*International Journal of Computing Science and Information Technology*, 2018

ISSN: 2278-9669, April 2018 (http://ijcsit.org)

# Data Stream Prediction Using Ddsmi Algorithm

**Priya.S**[1]

1Research Scholar, Bharathiar University and Assistant Professor, Department of Computer Science, Government First Grade, College, KGF, priyapunith@yahoo.com

***Abstract***

*Dynamic Data Stream Multiple Imputation is divided into m equal-sized segments of s transactions, and processes the Imputation/update of data stream incrementally in a segment-based manner. DDSMI uses the simplified Hoeffding's bound concepts to calculate the appropriate data stream size for the mining of Missing Data. It then uses the comparison of the two data stream sub-range observations and the Data counts when a segment occurs within the data stream and then adjusts the data stream size appropriately. As a result, the usability of DDSMI is not affected by one variable ms.(minimum support threshold) A user may set then different ms at each time he/she makes a mining request, while the DDSMI has to work for defining the sizes of the data streams.*

*Keywords:* DDSMI algorithm,Multiple Imputation,Data Stream,Missing data

## I INTRODUCTION

Dynamic Data Stream Multiple Imputation is one of the main models for Missing Data Imputation in Data stream Mining , in which a fixed length of recently arrived data is considered. In a Dynamic Data Stream Multiple Imputation over a transactional data stream, by the arrival of a new transaction, the oldest transaction is removed from the data stream and the new transaction is inserted into the data stream. Therefore, it always contains the newest transactions. The data stream is usually stored and maintained within the main memory for fast processing. Due to unbounded amount of incoming traansactions and limited amount of memory, the data stream size must be limited. Since the cost of insertion and deletion of transaction is significant, segments of transactions can be added or removed from the data stream instead of individual transactions.

### A.*Missing Data Mining*

Missing Data mining is a traditional and important problem in data mining. Data is frequent if its support is not less than a threshold specified by users. Traditional Missing Data mining approaches have mainly considered the problem of mining static transaction databases. In these methods, transactions are stored in secondary storage so that multiple scans over the data can be performed.

**Dr.Antony Selvadoss Thanamani**[2]

2 Professor and Head, Research Department of Computer Science, NGM College, 90, Palghat Road, Pollachi - 642 001 Coimbatore District, Tamilnadu, India

### *B Types of Dynamic Data streams*

Instead of using synopses to compress the characteristics of the whole Dynamic Data Streams, Dynamic Data Stream techniques only look on a portion of the data. This approach is motivated by the idea that only the most recent data are relevant. Therefore, a Dynamic Data Stream continuously cuts out a part of the Dynamic Data Stream, e.g. the last ten Dynamic Data Stream elements, and only considers these elements during the processing. There are different types of Dynamic Data Stream models involved in mining Missing Data over data streams. They are namely, Landmark Data stream model, Tumbling Data stream model, Damped Data stream model, Titled Time Data stream model and Dynamic Data Stream Multiple Imputation model. Furthermore, the data streams can also be differentiated into element based to consider, e.g. the last ten elements, or time based data streams, e.g. to consider the last ten seconds of data.

### *C.Mathematical Models*

Due to the little space allowed for mining Missing Data from Dynamic Data Streams, the key

point becomes how to fix the boundary of the Dynamic Data Streams. Hoeffding's Bound is used for initializing the Dynamic Data Stream sizes with two values upper bound and lower bound. The Heine-Borel, The Bolzano–Weierstrass and the Bounded-Monotone sequence uses the combinatorial approach for describing the changes of Dynamic Data Stream sizes. Whenever the speed of transactional has been increased, the size of the Dynamic Data Stream should be updated. This can be achieved by defining the bounds infimum and supermum from these mathematical models.

Overview of Research

In the domain of high speed Dynamic Data Streams where the volumes of incoming data is enormous, an effective data structure is required to store, update and retrieve essential information. This is mainly because of memory constraints and the huge amount of incoming data in case of Dynamic Data Streams. The research work further is divided into following phases.

- Segment Initialization
- Hoeffding's-Bound Based Dynamic Data Stream prediction
- Data stream limits estimated by Mathematical models

*DDSMI Algorithm*

Problem Definitions

Let I = {x1, x2,…xz} be the set of items (or attributes) which may occur in a source of data stream. An Data (or a pattern) X is a subset of I and written as X = xi,xj,... xm. The length (i.e., number of items) of an Data X is denoted by |X|. And a transaction T is a set of items, and T supports an Data X if $X \subseteq T$. A transactional data stream DS is a sequence of continuously incoming transactions, in which every transaction is one basic element of DS. A data stream in the data stream is a time interval which covers a set of successive w transactions. A Dynamic Data Stream Multiple Imputation W in the data stream is a data stream of most recent w transactions which Dynamic Data Stream forward for transactions, where w denotes the size of W. The notation Il used to denote the set of all possible Data of length l (that is, l- itemsets) together with their respective counts in a set of transactions. In addition, Tnuse to denote the latest transaction in the current data stream. Thus, the current data stream is W = {Tn-w+1,…Tn}.

Definition (Segment in-out): Let Sc denotes the current segment which is going to be inserted into the data stream next (after it is full of sseg transactions). A segment in-out operation (of the data stream) first insert Sc into and then extract Sn-m+1 from the original data stream, where n denotes the ID of latest segment in the original data stream. Therefore, the data streams before and after a Imputation are W = {Sn-m+1,….,Sn} and W = {Sn-m+2,…Sn, Sc}, respectively.

Now by this segment-based manner of Imputation, at each Imputation we insert the new segment and delete (or drop out) the earliest segment, which respectively contain the base summaries (i.e., I1 and I2, 1-items and 2-Data together with their respective counts) of data stream belonging to both segments. With this approach, there is no need to maintain the whole transactions within the current data stream in memory all along, while the Imputation of (segment-based) data stream is still feasible. In addition, the parameter m directly affects the consumption of memory and is now remarked. A larger value of m means the data stream will Dynamic Data Stream/update more frequently (since each segment on an average contains fewer transactions), while the increasing overhead of the memory space is also considerable. In our opinion, an adequate size of m that falls in the range between 5 and 20 may be suitable for the general data streams.

The proposed method would approximate the counts of Data and discover FIs over the Dynamic Data Stream Multiple Imputation of a data stream. This method now processes data stream Imputation in a segment-based fashion. The proposed data-stream mining algorithm, namely the Dynamic Data Stream Multiple Imputation with Combinatorial Approximation (DDSMI) algorithm is described as follows.

**II Algorithm DDSMI**

**Input**: A transactional data stream (DS), a minimum-support threshold (ms), and a Imputation-data stream size (w)

**Output:** A list of Missing Data (F)

**Method**:

Divide the data stream conceptually into m segments (where $5 \leq m \geq 20$);

Set up a segment in-out pointer sp;

while data of DS is still streaming in do begin

set F to be empty;

while no request from the user do begin

Fetch the next incoming transaction T from DS;

Enumerate all subsets of T and record increase their counts in Sc;

if the length of T is over 2 then begin

Enumerate all 3-subsets of T and record their counts in Sc;

end if

if Sc is full of sseg transactions then begin

Discard all kept 3-Data in Sc;

Insert Sc as the latest segment Sn into L according to sp;

if the number of segments in L is greater than m then begin

Delete the oldest segment from L according to sp;

end if

Create a new Sc for the next round;

Set sp circularly to point to the next filed;

end if

end while

Merge the m segments (in L) to obtain the count- values for Data in the

current data stream;

Find all large 1-items and 2-Data and insert them into F;

foreach frequent 2-Data X in F do begin

foreach segment S in the data stream do begin

Aggregate the count of each 3-superset of X with its sum of counts so far respectively;

end foreach

Insert every frequent 3-Data into F;

end foreach

foreach frequent 3-Data Y in F do begin

repeat

$n \rightarrow 4$;

Calculate the counts of n-Data with m = n and k = 3;

Insert every frequent n-Data into F;

$n \leftarrow n + 1$;

until there is no frequent n-Data generated

end foreach

**Output F as the mining result;**

end while

In the DDSMI method, the Dynamic Data Stream Multiple Imputation is divided into m equal-sized segments of s transactions, and processes the Imputation/update of data stream incrementally in a segment-based manner. The data structure used to keep the base summary of data stream is now a lexicographic-ordered prefix tree modified. This data structure maintains the base summary, i.e., I1 and I2 (in this research), over the current Dynamic Data Stream Multiple Imputation in a segment-based fashion too. Besides, for the current segment of transactions, i.e., Sc, its base summary is further kept separately in an array. In the array, we also maintain the whole I3 of Sc for the purpose of calculating and finding the Missing Data.

The DDSMI algorithm processes on an on-line transactional data stream. As long as there is no mining request from the user, the DDSMI continues receiving and processing the incoming transactions one by one, and handles the data stream Imputation in a segment- based manner. For each incoming transaction T in the current segment Sc, the DDSMI enumerates and records the first- three orders of subsets contained in T. When Sc is full of sseg transactions, for each 2-Data X which is not in Sc, DDSMI calculates FI. After the chosen I3 is obtained totally, a segment in-out operation is finally performed. To insert Sc into the data stream, only I1, I2, and the chosen I3 are updated into the tree, while the information of original I3 over Scis discarded.

For each Data X belonging to the base summary, the corresponding node in the tree includes then a circular array of size m which corresponds to the m segments of the Dynamic Data Stream Multiple Imputation , andX's count over the current data stream is recorded respectively in these m fields of the array. If we combine the counts of all the segments, the count of X over the current data stream is obtained. There is a (global) pointer to indicate in which field the count of X over Sc will be stored in. When the current segment Sc is full and a segment in-out operation is going to be performed, the count of X over Sc is then stored into the field of array indicated by the pointer, and that which is over the earliest segment is then dropped out naturally since it is replaced by the newly stored value as regards Sc.

In the rest of the portion of the approximation task, the DDSMI uniformly employs I1 and I2 (i.e., the base summary) plus the approximated I3 to approximate for the Data with longer length. The process of approximation proceeds in both depth-first order and lexicographic order.

That is, for any two 3-Data Y1 and Y2, if then Y2 comes after Y1 in lexicographic order, then before DDSMI starts processing Y2, all Data having Y1 as their prefix have then been processed already.

As the data-stream mining method should work with a changeable value of ms, which is a requirement that our proposed method needs to satisfy. Now DDSMI meets this requirement adequately. As mentioned before, in the data structure of the proposed method, we maintain no more than I1 and I2 (and the chosen I3) over the current data stream, and the mining process proceeds by approximating the counts of Data from such a base-summary data structure. It uses embedded techniques to approximate itemsets' counts according to its base-summary data structure, and then selects the frequent ones in terms of the respective ms. As a result, the usability of DDSMI is not affected by one variable ms. A user may set then different ms at each time he/she makes a mining request, whilst the DDSMI has to work for defining the sizes of the data streams.

III CONCLUSION

The DDSMI algorithm relies mainly on a verifier function and it is an exact and efficient algorithm for mining very large Dynamic Data Stream Multiple Imputations over data streams. The performance of the DDSMI improves when small delays are allowed in reporting new Missing Data; however this delay can be set to 0 with a small performance overhead. In this paper DDSMI algorithm is very efficient and the result is good and accurate.future work is to improve DDSMI algorithm using hoeffdings bound to gradually reduce the average of data stream size and to reduce cost.

***REFERENCES***


[1] Tensor Voting Techniques and Applications in Mobile Trace Inference, IEEE Access Special Section On Artificial Intelligence Enable Networking, Volume 3, 2015 Received October 30, 2015, accepted November 16, 2015, date of publication December 24, 2015, date of current version January 7, 2016.ERTE

PAN, (Student Member, IEEE), MIAO PAN, (Member, IEEE), AND ZHU HAN, (Fellow, IEEE)

[2] Cluster Based Mean Imputation International Journal of Research and Reviews

in Applicable Mathematics & Computer Science. Vol 2.No.1,2012,Ms.R.Malarvizhi and Dr.Antony Selvadoss Thanamani

[3] Bayesian Learning of Noisy Markov Decision Processes, ACM Transactions on Modeling and Computer Simulation Vol. 23, No. 1, Article 4, Publication date: January 2013.SUMEETPAL S. SINGH, University of Cambridge

[4] Estimating Burned Area in Mato Grosso, Brazil, Using an Object-Based Classification Method on a Systematic Sample of Medium Resolution Satellite Images ,IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, VOL. 8, NO. 9, SEPTEMBER 2015,Yosio Edemir Shimabukuro, Jukka Miettinen, René Beuchle, Rosana

Cristina Grecchi,Dario Simonetti, and Frédéric Achard

[5] On-Line PMU-Based Transmission Line Parameter Identification, CSEE JOURNAL OF POWER AND ENERGY SYSTEMS,VOL. 1, NO. 2, JUNE 2015,Xuanyu Zhao, Huafeng Zhou, Di Shi, Huashi Zhao, Chaoyang Jing, Chris Jones

[6] Cluster Based Mean Imputation, International Journal of Research and Reviews

in Applicable Mathematics& Computer Science. Vol 2.No.1,2012,Ms.R.Malarvizhi and Dr.Antony Selvadoss Thanamani

[7] K-NN Classifier Performs Better Than K-Means Clustering in Missing Value Imputation, International Journal for Research in Science & Advanced Technologies,Vol 1.Issue- 2,2013,Ms.R.Malarvizhi and Dr.Antony Selvadoss Thanamani.

[8] Classification of Efficient Imputation Method for Analyzing Missing Values,

International Journal of Computer Trends and Technology(IJCTT),Vol 12.No.4-Jun 2014 ,S.Kanchana and Dr.Antony Selvadoss Thanamani.