CUFPR : Closed Uncertain Frequent Pattern Mining in Running Database Based on CUFP-Tree and Ds-Tree.

by
Maliha Momtaz
Exam Roll No: 65
Registration No: RK Ha-2903
Session: 2009-2010
and
Mahbuba Maliha Mourin
Exam Roll No: 69
Registration No: 2010912069

Session: 2010-2011

A thesis presented for the fulfillment of the requirements for the Degree of Bachelor of

Science in Computer Science and Engineering

Department of Computer Science and Engineering
University of Dhaka
Dhaka,Bangladesh

March, 2015

Acknowledgements

Finally we are at the point of accomplishing the goal of our education period of the Bachelor of Science degree through variety of difficulties and a kind help from some respected people to overcome those difficulties. Here we come to express our thankfulness to the people without whom it the thesis would not have been possible, although written words are not enough to convey the utter gratefulness to them .

First of all, we are thankful to Almighty Allah, for his blessings that allowed us to complete this work.

Next we would like to utter our heartiest gratitude and special thanks to our supervisor, Mohammad Samiullah, Lecturer, Department of Computer Science and Engineering ,University of Dhaka ,who gave us opportunity to work with data mining,directed us the right path, instructed us when we faced difficulties, provided great study materials to reach the goal of our research . Actually his guidance, valuable suggestion and criticism and active supervision helped us to complete this research.

We are highly grateful to our parents, family members , senior brothers and sisters and fellow classmates of our department for their constant encouragement and moral support.

Last, but not the least ,we would like to give thanks to the Department of Computer Science and Engineering ,University of Dhaka, for giving us opportunity to perform this research work and facilitating us throughout the whole undergraduate program.

ABSTRACT

Increasing applications of web searching, telecommunication system, retail system ,various sensor networks etc. are continuously producing a huge streams of uncertain data where data are partial , inaccurate and confusing . So mining quality information as well as sidestepping spurious patterns from the large uncertain databases have been a major concern for decision making in our real life . Data mining paves the way to the solution to this problem by introducing mining closed frequent itemsets and mining exact correlation among the patterens. To deal with these situations, A few researchers have been conducted research work on uncertain data stream .Here we have presented a tree-based mining algorithms CUFPR-Tree (based on CUFP[8] and DSTree[19])to mine frequent itemsets using sliding window model, where each item in the transactions in the streams is associated with an existential probability. Key contribution of our thesis is our proposed CUFPR-Tree maintains probability values of each transactions to a two dimensional array, compresses each transaction to CUFPR-Tree in the same manner as a CUFP-Tree (so it is as compact as an CUFP-Tree[8]) and maintains the associated changing probability values of each transaction to the corresponding tail-nodes in an array ; then it mines frequent itemsets from the CUFPR-Tree[8] without further scan of Data Streams. Another contribution is, as CUFP[8] growth algorithm can generate spurious false positive patterns due to its over estimation of the support count of the itemsets ,so we have proposed here to apply an existing correlation measure called h-confidence to remove the false positive patterns drastically. Our experimental results demonstrate that our algorithm has the capability to convey maximal meaningful information without redundancy and with pervasive performance comparison.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Data Mining is a systematic procedure aimed to explore data in search of consistent patterns and/or logical associations between variables, and then to validate the findings. This process is also known as "Big Data" [23, 15]. Data mining can be applicable in medical and chemical research, education, banking, finance, government policy making ,business companies for customer behavior analysis, and most importantly in world wide web. Moreover , due to the huge range of these applications ,data mining has become one of the most important and prevalent field in the study of artificial intelligence(neural network). The ultimate goal of data mining is prediction in direct business applications and that's why many industries now often use data mining technique to make optimal business decision to increase sales and to estimate their future investment. A large number of research have been conducted to generate frequent closed patterns[5, 11, 13, 14], asscociation rule mining[4, 16], sequential patterns identifying [12, 9], time-series data analysis etc. on certain static or uncertain running databases .Among these fields, closed frequent item mining and discovering correlations among the associated patterns are two foremost sections that assist to focus on predictive data mining .

1.1 Overview

Pattern Mining:
In Data Mining the task of frequent-pattern mining in large databases is very important and has been studied extensively in large scale in the past few years. Unfortunately, this task is computationally expensive, especially when a large number of patterns exist.In recent past many algorithms were proposed and implemented to solve this challenge(for example,Apriori[1], FPgrowth[27], CLOSET[13],and CHARM[29]). The FP-Growth Algorithm, proposed by Han in [1], is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree[27]). In this study, Han proved that his method outperforms other popular methods for mining frequent patterns, e.g. the Apriori Algorithm [1] and the TreeProjection.

In recent years, there have been a revolutionary advancement in the realm of data stream mining as well as mining correlation.Different techniques of mining frequent patterns and correlation mining have been proposed. Some of them were designed to mine frequent patterns ,regular patterns , association rule etc. But all of them were on static certain databases or static uncertain databases . Very few researches are done on uncertain growing databases . In this section, we provide background information about mining frequent patterns from static databases of uncertain data and from data certain stream.

Pattern Mining from uncertain data :
Nowadays a wide number of real-life application now introduce uncertain data,where a specific item is either present or absent-the existence of item is not known in transaction.Such an item has a special attribute called *Existential Probability* ranging from 0 to 1 that represents the probability of that item's presence in a particular transaction. Actually processing data from uncertain databases is quite different than that of precise database.

For example ,a satellite image processor identifies the presence of any object in the earth or space .However ,due to the limitation of bandwidth access ,image resolution , noisy electronic media and changing features of spatial area and environment ,the presence of the target object can be expressed as probability.

As this field is getting more and more attention ,researchers have proposed different algorithms to mine frequent patterns from uncertain databases. U-Apriory proposed by Chui.et al[1] was the first algorithm in this field.Later, Leung et al.proposed the UF-growth[22] algorithm[22] To discover frequent patterns, UF-growth generates a UF-tree to capture the contents of uncertain data where each tree node stores an item X and its existential probability P(X, ti), with its frequency count. The UF-tree is constructed in a similar manner to that of the FP-tree[27] in contrast of that nodes in the UF-tree are merged and shared only if they represent the same item X and P(x, ti). Once the UF-tree is constructed, UF-growth extracts appropriate tree paths to mine frequent patterns using the possible world interpretation.In the next ,the modified and efficient version of UF-growth algorithm -Capped Uncertain Frequent Pattern Tree- was proposed called CUFP-growth[8] which introduces a term called *Transaction Cap* that stands for the expected probability count for the whole transaction ,calculated as the product of the largest two probabilities in the transaction and as compact as the Fp-tree.It is to be mentioned that, CUFP-growth[7] discovers frequent patterns from uncertain data, it mines from static databases (instead of dynamic data streams).

Pattern Mining from uncertain data :

Recent emerging applications, such as network traffic analysis, Web click stream mining, power consumption measurement, sensor network data analysis, and dynamic tracing of stock exchange data, call for study of a new kind of data, called stream data, where data takes the form of continuous, potentially infinite data streams, as opposed to finite, statically stored data sets. Stream data management systems and continuous stream query processors are under popular investigation and development. Besides querying data streams, another important task is to mine data streams for interesting patterns.In the busy field of data stream mining DS-Tree[19],CPS-Tree[26],RPS-tree[2] are some of the recent algorithms or

methods which mine useful frequent patterns from streams of data. DSTree[19] uses a Fp-Tree based approach to mine frequent itemsets with sliding window protocol but required higher run-time for constructing tree as it used reconstruction of tree reconstruction each time from the new windows. This algorithm has been outperformed by CPS -Tree by dynamic construction of tree. RPS-Tree mines the frequent patterns almost as the same way in CPS-Tree[26].

Pattern Mining from uncertain stream :
Since the publication of FP-Growth, many well-known algorithms have been developed based on it handle uncertain data stream such as FP-Streaming [22], UF-Growth[22]and SUF-Growth[12], the UDS-FIM[17] algorithm for fast FIM on uncertain data streams. Based on the algorithms FP-Streaming and UF-Growth, Leung et al.[14]proposed two algorithms UF-Streaming[20] and SUF-Growth[9]. UF-Streaming employs the same method as the FP-Streaming: mining frequent itemsets using two minimum support numbers PreMinsupand Minsup. Thus UF-Streaming may lose some frequent itemsets and it also requires an extra data structure UF-Stream[20] to maintain the pre-frequent itemsets of the current window. SUF-Growth[20] is a precise algorithm and uses only a user specified minimum support number Minsup, it directly finds all frequent itemsets with support number that is not less than Minsup from a window, and does not lose any frequent itemsets.

Among the frequent pattern mining algorithms that mine useful knowledge from static databases of uncertain data (e.g., UF-growth algorithm [3],UH-Mine algorithm [3], U-Eclat approach [6], UV-Eclat mining [21]),the tree-based UF-growth algorithm is used in UF-streaming for uncertain data stream mining.In contrast to UFP-growth[3] (which actually does not handle data streams) or FP-streaming [10] (which does not deal with uncertain data), the UFP-streaming algorithm [20]. mines frequent patterns from uncertain data streams by using a fixed-size sliding window of W-consisting of recent batches.In UF-streaming algorithm it first calls CUFP-growth to find frequent patterns from the current batch of transactions in the streams. Frequent Pattern Mining from Time-Fading Streams of Uncertain Data [18]. (uses a predefined minimum support as the threshold value). A pattern is frequent (i.e., subfrequent) if its expected support $>$ threshold value.

CUFP-streaming then stores the mined frequent patterns and their expected support values in a tree structure, in which each tree node X keeps a list of W support values. When a new batch flows in, the window slides and support values shift so that the frequent patterns (and their expected support values) mined from the newest batch are inserted into the window and those representing the oldest batch in the window are deleted. This process is repeated for each batch in the stream.

It is to be noted that,here users are interested in truly frequent patterns (i.e., patterns with expected support $\geq$ threshold value>predefined minimum support),but data in the continuous streams are not necessarily uniformly distributed, so predefined minimum support which has been used in attempt to avoid pruning a pattern too early ,is not feasible technique to pruning redundant patterns in data stream.

But there are some major drawbacks of using CUFP in uncertain data stream . CUFP-Growth[8] algorithm uses the maximal and second maximal probabilities of each transaction to form the transaction cap. Thus, an item which is much lower probability can

get a higher transaction cap and causes over estimation .For that stated reason, the expected count of some itemsets are assigned higher than they actually are.This issue leads to generate false positive patterns.

Rule Mining among Frequent Pattern

There are bundle of algorithms used to mine frequent itemsets. Some of them, very well known, started a whole new era in data mining. They made the concept of mining frequent itemsets and association rules possible. Others are variations that bring improvements mainly in terms of processing time.The algorithms vary mainly in how the candidate itemsets are generated and how the supports for the candidate itemsets are counted. To formalize the problem of mining highly correlated patterns in data sets with random support distribution *h-confidence*[28] measure is used. Actually h-confidence it is a measure that reflects the overall affinity among items within the itemset and this property can be used to avoid generating spurious patterns involving items from different support levels.

1.2 Motivation

Uncertain data stream is a challenging field. Because data streams are continuous and infinite. To find frequent itemsets from streams, we do not have the chance to perform multiple data scans as once the streams flow through, we lose them. Hence, we need some techniques to capture the important contents of the streams. Again in the streams data are not evenly scattered,usually their distributions are changeable in nature with time which indicates-a currently infrequent pattern may be frequent in the future or vice versa. So we need exact algorithm to handle such challenges and mine frequent patterns .But frequent pattern mining algorithm sometimes generate false positive patterns -patterns that may appear to be interesting even though they are actually more likely to be noises . So we need an efficient algorithm that satisfies the goal of any frequent pattern mining algorithm which is actually to find interesting association rules by removing all undesirable false positive patterns .Tree based structure is a simple but yet powerful technique to handle this issue.But the methods proposed so far is either not as compressed as CUFP [8] algorithm or generates redundant patterns due to the limitation of CUFP algorithm . This drawback motivated us to our research work to propose a better algorithm to overcome the defficiency of CUFP[8] in running databases.

1.3 Objective

To reinforce the very few previous research work on uncertain data stream we conducted our research work. Our objectives were:

- to propose a simple but effective tree based structure in mining uncertain data stream.
- to propose a way mine frequent itemsets from the tree.

- to propose a measure which will mine closely associated patterns as well as reduce the redundant patterns.

We have tried to improve an algorithm that unifies the solution to the mining problems of the uncertain data stream to some extent.

1.4 Contribution

In this paper, a simple , well organized but effective algorithm named **CUFPR** [Closed Uncertain Frequent Pattern Mining in Running Database] is established to mine frequent patterns and a way to use an existing correlation measure(h-confidence) find the actual closely associated patterns from the continuous flow of uncertain data.

It introduces a new tree based technique to mine form uncertain data stream easily which was not proposed before.Our proposed CUFPR-Tree is as compact as an CUFP-Tree[8],as it handles probability value of each item in every transactions to an array, compresses each transaction to CUFPR-Tree in the same manner as a CUFP-Tree and maintains the associated changing probability values of each transaction to the corresponding leaf-nodes in an array and in a systematic manner ; then it mines frequent itemsets from the CUFPR-Tree without rescan of Data Streams as the same way as fp-growth algorithm.

Again here we have proposed an way how to apply an existing correlation measure called h-confidence to remove the false positive(which sometimes are produced in CUFP-growth[8]).

1.5 Thesis organization

The remaining part of this paper is organized as follows :

Chapter 2. This chapter exhibits our literature review, deep background study and preliminaries of related work.It is the next chapter.

Chapter 3. It describes and analyses our proposed algorithm, "CUFPR" with suitable examples and with required pseudo code necessary simulation .

Chapter 4. Here the experimental result and performance comparison on different dataset of our proposed algorithm is presented .

Chapter 5. Finally, we discuss about some limitations and future possibilities of this research work and conclude with this chapter.

Chapter 2

Background Study

In this chapter, we overview some background researches for mining frequent patterns from uncertain static database,from certain data stream and also focus on association rule mining from databases which motivated us to construct a new approach "CUFPR" for uncertain stream mining in an efficient way.

2.1 Literature Survey

For mining closed frequent patterns from uncertain database we have revisited different algorithm among which we chose CUFP-Mine[8] , for mining frequent patterns from data stream we selected the DS-Tree[19] and for mining strong affinity among patterns we preferred the Affnity[28] algorithm which we have used to improve the algorithm for mining frequent patterns from uncertain data stream.

2.1.1 Mining Frequent Itemsets from Static Databases of Uncertain Data: CUFP-Algorithm

The CUFP-mine algorithm is pro-posed based on the tree structure to find uncertain frequent patterns. The tree-construction algorithm holds that if a transaction includes the same items as those in a branch of the tree, the items will be merged into the branch although it has a different existential probability. The expected count (instead of only the count of frequency) of an itemset is desired. Thus, each node at the end of a path of the tree has to store the expected count of the item in it as well as those of its super-itemsets existing in the path. An array is then attached to a node to keep the values. The construction algorithm and an example are then stated below-

- From a database consisting of n uncertain transactions, and a predefined minimum

support threshold s,the algorithm first calculates the expected count *expCount(I)* and the transaction frequency *f(I)* of each item I in the uncertain database,where the transaction frequency is the number of transactions in which I appears with non-zero probabilities.

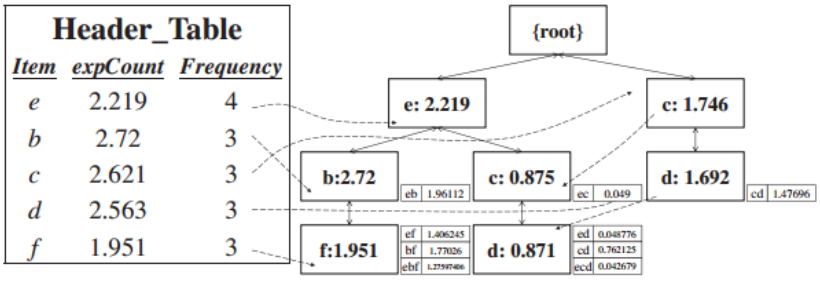
- Then it checks whether the value *expCount(I)* of each item I is larger than or equal to the predefined minimum count, n∗s.
- If *expCountI* satisfies the above condition,I is an uncertain frequent item and puts I with its *expCountI* in the set of large $frequent$ 1-itemsets (L1).
  That is: L1={I:expCount$I$ | {expCount$I$}≥ n ∗s}
- next it sorts the items in L1 according to their transaction frequencies.
- then it builds the *HeaderTable* by keeping the 1-itemsets (items) in L1 in the sorted order.
- it initially set the root node of the CUFP tree as root.
- then the tree removes the items not in L1 from the transactions and sort the remaining items in each transaction according to the sorted order.

The construction of CUFP tree is shown below by the following substeps.

Figure 2.1: A transactional database.

Transaction ID	Original items	Updated items
1	*a*: 0.8, *b*: 0.92, *e*: 0.721, *f*: 0.718	*e*: 0.721, *b*: 0.92, *f*: 0.718
2	*b*: 0.9, *e*: 0.719, *f*: 0.723	*e*: 0.719, *b*: 0.9, *f*: 0.723
3	*c*: 0.87, *d*: 0.872	*c*: 0.87, *d*: 0.872
4	*b*: 0.9, *e*: 0.723, *f*: 0.51	*e*: 0.723, *b*: 0.9, *f*: 0.51
5	*c*: 0.875, *d*: 0.871, *e*: 0.056	*e*: 0.056, *c*: 0.875, *d*: 0.871
6	*a*: 0.012, *c*: 0.876, *d*: 0.82	*c*: 0.876, *d*: 0.82

Figure 2.2: CUFP-Tree build from the database.



2.1.2 Mining from Uncertain Data Streams with Sliding Windows

The DSTree is designed for (exact) stream mining. The construction of the DSTree only requires one scan of the streaming data (as we do not have the luxury to scan the stream-

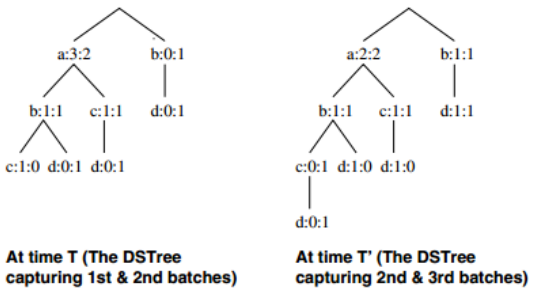
ing data multiple times). The tree captures the contents of transactions in each batch of streaming data (in the current window.

- from a data stream ,DSTree algorithm captures each transaction to form batches.
- from batches it forms a window.
- then it contructs the DSTree where each item has a procedure to store its frequency count and associated batch number.
- when window slides or new window comes the tree is reconstructed and older information is updated. the illustration is shown below...

Figure 2.3: A transactional database.

Batch	Transactions	Contents
first	t_1	$\{a, b, c\}$
	t_2	$\{a\}$
	t_3	$\{a, c\}$
second	t_4	$\{a, c, d\}$
	t_5	$\{b, d\}$
	t_6	$\{a, b, d\}$
third	t_7	$\{b, d\}$
	t_8	$\{a, b, c, d\}$
	t_9	$\{a, c\}$

Figure 2.4: DS-Tree for the data first stream.



2.1.3 Association Rule Mining from databases

The concept of a hyperclique pattern was introduces toestablish a novel tree structure that can strike to a balance between to ability to identify hiighly associated patterns at low support levels,and the ability to remove the spuriouss pattern involving items from different support levels.Basically hyperclique patterns are those patterns which supports the h-confidence measure.

- h-confidence is a interesting measure to evaluate the overall affinity within a hyperclique pattern.
- hyperclique patterns maintains three properties : anti monotone property, cross support property and strong affinity property.
- if the itemset support all these properties they are called hyperclique patterns.

2.1.4 preliminaries

1. Definition.***"Itemset"***: *An itemset* $\boldsymbol{X = \{x_1, x_2, x_3,..., x_n\}}$ *is a set of one or more items (* x_1, x_2, x_3,....,x_n *)where*$X \subseteq I$. *Here,I is the set of all possible items in the database. An item-set with k items is called a k-itemset.*
for example ,{twitter, likedin,gmail} is a 3-itemset.

2. Definition.***"Frequent Itemset"***:*An itemset is said to be frequent if all the items in the set is available in the transaction database D, at least more than or equal to the minimum support threshold,* σ. for instance,an itemset I=$\{I_1,I_2,I_3,...,I_n\}$is a frequent if frequency$I \geq \sigma$, Minimum support threshold. If a database holds a minimum support threshold of 60% and number of transactions in a database is 400,any itemset$\{i_1,i_2,...,i_n\}$ will be frequent if it satisfies the minimum support threshold *i.e.,* the frequency of the itemset must be at least 240 out of the total 400 transactions.

3. Definition.*"* ***Association rule"***:*Let* $\boldsymbol{X = \{x_1, x_2, x_3,...,x_n\}}$ *and* $\boldsymbol{Y = \{y_1, y_2, y_3,...,y_n\}}$ *be two sets where* $X,Y \subseteq I$ *and* $X,Y \cap = \phi$. *Here* $I=\{I_1,I_2,...,I_n\}$ *be a finite set of n items. Thus an association rule is defined as representation of the form* $X \rightarrow Y$. *The set of items X is here called the antecedent and the set of items Y is called the consequent.*

4. Definition.***"Minimum Support and Confidence threshold"***: *The support confidence for patterns in uncertain databases, uConf is defined as* $uConf(X)=\frac{expSup(X)}{max_{x \in X} expSup(X)}$

5. Definition.***"Expected support"***:*The expected support count (expSup) of an itemset* $\boldsymbol{X}$ *in an uncertain transaction dataset is the sum of its probability values in all transaction itemsets containing X, denoted as expSup(X),and is defined by* $expSupX= \sum P(X,t)$.

6. Definition.***"Uncertain Data Stream"***:*In data mining uncertain data stream is an instance of dynamic and continuous data where occurrance of each data is expressed by it existential probability value and the data can be read only a limited number of times using constrained resources to mine useful information. An uncertain dynamic database is rapidly growing database that continues to increase with time.*
A continuous sequence of uncertain data elements with their existential probability generated from a specified source.
i.e., network traffic analysis, web clicking stream, telecommunication call detail record, network intrusion detection etc.

7. Definition. ***"Mining frequent patterns from U-Database"*** *The mining technique from uncertain data stream is based on windows and batches. A batch is a set of transactions and a window-sliding unit. A window is a set of batches. In a single run of the algorithm all the transaction in a window is computed. After the computation is done , the window slides to the next batch. This means , the topmost batch is taken out of account and a new batch is taken into the window.*

8. Definition. ***"Mining strong affinity patterns from Uncertain Databases"***:*Given an uncertan database UDB=*$\{ T_1,T_2,....,T_N\}$*, with N transactions where minimum correlation or expected support confidence threshold is* σ*,The problem is to mine correlated itemsets* $CI \subseteq$ *UDB, where Confidence of the database ,*$uConf \geq \sigma$ *and* $CI_i \subseteq CI$.

9. Definition. ***"h-confidence measure"***:*We illustrate three important properties of the h-confidence measure in this subsection.*

Property 1. ***"Anti monotone property"***: *The h-confidence measure is mathematically equivalent to the all-confidence measure proposed by Omiecinski[24], even though both measures are developed from different perspectives. If P=* $\{i_1,i_2,i_3,..., i_n\}$ *is an itemset, then the h-confidence of P is calculated. by*
$hConf(P)=\frac{supp\{i_1,i_2,i_3,...,i_n\}}{max_{1\leq k\leq m} supp\{i_k\}}$ *Since h-confidence is mathematically identical to all-confidence, it is also monotonically non-increasing as the size of the hyperclique pattern increases. This anti-monotone property allows us to push the h-confidence constraint into the search algorithm. Thus, when searching for hyperclique patterns, the support of a candidate pattern is counted only if all its subsets of size m-1 are hyperclique patterns.*

Property 2. ***"Cross support property"***:*In this section, we introduce the concept of cross-support property. This property is useful to avoid generating cross-support patterns, which are patterns containing items from substantially different support levels. If I=* $\{i_1,$ $i_2,$ $i_3,...,$ $i_n\}$ *is an ordered set of items, sorted according to their support values, i.e.,* $\forall k : supp(i_k) < supp(i_k + 1)$. *In addition for each item* $x \in I$, *Let* $L(x)=x'|supp(x') \leq supp(x)$ *and* $U(x)=x'|supp(x') \geq supp(x)$.
A function f satisfies the cross-support property if f implies where is an itemset containing at least one item from and at least one item from $\exists x, y \in I$ *such that supp(x)* $\leq$ *supp(y) and upper f(x, y)<t.*
where P is an itemset containing at least one item from Lx and one item from Uy and t is the specified threshold.The h-confidence measure satisfies the cross support property. Furthermore, the h-confidence value for any cross-support pattern P= $\{x_{i1},$ $x_{i2},$ $x_{i3},...,x_{ik},y_{j1},$ $y_{j2},$ $y_{j3},...,y_jl\}$*has an upper bound and it is calculated as*
$upper(hCon(f))=\frac{max_{1\leq p\leq m} suppx_p}{min_{1\leq q\leq n} suppy_q}$

Property 3. ***"Strong affinity property"***:*In this subsection, we investigate the relationships between h-confidence and other similarity measures such as cosine[25] and Jaccard[25] measures. Our goal is to derive the lower bounds for these similarity measures in terms of the h-confidence threshold* h_c*Given a pair of items P=* $\{i_1,$ $i_2\}$, *the co-*

sine of P is computed as $cosine(P)=\frac{suppi_1,i_2}{\sqrt{supp(i_1)*supp(i_2)}}$,*while the Jaccard measure for P is* $jaccard(P)=\frac{suppi_1,i_2}{supp(i_1)+supp(i_2)-supp(i_1,i_2)}$

Chapter 3

Our Proposed Algorithm

CUFPR: Closed Uncertain Frequent Pattern Mining in Running Database based on CUFP-Tree and Ds-Tree.

3.1 Overview

In this section ,we have introduced our proposed algorithm for mining closed frequent pattern mining from *Uncertain Data Streams*. We have used here sliding window technique to get the most updated input . At first the algorithm takes the uncertain data from Running Database, forms batches to form window. Then it forms a tree to mine closed frequent patterns using CUFP approach. In the next step the algorithm generates association rule for discovering strong affinity patterns and removes spurious false positive patterns generated from the tree.

3.2 Problem Definition

In this section we visit the problem definition and our strategies to achieve the goal of our algorithm .We are given with a transaction database where each item holds its probabilistic value to indicate its chance to exist in the transaction . The transaction is always updated with new transactions.The window size, batch size ,minimum support ,minsup and h-confidence thresholds ,uh_c are set.

3.3 Building Blocks of our Proposed Approach

While working with correlated patterns (both support and weighted confidence), we have come to some valuable findings regarding redefinition of correlation measures for uncertain stream databases. These findings can be described with the following observations.

3.3.1 Building blocks

10. Definition. ***"h-confidence measure for uncertain data stream"***: *We rename the h-confidence measure for uncertain data stream as* uh_c.

Property 4. ***"Anti monotone property for*** uh_c***"***: The uh_c is similar to h-confidence measure for certain database and mathematically equivalent to the all-confidence measure proposed by Omiecinski[24].
If P= $\{i_1, i_2, i_3,..., i_n\}$ is an itemset, then the $uh_c$of P is calculated by. $uh_c(P)=\frac{expSup(\{i_1,i_2,i_3,...,i_n\})}{max_{1\leq k\leq n} expSupp\{i_k\}}$.

Property 5. ***"Cross support property for*** uh_c***"***:*In this section, we introduce the concept of cross-support property. This property is useful to avoid generating cross-support patterns, which are patterns containing items from substantially different support levels. If I=* $\{i_1, i_2, i_3,..., i_n\}$ *is an ordered set of items, sorted according to their support values, i.e.,* $\forall k : expSup(i_k) < expSup(i_k + 1)$. *In addition for each item* $x \in I$, *Let* $L(x)=\{x'|expSup(x') \leq expSup(x)\}$ *and* $U(x)=\{x'|expSup(x') \geq expSup(x)\}$.
A function f satisfies the cross-support property if f implies where is an itemset containing at least one item from and at least one item from $\exists x, y \in I$ *such that expSup(x)* $\leq$ *expSup(y) and upper f(x, y)<t.*
where P is an itemset containing at least one item from Lx and one item from Uy and t is the specified threshold.The h-confidence measure satisfies the cross support property. Furthermore, the h-confidence value for any cross-support pattern P= $\{x_{i1}, x_{i2}, x_{i3},...,x_{ik},y_{j1}, y_{j2}, y_{j3},...,y_{jl}\}$*has an upper bound and it is calculated as*
$upper(uh_c(f))=\frac{max_{1\leq p\leq m} expSupx_p}{min_{1\leq q\leq n} expSupy_q}$

Property 6. ***"Strong affinity property for*** uh_c***"***:*In this subsection, we investigate the relationships between h-confidence and other similarity measures such as cosine[25] and Jaccard[25] measures. Our goal is to derive the lower bounds for these similarity measures in terms of the h-confidence threshold* uh_c*Given a pair of items P=* $\{i_1, i_2\}$, *the cosine of P is computed as .* $cosine(P)=\frac{expSupi_1,i_2}{\sqrt{expSup(i_1)*expSup(i_2)}}$,
while the Jaccard measure for P is . $jaccard(P)=\frac{expSupi_1,i_2}{expSup(i_1)+expSup(i_2)-expSup(i_1,i_2)}$

3.4 Our Proposal:Our proposed approach -CUFPR

In order to propose an algorithm that uses *CUFP Tree* structures, we need to prove that, In our prposed algorithm we have changed the definition of h-confidence as uh_c.

3.5 Steps of the Algorithm

Our proposed algorithm is divided into three major phases :

1. Construction of the CUFPR Trees: The phase scans the uncertain data streams and build the CUFPR-Tree by inserting the transactions of current window and next updates the tree by inserting new transactions when window slides.

2. Frequent Itemset Mining :This phase mines the frequent itemsets from the tree by using a recursive method.

3. Removal of spurious false positive patterns :This phase takes the frequent itemset discovered by step two and checks for strong affinity conditions to remove the spurious patterns.
In the following subsections, the stages of the algorithm are explained briefly.

3.5.1 Components of the Algorithm

The major components of the algorithm are listed below:

1. **Header Table:** A Header table is used to keep track of the current status of the *CUFPR-Tree*. Header table contains the information regarding each item and its expected support ,expSup$I$. At each stage , we can iscard infrequent items just by looking at the header table . Header table also maintains a linked list among items. This allows us to find any item in the tree quckly.

2. **Tree Data Structure:** The tree data structure is the major part of the algorithm . It is a rooted tree and initially , only the root exists with NULL value. The tree is built progressively by adding transactions of the current window to it and update the branches of the tree when new window comes. There are two types f pointers for each tree node. One of them helps to create the tree structure, the other pointer is used for the linked list formed from the header table.

3. **Hashtable of Items**:An additional data structure for each item is used to store batch no and corresponding expSup so that by doing a query on transaction ,item, we can

get the existential probability of that item in constant time.

4. **Itemsets Holder:** This linear data structure *implementedasanarray* holds the frequent itemsets mined from the algorithm. This can be used later for association rule mining , or can be written back to database.

3.5.2 Construction of the CUFPR-Tree

Algorithm 3.1: CUFPR-tree construction.

```
1  begin
2  Scan the transaction database UDB ,form batches b_j for each window w_i
3  Sort F in support-descending order as FList, the list of frequent items
4  Create the root of an FP-tree, T, and set its label as null
5  for each w_i in UDB do
6      for each b_j in UDB do
7          calculate the expected support(expSup) of each item
8          if expSup(x_k < min_sup then
9              remove the item
10         else
11             keep the item in the FI-List with their transaction id t_p
12         Sort each item in t_p of FI-List according to their expSup-descending
           order
13     create the root node R=NULL of a CUFPR-Tree; for each b_j in each
       w_i do
14         for each t_p in each b_j do
15             Call Insert-CUFPR-Tree(R,t_p,b_j);
16 Return the CUFPR-tree.
17 end.
```

Algorithm 3.2: Procedure Insert-CUFPR-Tree(R,t_p,b_j)

```
1  :
2  begin.
3  let the sorted items in t_p be [x|X] ,where x is the first item and X is the
   remaining items in the FI-list;
4  if a child C of R such that C.item=x.item then
5      select C as the current node;
6      C.expSup=C.expSup+expSup;
7  else
8      create new nodw C as a child of R;
9      C.expSup=C.expSup+expSup;
10 else
11     X is not empty then call Insert-CUFPR-Tree(R,t_p,b_j);
12 end.
```

Algorithm 3.3: Procedure Frequent-Itemset-Mining

```
1  :
2  begin
3  for each item α ∈ FI − List do
4      PB_α =Build-PB(CUFPR-tree),call Mine(PB_α,α )
5  function Build-PB(CUFPR-tree,a)
6  for each node N_α of the last item of α in CUFPR-tree do
7      Project path P_α from the parent of N_α up to the root with N_(α).
       expSup* P_α for each node in the conditional pattern base PB_α of α ;
       HereP_α returns the existential probability of α for this branch.
8  Let FI-Lista be the FI-list for PB_α;
9  call Mining_correlated_Patterns(PB_α); return PB_α; function Mine(PB_α,α)
10 CT_α =Build-CT(PB_α);
11 if CT_α ! =NULL then
12     for each item βinFI − List of CT_α do
13         generate β = βunionα as a candidate frequent itemset.
14         PB_α = Build PB(CT_α,β); call Mine(PB_α,β);
15 function Build-CT(PB_α)
16 for each item in βinFI − Lista do
17     if expSup(β¡min_sup) then
18         delete β from FI-Lista ;
19         delete all N_β nodes form PB_α.
20 return CT_α(which is the conditional tree constructed from PB_α);
21 function Mining_correlated_Patterns(PB_α)
22 for each item in PB_α of FI -List do
23     check-antimonotone-property(PB_α)
24     check-cross-support-property(PB_α)
25     check-strong-affinity-property(PB_α)
26 return itemsets;
27 end
```

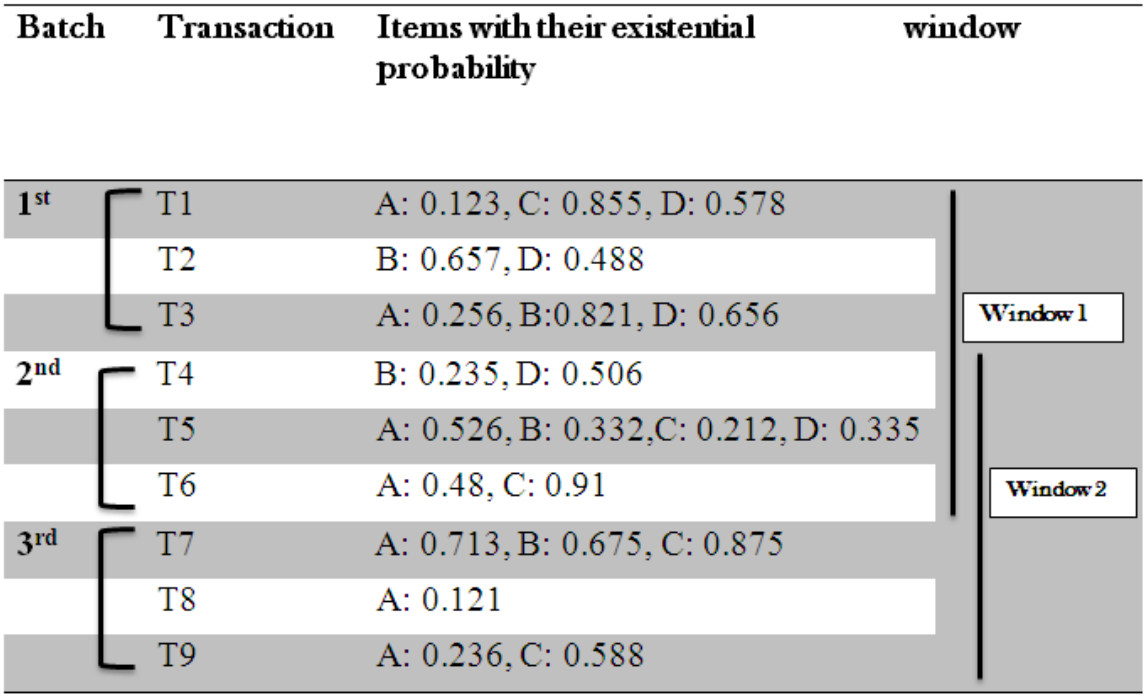


Figure 3.1: Uncertain stream database sample

3.5.3 Illustration

Here is shown the implementation of our proposed CUFPR-Tree algorithm on an uncertain transactional database.The database is shown in Figure 3.1:

The CUFPR-tree is constructed on the data stream for first window and first two batches applying our algorithm (shown in table Figure 3.2:).

After construction of the tree, we get the frequent itemsets (itemset including false positive patterns) are here shown in Table 3.2: Then we apply our rest part of the algorithm to find strong affinity patterns to remove the spurious false positive patterns. Here we see itemset {a,d}and {a,c,d} have been pruned as the have been proved to be false positive patterns,shown in figure 3.3: Here is shown the filtering comparison of mined frequent itemset and closely associated patterns.

3.5.4 Summary

Through our research, we have been able to find a way to fine closed frequent itemsets from uncertain data stream and reducing false positive patterns based on the theme of CUFP-Tree[8] , DsTree[19] and h-confidence[28] .Our primary goal was to construct an effective tree which can capture the transactions of uncertain stream and then improve the mining approach to reduce the spurious patterns by mining correlation.

Figure 3.2: CUFPR-Tree constructed from the given sample stream

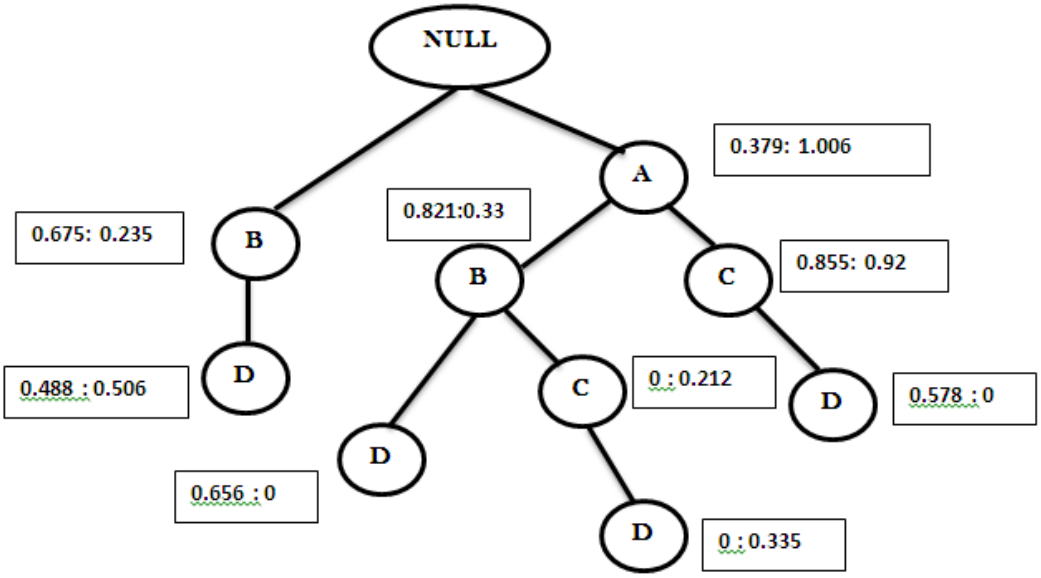


Table 3.1: Filtering of false positive patterns

Table 3.2: Frequent pattern

A	1.385
B	2.316
C	1.977
D	2.563
A,B	1.597
A,D	3.769
A,C	2.738
C,D	1.73
B,D	2.067
A,C,D	2.396

Table 3.3: Strong Affinity pattern

A	1.385
B	2.316
C	1.977
D	2.563
A,B	1.597
A,C	2.738
C,D	1.73
B,D	2.067

In the next chapter, we have presented our experimental results with speak of our claim and assumption made in this chapter.

3.5.5 Application of our Algorithm:

In this section we will discuss about the real life applicability and use of our proposed algorithm . Actually in our everyday life most of the databases we deal with is uncertain . For example we can say the modern applications of sensor data, web application data, geospatial data, medical and bio-informatics data, market basket data and so on, everywhere data are uncertain. Real life applicability of mining frequent itemsets and mining correlation from uncertain data stream can be realized by some use cases. In medical diagnosis system for automated disease prediction, databases of medical treatment diagnosis contain uncertain data(70% probality of heart disease)having tuples like {high blood pressure : 0.7, chest pain:

0.1, genetic cause : 0.6 ,High density cholesterol : 0.8, high pulpetation of heart :0.9.} In an automated disease prediction from the heart disease of symptoms and findings, we can and frequent itemset f of heart disease, genetic cause that states heart disease due to genetic causes are frequent cases. Moreover, in such systems, the correlated patterns can detect the correlated affinity patterns can provide information of correlated disease those have severe effects on others.

In web click streams , for user behavior analysis mining frequent and correlated patterns can help to suggest a particular web item for that user. For instance a user browse you tube .His watching list can be {music video: 0.9, tutorial : 0.8, movie song :0.7, drama : 0.6}. A frequent itemset mined { music video, movie song} indicate that we can suggest more latest correlated video songs to that user rather suggesting him learning tutorials.

Again in geospatial data analysis for sending any missiles or searching for moving objects or for predicting path of cyclones use image data and produces data that is uncertain. So, efficient uncertain data mining techniques are required to give them the desired outcomes. Hence, we need uncertain data mining techniques to find highly correlated patterns and in this case our algorithm can help us a lot .

Chapter 4

Experimental Result

In this chapter, we have presented the performance evaluation of our proposed algorithm. At first we have tested our algorithm using four different datasets. We have taken four different parameters and derived the curves for each parameter of each dataset. Finally we have derived the curve of filtering percentage of each dataset. All the curves of same parameter represent a similar nature. The curves for required time and memory against different number of transactions go upward. Because more time and memory is needed for increasing number of transactions. On the other hand, the curves for required time and memory against different values of h_confidence threshold go downward. Because less time and memory is needed for increasing value of h_confidence threshold.

4.1 Experimental Environment :

We have done our experimental evaluation in the following environment :

- **Hardware specification:**
  - CPU : Intel Core i3 2.40 GHz
  - RAM : 4 GB.
- **Software specification:**
  - Operating System : Windows 7 (x86-32 : 32 bit environment)
  - Source Language : C++ for algorithm implementation, data synthesis and performance benchmarking.

– Drawing Tools : Microsoft Excel

– Latex Environment : MikTex

Changes in hardware and software specification may produce results of different magnitude.

4.2 Data Synthesis

In order to test our algorithm, we needed to use large datasets from well known data repository. Unfortunately, large data repository for uncertain data is not available online. For this reason, we have generated uncertain dataset from the certain dataset. We have generated random character from each item of a transaction and derived the probability of the items.

Database	Transaction	Item
Chess	3545	75
Mushroom	30032	114
Connect	35670	3210
Mushroom	40034	11267

Parameter	Curve
Database Size vs Time	Figure 4.a.1, Figure 4.a.2, Figure 4.a.3, Figure 4.a.4
Database Size vs Memory	Figure 4.b.1, Figure 4.b.2, Figure 4.b.3, Figure 4.b.4
Threshold (h_c) vs Time	Figure 4.c.1, Figure 4.c.2, Figure 4.c.3, Figure 4.c.4
Threshold (h_c) vs Memory	Figure 4.d.1, Figure 4.d.2, Figure 4.d.3, Figure 4.d.4
Filtering percentage	Figure 4.e.1

Among the four datasets chess, mushroom and connect is widely used as dense dataset. On the other hand T10 is widely used as sparse dataset.We have done our evaluation on each datasets using four different parameters :

Here, we have shown our evaluation for four comparison measurement scales :

- Size vs Time
- Size vs Memory
- Threshold vs Time
- Threshold vs Memory

4.3 Experimental evaluation :

4.3.1 Size vs Time curve of chess, mushroom, connect and T10 datasets :

At first, we have taken size vs time parameter and applied it on the four datasets. The minimun support confidence (minsup_conf) threshold and h-confidence (h_conf) threshold is constant in this case.

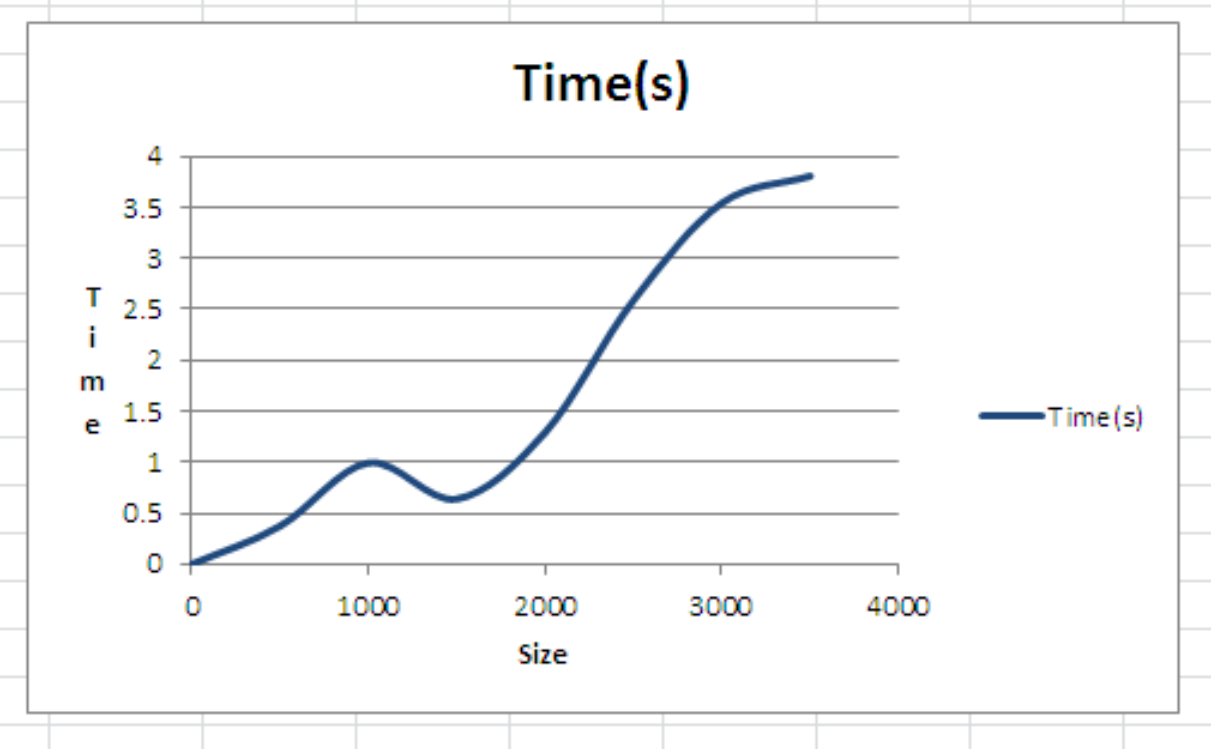


Figure 4.1: a) Size vs Time curve for chess dataset

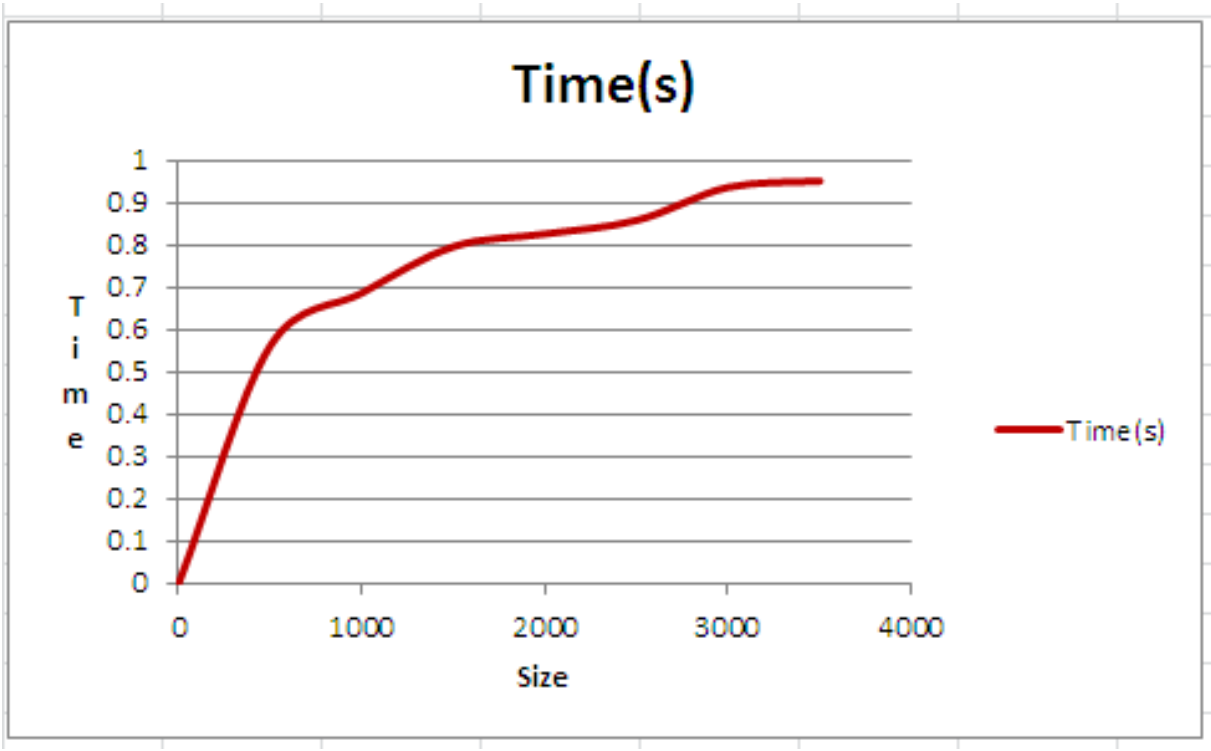


Figure 4.2: a) Size vs Time curve for chess dataset

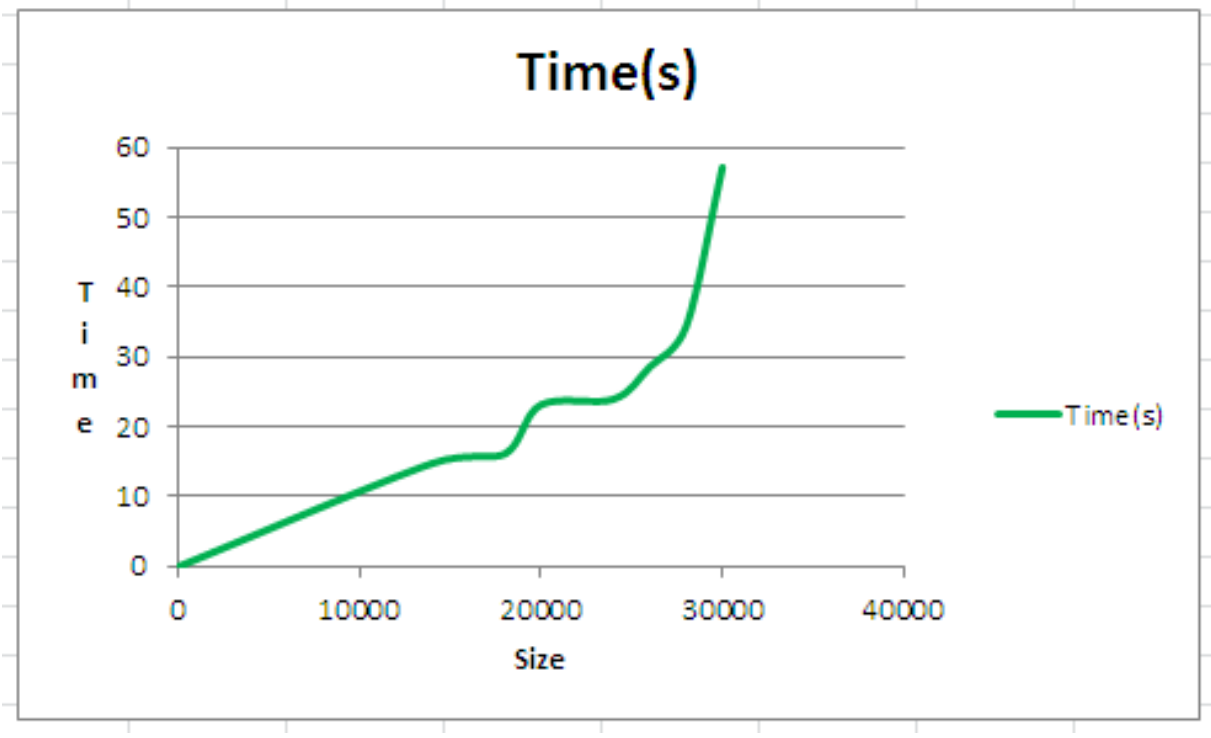


Figure 4.3: a) Size vs Time curve for chess dataset

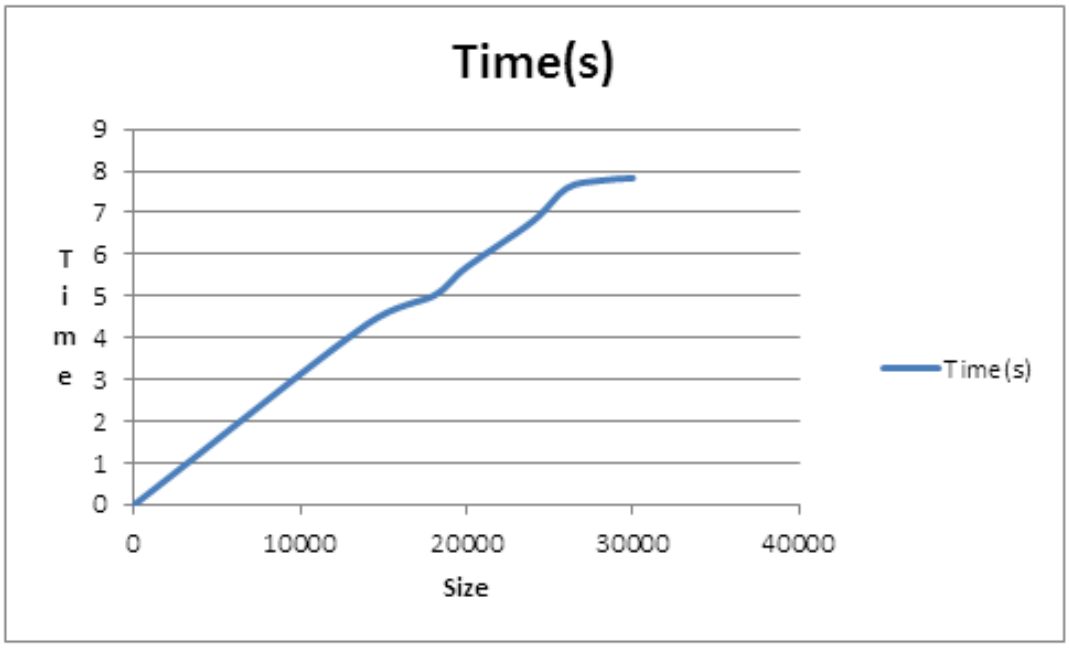


Figure 4.4: a) Size vs Time curve for chess dataset

In the above graphs from Fig 4.1 to Fig 4.4, x-axis represents the transaction size and y-axis represents the time needed for particular transaction size. Here we notice that the amount of time increases with the size of transactions in all four datasets. Because increasing number of transactions produce increasing number of frequent itemsets and increasing number of strong affinity patterns.

4.3.2 Size vs Memory curve for chess, mushroom, connect and T10 datasets:

In the graphs Fig 4.5 to Fig 4.8, x-axis represents the transaction size and y-axis represents the memory needed for particular transaction size. Here, we notice that the amount of memory increases with the size of transactions. Because increasing number of transactions produces increasing number of frequent itemsets and increasing number of strong affinity patterns.

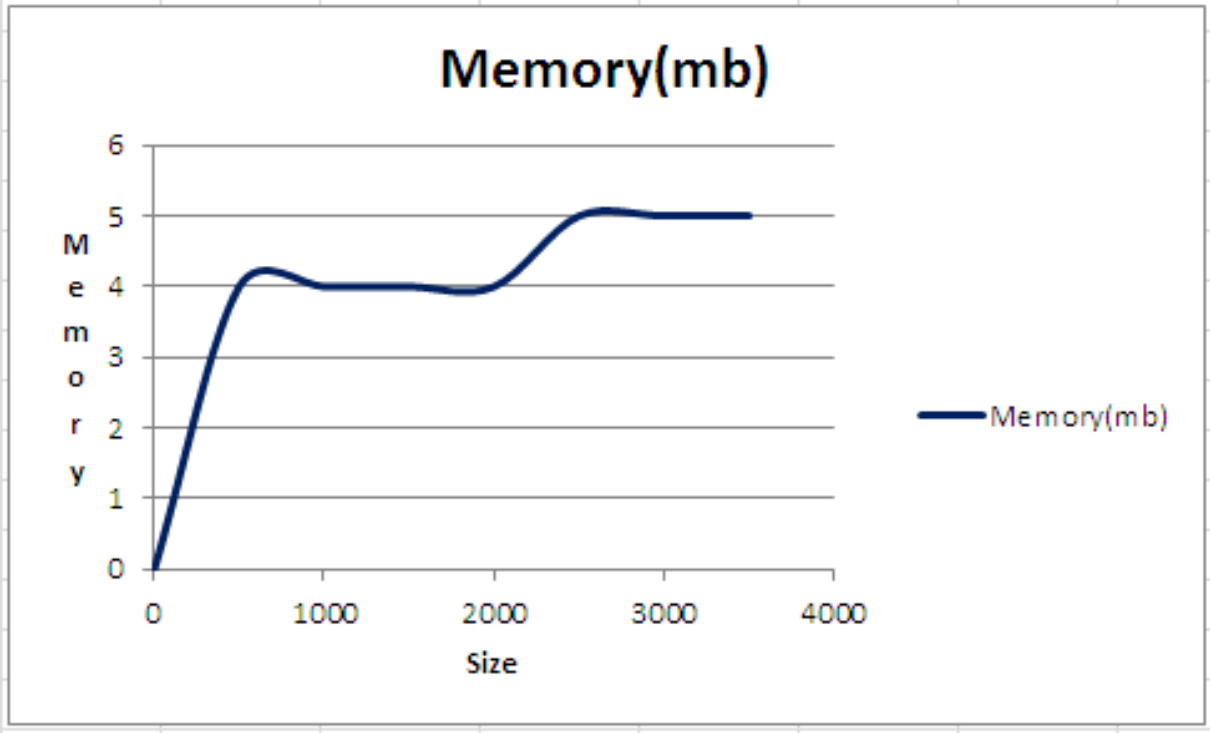


Figure 4.5: a) Size vs Time curve for chess dataset

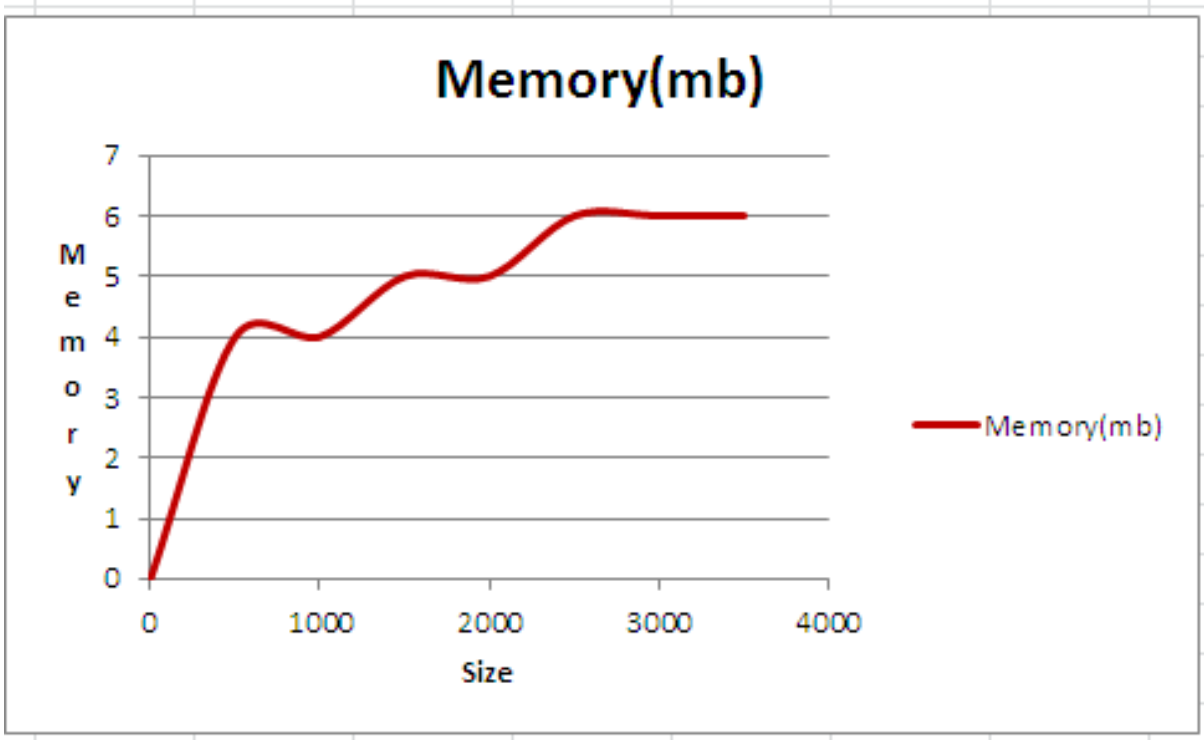


Figure 4.6: a) Size vs Time curve for mushroom dataset

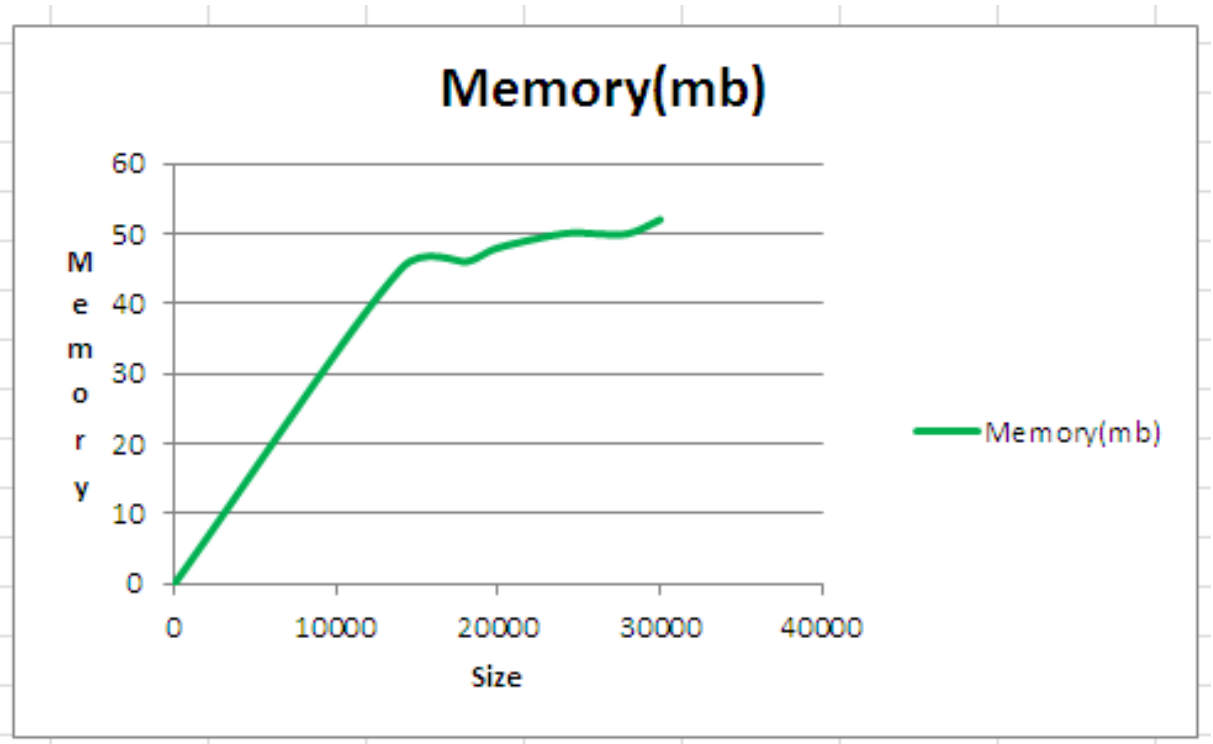


Figure 4.7: a) Size vs Time curve for connect dataset

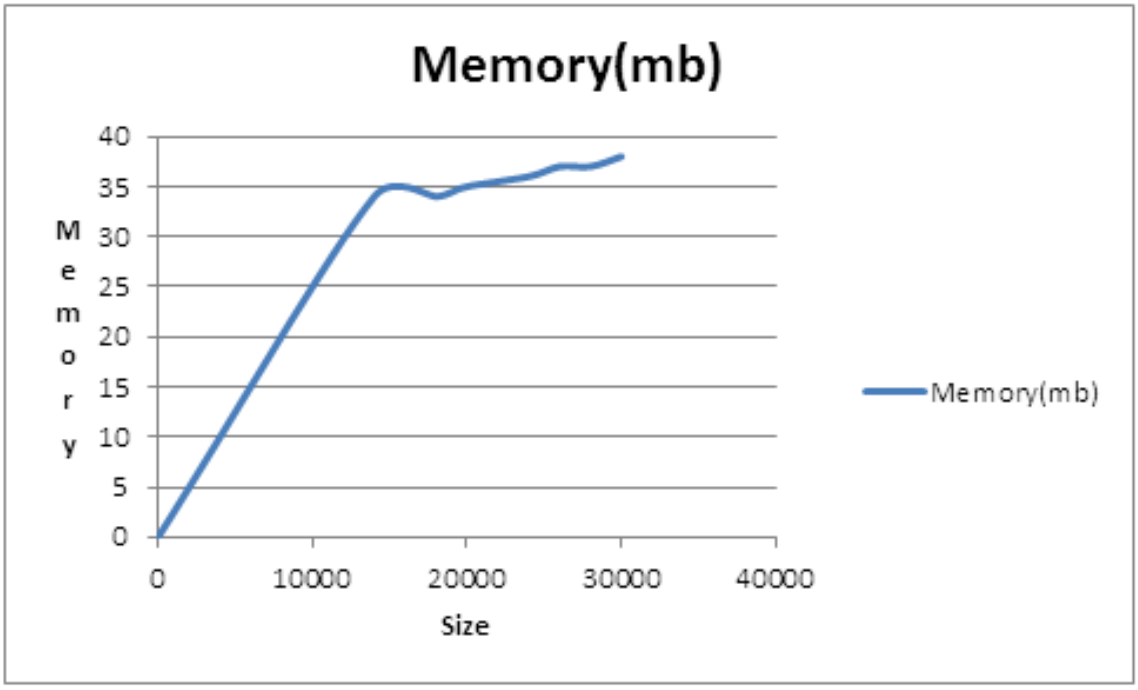


Figure 4.8: a) Size vs Time curve for T10 dataset

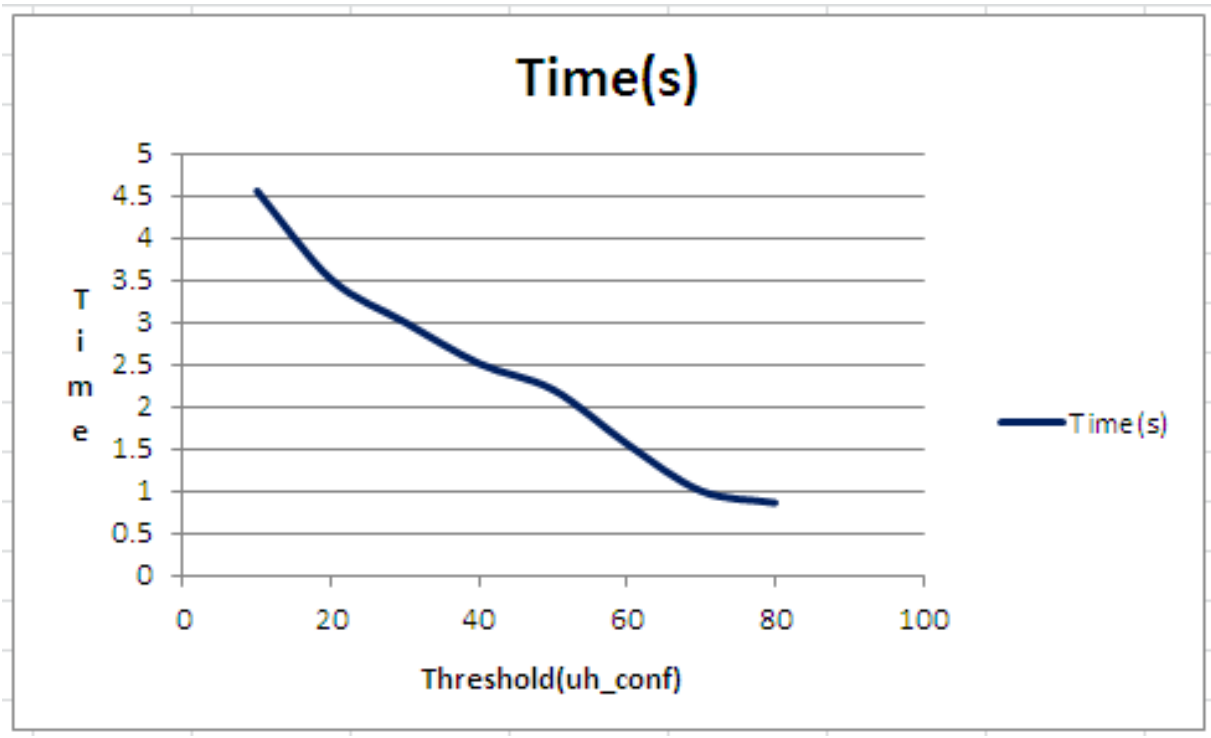


Figure 4.9: a) Size vs Memory curve for chess dataset

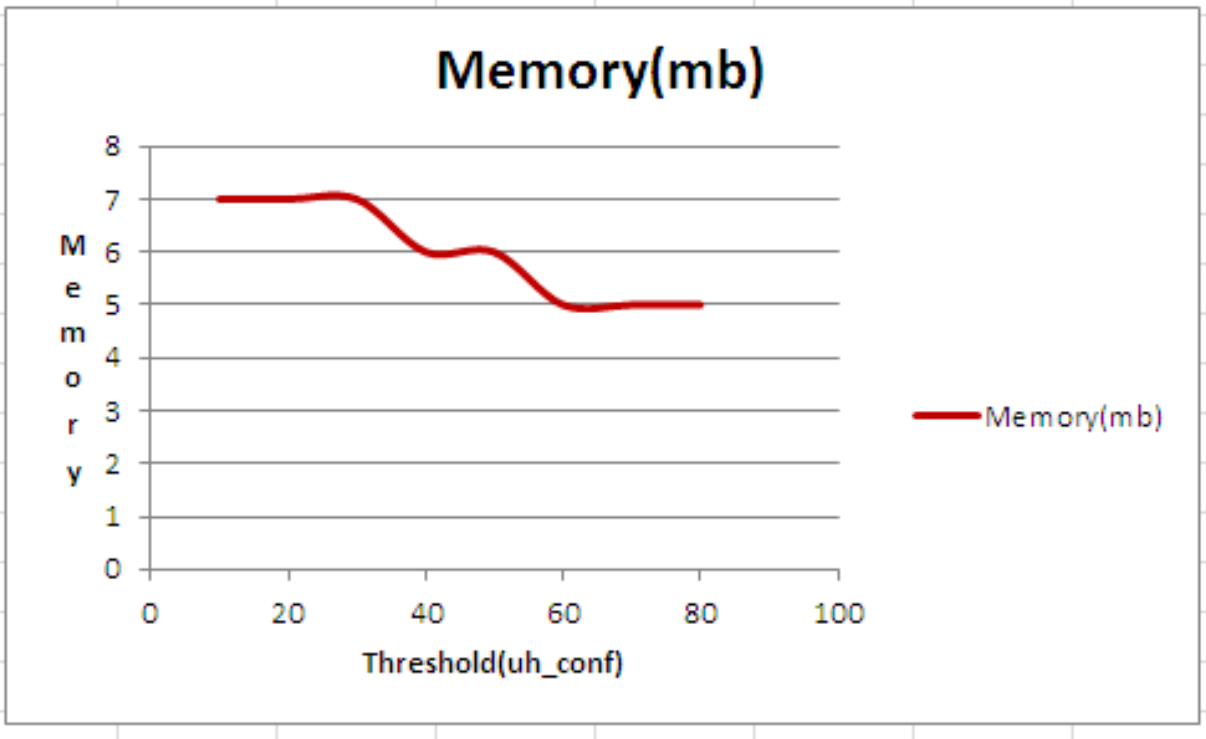


Figure 4.10: a) Size vs Memory curve for mushroom dataset

4.3.3 Threshold (uh_conf) vs Time curve for chess, mushroom, connect and T10 datasets :

In the above graphs [Fig 4.9 to Fig 4.12], x-axis represents the uh_confidence threshold and y-axis represents the time needed for particular uh_confidence threshold. Number of transaction size and minsup_conf threshold is constant in this case. Here, we notice that the amount of time decreases with the increasing amount of uh_confidence threshold. Because when minimum uh_confidence threshold increases, number of itemsets satisfying the minimum uh_confidence threshold decreases. So, the calculation becomes smaller. Here is one more thing to concentrate. In the experiment, here are two thresholds. One is minsup_threshold and other is uh_confidence threshold. But when we are doing our experiment, we will consider the h_confidence threshold. Because actually finally we are deriving correlation from frequent itemsets.

4.3.4 (uh_conf) vs Memory curve for chess, mushroom, connect and T10 datasets :

In the mentioned graphs [Fig 4.13 to Fig 4.16], x-axis represents the uh_confidence threshold and y-axis represents the memory needed for particular uh_confidence threshold. Number of transaction size and minsup_conf threshold is constant in this case. Here, we notice that the amount of memory decreases with the increasing amount of uh_confidence threshold. Because when minimum uh_confidence threshold increases, number of itemsets satisfying the minimum uh_confidence threshold decreases. So, the calculation becomes smaller. Here we are taking uh_confidence threshold between the two thresholds for the same reason mentioned before.

4.3.5 Filtering percentage curves for chess, mushroom, connect and T10 datasets :

In the graph [Fig:4.17], x-axis represents the uh_confidence threshold and y-axis represents the filtering percentage of the four datasets. The blue curve represents chess dataset, red curve represents mushroom dataset, green curve represents T10 dataset and purple curve represents connect dataset. Here minsup_conf threshold and transaction size is constant. Filtering percentage means the number of itemsets generated for different values of uh_confidence threshold. For all four datasets, number of generated itemsets have decreased with the increment of uh_confidence threshold. Because when the uh_confidence threshold is higher, less number of itemsets satisfy the minimum uh_confidence threshold.

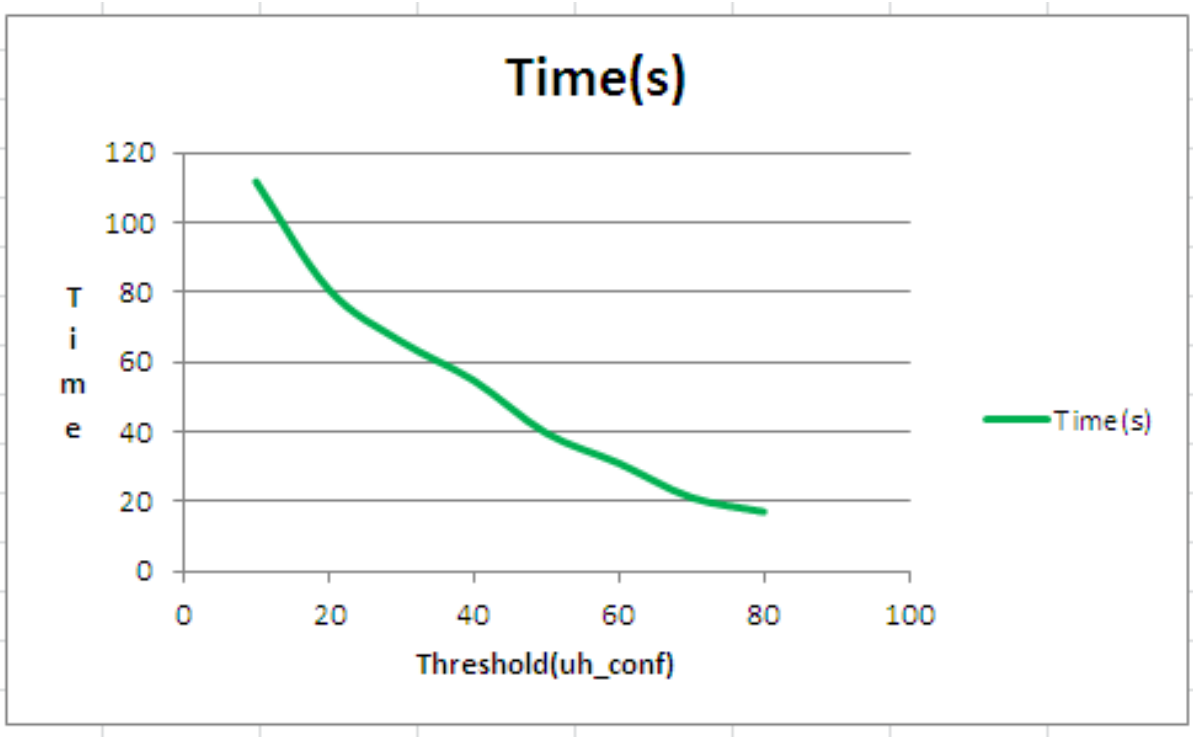


Figure 4.11: a) Size vs Memory curve for T10 dataset

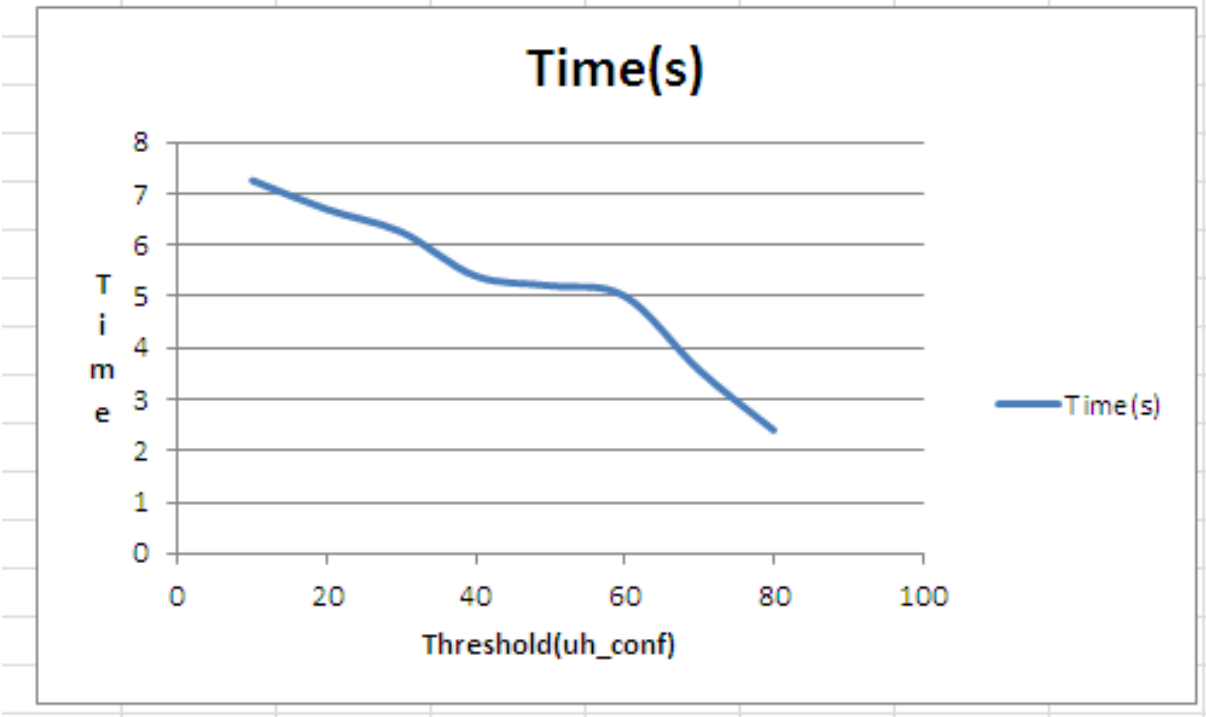


Figure 4.12: a) Size vs Memory curve for T10 dataset

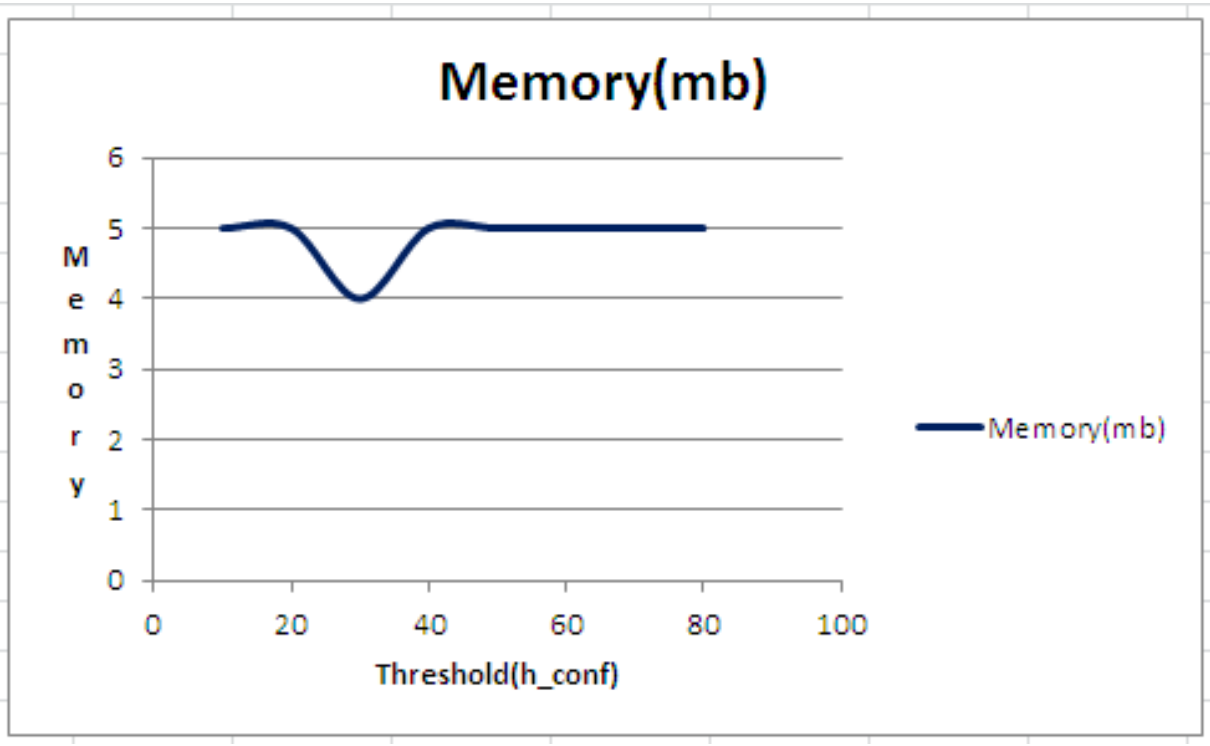


Figure 4.13: a) (h_conf) vs Time curve for chess dataset

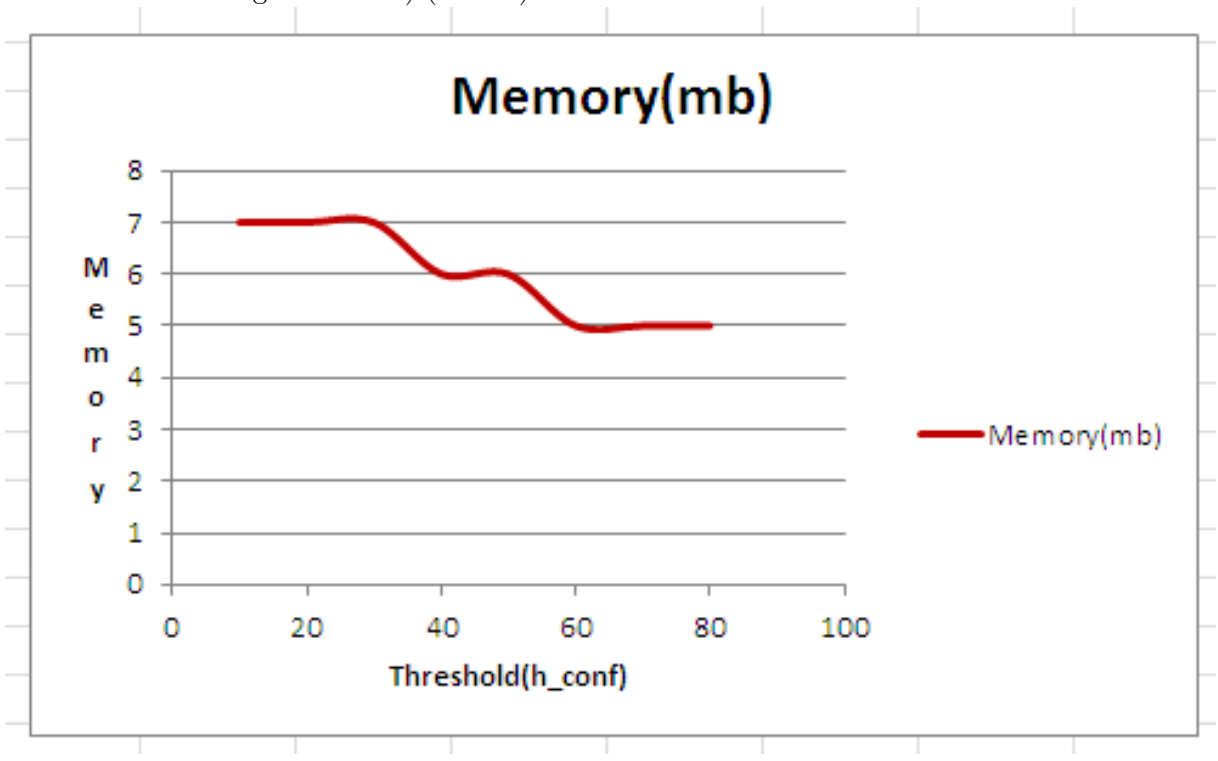


Figure 4.14: a) (h_conf) vs Time curve for mushroom dataset

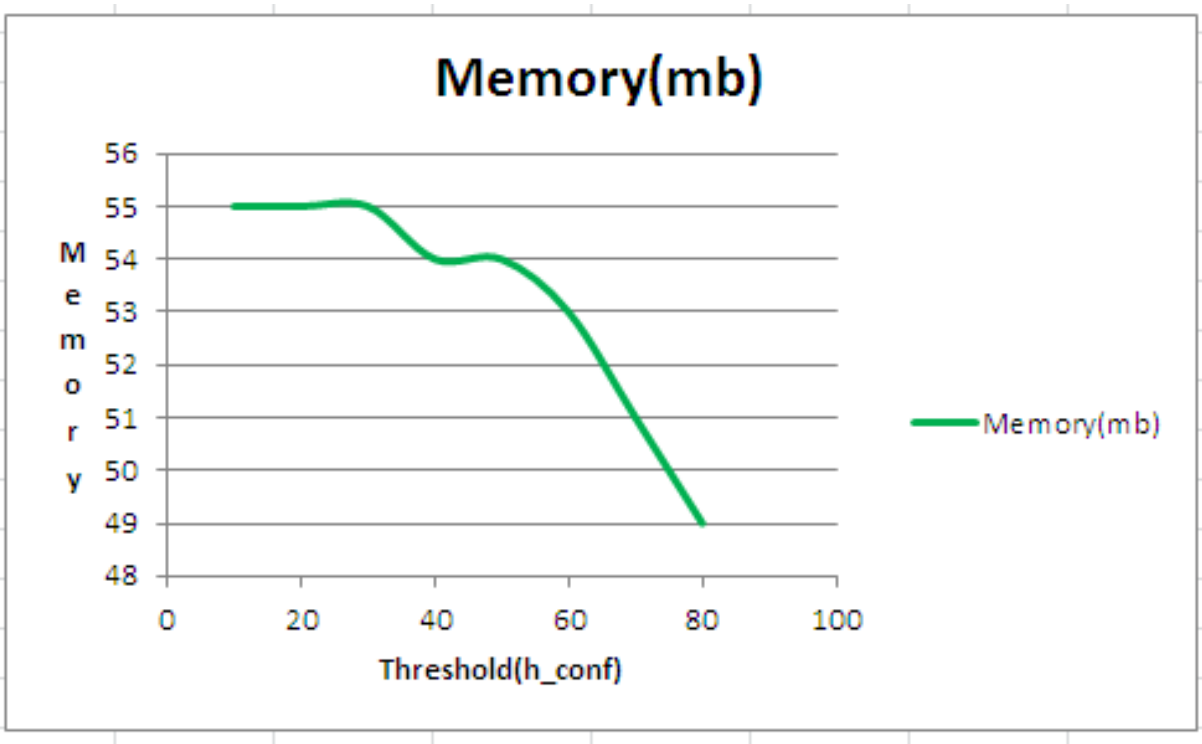


Figure 4.15: a) (h_conf) vs Time curve for connect dataset

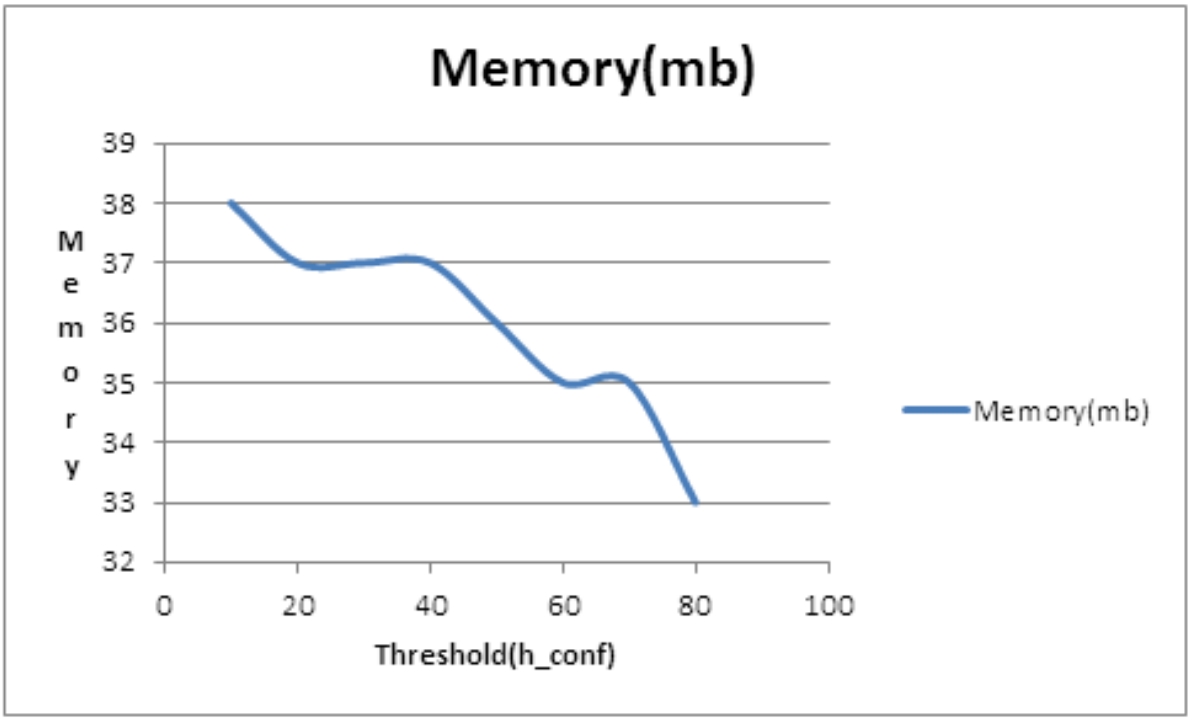


Figure 4.16: a) (h_conf) vs Time curve for T10 dataset

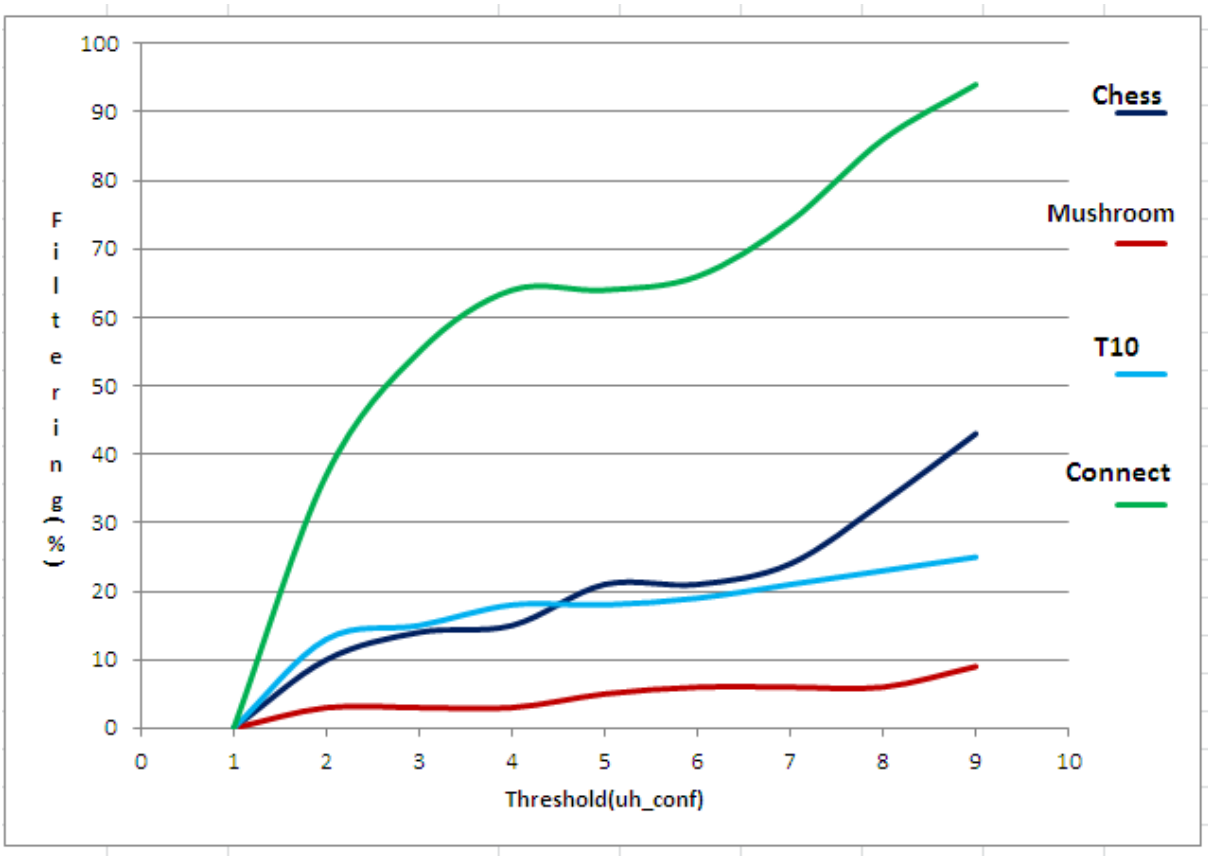


Figure 4.17: Filtering percentage curves for chess, mushroom, connect and T10 datasets

Chapter 5

Conclusion

The mining process in an uncertain data stream is more challenging and complicated than in precise database. In recent years, there have been proposed a few algorithms to mine frequent itemsets from uncertain data stream. In this research work, we have developed an idea to improve an algorithm using existing CUFP[8],DsTree[19] and Affinity[28] property to make it work for evolving running uncertain database.As we have used CUFP[8] approach our **CUFPR-Tree** can generate many false positive patterns as well as imposes higher execution time complexity. To overcome this situation , in our proposed algorithm we have introduced a strong affinity property check method which eliminates a huge amount of false positive patterns during the mining process and reduces complexity.

5.1 Research Summary

In this paper, we proposed tree-based mining algorithms that can be used for mining frequent patterns from dynamic streams of uncertain data with sliding window model. All algorithms apply CUFP-growth with pre_Minsup to find frequent patterns. The mined patterns are then stored in the CUFPR-stream structure together with their expected support values and with their batch id. Then, when the next batch of streaming transactions flows in, the algorithms update the CUFPR-stream structure differently. The naive algorithm keeps a potentially infinite list of expected support values for each node in CUFPR-stream. The CUFPR-Tree reduces the memory consumption by keeping the items with different existential probability and with different batched on the same node.
In addition, we also proposed an enhancement algorithm to check strong affinity that reduces both space and time consumption by keeping only only really correlated patterns for the sliding window model.
We have presented some experimental results to prove our algorithm efficiently. A comparison of frequent itemset and filtered frequent itemset was shown. Datasets used in this thesis also comes from well known frequent itemset mining data repository.We have also showed how our algorithm performs in terms of execution and memory consumption. A few observations were made to build the foundation of this research work. Experimental results supports our logical assumptions to the point and justifies our claim for an improved approach.

Implementing our algorithm requires some more space for storing some extra fragment of information to produce desired result in mining frequent itemsets from uncertain data stream. As in our algorithm each node has to keep track of each batch id with its existential probability value, the tree nodes become primarily overhead. At this point, this can be stated as a weakness of our algorithm, although the difference in memory consumption is quite subtle considering the processing powers of today's modern computers.

5.2 Scope of future studies

The ĆUFPR' algorithm is a new algorithm to mine frequent*reallyassociated* itemsets form uncertain data stream. the algorithm used a very efficient data structure which provide great output regarding the runtime and memory consumption.The main goal of the paper is to target fast and memory efficient computation of the frequent itemset mining approach with appropriate way of removal of false positive patterns. Thus a new area of data mining studies has been explore with newer possibilities.However some points to be focussed for future improvements over our algorithm.
Fist, we have to scan the uncertain data stream tree times to construct our CUFPR-tree.If the databse scanning can be reduced to two times the algorithm will be more efficient in response to time and memory consumption.
Second , the amount of memory consumption can be reduced by implementing mechanisms that does not store the whole tree information in the main memory, instead utilizes the disc cashes using hash based technique.But this is contradiction as it will impose additional overhead for accessing disc memory.
In this chapter , we have discussed about the research summary,we have also mentioned some limitations of our work and indicated some future scopes of research opportunity. This research is a small step toward the development of an efficient algorithm which would be proficiently able to mine frequent itemset and remove the false positive patterns from most challenging uncertain continuously flowing databases.

Bibliography

[1] A. C. Agarwal, R. and Prasad. A tree projection algorithm for generation of frequent itemsets. Journal of Parallel and Distributed Computing, pages 350–371, 2001.

[2] A. C. Agarwal, R. and Prasad. A tree projection algorithm for generation of frequent itemsets. Journal of Parallel and Distributed Computing, pages 145–341, 2006.

[3] L. Y. Aggrawal C.C and W. J. Frequent pattern mining with uncertain data. ACM KDD, pages 29–37, 2009.

[4] B. L. B. Vo. Mining minimal non-redundant association rules using frequent itemsets lattice. Int. J. Intelligent Systems Technologies and Applications, 10(1), 2011.

[5] B. L. Bay Vo. A frequent closed itemset lattice-based approach for mining minimal non-redundant association rules. Internaiona Journal of Database Theory and Application, 4(2), June,2011.

[6] G. C. G. B. Calders, T. Efficient pattern mining of uncertain data with sampling. Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part I. LNCS (LNAI),Springer, Heidel-berg, 6118:480–487, 2010.

[7] L. C.K.S. and T. S.K. Fast tree based mining of frequent itemssets from uncertain data. DASFAA,Part 1, LNCS,7238, pages 272–287, 2012.

[8] L. C.W. and ong T.P. A new mining approach for uncertain databases using cufp trees. ELSEVIER,Expert System with App., 2011.

[9] C. R. L. W. S.-H. L. J. E. Keogh, S. Lonardi. Compression-based data mining of sequential data. Data Mining and Knowledge Discovery, Dallas,Texas,USA, 14(1):99–129, 2007.

[10] H. J. P. J. Y. X. Y.-P. Giannella, C. Mining frequent patterns in data streams at multiple time granularities. AAAI/MIT Press, page 105–124., 2004.

[11] i. Discovering frequent closed itemsets for association rules. Proc. Of the 5th International Conference on Database Theory, LNCS, Springer-Verlag, Jerusalem, page 398–416, 1999.

[12] B. M.-A. H. P.-Q. C. U. D. M.-C. J. Pei, J. Han. Prefixspan:mining sequential patterns efficiently by prefix-projected pattern growth. IEEE 29th International Conference on Data Engineering(ICDE), 0:215, 2001.

[13] R. M. J. Pei, J. Han. Closet: An efficient algorithm for mining frequent closed itemsets. Proc. of the 5th ACM-SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Dallas,Texas,USA, page 11 – 20, 2000.

[14] J. P. J. Wang, J. Han. Closet+: Searching for the best strategies for mining frequent closed itemsets. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 236 – 245, 2003.

[15] M. K. Jiawei Han. Data Mining Concepts and Techniques. Morgan Kaufmann, 2 edition, 2006.

[16] Y. F. Jiawei Han. Discovery of multiple-level association rules from large databases. the 21st VLDB Conference Zurich, Swizerland, 1995.

[17] L. F. Le Wang and M. Wu. Uds-fim: An efficient algorithm of frequent itemsets mining over uncertain transaction data streams. JOURNAL OF SOFTWARE, 9, 2014.

[18] C. K.-S. Leung and F. Jiang. Frequent pattern mining from time-fading streams of uncertain data. ACM SAC, page 983–984, 2011.

[19] C. K. S. Leung and Q. I. Khan. Stree: a tree structure for the mining of frequent sets from data streams. IEEE International Conference on Data Mining ,, pages 928–932, 2006.

[20] C.-S. Leung and B. Hao. Mining of frequent itemsets from streams of uncertain data. In Data Engineering, 2009. ICDE '09. IEEE 25th International Conference on, pages 1663–1670, March 2009.

[21] S. L. Leung, C.K.-S. Equivalence class transformation based mining of frequent itemsets from uncertain data. ACM SAC, page 983–984, 2011.

[22] M. M. Leung C.K.S. and B. D.A. A tree based approach for frequent pattern mining from uncertain data. pages 553–561, 2008. Lecture Notes in Computer Science(LNCS),5012.

[23] G. S. L. Michael J.A. Berry. Data Mining Techniques for Marketing,Sales and Customaer Relationship Management. Wiley Publishing, Inc., 2 edition, 2004.

[24] E. Omiecinski. Alternative interest measures for mining as-sociations. January/February 2003.

[25] C. J. V. Rijsbergen. Information Retrieval. Butterworths, London, 2 edition, 1979.

[26] B. S. J. o. S. K. Tanbeer, C. F. Ahmed and Y. K. Lee. Sliding window-based frequent pattern mining over data streams. Information Sciences, 179:3843–386, 2009.

[27] Website. The fp-growth algorithm.

[28] H. Xiong, P.-N. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support. In In Proceedings of the 3rd IEEE International Conference on Data Mining, pages 387–394, 2003.

[29] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. CiteSeerx, 2002.