Using MSMIA Algorithm For Finding Missing Value Handleing Boeing Data Set

P. Logeshwari, Dr. Antony Selvadoss Thanamani

Abstract

Main Stream Data 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 Main Stream Data 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 transactions 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. The MSMIA Algorithm

The Main stream Data Multiple Imputation Algorithm (MSMIA) always maintains a union of the Missing Data of all Imputes in the current data stream W, called Segment(S), which is guaranteed to be a superset of the Missing Data over W. Upon arrival of a new Impute and expiration of an old one, we update the true count of each segment in S, by considering its frequency in both the expired Impute and the new slide. To assure that S contains all Data that are frequent in at least one of the Imputes of the current data stream $*''_i(\sigma_{\alpha}(S_i))$, we must also mine the new Impute and add its Missing Data to S. The difficulty is that when a new segment is added to S for the first time, its true frequency in the whole data stream is not known, mostly since this segment wasn't frequent in the previous n -1 Imputes. To address this problem, MSMIA uses an auxiliary array, aux array, for each new segment in the new slide.

The aux array now stores the frequency of a segment in each data stream starting at a particular Impute in the current data stream. In other words, the aux array stores the frequency of a segment for each data stream, for which the frequency is not known. The key point in this is that this counting can either be done eagerly or lazily. Under the laziest approach, we wait until a Impute expires and then compute the frequency of such new Data over this Impute and update the aux arrays accordingly.

This further saves many additional passes through the data stream. The pseudo code for the MSMIA algorithm is given in Figure A1. At the end of each slide, MSMIA outputs all Data in S whose frequency at that time is. However few Data will be missed due to the lack of knowledge at the time of the output, but it will then be reported as delayed when other Imputes expire.

[1]Research scholar, Department of Computer Science, NGM College Pollachi. Email : tppselvalogu@gmail.com

[2]Associate Professor and Head, Department of Computer Science, NGM College Pollachi. Email : selvdoss@gmail.com

For Each New Impute S

1: For each segment $s \in S$
update $s.freq$ over S

2: Mine S to compute $\sigma_\alpha(S)$

3: For each existing segment $s \in \sigma_\alpha(S) \cap S$
remember S as the last Impute in which s is frequent

4: For each new segment $s \in \sigma_\alpha(S) \backslash S$
$S \leftarrow S \in \{s\}$
remember S as the first Impute in which s is frequent
create auxiliary array for s and start monitoring it

For Each Expiring Impute S

5: For each segment $s \in S$
update $s.freq$, if S has been counted in
update $s.aux\ array$, if applicable

report s as delayed, if frequent but not reported

at query time

delete $s.aux\ array$, if s has existed since arrival of S

delete s, if s no longer frequent in any of the current slides

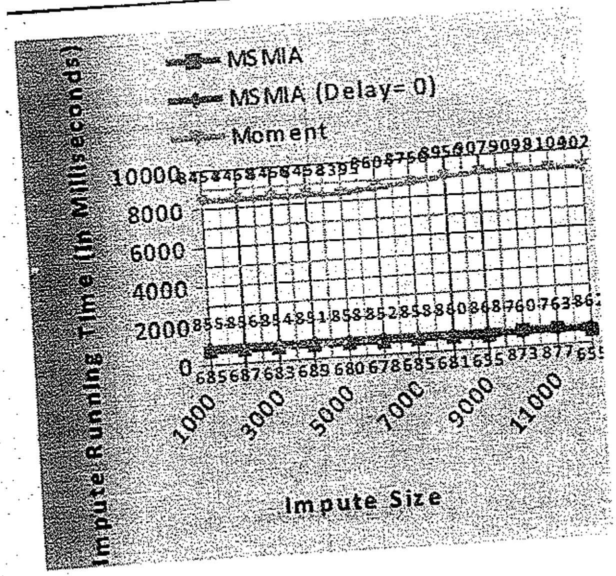
Fig.A1 MSMIA pseudo code.

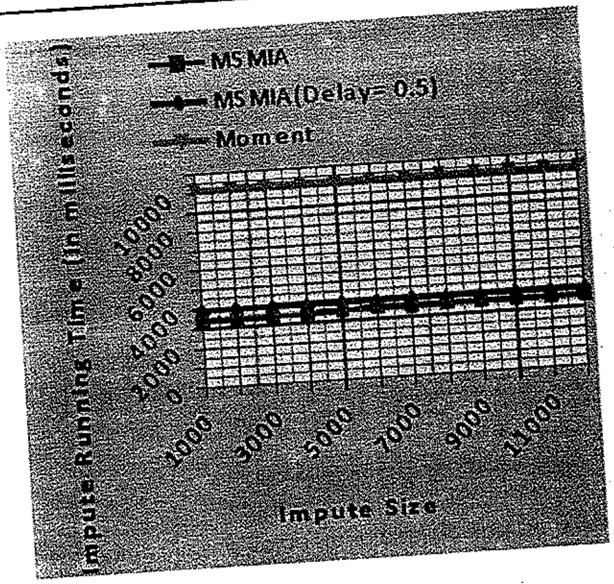
Implications of MSMIA algorithm with Boeing Data Set

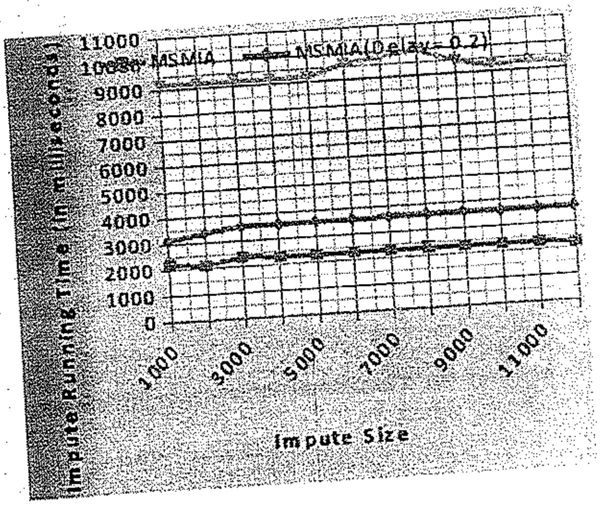
The MSMIA algorithm compared with the Moment, is applied to the Real-world Normalized Large dataset of Boeing which fixes the data stream size to 100K transactions. Furthermore, the support thresholds set to 2% and vary the Impute size to measure the scalability of these algorithms. As shown in Figure A3 (a), (b), (c) and (d) MSMIA is much more scalable with versions MSMIA and MSMIA (Delay) algorithms, one with maximum data stream size delay and the other one without any delay, are much faster than Moment. The MSMIA algorithm is intended for incremental maintenance of Missing Data, it is best suitable for online and real-time processing of millions of transactions. The proposed algorithm however is aimed at maintaining Missing Data over large main stream Data Multiple Imputations. In fact, the proposed algorithm can handle a Impute size of up to 1000 million transactions (Large Data).

Inpute sizes	1000	2000	3000	4000	5000	6000	7000	8000	9000	16000	11000	12000
MSMIA	685	687	683	689	680	678	685	681	655	873	877	655
MSMIA (Delay=0)	855	856	854	851	858	852	858	860	868	760	763	862
Moment	8454	8458	8456	8458	8395	8604	8750	8950	9075	9098	9104	9025
MSMIA	2190	2110	2370	2304	2311	2307	2311	2329	2306	2304	2311	2230
MSMIA (Delay=0.2)	3115	3352	3553	3551	3566	3574	3585	3596	3627	3588	3603	3622
Moment	9154	9174	9187	9156	9145	9587	9765	9944	9462	9204	9306	9225
MSMIA	3165	3185	3223	3258	3246	3276	3254	3232	3229	3198	3265	3289
MSMIA (Delay=0.4)	3535	3555	3575	3575	3576	3577	3577	3577	3578	3578	3578	3579
Moment	9078	9142	9144	9204	9242	9234	9342	9351	9415	9427	9515	9504
MSMIA	3405	3435	3445	3456	3546	3576	3555	3613	3609	3616	3627	3646
MSMIA (Delay=0.5)	3930	3950	3970	3971	3968	3973	3972	3969	3978	3980	3984	3998
Moment	10179	10199	10219	10217	10207	10216	10198	10231	10229	10236	10242	10249

Table T2 Comparison of MSMIA, MSMIA (Delay) and Moment with various Impute sizes for Boeing Data Set.







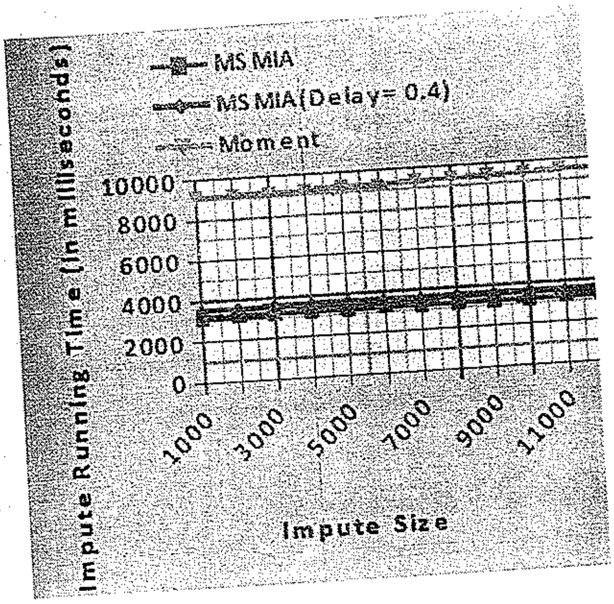


Figure A3 Implications of MSMIA algorithm with Boeing Data Set

The result reveals that the MSMIA algorithm holds well even in cases of Large and Normalized data set. As Boeing dataset belongs to the category of Large data set, being the data size more than 160 million bytes – A data set is declared as Large data set if the size is more than 100 million bytes. The performance of MSMIA algorithm is good for the cases of delay < 0.4 where there the deflections are tangible in the graph scale.

Conclusions

The Data Stream is usually stored and maintained within the main memory for fast processing. Due to unbounded amount of incoming transactions 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. In this paper I am Implementing MSMIA algorithm with Boeing Data

Set, used to Comparison of MSMIA, MSMIA (Delay) and Moment with various Impute sizes for Boeing Data Set.

References

[1] 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.

[2] 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.

[3] 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

[4] Multible Imputation of Missing Data Using Efficient Machine Learing Approach International Journal of Applied Engineering Research,Vol 1.No.1, 2015, S.Kanchana and Dr.Antony Selvadoss Thanamani

[5] 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.

AUTHORS BIOGRAPHY:

Mrs. P. Logeswari received her MCA., degree in computer Science from Sree Saraswathi Thiyagaraja College of arts and science, Pollachi, India in 2010. She completed her M.Phil., degree in computer Science from Sree Saraswathi Thiyagaraja College of arts and science, Pollachi, India on 2012. Presently she is pursuing PhD (Full Time) degree in Computer Science in NGM College (Autonomous), Pollachi under Bharathiar University, Coimbatore. She served as a Faculty of Computer Science at Government Arts College Udumalpet, from 2012 to 2013 and she served as a Faculty of Computer Science at Sree Ramu College of Arts and Science, NM Sunggam, Pollachi ,India. from April 2013 to August 2014. She has presented papers in International/National conferences and published two papers in International journal. Her research focuses on Data Mining.

Dr. Antony Selvadoss Thanamani is presently working as Professor and Head, Dept of Computer Science, NGM College, Coimbatore, India (affiliated to Bharathiar University, Coimbatore). He has published more than 100 papers in international/ national journals and conferences. He has authored many books on recent trends in Information Technology. His areas of interest include E-Learning, Knowledge.