Reconstruction of a Complete Dataset from an Incomplete Dataset by ARA (Attribute Relation Analysis): Some Results

Sameer S. Prabhune ssprabhune@ssgmce.ac.in

Assistant Professor & Head,
Department of Information Technology
S.S.G.M. College of Engineering,
Shegaon-444203, Maharashtra, India

Dr. S. R. Sathe srsathe@cse.vnit.ac.in

Professor,Department of E & CS,
V.N.I.T., Nagpur, Maharashtra, India

Abstract

Preprocessing is crucial steps used for variety of data warehousing and mining Real world data is noisy and can often suffer from corruptions or incomplete values that may impact the models created from the data. Accuracy of any mining algorithm greatly depends on the input data sets. Incomplete data sets have become almost ubiquitous in a wide variety of application domains. The incompleteness in the data sets may arise from a number of factors: in some cases it may simply be a reflection of certain measurements not being available at the time; in others the information may be lost due to partial system failure; or it may simply be a result of users being unwilling to specify attributes due to privacy concerns. When a significant fraction of the entries are missing in all of the attributes, it becomes very difficult to perform any kind of interpretation on the original data. And also it overall deteriorates the accuracy of any classifiers, algorithm for further analysis. For such cases, we introduce the novel idea of attribute weightage, in which we give weight to every attribute for prediction of the complete data set from incomplete data sets, on which the data mining algorithms can be directly applied. This simple but robust mechanism of weighted average gives very precise results, when tested on variety of real world datasets. This paper describes a theory and implementation of a new filter ARA (Attribute Relation Analysis) to the WEKA workbench, for finding the complete dataset from an incomplete dataset.

Keywords: Data Mining, Data Preprocessing, Missing Data

1. INTRODUCTION

Many data analysis applications such as data mining, web mining, and information retrieval system require various forms of data preparation. Mostly all this worked on the assumption that the data they worked is complete in nature, but that is not true! In data preparation, one takes the data in its raw form, removes as much as noise, redundancy and incompleteness as possible and brings out that core for further processing. Common solutions to missing data problem include the use of default [16],imputation, statistical or regression based procedures [11,15]. We note that, the missing data mechanism would rely on the fact that the attributes in a data set are not independent from one another, but that there is some predictive value from one attribute to another [1]. Therefore we used the well-known principle namely, weightage on attribute instance [11], for predicting the missing values. This paper gives the theory and implementation details of addition of an ARA filter in the WEKA workbench for estimating the missing values.

1.1 Contribution of this paper

This paper gives the theory and implementation details of ARA filter addition to the WEKA workbench. Also it gives the precise results on real datasets.

2. PRELIMENARY TOOLS KNOWLEDGE

To complete our main objective, i.e. to develop the ARA filter for the WEKA workbench we have used the following technologies. These are as follows:

2.1 WEKA 3-5-4

Weka is an excellent workbench [4] for learning about machine learning techniques. We used this tool and the package because it was completely written in java and its package gave us the ability to use **ARFF** datasets in our filter. The weka package contains many useful classes, which were required to code our filter. Some of the classes from weka package are as follows [4].

- weka.core
- weka.core.instances
- weka.filters
- weka.core.matrix.package
- weka.filters.unsupervised.attribute;
- weka.core.matrix.Matrix;
- weka.core.matrix.Eigenvalue Decomposition; etc.

We have also studied the working of a simple filter by referring to the filters available in java [9,10].

2.2 JAVA

We used java as our coding language because of two reasons:

1. As the weka workbench is completely written in java and supports the java packages, it is useful to use java as the coding language.
2. The second reason was that we could use some classes from java package and some from weka package to create the filter.

3 PSEUDO CODE

This pseudo code is designed to give the user an understanding of the ARA algorithm. ARA is a simple yet robust technique of weighted average of attribute values. [11,13].

Attribute Relation Analysis Pseudo code

Input: Dataset D with Missing attributes.

Output: Completed dataset D´ with estimated values.

ARA (I_k , J, AV^k, I_t , MA_k , PA_k , AC^k)

Where

I_K - Instant in AV^K

J - Element in AV^K

AV^K - Given attributes with Missing values

MA_K - Missing attributes

PA_k - Predictable attribute

I_t - Iteration

Step 1

Get input data:-

a. Take ARFF file as input.

b. It will be taken as one-dimensional array.

Step 2

Repeat Until (iteration > CK_j)

a. Initially all attribute and instance in iteration is null.

b. Array of instance is initially null.

c. Iteration start from first instance.

d. Initially missing values are null.

e.

Step 3

(Ai_i…….. Ai_j (MA_K) ;

Ai_i…….. Ai_j ; { Ai_i…….. Ai_j} $\neq \Omega$ [ ] ;

a. Check iteration until missing instances

b. If any missing instances is getting stop the procedure.

Step 4

$? \quad \leftarrow \quad AV_i^k \ldots\ldots ? \leftarrow \quad AV_j^k$

a. In iteration number of instances from {Ai……..Aj} so check entire instances in iteration.

b. Check any missing instance in iteration.

c. In given iteration if no missing value, so stop the procedure and print given iteration in array.

Step 5

If { correct classify (I_K.T)} ;

a. Replace the missing value or missing instance by X.
b. And existing instances by Y.

Step 6

(V[chg Num] = $AV_i^k + ... + AV_j^k$

Ω [chg Num] $\leftarrow$ { Ai_i --- Ai_J }

a. Check given instances is correctly classified or not.
b. If not correctly classify.
c. Restore instances using given dataset.

Step 7

Ai_K …. Ai_j; chg Num ++

a. Store the iteration and its changing attribute in an array.

Step 8

Iteration ++

a. Restore old value.

Step 9

if { chgnum ! = 0 }

a. If changing value contains 0, it will be replace by 0.
b. Otherwise replace it by new value.

Step 10

$$Av_i^k = \frac{\sum_{i=0}^{n}[Ai_1 \times w(Ai_n)] + [Ai_2 \times w(Ai_{n+1})] + ... + [Ai_n \times w(Ai_{n+1})]}{\sum_{i=0}^{n} w(Ai_1 + ... + Ai_n)}$$

a. Take upper sum of instances from missing instances and multiply with its weight.
b. Nearest instances get higher weight and weight will be decrease by 1 for each instance.
c. Sum of upper instances is divided by sum of weight.

Step 11

$$Av_j^k = \frac{\sum_{j=0}^{n}[Aj_1 \times w(Aj_n)] + [Aj_2 \times w(Aj_{n+1})] + ... + [Aj_n \times w(Aj_{n+1})]}{\sum_{i=0}^{n} w(Aj_1 + ... + Aj_n)}$$

a. Take lower sum of instances from missing instances and multiply with its weight.

b. Nearest instances get higher weight as compared to other and weight will be decrease by 1 for each instance.

c. Sum of lower instances is divided by sum of weight

Step12

$$\mathrm{MV}^k = \frac{AC_i^K + AC_J^K}{2}$$

Finally take average of upper instances and lower instances.

Result is replaced by new value called as predict instance in given attribute.

Step 13

After completion of one instance check next missing instance.

Procedure will be repeated until all value will be predicted.

Figure 1 Shows the ARA psedo code for prediction of the missing values.

4. IMPLEMENTATION

Coding Details

We were using datasets in ARFF format as an input to this algorithm and the ARA filter [2,7,8]. The filter would then take ARFF dataset as input and estimating out the missing values in the input dataset. After fixing out the missing values in the given dataset, it would apply the ARA algorithm and predict the missing values and also reconstruct the whole dataset from an incomplete dataset.

We have created an ARA filter class, which is an extension of the Simple Batch Filter class, which is an abstract class. Our algorithm first of all takes an ARFF format database as input then read how many attribute in given data set. It takes each attribute individually and writes it into array format. After that, it insert all instances into that array, including missing instances and find first missing instance, if it got the instance replace it by zero. After that it calculates average of all upper instances by using its weight effect on that particular instance. Nearest instance get more weight and weight will be decreases instance by instance. After that it also calculates average of all lower instances by using its weight effect on that particular instance, nearest instance get more weight and weight will be decreases instance by instance and vice-versa. Finally we are calculating the average of lower as well as upper instances.

5 EXPERIMENTAL SET UP

5.1 Approach

The objective of our experiment is to build the filter as a preprocessing step in Weka Workbench, which completes the data sets from missing data sets. We did not intentionally select those data sets in UCI

[12], which originally come with missing values because even if they do contain missing values, we don't know the accuracy of our approach. For experimental set up, we take the complete dataset from UCI repository [12], and then missing values are artificially added to the data sets to simulate MCAR missing values. To introduce m% missing values per attribute x_i in a dataset of size n, we randomly selected mn instances and replaced its x_i value with unknown i.e. ?(In WEKA, missing values are denoted as "?"). We use 10%,20% and 30% missingness for every dataset.

5.2 Results

After preprocessing steps, we use WEKA's M5Rules classifier for finding the error analysis. The classification was carried out on 10 fold cross validation technique. In Table 1, we have calculated the standard errors along with correlation coefficient on UCI[12] database repository. When observing the correlation coefficient of all the dataset i.e. CPU, Glass and Wine with missingness parameters – 10%,20% and 30%, it has been clear that, as missingness increases the accuracy of the classifier decreases.

S. N.	Error Analysis	Dataset								
		CPU			Glass			Wine		
		10% M	20% M	30% M	10% M	20% M	30% M	10% M	20% M	30% M
1	Correlation coefficient	0.88	0.66	0.47	0.62	0.59	0.62	0.78	0.78	0.75
2	Mean absolute error	36.03	53.85	60.50	0.77	0.76	0.81	144.89	140.69	147.70
3	Root mean squared error	74.03	116.22	144.70	2.31	2.39	2.18	198.67	188.66	204.41
4	Relative absolute error	43.63%	63.64%	79.63%	40.40%	39.99%	43.23%	57.76%	59.17%	62.67%
5	Root relative squared error	48.68%	75.88%	114.19%	78.06%	80.81%	79.32%	64.36%	63.10%	68.63%
6	Total Number of Instances	209	209	209	214	214	214	178	178	178

TABLE-1: After Applying the ARA Filter with M5Rules Classifier in WEKA Workbench on UCI [12] Datasets.

6. CONCLUSION

In this paper, we provided the theory and implementation details of a new filter viz. ARA in the WEKA workbench. As seen from the result, this simple but yet robust ARA filter works well to predict the missing data. We also demonstrate the efficacy of our approach by performing the analysis on real world UCI [12] data repositories. Thus it proves the extension as a preprocessing filter.

ACKNOWLEDGMENTS

Our special thanks to Mr. Peter Reutemann, of University of Waikato, fracpete@waikato.ac.nz, for providing us the support as and when required.

REFERENCES

1. S.Parthsarthy and C.C. Aggarwal, “On the Use of Conceptual Reconstruction for Mining Massively Incomplete Data Sets”,IEEE Trans. Knowledge and Data Eng., pp. 1512-1521,2003.

2. J. Quinlan, “C4.5: Programs for Machine Learning”, San Mateo, Calif.: Morgan Kaufmann, 1993.

3. http://weka.sourceforge.net/wiki/index.php/Writing_your_own_Filter

4. wekaWiki link : http://weka.sourceforge.net/wiki/index.php/Main_Page

5. S. Mehta,, S. Parthsarthy and H. Yang “Toward Unsupervised correlation preserving discretization”, IEEE Trans. Knowledge and Data Eng. pp.1174-1185 ,2005.

6. Ian H. Witten and Eibe Frank , “Data Mining: Practical Machine Learning Tools and Techniques” Second Edition, Morgan Kaufmann Publishers. ISBN: 81-312-0050-7.

7. http://weka.sourceforge.net/wiki/index.php/CVS

8. http://weka.sourceforge.net/wiki/index.php/Eclipse_3.0.x

9. weka.filters.SimpleBatchFilter

10. weka.filters.SimpleStreamFilter

11. R.J.A. Little and D. Rubin. “Statastical Analysis with Missing Data”. Ch. 3, pp-42-53,Wiley Series in Prob. and Stat., 2002.

12. UCI Machine Learning Repository, http://www.ics.uci.edu/umlearn/MLsummary.html

13. X. Zhu and X. Wu, “ Cost Constrained Data Acquisition for Intelligent Data Preparation”, IEEE Transactions on Knowledge and Data Engineering, Vol.17, Number 11, pp.1542-1556.

14. J. L. Schafer, “Analysis of Incomplete Multivariate Data”, Monographs on Stat and Applied Prob. 72, Chapman and Hall/CRC.

15. J. W. Grzymala-Busse and M.Hu. “A comparison of Several Approaches to Missing Attribute Values in Data Mining, Rough Sets and Current Trends in Computing”, 378-385, 2000.

16. C. J. Date and H. Darwen, “The Default Values approach to Missing Information,” Relational Database Writings 1989-1991, pp.343-354, 1989.