A Decision Tree-based Missing Value Imputation Technique for Data Pre-processing

Md. Geaur Rahman and Md. Zahidul Islam
Centre for Research in Complex System, School of Computing and Mathematics, Charles Sturt University, Locked Bag 588, Boorooma Street, Wagga Wagga, NSW 2678, Australia
{grahman,zislam}@csu.edu.au

Abstract
Data pre-processing plays a vital role in data mining for ensuring good quality of data. In general data pre-processing tasks include imputation of missing values, identification of outliers, smoothing out of noisy data and correction of inconsistent data. In this paper, we present an efficient missing value imputation technique called DMI, which makes use of a decision tree and expectation maximization (EM) algorithm. We argue that the correlations among attributes within a horizontal partition of a data set can be higher than the correlations over the whole data set. For some existing algorithms such as EM based imputation (EMI) accuracy of imputation is expected to be better for a data set having higher correlations than a data set having lower correlations. Therefore, our technique (DMI) applies EMI on various horizontal segments (of a data set) where correlations among attributes are high. We evaluate DMI on two publicly available natural data sets by comparing its performance with the performance of EMI. We use various patterns of missing values each having different missing ratios up to 10%. Several evaluation criteria such as coefficient of determination ($R^2$), Index of agreement ($d_2$) and root mean squared error (RMSE) are used. Our initial experimental results indicate that DMI performs significantly better than EMI.

Keywords: Data pre-processing, Data cleansing, Missing value imputation, Decision tree algorithm, EM algorithm.

1 Introduction
Organisations are extremely dependant nowadays on data collection, storage and analysis for various decision-making processes. Data are collected in various ways such as paper based and online surveys, interviews, and sensors (Chapman 2005, Cheng, Chen and Xie 2008, Apiletti, Bruno, Ficarra and Baralis 2006). For example, temperature, humidity, and wind speed data in a habitat monitoring system (HMS) are often acquired through different sensors. Due to various reasons including human error and misunderstanding, equipment malfunctioning, and introduction of noise during transformation and propagation data can often be lost or perturbed. For example, data in the HMS can be lost due to limited battery power of sensing devices, and other electro-mechanical problems of sensors.

Data anomalies and impurities can cause inefficient data analyses, inaccurate decisions and user inconveniences. Careless use of erroneous data can be misleading and damaging making it useless for the users (Muller and Freytag 2003, Abbas and Aggarwal 2010, Han and Kamber 2006). For instance, wrong quality data in genomic databases can have serious impact on end users. Errors in genome data can result in inaccurate outcomes from biological and pharmaceutical experiments costing billions of dollars for the pharmaceutical companies for developing only a few useful drugs. Hence, it is of great importance to have high quality data for safety critical data analyses (Muller, Naumann and Freytag 2003, Hensley 2002).

Therefore, a data pre-processing framework is crucial to deal with inaccurate and incomplete data for ensuring high quality data through effective data cleansing. One important task in data pre-processing is the imputation of missing values as accurately as possible. In the last few years a number of imputation methods have been proposed (Tseng, Wang and Lee 2003, Zhang, Qin, Zhu and Zhang 2006, Junninen, Niska, Tuppurainen, Ruuskanen and Kolehminen 2004, Schneider 2001, Dempster, Laird and Rubin 1977, Dellaert 2002, Li, Zhang and Jiang 2005, Pyle 1999, Little and Rubin 1987).

Generally imputation performance heavily depends on the selection of a suitable technique (Zhang et al. 2006). Different imputation techniques perform well on different types of data sets and missing values. Existing imputation techniques therefore have room for further improvement.

We argue that since EMI algorithm relies on the correlations among the attributes while imputing missing values, it performs better on a data set having high correlations among the attributes. Correlations among the attributes are natural properties of a data set and they cannot be improved or modified. However, we realise that it is often possible to have horizontal segments within a data set where there are higher correlations than the correlations over the whole data set.

This work was supported by the second author’s CRiCS seed grant.

Copyright © 2011, Australian Computer Society, Inc. This paper appeared at the 9th Australasian Data Mining Conference (AusDM 2011), Ballarat, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 121. Peter Vamplew, Andrew Stranieri, Kok-Leong Ong, Peter Christen and Paul Kennedy, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

41
For example, consider a sample data set containing two attributes: age and height (Figure 1). For the whole data set (Figure 1a) the correlation among the attributes is very low since height of a person always does not increase with age. However, age and height are highly correlated for people younger than say 17 years. Therefore, if we partition the whole example data set into two segments where one segment contains records for all people who are younger than 17 years and the other segment contains the remaining records then we have a higher correlation among age and height in Partition 1 of Figure 1b. Therefore, the identification of the horizontal segments having high correlations and application of EMI algorithm within the segments is expected to produce a better imputation result.

In this paper, we propose a novel hybrid imputation technique called DMI that makes use of an existing decision tree algorithm such as C4.5 (Quinlan 1993, Quinlan 1996, Kotsiantis 2007), and an Expectation Maximisation (EM) based imputation technique called EMI (Junninen et al. 2004, Schneider 2001, Dempster et al. 1977, Dellaert 2002) for data sets having both numerical and categorical attributes. In order to impute categorical missing values our technique uses a decision tree algorithm. However, for numerical missing values our technique with the help of a decision tree algorithm first identifies horizontal segments of records having high correlations among the attributes and then applies the EM algorithm within various horizontal segments.

We evaluate our technique (DMI) on two publicly available natural data sets by comparing its performance with the performance of EMI. We first prepare a data set having no natural missing values. We then generate data sets with artificial missing values (the original values for which are known to us) using various patterns of missing values such as simple, medium, complex and blended. In a simple pattern a record can have at most one missing value, whereas in a medium pattern if a record has any missing values then it has minimum 2 attributes with missing values and maximum 50% of the attributes with missing values. Similarly a record having missing values in a complex pattern has min 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 (Junninen et al. 2004).

In this paper, we also use different missing ratios ranging from 1% to 10% of total attribute values of a data set. Moreover, we use two missing categories/models called uniformly distributed (UD) missing values and overall missing values. In UD each attribute has the same number of missing values, whereas in an overall category an attribute can have higher number of missing values than the number of missing values in another attribute. Moreover, several well known evaluation criteria such as coefficient of determination ($R^2$), Index of agreement ($d_2$) and root mean squared error (RMSE) are used. Our experimental results indicate that DMI performs significantly better than EMI.

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

2 Background Study

Many missing value imputation methods have been proposed recently (Tseng et al. 2003, Zhang et al. 2006, Junninen et al. 2004, Schneider 2001, Dempster et al. 1977, Dellaert 2002, Li et al. 2005, Pyle 1999, Little and Rubin 1987). However, most of the existing techniques are not suitable for a data set having both numerical and categorical attributes (Tseng et al. 2003).

A simple technique is to impute a missing value of an attribute by the mean of all values of the attribute (Schneider 2001). Several missing value imputation techniques such as Nearest Neighbour (NN), Linear Interpolation (LIN), Cubic Spline Interpolation, Regression based Expectation Maximization (REGEN) imputation, Self-organizing Map (SOP) and Multilayer Perceptron (MLP) have been proposed in the literature (Junninen et al. 2004). Hybrid models such as LIN+MLP and LIN+SOP have also been presented by Junninen et al. (2004) in order to improve the imputation accuracy. Their experimental results show a slight improvement due to the use of hybrid methods. Moreover, it was pointed out that a single imputation method has several limitations while a combination of a few imputation techniques can improve accuracy significantly.

Two interesting imputation methods, among many existing techniques, are EM algorithm (Dempster et al. 1977) and Kernel Function (Zhang et al. 2006). We discuss the techniques as follows.

For imputing numerical missing values of a data set EM algorithm relies on mean and covariance matrix of the data set. First the mean and covariance matrix are estimated from a data set having some missing values. The process of estimating the mean and covariance matrix continues recursively until we get a mean and covariance matrix having difference with the previous mean and covariance matrix under user defined thresholds.

In the imputed data set $x_m = \mu_m + (x_a - \mu_a)B + e$ (1)

Where $x_m$ and $x_a$ are vectors of missing and available values of a record $x$, respectively. Moreover, $\mu_m$ and $\mu_a$ are the mean vectors of missing values and available values, respectively. $B = \Sigma_m^{-1} \Sigma_{am}$ is a regression coefficient matrix, which is the product of the inverse of covariance matrix of available attribute values ($\Sigma_{aa}$), and cross covariance matrix of available and missing values ($\Sigma_{am}$). Besides, $e$ is a residual error with mean zero and unknown covariance matrix (Schneider 2001).

Using the imputed data set EM algorithm again estimates the mean and covariance matrix. The process of imputing missing values, and estimating mean and covariance matrix continues recursively until we get a mean and covariance matrix having difference with the previous mean and covariance matrix under user defined thresholds.

Kernel imputation method for numerical missing values was originally proposed by Wang and Rao (2002) and later on discussed by Zhang et al. (2006). In the technique the mean, covariance matrix and Gaussian kernel function of a data set are used to impute missing
values. Initially all missing values belonging to an attribute are imputed by the mean value of the attribute. Let $d$ be the number of attributes, $n$ be the number of records of a data set $D$ and $m$ be the total number of missing values in the whole data set. An attribute $A_j$ ($1 \leq j \leq d$) may have missing values for more than one record. All missing values belonging to an attribute $A_j$, $\forall j$ are first imputed using its average value.

At this stage the originally missing values are again imputed one by one considering only one imputed value as missing at a time. All other values including the remaining ($m-1$) imputed values are considered as non-missing. A missing value $A_{ij}$ of an attribute $A_j$ and record $R_i$ is calculated as follows (Zhang et al. 2006).

$$A_{ij} = m(A_{i1}, A_{i2}, ..., A_{id}; \forall A_{ik} \neq A_{ij}) + \epsilon_i$$  \hspace{1cm} (2)

where $m(A_{i1}, A_{i2}, ..., A_{id}; \forall A_{ik} \neq A_{ij})$ is computed as follows.

$$m(A_{i1}, A_{i2}, ..., A_{id}; \forall A_{ik} \neq A_{ij}) = \frac{\sum_{a=1}^m \delta_{a} \sum_{k=1}^{d} f_{A_{ik}} \prod_{j=1}^{d} K(\frac{A_{ij} - A_{ij}}{\sigma})}{\sum_{a=1}^m \delta_{a} \prod_{j=1}^{d} K(\frac{A_{ij} - A_{ij}}{\sigma}) + m - 2}$$  \hspace{1cm} (3)

where $\delta_a$ is either 0 or 1. If $A_{ij}$ is missing then $\delta_a=0$, and otherwise $\delta_a=1$. $K(x) = \frac{1}{\sqrt{2\pi}} \exp \left(-\frac{x^2}{2}\right)$ is the Gaussian kernel function, and $h$ is typically considered as a constant value say 5 (Zhang et al. 2006).

### 3 Our Technique

We present a novel “Decision tree based Missing value Imputation technique” (DMI) which makes use of an EM algorithm and a decision tree (DT) algorithm. We first introduce the main ideas of the proposed technique as follows before we introduce it in detail.

EM based imputation techniques rely on the correlations among the attributes in a data set. We realised that the imputation accuracy is supposed to be high for a data set having high correlations among the attributes. Besides correlations among the attributes are natural properties of a data set and they cannot be improved or modified for the sake of achieving better imputation accuracy. However, we argue that correlations among the attributes can be higher within a horizontal partition of a data set than within a whole data set as shown in Figure 1.

Figure 5a gives an example of a decision tree obtained from a toy data set shown in Figure 4b. The nodes of the tree are shown as the rectangles and the leaves are shown as the ovals (Figure 5a). In each node a tree tests an attribute which we call the “test attribute” of the node. A decision tree divides the records of a data set into a number of leaves, where ideally all records belonging to a leaf have the same class value. However, in reality a leaf may contain records where majority of them have the same class value, but a few of them can have a different class value/s. Such a leaf is generally known as heterogeneous leaf. The class values in a heterogeneous leaf are considered to be similar to each other (Estivill-Castro and Brankovic 1999, Islam and Brankovic 2003). Moreover, the records belonging to a leaf are considered to be a cluster as they are shown to be similar to each other as well (Islam and Brankovic 2011, Islam 2008). The main justifications for considering the records similar to each other are that they share the same or similar class value/s, same values for all categorical test attributes for the leaf and similar values for all numerical test attributes for the leaf. For example, the records in the Leaf 8 of Figure 5c have the same class value (“7-10”) and the same test attribute value (“a12”). Another justification for considering the records to be similar is that the entropy of the class values for the records within a leaf is the minimum (Islam and Brankovic 2011, Islam 2008).

Therefore, we argue that attribute correlations within the records belonging to a leaf are likely to be higher than attribute correlations within a whole data set. We test attribute correlations for the records within a leaf and for all records of “Credit Approval” data set (UCI Repository). Applying C4.5 algorithm (Quinlan 1993, 1996) on Credit Approval data set we build a decision tree that has seven leaves. We then prepare seven correlation matrices, each for the records within a leaf. We also prepare a correlation matrix for all records. We observe that correlations among attributes within a leaf are generally higher than within the whole data set. Considering all seven leaves, on an average 66% of the correlations among the attributes have higher values within the records of a leaf than the correlation values that are calculated for all records. Figure 2 shows two correlation matrices for the six numerical attributes of Credit Approval data set. For six numerical attributes there are 15 correlations among the attributes. Only 3 out of 15 correlations have lower values in Figure 2b than the corresponding correlation values in Figure 2a.

Figure 2: Correlation matrix for the six numerical attributes of Credit Approval data set.

- **(a)** Full data set.
- **(b)** Within a leaf.

<table>
<thead>
<tr>
<th></th>
<th>1.00</th>
<th>0.23</th>
<th>0.41</th>
<th>0.19</th>
<th>-0.09</th>
<th>0.02</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>1.00</td>
<td>0.30</td>
<td>0.37</td>
<td>0.21</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>0.23</td>
<td>0.30</td>
<td>1.00</td>
<td>0.33</td>
<td>-0.06</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>0.41</td>
<td>0.30</td>
<td>1.00</td>
<td>0.33</td>
<td>-0.06</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>0.19</td>
<td>0.27</td>
<td>0.33</td>
<td>1.00</td>
<td>-0.12</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>-0.09</td>
<td>-0.21</td>
<td>-0.06</td>
<td>-0.12</td>
<td>1.00</td>
<td>0.07</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td>0.12</td>
<td>0.05</td>
<td>0.06</td>
<td>0.07</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

**Figure 3: The overall block diagram of our DMI.**

We first mention the main steps of DMI as follows and then explain each of them in detail.

- **Step-1:** DMI divides a full data set ($D_F$) into two sub data sets ($D_C$ (having only records without missing values) and $D_I$ (having only records with missing values)).

**Step-2:** Build a set of decision trees on $D_C$ considering the attributes, having missing values in $D_I$, as the class attributes.

**Step-3:** Assign each record of $D_I$ to the leaf where it falls in for the tree that considers the attribute, which has a missing value for the record, as the class attribute. If the record has more than one attributes with missing values it will be assigned to more than one leaves.
Step-4: Impute numerical missing values using EM algorithm and categorical missing values using majority class values within the leaves.

Step-5: Combine records to form a completed data set \( (D_F) \) without any missing values.

The overall block diagram of DMI is shown in Figure 3.

Step-1: DMI divides a full data set \( (D_F) \) into two sub data sets \( D_C \) (having only records without missing values) and \( D_I \) (having only records with missing values).

To impute missing values in a data set, we first divide the data set \( D_F \) into two sub data sets \( D_C \) and \( D_I \), where \( D_C \) contains records having no missing values and \( D_I \) contains records having missing values as shown in the DMI algorithm (Step-1 of Figure 8). For example, Figure 4 shows an example data set \( D_F \), sub data set \( D_C \), and sub data set \( D_I \). The data set \( D_F \) has 9 records and 4 attributes out of which two are numerical and two are categorical. Two records (R3 and R5) have missing values for attributes C3 and C4, respectively. Therefore, we first move records R3 and R5 from \( D_F \) to sub data set \( D_I \) (Figure 4c) and the remaining records into sub data set \( D_C \) (Figure 4b).

(a) A sample data set \( D_F \).

(b) Sub data set \( D_C \).

(c) Sub data set \( D_I \).

Figure 4: A sample data set \( D_F \) with sub data sets \( D_C \) and \( D_I \).

Step-2: Build a set of decision trees on \( D_C \) considering the attributes, having missing values in \( D_I \), as the class attributes.

In this step we first identify attributes \( A_i \) \((1 \leq i \leq M)\) where \( M \) is the total number of attributes, in \( D_I \), having missing values. We make a temporary copy of \( D_C \) as \( D_F \). For each attribute \( A_i \) we build a tree using the C4.5 decision tree algorithm for the sub data set \( D_C \) considering \( A_i \) as class attribute. If \( A_i \) is a numerical attribute, we first generalize \( A_i \) of \( D_C \) into \( N_C \) categories, where \( N_C \) is the squared root of the domain size of \( A_i \) (Step-2 of Figure 8).

Step-3: Assign each record of \( D_I \) to the leaf where it falls in for the tree that considers the attribute, which has a missing value for the record, as the class attribute. If the record has more than one attributes with missing values it will be assigned to more than one leaves.

Each record \( r_k \) from the sub data set \( D_I \) has missing values. If \( r_k \) has a missing value for the attribute \( A_1 \) then we use the tree \( \mathcal{T}_1 \) (that considers \( A_1 \) as the class attribute) in order to indentify the leaf \( L_1 \), where \( r_k \) falls in. Note that \( r_k \) falls in \( L_1 \) if the test attribute values of \( r_k \) match the attribute values of \( r_k \). We add \( r_k \) in the data set \( D_I \) representing \( R_1 \) of \( \mathcal{T}_1 \) (Step 3 of Figure 8). If \( r_k \) has more than one attributes with missing values then it is added in more than one data sets.

In this step, \( D_I \) represents a sub data set containing all records of the \( j^{th} \) leaf/logic rule. In our example, attribute C3 of record R3 is missing (Figure 4c). In order to assign the record R3 into a leaf, the tree shown in Figure 5a is the target tree since it considers C3 as the class attribute. Matching the test attribute values with the attribute values of R3 we assign R3 in Leaf 2 of Figure 5a. The resultant leaf records are shown in Figure 6a. Similarly, we assign R5 into Leaf 10 of Figure 5c which is shown in Figure 6b.

Step-4: Impute numerical missing values using EM algorithm and categorical missing values using majority class values within the leaves.

We impute the missing values for all records in \( D_I \) one by one. For a record \( r_k \) we identify an attribute \( A_i \) having a missing value. We also identify the data set \( D_j \) where \( r_k \) has been added (in Step 3) for imputation of missing value in the attribute \( A_i \). If \( D_j \) has not been imputed before (i.e. if \( L_j \) is FALSE) then we apply either EM algorithm or DT based imputation depending on the type (numerical or categorical) of the attribute \( A_i \) and thereby impute the values of \( A_i \) in \( D_j \). Finally we update the attribute value for \( A_i \) of \( r_k \) in \( D_I \) from the imputed attribute value for \( A_i \) of \( r_k \) in \( D_j \). We continue the process for all \( r_k \) in \( D_I \) and thereby impute all missing values for all \( r_k \) in \( D_F \) (Step-4 of Figure 8).
In our example, we impute the attribute C3 of record R3 (Figure 6a) using the class value “a32” of Leaf 2 (Figure 5a). We apply EM algorithm on the data set belonging to Leaf 10 (Figure 5c and Figure 6b) to impute the value of attribute C4 of record R5. Figure 7 shows the data sets after imputation.

![Figures](image)

**Figure 7: Impute missing values and combine records to form a completed data set (D_f).**

There are two problematic cases where EM algorithm needs to be used carefully in order to get a proper imputation result. First, EM algorithm does not work if all records have the same value for a numerical attribute. Second, EM algorithm is also not useful when we have all numerical values missing in a record. DMI initially ignores the attribute having same value for all records. It also ignores the records having all numerical values missing. DMI then imputes all others values as usual. Finally, it uses the mean value of an attribute to impute a numerical missing value for a record having all numerical values missing. It also uses the mean value of an attribute to impute a missing value belonging to an attribute having the same value for all records.

**Step-5: Combine records to form a completed data set (D_f).**

We finally combine D_c and D_l in order to form D_f which is the imputed data set. In our example, we integrate sub data sets of Figure 4b and Figure 7a to represent the complete data set D_f shown in Figure 7b.

**4 Experimental Result**

We implement both our novel technique DMI and an existing technique EMI (Junninen et al., 2004), which is an EM based imputation over all records of a data set. We apply the techniques on two real life data sets, namely Adult and Credit Approval data set. The data sets are publicly available in UCI Machine Learning Repository (UCI Repository).

The Adult data set contains census information of United States and has 32561 records with 15 attributes including the natural class attribute. There are all together 9 categorical attributes, and 6 numerical attributes. We remove all records having missing values, and thereby work on 30162 records with no missing values. On the other hand, the Credit Approval data set contains information about credit card applications and has 690 records with 16 attributes including the class attribute. There are 10 categorical attributes, and 6 numerical attributes. There are also a number of records having missing values. We remove all records with missing values, and thereby produce 653 records with no missing values.

Note that the missing values that naturally exist in the data sets are first removed to prepare a data set without any missing values. We then artificially create missing values which are imputed by the different techniques. Since the original values of the artificially created missing data are known to us we can easily evaluate the efficiencies of the imputation techniques.

![Algorithm](image)

**Algorithm: DMI**

**Input:** A data set D_F having missing values

**Output:** A data set D_F with all missing values imputed

**Method:**

- D_c ← ø.  //Sub data set having records without missing values
- D_l ← ø.  //Sub data set having records with missing values
- L ← ø.  //Total no. of leaves of all trees

**Step-1:** Divide data set D_F as follows

- D_c ← all missing-valued records of D_F.
- D_l ← D_F - D_c.

**Step-2:** Find A_i, i = 1, ..., M, where M is the total no. of attributes, in D_l, having missing values.

For all attributes A_i ∈ A = {A_1, A_2, ..., A_M} do

1. Set D_c ← D_c.
2. If A_i is numerical then
   1. Find no. of categories N_c ← √|A_i| where |A_i| is the domain size of A_i.
   2. Generalize values of A_i for all records in D_c into N_c categories.
   3. Call C4.5 to build a tree T_i from D_c considering A_i as class attribute.
   4. Set S_i ← No. of leaves of T_i.
   5. For j = L to L + S_i do
      1. Define logic rule R_j from T_i.
      2. Generate data set d_j having all records belonging to R_j.
      3. Set l_j ← FALSE.
   6. End for
   7. L ← L + S_i.
   8. End for
3. Else if A_i is categorical then
   1. Impute A_i using EM algorithm on d_j.
   2. Else if A_i is categorical then
      1. Impute A_i from R_j considering the majority class value of d_j as the imputed value.
   3. End if
   4. If l_j is TRUE then
      1. Update A_i of r_k in D_l from A_i of r_k in d_j.
   5. End if
   6. End if
5. End for

**Step-3:** For each record r_k ∈ D_l do

For attributes A_i ∈ A do

1. If A_i is numerical and missing then
   1. Find the data set d_j for the logic rule R_j belonging to the tree T_i where the record r_k falls in R_j.
   2. d_j ← d_j ∪ r_k.
   3. End if
   4. End for
5. End for

**Step-4:** For each record r_k ∈ D_l do

For attributes A_i ∈ A do

1. If A_i is missing then
   1. Find d_j.
   2. If l_j is FALSE then
      1. If A_i is numerical then
         1. Impute A_i using EM algorithm on d_j.
      2. Else if A_i is categorical then
         1. Impute A_i from R_j considering the majority class value of d_j as the imputed value.
      3. End if
   3. End if
   4. End if
   5. End if
5. End for

**Step-5:** Complete data set D_f ← D_c ∪ D_l.

Return D_f.

**Figure 8:** The DMI missing values imputation algorithm.
An imputation performance does not only depend on the amount of missing data, but also depends on the characteristics of missing data patterns (Junninen et al. 2004). For example, in one scenario (pattern) 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. Note that both data sets may have the same number of total missing values.

Typically the probability of a value being missing does not depend on the missing value itself (Rubin 1976, Schneider 2001) and hence missing values often can have a random nature which can be difficult to formulate. Therefore, we use various patterns of missing values such as simple, medium, complex and blended. In a simple pattern a record can have at most one missing value, whereas in a medium pattern a record can have missing values for number of attributes ranging from 2 to 50% of the total number of attributes. Similarly a record having missing values in a complex pattern has minimum 50% and maximum 80% attributes having 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 patterns, we use different missing ratios (1%, 3%, 5% and 10% of total attribute values of a data set) as shown in Table 1.

We also 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.

For each combination of Missing Pattern, Missing Ratio and Missing Model we create 10 data sets with missing values. For example, for the combination having “simple” missing pattern, “1%” missing values and “overall” model (see Table 2) we generate 10 data sets with missing values. We therefore create all together 320 data sets (32 combinations x 10 data sets). For each natural data set we perform Adult and Credit Approval.

Imputation accuracy is evaluated using a few well known evaluation criteria namely co-efficient of determination ($R^2$), Index of agreement ($d_2$) and root mean squared error (RMSE).

Table 1: Settings of missing data simulation.

<table>
<thead>
<tr>
<th>Missing Pattern</th>
<th>No. of attributes having missing values</th>
<th>Missing Ratios</th>
<th>Missing Model</th>
<th>No. of simulations for each pattern combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>1</td>
<td>1%</td>
<td>Overall and Uniformly distributed (UD)</td>
<td>10</td>
</tr>
<tr>
<td>Medium</td>
<td>2</td>
<td>3%, 5% and 10%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complex</td>
<td>50%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blended</td>
<td>simple 25%, medium 50% and complex 25%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Average performance of DMI and EMI on Adult data set.

<table>
<thead>
<tr>
<th>Missing Pattern</th>
<th>$R^2$ (higher value is better)</th>
<th>$d_2$ (higher value is better)</th>
<th>RMSE (lower value is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>Simple 0.559</td>
<td>0.156</td>
<td>0.752</td>
</tr>
<tr>
<td>Medium</td>
<td>0.470</td>
<td>0.444</td>
<td>0.433</td>
</tr>
<tr>
<td>Complex</td>
<td>0.414</td>
<td>0.395</td>
<td>0.388</td>
</tr>
<tr>
<td>Blended</td>
<td>0.414</td>
<td>0.395</td>
<td>0.388</td>
</tr>
</tbody>
</table>

Table 3: Average performance of DMI and EMI on Credit Approval data set.

<table>
<thead>
<tr>
<th>Missing Pattern</th>
<th>$R^2$ (higher value is better)</th>
<th>$d_2$ (higher value is better)</th>
<th>RMSE (lower value is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>Simple 0.469</td>
<td>0.100</td>
<td>0.766</td>
</tr>
<tr>
<td>Medium</td>
<td>0.470</td>
<td>0.444</td>
<td>0.433</td>
</tr>
<tr>
<td>Complex</td>
<td>0.414</td>
<td>0.395</td>
<td>0.388</td>
</tr>
<tr>
<td>Blended</td>
<td>0.414</td>
<td>0.395</td>
<td>0.388</td>
</tr>
</tbody>
</table>

We now define the evaluation criteria briefly. Let $N$ be the number of artificially created missing values, $Q_i$ be the actual value of the $i$th artificially created missing value ($1 \leq i \leq N$), $P_i$ be the imputed value of the $i$th missing value, $\bar{Q}$ be the average of actual values $Q_i, \forall i \in N$. Let $\bar{P}$ be the average of the imputed values, $\sigma_Q$ be the standard deviation of the actual values and $\sigma_P$ be the standard deviation of the imputed values.
The most commonly used imputation performance indicator is the coefficient of determination ($R^2$), which determines the degree of correlations between actual and imputed values. It varies between the range 0 and 1, where 1 indicates a perfect fit.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (p_i - \hat{p}_i)^2}{\sum_{i=1}^{N} (p_i - \bar{p})^2}$$  \hspace{1cm} (4)

The index of agreement (Willmott 1982) determines the degree of agreement between actual and imputed values. Its value ranges between 0 and 1. Higher value indicates a better fit.

$$d = 1 - \frac{\sum_{i=1}^{N} (p_i - \hat{q}_i)^2}{\sum_{i=1}^{N} (|p_i - \bar{q}| + |q_i - \bar{q}|)^2}$$  \hspace{1cm} (5)

where $k$ is either 1 or 2. The index ($d_2$) has been used throughout this experiment with $k$ equal to 2.

The root mean squared error (RMSE) aims to explore the average difference of actual values with the imputed values as shown in Equation 6. Its value ranges from 0 to $\infty$, where a lower value indicates a better matching.

$$RMSE = \left(\frac{1}{N}\sum_{i=1}^{N} [p_i - \hat{q}_i]^2\right)^{\frac{1}{2}}$$  \hspace{1cm} (6)
The residual error $\mathbf{e}$ in Equation 1 is calculated in this experiment as follows (Muralidhar, Parsa and Sarathy 1999):

$$ e = [\mu_0 + HZ]^T $$

(7)

where $\mu_0$ is a mean vector having zero values, $H$ is a cholesky decomposition of covariance matrix $C$ (Equation 8), and $Z$ is a vector having Gaussian random values. $C$ is calculated as follows (Schneider 2001).

$$ C = \Sigma_{mm} - \Sigma_{ma}\Sigma_{ad}^{-1}\Sigma_{am} $$

(8)

where $\Sigma_{mm}$ is the covariance matrix of missing variables, $\Sigma_{ma}$ is a covariance of missing and available variables, and $\Sigma_{ad}^{-1}$ is the inverse of a covariance of available variables.

Moreover, in order to evaluate the changes in covariance matrices of the consecutive iterations (in EMI and DMI) we calculate the sample covariance matrices from the data sets generated by the iterations. The termination threshold used in EM algorithm is considered to be $10^{-16}$ in the experiments. If the difference between the determinants of two consecutive covariance matrices is below the threshold, and the difference between the average-values of two consecutive mean vectors are below the threshold then the termination condition used in the experiments is considered to be satisfied.

We present performance of DMI and EMI based on $R^2$, $d_2$ and RMSE for both Adult and Credit approval data set in Table 2 and Table 3, respectively.

The average values of the performance indicators on 10 data sets having missing values for each combination of missing pattern, missing ratio and missing model is presented in the tables. For example, there are 10 data sets having missing values with the combination $Com_1$ of “Simple” missing pattern, “1%” missing ratio and “Overall” missing model. The average of $R^2$ for the data sets having $Com_1$ is 0.763 for DMI as reported in Table 2. Bold values in the tables indicate better results between the two techniques. DMI performs significantly better than EMI on both data sets.

We now discuss the experimental results on Adult data set presented in Table 2. For $R^2$ DMI performs better than EMI in 26 out of 32 missing pattern combinations. In 3 occasions it has the same performance as the performance of EMI (Table 2, and Graph 1). In terms of $d_2$ DMI performs better than EMI for all 32 missing pattern combinations (Graph 2). Moreover, for RMSE it performs better in 17 combinations and similar in 8 combinations (Graph 3).
Similarly for Credit Approval data set (Table 3) DMI performs better than EMI for 19 out of 32 combinations in terms of $R^2$ (Graph 4). DMI out performs EMI for all 32 combinations based on $d_z$ (Graph 5). However, for RMSE it performs better in 13 combinations and similar in 5 combinations (Graph 6).

The average performances (for all 32 missing pattern combinations) of DMI and EMI on both data sets are shown in Graph 7 and Graph 10.

It is inspiring to observe that DMI performs significantly better than EMI on Adult data set. Although DMI achieves overall better result than EMI on Credit Approval data set as well, it does not perform as good as its performance on Adult data set. One possible explanation can be the size (30162 records) of Adult data set, which is larger than the size (653 records) of Credit Approval data set. Since in DMI we apply EM based imputation on the records belonging to a leaf we often may end up having insufficient number of records for getting a good result from the EM algorithm. Therefore, it is understandable that DMI’s performance on a larger data set is better than its performance on a smaller data set.

Graph 8 and Graph 11 show a performance comparison of the techniques on both data sets based on four missing patterns namely simple, medium, complex and blended. Both graphs show that the overall performances of both techniques are better for simple pattern type than blended pattern type. We also present a comparison of performances based on missing models in Graph 9 and Graph 12. DMI performs clearly better than EMI in all different analyses, especially for Adult data set.

We now assign a score to an algorithm for each combination of Table 1 and Table 2. If an algorithm performs better than the other then we assign 1, otherwise 0. If the performances of both algorithms are equal then both of them score 0.5. The overall scores of DMI and EMI are presented in Table 4. For both Adult data set and Credit Approval data set in terms of $d_z$ DMI scores 32 while EMI scores zero.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Evaluation Criteria</th>
<th>EMI</th>
<th>DMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adult</td>
<td>$R^2$</td>
<td>4.5</td>
<td>27.5</td>
</tr>
<tr>
<td></td>
<td>$d_z$</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>11</td>
<td>21</td>
</tr>
<tr>
<td>Credit Approval</td>
<td>$R^2$</td>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>$d_z$</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>16.5</td>
<td>15.5</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>45</td>
<td>147</td>
</tr>
</tbody>
</table>

Table 4: Overall performance (score comparison).

5 Conclusion

In this paper we present a novel missing value imputation technique called DMI that makes use of an entropy based decision tree algorithm and expectation maximisation based imputation technique. The main contributions of the paper are as follows. We realise that the EM algorithm produces better imputation result on data sets having higher correlations among the attributes. Besides correlations among the attributes are natural properties of a data set. A data set should not be modified in order to improve the correlations for the sake of achieving better imputation accuracy. However, correlations among the attributes can be higher within a horizontal partition of a data set than within the whole data set. We propose the use of an entropy based decision tree algorithm in order to identify the horizontal segments having higher correlations.

On each horizontal segment DMI algorithm applies an existing imputation technique, which heavily relies on the correlations among the attributes, in order to take advantage of higher correlations within the segments. Thus, DMI is expected to impute numerical missing values with higher accuracy. Moreover, for categorical missing values DMI applies a decision tree based imputation approach within each horizontal segment separately. It applies the decision tree algorithm to build a tree for each attribute having missing value/s. Therefore, it uses an attribute-specific horizontal segments that results in better imputation accuracy. DMI is capable of imputing both numerical and categorical missing values.

DMI also handles the two problematic cases where EMI algorithm may not provide reasonable results. The cases are, all records having the same value for a numerical attribute and all numerical values are missing in a record.

We evaluate DMI on two publicly available natural data sets by comparing its performance with the performance of EMI. We use various missing patterns such as simple, medium, complex and blended each having different missing ratios ranging from 1% to 10%. We also use two missing models/categories namely Uniformly Distributed (all attributes have equal number of missing values) and Overall. Several evaluation criteria such as coefficient of determination ($R^2$), Index of agreement ($d_z$) and root mean squared error (RMSE) are also used. Our initial experimental results indicate that DMI performs significantly better than EMI.

DMI performs clearly better on Adult data set for all evaluation criteria, whereas for Credit Approval data set DMI performs better for $R^2$ and $d_z$ (Table 4). Moreover, we compare DMI with EMI based on missing ratios (Graph 7 and Graph 10), missing patterns (Graph 8 and Graph 11) and missing models (Graph 9 and Graph 12). For Adult data set DMI performs better than EMI for all comparisons. It also performs better than EMI in most cases for Credit Approval data set.

Although DMI performs significantly better than EMI on both data sets, its performance on Adult data set is clearly better than its performance on Credit Approval data set. It is worth mentioning that Adult data set has 32561 records which is approximately 47 times larger than the Credit Approval (690 records) data set. It appears that DMI performs better on larger data sets. Since DMI applies EM based imputation on the records belonging to a leaf separately, for a small data set we often may end up having insufficient number of records for getting a good result from the EM algorithm. However, in our experiments DMI still performs better than EMI even on Credit Approval data set in most of the cases.

Our future work plans include further improvement of the DMI algorithm, and extensive experiments to compare DMI with many other existing techniques such as kernel function, regression analysis and a number of existing hybrid models.
6 References


