Reconstruction of a Complete Dataset from an Incomplete Dataset by Expectation Maximization Technique: Some Results

Sameer S. Prabhune[1], Dr. S.R. Sathe [2]

[1]Assistant Prof. & HOD- I.T S.S.G.M. College Of Engineering, Shegaon, 444 203 Maharashtra, India
[2]Professor, Dept. of E & CS, V.N.I.T.Nagpur, Maharashtra,India

Summary
Preprocessing is a crucial step 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 datasets. In this paper we describes an novel idea of predicting the missing values in the dataset by a well known principle of EM (Expectation Maximization) . After implementing and applying the EM filter, the dataset is completed with the estimated values, based on the well known principle of expected maximization of attribute instance. We demonstrate the efficacy of the approach on real data sets as a preprocessing step. The first section gives a brief introduction of the topic chosen for the implementation. In the second section we describe the preliminary tools that are required to develop this filter based on EM approach. In the third section we give the pseudo code for the EM technique for estimating the missing values. In the fourth section we discuss the implementation details for design and addition of this EM filter to WEKA workbench (WEKA 3-5-4 ver.). Lastly experimental results from real-world data sets demonstrate the effectiveness of our method.

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 imputation, statistical or regression based procedures [11].

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 machine learning estimation technique, expectation maximization i.e. EM [11], for predicting the missing values.

1.1 Contribution of this paper

This paper gives the novel idea for reconstruction of a complete dataset from an incomplete dataset by using well know principle of Expectation Maximization i.e. EM implemented in WEKA workbench. It gives the very precise results on real datasets of UCI [12].

2. Preliminary Tools Used

To complete our main objective, i.e. to develop the EM 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.EigenvalueDecom position; 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:

a) As the weka workbench is completely written in java and supports the java packages, it is useful to use java as the coding language.

Manuscript received November 5, 2010
Manuscript revised November 20, 2010

b) The second reason was that, we could use some classes from java package and some from weka package to create the filter.

3. Algorithm

This algorithm is designed to give the user an understanding of the EM algorithm. EM is a common technique for finding missing values to:
i) Predict missing values by most probable estimated values,
ii) Estimate parameters,
iii) Re- estimate the missing values assuming the new parameter estimates are correct,
iv) Re-estimate parameters, and so forth, iterating until convergence [11].
An expectation-maximization (EM) algorithm is used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. EM alternates between performing an expectation (E) step, which computes an expectation of the likelihood by including the latent variables as if they were observed, and maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step. The parameters found on the M step are then used to begin another E step, and the process is repeated. Refer Algorithm 1.

Algorithm 1: EM Algorithm for Missing Value Prediction

Missing Value predication

Ni[][]	-	The elements of AIFF file
Ai[][]	-	Array of instances
Rv	-	Number of rows
Cv	-	Number of columns
r	-	Iteration row counter
c	-	Iteration Column counter
miss	-	Instance of Ni
θ()	-	A function which varies according to the factor ruling the population
t	-	Iteration

Step 1:
Get data form file:
a) In our implementation we take ARFF format as input
b) It will be taken as one-dimensional array.

Step 2

Repeat Until (r<=Rv and c<=Cv)

a. Fetch the attribute from the data at first instance from Ni.
b. Array of instance is initially null.
c. Iteration starts from first instance.
d. Check. If(miss= =null)
e. If (miss= =null) then Ai[r][c]=0
f. Else Ai[r][c]=Ni[r][c]

Step 3:

Repeat Step 4 to Step 10 until all columns are called.

Step 4:

Take the probabilities as (ai+bi θ)……(an +bn θ) for each occurrence.

Step 5:

Calculate θ^{hat}=d/dθ (log(Ai(ai+bi θ+…………. Ai(ai+bi θ)))
$F(\theta) = Ai_{r-1}(b_{r-1})/(a_r + b_r)$

Step 6:

E step:
$Ai_r = Ai_{r-1}(F(\theta))$.

Step 7:

M-step
$\theta_{t+1} = \theta_{hat}$ at t^{th} iteration.

Step 8:

Convergence Step
Take θ_t=0.5
Repeat till $(\theta_{t+1} - (\theta_{hat} / \theta_t - \theta_{hat})) != (\theta_{t+2} - \theta_{hat} / \theta_{t+1} - \theta_{hat})$

Step 9:

At convergence, get actual θ_{hat}.
Put this in step 6 and 7 to get actual values.

Step 10:

a) After completion of one iteration, check next missing instance.
b) Repeat the procedure.

4. Implementation

We were using datasets in ARFF format as an input to this algorithm [2,7,8]. The filter would then take ARFF dataset as input and after finding out the missing values in the given dataset, we apply the EM filter and predict the missing values and also reconstruct the whole dataset.

Our code works only for numerical values. We have created an EM 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, apply the steps given in Algorithm 1.

5. Experimental Results

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], then deliberately deleted some values for making it as an incomplete datasets.

5.2 Results

In Table 1, we used the UCI [12] dataset CPU, in the original dataset, there are seven numeric attributes. The first column of Table 1 gives the original dataset values. In the second column of Table 1, we purposely deleted seven values for making it incomplete datasets. Finally in the third column, after applying the EM filter, we get the estimated values. These estimated values as compared to the original values are in the same domain, therefore, gives the expected results.

5.3 Limitations

There are two major limitations to EM during experimentation:

a) In some cases, with large fractions of missing information, it can be very slow to converge, and

b) In some problems, the M step is difficult (i.e. has no closed form) and then the theoretical simplicity of EM does not convert to practical simplicity.

Table 1

Missing Attribute Value Prediction Result *

Original dataset CPU	**Dataset with 7 missing values**	**Output after applying filter**
@relation 'cpu' @attribute MYCT real @attribute MMIN real @attribute MMAX real @attribute CACH real @attribute CHMIN real @attribute CHMAX real @attribute class real @data	@relation 'cpu' @attribute MYCT real @attribute MMIN real @attribute MMAX real @attribute CACH real @attribute CHMIN real @attribute CHMAX real @attribute class real @data	@relation cpu-weka.filters.unsupervised.attribute.EM @attribute MYCT numeric @attribute MMIN numeric @attribute MMAX numeric @attribute CACH numeric @attribute CHMIN numeric @attribute CHMAX numeric @attribute class numeric @data
125, 256, 6000, 256,16, 128,199	125, 256, 6000, 256, 16, 128, 199	125, 256, 6000, 256, 16, 128, 199
29, 8000,32000,32, 8, 32, 253	29, 8000,32000, 32, 8, 32, 253	29, 8000, 32000, 32, 8, 32, 253
29, 8000,32000,32, 8, 32, 253	29, **?,** 32000, 32, 8, **?**, 253	29, 8001.02, 32000, 32, 8, 33.00, 253
29, 8000,32000,32, 8, 32, 253	29, 8000,32000, 32, 8, 32, **?**	29, 8000, 32000, 32, 8, 32, 254.00
29, 8000,16000,32, 8, 16, 132	29, 8000, **?,** 32, 8, 16, 132	29, 8000, 32001.04 ,32, 8, 16, 132
26, 8000,32000,64, 8, 32, 290	26, 8000,32000, 64, 8, 32, 290	26, 8000, 32000, 64, 8, 32, 290
23, 16000,32000,64, 16, 32, 381	**?,** 16000,32000, **?,** 16, 32, 381	27.42,16000, 32000, 65.00,16, 32, 381
23, 16000,32000,64, 16, 32, 381	23, 16000,32000, 64, **?** , 32, 381	23, 16000, 32000, 64, 17.00, 32, 381
23, 16000,64000,64, 16, 32, 749	23, 16000, 64000, 64, 16, 32, 749	23, 16000, 64000, 64, 16, 32, 749
23, 32000,64000,128,32 , 64, 1238	23, 32000,64000,128 , 32, 64, 1238	23, 32000, 64000, 128, 32, 64, 1238

***CPU dataset (for 10 instances only) at UCI [12] repository.**

6. Conclusion

Expectation Maximization is used to recommend incomplete instances in a dataset for information completion, where attribute of instances mixing the missing information of different attributes are inheritably different and data is bounded by specific budget.

The design of our Algorithm distinguishes our work from existing approaches including basic two components:

a. The predicted value using EM algorithm is found to be either lying very close to real value or show an attribute relation.

b. We combine the weight and efficiency of each instance into unique economical, factor to explore the economical attribute for effective data acquisition.

7. Acknowledgements

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] wekaWikilink: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] 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. Little, D. Rubin. Statastical Analysis with Missing Data. Ch.8 , pp 164-172,Wiley Series in Probability and Statistics, 2002.
[12] UCI Machine Learning Repository, http://www.ics.uci.edu/umlearn/MLsummary.html

Dr. S. R. Sathe received the B.E.in 1987 and M.Tech. degrees in Computer Science and Engineering from IIT, Bombay in 1989. He completed his Ph.D.in Computer Sci. from Nagpur University in 2004.
He is currently working as a Professor in Computer Sci. in the Dept. Of Electronics and Computer Science Engineering at Visvesvaraya National Institute Of Technology, Nagpur. His research interests include Database System, Data Mining, Mobile Agent and Parallel Computing.

Sameer S. Prabhune received the B.E. and M.E. degrees in Computer Science and Engineering from SGB Amravati University in 1993 and 2000, respectively.
He is pursuing his Ph.D in Computer Science from Visvesvaraya National Institute Of Technology, Nagpur. He now with Shri Sant Gajanan Maharaj College Of Engineering, Shegaon,M.S., India. His research interests include Database System, Data Mining and Temporal Database. He is currently working on developing techniques to complete the incomplete data.