Improve Frequent Pattern Mining in Data Stream using Sliding window

A Dissertation Submitted

By

Himanshu Shah

To

Department of Technology & Science

In partial fulfillment of theRequirement for the

Award of the degreeof

MASTER OF TECHNOLOGY

IN

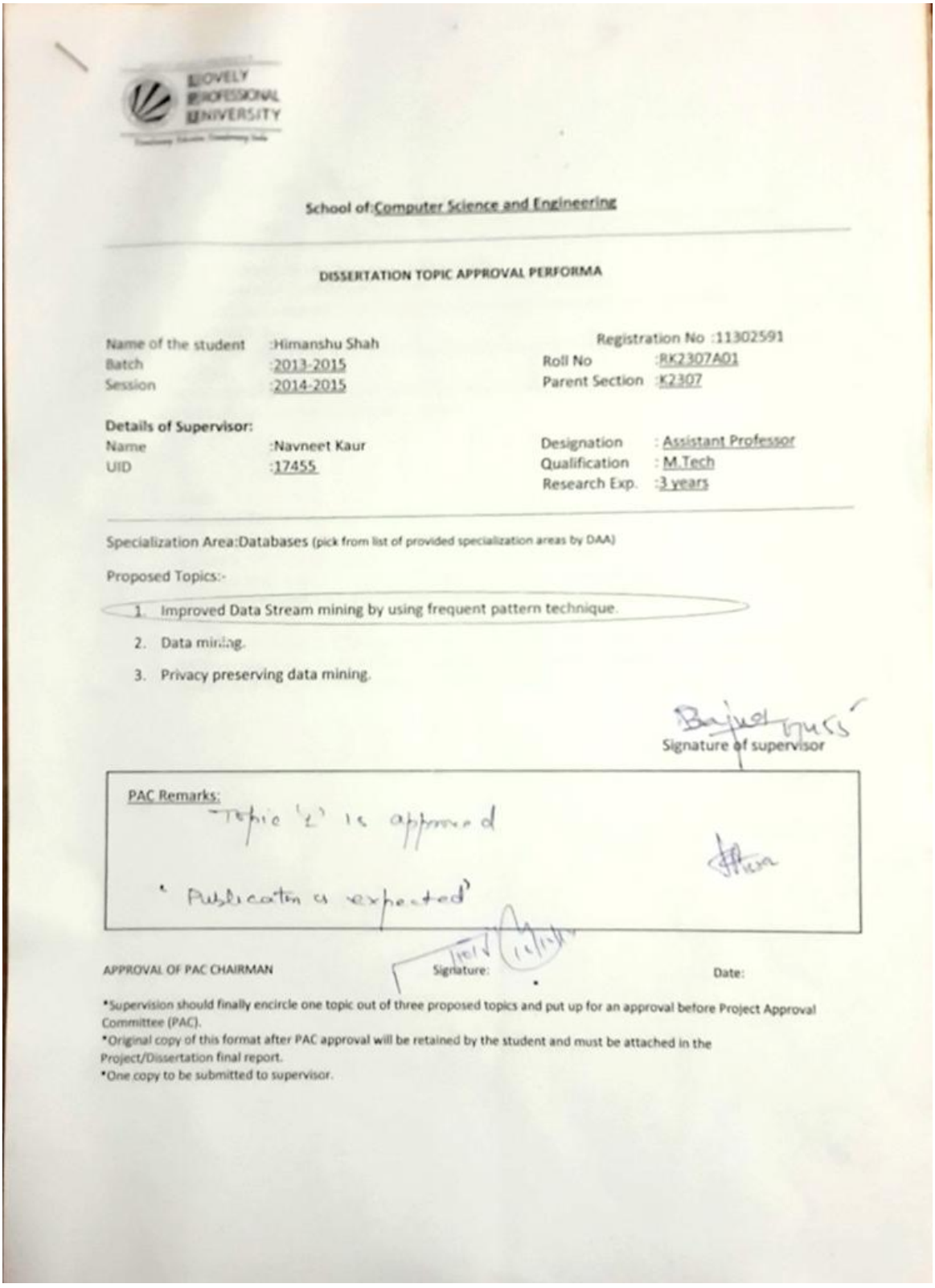
Computer Science and Engineering

Under the Guidance of

Miss. Navneet Kaur

(April 2015)

PAC FORM



LOVELY PROFESSIONAL UNIVERSITY

School of:Computer Science and Engineering

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the student :Himanshu Shah
Batch :2013-2015
Session :2014-2015

Registration No :11302591
Roll No :RK2307A01
Parent Section :K2307

Details of Supervisor:
Name :Navneet Kaur
UID :17455

Designation : Assistant Professor
Qualification : M.Tech
Research Exp. :3 years

Specialization Area:Databases (pick from list of provided specialization areas by DAA)

Proposed Topics:-

1. Improved Data Stream mining by using frequent pattern technique.
2. Data mining.
3. Privacy preserving data mining.

Signature of supervisor

PAC Remarks:
Topic '1' is approved
Publication is expected

APPROVAL OF PAC CHAIRMAN Signature: Date:

*Supervision should finally encircle one topic out of three proposed topics and put up for an approval before Project Approval Committee (PAC).
*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.
*One copy to be submitted to supervisor.

Abstract

Data mining refers to “mining” or extracting knowledge from large amounts of data. It is the process of finding correlations or patterns among dozens of fields in large relational databases. Stream data are continuous, rapid, time varying and unpredictable that continuously arrives to store or process at a system and requires quick response. Frequent pattern in stream data is very challenging because data can be scan one time only. Due to this reason traditional approach cannot be use for data stream. The sliding window model which can perform mining operations focusing on recently accumulated parts over data streams by temporary storing part of the data and processed. It is hard to mine all of the frequent patterns in the data stream environment since generated patterns are remarkably increased as data streams are continuously extended which result in memory storage overflow and performance degradation.Many algorithms are based on FP-Growth which compresses the database representing frequent items into a frequent-pattern tree (FP-tree) which retains the itemset association information.

Hence our main Approach is to improve accuracy of Stream Data by developing tree structure similar to FP Tree and also optimize time and memory complexity. For Implementation of this particular concept the Methodology & Tools which would be used by us are C#.net.

CERTIFICATE

This is to certify that “Himanshu Shah” has completed M.Tech dissertation titled “ImproveFrequent Pattern mining over data stream using sliding window” under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigation and study. No part of the dissertation proposal has even been submitted for any other degree or diploma.

The dissertation is fit for the submission and the partial fulfillment of the conditions for the award of M.Tech Computer Science &Engineering.

Date:

SignatureofAdvisor

Name:

UID:

ACKNOWLEDGEMENT

I would like to thank **LOVELY PROFESSIONAL UNIVERSITY** for giving me opportunity to use their resource and work in such a challenging environment. I am grateful to the individuals whom contributed their valuable time towards my thesis.

I wish to express my sincere and heart full gratitude to my guide “Miss. Navneet Kaur” Assistant professor, who guides me to take up this thesis in sync with global trends in scientific approach. I would also like to extend my gratitude to my friends and family who always encouraged and supported me in this thesis work.

Last but not the least; I would like to thank all the staff members of department of Computer Science & Technology who have been very patient and co-operative with us.

DECLARATION

I hereby declare that the dissertation entitled,**"Improve Frequent Pattern mining over data stream using sliding window"**submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma

Date:

Investigator

Regn. No.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1
Introduction

Data mining is the process of extracting valuable information from large amount of data. This valuable information is helpful for business perspective. This information can be categorize, summarize and can also be represent in chart form. Data mining is series of steps start from extracting data from various location then pass through selection, cleaning, integration, Transformation, mining which generates patterns that give knowledge. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases [1]

Pattern generation is time consuming task. it takes time but very helpful for business planning, Trends analysis, future forecasting and Market basket analysis. Pattern generation involves scanning of data more than one time. In mining, all patterns are not important, it is important to apply interesting measure so after mining, one can have only interesting pattern.

Data Mining is the knowledge discovery process also known as KDD process.

The Process of extracting knowledge from raw data is as shown in figure below.

- Data Cleaning, Integration, Selection, Transformation, Mining, Evaluation and Presentation

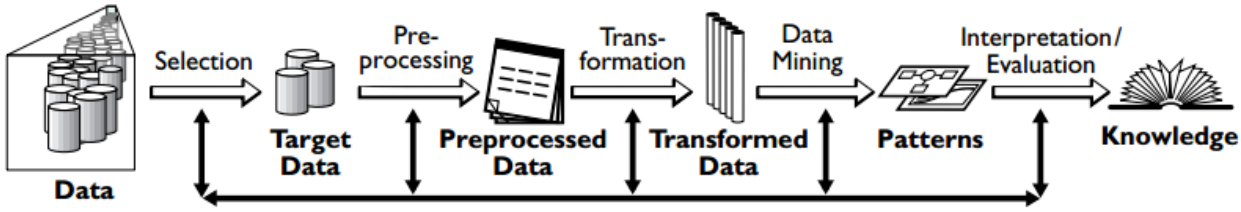


Figure 1-1 KDD Process

1.2 Introduction to Data Stream Mining:

Today many organizations have huge amount of data store in database and it continuously grow. Not only social network site, there are many other sources which generate

enormous amount of data like sensor network, telephone records-commerce site, stores, Internet traffic, on-line transactions in the financial market and Real-time surveillance systems, remote sensors and other active environments. Everyday 10tb of data stored in social site so from all over world approx 100tb of data are generate.

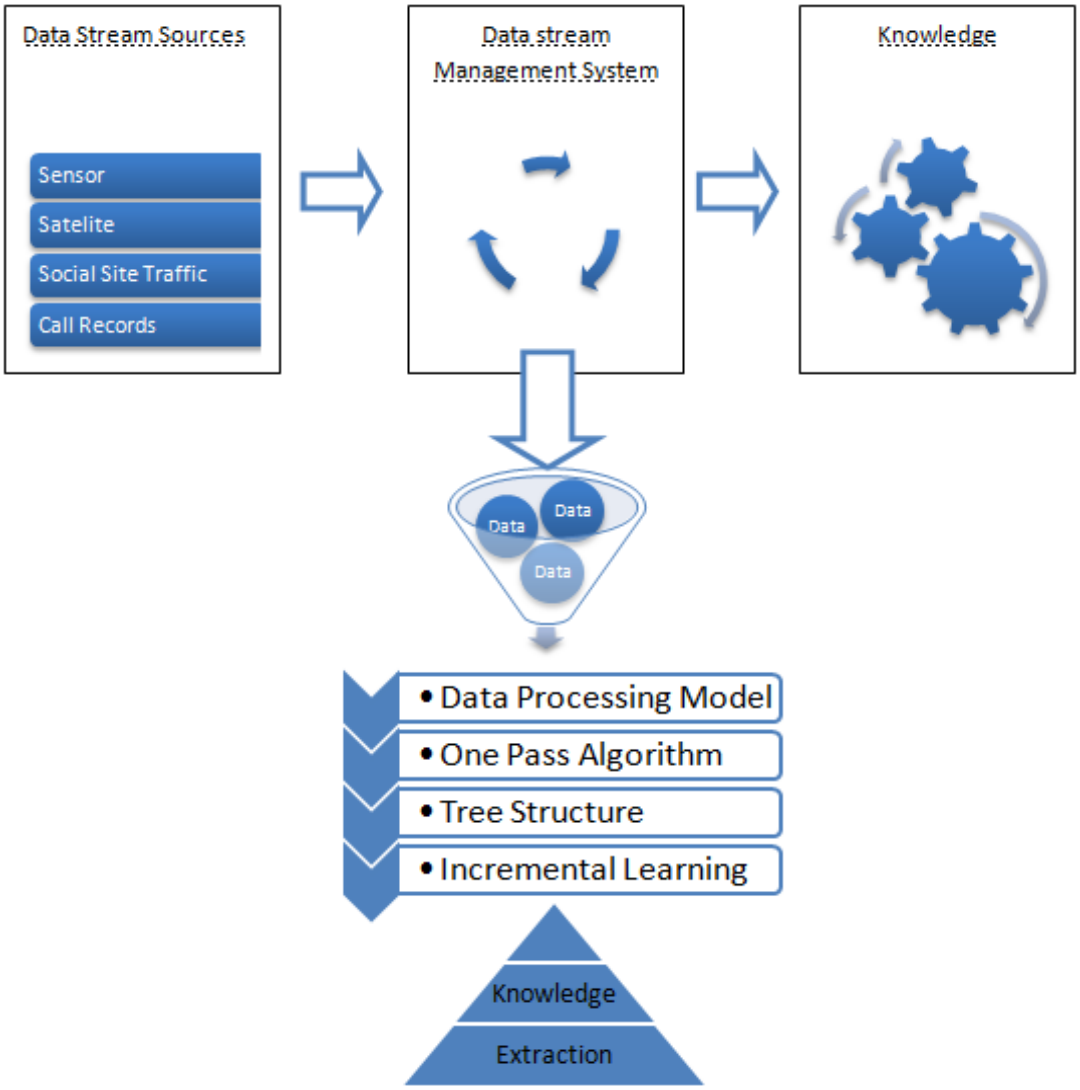


Figure 1-2 Process of Stream Data Mining

High speed uninterrupted sequence of data that arrive continuously know as data stream. Mining frequent pattern in stream data is very challenging because data can be scan ones. It's very challenging to processing and turning this data into useful information. Many author proposed various techniques to handle stream data. Based on application need data processing model is used for data mining.

The main challenges for Stream Data Mining are:

1. Developing and Designing robust and fast mining methods for such high speed data streams.

2. Computational challenges.
3. Need to detect concept drift and data distribution due to active nature.
4. Storage, querying and mining challenges.

Various strategies are introduced to overcome data stream challenge of Processing along with storage of high speed and uninterrupted data streams. The data are very large like millions of transaction in volume, fast changing, potentially unbounded and temporally structured cause challenging problems in stream data mining field.

The conventional OLAP and mining methods cannot be use due to requirement of several scans of the data and so it's not feasible for stream data applications. The established methods expect data to be stored first. After that data is process as offline using composite algorithms. These algorithms cannot be used because it makes numerous scan over the data. But data stream is limitless and generates with high rates. So it is impractical to store such data so it is necessary to design such algorithm which is light and speedy mining methods used for data stream.

For that algorithms should be designs such a way that expect only one pass over the data and it should work with restricted memory. Due to highly active nature of stream data it is difficult to detect and work with concept-drift by stream mining algorithms.

	DBMS	DSMS
Data type	Static data	Stream Data
Relationship	Persistent data	Volatile data stream
Access	Random	Sequential
Query	One time	Continuous
Storage	Passive repository	Active repository
Available memory	Flexible	Limited
Algorithms	Processing time is not a constraint	Processing time is most important as data may skip
Results	Accurate	Approximate
Response speed	No time requirements	Real-time requirements
Data scan	Flexible	One time scan only
Data Schema	Static	Dynamic

Table 1 Difference between DBMS and DSMS

As shown in table 1[10][15], Stream data are continuous, rapid, time varying and unpredictable and unbounded and require quick repose. Therefore traditional DBMS and algorithms which are designed for static data are not suitable for mining stream data because it cannot fulfil the requirement of stream data mining.

Due to high storage cost and huge amount of data, it is not possible to accumulate whole stream data or to scan through data various times. So there are various challenge in computational, communication, and storage capabilities of computational systems take place. Due to large volume and rapid data, it is needed to utilize semi automatic interactional technique to elicit embedded knowledge from data.

GENERAL ISSUES IN DATA STREAM MINING

There are some crucial issues that need to be taken into account when developing association rule for stream data.

1. Data Processing Model

According to the research of Zhu and Shasha[30] , there are three data stream processing models, Landmark, Damped and Sliding windows[11].

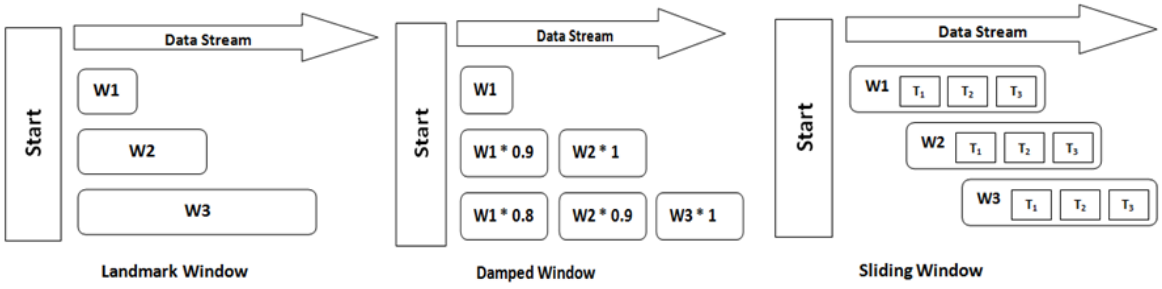


Figure 1-3 Processing model

The Landmark model mines all frequent itemsets over the entire history of stream data from a specific time point called landmark to the present. In this model, we treat each time point after the starting point equally important. This model is not suitable for mining where most recent information and real time data are very important such as stock market.

The Damped model mines frequent itemsets over stream data. In stream data, each transaction has weight and this weight decreases with time. So in this model new and old

transaction has different weights. Due to above characteristic of damped model, It is known as Time Fading model.

The Sliding window model mines frequent itemset over stream data by temporary storing part of the data and processed. In this model, size of sliding window decided by need of application and system resources.

Besides above mention windows, Jiawei Han et. al. proposed tilted time window model. In this model, we are interested in frequent itemsets over a set of windows[29]. Each window corresponds to different time granularity for example we are interested in every ten minutes for the hour before that. Each transaction in this window has weight.

2. Memory Management

This is major issue in mining stream data. This includes choosing of efficient and compact data structure algorithm which can efficiently stored, updated and retrieved data.
In traditional algorithm, we do multiple scan over available data. This is not possible in data stream because there is not enough memory space to store all the transaction and their counts. In simple terms, memory size is bounded and Hugh amount of data are arrives continuously.
If we store the information in disks, the additional I/O operation will increase the processing time.

3. Data preprocessing

Data preprocessing is crucial aspect in the process of data mining. If data input to algorithm is not in proper format then it cannot process efficiently. So preprocessing is needed and in which existing data transform into new data which is in proper format and suitable for processing. Different data mining tools available in the market have different formats for input which makes the user forced to transform the existing input dataset into the new format.

4. One Pass algorithm

There are many algorithms for mining stream data. Based on result, they are categorizing as exact algorithms or approximate algorithms.

In exact algorithms, The result consist of all the itemsets which satisfy support values greater than or equal to threshold support. To Produced accurate result in stream data, additional cost is needed.

In approximate algorithms, the result is approximate result with or without an error guarantee.

5. Concept Drift

Since stream data are rapid and time varying, we cannot assume that total number of class are fixed because itemsets which are frequent can change as well with arrival of new data. So there is need of frequent updating of model, because old data are inconsistent with the new data. This problem is known as Concept drift. If we neglect non frequent itemsets from consideration which can be frequent itemset later, we cannot get this information. Therefore technique is needed to handle concept drifting.

6. Producing and maintain association rules

Mining Association rule involves a lot of memory and CPU costs. There is also one problem; processing time is limited to only one online scan. So there is need of real time maintaining and updating association rule. However stream data, if we update association rules too frequently, the cost of computation will increase drastically.

7. Resource Aware

Resources such as memory space, CPU, and sometimes energy are very precious in data stream mining. One cannot ignore the resources availability, for example when main memory is totally used up in processing algorithm, data will be lost and it lead to inaccuracy of results.

In general, if we don’t consider this problem, it will degrade the performance of the mining algorithm.

Application Dependent Issue

Based on need of application environments, association rule mining algorithms has different needs[11].

1. Timeline Query

In some application, recent data are important while in other certain period of time data is important base on user interest. So it leads to issue of efficiently storing and retrieving with timeline.

2. Multidimensional Stream Data

There are some applications where stream data are multi dimensional such as sensor data network. So there is need of multi-dimensional processing techniques for mining association rule. How to efficiently store, update and retrieve the multidimensional information to mine association rules in multidimensional data streams is an issue.

3. Distributed Environment

In a distributed environment, stream data comes from multiple remote sources. So in this type of environment, there are various issues that need to be consider such as communication overhead, computation overhead, and resource overhead. Therefore there is need of distributed algorithm which imposes low communication overhead, controlled interactive response time and which can parallel and incrementally generate frequent itemsets.

4. Online Interactive Processing

There are various algorithms which allow user to modify the mining parameters during the processing period.Therefore, how to make the online processing interactive according to user inputs before and during the processing period is another important issue.

5. Visualization

In some data stream applications, especially monitoring applications, there is a demand for visualization of association rules to facilitate the analysis process. Visualization of data in form of graph and plot helps user to understand the relationship between related associations rules better so that they can further select and explore a specific set of rules from the visualization.

In Real World, Items have their importance called weight. For example in market, each items like milk, chips and any food or gadget etc. are having their weight. we can simply

consider by their price or profit. Therefore instead of finding frequent itemset by their frequency, go for utility. This is known as utility mining.

In High utility mining, transactions are having two type of utility like internal and external utility. Internal utility shows importance of items in transaction whereas external utility of items are distinct. To calculate utility of itemset, simply multiply external utility with internal utility.

This high utility mining is very useful and there are many business areas like retail, inventory and online e-commerce as well.

Consider an example

TID	**ITEMSET**
1	(A, 2),(B,1),(C,3),(E,2)
2	(A, 1),(D,1),(F,2)
3	(B,2),(C,1),(D,3),(E,3)
4	(C, 3),(F,2)
5	(A, 2) ,(B, 1), (D, 1) ,(E,2)
6	(A, 2),(B,3),(C,2)

Table 2 Transaction list

Item name	A	B	C	D	E	F
External Utility	5	3	6	1	2	8

Table 3 External utility

In this example, A,B,C,D,E,F occurred 4,4,4,3,3,2 times respectively.

Here minimum support=4 is set. Therefore only 3 item A, B, and C satisfy minimum support threshold. Therefore this items are frequent items because their occurrence values are equal to the threshold value. Items A, D and F are called as infrequent item sets. So they are omitted. Thus this is called as frequent pattern mining.

In High utlity mining.

$U_D(A)= 7 \times 5= 35, U_D(B)= 21, U_D(C)= 42, U_D(D)= 5, U_D(E)= 14, U_D(F)= 36,$

Chapter 2
Review of Literature

Frequent pattern mining is first start with static database. In Traditional algorithm, Aprior (Agrawal & Srikant,1994) and FP-growth (Han et. al.,2004) are pioneer algorithms. Apriori uses Breadth first search(BFS) and generate numbers candidate pattern in mining process and it need to perform scanning repeatedly. FP growth algorithm is based on Depth First Search(DFS) which improves mining process. FP-growth algorithm overcome the problem of repeated scan and able to mine pattern with two fixed database scan. Till now, Most of algorithms are based on FP-growth.

These traditional algorithm are not suitable for data stream environment. Though FP-growth able to mine pattern with two scan, it is not suitable for data stream, Since data in data stream can be scan one time only .Based on this fact, There are three processing model proposed .

Sliding window based frequent pattern mining over data stream.

Sliding window consider recent data as more important and based on that there are many approaches (ahmed et al.,2009;Chen et al.,2012;Deypir et al.,2012;Farzanyar et al.,2012;Li,2011;Mozafari et al.,2008;Shie et al.,2012;Tanbeer et al.,2009b;Zhang & Zhang,2011) have been proposed. An Efficient approach proposed by(Tanbeer et al., 2009a, 2009b),which uses tree restructuring technique, BSM which perform restructuring operation more effectively by path adjusting method, etc..Algorithm estWin Proposed by Chang and Lee(2005) which finds recent frequent pattern and uses reduced minimum support threshold named significance to early monitoring of itemsets before they become actually frequent itemset.The Moment algorithm (Chi et al.,2006) finds closed frequent pattern bt maintaining a boundary between frequent closed itemset and other itemset.

Concept Drift.

Koh and Lin(2009) proposed an innovative approach which continuously monitors the incoming transactions to detect the occurrence of a concept shift. The frequent itemset are

mined when a concept shift observed. F.Nort et al.(2013) proposed an approache based on concept drift named TMoment which uses sliding window

Weighted Condition is Frequent Patter mining over data stream

In real world, Each item have their importance known as weight or utility(Price or profit).Not only support but weight also play crucial factor in mining process. But the main challenge is to maintain anti-monotone property. Due to the fact that wighted infrequent pattern can become weighted frequent which normally destroy that property, Many researcher able to maintain this property by variety of methods(Ahmed et al.,2009,2012;Wang & Zeng,2011;Yun & Ryu,2011;Yun et. al.,2011,2012).Chang and Lee(2006) introduced an algorithm called estDec based on time decay model in which each transaction has a weight decreasing with age. In this method, in order to reduce the effect of old transactions in the set of frequent patterns a decay rate is defined. Up-Growth(Tseng et al.,2010) which is novel algorithm that uses various pruning and counting strategies during mining process. IWFP Algorithm (ahmed et al., 2012) is a weighted frequent pattern mining, Applying BSM method. THUI-Mine (Tseng et al.,2006) is the first algorithm for mining high utility itemsets. WFPMDS (Ahmed et al.,2009) mines weighted frequent pattern using sliding window. The algorithm produce recent mining result by using sliding window and uses tree restructuring work with BSM technique.

This section will describe about literature review or the studies by paper wise

In paper [1] Author uses prefix tree structure called as Compact pattern Stream (CPS)-tree [20]. They use a novel technique which can restructure tree dynamically to handle stream data. For Restructuring, they use BSM method and Path adjustment method. It finds exact set of latest frequent pattern with the use of Sliding window. For each new item arrive in window, it restructure the tree. This is main disadvantage because it needs more memory and time for reconstruction.

In paper [2] Author proposed weighted sliding window (WSW) technique in the year 2009. This Technique allow user to state various parameter for data mining like size of window, weight of window and number of window.

In each window, every transaction has weight and if the weight satisfies smallest weighted threshold rate then it is consider as frequent itemset. For large window size,

running time of this model decreases. This is happened for the reason that small window size, number of frequent itemset is small due to small number of transactions. This model uses Apriori algorithm for candidate generation and this may take more memory and time. Therefore instead of Apriori, we can use another algorithm like ‘eclate’ to improve mining.

In paper [3] Author proposed a technique which is based on bit sequence and it worked on 3 phase. They are 1) Window Initialization 2) Window Sliding and 3) Pattern creation. Based on the MFITransSW they introduce one more algorithm called MFI Time SW to discover the set of frequent itemsets with the use of time sensitive sliding window. Both of the technique takes more memory usage when window size increased. So hybrid approach or new technique can save time.

In paper [4] Author proposed an efficient technique known as WSFI mining with normalized weight over data stream. This algorithm works in 3 phases. First phase is to divide stream data into 3 categories. They are frequent itemset, recent itemset, and infrequent itemset. Second phase is to store compress information of frequent itemset. In the Last phase, algorithm discovers frequent itemsets. This is very efficient algorithm. This algorithm can be improved by careful pruning of infrequent itemset.

In paper [5] Author introduce new algorithm called as Hybrid streaming, H-stream for short in the year 2011. They uses H tree for storing and maintaining historical and potential frequent itemset. It is used for collaborative and comparative frequent pattern mining. Author uses real time news paper data for analyzing. Main advantage of this algorithm is that it can efficiently mine frequent pattern from multiple streams. Data may have confidential information, so one can apply privacy preserving technique to protect confidential data. It is often challenge to perform privacy preserving in high speed data stream.

In Paper [6] Author mainly focused on data compression and data privacy. They uses Moment algorithm and it works on fixed sized sliding window model. This algorithm can be improved by adding noise for security purpose.

In Paper [7] Author believe that stream data are very fast, dynamic and time varying in nature and one cannot assume that there is only fixed number of class. This is major problem because of model built on old data becomes incompatible with the arrival of latest data and therefore frequent updating of the model is essential. This problem is identified as concept drifting. This problem address by the author and proposed general framework which worked on drifting. The proposed method combine weighted multiple classifiers by their estimated forecast accuracy to mine stream data.

In Paper [8] Author uses Landmark model and stores the majority frequent items and their count It gives accurate result but for the accurate result additional cost is needed. This algorithm takes 2 scan to generate exact result items set. Main advantage of this algorithm is that its memory usage is low. This algorithm rejects infrequent items and all their information. Infrequent items may become frequent in future therefore it needs to be stored. It requires 2 passes to generate exact result so one can improve this by single scan. In additional it gives no guarantees regarding false positive.

In Paper [9] Author proposed classic algorithm for mining known as "Lossy Counting Algorithm". This model is based on Apriori property [1] which says that "All nonempty subsets of a frequent itemset must also be frequent."It uses landmark window for processing data stream. Algorithms have good average space complexity. It generates all frequent itemset and there is no false negative.

In Paper [10] Author proposed new method (named T-HUDS) for finding top-k high utility patterns using sliding windows of a data stream. They uses compressed tree structure called HUDS-tree which is similar to FP-Tree like data structure. They proposed a strategies for dynamic adjusting the minimum utility threshold. Also they claim that no top-k high utility items are missed. Their algorithm also reducing the number of data scans can reduce runtime and also fit into main memory.

In Paper [11] Author introduced algorithm known as MAFIA for mining Maximal frequent itemset. They introduced novel search strategy of algorithm that integrates a depth first traversal of the itemset lattice with effective pruning mechanism. They represent data in vertical bitmap form and uses relative bitmap compression schema.

Chapter 3
Present Work

Problem Formulation

Now a day's many organization and researcher taking interest in stream data mining. Currently there are many sources produces stream of data continuously which are unbounded, rapid and highly dynamic in nature. Example include sensor networks, wireless networks, radio frequency identification (RFID), customer click streams, telephone records, multimedia data, scientific data, sets of retail chain transactions, etc. These sources are called data streams.

Data stream mining is important research topic in research community and the number of researches also growing in this field. Many algorithm and technology developed and evolved for handling complexity and volume of data, still there is need of general purpose algorithm, model with smaller space complexity, smaller time complexity and high performance in nature[15].Since it is under research area there are wide chances of exploration.

Objectives

- To analyze the performance of current algorithm.
- To design and develop algorithm that reduce memory space and runtime.
- There is tradeoff between response time and accuracy of result. So we would try to optimize them.

Methodology

A data stream $D=\{B_1,B_2,\ldots,B_N\}$ is a an infinite sequence of batches where each batch Bi contains a set of transactions i.e. $Bi=\{T_1,T_2,\ldots,T_K\}$ where $k > 0$.Each transaction $T=\{(i_1,q_1), (i_2,q_2), \ldots, (i_n, q_n)\}$ is a set of items where i represent an item such as $i \in I$ and q represent quantity of item in transaction. An itemset is a non-empty set of items. An

itemset with size k is called k itemset. A window W is a sliding window which has number of batches W = {B1, B2, …,B_m}.

Definition 1.Utility of an item I in transaction Tj

Tj is defined as :u(i,Tj)=q(i,Tj) x p(i) where q(I,Tj) is quantity of an item and p(i) is external utility of item

Definition 2. Utility of an itemset X in a transaction

Tj is defined by: $u(X,Tj) = \sum_{i \in X} u(I,Tj)$.

For example,$u(\{bc\},T_3)$=2 x 6 +3 x 5 =27 in fig 1.

Definition 3. Utility of an itemset X in a data set D

$u_D(X) = \sum_{X \subseteq Tj \wedge Tj \in D} \sum_{i \in x} u(I,Tj)$.

We use u(X) to denote u_D (X0 when data set D is clear in the context.

Definition 4. Utility of a transaction

T_j is denoted as $TU(T_j)$ and computed as $u(T_j , T_j)$.

Definition 5. (High Utility Itemset (HUI))

An itemset X is called a high utility itemset (HUI) on a data set D if and only if $u_D(X) \geq$ min_util is called a minimum utility threshold.

Definition 6. Transaction-Weighted Utility (TWU)

TWU of an itmset X over a dataset D is defined as $TWU_D(X) = \sum_{X \subseteq Tj \wedge Tj \in D} TU(Tj)$.

Definition 6. Prefix Utility of an itemset X in a Transaction

$PrefixUtil_D(X,T) = \sum_{i \in PrefixSet(X,T)} u(i,T)$

Example $PrefixUtil(\{ac\},T_3) = u(a,T_3) + u(b,T_3) + u(c,T_3)$ =3+12+15=30

Definition 7. Prefix Utility of an itemset X in a Dataset D

Example PrefixUtil({ac})=$PrefixUtil(\{ac\},T_1)$+$PrefixUtil(\{ac\},T_2)$+

$PrefixUtil(\{ac\},T_3)$=8+36+30=74

Definition 8.Support number(sn) of itemset X is a number of transaction containing x in dataset

Property 1.For any Itemset X in ad dataset D, the following relationship holds :
$TWU_D(X) \geq PrefixUtil_D(X) \geq u_D(X)$

Clearly, $TWU_D(X) \geq u_D(X)$.

TWU satisfies the downward closure property, that is, for all $Y \subseteq X$, $TWU_D(Y) \geq TWU_D(X)$.Most of HUI mining methods uses the TWU values of the itemsets to prune the search space. That is, they find all the itemsets whose TWU is no less than the minimum utility threshold. Since $TWU_D(X)$ is an overestimate of $u_D(X)$, The procedure does not miss any high utility itemset.

Property 2. Anti-monotone Propoerty
It says that support for an itemset never exceed the support for its subsets.
For example If x & y are 2 itemsets such that $x \cap y = \emptyset$, then supp(x U y) < supp(x) and
It makes the search in the itesmet lattice easier by avoiding large number of useless cases.

Property 3. Apriori Property
Every subset of a frequent itesmet is also frequent

TID	**ITEMSET**	**tu**
1	(A, 2),(B,1),(C,3),(E,2)	35
2	(A, 1),(D,1),(F,2)	8
3	(B,2),(C,1),(D,3),(E,3)	21
4	(C, 3),(F,2)	20
5	(A, 2) ,(B, 1), (D, 1) ,(E,2), (F,2)	20
6	(A, 2),(B,3),(C,2)	31

Table 4 Dataset and External Utility Table

Item name	A	B	C	D	E	F
External Utility	5	3	6	1	2	1

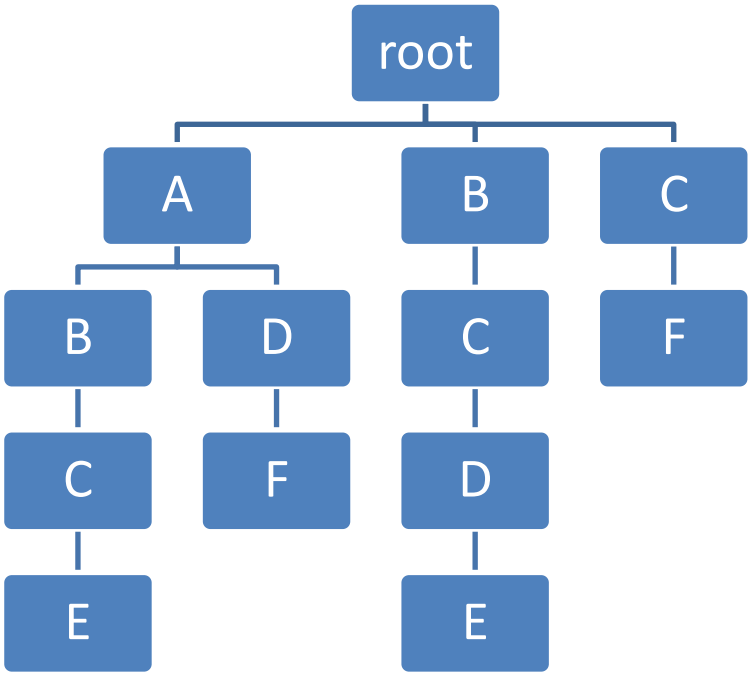


Figure 3 - 1Tree for Dataset

Tree is very useful structure to store data and process. The more data store the more memory consumes. Researcher mostly uses tree structure which is based on FP-Growth tree model. Every Research paper has slight difference in using and processing transaction and maintains tree structure.

Processing of Transaction is more important. From data stream perspective view ,data comes at very high speed, So processing must be fast and scan only ones when transaction arrive. The more task algorithm performs the more time it takes and gives more accuracy. So Algorithm has to balance between accuracy and speed.

There are many Research papers which suggest resource aware mining is very useful. As data stream contains unbounded and unlimited transaction, resource must be maintain rather than overloading and total hamper processing which lead to faulty results.

As discuss in Introduction Chapter, There are three types of window model. Among them sliding window model is best preference if recent transaction is more important. Using Sliding window model the data are mines based on recent data, and current mining result is more important than older.

DFS and BFS are two techniques which most data mining researcher are using. Both techniques are best in certain scenario.

Here we are using two structures as shown below.

1.] RankItem Node

2.] Transaction Node

Table 5 RankItem Node's parameter

RankItem Node		
1	**Name**	**Item Name**
2	**UnitList[win_Size]**	**Store Utility of item of window size**
3	**TotalUnit**	**Total utility of window**
4	**TotalTrans**	**Total Transaction in window that contain item**
5	**ExternalUtility**	**External Utility of item**
6	**TotalProfit**	**Total Profit generated by item**
7	**Hot**	**Shows that item is recent or not**

Table 6 Transaction Node's Parameter

Transaction Node		
1	**TransNum**	**Transaction Number**
2	**ItemList**	**Contain Item list of transaction**
3	**TotalItem**	**Total Item transaction contain**
4	**TransUtility**	**Total Transaction Utility**

As shown above Data are store in above structure. Structure is very compact but powerful in nature as it contains information of item based on that mining is performed.

In the algorithm, the user provide only two input which are ‘k’ and ‘Win_Size’ . Here ‘K’ value defines top k high utility item user want. And ‘Win_Size’ define window size. In Window, Transactions are in batch wise. Each batch contains certain number of transaction.

Below is Flow chart of Algorithm.

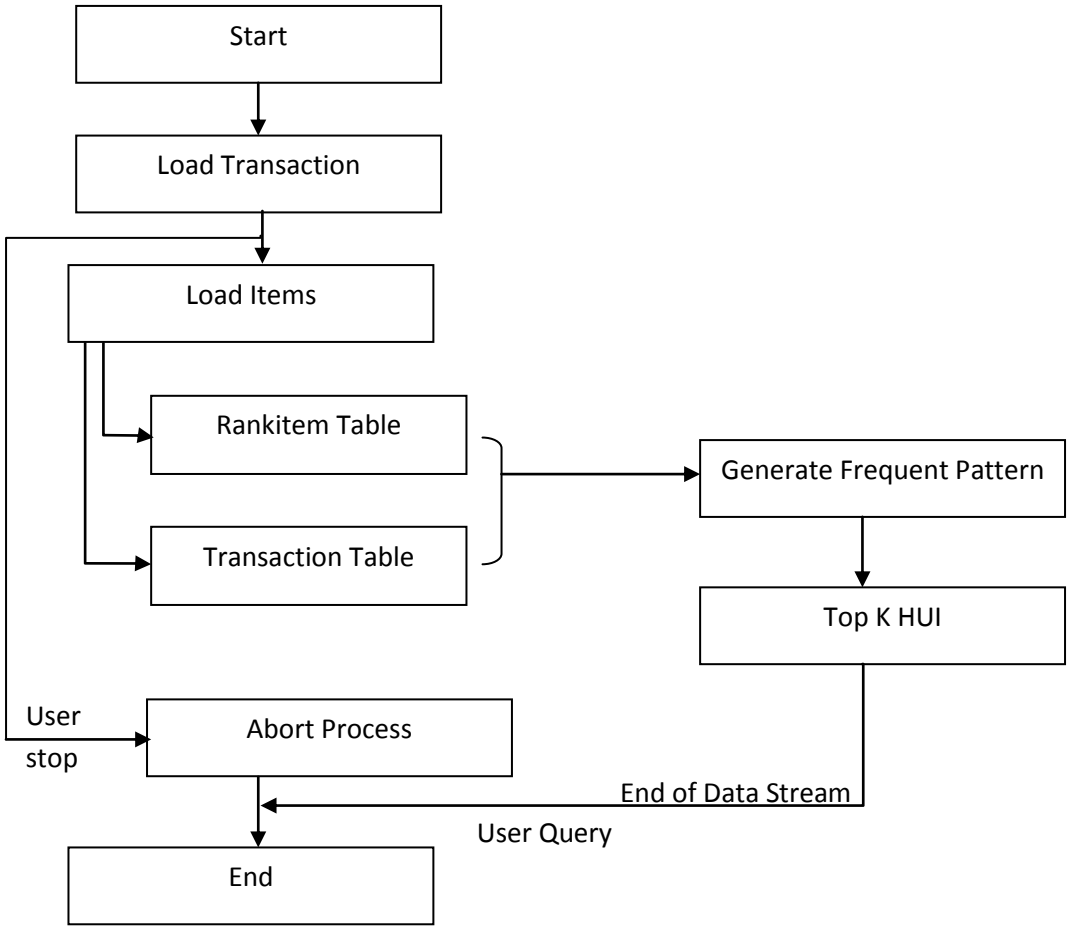


Figure 3- 2 Flowchart of Methodology

As seen in Flowchart data comes in continuously. Here we are using sliding window model, we first load data in transaction node List. Also simultaneously add item information in rankitem node List. Whenever user requires result, the mining result shown to user and generated after each batch, it means after each batch mining is performed. Result will show to user only when user perform query or at the end of Data Stream. The reason is that producing result to screen takes more time and consumes memory and it is overload to algorithm. If user doesn't want the result right now why should display to screen.

The process is very simple yet effective. Here we are not using tree structure, because this algorithm is based on recent item and item which generate high profit is most valuable also our structure effective track item which generate higher profit in recent window. Here Pruning is simultaneously done when new transactions arrive.

Itemset which generate higher profit are sorted in Rankitem List .Based on that potential itemsets are generated.

Algorithm 1. Loading Transaction and data into Structure

Input: Win_size,K, Transaction T,Batch Bi,
Output: Top-K HUIs

```
1.  TxNum← Tnum % Win_Size
2.  For every T ∈ Bi do
3.      Transaction[TxNum].TransNum = TxNum;
4.      For every Item I ∈ T do
5.          Transaction[TxNum].ItemList.Add(I);
6.          Transaction[TxNum].TotalItem++;
7.          IF Item I ∈ RankItem List then
8.              RankItem. UnitList [TxNum]=I.InternalUtility
9.              RankItem. ExternalUtility =I. ExternalUtility
10.             RankItem. Hot =True
11.         Else
12.             RankItem.Name=I.Name
13.             RankItem. UnitList [TxNum]=I.InternalUtility
14.             RankItem. ExternalUtility =I. ExternalUtility
15.             RankItem. Hot =True
16.         If RankItem.totalTrans != Win_Size
17.             RankItem.totalTrans++;
18.     Utility= I. ExternalUtility * I.InternalUtility;
19.     Transaction[TxNum].utility +=Utility;
20.     If it is not First Slinding window then
21.         For every rankItem node I in List do
22.             If rankItem node I is not hot then
23.                 If rankItem totalTrans != 0 then
24.                     RankItem.totalTrans--;
25.                 Else
26.                     Remove RankItem;
27.             Else
28.                 RankItem.Hot =False
29.         For every New Batch do
30.             Call Algorithm 2 Mining
31.             Return Top K HUI
```

In the First Algorithm, we just add data to our structure. Here RankItem list is class structure which has 7 fields and utility list field is of window size. Each item information store in rankItem class structure. So there are many item in data stream, so it create array of RankItem class.

Here TxNum is transaction number that cannot greater than window size. Each Transaction is loaded into Transaction Class structure than added to list of transaction.

Here each item have internal and external utility. Both utility are store in RankItem class structure which is in the RankItem List

Here we store and maintain information of recent window. When Window moves or new batch arrives, RankItem node in list also update and same for transaction list. Advantages is that it remain compact in size.

In algorithm 1, for each transaction we add transaction number to Transaction node. For every item in transaction we added into transaction's itemlist field. If item found in RankItem list then we update information like its internal utility and total transaction value also hot field which says that whether it is recently arrives or not.

After First window, for every transaction we check for the item which is not hot.For that item we decrement the totalTransaction field of RankItem node .This way keep track of recent item. For each item whose value is true for hot field, it is updated to false.

Mining is perform after every batch and after first window complete.

Algorithm 2 is mining where actual work is done. In this algorithm, for every Rankitem in List update RankItem ToalUnit by summing up all the unit from UnitList field. After that RankItem.TotalProfit calculated. Here we min_supp taken as avg of all totalprofit by all item. AvgTotalUnit is also used to filter data items. Potential is list of RankItem node which generate higher profit and sorted descending order according to total profit.

After that HuiList is generated by taking each item at a time and adding second highest and possible all variation. Also side by side we calculated utility also of itemset. It generate very high Itemset and their profit. After that it check for min_Supp. If it is greater than min_Supp then added to topKHui List and after that sorting is done based on profit.

Algorithm 2. Mining

Input : Batch Bi

Output : Top-K HUIs

1. **If** it is not First Slinding window **then**
2. **For** every New Batch **do**
3. topKHui=∅
4. **For** every RankItem node in List **do**
5. RankItem.TotalUnit=∑ RankItem.UnitList
6. RankItem.TotalProfit=RankItem.TotalTransaction * RankItem.totalUnit * RankItem.ExternalUtility
7. Min_Supp=∑ Rankitem.ToalProfit / Total RankItem node in List
8. AvgTotalUnit=∑ Rankitem.ToalUnit / Total RankItem node in List
9. Potential=List all Rankitem node where RankItem.ToalProfit >Min_Supp and RankItem.TotalUnit >AvgTotalUnit
10. Potential = sorting all rankitem Node by decending order of Total Profit
11. HUI List = generate a set of top k HUIs by combination taking each item at ones and adding second highest profitable item and so on. And calculate utility also simultaneously
12. **For** every Itemset in Hui List **do**
13. **If** utility > Min_Supp **then**
14. Add to topKHui List
15. Sort by decending order of their profit
16. **If** K > topKHui list **then**
17. Display all Itemset with Their utility profit.
18. **Else**
19. Display top K item and prune all itemsets
20. **Return topkHUI**

If k value is higher than topKHui List then print all itemset with their utility profit otherwise display only k itemset and prune all itemset form topkItem list.

Chapter 4
Results and Discussions

In this chapter, The proposed method is evaluated. All the algorithm are implemented in C#. The experiments are conducted on Intel core i3 2.53 GHz computer with 3GB RAM.

1. About C#

C# is a programming language. It is developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006).It encompassing strong typing, imperative, declarative, functional, generic,object-oriented (class-based), and component-oriented programming disciplines.

The C# language is intended to be a simple, modern, general-purpose, object-oriented programming language. The most recent version is C# 5.0

Its features are as below

- Source code portability
- suitable for writing applications for both hosted and embedded systems,
- economical with regard to memory and processing power requirements,
- provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection.
- suitable for deployment in distributed environments.
- C# supports Data Encapsulation, inheritance,polymorphism, interfaces.
- C# has been based according to the current trend and is very powerful and simple for building interoperable, scable, robust applications.
- C# includes built in support to turn any component into a web service that can be invoked over the internet from any application runing on any platform.
- C# includes native support for the COM and windows based applications.
- SCALABLE AND UPDATEABLE

2. Datasets and Performance measure

For experiment, four datasets are used. They are chess, mushroom, connect4 and T10I4D100K.First three datasets were prepared by Roberto Bayardo from the UCI datasets. The Last dataset is the IBM Synthetic dataset T10I4D100K,where the numbers after T, I, and D represent the average transaction size, average size of maximal potential frequent patterns and the number of transaction, respectively. In all the dataset, External utility of item is not provided and also quantity of each item in transaction as well.Therefore External utility of item is generated using log normal distribution and quantity of item in transaction is generated randomly between 1 and 10.

Table 7 Dataset and Its Information

	Transaction	**Items**	**Max object in Trans.**
Chess	3196	75	37
mushroom	8124	119	23
connect4	67557	129	43
T10I4D100K	100000	870	29

As the K size and winSize Varies runtime and performance also change. K value defines maximum number of top High utility pattern. For small value of k, Number of itemset store and maintain in top K HUI list are very less and therefore less memory consumption. Less memory consumption improves performance.

WinSize is also important. For high value of WinSize, High utility itemset is having more potential to generate more profit. For small value of WinSize, it gives best result based on current window size transaction only.

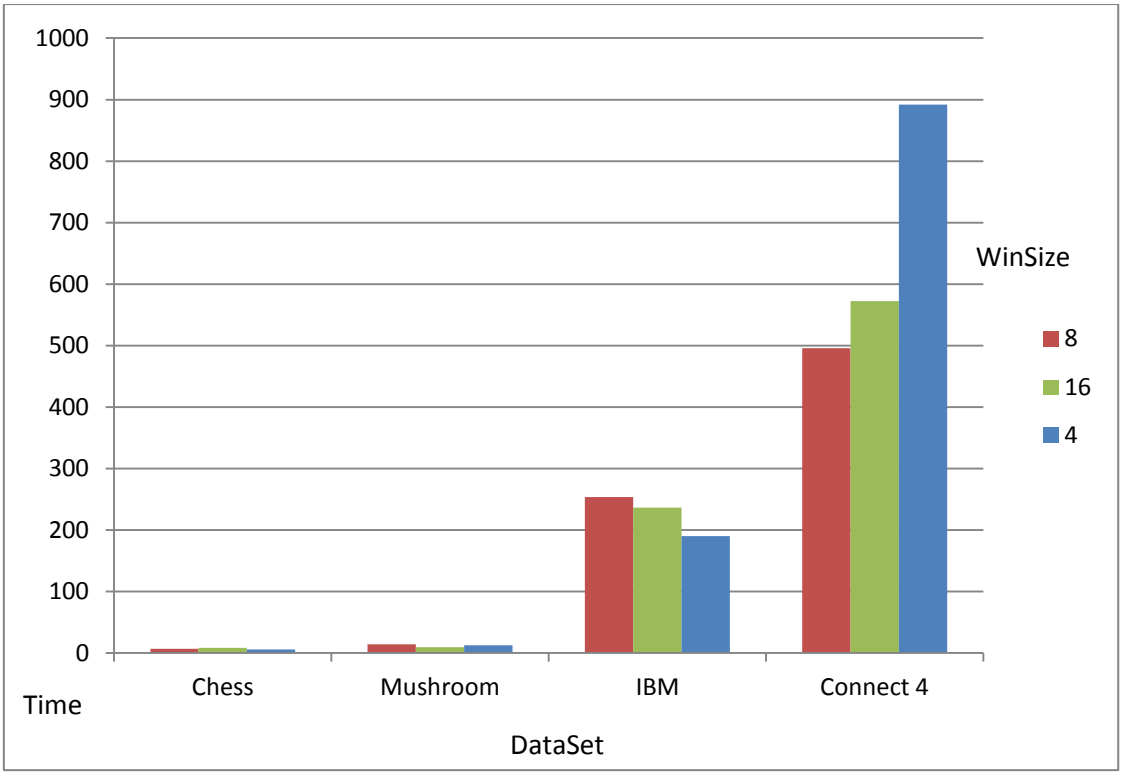


Figure 4- 1 Window Size effect on Runtime

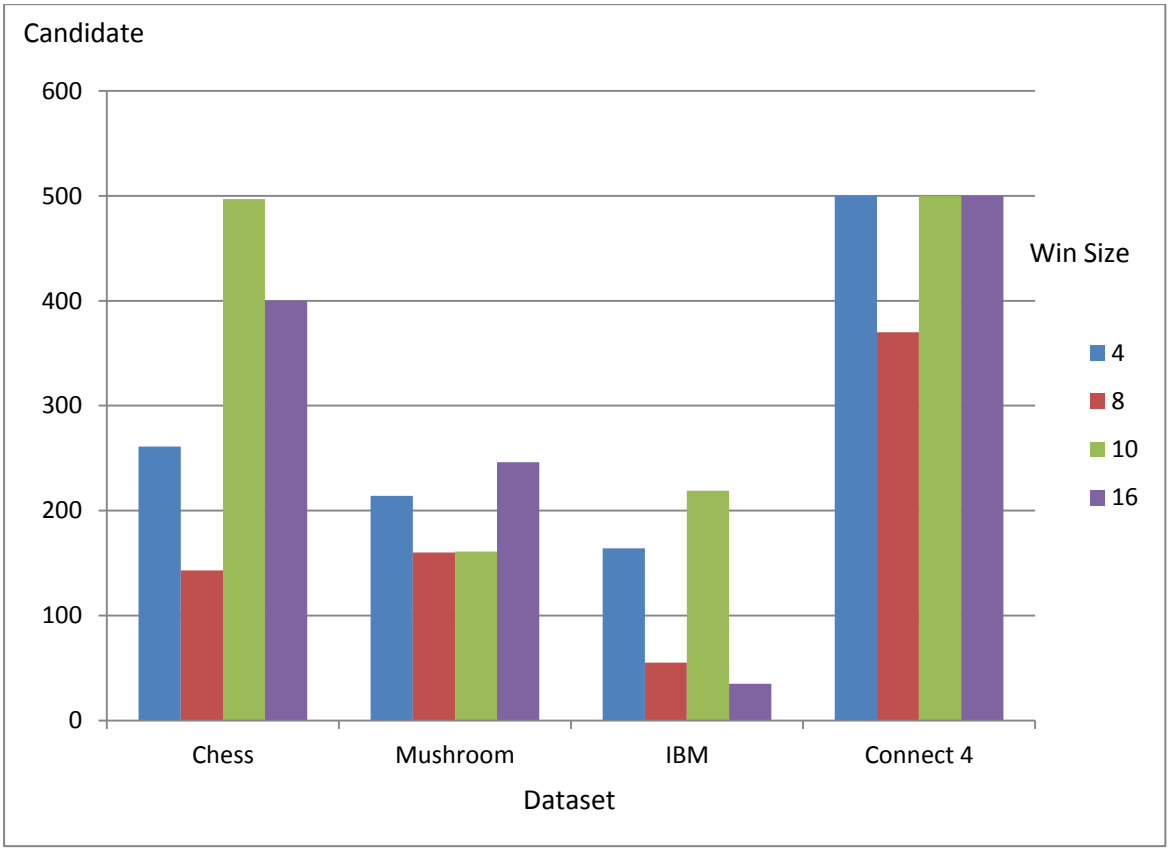


Figure 4- 2 Window Size effect on Candidate Itemset Generation

As seen in above chart, Change in winSize value affects timing. Timing is also affected by number of items in RankItem list and potential Items. The more number of Potential Item the more candidate itemset generated and takes more time.

Free up RankItem node which have 0 Total transaction values reduce memory and improves performance. In this algorithm Mining is done at every batch. This is also affecting performance and timing.

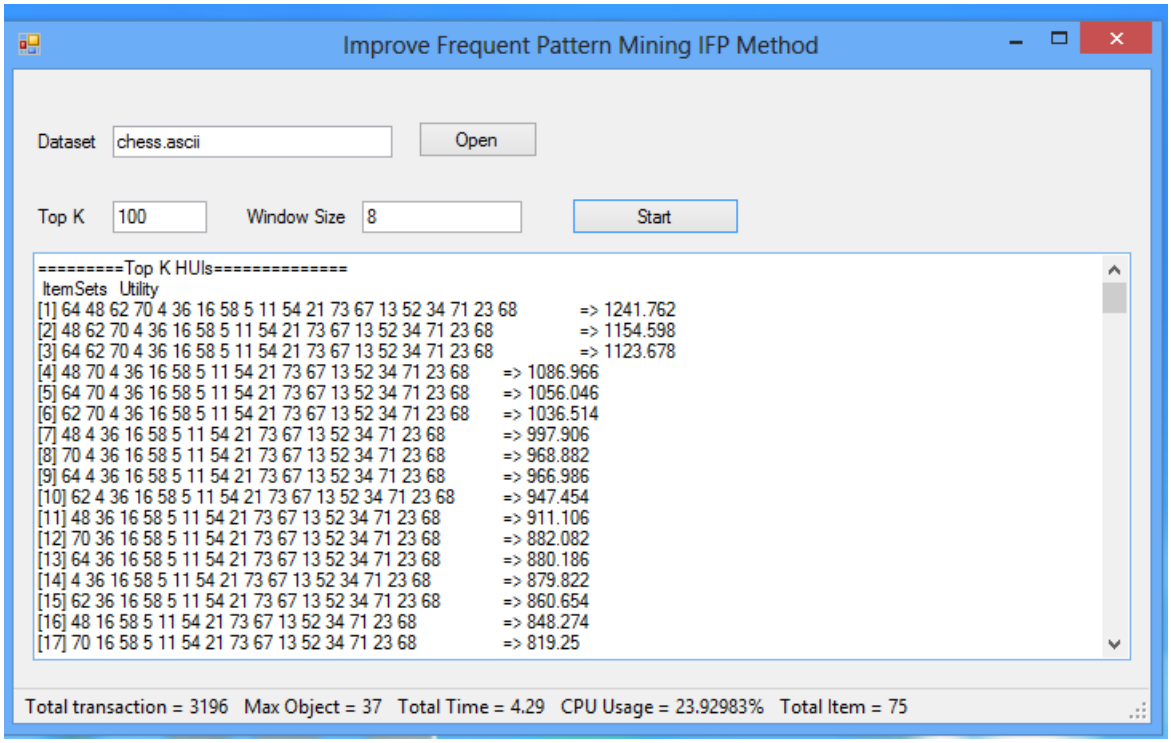


Figure 4- 3 User Interface

This is the interface for IFP algorithm. In this dataset first need to select by opening and locating path of repository. After that Top K value which is number of HUI require is provide and window size value also.

Interface Shows every detail of dataset as shown in status bar of interface.Details include total number of transaction, Maximum object per transaction , Total time taken by algorithm, Cpu usage taken by algorithm, Total item contain in dataset.

Chapter 5

Conclusion and Future Scope

Our research focuses on Designing computation and memory efficient algorithms to provide approximate results in high accuracy and confidence and developing system support help to mine useful information from data streams. Some specific research problems are identified. Result is as expected also can be improved by updating support value frequently based on window maximum and minimum value of itemset. Potential itemset generation is based on high value of frequent item. The itemset is for current window only so it can be improved by updating and maintain older itemset and prune based on time fading model. Meanwhile, the new techniques and algorithms we have developed for frequent itemset mining on streaming data are presented, and some preliminary ideas of our on-going work are discussed. Currently, we continue working on some of these problems.

Chapter 6
References

Books

1. Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
2. Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems).* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Research Papers

1. Morteza Zihayat, Aijun An, Mining top-k high utility patterns over data streams, Information Sciences, Volume 285, 20 November 2014, Pages 138-161, ISSN 0020-0255
2. Doug Burdick, Manuel Calimlim and Johannes Gehrke, MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases. In *Proceedings of the 17th International Conference on Data Engineering*. Heidelberg, Germany, April 2001.
3. Fatemeh Nori, Mahmood Deypir, and Mohamad Hadi Sadreddini. 2013. A sliding window based algorithm for frequent closed itemset mining over data streams. *J. Syst. Softw.* 86, 3 (March 2013), 615-623. DOI=10.1016/j.jss.2012.10.011
4. Gangin Lee, Unil Yun, Keun Ho Ryu, Sliding window based weighted maximal frequent pattern mining over data streams, Expert Systems with Applications, Volume 41, Issue 2, 1 February 2014, Pages 694-708, ISSN 0957-4174
5. Bai-En Shie, Philip S. Yu, Vincent S. Tseng, Efficient algorithms for mining maximal high utility itemsets from data streams with different models, Expert Systems with Applications, Volume 39, Issue 17, 1 December 2012, Pages 12947-12960, ISSN 0957-4174,
6. Unil Yun, Gangin Lee, and Keun Ho Ryu. 2014. Mining maximal frequent patterns by considering weight conditions over data streams. *Know.-Based Syst.* 55 (January 2014), 49-65. DOI=10.1016/j.knosys.2013.10.011

7. Mahmood Deypir, Mohammad Hadi Sadreddini, and Sattar Hashemi. 2012. Towards a variable size sliding window model for frequent itemset mining over data streams. *Comput. Ind. Eng.* 63, 1 (August 2012), 161-172. DOI=10.1016/j.cie.2012.02.008
8. Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed, Byeong-Soo Jeong, and Young-Koo Lee. 2008. Efficient frequent pattern mining over data streams. In Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08). ACM, New York, NY, USA, 1447-1448.
9. Pauray S. M. Tsai. 2009. Mining frequent itemsets in data streams using the weighted sliding window model. Expert Syst. Appl. 36, 9 (November 2009), 11617-11625.
10. Hua-Fu Li, Suh-Yin Lee, Mining frequent itemsets over data streams using efficient window sliding techniques, Expert Systems with Applications, Volume 36, Issue 2, Part 1, March 2009, Pages 1466-1477, ISSN 0957-4174
11. Younghee Kim, Wonyoung Kim and Ungmo Kim, "Mining Frequent Itemsets with Normalized Weight in Continuous Data Streams," Journal of Information Processing Systems, vol. 6, no. 1, pp. 79~90, 2010
12. Jing Guo, Peng Zhang, Jianlong Tan, and Li Guo. 2011. Mining frequent patterns across multiple data streams. In Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM '11), Bettina Berendt, Arjen de Vries, Wenfei Fan, Craig Macdonald, Iadh Ounis, and Ian Ruthven (Eds.).
13. Anushree Goutam Ringne, Deeksha Sood, and Durga Toshniwal, "Compression and Privacy Preservation of Data Streams using Moments," International Journal of Machine Learning and Computing vol. 1, no. 5, pp.473-478, 2011.
14. Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. 2003. Mining concept-drifting data streams using ensemble classifiers. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03).
15. Richard M. Karp, Scott Shenker, and Christos H. Papadimitriou. 2003. A simple algorithm for finding frequent elements in streams and bags. ACM Trans. Database Syst. 28, 1 (March 2003), 51-55.
16. Motwani, R; Manku, G.S (2002). "Approximate frequency counts over data streams". VLDB '02 Proceedings of the 28th international conference on Very

Large Data Bases: 346–357.

17. Boutsis, I.; Kalogeraki, V., "Resource management using pattern-based prediction to address bursty data streams," Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), 2013 IEEE 16th International Symposium on , vol., no., pp.1,8, 19-21 June 2013
18. HebaTallah Mohamed Nabil, Ahmed Sharaf Eldin and Mohamed Abd El-Fattah Belal, "Mining Frequent Itemsets from Online Data Streams: Comparative Study" International Journal of Advanced Computer Science and Applications(IJACSA), 4(7), 2013
19. Dr. S. Vijayarani, Ms. P. Sathya "An Efficient Algorithm for Mining Frequent Items in Data Streams",in International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 3, May 2013

Websites

1. Data Mining: What is Data Mining
   http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm
2. Mafia algorithm information and source
   http://himalaya-tools.sourceforge.net/Mafia/#intro
3. Data Mining Concept
   http://www.hypertextbookshop.com/dataminingbook/public_version/contents/chapters/chapter002/section004/blue/page002.html

Chapter 7
Appendix

Publish Paper

1. Himanshu Shah,Navneet Kaur, "Improve Frequent Pattern Mining in Data Stream", IMPACT: International Journal Research in Engineering & Technology (IMPACT: IJRET), May-2014

Abbreviations

DBMS =Database management system

DSMS= Data stream Management system

U(x) = Utility of item in dataset

TU = Transaction Utility

TWU=Transaction Weighted Utility

HUI = High Utility Itemset