

PRIVACY PRESERVING TEMPORAL DATA MINING

Dr K Venkatesh Sharma¹, Behailu Shewandagn², Worku Birhanie³

¹Asst Professor, Jimma University

²Asst Professor, School of Computing, Jimma University

³Asst Professor, School of Computing, Jimma University

ABSTRACT

Due to Rapid development in Data mining, Evolution of Temporal data mining is newly evolving research area in vast field of Data Mining. Temporal Data Mining having applications in following fields such as biomedicine, geographical data processing, financial data forecasting and Internet site usage monitoring. Temporal data mining deals is a process of extracting of knowledgeable information from temporal data, where the definition of Knowledge depends on the user application. The most common form of temporal data is time series data, which consist of real values sampled at regular time intervals. Temporal Data Mining is a rapidly evolving area of research in following field such as statistics, temporal pattern recognition, temporal databases, optimization, and visualization, high performance computing, and parallel computing. This paper is first intended to serve as an overview of the temporal data mining and provide an algorithm to achieve privacy in temporal mining process.

Keywords: Data Mining, Utility, Sensitive data items, Temporal Data mining etc

I. INTRODUCTION

Data mining is a process to extract some knowledge full information contained in large databases. The goal is to discover hidden patterns, unexpected trends or other subtle relationships in the data using a combination of techniques from machine learning, statistics and database technologies. This new discipline today finds application in a wide and diverse range of business, scientific and engineering scenarios. For example, large databases of loan applications are available which record different kinds of personal and financial information about the applicants (along with their repayment histories). Several terabytes of remote-sensing image data are gathered from satellites around the globe.

Due to rapid increase in storage of data, the interest in the discovery of hidden information in databases has exploded in the last decade. This discovery has mainly been focused on association rule mining, data classification and data clustering. One major problem that arises during the mining process is treating data with temporal feature i.e. the attributes related with the temporal information present in the database. This temporal attribute requires a different procedure from other kinds of attributes. However, most of the data mining techniques tend to treat temporal data as an unordered collection of events, ignoring its temporal information.

1.1 Temporal data mining:

Temporal Data Mining (TDM) is defined as the activity of looking for interesting correlations or patterns in large temporal datasets. TDM has evolved from data mining and was highly influenced by the areas of temporal databases and temporal reasoning. Several surveys on temporal knowledge discovery exist [5].

Most temporal data mining techniques convert the temporal data into static representations and exploit existing 'static' machine learning techniques, thus potentially missing some of the temporal semantics. Recently there is a growing interest in the development of temporal data mining techniques in which the temporal dimension is considered more explicitly. Console et al. proposed an extension of the known Decision Trees induction algorithm to the temporal dimension [1]. One advantage of temporal decision trees is that the output of the induction algorithm is a tree that can immediately be used for pattern recognition purposes. However, the method can only be applied to time points, not to time intervals.

TID	A	B	C
T1	0	0	18
T2	0	6	0
T3	2	0	1
T4	1	0	0
T5	0	0	4
T6	1	1	0
T7	0	10	0
T8	3	0	25
T9	1	1	0
T10	0	6	2

Table2.1: External Utility Table

II. RELATED WORK

In association with rules mining, Apriori (Agrawal and Srikant, 1995), DHP (Park et al., 1997) and partition-based ones (Lin and Dunham, 1998; Savasere et al., 1995) were proposed to find frequent itemsets. Many important applications have called for the need of incremental mining due to the increasing use of record-based databases to which data are being added continuously. Many algorithms like FUP (Cheung et al., 1996), FUP2 (Cheung et al., 1997) and UWEP (Ayn et al., 1999; Ayn et al., 1999) have been proposed to find frequent itemsets in incremental databases. The FUP algorithm updates the association rules in a database when new transactions are added to the database. Algorithm FUP is based on the framework of Apriori and is designed to discover the new frequent itemsets iteratively. The idea is to store the counts of all the frequent itemsets found in a previous mining operation. Using these stored counts and examining the newly added transactions, the overall count of these candidate itemsets are then obtained by scanning the original database. An extension to the work in Cheung et al. (1996) was reported in Cheung et al. (1997) where the authors propose an algorithm FUP2 for updating the existing association rules when transactions are added to and deleted from the database. UWEP (Update with Early Pruning) is an efficient incremental algorithm, that counts the original database at most once, and the increment exactly once. In addition, the number of candidates generated and counted is minimized.

In recent years, processing data from data streams becomes a popular topic in data mining. A number of algorithms like Lossy Counting (Manku and Motwani, 2002), FTP-DS (Teng et al., 2003) and RAM-DS (Teng et al., 2004) have been proposed to process data in data streams. Lossy Counting divided incoming stream conceptually into buckets. It uses bucket boundaries and maximal possible error to update or delete the itemsets with frequency for mining frequent itemsets. FTP-DS is a regression-based algorithm for mining frequent temporal patterns from data streams.

ITEM	PROFIT(\$)
A	3
B	10
C	1

Table 2.2 Transaction table

ITEM	Quantity value
A	2
B	6
C	3

Table 2.3 transaction Quantity utility

C.-J. Chu et al. / pattern mining tasks for data streams by exploring both temporal and support count granularities.

Some algorithms like SWF (Lee et al., 2001) and Moment (Chi et al., 2004) were proposed to find frequent itemsets over a stream sliding window. By partitioning a transaction database into several partitions, algorithm SWF employs a filtering threshold in each partition to deal with the candidate itemset generation. The Moment algorithm uses a closed enumeration tree (CET) to maintain a dynamically selected set of itemsets over a sliding window [25].

A formal definition of utility mining and theoretical model was proposed in Yao et al. (2004), namely MEU, where the utility is defined as the combination of utility information in each transaction and additional resources. Since this model cannot rely on downward closure property of Apriori to restrict the number of itemsets to be examined, a heuristic is used to predict whether an itemset should be added to the candidate set. However, the prediction usually overestimates, especially at the beginning stages, where the number of candidates approaches the number of all the combinations of items. The examination of all the combinations is impractical, either in computation cost or in memory space cost, whenever the number of items is large or the utility threshold is low. Although this algorithm is not efficient or scalable, it is by far the best one to solve this specific problem. Another algorithm named Two-Phase was proposed in Liu et al. (2005), which is based on the definition in Yao et al. (2004) and achieves the finding of high utility itemsets. The Two-Phase algorithm is used to prune down the number of candidates and can obtain the complete set of high utility itemsets. In the first phase, a model that applies the “transaction-weighted downward closure property” on the search space is used to expedite the identification of candidates. In the second phase, one extra database scan is performed to identify

the high utility itemsets. However, this algorithm must rescan the whole database when new transactions are added from data streams. It incurs more cost on I/O and CPU time for finding high utility itemsets. Hence, the Two-Phase algorithm is focused on traditional databases and is not suited for mining data streams.

Although there existed numerous studies on high utility itemsets mining and data stream analysis as described above, there is no algorithm proposed for finding temporal high utility itemsets in data streams. This motivates our exploration of the issue of efficiently mining high utility itemsets in temporal databases like data streams in this research.

III. PROPOSED METHOD:

Hiding sensitive data items using temporal data mining proposed new algorithm:

In the proposed method each itemset having two major factors Quantity and profit. Based on these factors calculate total utility of itemset. An itemset is called highly utility itemset if total utility of itemset is greater than user specific threshold (ϵ). To compute the profit utility each itemset belongs to $Ip \in DB$ and

Profit utility = $\sum_{ip \in DB} T(ip, tq) * eu(ip)$ and compute

Quantity utility = $\sum_{ip \in DB} T(ip, tq) * Q(ip)$ and compute total utility = profit utility + Quantity utility

3.1 Privacy preserving Temporal data mining Algorithm (PPTDM): In this algorithm collected original Database DB that's equals sanitized database DB'

Input: collected original database DB

Output: produces sanitized database DB'

Algorithm: for each database DB contains the following data items $DB = \{I1, I2, I3, I4, \dots, In\}$

1. For each data item ip having utility on transaction Tq $T(ip, tq)$.

2. for each itemset ip having external utility eu_{ip}

3. Compute utility factor of each item set ip

$$U(ip, tq) = \sum_{i \in ip} eu_{ip} * T(ip, tq)$$

4. for each item set there associated Quantity of each itemset Q_{ip} .

5. compute Quantity factor of each item such that

$$Q(ip, tq) = T(ip, tq) * Q_{ip}$$

6. Summation of both factors associated with each itemset called total utility of each itemset.

$$Total\ Utility\ (Tu) = \sum_{i \in ip} \{Q(ip, tq) + U(ip, tq)\}$$

7. Assume a user threshold ϵ which specifies itemset is to be sensitive itemset.

8. Compute a difference of total Utility of each itemset and user specify threshold.

$$diff = Total\ utility(TU) - threshold(\epsilon)$$

9. Now modify each itemset such that

$$O(ip, tq) = arg\ max(i \in ip\ T(ip, tq))$$

10. While (diff > 0)

11. Modify each item set such that $O(ip, tq)$

$$= \begin{cases} 0, & \text{if } TU(ip, tq) > diff \\ O(ip, tq) - \lceil \frac{diff}{S(ip)} \rceil, & \text{if } TU(ip, tq) < diff \end{cases}$$

$$12. diff = \begin{cases} 0, & \text{if } TU(ip, tq) > diff \\ diff - TU(ip, tq), & \text{if } TU(ip, tq) < diff \end{cases}$$

13. Return the result sanitized database DB'

$$\text{Total Utility (Tu)} = \sum_{i \in ip} \{Q(ip, tq) + U(ip, tq)\}$$

7. Assume a user threshold £ which specifies itemset is to be sensitive itemset.

8. Compute a difference of total Utility of each itemset and user specify threshold.

$$\text{diff} = \text{Total utility(TU)} - \text{threshold(€)}$$

3.2 Implementation of Algorithm: for implementation of PPTDM algorithm following strategies is estimated

$$\text{Totalutility(TU)A} = [\{Q(A, T3) + Q(A, T4) + Q(A, T6) + Q(A, T8) + Q(A, T9)\} * Q(A) + \{U(A, T3) + U(A, T4) + U(A, T6) + U(A, T8) + U(A, T9)\} * eu(A)]$$

$$= [\{2+1+1+3+1\} * 2 + \{2+1+1+3+1\} * 3]$$

$$= 16 + 24$$

$$= 40$$

$$\text{Totalutility(TU)B} = [\{Q(B, T2) + Q(B, T6) + Q(B, T7) + Q(B, T9) + Q(B, T10)\} * Q(B) + \{U(B, T2) + U(B, T6) + U(B, T7) + U(B, T9) + U(B, T10)\} * eu(B)]$$

$$= [\{6+1+10+1+6\} * 10 + \{6+1+10+1+6\} * 6]$$

$$= 240 + 144 = 384$$

the same way we calculate total utility of each item set {C}, {AB}, {AC} and {BC}

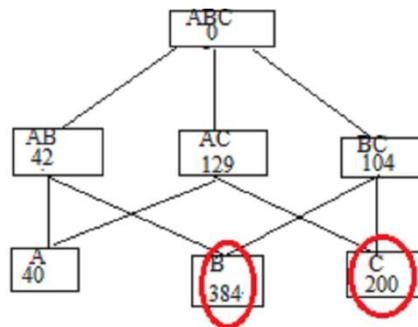


Fig 3.2.1 shows total utility of various data items

Now user specific threshold is 150 so {B} and {c} data items called sensitive data items. Now we reduce total utility factor of itemset {B} and {C}.

High utility item	Total Utility
{B}	384
{c}	200

Table 3.2.1 Sensitive (high utility item set)

ITEM \ Tid	B
T2	6
T6	1
T7	10
T9	1
T10	6

Table 3. 2.2 shows itemset {B} Transaction table

Modify O(ip,tq) such that calculate diff such that

$$= 384 - 150 (\text{user specified threshold})$$

$$= 234$$

Now modify each itemset such that $O(B,T7)$ value 10 to

0 because $TU(B,T7) > \text{diff}$, such that so new specified value of each itemset $\{B\}$ 384 to modified

$$TU(B) = [(6+1+0+1+6) * 10 + (6+1+0+1+6) * 6]$$

$$= 140 + 84 = 224$$

Now this value is greater than user threshold which is 150

ITEM \ Tid	B
T2	6
T6	1
T7	0
T9	1
T10	6

Table 3. 2.3 shows modified itemset $\{B\}$ Transaction table

So now modify next high maximum utility item which is $O(B,T2)$ that is 6, modify it calculate its max average

$$= O(B,T2) * eu(B) * Q(B) = 6 * 10 * 6$$

= 360 which greater then $\text{diff}(224-150)$ so $(B,T2)$ is zero.

So modified table is

ITEM \ Tid	B
T2	0
T6	1
T7	0
T9	1
T10	6

Table 3. 2.4 shows Modified item set $\{B\}$ Transaction table

So newly computed value of $\{B\}$ is

$$TU(B) = [(0+1+0+1+6) * 10 + (0+1+0+1+6) * 6]$$

$$= 80 + 48$$

$$= 128$$

High utility item	Total Utility
$\{B\}$	128
$\{c\}$	200

Table 3. 2.5 modified total utility Sensitive (high utility item set)

Now same way modified value of $\{C\}$ which is 200 to 128.

High utility item	Total Utility
$\{B\}$	128
$\{c\}$	128

Table 3. 2.6modified total utility Sensitive (high utility item set)

IV. SIMULATION ANALYSIS AND RESULT

To measure the effectiveness of PPTDM algorithm, the experiments were conducted on synthetic dataset and compared by the hiding failure. All experiments were performed on a Dell workstation with 3.40 GHz Intel Pentium 4 processor and 2 GB of main memory running the Windows XP professional and simulation performed on WEKA simulation tool and use netbeans7.3 as IDE(Integrated development environment) for deploy of project.

WEKA is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code.

WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes[22].

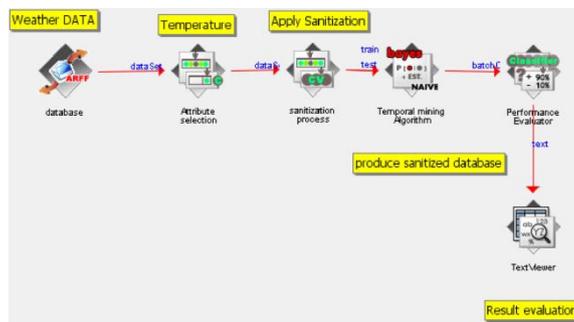


Fig 4.1 knowledge flow of Temporal mining process

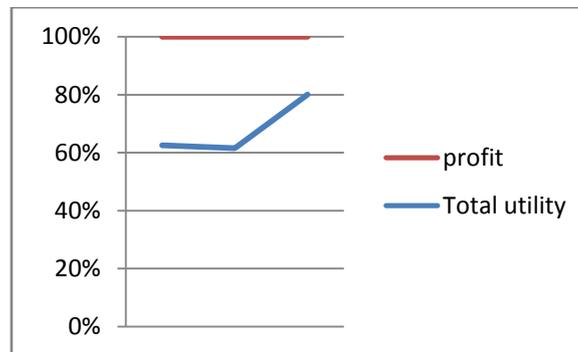


Fig 3.2.2 shows total utility of various data items

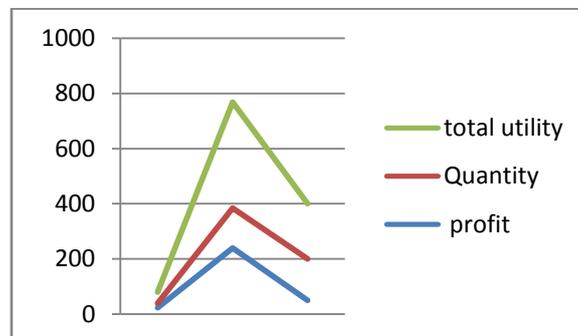


Fig 3.2.3 shows toatal utility of various dataitems

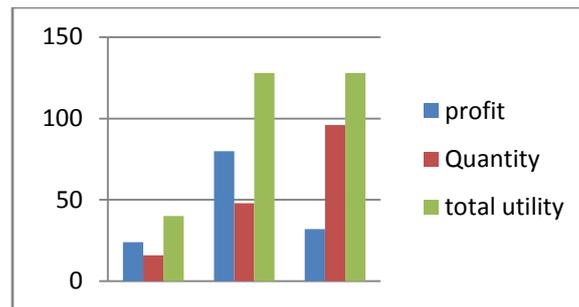


Fig 3.2.4 shows total utility of various data items after applying sanitization

V. EFFECTIVE MEASUREMENT

(a) **Hiding failure (HF)**: The ratio of sensitive item sets that are disclosed before and after the sanitizing process. The hiding failure is calculated as follows:

$$HF = \frac{|U(DB')|}{|U(DB)|}$$

denote the sensitive itemsets discovered from the original database DB and the sanitized database DB' respectively. The cardinality of a set S is denoted as |S|.

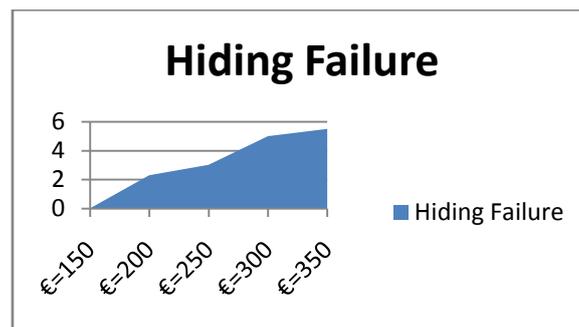


Fig 3.2.4 shows Hiding failure of various data items after applying sanitization

VI. CONCLUSION AND FUTURE WORK:

PPTDM (privacy preserving temporal data mining) algorithm is more sophisticated than traditional based on high utility mining algorithm. Hiding Failure is less than traditional Algorithm like HHUIF (Hiding High utility item set). In the further, develop more superior sanitization algorithms to minimize the impact on the sanitized database in the process of hiding sensitive itemsets. We are also expanding our work with a probabilistic to supplement the empirical, which require further exploration.

REFERENCES

- [1] D. Brillinger, editor. Time Series: Data Analysis and Theory. Holt, Rinehart and Winston, New York, 1975.
- [2] P. Cheeseman and J. Stutz. Bayesian classification (AUTOCLASS): Theory and results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining. AAAI Press / MIT Press, 1995.
- [3] T. Fulton, S. Salzberg, S. Kasif, and D. Waltz. Local induction of decision trees: Towards interactive data mining. In Simoudis et al. [21], page 14.

- [4] B. R. Gaines and P. Compton. Induction of metaknowledge about knowledge discovery. *IEEE Trans. On Knowledge And Data Engineering*, 5:990–992, 1993.
- [5] C. Glymour, D. Madigan, D. Pregibon, and P. Smyth. Statistical inference and data mining. *Communications of the ACM*, 39(11):35–41, Nov. 1996.
- [6] F. H. Grupe and M. M. Owrang. Data-base mining -discovering new knowledge and competitive advantage. *Information Systems Management*, 12:26–31, 1995.
- [7] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: An attribute-oriented approach. In *Proceedings of the 18th VLDB Conference*, pages 547–559, Vancouver, British Columbia, Canada, Aug. 1992.
- [8] J. W. Han, Y. D. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. *Ieee Trans. On Knowledge And Data Engineering*, 5:29– 40, February 1993.
- [9] J. W. Han, Y. Yin, and G. Dong. Efficient mining of partial periodic patterns in time series database. *IEEE Trans. On Knowledge And Data Engineering*, 1998.
- [10] D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors. *Learning bayesian networks: the combination of knowledge and statistical data*. AAAI Press, 1994.
- [11] D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*. AAAI Press, 1997.
- [12] E. Keogh and P. Smyth. A probabilistic approach to fast pattern matching in time series databases. Page 126.
- [13] A. Ketterlin. Clustering sequences of complex objects. In Heckerman et al. [11], page 215.
- [14] C. Li and G. Biswas. Temporal pattern generation using hidden markov model based unsupervised classification. In *Proc. of IDA-99*, pages 245–256, 1999.
- [15] M. J. Zaki. Fast mining of sequential patterns in very large databases. *Uni. of Rochester Technical report*, 1997.
- [16] S. a. O. Etzion, editor. *Temporal databases: Research and Practice*. Springer-Verlag, LNCS 1399, 1998.
- [17] B. Padmanabhan and A. Tuzhilin. Pattern discovery in temporal databases: A temporal logic approach. In Simoudis et al. [21], page 351.
- [18] P. sprites, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. Springer-Verlag, 1993.
- [19] J. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 2002.
- [20] R. O. Duda and P. Hart. *Pattern classification and scene analysis*. John Wiley and Sons, 1973.
- [21] E. Simoudis, J. W. Han, and U. Fayyad, editors. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press, 1996.
- [22] *the WEKA Data Mining Software: An Update* Mark Hall Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer Peter Reutemann, Ian H. Witten Pentaho Corporation Department of Computer Science Suite 340, 5950 Hazelton National Dr. University of Waikato Orlando, FL 32822, USA Hamilton, New Zealand

International Conference on Emerging Trends in Engineering, Science and Management

Sphoorthy Engineering College, Hyderabad, India

(ESM-17)

17th and 18th March 2017 , www.conferenceworld.in

ISBN: 978-93-86171-32-0

- [23] an evaluation of the netbeans module system as a product line implementation technology hans martin mærsk-møller The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Odense, Denmark
- [24] HHUIF and MSICF: Novel algorithms for privacy preserving utility mining Jieh-Shan Yeh, , Po-Chiang Hsu
- [25] Sliding-Window Filtering: An Efficient Algorithm for Incremental Mining Chang-Hung Lee, Cheng-Ru Lin, and Ming-Syan Chen