
We recommend you cite the published version.
The publisher’s URL is: http://eprints.uwe.ac.uk/34658/

Refereed: Yes

(c) 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Disclaimer

UWE has obtained warranties from all depositors as to their title in the material deposited and as to their right to deposit such material.

UWE makes no representation or warranties of commercial utility, title, or fitness for a particular purpose or any other warranty, express or implied in respect of any material deposited.

UWE makes no representation that the use of the materials will not infringe any patent, copyright, trademark or other property or proprietary rights.

UWE accepts no liability for any infringement of intellectual property rights in any material deposited but will remove such material from public view pending investigation in the event of an allegation of any such infringement.

PLEASE SCROLL DOWN FOR TEXT.
Missing Data Imputation based on the Evidence Chain

Xiaolong Xu
Jiangsu Key Laboratory of Big Data Security & Intelligent Processing
Nanjing University of Posts and Telecommunications
Nanjing, China
xuxl@njupt.edu.cn

Weizhi Chong
State Key Laboratory of Information Security,
Chinese Academy of Sciences
Beijing, China
1215043122@njupt.edu.cn

Shancang Li, Abdullahi Arabo
School of Computer Science
University of the West of England
Bristol, UK
{Shancang.Li, Abdullahi.Arabo}@uwe.ac.uk

Abstract—Missing or incorrect data caused by improper operations can seriously compromise security investigation. Missing data not only damages the integrity of the information but also can lead to the deviation of the p data mining and analysis. Therefore, it is necessary to implement the imputation of missing value in the phase of data preprocessing to reduce the possibility of data missing as a result of human error and operations. The performances of current imputation approaches of missing value cannot satisfy the analysis requirements due to its low accuracy and poor stability, especially the rapid decreasing imputation accuracy with the increasing rate of missing data. In this paper, we propose a novel missing value imputation algorithm based on the evidence chain (MAIEC), which can mine all relevant evidence of missing values in each data tuple, and then combine this relevant evidence to build the evidence chain for further estimation of missing values. To extend MAIEC for large-scale data processing, we apply the Map-Reduce programming model to realize the distribution and parallelization of MAIEC. Experimental results show that our proposed approach has higher imputation accuracy compared with the missing data imputation algorithm based on naive Bayes, the Mode imputation algorithm, and the proposed missing data imputation algorithm based on K-nearest neighbor (KNN). MAIEC has higher imputation accuracy, and its imputation accuracy is also assured with the increasing rate of missing value or the position change of missing value. MAIEC is also proved to be suitable for the distributed computing platform and can achieve an ideal speedup ratio.

Keywords: Missing value imputation; data preprocessing; Chain of evidence; Map-Reduce

I. INTRODUCTION

In existing industrial applications, many data mining analysis and processing tools are used to mine potential useful information for further analysis to improve industry competitive advantage and important findings in scientific research [1, 2]. However, the limitations of data acquisition or improper operation of the data, often lead to data errors, incomplete results, and inconsistencies. Data preprocessing is an indispensable step before data mining. Data preprocessing includes missing data imputation, entity identification, and outlier detection. Missing data padding is an important problem to be solved in data preprocessing. Because of lack of information, the omission of information and man-made operation, some data are missing in the dataset. These incomplete data will affect the quality of data mining and even lead to the establishment of the wrong data mining model, making the data mining results deviate from the actual data.

Pearson R K [3] pointed out that missing data that will lead to three major problems: (1) most of the data processing algorithms at this stage do not have the ability to process data sets with missing data. Faced with these datasets are missing data, these have been widely used algorithms or systems are often powerless; (2) In the data mining process, in order of performing simple operation to save time, the issue of missing records has often been overlooked, which will lead to poor statistical results; (3) mining data contains missing records, some mining methods have a high sensitivity to the data missing rate in the data set. The reduction of the effective record in the data set will result in a rapid decrease or even a deviation of the mining effect.

In industrial applications, more than 80% of the effort is devoted to data preprocessing, so it is necessary to process missing data in order to make full use of the data already collected in machine learning and data mining [references needed].

Little, R. et al. [4], proposed three types of data loss according to the degree of randomness of deletion: missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR). The missing type of a data set affects the choice of imputation algorithm and the final effect of imputation. The mean-imputation method is a common missing value imputation method for MCAR. Since the imputation of NMAR data depends on the prior knowledge of the data set itself. The effect of different missing types on data classification is discussed in [5].
Due to the difficulty of NMAR data imputation, the current missing value imputation algorithm is for MAR data. Relevant missing attribute values are used to estimate the value of the missing data. However, these methods have their own shortcomings. For example, the linear regression algorithm based on statistical probability and the maximum expectation algorithm must have enough knowledge of the data distribution in the data set. However, our understanding of the most data sets is limited. Based on data mining, such as Bayesian network and k-neighborhood algorithm, the Bayesian network should have a certain knowledge of the domain and knowledge of the data. It is necessary to clear the dependence between various attributes, training Bayesian networks directly with the data set is quite complex, and KNN algorithm with a high rate of missing cases, the imputation effect will be greatly reduced.

In this paper, we propose a missing data imputation approach based on a chain of evidence (MIBOCE). MIBOCE obtains all relevant evidence of missing data in each tuple of missing data by mining, which is further combined to form a chain of evidence to estimate the missing attribute value. Finally, the value of the missing data is estimated by the chain of evidence. We do not need to master the distribution of data in the data set, domain knowledge, and do not need to train the data set estimation model for the imputation to save a lot of time. Most existing algorithms are designed to deal with small data sets on a single machine, but now with the development of information technology, the rapid growth of data, large-scale data processing on a single machine is clearly inappropriate. The proposed MIBOCE algorithm is based on the Map-Reduce programming model. In this paper, Map-Reduce programming model implements MIBOCE algorithm, imputation of large-scale data sets on distributed platforms. We use UCI machine learning data to carry out the random exclusion of different proportions of the attribute values to get the experimental data set, and the imputation of the missing data. The results show that MIBOCE has high imputation accuracy and imputation stability when imputation values are missing.

II. RELATED WORKS

Zhu et al. [6,7], divide the processing of missing data sets into three main approaches:(1) Delete the missing records, (2) Directly analyze missing datasets, ignoring missing values, (3) Imputation the missing value.

Deleting missing values is to delete the tuples or records from the missing data, using only the complete data tuple or record, so that the analyzed data set is complete. This approach is simple, especially if the number of missing data is small for the entire dataset, but it is clear that this method will miss some important information because the deleted tuples have missing attributes, leaving only the complete tuple, will lead to the analysis of the dataset smaller. D.B. Rubin [8,9], describes in detail the dangers of deleting tuples of missing data. The second idea is not to delete the missing attribute tuple, nor perform imputation of the missing value, directly in the missing data in the dataset analysis and mining. As described in [10], Bayesian networks are used to process the data. [11, 12] using artificial neural networks to process the data. This idea can save a lot of time and avoid the noise caused by the imputation of the missing data. However, the missing data in the dataset leads to the loss of the useful information, which deviate the mining result. Moreover, [13] concluded that this method is poor in the accuracy of classification when classifying discrete datasets. The work uses the third idea to amputate the missing value.

For the time being, most of the research when considering missing data imputation methods are divided into two categories, based on the probability of statistical analysis and data mining.

Imputation the missing data is to start from the probability of statistical analysis, the probability of missing data sets are analyzed to obtain the missing value of the relevant information, and based on this probability information of imputation the missing value. Linear regression (LR) [14, 15] and Expectation-Maximization (EM) [16] are the most common methods based on probability. For LR and EM parameter methods, if the data distribution of the dataset being processed is not well understood, this will result in an estimate of the deviation. However, in real life, we have a limited understanding of the dataset to be processed. If we compare the data distribution of the dataset, choose the appropriate parameters, and then the missing value of the imputation effect is quite good. But like the EM method even for the dataset of data distribution to understand, the parameter convergence is very slow and time-consuming.

Bayesian classification based on statistics is also a common algorithm for imputation of missing data. Naive Bayes classification results can be compared with the decision tree algorithm and neural network classifier [17]. The naive Bayesian classification algorithm used for imputation of the missing value is based on the Bayesian theorem formula to estimate the value of missing data [18]. The estimated value of each missing data depends on the value of the other attributes in its tuple. The core formula is as follows:

\[ P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)} \]  \hspace{1cm} (1)

In Equation (1) 1, \( C_i \) represents an imputation value of the missing value estimate, \( X \) represents the value of other attributes in the tuple where the missing value is located. Since \( P(X) \) is a constant for all classes, as long as \( P(X | C_i)P(C_i) \) is computed. Naive Bayes has a conditional independence hypothesis. Therefore, the calculation of \( P(X | C_i) \) is given by the following formula:

\[ P(X | C_i) = \prod_{k=1}^{n} P(X_k | C_i) \]  \hspace{1cm} (2)
The algorithm searches the entire dataset for each missing value \( P(C_i | X) \), taking the maximum probability \( C_i \) as imputation value of the missing value. The method is simple and easy to implement, and the imputation effect is also good. But in real life, the relationship between the attributes of datasets is not independent of each other, hence the limitations of using Naive Bayes.

It is the most commonly used method for imputation of missing data in missing dataset by using machine learning or data mining methods. The idea of using data mining for imputation in missing data is to extract useful information from the original dataset for building the prediction model based on this useful information to estimate the value of missing value from the predicted model. A number of algorithms based on data mining have been reported, such as k-nearest neighbor (KNN) [19, 20], Bayesian network [21], rough set method [22], etc. There are various algorithms to form a complex combination of algorithms; in [23] the decision tree algorithm and the maximum expectation algorithm are combined to form a new algorithm. This algorithm makes use of the higher classification accuracy of the decision tree, which makes the filling effect very good, with higher classification accuracy; as well as in the original algorithm on the innovative algorithm. In [24], an incomplete data clustering algorithms, MBOI, was proposed, which defines the set of constraint tolerance sets for the incomplete data set of categorical variables. It can be used to estimate the overall difference of incomplete data objects from the set point of view degree, and further imputation in the missing data based on the result of incomplete data clustering. The algorithm has a good effect on the accuracy of the imputation and the time efficiency of the algorithm.

KNN is based on distance data classification algorithm, which searches the entire data set to find the k data tuples closest to a given data tuple. KNN is able to mark the tuples with missing data as missing tuples, and then the complete data tuple in the data set is used as the training data set, searching for the k data tuples nearest to the missing tuples in the entire training dataset, and finally performing imputation of the missing tuples according to the k data tuples. In practice, KNN has very high classification accuracy, experiments in [20] demonstrated that the K-nearest neighbor algorithm performs well in the absence of DNA microarray data. However, KNN algorithm has its own shortcomings, when the dataset has a large percentage of missing data, KNN filling accuracy will be greatly reduced.

### III. Missing value imputation algorithm

#### A. Related Definitions

Let D denote a dataset with \( m \) rows and \( n \) columns. in another word, \( D \) has \( m \) data tuples, each data tuple has \( n \) attributes, then the data set can be defined as

\[
D = \{A_i | 1 \leq i \leq n\}
\]

in which \( A_i (1 \leq i \leq n) \) represents the \( i \)-th column attribute of the dataset \( D \).

Data tuples in the dataset can be defined as

\[
D_i = \{V_j(A_i) | 1 \leq j \leq m, 1 \leq i \leq n\}
\]

where \( X_{j,i} (1 \leq j \leq m, 1 \leq i \leq n) \) represents the value of the \( j \)-th row \( i \)-th column attribute of the dataset \( D \). Let the \( i \)-th attribute of the \( j \)-th tuple in \( D \) be \( V_j(A_i) \).

**Definition 1.** The missing model is defined as follows

\[
V_j(A_i) = \begin{cases} X_{j,i} & (1 \leq j \leq m, 1 \leq i \leq n) \\ ? & (? \leq j \leq m, 1 \leq i \leq n) \end{cases}
\]  

(5)

in which \( V_j(A_i) = ? \) means that the \( i \)-th attribute value of the \( j \)-th tuple is missing.

When there is \( V_j(A_i) = ? \) in the data tuple, the data tuple is an incomplete data tuple, denoted as:

\[
Z_j = \{V_j(A_i) | \exists V_j(A_i) = ?, 1 \leq j \leq m, 1 \leq i \leq n\}
\]

(6)

When there is no \( V_j(A_i) = ? \) in the array tuple, then the array tuple is believed as complete data tuple, denoted as:

\[
R_j = \{V_j(A_i) | V_j(A_i) \neq ?, 1 \leq j \leq m, 1 \leq i \leq n\}
\]

(7)

The non-missing data in the incomplete data tuple is the associated attribute value of the missing data in the incomplete data tuple.

**Definition 2:** Let the set of combinations of the associated attribute values for the missing data as the chain of evidence to estimate the value of the missing value:

\[
S_j = \{C(y,u) | 1 \leq y \leq n, 1 \leq u \leq y\}
\]

(8)

in which \( C(y,u) \) is the combination of the associated attribute values of the estimated missing values, which means the \( u \) unregistered attribute values are selected from the \( y \) complete attribute values, which is recorded as the evidence of the estimated missing value.

The main goal of the algorithm is to estimate the value of the missing data in the \( j \)-th data tuple by the set \( S_j \).
In each data tuple $M_i (1 \leq i \leq m)$ of dataset $D$, there is a set of attributes, which is assumed to be $A$. The data tuple $M_i (1 \leq i \leq m)$ contains $A$, if and only if $A \subseteq M_i (1 \leq i \leq m)$, the rule is formed as $A \Rightarrow B$ in the tuple $M_i (1 \leq i \leq m)$ of the dataset $D$. Among them, $A \subset D$, $B \subset D$, $A \neq \emptyset$, $B \neq \emptyset$, and $A \cap B \neq \emptyset$, all tuples in dataset $D$ contain set $A$ and set $B$ with the ratio $P(A \cup B)$.

Definition 3: the support count $S$ represents the number of a set in the dataset, and then the support count of the rule $A \Rightarrow B$ in the whole dataset $D$ is defined as:

$$S(A \Rightarrow B) = P(A \cup B)$$

(9)

Definition 4: Dataset $D$ contains both attribute value set $A$ and contains attribute value set $B$ data tuple ratio is $P(B \mid A)$, the credibility of the rule $A \Rightarrow B$ in the whole dataset $D$ is defined as:

$$F(A \Rightarrow B) = P(B \mid A)$$

(10)

Credibility is calculated as:

$$F(A \Rightarrow B) = P(B \mid A) = \frac{S(A \cup B)}{S(A)}$$

(11)

B. Algorithm Flow

MIBOCE is based on the set of associated attribute value combinations of missing data as a chain of evidence to estimate the value of missing data. According to the idea of data mining, there is a certain relationship between data attribute values in large-scale data sets. In the imputation process, the algorithm will first estimate the missing value of the imputation value, the algorithm scans each data tuple in the entire data set, marking tuples with missing values ‘?’ as incomplete data tuples, and combine the different associated attribute values of the missing data in the incomplete data tuple as evidence of the estimated missing value. So that a large number of combination of the relevant attributes for missing data in the incomplete tuple constitute the estimated missing data value of the chain of evidence. The algorithm scans the entire dataset again and counts the combination of attribute values in all data tuples, and uses the relevant theorem in the prerequisite knowledge to calculate the value of the missing data. The core task of the algorithm is to calculate the reliability of each estimated missing value in the chain of evidence. Thus, gives the sum of the confidence of all the evidence for the estimated missing data, The maximum estimate of the sum of the confidence values is selected as the imputation value.

MIBOCEF algorithm is divided into five stages:

Stage 1. The algorithm scans the data set $D$ to uniquely identify $I_k (1 \leq k \leq m)$ for each data tuple $D_j$, then gives the position $M_i (1 \leq i \leq n)$ of missing data for each incomplete data tuple to determine which attribute data in the tuple is missing and output the marked data set. The output data format is $(I_k, M_n, \hat{D}_j)$.

Stage 2. This stage contains four parallel modules.

Module 1. The algorithm first scans the result file created in Stage 1 to compute the set $S_j$ of combinations the complete data $R_j$ in incomplete data tuple $Z_j (1 \leq j \leq m)$, which will serve as the evidence chain for estimating the missing data values. The result will be used as the chain of evidence to estimate the missing data. The output data format is $(I_k, M_n, S_j)$.

Module 2. The module calculates the probability $P(p)$ of the possible values of the missing values $p$ in each missing tuple from the complete data tuple and the output data $(p, P(p))$.

$$P(p) = \frac{K(p)}{m}$$

(12)

The function in the formula $K()$ represents the count, $K(p)$ denotes the possible value of the missing value $p$ the number of occurrences of the same missing location in each data tuple, and $m$ denotes the number of data tuples.

Module 3. The module counts the number $O_j$ of the data set of complete data combination $S_j$ for each data tuple in the entire data set and will be used in the probability query for the missing data value estimation in the following steps. The output data format is $(S_j, O_j)$.

Module 4. The algorithm counts the number of complete sets of data $S_j$ and missing data in the incomplete data tuple $Z_j (1 \leq j \leq m)$ in the same data tuple, that is $T_j$.

$$T_j = K(V_j (A_j) = '?' (1 \leq j \leq m, 1 \leq i \leq n) \cup S_j = \{C(y,u) \mid 1 \leq y \leq n, 1 \leq u \leq y\}$$

(13)

The format of the output is $(S_j, M_n, p, T_j)$.

Stage 3. In this stage, the evidence chain $(I_k, M_n, S_j)$ of estimated missing data created in module 1 in stage 2 will be connected with the possible value $(p, P(p))$ of output missing data in module 2 in stage 2. The probability $P(p)$ that the associated attribute value combination $C(y,u)$ of missing data in each incomplete data tuple $Z_j (1 \leq j \leq m)$ and each possible padding value $p$ occur simultaneously in the entire data set. The output data format is $(S_j, M_n, p, P(p))$.

Stage 4. In the stage to the number $O_j$ of the set $S_j$ of the associated attribute value combinations in the incomplete data $Z_j (1 \leq j \leq m)$ in which the missing data $V_j (A_j) = '?' (1 \leq j \leq m, 1 \leq i \leq n)$ is located in the result file of the Stage 2 module 3 according to $e$ in the Stage 3 result file.
According to the results file of Stage 3 $S_i$ and $p$, and in the results of the Stage 2 module 4 to find the associated attribute value set $S_i$ and missing data possible values $p$ in the entire data set the number of simultaneous appearances $T_j$. According to the above preliminary knowledge, in the incomplete data tuple $Z_j(1 \leq j \leq m)$, we can compute the probability that all possible missing data values are taken from the set $S_i$ of the associated attribute value combinations.

$$F(S_i \Rightarrow p) = P(p | S_i) = \frac{S(p \cup S_i)}{S(S_i)}$$ (14)

We choose the maximum estimate of the confidence $F(S_i \Rightarrow p)$ as the final imputation value.

Stage 5. The algorithm imputation the original missing data set $D$ with the possible values of the missing data estimated in Stage 4.

The timing diagram of the algorithm is shown in Figure 1.
C. Map-Reduce Parallelization of Algorithms

Map-Reduce adopts a divide-and-conquer strategy for large-scale data sets. The core of MapReduce's parallelization is Map and Reduce, the Map-Reduce computing framework divides the data set into many small files of the same size and assigns them to different nodes. Each node performs Map calculation, and the results are sorted and merged. The same Key values are put in the same set Reduce calculation [10]. Figure 2 shows the implementation process of Map-Reduce.

Stage 1. Tag missing data sets
Input: A data file with missing values
Output: Data tuple labels, and data tuples

Map<Object,Text,Text,Text>
Input:key=offset,value=tuple
1. FOR each key,value> DO
2. ADD tupleindex into tuple
3. FOR each attri-v,tuple> DO
   IF tuple contains missingvalue THEN
      Outkey:tupleindex
      Outvalue:tuple

Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO

Stage 2. This Stage is divided into 4 modules, each module can be simultaneous.
Module 1. The missing data-related attribute combination set.
Input: Stage 1 results file.
Output: The data tuple label, the missing value location, missing value a set of attribute value combinations.
Map<Object,Text,Text,Text>
Input:key=offset,value=tuple
1. FOR each key,value> DO
2. IF tuple contains missingvalue THEN
   ADD rest complete attribute into set comple-attri
3. Calculation complete attribute combination combi-attri in comple-attri
4. Outkey:tupleindex+missingindex
   Outvalue:combi-attri

Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO
   Outkey: key
   Outvalue: tuple

The Map function in Stage 1 scans the dataset to add a tag tupleindex for each data tuple. The final output data format of the Reduce function is (tupleindex, tuple).

Stage 2. This Stage is divided into 4 modules, each module can be simultaneous.
Module 1. The missing data-related attribute combination set.
Input: Stage 1 results file.
Output: The data tuple label, the missing value location, missing value a set of attribute value combinations.
Map<Object,Text,Text,Text>
Input:key=offset,value=tuple
1. FOR each <key,value> DO
2. IF tuple contains missingvalue THEN
   ADD rest complete attribute into set comple-attri
3. Calculation complete attribute combination combi-attri in comple-attri
4. Outkey:tupleindex+missingindex
   Outvalue:combi-attri

Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO
   Outkey: key
   Outvalue: comple-attri

In module 1, missing index is the position of the missing data in the incomplete data tuple. The Map function adds the associated attribute value combinations of the missing data in the incomplete data tuple comple-attri to the set combi-attri. Reduce function final output data format is (tupleindex, missingindex, combi-attri).

Module 2. Possible values for missing data
Input: Stage 1 results file
Output: Array tuple labels, missing data possible values, missing data possible probability of the value.
Map<Object,Text,Text,Text>
Input:key=offset,value=tuple
1. FOR each <key,value> DO
2. FOR each <attri-v,tuple> DO
   Outkey: attriindex
   Outvalue: attri-v
Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO
   Add value into list-pro
2. Calculation valuelist-pro length divided by m as pro
3. Outkey: attriindex
   Outvalue: attriindex+ list-pro+ pro
The algorithm in Module 2 scans each data tuple. The Map function records the value of each attribute attri-v and outputs each attribute number attri-index. The Reduce function returns the list of possible values for each attribute list-pro and the probability of each possible value probability.

Module 3: Counts the number of attribute value combinations
Input: Stage 1 results file
Output: The set of attribute value combinations, the number of attribute value combinations
Map<Object,Text,Text,Text>
Input: key=offset,value=tuple
1. FOR each <key, value> DO
   Calculation $C(y,u)$ in tuple as combi-attri
2. Outkey: combi-attri
   Outvalue: 1
Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO
   Calculation number of combi-attri
2. Outkey: combi-attri
   Outvalue: num_c

The Map function in Module 3 calculates the combination of attribute values in each data tuple combi-attri, the Reduce function calculates the number of each attribute value combination in each data tuple in the entire data set num_c.

Module 4: The number of data tuples that have a combination of attribute values and an attribute value are counted throughout the data set.
Input: Stage 1 results file
Output: The combination of attribute values, the position of an attribute value, the value of an attribute, the number of data tuples that have a combination of attribute values and an attribute value.
Map<Object,Text,Text,Text>
Input: key=offset,value=tuple
1. FOR each <key, value> DO
2. FOR each <attri-v, tuple> DO
   Calculation $C(y,u)$ in rest complete attribute as combi-attri
3. Outkey: combi-attri+attriindex+attri-v
   Outvalue: 1
Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO
2. Calculation number of combi-attri+attriindex+attri-v
3. Outkey: combi-attri+attriindex+attri-v
   Outvalue: num_caa

The Map function in Module 4 scans each data tuple, selects the attribute value attri-v in the data tuple, in turn, calculates the attribute value combo-attri combi-attri in the remaining attribute values. In the Reduce function, the number of data tuples num_caa that have a combination of attribute values combi-attri and an attribute value attri-v are counted throughout the data set.

Stage 3: The set of associated attribute value combinations for the missing data is concatenated with the possible values for the missing data.
Input: The resulting file for Module 1 in Stage 2, and the result file for Module 2 in Stage 2
Output: Associated attribute value combinations, missing values in incomplete data tuples, possible values.
Map<Object,Text,Text,Text>
Input: key=offset,value=missingindex+combi-attri
1. FOR each <key, value> DO
   Split the value
2. Outkey: missingindex
   Outvalue: combi-attri
Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO
2. Outkey: offset
   Outvalue: combi-attri+missingindex+pro-v

In stage 3, pro-v is the possible value of the missing value. The first map will split the data of the result file of Module 1, missingindex as key and combi-attri as value. The second Map will be missing the value of the possible value of each line of the data file to be separated missingindex as a key, pro-v as value. In Reduce the value of the same key will be placed in the same valuelist, Reduce will be missing data associated attribute value combinations combi-attri and possible values pro-v to connect, the final output data format (combi-attri, Missingindex, pro-v).

Stage 4: Estimates the possible values of missing values
Input: Module 3 Results File CAacount, Stage 2 Module 4 Results File CA-Aaccount, Stage 3 Results File.
Output: Missing data estimates
Map<Object,Text,Text,Text>
Input: key=offset,value=missingindex+combi-attri+pro-v
1. FOR each <key, value> DO
   Split the value
2. Research account of combi-attri in CAacount recorded as num-combi-attri
3. Research account of combi-attribi+pro-v in CA-Aaccount recorded as num-combi-attribi-a
4. Calculation num-combi-attribi-a/num-combi-attribi as credibility
5. Outkey:tupleindex+missingindex
   Outvalue:credibility+pro-v

Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO
   Sum of credibility
2. IF sum of credibility is maximum THEN
   Outkey:offset
   Outvalue:pro-v

Stage 4 is the core part of the algorithm and will be used to estimate the value of the missing value. The Map function first divides each row of the result file of Stage 3 and finds the number num-combi-attribi of associated attribute value combinations combi-attribi of missing value in the file CA-Aaccount as S(comb i-attribi). Finds the number of data tuples in the file CA-Aaccount that have a combination of attribute values of missing data and possible values combi-attribi+pro-v as S(comb i-attribi∪pro-v). And the credibility of the missing data may be calculated. The Reduce function adds all the evidence of confidence to estimate the missing value and takes the possible value of the maximum sum of the evidence as pro-v as the final filling value.

Stage 5. The value of the missing value estimate of stage 4 is filled into the original missing data set.

Input: Missing data set file. Stage 4 Missing value estimates

Output: Full data set

Map<Object,Text,Text,Text>
Input: key=offset, value=tuple
1. FOR each <key, value> DO
2. Outkey: offset
   Outvalue: value

Map<Object,Text,Text,Text>
Input: key=offset, value=missingindex+pro-v
1. FOR each <key, value> DO
2. Outkey: offset
   Outvalue: missingindex+pro-v

Reduce<Text,Text,Text,Text>
1. FOR each in valuelist DO
   missingindex+pro-v in listA
2. FOR each in listA DO
   pro-v append to value
3. Outkey: key
   Outvalue: com-tuple

In Stage 5, the Map function takes the offset of the original missing dataset file and the offset of the missing value file as the key. The "Missing + Possible Value" of the original missing data set's value and the missing value's estimated file, As the value of the output value. The Reduce function stores "missingindex + pro-v" in listA in each valuelist and populates all the values in ListA into the value of the missing data set, and finally outputs the complete data tuple com-tuple.

IV. EXPERIMENTS

A. Experimental Data

The experimental data in this thesis are from the real data set of UCI (www.archive.ics.uci.edu/ml/datasets.html). The UCI database is a database for machine learning developed by the University of California Irvine. The UCI dataset is a commonly used standard test data set. The Adult dataset is from the American Census Database, completed by Barry Becke et al. The data are 32561 data, a total of 15 different attributes of people's age, work type, weight, education, education duration, marital status, occupation, interpersonal, race, gender, capital status, and so on information. The experiment will be in accordance with a certain percentage of data randomly removed from the data. About final criteria for filling the missing values, there is no unified answer in the field of data mining because the data object and purpose are usually different. This work mainly analyzes the algorithm from two aspects: accuracy of imputation and speedup.

B. Experimental Results

In this experimental, we select five discrete attributes, which are sex, race, education, occupation, and workclass. The values of these attributes are randomly removed in different proportions, and data sets with different missing rates are obtained. The missing rates were 10%, 20%, and 30%, respectively. For each deletion rate, 5 groups of experiments were performed. The experimental results are shown in Figure 3.
From the above results, we can see that MIBOCE uses the other attribute values in the tuples where the missing values are as evidence to estimate the value of missing data. As more than one calculated evidence, so that the algorithm can guarantee the stability of the imputation accuracy. Will not change the location of the missing data in the dataset and the larger fluctuations.

The 5 groups of each missing rate were averaged and the imputation accuracy of MIBOCE under different missing rate was obtained. As shown in Figure 4.

![Figure 4](image)

**Figure 4.** Mean imputation accuracy of MIBOCE algorithm with different missing rates

It can be seen from Figure 4 that the MIBOCE algorithm has stable imputation accuracy. The accuracy of imputation does not fluctuate significantly with the missing rate in the data set, which means that the imputation accuracy of the algorithm will not decrease with the increase of data set missing rate.

We further choose a typical imputation algorithm based on Naive Bayes, imputation algorithm based on Mode, imputation algorithm based on k-Nearest Neighbor comparative experiment with MIBOCE. Imputation algorithm based on Naive Bayes is based on probabilistic statistics. Imputation algorithm based on Mode is to use the most frequently occurring value as the imputation value. Imputation algorithm based on k-Nearest Neighbor is to cluster the entire data set, and then use the representative of each cluster to imputation.

Thesis [19] also uses the data set Adult and gives the imputation algorithm based on Mode, imputation algorithm based on k-Nearest Neighbor experimental results. The experimental results are shown in Figure 5.

![Figure 5](image)

**Figure 5.** The imputation accuracy of missing data under different algorithms

It can be seen from the experimental results that the imputation accuracy of MIBOCE is better than that of other algorithms. Imputation algorithm based on Naive Bayes in the low missing rate conditions can play a very good effect, but with the increase of the missing rate, the imputation accuracy will be decreased significantly. The imputation algorithm based on Mode is better in the anti-miss rate, that is, with the increase of the missing rate, the imputation accuracy of the algorithm will not be greatly reduced. The Imputation algorithm based on k-Nearest Neighbor is poor in anti-miss rate, and the accuracy of imputation will decrease sharply with the increase of missing rate.

This experiment is to test the distributed parallelism of the algorithm, the algorithm is based on Hadoop platform to implement its Map-Reduce. We test the speedup of the algorithm on the generated data.

The data sets with missing rates of 10%, 20%, and 30% were used in this experiment. The experiment results show that the algorithms with different missing data under a different number of nodes are implemented. Table 1 shows the time of algorithm execution for the data sets with different missing rates at different nodes. Figure 6 shows the speedup of the algorithm for different clustering nodes with different missing data rates. It can be seen that the algorithm is linearly accelerated and has more concurrency advantages under multiple nodes, And the speedup of the algorithm decreases with the increase of the missing rate. This is in line with expectations because the main calculation of the algorithm works on the complete data when missing too much data will lead to the availability of complete data reduction, the implementation of the algorithm will be affected to some extent.

<table>
<thead>
<tr>
<th>Missing rate</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. Algorithm execution time at different missing rate and different cluster numbers
V. CONCLUSION

The problem of data loss has caused many difficulties in data mining, especially in the large-scale data set, the data loss becomes an important factor affecting the quality of data. The processing of missing data has become a very important step in the process of data preprocessing. In this work, we use the evidence chain to predict the value of missing data and estimate the missing value by using the set of combinations of missing attribute values. In order to make the algorithm applicable to large data processing, this algorithm combined with a Map-Reduce framework to achieve the parallel processing of imputation missing data. Experimental results show that the algorithm has high imputation accuracy of missing data, and the imputation accuracy is still better with the increase in missing rate. It does not fluctuate greatly with the position of missing data, and can be applied to the distributed computing environment, and can achieve the ideal algorithm to speed up and so on.

But this algorithm is mainly for the discrete missing data, has some limitations. In addition, since the main computation time of the algorithm lies in the choice of data set attribute combination, when the dataset attribute dimension increases, the time consumption of the algorithm will increase rapidly. In the future research, we will focus on the imputation algorithm of missing values in the case of continuous data loss and the data reduction method of large data.

REFERENCES


![Figure5. Speedup of the algorithm under different missing rate data set and different node numbers](image)

