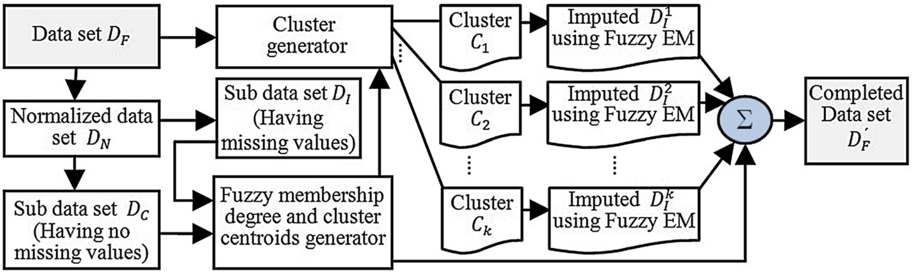
Data set D_F
Cluster generator
Normalized data set D_N
Sub data set D_I (Having missing values)
Sub data set D_C (Having no missing values)
Fuzzy membership degree and cluster centroids generator
Cluster C_1
Cluster C_2
Cluster C_k
Imputed D_I^1 using Fuzzy EM
Imputed D_I^2 using Fuzzy EM
Imputed D_I^k using Fuzzy EM
Σ
Completed Data set $D_F^{'}$

Fig. 3 The overall block diagram of our proposed technique

membership degrees and the cluster centers continues iteratively until a termination condition is met.

Each iteration of GFCM is expected to improve the cluster quality from the previous iteration. The termination condition in GFCM is met when the improvement of cluster quality stops over the two consecutive iterations. The cluster quality is evaluated through the dissimilarity (as calculated by Eq. (5)) of the records within a cluster. The lower the total dissimilarity, the better the cluster quality. Equation (5) computes the total dissimilarity of the clustering solution of the tth iteration.

$$d_t = \sum_{j=1}^{k} \sum_{i=1}^{N} U_{ij}^{m} \sum_{l=1}^{M} \delta(R_{il}, v_{jl})^2 \tag{5}$$

3.3 The main steps of our proposed technique

We now introduce the main steps of the FEMI framework. We also present an overall block diagram of the FEMI framework as shown in Fig. 3. Besides, we present a running example to illustrate the steps of FEMI. Note that the missing values of the running example (the main purpose of which is to illustrate the main steps clearly) may appear to be straightforward and can be imputed using a simple functional dependency analysis without requiring the sophistication of FEMI. However, the real data sets are typically a lot more complicated, than the running example, requiring the extra elegance of FEMI as evident from the experimental results presented in Sect. 4.

Step 1: Copy a full data set D_F into D_N and normalize all numerical attributes of D_N within a range between 0 and 1.
Step 2: Divide the data set D_N into two sub data sets D_C (having only records without missing values) and D_I (having only records with missing values).
Step 3: Find membership degrees of all records of D_C and D_I with all clusters.
Step 4: Apply our novel *FuzzyEM* method to impute numerical missing values using all clusters.
Step 5: Find the combined imputed value of a numerical attribute. Find the imputed value of a categorical attribute.
Step 6: Combine records to form a completed data set (D'_F) without any missing values.

Step 1: Copy a full data set D_F into D_N and normalize all numerical attributes of D_N within a range between 0 and 1.

We first make a copy of a data set D_F having $|R|$ records and $|A|$ attributes into D_N. $R = \{R_1, R_2, ...R_n\}$ is the set of records and $A = \{A_1, A_2, ...A_m\}$ is the set of attributes

Algorithm 1: FEMI

```
Input   : Data set $D_F$ having $|R|$ records and $|A|$ attributes
Output  : Imputed data set $D'_F$ having $|R|$ records and $|A|$ attributes
Step 1:
    Set $D_N \leftarrow D_F$;
    $D_N \leftarrow Normalize(D_N)$; /*Normalize all numerical attributes of $D_N$ into a range between 0 and 1*/
end
Step 2:
    Divide $D_N$ into $D_C$ having $|R'|$ records without missing values and $D_I$ having $|(R - R')|$ records with missing
    values;
end
Step 3:
    Apply a fuzzy clustering algorithm like GFCM on $D_C$ to find a set of $k$-cluster centroids $V = \{V_1, V_2, \ldots, V_k\}$
    and a membership degree matrix $U^C$ having $|R'|$ rows and $k$ columns;
    For the records of $D_I$, find fuzzy membership degree matrix $U^I$ having $|(R - R')|$ rows and $k$ columns for the
    same $k$-cluster centroids $V = \{V_1, V_2, \ldots, V_k\}$;
    Set $U \leftarrow U^C \cup U^I$;
end
Step 4:
    Set $D_I \leftarrow Denormalize(D_I)$;
    foreach cluster $C_k$ do
        $D_I^k \leftarrow$ FuzzyEM($k, D_I, D_F, U$) ; /*$k$-cluster index*/
    end
end
Step 5:
    foreach record $R_i \in D_I$ do
        foreach attribute $A_z \in A$ do
            if $A_z$ is missing then
                if $A_z$ is numerical then
                    Impute $R_{iz}$ using Equation (9);
                end
                else if $A_z$ is categorical then
                    For each domain value $pl \in P$, calculate the vote $G_{pl}$ using Equation (11);
                    Find the maximum vote, $G_{max} \leftarrow \max(G_{pl}; \forall pl \in P)$;
                    Impute $R_{iz}$ by the domain value associated with $G_{max}$;
                end
            end
        end
    end
end
Step 6:
    Set $D_C \leftarrow Denormalize(D_C)$;
    Completed data set $D'_F \leftarrow D_C \cup D_I$;
    Return $D'_F$;
end
```

in D_F. We then normalize each numerical attribute $A_i \in A$ of D_N into a range between 0 and 1 as shown in the Step 1 of the FEMI algorithm (Algorithm 1). Table 2a shows an example data set D_F. The data set D_F has 15 records and 4 attributes out of which two are numerical and two are categorical. The records R2, R8, R10, and R13 have missing values for attributes Education (Edu.), Position (Pos.), Age and Salary (in thousand), respectively. The normalized data set D_N is shown in Table 2b.

Step 2 Divide the data set D_N into two sub data sets D_C (having only records without missing values) and D_I (having only records with missing values).

In this step, we divide the data set D_N into two sub data sets namely D_C and D_I, where D_C contains $|R'|$ records without missing values (see Table 2c) and D_I contains $(|R| - |R'|)$ records with missing values (see Table 2d) as shown in the FEMI algorithm (Step 2 of Algorithm 1).

Step 3: Find membership degrees of all records of D_C and D_I with all clusters.

Algorithm 2: Procedure FuzzyEM()

```
Input  : Cluster index $k$, data set $D_I$, data set $D_F$, membership degree matrix $U$
Output : Imputed data set $D_I^k$

Step 1:
    initialize termination criteria, $\epsilon \leftarrow 10^{-10}$;
end
Step 2:
    Set $D_I^k \leftarrow \phi$;
    foreach record $R_i \in D_I$ do
        Set $m \leftarrow$ Number of numerical attribute(s) of $R_i$ having a missing value;
        Set $a \leftarrow$ Number of numerical attribute(s) of $R_i$ having a non-missing value;
        Calculate weighted mean vector, $\mu_{mis}^k$ using Equation (6) for $m$ number of attributes having missing values
        in $R_i$;
        Calculate weighted mean vector, $\mu_{av}^k$ using Equation (6) for $a$ number of attributes having available values
        in $R_i$;
        Calculate weighted covariance matrix $\mathbf{B} = (\theta_{aa}^k)^{-1}\theta_{am}^k$ using Equation (7);
        Calculate residual error, $e \leftarrow [\mu_0 + H.Z^T]^T$;
        Set $x_m$ and $x_a$ as $(1 \times m)$ and $(1 \times a)$ size vectors having missing and available attribute values of $R_i$;
        Impute $x_m \leftarrow \mu_{mis}^k + (x_a - \mu_{av}^k)\mathbf{B} + e$;
        Update $R_i$ with the imputed values $x_m$;
        $D_I^k \leftarrow D_I^k \cup R_i$;
    end
    Replace missing values of $D_F$ by corresponding values of $D_I^k$;
end
Step 3:
    Repeat Step 2 until the change of mean and covariance, in $D_F$, obtained from two consecutive iterations is less
    than $\epsilon$;
end
Step 4:
    Return imputed data set $D_I^k$;
end
```

In Step 3, we apply a fuzzy clustering technique such as GFCM [26] on D_C with a user-defined k number of clusters as an input in order to produce a set of k cluster centers V = $\{V_1, V_2, \ldots, V_k\}$ as shown in Step 3 of Algorithm 1. The clustering technique also produces a membership degree matrix U^C having $|R'|$ rows and k columns, where U_{ij}^C is the membership degree of the ith record ($R_i \in D_C$) with the jth cluster.

Using the membership degree calculation equation of the fuzzy clustering technique, we then find the membership degree matrix U^I for all records of D_I and the same k clusters having the same cluster centers V = $\{V_1, V_2, \ldots, V_k\}$ as mentioned before. U_{lm}^I is the membership degree of the lth record ($R_l \in D_I$) with the mth cluster. Membership degree of a record with a cluster is inversely proportionate to the distance between the record and the center of the cluster. We next combine U^C and U^I into U having $|R|$ number of rows and k number of columns.

For the example data set (as shown in Table 2c), let us use k = 3 in order to produce 3 clusters. The membership degrees of each record of $R_i \in D_C$ with 3 clusters are shown in Table 3a. Each cluster has a centroid which contains the weighted mean of a numerical attribute and the frequency of each attribute value of a categorical attribute as shown Table 3b.

We then calculate the membership degrees of each record $R_l \in D_I$ with a cluster C_j; $\forall j$, using the centroid of C_j and the available values of R_l. Membership degrees are calculated based on the equations used in GFCM [26]. In our example, the membership degrees U^I, of $R_l \in D_I$; $\forall R_l$ with the 3 clusters are shown in Table 3c. Finally, we combine the membership degrees U^C, and U^I into U, where U_{ij} is the membership degree of the ith record $R_i \in D_F$ with the jth cluster. Note that the membership degree of a record $R_i \in D_F$ is the same as the

Table 2 A sample data set D_F and normalized data set D_N

Rec.	Age	Edu.	Salary	Pos.
(a) A sample data set D_F				
R1	27	MS	85	L
R2	45	**?**	145	P
R3	42	PhD	145	P
R4	25	MS	85	L
R5	50	PhD	146	P
R6	28	MS	85	L
R7	38	PhD	140	P
R8	43	PhD	147	**?**
R9	44	PhD	146	P
R10	**?**	MS	86	L
R11	42	PhD	142	P
R12	26	MS	84	L
R13	42	PhD	**?**	P
R14	25	MS	86	L
R15	43	PhD	143	P
(b) Normalized data set D_N				
R1	0.08	MS	0.01587	L
R2	0.80	?	0.96825	P
R3	0.68	PhD	0.96825	P
R4	0.00	MS	0.01587	L
R5	1.00	PhD	0.98413	P
R6	0.12	MS	0.01587	L
R7	0.52	PhD	0.88889	P
R8	0.72	PhD	1.00000	?
R9	0.76	PhD	0.98413	P
R10	?	MS	0.03175	L
R11	0.68	PhD	0.92063	P
R12	0.04	MS	0.00000	L
R13	0.68	PhD	?	P
R14	0.00	MS	0.03175	L
R15	0.72	PhD	0.93651	P
(c) Data set D_C having no missing values				
R1	0.08	MS	0.01587	L
R3	0.68	PhD	0.96825	P
R4	0.00	MS	0.01587	L
R5	1.00	PhD	0.98413	P
R6	0.12	MS	0.01587	L
R7	0.52	PhD	0.88889	P
R9	0.76	PhD	0.98413	P

Table 2 continued

Rec.	Age	Edu.	Salary	Pos.
R11	0.68	PhD	0.92063	P
R12	0.04	MS	0.00000	L
R14	0.00	MS	0.03175	L
R15	0.72	PhD	0.93651	P
(d) Data set D_I				
R2	0.80	?	0.96825	P
R8	0.72	PhD	1.00000	?
R10	?	MS	0.03175	L
R13	0.68	PhD	?	P

Table 3 Membership degree of the records of D_C and D_I with three clusters and clusters centroids V

Rec.		Cluster 1	Cluster 2	Cluster 3
(a) Membership degrees U^C of each record of D_C with three clusters				
R1		0.00000002	0.99999976	0.00000022
R3		0.99999986	0.00000009	0.00000005
R4		0.00000013	0.99998951	0.00001036
R5		0.00000011	0.00000252	0.99999737
R6		0.00000005	0.99999923	0.00000072
R7		0.99941392	0.00000013	0.00058595
R9		0.99733209	0.00000032	0.00266759
R11		0.99999438	0.00000362	0.00000200
R12		0.00000122	0.99999301	0.00000577
R14		0.00000044	0.99999122	0.00000834
R15		0.99999270	0.00000141	0.00000589
(b) Centroids of three clusters				
Attr. name	Attr. value	Cluster 1	Cluster 2	Cluster 3
Age		0.6719620190601	0.0480000000929	0.9998616492793
Edu.	MS	0.0000000000002	0.9999999997743	0.0000000000005
	PhD	0.9999999999998	0.0000000002257	0.9999999999995
Salary		0.9396589053030	0.0158720001969	0.9841240084866
Pos.	L	0.0000000000002	0.9999999997743	0.0000000000005
	P	0.9999999999998	0.0000000002257	0.9999999999995
(c) Membership degrees U^I of each record of D_I with three clusters				
Rec.		Cluster 1	Cluster 2	Cluster 3
R2		0.944139470580681	0.000000060804250	0.055860468615069
R8		0.999816462968573	0.000000002003948	0.000183535027478
R10		0.000000000000032	0.999999999999939	0.000000000000029
R13		0.999999999978393	0.000000000000001	0.000000000021606

membership degree of the record $R_i^N \in D_N$ with a cluster C_j, where R_i^N is the normalized form of R_i.

Step 4 Apply our novel *FuzzyEM* method to impute numerical missing values using all clusters.

In this step, we impute the missing numerical attribute values using a novel Fuzzy Expectation Maximization approach, which is a modification (fuzzy version) of an existing approach [43]. The basic idea of this step is to use the membership degrees of the records with a cluster, in order to impute the missing values as shown in Step 4 of the FEMI algorithm (see Algorithm 1) and Procedure FuzzyEM (see Algorithm 2). The missing values are imputed using the membership degrees of all clusters one by one. For example, we impute the missing values using the membership degrees of the records with Cluster 1 and store the imputed data set in D_I^1. Similarly, we produce D_I^2, $D_I^3 \dots D_I^k$ from Cluster 2, Cluster3, ... Cluster k, respectively.

FuzzyEM takes four input parameters namely the index k of a cluster, data set having records with missing values D_I, full data set D_F, and membership degree matrix U. We next describe the proposed iterative imputation process as follows.

In this step, we first denormalize D_I data set, where each record has one or more missing values. Let R_i be a record of D_I. R_i has m number of numerical attributes with missing values and a number of numerical attributes with available values.

Let μ_{mis}^k and μ_{av}^k be the fuzzy mean vectors of the m number of missing values and a number of available values, respectively. The fuzzy mean is calculated considering the membership degree of each record. It is the average value weighted by the membership degrees as shown in Eq. (6). We calculate fuzzy mean, μ_p^k, for the pth attribute according to the kth cluster as follows. Note that the fuzzy mean of an attribute according to a cluster can be different to the fuzzy mean of the attribute according to another cluster.

$$\mu_p^k = \frac{\sum_{i=1}^{|R|} U_{ik} R_{ip}}{\sum_{i=1}^{|R|} U_{ik}} \tag{6}$$

where $|R|$ is the number of records in D_F, U_{ik} is the membership degree of the ith record with the kth cluster, and R_{ip} is the pth attribute value of the ith record.

Using μ_{mis}^k and μ_{av}^k, we now calculate fuzzy covariance matrix $\mathbf{B} = (\theta_{aa}^k)^{-1}\theta_{am}^k$, where the element σ_{pq}^k, of θ_{aa}^k or θ_{am}^k, is a covariance between the p-th and q-th attributes according to the kth cluster and is calculated as follows. It is a covariance value weighted by the membership degrees of the records with a cluster as shown in Eq. (7). Note that the covariance of two attributes according to a cluster can be different to the covariance of the attributes according to a different cluster.

$$\sigma_{pq}^k = \frac{\sum_{i=1}^{|R|} U_{ik}(R_{ip} - \mu_p^k)(R_{iq} - \mu_q^k)}{\sum_{i=1}^{|R|} U_{ik}} \tag{7}$$

Now, let x_m and x_a be the vectors of missing attribute values and available attribute values in R_i, respectively. We impute the missing values of x_m as follows.

$$x_m = \mu_{mis}^k + (x_a - \mu_{av}^k)\mathbf{B} + e \tag{8}$$

where e is a residual error with mean zero and unknown covariance matrix $Q(= \theta_{mm} - \theta_{ma}\theta_{aa}^{-1}\theta_{am})$. Following an existing approach [43], we use a residual error $e = [\mu_0 + H.Z^T]^T$, where μ_0 is a mean vector having zero value/s meaning that the elements of the vector μ_0 represent the mean values, which are all equal to zero in this case, H is a cholesky decomposition of the covariance matrix Q, and Z is a vector having Gaussian random values that have mean zero and variance equal to one. Since H is multiplied by Z for the calculation of e, the e value is obtained through a randomization of the cholesky decomposition of the covariances. Note that the residual error e is used in Eq. (8) for the first iteration only.

After imputing all the records of $R_i \in D_I, \forall i$ we get an imputed data set D_I^k. We replace the missing values of D_F with the corresponding imputed values in D_I^k. We then re-calculate μ_{mis}^k, μ_{av}^k and **B** using the updated D_F. We next re-impute x_m using (8) for all the records of D_I. We repeat this process until the change of the means and covariances in D_F of two consecutive iterations is less than a user-defined threshold ϵ. We use 10^{-10} as a value for the termination threshold (ϵ) in the experiments. The Procedure *FuzzyEM* then returns the imputed data set D_I^k. Following [43], we use the residual error only in the first iteration.

We impute our example data set according to three clusters using the Algorithm 2. The imputed data sets are shown in Table 4a–c. For easy understanding, the imputed values are presented in bold font.

Note that for a non-fuzzy data set where U_{ik} can be either 1 or 0, μ_{miss}^k (see Eq. (6)) becomes equal to μ_m (see Eq. (1)) for some k. Similarly, μ_{av}^k becomes equal to μ_a and σ_{pq}^k becomes equal to σ_{pq} for some k. Therefore, Eq. (8) becomes equal to Eq. (1), suggesting that the existing EMI [43] technique is a special case of the proposed FEMI algorithm.

Step 5 Find the combined imputed value of a numerical attribute. Find the imputed value of a categorical attribute.

In this step, we impute both numerical and categorical missing values of D_I (see Fig. 3 and Step 5 of Algorithm 1). For imputing numerical missing value of D_I, we combine all $D_I^s; \forall s \in K$, where $K = \{1, 2, \ldots k\}$. Let R_{ip} be the pth attribute value (numerical) of the ith record in D_I. Let v_{ip}^s be the pth attribute value of the ith record in the data set D_I^s, which is imputed according to sth cluster as explained in Step 4. If R_{ip} is a missing value, then it is computed as follows.

$$R_{ip} = \frac{\sum_{s=1}^{k} U_{is} v_{ip}^s}{\sum_{s=1}^{k} U_{is}}, \quad \text{where } U_{is} \in U^I \tag{9}$$

We now explain our approach toward imputing a categorical attribute value. Many fuzzy clustering techniques such as GFCM produce a fuzzy seed (center) for a cluster where the seed contains each value of the domain of an attribute according to a confidence degree [24,26]. The confidence degree of an attribute value p_l in a cluster s is the sum of the membership degrees for the records, having p_l, with the cluster. So the confidence degree $C_{p_l}^k$ for the value p_l of the p-th attribute (categorical) in the kth cluster can be calculated as follows.

$$C_{p_l}^k = \sum_{i=1}^{|R|} U_{ik} | v_{ip}^k = p_l \tag{10}$$

Similarly, confidence degree for all domain values of an attribute can be calculated. A fuzzy seed of a cluster can contain all the domain values of a categorical attribute and their corresponding confidence degrees with the cluster. Naturally, the value having the highest confidence degree is considered to be the most likely value of the attribute in the cluster.

While imputing a missing value of a record R_i, we calculate the vote (G_{p_l}) for a value p_l by multiplying its confidence degree ($C_{p_l}^s$) in terms of the s-th cluster, and the membership degree U_{is} of R_i with the s-th cluster, for all clusters as follows.

Table 4 Imputed data set D_I

Rec.	Age	Edu.	Salary	Pos.
(a) Imputed data set D_I^1 using *FuzzyEM* on cluster 1				
R2	45	?	145	P
R8	43	PhD	147	?
R10	**2.2**	MS	86	L
R13	42	PhD	**143.6**	P
(b) Imputed data set D_I^2 using *FuzzyEM* on cluster 2				
R2	45	?	145	P
R8	43	PhD	147	?
R10	**25.7**	MS	86	L
R13	42	PhD	**82**	P
(c) Imputed data set D_I^3 using *FuzzyEM* on cluster 3				
R2	45	?	145	P
R8	43	PhD	147	?
R10	**3.4**	MS	86	L
R13	42	PhD	**144.4**	P
(d) The final imputed data set D_I				
R2	45	**Phd**	145	P
R8	43	PhD	147	**P**
R10	**25.69**	MS	86	L
R13	42	PhD	**143.6**	P

$$G_{p_l} = \sum_{s=1}^{k} C_{p_l}^{s} \times U_{is} \tag{11}$$

Similarly, we calculate $G_{p_l}; \forall p_l \in P$, where P is the domain of the p-th attribute. Finally, the value having the maximum vote is considered to be the imputed value R_{ip} for the p-th attribute of the i-th record. The final imputed data set of D_I (in our example) is shown in Table 4d.

Step 6 Combine records to form a completed data set (D'_F) without any missing values.

We finally combine denormalized D_C and imputed D_I in order to form D'_F, which is the imputed data set.

3.4 A few possible approaches for automatically determining the number of clusters, k

Since FEMI requires a number of clusters k for the fuzzy clustering technique (as discussed in Step 3 of Sect. 3.3), a suitable approach (for automatically determining the most appropriate k value) could be incorporated into the FEMI algorithm. In the following paragraphs, we discuss a few possible approaches to automatically determine the k value. However, they need to be carefully evaluated in order to find the best of them and we plan to carry out the evaluation in our future study. In the experiments of this study, we use k = 20 based on our initial empirical analysis as discussed in Sect. 4.4.

Approach 1: The full data set D_F can be divided into two (mutually exclusive) horizontal segments D_C (having only records without any missing value/s) and D_I that has only the records with missing value/s. Artificial missing values can be created in D_C randomly, using the same approach that we have taken in this study to simulate missing values (see Sect. 4.2). The artificial missing values can then be imputed many times, where each time FEMI can use different k values. Finally, the k value resulting in the best imputation accuracy can be chosen as the k value for FEMI in order to impute the real missing values. Note that the process of generating artificial missing values and imputation can be run many times, and the average result for each k value can be used for finding the best k.

Approach 2: Instead of GFCM [26], an existing fuzzy clustering algorithm such as FBSA [45], that automatically finds the k value, can be used. Note that FBSA finds the best k value by comparing the quality of clusters obtained for different k values.

Approach 3: The k-means algorithm [21] can be applied many times for different k values and thereby produce different clustering results. All clustering results can be evaluated using any evaluation metric such as the Silhouette coefficient [41] and Davies?Bouldin Index (DBI) [13]. The k with the best clustering result can then be used in FEMI for the imputation of missing values.

Approach 4: Instead of k-means a basic fuzzy c-means such as FCM [5] can be applied many times in order to find the best k value. This approach does not require FEMI to use the GFCM technique for clustering since a fuzzy clustering technique like FCM computes the membership degrees as well. The set of membership degrees for the best k value can be directly used in the FEMI algorithm.

An advantage of the first approach is that the best k value is determined by evaluating the imputation accuracy instead of evaluating the cluster quality for each k value as required by the other approaches. The k value determined based on the imputation accuracy may produce better end result than the k value determined based on the cluster quality. However, a disadvantage of the first approach is its time complexity since it requires the complete imputation many times in order to determine the best k value. An advantage of the second approach is a relatively lower time complexity than the first approach as it does not require to run all steps of FEMI in order to identify the best k value. Since k-means is well known for its low time complexity [20], the third approach can also enjoy its simplicity in terms of time complexity compared with the first approach. Since the fourth approach skips the step involving GFCM, it can save some computation time. However, a possible disadvantage of Approach 2, Approach 3, and Approach 4 can be the use of clustering quality instead of imputation quality.

3.5 Complexity analysis

We now analyze complexity for FEMI, EMI, GkNN, FKMI, SVR, and IBLLS. We consider that we have a data set with n records, and m attributes. We also consider that there are n_I records with one or more missing values, and n_c records ($n_c = n - n_I$) with no missing values. FEMI uses a fuzzy clustering technique such as GFCM [26] to create clusters with a user-defined number of clusters k.

Complexity of Step 1 for normalizing all numerical attributes is $O(nm)$. In Step 2, the complexity for preparing D_I and D_C is $O(nm)$. The complexity of Step 3 is dominated by the GFCM algorithm [26] which has a complexity $O(k^2 n_c m d)$, where we consider that the domain size of each attribute is d.

Step 4 uses the $FuzzyEM()$ procedure, which has a complexity $O(nn_I m^2 + n_I m^3)$. $FuzzyEM()$ is applied repeatedly k times which makes the complexity of this step is $O(knn_I m^2 + kn_I m^3)$. The complexity for imputing missing values in Step 5 is $O(knn_I m)$. Complexity of Step 6 for denormalizing all numerical attributes is $O(n_C m)$.

Table 5 Data sets at a glance

Data set	Records	Num. attr.	Cat. attr.	Missing	Pure Rec.
Adult	32,561	6	9	Yes	30,162
Chess	28,056	3	4	No	28,056
Yeast	1,484	8	1	No	1,484
CMC	1,473	2	8	No	1,473
GermanCA	1,000	7	14	No	1,000
Pima	768	8	1	No	768
Housing	506	11	3	No	506
Autompg	398	5	3	Yes	392

Therefore, the overall complexity of FEMI is $O(nm + k^2n_cmd + knn_Im^2 + kn_Im^3 + knn_Im$. However, typically k, and d values are very small, especially compared with n. Besides, we can also consider n_I to be very small and therefore $n_c \approx n$. Hence, the overall complexity of FEMI is $O(nm^2 + m^3)$. Moreover, for low-dimensional data sets such as those used in this study, the complexity is $O(n)$. We estimate the complexities of EMI, FKMI, SVR and IBLLS (i.e., the techniques that we use in the experiments of this study) as $O(nm^2+m^3)$, $O(nm)$, $O(n^3m)$ and $O(n^3m^2 + nm^4)$, respectively. Moreover, the complexity of GkNN is $O(n^2mlog(n))$ [53]. This is also reflected in the execution time complexity analysis in the next section (see Table 8).

4 Experimental result

We compare FEMI with four existing techniques namely EMI [22,43], GkNN [53], FKMI [27, 31] and IBLLS [12]. Moreover, we use the Java implementation (LibSVM [10]) of an existing technique SVR [44,49]. The existing techniques have been shown to be better than Bayesian principal component analysis (BPCA) [33], LLSI [25], and ILLSI [9].

4.1 Data sets

We apply the techniques on eight real data sets, namely Adult, Chess, Yeast, Contraceptive Method Choice (CMC), GermanCA, Pima, Housing and Autompg data set that are available from UCI Machine Learning Repository [15]. A brief description of the data sets is presented in Table 5. For instance, the Adult data set has 32,561 records, 6 numerical and 9 categorical attributes. There are a number of records having missing values. We first remove all records having missing values. Therefore, we get a pure data set having 30162 records without any missing values. In our experiment we use the pure data set.

4.2 Simulation of missing values

In pure data sets, we artificially create missing values, which are then imputed by different techniques. Since the original values of the artificially created missing data are known to us, we can evaluate the performances of the techniques.

Both the amount and type of missing data influence the imputation performance [22]. Missing data can have many different types. For example, in one scenario (type), we may have a data set where a record has at most one missing value, and in another scenario, we may have records with multiple missing values, but both data sets may have the same number of total missing values. Moreover, the probability of a value being missing typically does not depend on the missing value itself [42,43], and hence, missing values often can have a random nature, which can be difficult to formulate. Therefore, in this experiment, we use various types of missing values such as simple, medium, complex and blended, as explained below.

A record can have at most one missing value for a simple pattern, whereas in a medium pattern, if a record has any missing value, then it has minimum 2 attributes with missing values and maximum 50 % of the attributes with missing values [22,35,40]. Similarly, a record having missing values in a complex pattern has minimum 50 % and maximum 80 % attributes with missing values. In a blended pattern, we have a mixture of records from all three other patterns. A blended pattern contains 25 % records having missing values in simple pattern, 50 % in medium pattern and 25 % in complex pattern. Blended pattern simulates a natural scenario where we may expect a combination of all three missing patterns.

For each of the missing patterns, we use different missing ratios (1, 3, 5 and 10 %) where x % missing ratios means x % of the total attribute values (not records) of a data set are missing. For example, if a data set has 5 records and 20 attributes then a missing ratio of 10% means that 10 values, out of the total 100 (= 5 × 20) attribute values, are missing. Therefore, for 10 % missing ratios and simple missing pattern, the total number of records having missing values may exceed the total records in some data sets. In the experiments, we therefore use 6 % missing ratios (instead of 10 % missing ratios) only for simple missing pattern with 10 % missing ratios, for all data sets.

Moreover, we use two types of missing models namely Overall and Uniformly Distributed (UD). In the overall distribution, missing values are not equally spread out among the attributes, and in the worst case scenario, all missing values can belong to a single attribute. However, in the UD model each attribute has equal number of missing values.

Note that there are 32 combinations (id 1, 2, . . . , 32) of Missing Ratio, Missing Model, and Missing Pattern. For each combination, we create 10 data sets with missing values. For example, for the combination having “1 %” missing values , “overall” missing model , and “simple” missing pattern (id 1, see Table 6), we generate 10 data sets with missing values. We therefore create all together 320 data sets for each natural data set namely Adult, Chess, Yeast, Contraceptive Method Choice (CMC), GermanCA, Pima, Housing and Autompg.

4.3 Evaluation criteria

The imputation accuracy of FEMI is evaluated using two well-known evaluation criteria namely root mean squared error (RMSE) and mean absolute error (MAE).

We now define the evaluation criteria briefly. Let N be the number of artificially created missing values, O_i $(1 \leq i \leq N)$ be the actual value of the ith artificially created missing value, P_i be the imputed value of the ith missing value.

The most commonly used imputation performance indicator is the root mean squared error (RMSE) [22], which aim to explore the average difference of actual values with the imputed values as shown in (12). Its value ranges from 0 to ∞, where a lower value indicates a better matching.

$$\text{RMSE} = \left(\frac{1}{N}\sum_{i=1}^{N}[P_i - O_i]^2\right)^{\frac{1}{2}} \tag{12}$$

Table 6 Performance of FEMI, GkNN, FKMI, SVR, EMI, and IBLLS based on RMSE, and MAE for 32 missing combinations on Adult data set

Missing combination			Id	RMSE (Lower value is better)						MAE (Lower value is better)					
				FEMI	GkNN	FKMI	SVR	EMI	IBLLS	FEMI	GkNN	FKMI	SVR	EMI	IBLLS
1 %	Overall	Simple	1	**0.103**	0.156	0.182	0.120	0.123	0.149	**0.072**	0.119	0.135	0.095	0.082	0.113
		Medium	2	**0.106**	0.147	0.173	0.128	0.126	0.175	**0.074**	0.113	0.128	0.100	0.084	0.123
		Complex	3	**0.106**	0.151	0.174	0.137	0.126	0.211	**0.071**	0.111	0.124	0.105	0.081	0.136
		Blended	4	**0.109**	0.151	0.176	0.130	0.129	0.179	**0.074**	0.110	0.127	0.101	0.084	0.125
	UD	Simple	5	**0.106**	0.146	0.172	0.122	0.126	0.169	**0.074**	0.112	0.125	0.096	0.084	0.133
		Medium	6	**0.107**	0.149	0.174	0.129	0.127	0.176	**0.072**	0.110	0.126	0.100	0.082	0.125
		Complex	7	**0.106**	0.148	0.173	0.135	0.126	0.231	**0.071**	0.109	0.125	0.104	0.081	0.164
		Blended	8	**0.103**	0.146	0.171	0.129	0.123	0.179	**0.072**	0.111	0.127	0.100	0.082	0.128
3 %	Overall	Simple	9	**0.107**	0.148	0.173	0.121	0.126	0.181	**0.078**	0.115	0.130	0.096	0.087	0.151
		Medium	10	**0.105**	0.146	0.172	0.127	0.125	0.209	**0.074**	0.112	0.128	0.100	0.085	0.159
		Complex	11	**0.105**	0.149	0.174	0.137	0.126	0.232	**0.073**	0.113	0.128	0.105	0.084	0.170
		Blended	12	**0.106**	0.146	0.173	0.128	0.125	0.216	**0.074**	0.111	0.129	0.100	0.084	0.159
	UD	Simple	13	**0.103**	0.144	0.172	0.121	0.125	0.193	**0.075**	0.113	0.129	0.096	0.086	0.151
		Medium	14	**0.107**	0.147	0.175	0.129	0.126	0.271	**0.076**	0.113	0.132	0.101	0.086	0.199
		Complex	15	**0.105**	0.149	0.174	0.136	0.126	0.216	**0.072**	0.113	0.127	0.105	0.083	0.159
		Blended	16	**0.107**	0.148	0.174	0.130	0.127	0.206	**0.077**	0.114	0.130	0.101	0.087	0.160
5 %	Overall	Simple	17	**0.107**	0.148	0.173	0.121	0.126	0.184	**0.079**	0.118	0.132	0.096	0.088	0.133
		Medium	18	**0.105**	0.146	0.175	0.128	0.126	0.208	**0.076**	0.114	0.132	0.100	0.086	0.160
		Complex	19	**0.107**	0.148	0.174	0.137	0.127	0.248	**0.075**	0.113	0.128	0.105	0.085	0.183
		Blended	20	**0.108**	0.149	0.175	0.130	0.128	0.241	**0.076**	0.114	0.131	0.101	0.087	0.195
	UD	Simple	21	**0.107**	0.147	0.177	0.122	0.127	0.193	**0.078**	0.115	0.136	0.096	0.089	0.152
		Medium	22	**0.107**	0.147	0.175	0.129	0.127	0.190	**0.076**	0.114	0.132	0.101	0.086	0.144
		Complex	23	**0.107**	0.149	0.176	0.139	0.127	0.257	**0.074**	0.113	0.130	0.106	0.084	0.193
		Blended	24	**0.104**	0.146	0.175	0.128	0.125	0.230	**0.074**	0.113	0.134	0.100	0.085	0.173
10 %	Overall	Simple	25	**0.107**	0.147	0.177	0.123	0.127	0.239	**0.077**	0.115	0.138	0.097	0.088	0.199
		Medium	26	**0.106**	0.147	0.174	0.128	0.127	0.234	**0.078**	0.116	0.131	0.100	0.088	0.179
		Complex	27	**0.107**	0.149	0.175	0.140	0.127	0.259	**0.076**	0.116	0.130	0.107	0.086	0.190
		Blended	28	**0.106**	0.148	0.174	0.129	0.127	0.235	**0.077**	0.117	0.131	0.100	0.088	0.173
	UD	Simple	29	**0.106**	0.146	0.175	0.121	0.125	0.209	**0.077**	0.115	0.135	0.096	0.087	0.163
		Medium	30	**0.106**	0.148	0.174	0.128	0.127	0.232	**0.078**	0.117	0.134	0.100	0.089	0.172
		Complex	31	**0.106**	0.148	0.174	0.142	0.127	0.256	**0.075**	0.113	0.128	0.109	0.085	0.194
		Blended	32	**0.106**	0.148	0.175	0.128	0.127	0.224	**0.078**	0.118	0.135	0.100	0.088	0.175

The mean absolute error (MAE) [22] determines the closeness between actual and imputed values. Similar to RMSE, its value ranges from 0 to ∞, where a lower value indicates a better matching.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|P_i - O_i| \tag{13}$$

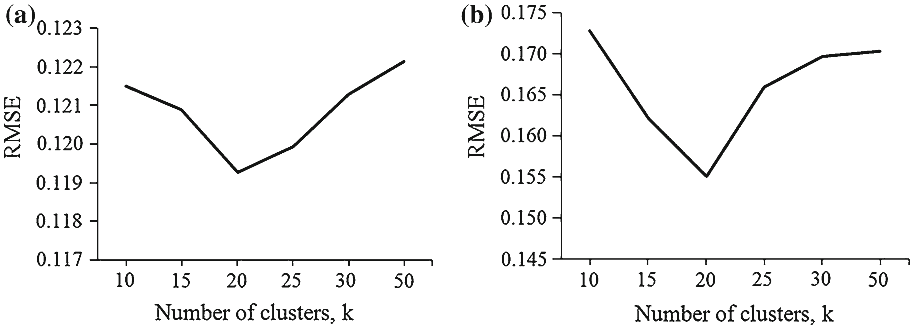

Fig. 4 Performance of FEMI based on RMSE for different number of clusters. **a** Credit Approval data set, **b** CMC data set

4.4 Justification of k

FEMI uses a user-defined value k (the number of clusters) as explained in Sect. 3. In order to explore a suitable value for k, we first evaluate the performances of FEMI for the Credit Approval data set and CMC data set by using different values of k as shown in Fig. 4. For both data sets, $k = 20$ gives the best result based on the evaluation criteria RMSE. Note that, $k = 20$ also gives the best result based on MAE for the data sets. Therefore, we use the default value of k equal to 20 for the experiments in this study.

4.5 Experimental result analysis for the adult data set

In Table 6, we present the performance of FEMI, GkNN, FKMI, SVR, EMI and IBLLS, on Adult data set, based on RMSE and MAE for 32 missing combinations. The average values of the performance indicators on 10 data sets having missing values for each combination of missing ratios, missing model, and missing pattern are presented in Table 6. For example, there are 10 data sets having missing values with the combination ($id = 1$) of “1 %” missing ratio, “Overall” missing model and “Simple” missing pattern. The average of RMSE for the data sets having $id = 1$ is 0.103 for FEMI as reported in Table 6. The average RMSE values for the data sets having $id = 1$ are 0.156, 0.182, 0.120, 0.123 and 0.149 for GkNN, FKMI, SVR, EMI, and IBLLS, respectively. Bold values in the table indicate the best results among the three techniques. FEMI performs significantly better than all other techniques on the data set. In 32 out of 32 combinations of missing patterns, FEMI performs better than all other techniques in terms of all evaluation criteria.

4.6 Statistical significance analysis on adult data sets

We now present the confidence interval analysis on Adult data set to evaluate the statistical significance of the superiority of FEMI over the five existing techniques as evident from our empirical assessment. However, we first briefly explain the essence of confidence interval and its actual meaning.

Suppose that we have ten different copies of the Adult data set with different missing values, and we impute the missing values of each data set using FEMI. We thereby get ten imputed data sets. We can evaluate the imputation quality of the ten data sets through a metric such as RMSE. Therefore, we get ten RMSE values and can calculate the mean of the ten values. Let us call the mean RMSE value as the population mean. Now if we collect another set of ten copies of the Adult data set with missing values and impute them using FEMI, we are likely to get a different population mean. An obvious question is then which of the two population means is the correct one. Is any of the two population means is the same as the true mean? A true mean is the theoretical population mean obtained from the imputation of infinite number of data sets [46].

The 95 % confidence interval analysis allows us to compute an interval (i.e., an upper limit and a lower limit) around a mean value, suggesting that if we run the tests 100 times (where each test imputes say ten data sets and thereby obtain a population mean) and compute the 100 intervals from the tests then the true mean will be within the intervals for 95 tests. By running a test, we mean the collection of ten missing data sets and creating ten imputed data sets. Therefore, 95 % confidence interval tells us that the true mean falls within the interval obtained from a test with 95 % chance [46].

If we have ten data sets with missing values that are imputed by two different imputation techniques X and Y, then we get ten imputed data sets that are imputed by a technique. We can then compute the population mean and confidence interval for each technique. If the population mean of X is higher than the population mean of Y and the confidence intervals are non-overlapping, then it tells us that there is 95 % chance that the true mean of X is higher than the true mean of Y. That is, Y is clearly better than X (since the RMSE value lower the better) with 95 % probability.

We now present 95 % confidence interval analysis of FEMI with other techniques in terms of RMSE, and MAE for all 32 missing combinations in Fig. 5. It is clear from the figure that FEMI performs better (i.e., better average value and no overlap of confidence intervals) than other techniques for all missing combinations. We can see from the figures that IBLLS in general performs worse for a high missing ratios, whereas FEMI maintains almost the same performance even for a high missing ratios.

4.7 Statistical significance analysis for all data sets

We present 95 % confidence interval analysis in terms of RMSE for all seven remaining data sets as shown in Fig. 6. FEMI performs significantly better (i.e., better average value and no overlap of confidence intervals) than other five techniques in terms of RMSE, for all missing combinations in Chess (Fig. 6a), Yeast (Fig. 6b), GermanCA (Fig. 6d) and Pima (Fig. 6e) data sets. In CMC (Fig. 6c), Housing (Fig. 6f) and Autompg (Fig. 6g) FEMI performs significantly better than EMI and IBLLS for all missing combinations except for those marked by the circles.

We also present the number of overlapping cases between FEMI and other techniques for all evaluation criteria in Table 7. Out of 256 cases (i.e., 32 combinations for each data set, of 8 data sets), confidence interval of FEMI overlaps with other techniques in only 38 and 51 cases in terms of RMSE, and MAE, respectively (see Table 7).

From Fig. 6, we realize that generally overlapping happens for low missing ratio and simple missing pattern. Although GkNN, FKMI, SVR, IBLLS, and EMI generally perform worse than FEMI for all patterns, they perform comparatively better in the simple pattern than in the medium and complex patterns. However, FEMI performs almost equally good for the low and high missing ratio.

All six techniques demonstrate a similar tendency of better performance for the simple pattern than the medium and complex patterns. In the simple pattern, a record can have at most one missing value. Therefore, an imputation technique can take advantage of a higher number of available values of a record in order to impute the missing value of the record. On the other hand, in the medium and complex patterns, a record may have more than one missing

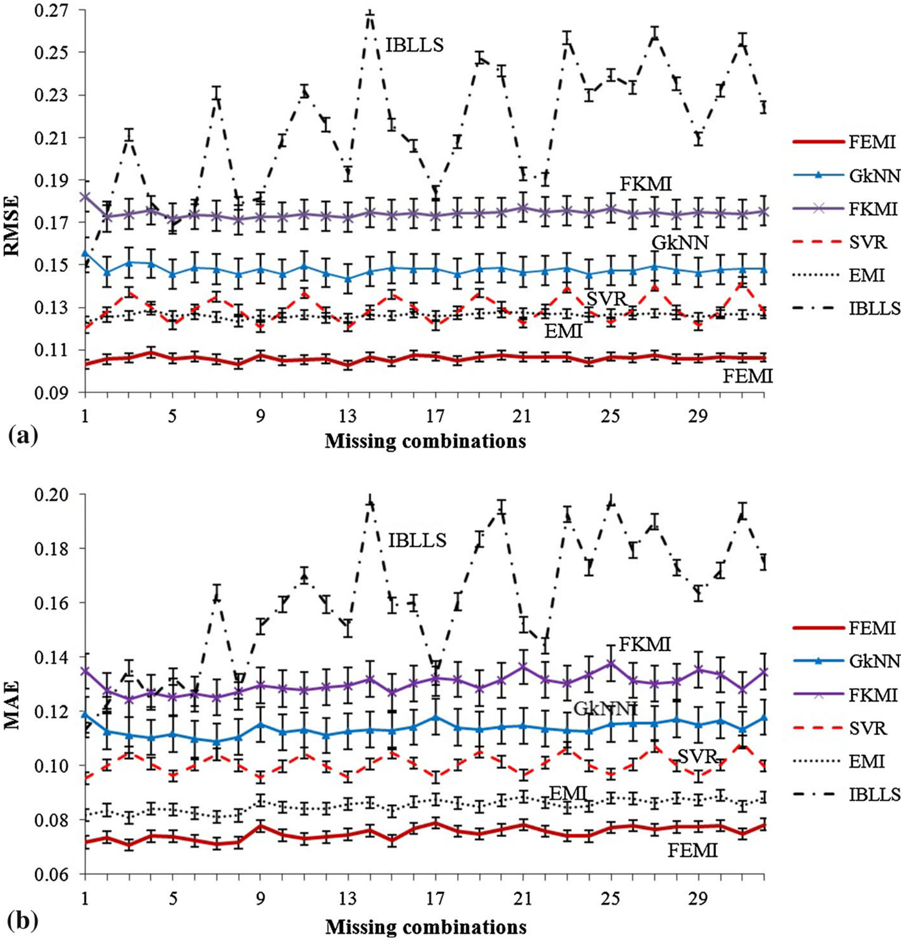

Fig. 5 95 % confidence interval analysis on Adult data set. **a** RMSE, **b** MAE

values. When a record has a big number of missing values, then an imputation technique has a lower number of available values in the record to make a more precise estimation of the missing values. An extreme example can be the case where all values of a record are missing. Naturally, the imputation accuracy drops with the increased number of missing values in a record. However, the imputation techniques generally perform better for the blended pattern (as evident from Fig. 6) due to the existence of the simple pattern inside the blended pattern as explained in Sect. 4.2.

In Fig. 7, we present the overall average values (average of all 32 combinations) for the techniques on all eight data sets. FEMI performs clearly better than five other techniques.

Figure 8 shows the percentage of combinations (out of 256 combinations for all data sets) where FEMI, GkNN, FKMI, SVR, EMI, and IBLLS perform the best. For example, FEMI performs the best in 88.67 % cases (combinations) in terms of RMSE as shown in Fig. 8a.

In Fig. 9, we now present another statistical significance analysis called the t-test for all 32 missing combinations of all data sets. Before we present the results, we first briefly explain the *t* test analysis. Like the confidence interval test, a *t* test is also used to evaluate

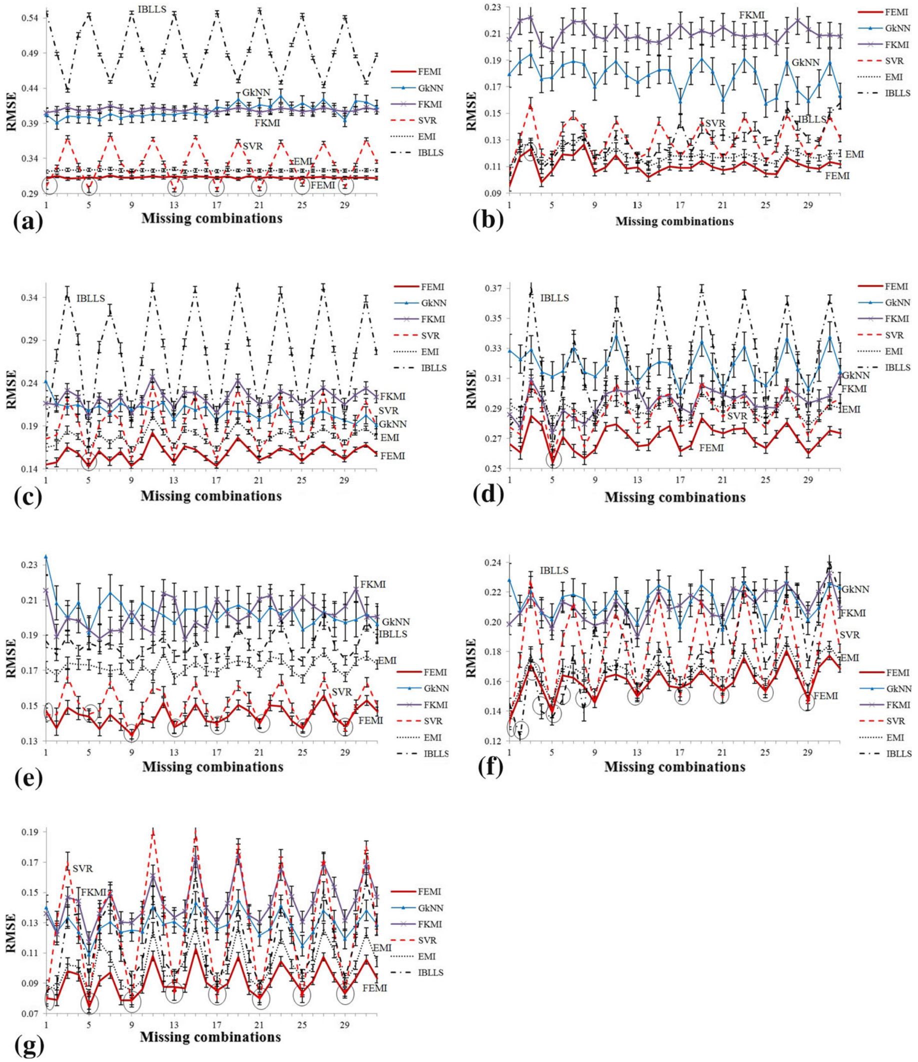

Fig. 6 95 % confidence interval analysis on Chess, Yeast, CMC, GermanCA, Pima, Housing and Autompg data sets in terms of d_2. **a** Chess data set. **b** Yeast data set. **c** CMC data set. **d** GermanCA data set. **e** Pima data set. **f** Housing data set. **g** Autompg data set

the significance of the superiority of a technique X over another technique Y [46]. Let X and Y be evaluated through RMSE. For 32 runs (i.e., the sample size $n_X = 32$) of X, we get 32 different RMSE values for X. Similarly, if we run the technique Y 32 times (i.e., the sample size $n_Y = 32$), then we get 32 different RMSE values for Y. Let $\overline{X}$ and $\overline{Y}$ be the averages of X and Y, respectively, and s_X and s_Y be the variances of X and Y, respectively. Also, let $df_X(= n_X - 1)$ and $df_Y(= n_Y - 1)$ be the degrees of freedom for X and Y, respectively. Using the averages, variances, and degrees of freedom, we can calculate a t value for X and Y as follows [2].

Table 7 The number of overlapping cases (out of the total of 256 cases for each evaluation criterion) between FEMI and other techniques in terms of 95 % confidence interval analysis

Data set	RMSE	MAE
Adult	0	0
Chess	7	10
Yeast	1	0
CMC	1	0
GermanCA	1	16
Pima	8	0
Housing	12	14
Autompg	8	11
Total (Out of 256 cases)	38	51

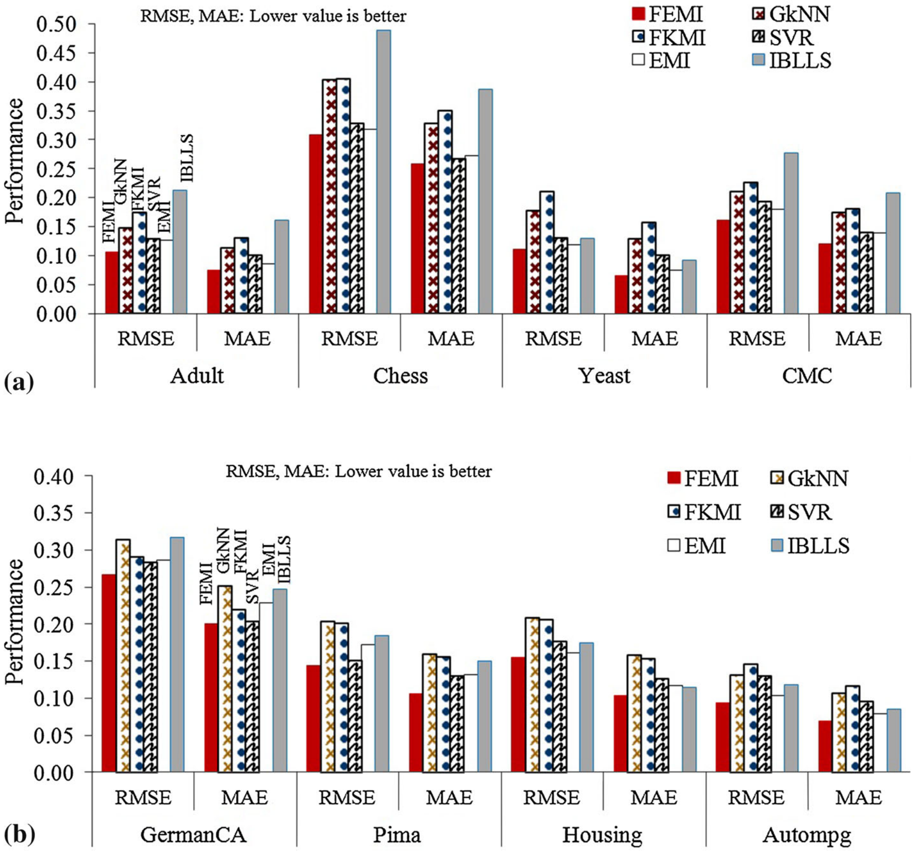

Fig. 7 Performance comparison on eight data sets. **a** Performance on Adult, Chess, Yeast and CMC data sets. **b** Performance on GermanCA, Pima, Housing and Autompg data sets

$$t_{XY} = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{s_X}{df_X} + \frac{s_Y}{df_Y}}} \tag{14}$$

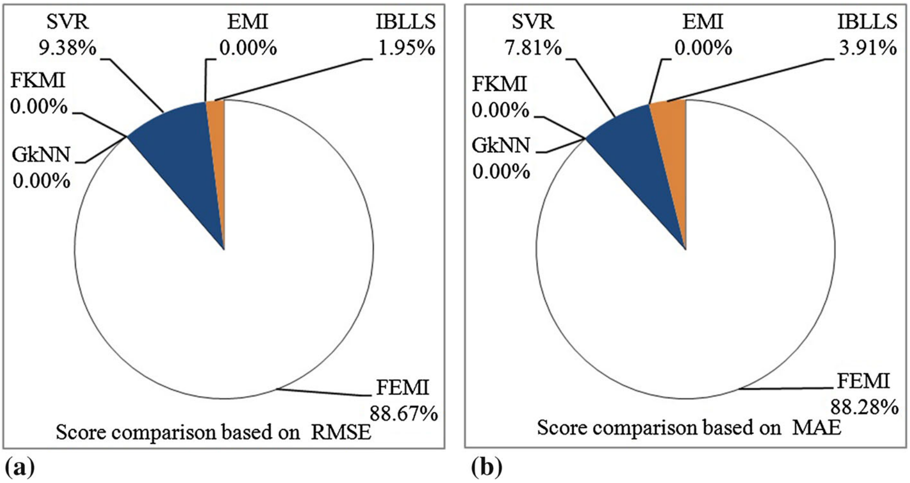

Fig. 8 Score comparison on eight data sets. **a** RMSE. **b** MAE

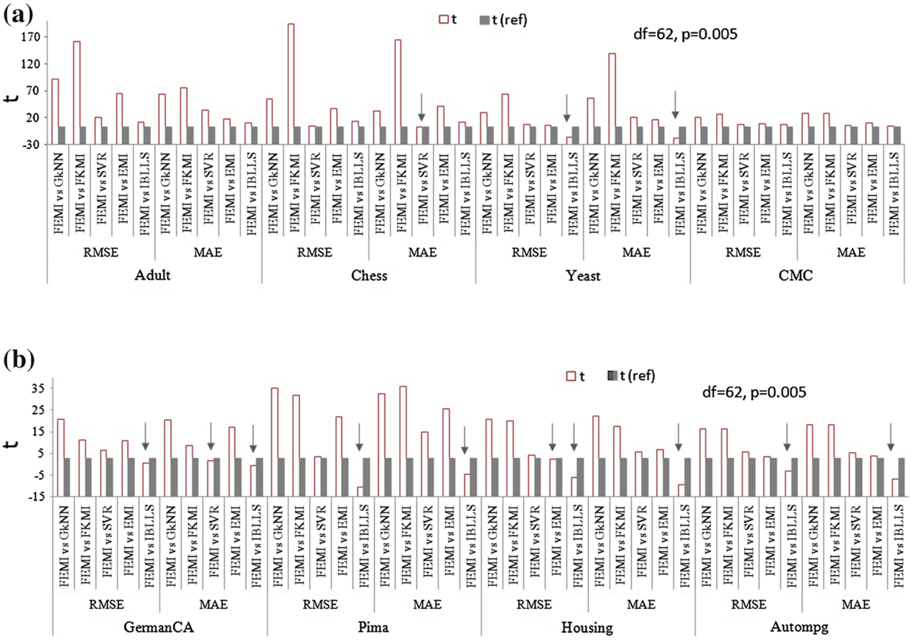

Fig. 9 *t* test analysis on eight data sets. **a** *t* test analysis on Adult, Chess, Yeast and CMC data sets. **b** *t* test analysis on GermanCA, Pima, Housing and Autompg data sets

The statistical superiority of X over Y is evaluated by comparing the t_{XY}-value with the t(ref) value. The t(ref) value can be obtained from the Student *t* distribution Table [1] through the degree of freedom and the confidence level. For example, the t(ref) value of the two-tailed test at significance level $p = 0.05$ (i.e., 95 % confidence level) and degree of freedom=31

Table 8 Average execution time (in ms) of different techniques on the eight data sets

Data set	FEMI	GkNN	FKMI	SVR	EMI	IBLLS	Machine used
Adult	1,249,433	1,075,760	1,101,046	1,475,522	82,189	53,947,274	Machine 1
Chess	48,784	11,994	114,254	410,324	8,667	15,537,849	Machine 1
Yeast	1,603	51,102	2,827	1,260	92	173,209	Machine 1
CMC	34,383	6,711	39,993	150,341	469	233,994	Machine 2
GermanCA	785	1,740	11,125	4,006	62	58,044	Machine 1
Pima	702	11,327	13,204	562	47	39,443	Machine 1
Housing	15,331	9,767	88,120	54,987	3,087	1,268,431	Machine 2
Autompg	970	1,170	2,013	167	18	8,861	Machine 1
Average	168,999	146,196	171,573	262,146	11,829	8,908,388	

Table 9 Average execution time (in ms) on CMC data set for a complex pattern (30 runs)

FEMI	EMI	IBLLS	Machine used
33,879.00	482.63	312,005.90	Machine 2

(since $n_X = n_Y = 32$) is 1.96 [1]. The performance of X is significantly better than Y if the t_{XY}-value is greater than the t ref (which is 1.96 in this case).

In Fig. 9, we present the statistical significance analysis using t test for all 32 missing combinations of all data sets. If a t_{XY} value (denoted as the t value in the figure) is greater than the t(ref) value, then it indicates the superiority of FEMI over the other techniques being evaluated. Please note that in this experimentation, we evaluate the superiority of FEMI over the existing techniques for 99.5 % confidence since we use ($p = 0.005$). Therefore, Fig. 9 demonstrates a considerably better performance of FEMI over other techniques based on all evaluation criteria for all eight data sets except for a few cases indicated by a downhead arrow in the figure.

4.8 Execution time complexity analysis

We now present the average execution time (in milliseconds) for 320 data sets (32 combinations × 10 data sets per combination) with missing values for each real data set in Table 8. We carry out the experiments using two different machines. However, for one data set, we use the same machine for all techniques. The configuration of Machine 1 is 4 × 8 core Intel E7-8837 Xeon processors, 256 GB RAM. The configuration of Machine 2 is Intel Core i5 processor with 2.67 GHz speed and 4 GB RAM. The last row of Table 8 indicates that FEMI takes less time than IBLLS, SVR, and FKMI, whereas it takes considerably more time than EMI to pay the cost of a significantly better quality imputation. Moreover, the execution time of FEMI is comparable with GkNN. However, FEMI performs significantly better than GkNN as shown in Figs. 7 and 8.

For CMC data set, we next create 30 noisy data sets for a missing combination (id=27, missing ratio=10 %, missing model = Overall, and missing pattern = Complex) and apply FEMI, EMI, and IBLLS on all data sets. We then calculate average execution time for 30 runs for each of the three techniques. FEMI requires lower time than IBLLS, but higher time than EMI (see Table 9).

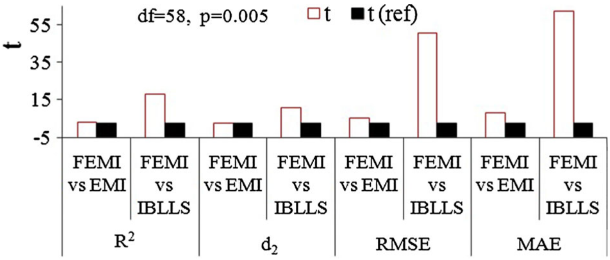

Fig. 10 *t* test analysis on CMC data set for a complex pattern (30 runs)

For the same missing combination, we also perform *t* test analysis for all four evaluation criteria as shown in Fig. 10. FEMI performs significantly better than both other techniques in terms of R^2, d_2, RMSE, and MAE.

4.9 Experimentation on categorical missing value imputation for all data sets

Unlike EMI and IBLLS, FEMI can impute categorical missing values in addition to numerical missing values. Therefore, in order to evaluate the performance of FEMI for imputing categorical values, we now compare its performance with GkNN, FKMI, and SVR that imputes categorical missing values. Figure 11 shows that FEMI achieves lower RMSE (Fig. 11a) and MAE (Fig. 11b) values than GkNN and FKMI for all eight data sets. FEMI also performs better than SVR in terms of RMSE and MAE for most of the data sets except Yeast and Autompg. For each data set, RMSE and MAE values are computed using all 32 combinations. Note that for RMSE and MAE, a lower value indicates a better imputation.

Note that for categorical attributes, FEMI provides the votes $V_{p_l}; \forall p_l \in P$ where P is the domain of the pth attribute (see Eq. 11). For the RMSE and MAE calculation in this study (see Eqs. 12 and 13), the value p_l having the highest vote is considered as the imputed value P_i of the ith missing value. For the RMSE and MAE calculation, if the imputed value (P_i) and the actual/observed value (O_i) are the same, then the distance between them $(P_i - O_i)$ is considered 0 and otherwise 1. However, if the votes for two possible values (such as $p_l \in P$ and $p_m \in P$) are high, then a user is less certain about the imputed value P_i. When the votes are close to each other, then a user is more uncertain than when only one value has the highest vote, and all other values have very low votes. The uncertainty of a user can be captured through the entropy of the votes. We considered in this study the value having the highest vote as the imputed value just for the sake of RMSE and MAE calculation. However, FEMI actually provides a user with all votes and a user can take his/her decision on the imputation.

We understand that when the observed value (O_i) does not receive the highest vote, the following two cases are not the same. In the first case, the observed value (O_i) receives a high vote, and in the second case, it receives a low vote. Obviously, the first case is better than the second case in terms of the imputation accuracy. In order to take this into consideration in the RMSE and MAE calculation, we use all votes and thereby calculate the probability of the actual value to be imputed, $p(O_i) = \frac{V_{p_m}|p_m=O_i}{\Sigma_{p_l \in P; \forall p_l} V_{p_l}}$. The best imputation is when $p(O_i) = 1$ and $p(p_l) = 0; \forall p_l \neq O_i$. Therefore, we introduce two new metrics called new RMSE (see Eq. 15) and new MAE (see Eq. (16)), where N is the total number of missing values.

$$n\text{RMSE} = \left(\frac{1}{N} \sum_{i=1}^{N} [1 - p(O_i)]^2 \right)^{\frac{1}{2}} \tag{15}$$

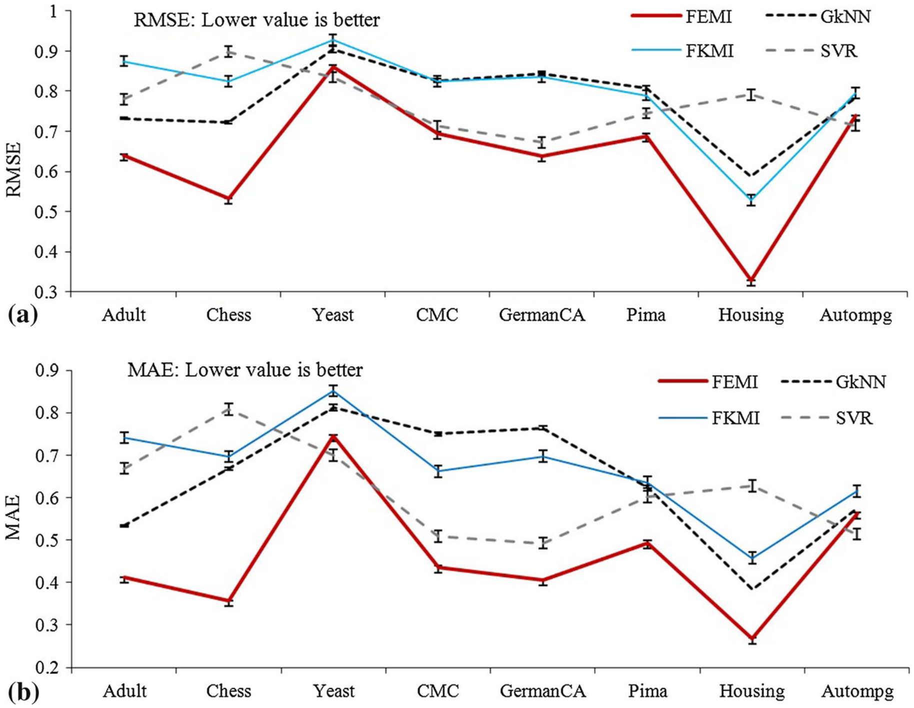

Fig. 11 Performance comparison on eight data sets in terms of categorical imputation. **a** RMSE, **b** MAE

Table 10 Overall performance comparison of FEMI and DMI for categorical imputation based on RMSE, *n*RMSE, MAE and *n*MAE

Data set	DMI RMSE	FEMI RMSE	FEMI nRMSE	DMI MAE	FEMI MAE	FEMI nMAE
CMC	0.779	0.694	0.569	0.617	0.436	0.502
Housing	0.480	0.328	0.316	0.413	0.269	0.310

$$nMAE = \frac{1}{N}\sum_{i=1}^{N} |1 - p(O_i)| \tag{16}$$

In Table 10, we present the overall average values (average of all 32 combinations) of RMSE, *n*RMSE, MAE and *n*MAE for FEMI and DMI on the CMC and Housing data sets. The results in the table indicate a strong superiority of FEMI over DMI for categorical imputation.

5 Conclusion

In this paper, we propose a framework for missing value imputation using a fuzzy clustering technique and a proposed fuzzy expectation maximization algorithm. The basic idea of the technique is to make an educated guess for a missing value using the most similar records. It takes the fuzzy nature of clustering into consideration while identifying the group of most similar records. Therefore, it considers all groups of records (clusters) as similar, with some degree of similarity. Moreover, while imputing a missing value based on a group, it also considers the fuzzy nature of all records for belonging to the group. Therefore, it uses a novel fuzzy expectation maximization algorithm to impute missing values.

The proposed technique utilizes a user-defined value for k (i.e., the number of clusters), the default value of which is $k = 20$. We use $k = 20$ in all experiments. While a data set specific k value would favor our technique, it achieves higher quality of imputation than the existing techniques even for the default k value. Moreover, in this study (see Sect. 3.4), we present a number of possible approaches for finding a suitable k value automatically. Note that the main focus of the proposed technique is imputation of missing values and not clustering the records. We are using clustering only to find a group of similar records possibly having high correlations among the attributes. Therefore, it is okay if we do not find the actual clusters very precisely (that is, the best clustering solution with the best k) as long as the clusters found contain similar records. For example, if there are actually three clusters in a data set (i.e., $k = 3$) while FEMI finds two groups of records from each cluster and thereby obtains six groups (i.e., $k = 6$), the records within each group are still similar to each other. Therefore, the imputation quality of the proposed technique should still remain high. Hence, choosing a high k value (such as $k = 20$) can be considered acceptable when the typical data sets [34] such as those used in this study have low number of clusters. Our future research plans include the automation of the k value as indicated in Sect. 3.4.

We compare the proposed technique with five other high-quality existing techniques. In our experiments, we use eight publicly available natural data sets and two evaluation criteria. The data sets used in the experiments of this study contain non-time series data. However, our technique can also be used on time series data provided the data set has two or more attributes. Two or more attributes are required to facilitate the correlation calculation needed by the fuzzy EMI technique used in FEMI. Since FEMI was not tailored for time series data, it does not take advantage of time information, while imputing missing values. Nevertheless, many imputation techniques similar to our approach have been applied on time series data [7,18,19].

The experimental results indicate that the proposed technique performs significantly better than the other five techniques. We use average values and confidence interval and t test analyses to compare the performance of our technique with other techniques. We also carry out a test on execution time complexity. While the proposed technique takes less time than IBLLS, SVR, and FKMI, it takes more time than EMI. Our future research plan is to reduce the time complexity of the technique and improve its imputation accuracy.

Acknowledgments The authors would like to thank anonymous reviewers for their valuable comments.

References

1. Distribution table: students t [online available: http://www.statsoft.com/textbook/distribution-tables/] (2012). Accessed 17 July 2012
2. Tests for significance [online available: http://www.csulb.edu/msaintg/ppa696/696stsig.htm] (2014). Accessed 12 May 2014
3. Banerjee A, Merugu S, Dhillon IS, Ghosh J (2005) Clustering with bregman divergences. J Mach Learn Res 6:1705–1749
4. Batista G, Monard M (2003) An analysis of four missing data treatment methods for supervised learning. Appl Artif Intell 17(5–6):519–533
5. Bezdek JC, Ehrlich R, Full W (1984) FCM: The fuzzy c-means clustering algorithm. Comput Geosci 10(2):191–203
6. Bilmes JA et al (1998) A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Int Comput Sci Inst 4(510):126

7. Bø TH, Dysvik B, Jonassen I (2004) Lsimpute: accurate estimation of missing values in microarray data with least squares methods. Nucleic Acids Res 32(3):e34–e34
8. Branch JW, Giannella C, Szymanski B, Wolff R, Kargupta H (2013) In-network outlier detection in wireless sensor networks. Knowl Inf Syst 34(1):23–54
9. Cai Z, Heydari M, Lin G (2006) Iterated local least squares microarray missing value imputation. J Bioinform Comput Biol 4(5):935–958
10. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2: 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm
11. Chatzis SP (2011) The fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional. Expert Syst Appl 38:8684–8689
12. Cheng K, Law N, Siu W (2012) Iterative bicluster-based least square framework for estimation of missing values in microarray gene expression data. Pattern Recognit 45(4):1281–1289. doi:10.1016/j.patcog.2011.10.012
13. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 2:224–227
14. Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the em algorithm. J R Stat Soc Ser B (Methodological) 39(1):1–38
15. Frank A, Asuncion A (2010) UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed 7 June 2012
16. Han J, Kamber M (2000) Data: mining Concepts and techniques. The Morgan Kaufmann Series in data management systems 2
17. Hido S, Tsuboi Y, Kashima H, Sugiyama M, Kanamori T (2011) Statistical outlier detection using direct density ratio estimation. Knowl Inf Syst 26(2):309–336
18. Honaker J, King G (2010) What to do about missing values in time-series cross-section data. Am J Polit Sci 54(2):561–581
19. Hourani M, El Emary IM (2009) Microarray missing values imputation methods: critical analysis review. Comput Sci Inf Syst ComSIS 6(2):165–190
20. Huang Z (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min Knowl Discov 2(3):283–304
21. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall, Inc, Englewood Cliffs NJ
22. Junninen H, Niska H, Tuppurainen K, Ruuskanen J, Kolehmainen M (2004) Methods for imputation of missing values in air quality data sets. Atmos Environ 38(18):2895–2907
23. Khoshgoftaar T, Van Hulse J (2005) Empirical case studies in attribute noise detection. In: IRI-2005 IEEE international conference on information reuse and integration, conf, 2005. IEEE, pp 211–216
24. Kim DW, Lee KH, Lee D (2004) Fuzzy clustering of categorical data using fuzzy centroids. Pattern Recognit Lett 25(11):1263–1271
25. Kim H, Golub G, Park H (2005) Missing value estimation for dna microarray gene expression data: local least squares imputation. Bioinformatics 21(2):187–198
26. Lee M, Pedrycz W (2009) The fuzzy c-means algorithm with fuzzy p-mode prototypes for clustering objects having mixed features. Fuzzy Sets Syst 160(24):3590–3600
27. Li D, Deogun J, Spaulding W, Shuart B (2004) Towards missing data imputation: a study of fuzzy k-means clustering method. Tsumoto S, Słowiński R, Komorowski J, Grzymała-Busse JW (eds) RSCTC 2004, LNAI, vol 3066. Springer, Berlin, Heidelberg, pp 573–579
28. Li L, Huang L, Yang W, Yao X, Liu A (2013) Privacy-preserving lof outlier detection. Knowl Inf Syst 42(3):579–597
29. Liu B, Xiao Y, Cao L, Hao Z, Deng F (2013) SVDD-based outlier detection on uncertain data. Knowl Inf Syst 34(3):597–618
30. Lu Y, Roychowdhury V (2008) Parallel randomized sampling for support vector machine (SVM) and support vector regression (SVR). Knowl Inf Syst 14(2):233–247
31. Luengo J, García S, Herrera F (2011) On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowl Inf Syst 32:77–108
32. Maletic J, Marcus A (2000) Data cleansing: beyond integrity analysis. In: Proceedings of the conference on information quality. Citeseer, pp 200–209
33. Oba S, Sato M, Takemasa I, Monden M, Matsubara K, Ishii S (2003) A bayesian missing value estimation method for gene expression profile data. Bioinformatics 19(16):2088–2096
34. Pham DT, Dimov SS, Nguyen C (2005) Selection of k in k-means clustering. Proc Inst Mech Eng Part C J Mech Eng Sci 219(1):103–119
35. Rahman MG, Islam MZ (2011) A decision tree-based missing value imputation technique for data preprocessing. In: Australasian data mining conference (AusDM 11), CRPIT, vol 121, pp 41–50. ACS, Ballarat, Australia. http://crpit.com/confpapers/CRPITV121Rahman.pdf

36. Rahman MG, Islam MZ (2013) Data quality improvement by imputation of missing values. In: International conference on computer science and information technology (CSIT-2013). Yogyakarta, Indonesia, pp 82–88
37. Rahman MG, Islam MZ (2013) KDMI: a novel method for missing values imputation using two levels of horizontal partitioning in a data set. In: The 9th international conference on advanced data mining and applications (ADMA 2013) Hangzhou, China
38. Rahman MG, Islam MZ (2013) Missing value imputation using decision trees and decision forests by splitting and merging records: two novel techniques. Knowl Based Syst. doi:10.1016/j.knosys.2013.08.023
39. Rahman MG, Islam MZ (2013) A novel framework using two layers of missing value imputation. In: Australasian data mining conference (AusDM 13), CRPIT, vol 146. ACS, Canberra, Australia
40. Rahman MG, Islam MZ, Bossomaier T, Gao J (2012) Cairad: a co-appearance based analysis for incorrect records and attribute-values detection. In: The 2012 international joint conference on neural networks (IJCNN). IEEE, Brisbane, Australia, pp 1–10. doi:10.1109/IJCNN.2012.6252669
41. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
42. Rubin D (1976) Inference and missing data. Biometrika 63(3):581–592
43. Schneider T (2001) Analysis of incomplete climate data: estimation of mean values and covariance matrices and imputation of missing values. J Clim 14(5):853–871
44. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222
45. Sun H, Wang S, Jiang Q (2004) Fcm-based model selection algorithms for determining the number of clusters. Pattern Recognit 37(10):2027–2037
46. Triola MF, Goodman WM, LaBute G, Law R, MacKay L (2006) Elementary statistics. Pearson/Addison-Wesley, Reading, MA
47. Tseng S, Wang K, Lee CI (2003) A pre-processing method to deal with missing values by integrating clustering and regression techniques. Appl Artif Intell 17(5–6):535–544
48. Wang H, Wang S (2010) Mining incomplete survey data through classification. Knowl Inf Syst 24(2):221–233
49. Wang X, Li A, Jiang Z, Feng H (2006) Missing value estimation for dna microarray gene expression data by support vector regression imputation and orthogonal coding scheme. BMC Bioinform 7(1):32
50. Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Philip SY et al (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14(1):1–37
51. Zhang, C., Qin, Y., Zhu, X., Zhang, J., Zhang, S.: Clustering-based missing value imputation for data preprocessing. In: 2006 IEEE international conference on industrial informatics. IEEE, pp 1081–1086 (2006)
52. Zhang S (2011) Shell-neighbor method and its application in missing data imputation. Appl Intell 35(1):123–133
53. Zhang S (2012) Nearest neighbor selection for iteratively k-nn imputation. J Syst Softw 85(11):2541–2552
54. Zhang S, Jin Z, Zhu X (2011) Missing data imputation by utilizing information within incomplete instances. J Syst Softw 84(3):452–459
55. Zhu X, Zhang S, Jin Z, Zhang Z, Xu Z (2011) Missing value estimation for mixed-attribute data sets. IEEE Trans Knowl Data Eng 23(1):110–121

Md. Geaur Rahman is a PhD candidate in the School of Computing and Mathematics, Charles Sturt University, Australia. He received the B.Sc. (Honors) and M.Sc. degree in Computer Science and Engineering from Rajshahi University, Bangladesh, in 2001 and 2002, respectively. He has been working as an Assistant Professor at the Department of Computer Science and Mathematics, Bangladesh Agricultural University, Bangladesh, since 2005. His research interests include data cleansing/preprocessing, data mining and knowledge discovery, missing value analysis, machine learning, pattern recognition, and neural networks.

Md Zahidul Islam is a Senior Lecturer in Computer Science in the School of Computing and Mathematics, Charles Sturt University, Australia. He received his Bachelors degree in Engineering from Rajshahi University of Engineering and Technology, Bangladesh, Graduate Diploma in Information Science from the University of New South Wales, Australia, and PhD in Computer Science from the University of Newcastle, Australia. His main research interests include data preprocessing and cleansing, various data mining algorithms, applications of data mining techniques, privacy issues related to data mining, and privacy preserving data mining.