Final Report – Data Mining

Co-appearance, Correlation and Similarity Analysis - FIMUS

Professor
Prof. Roozbeh Razvi Far

Students
Feng Zhu - 104217106
Pranav Patel -104417311
Yuwie Liu – 104307195

Submission Date
8/19/2015

TABLE OF CONTENTS

LIST OF TABLES

ABSTRACT

With the development of Data Mining, Data Mining becomes widely used in artificial intelligence, machine learning, statistics, and database systems etc. In this project, we focus on using new technology, which is named FIMUS to impute missing data imputation. FIMUS is a framework for missing data imputing using co-appearance, correlation and similarity analysis. Based on this technology, we prefer to use JAVA computer language to write a code. When we complete the final project, we will demonstrate demos, final reports and make presentation. In conclusion, this report describes a new typical method of imputing missing data.

INTRODUCTION

It is common that original data sets have missing values. Therefore, one of most important pre-processing tasks is to impute the missing values. There are some frequency-used approaches, which are showed below for imputing missing value. Using the mean of all useful values of the attribute to have a missing value, as an imputed value is one early approach. However, mean imputation can cause some wrong outputs. Another inchoate approach is to delete the records, which have missing values, but it may also remove some useful data for a statistical analysis and data mining.

In this project, we design a new imputation technique, which is called A Framework for Imputing Missing Values Using Co-Appearance Correlation and Similarity Analysis (FIMUS). Making a reasonable guess, which is based on the co-appearances of the values correlations between attributes and similarity of values belonging to an attribute, is the main idea of the approach. Not only can FIMUS impute numerical missing values, but it also can impute categorical missing values.

The paper showed as follows: Section 2 presents some related works on missing value imputation. Section 3 shows the FUMIS and section 4 presents empirical evaluation.

FIMUS TECHNOLOGY

In our opinion, a data set D usually has some natural patterns in it. We plan to estimate the possibility by studying the co-appearance matrix C which is obtained from the data set D. In order to estimating the co-appearances of the values with other values belonging to other attributes, FIMUS generalizes the values of a numerical attribute into some categorizes. Categorizing the same numerical values into the same category instead of categorizing them into different categories is a advisable method. FIMUS studies multiple information from the co-appearance matrix, correlation

matrix, frequency and similarity of values so that it can impute a missing value.

DESIGN METHEDOLOGY

Number of missing value imputation techniques are recently proposed like kNNI and Local Weighted Linear Approximation Imputation (LWLA). But, these techniques have limitation like they are expensive for large datasets.

FIMUS generalizes the values of a numerical attribute into several categories in order to be able to compute the co-appearances of the values with other values belonging to other attributes. For categorizing the values of a numerical attribute, a number of methods have been proposed like PD (Proportional Discretizer) and FFD (Fixed Frequency Discretizer). Experiment results shows that FIMUS using its own categorization (i.e. FIMUS Categorization) produces a better imputation accuracy than FIMUS using PD and FFD for the categorization purpose, for the both data sets and the both evaluation criteria. Moreover, the categorization approach used in FIMUS is very simple. Categorization methods lead to loss of information. So, instead of imputing a missing value, using categorization approach, FIMUS imputes the missing value by taking into account other available values and their similar values.

Dataset that used in this report consists Numerical and Categorical attributes. In FIMUS method, missing data is imputed by calculating vote for each possible candidates based on Co-relation, Co-appearance and Similarity value of each and every other attributes of element. Therefore, FIMUS method is restricted to impute only one missing data into element. Dataset that used in this method is shown Table-1.

FIMUS systematically studies various information from the co-appearance matrix, correlation matrix, frequency and similarity of values in order to impute a missing value. In order to resolve missing data imputation by using the FIMUS technology, the FIMUS means several steps and algorithms as follows.

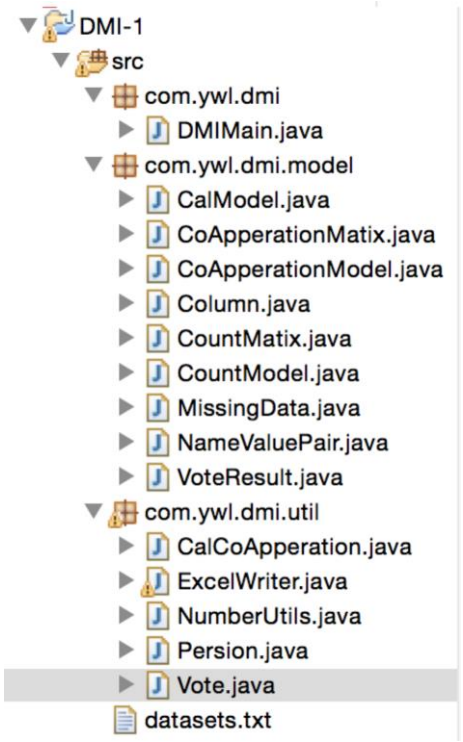
Age	Education	Salary	Position
27	MS	85	L
45	-	145	P
42	PhD	145	P
25	MS	85	L
50	PhD	146	P
28	MS	85	L
38	PhD	140	P
43	PhD	148	-
44	PhD	146	P
-	MS	86	L
42	PhD	142	P

26	MS	84	L
42	PhD		P
25	MS	86	L
43	PhD	143	P

Table 1 Target Dataset

JAVA DESIGN

Java can run on different systems like Windows and OS. Java is an explanatory system language, which can decrease the percentage of error. The construction as below:



Package com.ywl.dmi:

1. DMIMain - Main function

Package com.ywl.dmi.model

1. CalModel.java – Classify the range.

   From this class, we defined some information about the maximum value, the minimum value and the value of range.

2. CoAppearationMatix

   From this class, we defined some information, including value about table 3 in FIMUS paper.

3. CoAppearationModel

   From this class, we defined some information, including attribute name, data and account about table 3 in FIMUS paper.

4. Column

   Generalize each attribute, such as name, index, Integer. Like columns [0] = new Column(0, "Age", true); “0” means the number in

line. “Age” means the attribute, “true” means that the attribute needs to discretization, focusing on figure.

5. CountMatix.java – Defining the number type in matrix.
6. CountModel.java – The form of count number in matrix.
7. MissingData.java – Defining the form of missing data.
8. NameValuePair.java – Defining the form of name.
9. VoteResult.java – Defining the form of vote result.

Package com.ywl.dmi.util:

1. CalCoAppearation.java
   Focusing on the step 3: classify and satisfices the datasets.
2. Get the corresponding value for each column.
3. Classification for the value of each attribute.
4. Print the classify map which is the result in table 3 in FIMUS paper.
5. Persion.java – Defining the Pearson correlation
6. Vote – Defining the CSR for voting

STEP-1

```
Algorithm1: FIMUS
Input: Data set D⁰ having N records and |A| attributes
Output: Imputed data set D⁰ having N records and |A| attributes
Step1:
   1. B←CalculateMissingMatrix(D⁰)
   /* Initialize Missing matrix B where each element bᵢⱼ is
   calculated using Equation (5) for I = 1. N, and j = 1. |A|
   */
   2. T = 0 and rmse = 1;
   /* T is used as iteration counter and rmse is used as
   termination condition*/
   3. λ= user defined value between 0 and 1; default λ= 0.2;
end
```

We put data set D^0 to the missing matrix B, which is initialized. There is calculated using formula:

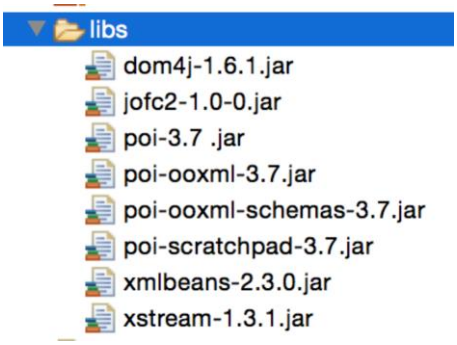
$$b_{ij} = \begin{cases} 1, if\ \ r_{ij} \in D^0\ is\ missing; \\ 0, if\ \ r_{ij} \in D^0\ \ is\ available; \end{cases} \forall i,j$$

From the formula, we can know that each element $b_{ij} \in B$ ($(1 \le i \le N)$ and $(1 \le i \le |A|)$), including either 0 or 1. When b_{ij} equal to 1, the value is missing and when b_{ij} equal to 0, the value is available.

Development procedure：

We input some .jar to realize reading excel files. When we read the excel, the data

in the excel is defined as object. If the data of column is integer, it should be saved as INT to a list; otherwise, if the data of column is missing, it should be signed and saved to the list. These jar are showed as below:



After getting the finished column, we should analyze the column. If the column is available, the column is signed as 0; otherwise, if the column is missing, the column is signed as 1.

When we define the attributes, we need to analyze the data of excel to certain it is hashed or not. If it is needed, it should be true; otherwise, it is false.

For target dataset, generated missing matrix B is shown in Table-2.

Age	Education	Salary	Position
0	0	0	0
0	1	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	1
0	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0
0	0	1	0
0	0	0	0
0	0	0	0

Table 2 Missing Matrix B

STEP-2

$D^P \leftarrow D^0$;

$D_G \leftarrow$ Generalize(D^0);

FIMUS method requires conversation of numerical attributes into categorical

attributes. Main advantage of conversation has reducing number of possible candidates.

All of the numerical attributes of D^0 should be generalized. In this step, we will make a copy of the data set D^0 into D^p. Then, by generalizing the numerical attributes A_j of D^0 into a group of $\sqrt{(|A_j|)}$, in which $|A_j|$ is the domain size of A_j, we create the generalized the data set D_j. Generalize datasets Dj for target dataset is shown in Table 3.

Development procedure：

First, we should analyze the attributes, which need to be hashed, or not. If it is needed, we should get the max and the min based on the formula $domainsize = \sqrt{upperlimit - lowerlimit + 1}$ to get the range. Then, according to the formula $ranges\ from\ a\ to\ a + domainsize - 1$, we can get the region. At last, we can print the results.

```
step 2
max=148, min=84, range=8
max=50, min=25, range=5
25-29   MS      84-91   L
45-49   ?       140-147 P
40-44   PhD     140-147 P
25-29   MS      84-91   L
50-54   PhD     140-147 P
25-29   MS      84-91   L
35-39   PhD     140-147 P
40-44   PhD     148-155 ?
40-44   PhD     140-147 P
?       MS      84-91   L
40-44   PhD     140-147 P
25-29   MS      84-91   L
40-44   PhD     ?       P
25-29   MS      84-91   L
40-44   PhD     140-147 P
```

Table 3 Generalize Dataset

STEP-3

```
C ← C0AppearanceMatrix(DG); /*Generate a Co-Appearance Matrix
CQ*Q.
 S ← null;
    for i= 1 to |A| do
        Si ← SimilarityMatrix(DG, i);
```

```
        Si ← Normalize(Si);
        S ← S ∪ Si
    end
K ← CorrelationMatrix(DG); /*Find correlation matrix K|A|*|A|*/
end
```

Obtain co-appearance matrix(C) regulated semblable matrix (S_j) for attribute A_j; $\forall A_j \in A$, and relevance matrix(K). In this step, the domain values of $A_j \in A$ is based on the co-appearance matrix C, which is generated from D_G. Co-apperance matrix for D_G is shown in Table-4. Besides, V_j like Cxm∈C, is the number of appearances of $X \in A_j$ and m∈Ap in D_G.

cal coapperation matix

	Age_25-29	Age_45-49	Age_40-44	Age_50-54	Age_35-39	Edu_MS	Edu_PhD	Salary_84-91	Salary_140-147	Salary_148-155	Pos_L	Pos_P
Age_25-29	--	--	--	--	--	5	0	5	0	0	5	0
Age_45-49	--	--	--	--	--	0	0	0	1	0	0	1
Age_40-44	--	--	--	--	--	0	6	0	4	1	0	5
Age_50-54	--	--	--	--	--	0	1	0	1	0	0	1
Age_35-39	--	--	--	--	--	0	1	0	1	0	0	1
Edu_MS	5	0	0	0	0	--	--	6	0	0	6	0
Edu_PhD	0	0	6	1	1	--	--	0	6	1	0	7
Salary_84-91	5	0	0	0	0	6	0	--	--	--	6	0
Salary_140-147	0	1	4	1	1	0	6	--	--	--	0	7
Salary_148-155	0	0	1	0	0	0	1	--	--	--	0	0
Pos_L	5	0	0	0	0	6	0	6	0	0	--	--
Pos_P	0	1	5	1	1	0	7	0	7	0	--	--

Table 4 Co-appearance matrix

```
Edu     PhD     0.023   0.052000000000000005
norMap{86_148=0.095, 140_142=0.089, 140_86=0.106, 140_143=0.089, 86_145=0.11, 86_146=0.11, 140_140=0.211, 85_143=0.09
out key : Salary_86
cat = 26,,, okey = Salary_145
Salary  145     0.0     0.0
cat = 26,,, okey = Salary_146
Salary  146     0.0     0.0
cat = 26,,, okey = Salary_148
Salary  148     0.0     0.0
cat = 26,,, okey = Salary_84
Salary  84      0.0     0.0
cat = 26,,, okey = Salary_85
Salary  85      0.0     0.0
cat = 26,,, okey = Salary_86
Salary  86      0.0     0.0
cat = 26,,, okey = Salary_140
Salary  140     0.0     0.0
cat = 26,,, okey = Salary_142
Salary  142     0.0     0.0
cat = 26,,, okey = Salary_143
Salary  143     0.0     0.0
norMap{P_P=0.5, L_L=0.5, P_L=0.5, L_P=0.5}
out key : Pos_L
cat = 26,,, okey = Pos_P
Pos     P       0.023   0.023
cat = 26,,, okey = Pos_L
Pos     L       0.029   0.052000000000000005
Age     26      0.05873873804263494
norMap{MS_PhD=0.5, PhD_MS=0.5, MS_MS=0.5, PhD_PhD=0.5}
out key : Edu_MS
```

Table 5 Similarity Analysis

In order to obtain A_j: $V_j \in A$, we generate a similarity matrix S_j in basis of size $|A_j| \times |A_j|$. By using similarity measures for categorical attributes Occurrence Frequency method, we calculated this similarity between the two values that belongs to an attribute A_j. It is also need to clarify that the similarity can vary from 0 to 1, in which 0 means no similarity while 1 indicates the maximum of it. Similarity Matrix

for each attribute is shown in Table 5. Table 6 shows co-relation matrix calculated using Pearson coefficient contingency method.

(a) Correlation matrix K				
Age	–	0.707	0.721	0.707
Edu.	0.707	–	0.707	0.707
Salary	0.721	0.707	–	0.707
Pos.	0.707	0.707	0.707	–

Table 6 Co-relation matrix

Development procedure：

First, we need to classify and count the hashed attributes in table 2. The same attributes should be one classification, and counting the relation between different attributes, and showing in number. Finally, for all of the attributes, printing the final results.

Getting similarity matrix and standard similarity matrix:

1. Showing the classification of all attributes, and putting the same attribute into one classification. Then, printing the table. If the classifications are the same, the output is 1.
2. Calculating the number of columns, for instance, the example in the reference, there are 15 set of values from R1 to R15; calculating the number of classification of attributes such as the example in the reference, from 25 to 29, it is 5; from 35 to 39, it is 1, and from 40 to 44, it is 6.
3. Using formula which is $S = \begin{cases} 1 & if\ X_k = Y_k \\ \frac{1}{1+\log\frac{N}{f_k(X_k)}\times\log\frac{N}{f_k(Y_k)}} & Otherwise \end{cases}$ to calculate.
4. After getting the similarity matrix, calculating the total of the same classification of the same attribute separately.
5. Using each term of similarity matrix divide the total which is get in the step 4, then we can get the standard similarity matrix.

STEP-4

```
for i= 1 to N do
    for j= 1 to |A| do
        if b_ij= 1 then
            T_i ← ReturnRecord(D_G,i): /*returns the ith record of
        D_G*/
            P_i ← ReturnRecord(D^0,i);
            /*impute categorical missing values*/
            x= CSR(r_i, j, A, C, S, K, λ, D_G);
```

```
            /*impute numerical missing values*/
            If Aj is numercal then
                DN ← SubDataSet(D0, DG, x, j);
                C′ ← CoAppearanceMatrix(DN);
                S′ ← null;
                for g= 1 to |A| do
                    S′g ← SimilarityMatrix(DN, g);
                    S′g ← Normalize(S′g);
                    S′ ← S′ ∪ S′g;
                end
                K′ ← CorrelationMatrix(DN);
                x= CSR(pi, j, A, C′, S′, K′, λ, DN);
            end
            Update D0ij = x; /*D0ij is the same as pij*/
            end
        end
    end
end
```

Type in missing values. In the step 4, there would exist a missing value in the record considering the collection of a record $R_i \in D_G$ and $P_i \in D^0$ by FIMUS. Therefore, the record and other essential parameters will transform into the CSR procedure for imputation. The imputed value obtained from CSR is called X. To be more specific, when the missing value $D_{ij} \in D^0$(i.e.….) pertains to a categorical attribute A_j, we impute D_{ij} with X.

For example, let us assume that for a record Ri there is a missing value for the attribute A_j (A_j ={x; y; z}), and an available value l for another attribute A_p (A_p = {l;m; n}) i.e. Rij is missing and Rip = l. FIMUS uses the Co-appearance of x and l; x and m, and x and n in order to estimate the possibility of x being the correct imputation. The influence of m and n are weighted according to their similarity with l.

$V^{N,p}{}_x$ is the vote in favor of x based on A_p considering only the available value l. We calculate $^{N,p}{}_x$ as follows.

$$V_X^{N,p} = \frac{C_{xl}}{f_l}$$

$V^{S,p}{}_x$ is the vote in favor of x based on A_p considering the available value along with its similar values. That is, $V^{S,p}{}_x$ is calculated considering l; m and n as follows.

$$V_X^{S,p} = \sum_{\forall a \in Ap} \frac{C_{xa}}{f_a} \times S_{la}^p$$

Calculate the weighted vote $V^p{}_x$ in favor of x based on attribute A_p as follow. Where $\lambda = 0.2$

$$V_x^p = \left\{ V_x^{N,p} \times \lambda + V_x^{S,p} \times (1-\lambda) \right\} \times k_{jp}$$

Development procedure：

1. **Pearson's correlation coefficient:**

We need to call the table 3 and get the relation of attributes in table 3. Then, using the data of table 3 to get the object and the value of expectation, which is based on the formula $E_{ij} = \frac{O_i O_j}{n}$, O_i: total observed frequency in the i-th row; O_j: total observed frequency in the j-th column; n: sample size. After getting the value of expectation, according to the formula $T = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$, we need to get each value of T separately and add them together. Finally, to get the Pearson's correlation coefficient which is based on formula $P = \sqrt{\frac{T}{N+T}}$.

2. **Realizing CSR algorithm:**

```
Input: Record r_i, Attribute index j, Attribute list A,
Co-appearance matrix C, Set of Similarity matrices S,
Correlation matrix K, threshold λ and data set D_G
Output: Imputed value r_ij
Step 1:
foreach category x ∈ A_j do
    V^T_x = 0; /*V^T_x is the total voting in favour of x*/

      /*Loop over all attributes in A excluding the jth
   attribute*/

   foreach attribute A_p ∈ A\A_j do
       V^N_x = 0, V^S_x = 0, and l = r_ip;

      /*notations are introduced in Section 1*/

       if λ > 0 then
          V^N_x = C_xl/f_l;
       end
       if λ < 1 then
          foreach category a ∈ A_p do
             H= C_xa/ f_a;
             V^S_x= V^S_x + H * S^p_la; /*see Section 3.1*/
          end
       end
       V^p_x = { V^N_x * λ + V^S_x* (1-λ)}* k_jp; /*k_jp ∈ K is the
correlation between the jth and pth
attribute*/
```

```
                V^T_x= V^T_x + V^P_x0
              end
          end
          end
```

```
Step 2:
Set r_ij ← Value(max(V^T_x; Vx ∈ A_j)); /*finds the attribute value
x 20. for which V^T_x is the Max*/
end
```

```
Step 3:
   Return imputed value r_ij;
   end
```

For each of attributes and the record of missing values. Calling the data of row which has missing values, and analyzing the attributes without the missing values orderly, for example, from the reference, focus on R2, the attribute of missing value is Education, so we need to analyze other three attribute-- Age, Salary and Position orderly.

Ergodic the classification of attribute of missing values, showing all of the possibility-- x, y and z of attribute of missing values separately.

In spite of x, y and z, finding the relation of other attributes and using formula to calculate.

Ergodic all of attributes orderly, using formula to vote.

STEP-5

```
T= T + 1;
    if T >= 2 then
       Calculate rmse from D^P and D^0 using Equation (7);
    end
    if rmse > 0 then
       Goto Step 2;
    end
 end
```

Doing the imputation process (Step2-4) again, and then, there is a change between two consecutive iterations.

STEP-6

```
Return completed data set D^0;
end
```

In the last step, after all the process (step2-5), we can output a completed data set D^0 and there are no missing values in this data set D^0. According to the requirement, we also manually get some diagrams to present and compare our results with the original datasets. Final output dataset which has imputed missing data by FIMUS technology is shown in Table 7.

Age	Education	Salary	Position
27	MS	85	L
45	PhD	145	P
42	PhD	145	P
25	MS	85	L
50	PhD	146	P
28	MS	85	L
38	PhD	140	P
43	PhD	148	P
44	PhD	146	P
25	MS	86	L
42	PhD	142	P
26	MS	84	L
42	PhD	145	P
25	MS	86	L
43	PhD	143	P

Table 7 Final output dataset

RESULT

To measure accuracy of FIMUS missing data imputation method, AE measurement technique is used. AE function is used to find accuracy for categorical dataset. In FIMUS technique, all numerical datasets are categorized. Therefore, AE function is used here.

$$AE = \frac{1}{n}\sum_{i=1}^{n} I(\hat{x}_i = x_i)$$

Where I(.) stands for function that returns 1 if the estimated value $\hat{x}i$ and real value xi are the same (i.e., if the condition is true), but 0 otherwise. AE table is shown in Table 8.

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	0	1
1	1	1	1
1	1	1	1

Table 8 AE table

Here, result value of AE is 0.983.

CONCLUSION

In this final project, FIMUS technology was implemented in JAVA programming language. FIMUS technology imputes missing data using co-occurrence, similarity analysis and co-appearance value. FIMUS technology can implement both type of attribute data, numerical and categorical value. Results shows that FIMUS technology is more accurate method other techniques for missing data imputation. We use Java as our program language. When using Java to realize the function algorithm, we found that Java is a good program language, but in data mining, it has some disadvantages so that we met some problems, for example, it will be more convenience and faster when getting the similarity matrix and standard similarity matrix. In conclusion, we would like to thanks Prof. Roozbeh Razvi Far to give a great opportunity to perform this project that help us to understand concepts of Data Mining and its applications.

REFERENCES

[1] Jiawei Han, Micheline Kamber, Jian Pei, “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2011.
[2] Md. Geaur Rahman, Md Zahidul Islam, “FIMUS: A framework for imputing missing values using co-appearance, correlation and similarity analysis,” Knowledge-Based Systems, vol. 56, pp. 311-327, 2014.
[3] Shyam Boriah Varun Chandola and Vipin Kumar, “Similarity Measures for Categorical Data: A Comparative Evaluation”, Department of Computer Science and Engineering, University of Minnesota.
[4] E.J. Krieg, Statistics and Data Analysis for Social Science, Pearson Education, 2012.