Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

The Analysis of PROC_IMA Using Standard Warehouse

Ashish Saxena[1], Prof. (Col).Gurmit Singh[2]
[1]Ph.D. Scholar Shepherd Institute of Engg, &Technology, SHIATS (AAI-DU), Allahabad, India
[2]Prof Emeritus, Deptt of Computer Sc. and Information Tech, SSET, SHIATS (AAI- DU) NAINI, Allahabad, India

Abstract—Incremental data mining is used for large olap .The algorithm plays an important role in the performance of data mining for this a new algorithm PROC _IMA has been introduced. It uses four passes one extra pass using count function.

Keywords— PROC_IMA ,DELTA, COUNT,INCREMENTAL MINING

I. INTRODUCTION

The incremental data mining algorithm which was earlier developed used only three passes .The efficiency of DELTA is Lesser as large no data is required for data processing in this research paper analysis of the existing DELTA incremental data mining algorithm is done with the proposed PROC_IMA .Evaluation of the incremental data mining algorithm (PROC_IMA) developed in the study and its comparison with the existing model (DELTA), using time series model and in terms of various statistical tests /models.

II. REVIEW OF LITERATURE

The large no of algorithm ware developed for incremental data mining for extracting the golden set of rules for pattern analysis. **Washio et.al.(2005)** worked on Association rules based on level wise subspace clustering.
X. Su et.al. (2009) worked on afast incremental clustering algorithm.**Saaret.al.(2007)** worked on *active feature-value acquisition*.**Sunget.al (2011**) worked on forecasting changes in Korea Composite Stock Price Index (KOSPI) using association rules. **Huang et.al(2012)** worked on a Short Chinese Text Incremental Clustering Algorithm Based on Weighted Semantics and Naive Bayes**Po et.al (2004)** worked on image Database Design Based On 4D-Spa Representation for Spatial Relations.

THE PROPOSED ALGORITHM (PROC _IMA)

For developing more efficient algorithm for incremental datamining, the new function count was introduced in (DELTA).The synthetic standard warehouse *T10I4D100K* of about 50 million data is used as incremental warehouse .The algorithms were applied on it by using programming these algorithm in MATLAB7.6 .The concurrent values of F_{DB} and N_{DB} are obtained which is the support of algorithm the support plays an important role in the performance of algorithm .The statistical analysis of the results obtained by MATLAB 7.6 has been done on SPSS16.0 .The comparative graphs of DELTA and PROC_IMA algorithm along with statistical analysis are generated in SPSS16.0 and comparison is made between them.
The DELTA algorithm used only three passes. The proposed algorithm in 1st pass reads previous database and then increments *d(database element)* (db is increment in datawarahouse , DB is existing datawarehouse and (DB + db) is the current datawarehouse)using the *updatecount function* .Some item sets becomes frequent in N(Negative border database element) .
In the IInd pass the infrequent itemsets are denoted in *fknown*. These itemsets are extracted using the function *Getfrequent* .The infrequent itemsets are calculated by *Infrequent =(fdb+ndb)- $f_{known.}$* These itemsets are used for pruning in the second pass of the algorithm. If some itemsets do not move from ndb to fknown , then the negative border of ndb of *fknown* is computed by using apriori function. Itemsets in *nknown* with unknown counts are stored in n1 thus the remaining counts are all infrequent. Any itemset in n1 and their extension are computed. If there are no element in n1 their extensions. Any itemset which is not frequent in db cannot be frequent in(DB+db).
In 3rd pass all possible extensions of f_{known} which are in form (f(DB+db)U n(DB+db))and store them in set count c. This is done by computing the layers of negative borders closure of fknown .It is expected that all the other remaining layers can be generated together since the number of the two itemsets in fknown is typically much smaller than the total number of 2-itemsets pairs. Initially C(Count) is reset to zero using the function of reset count. Then at every stage of computation of closure, those itemsets that are Infrequent and Infrequent db are removed so , that none of the extension is generated. Itemsets from f_{known} and n1 are removed from C. In this pass the counts within db of the remaining itemsets C are computed.

The 4th pass scans the set of f_{DBudb}(frequent itemsets) and n_{DBudb}(Negative Border) values are returned , which gives a unique value of support .This value of support is used to evaluate the performance of algorithm.

PROCEDURE PROC_IMA **(DB,db,f_{DB},n_{DB})**

Coding of Procedure IMA in MATLAB7.6

```
% proposed algorithm for incremental mining over set of frequent itemsets and negative border
function[f, n]= PROC_IMA (DB,db,fdb,ndb,n1)
tic
  load T40I10D100K.mat;
loada30.txt;   %IST PASS OF PROPOSED ALGORITHM
 scandb = 300;
   fdb=300;
   ndb=210;
   c=1;
   nu=1;
     updatecount=fdb+ndb;
  fknown= fdb-ndb;
  infrequent =(fdb+ndb)-fknown; %IIND PASS
  if (fdb==fknown)
     f= fdb;
     n=ndb;
     scandb;
  else
     nknown= updatecount;
     if nknown <= infrequent
        f=fknown;
        n=nknown;
     else
        nu= nknown-infrequent;
        updatecount=nu
        c=nu; ;%IIIRD PASS  OF PROC_IMA ALGORITHM
        scandb;
        infdb= nu-c;
        while  c>0
           c=c+ fknown;
              c=c+updatecount;
              c=c-(infrequent + infdb);
      c=c-fknown+nu;
            if (c>0)
             c = updatecount;
            end
        scandb=c+nu;
        n1= updatecount-infrequent;
        updatecount=n1+ scandb;
        fd= fknown+ scandb;
        nd=updatecount;
        scandb;          %  IVTH   PASS
        f=fd;
        n=nd;
        toc
  return
  end
     end
  end
```

The standard synthetic data warehouse T40I10d100K is used as incremental data warehouse containing online 50 billion incremental dataset.

THE UTILITY OF IIIRD EXTRA PROPOSED PASS

On performing the simulation of PROC_IMA , the results are obtained in form of (fdb,ndb) pair which is the unique value of support of the algorithm ,which is used for analysis and performance evaluation of a data mining algorithm .After the first pass over the increment, Namely, (fdb, ndb) the counts of these itemsets with respect to the increment is calculated by computing difference between the updated count and original count .After this computation ,the itemsets that turn out to be frequent in db are gathered together and their negative border is computed. If counts are within db and some itemsets in the negative border are unknown, these counts are detected during second and third pass by using count function .The counts are reset initially by using the ResetCount Function. After this computation of the Negative-border closure which are determined during the second pass. The counts within db of the itemsets in the closure are determined. During the third pass, the identities and counts within db of itemsets in (fdb+ndb) are extracted from closure . If any itemset is candidate then for computing (fdb+ ndb) as well as updated (f_{DBudb} + n_{DBudb}) then to ensure that there is no duplicate counting of all such common itemsets in support of DB , support in db is computed . Then, first count is incremented by this value. The optimization is performed only in routine that access the database and do not effect the structure of incremental mining algorithm. The second optimization is performed before each pass over the increment of previous database. The successor of items that are not from candidate are totally removed from the arrays of successor that are earlier computed during the first optimization.The third optimization is performed only once in the increment. At this stage the identities of all potentially frequent 2- item sets (DB+db) are found . Hence, no next candidate 2- itemsets will be generated. Among the potentially frequent 2- itemsets in third pass will be pruned. These newly pruned itemsets will have the same property which is not present in itemsets and its ancestors. Thus, the third pass increases the efficiency of PROC_IMA as optimization is performed in third pass which was not done in case of DELTA .This optimization causes the increase in the efficiency of the proposed algorithm. In the third pass the *count* initially resets to zero as soon as the itemsets are read by algorithm which are stored in array .Then for every increment ,the *re-mining* takes place which updates the existing association rules and scanning of whole database is done .It causes the increase in the support and there by increases its performance.

III. FINDINGS

The general incremental mining algorithm used for mining used only three passes. The new algorithm is proposed which is more efficient and less time consuming as compared to existing algorithm (DELTA) . The proposed algorithm IMA in 1st pass reads previous datasets and then increments it and again reads previous frequent itemsets and then calculates the negative boarder along with minimum support threshold or minsup. It then updates the negative boarder and returns itemset used for next pass (2nd Pass) .Second pass then causes pruning if negative boarder passes the closure property. Then pruning takes place until frequent item sets do not under grow.

The updation of count takes place in this pass. In third pass the counts are updated by using the resetcount function & scanning of database takes place by knowing parameters of frequent data itemsets and even if negative boarder exists then pruning is done till the end of negative boarder. The 4th pass scans the set of minsup *| DB + db | and returns various values in the form of support.

The PROC_IMA and DELTA were coded in MATLAB 7.6 and a large data synthetic standard data warehouse T10I4D100K was used. The small increment of data warehouse (db) was made of 25,50,100 etc set of data as incremental dataset was used then PROC_IMA performance with the DELTA algorithm was compared in terms of support which reveals that the performance of algorithm PROC_IMA is higher than DELTA. The following values are obtained while simulating MATLAB7.6 ON performing the ACF analysis of DELTA and PROC_IMA was done on SPSS16.0 we get the following result.

Table 1 ACF analysis of DELTA obtained in SPSS16.00

Input	Data	C:\Documents and Settings\ASHISH SAXENA\Desktop\FINALAMRESHPRINT16102014\ALGO1\2012pd.sav
Syntax		ACF VARIABLES=DELTA /NOLOG /MXAUTO 16 /SERROR=IND /PACF.
Time Series Settings (TSET)	Amount of Output	PRINT = DEFAULT
	Saving New Variables	NEWVAR = NONE
	Maximum Number of Lags in Autocorrelation or Partial Autocorrelation Plots	MXAUTO = 16

	Maximum Number of Lags Per Cross-Correlation Plots	MXCROSS = 7
	Maximum Number of New Variables Generated Per Procedure	MXNEWVAR = 60
	Maximum Number of New Cases Per Procedure	MXPREDICT = 1000
	Treatment of User-Missing Values	MISSING = EXCLUDE
	Confidence Interval Percentage Value	CIN = 95
	Tolerance for Entering Variables in Regression Equations	TOLER = .0001
	Maximum Iterative Parameter Change	CNVERGE = .001
	Method of Calculating Std. Errors for Autocorrelations	ACFSE = IND
	Length of Seasonal Period	Unspecified
	Variable Whose Values Label Observations in Plots	Unspecified
	Equations Include	CONSTANT

(a)

Model Description		
Model Name		MOD_5
Series Name	1	DELTA
Transformation		None
Non-Seasonal Differencing		0
Seasonal Differencing		0
Length of Seasonal Period		No periodicity
Maximum Number of Lags		16
Process Assumed for Calculating the Standard Errors of the Autocorrelations		Independence(white noise)a
Display and Plot		All lags
Applying the model specifications from MOD_5		
a. Not applicable for calculating the standard errors of the partial autocorrelations.		

(b)

Case Processing Summary		DELTA
Series Length		24
Number of Missing Values	User-Missing	0
	System-Missing	0
Number of Valid Values		24
Number of Computable First Lags		23

(c)

Autocorrelations					
Series:DELTA					
Lag	Autocorrelation	Std. Errora	Box-Ljung Statistic		
			Value	df	Sig.b
1	.377	.192	3.864	1	.049
2	.259	.188	5.760	2	.056
3	.036	.179	6.836	4	.145
a. The underlying process assumed is independence (white noise).					
b. Based on the asymptotic chi-square approximation.					

(d)

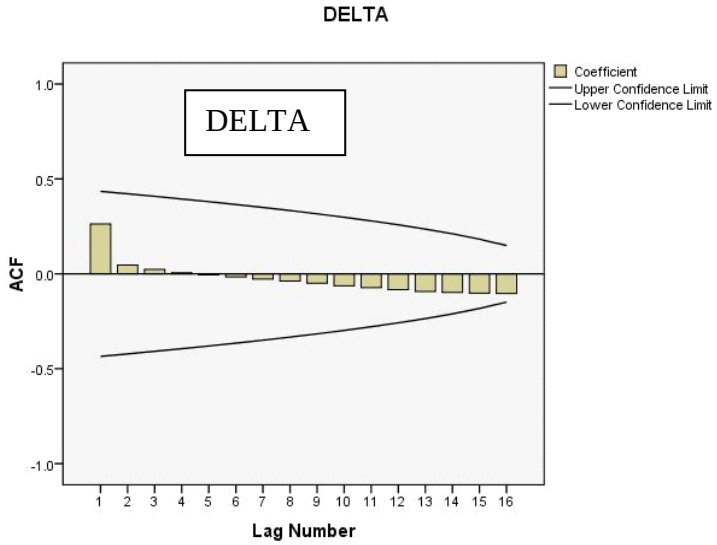

Fig 1: ACF of DELTA

The above figure of ACF of DELTA is generated on SPSS 16.0.It is clear that the DELTA has more deviation from normal there by having low performance .The value of upper confidence limit is less and the lower confidence limit increases as the lag number increases .It causes decrease in the efficiency of mining as the dataset increase in incremental data mining, from the figure 4.9 it is evident that DELTA efficiency decreases as the lag number increases.

Table 2ACF analysis of PROC_IMA on SPSS16.00

Case Processing Summary		IMA
Series Length		24
Number of Missing Values	User-Missing	0
	System-Missing	0
Number of Valid Values		24
Number of Computable First Lags		23
Output Created		2013-07-25T10:34:38.781
Comments		
Input	Data	C:\Documents and Settings\ASHISH SAXENA\Desktop\FINALAMRESH PRINT16102011\ALGO1\2012pd.sav
	Maximum Number of New Cases Per Procedure	MXPREDICT = 1000
	Treatment of User-Missing Values	MISSING = EXCLUDE
	Confidence Interval Percentage Value	CIN = 95
	Tolerance for Entering Variables in Regression Equations	TOLER = .0001
	Maximum Iterative Parameter Change	CNVERGE = .001
	Method of Calculating Std. Errors for Autocorrelations	ACFSE = IND
	Length of Seasonal Period	Unspecified
	Variable Whose Values Label Observations in Plots	Unspecified
	Equations Include	CONSTANT
Model Description		

Model Name		MOD_6
Series Name	1	IMA
Transformation		None
Length of Seasonal Period		No periodicity
Maximum Number of Lags		16
Process Assumed for Calculating the Standard Errors of the Autocorrelations		Independence(white noise)a
Display and Plot		All lags

Table 3 ACF analysis Using Box-Ljung Statistic of PROC_IMA on SPSS16.00

Autocorrelations					
Series:IMA					
Lag	Autocorrelation	Std. Errora	Box-Ljung Statistic		
			Value	df	Sig.b
1	.721	.192	14.107	1	.000
2	.476	.188	20.521	2	.000
3	.282	.183	22.879	3	.000
4	.054	.179	22.968	4	.000
5	-.018	.174	22.979	5	.000
6	-.024	.170	22.999	6	.001
7	-.055	.165	23.112	7	.002
a. The underlying process assumed is independence (white noise).					
b. Based on the asymptotic chi-square approximation.					

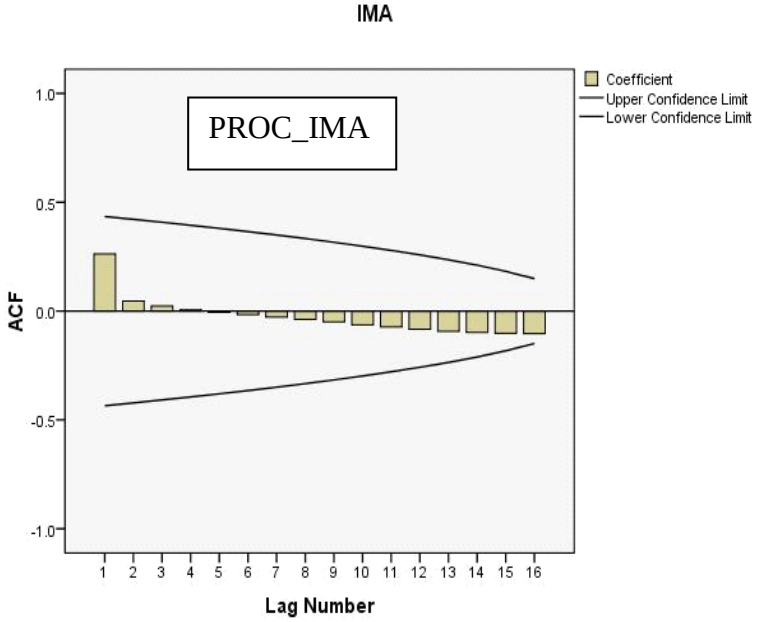

Fig 2 ACF of PROC_IMA

From the ACF it is clear that PROC_IMA is having less deviation from normal so ,PROC_ IMA has more efficient then as compared with DELTA. The value of upper confidence level for DELTA is higher as the autocorrelation function lag number increases .The lower confidence level gradually decreases as the lag number increases. Thus, it is evident from the figure 1 and 2 the PROC_IMA holds higher performance than that of DELTA.

IV. CONCUSION

From the above Auto correlation function analysis of DELTA and PROC_ IMA over standard synthetic data warehouse T40I10d100K, it is observed that lag increases in case of DELTA as data concentration increases in case large data warehouse which is decreases the efficiency of DELTA while in case of PROC _ IMA lag does not increase therefore the PROC _IMA has higher efficiency than that of DELTA.The PROC_IMA can be used for incremental data mining in the field of advance simulation, satellite imaging ,pattern recognition etc.

REFRENCFES

[1] **Washio T, Nakanishi K, Motoda H (2005)** Association rules based on level wise subspace clustering. In: Proceedings. of 9th European conference on principles and practice of knowledge discovery in databases. LNAI, vol 3721,692–700p, Springer, Heidelberg, ***2005***

[2] **X. Su, Y. Lan,R. Wan, Y. Qin, (2009**) A fast incremental clustering algorithm, Proc. of International Symposium on Information Processing, 2009, pp 175-178,***2009***

[3] **Saar-Tsechansky, M., Melville, P., and Provost, F. J.(2007)** *"Active feature-value acquisition"*. Tech. Rep. IROM-08-06, University of Texas at Austin, McCombs Research Paper Series, Sept. ***2007***

[4] **Sung, H.N., So, Y. S, (2011**) " Forecasting changes in Korea Composite Stock Price Index (KOSPI) using association rules", Expert Systems with Applications, vol. 38, no. 7, 9046–9049p,***2011***

[5] **P. Lin, Z. Lin, B. Kuang, P. Huang,(2012)** A Short Chinese Text Incremental Clustering Algorithm Based on Weighted Semantics and Naive Bayes, Journal of Computational Information Systems, 2012, 4257- 4268p,***2012***

[6] **Po-Whei Hung and Chu-Hui Lee (2004),** "Image Database Design Based On 4D-Spa Representation for Spatial Relations. *IEEE Transactions on knowledge and Data Engineering* , ***2004***

[Aberiviations:
ndb:negative border element
fdb; frequent itemset element
DB: Exsisting Warehouse
db: Current Warehouse
updatecount:function used for count increment
n1:updatecount with frequent item sets.]

