International Journal of Computer Applications (0975 – 8887)
Volume 107 – No. 16, December 2014

Survey on Constrained based Data Stream Mining

Lini Susan Kurien
Dept of Computer Science,
College of Engineering,
Poonjar.

Sreekumar K
Dept of Computer Science,
College of Engineering,
Poonjar.

Minu KK
Dept of Mathematics,
College of Engineering,
Poonjar

ABSTRACT

In most of the real time applications data may arrive as continuous ordered sequence of items, called Data streams. The main challenge in dealing with the Data Stream is its voluminous, complex and dynamically arriving stream of data. There are certain techniques to deal with data streams, in particular, finding the frequent or sequential patterns that occur repeatedly. These results retrieve huge number of patterns, which are hard to analyze and use them, also difficult to store these results and its intermediate results. The traditional pattern mining techniques fail to give the relevant details to the user. In order to obtain that, some constraint based mining techniques, which acts as a filter to the large result set retrieved from traditional pattern mining techniques. This paper investigates different data stream mining techniques and constrained based stream mining techniques from which the user gets the required information from the data stream.

General Terms

Data Stream, Pattern.

Keywords

Data stream mining, Frequent Pattern Mining, Constraints, Sequential Pattern Mining.

1. INTRODUCTION

In many real time applications huge amount of data accumulated as continuous sequence of items, in order to handle those items different data mining techniques are there. The goal of data mining is to analyze the incoming data and extract the useful information for the user from the large data set.

Frequent itemset mining algorithms retrieve frequent patterns that arriving frequently more than a threshold value, such algorithms give a number of patterns. From those huge numbers of patterns it is hard to analyze and chose the useful patterns/information for the user. In order to reduce the number of results, if increased the threshold value/support it may not retrieve any result or if decreased the threshold value users may get huge number of irrelevant results. So the requirement is a balanced number of results as well as useful results.

There are some techniques introduced to reduce the number of irrelevant patterns such as constraint based algorithms. Constraints play an important role in reducing the number of patterns and to increase the efficiency of the pattern mining. In real time applications data may not available prior instead it may appear as continuous stream of data at high speed. There are several techniques to handle data streams and pattern mining in those streams. So it is important to integrate constraint mining and pattern mining in order to get the user relevant information. In most of the constraint based algorithms are post-processing approaches; first retrieve all the patterns using unconstraint pattern mining algorithms and then apply constraints on to it. This paper deals with study of stream mining and constraint based data stream mining techniques.

2. ISSUES IN DATA STREAM MINING [15][13]

i. Since the data arrives as continuous streams as time progresses how to efficiently find and store the frequent itemsets.

ii. As the data arrival rate is very fast computation and communication capability is a major challenge in stream mining.

iii. To store all the incoming data unbounded memory is required.

iv. Data model and query semantics must be ordered and time based operations.

v. Most of the real time scenarios summary of the stream is stored not the complete stream, which leads to incorrect results when queries are executed and backtracking of results is also not feasible.

3. DATA STREAM MINING APPROACHES

Data stream can be classified as offline stream in which regular bulk arrivals and online stream in which real time data is updated one by one as time progresses.

Frequent Item set mining or Frequent Pattern Mining algorithms retrieve patterns that may appear more than a threshold value. [6]

3.1 Counter based algorithms[6]

3.1.1 Majority Algorithm

It selects each item and retrieves the majority vote, if any, and then the algorithm stores that item. The algorithm verifies by taking the number of occurrences of items stored.

3.1.2 Frequent Algorithm

It stores k-1 (item, counter) pairs ensure that the count associated with each item on termination is at most εn below the true value. It finds the items which occur more than the given fraction of time.

3.1.3 Lossy Algorithm

It stores an item and its counts. A ‘delta’ value is calculated such that the difference between the upper bound and the lower bound. If the delta value of an item is increased, all its count will be decreased by 1, and items with zero count will be deleted.

3.1.4 Space Saving Algorithm

It is a combination of Frequent algorithm and Lossy algorithm. It stores k items as a pair of (item-name, count). Algorithm initializes first distinct k items and their counts. If the next item matches with the k items stored then the count is incremented else the smallest count has its item value replaced with the new item, and the count is incremented.

3.2 Quantile Algorithm[6]

Find ϕ-quantiles of a sequence of item drawn from a totally ordered domain is to find an item i such that it is the smallest item which dominates ϕn- items from the input.

1.2.1 GK Algorithm

It stores tuples containing an item from the input, a frequency count g and a delta value.

1.2.2 Q Digest

A dyadic range and a count. A dyadic range is a range whose length is a power of two, and which ends at a multiple of its own length.

1.2.3 Count Sketch

This algorithm uses a hash function and hash table used for estimating frequent number of occurrences and storing the estimated values. The count is estimated for each element and it maintains a heap of top count k elements. The efficiency of the algorithm is improved by using the same technique is used for every update, which affects only a small subset instead of the whole data stored.

1.2.4 Count Min Sketch

It is similar to Count Sketch. An array of d × w counters is maintained, and pair wise independent hash functions hj which map e items onto [w] for each row and each update is mapped onto d entries in the array of d x w, each of which is incremented.

1.2.5 Finding Frequent Items using Hierarchy

The algorithm allows to model the removal of items as an update with negative weight.

1.2.6 Finding Frequent Items using Group Hierarchy

The idea of “group testing” randomly divides the input into buckets, so that at most one frequent item will be there in each group. In each bucket, the items are divided into groups, so that the “weight” indicates the identity of the frequent item in each group. This can be treated as an extension of the Count-Min sketch.

Another approach is Frequent Itemset mining on the MapReduce platform which focuses on speed and mining large databases using hybrid approaches. [7]

Clustering is another important technique in data mining, which partition the data set into group of similar objects. StreamKM++ algorithm constructs clustering of streams with small amount of memory and time.[17]

4. CONSTRAINED MINING

The usual mining algorithms retrieve number of results which may not be relevant for the user. By using constraints the data miner can optimize the results and get the useful results by applying some knowledge to the mining process.

4.1 Different types of constraints [2][3]

Type	Description	Example
Item Constraint	It specifies about the particular individual or group of item that should or should not be present.	A diary company may be interested in patterns containing only dairy products, when it mines transactions in a grocery store
Length Constraint	It specifies the requirement of length of the pattern, where it can be number of occurrences or number of patterns.	When mining classification rules for documents, a user may be interested in only frequent patterns with at least 5 keywords, a typical length
Super pattern Constraint	To find the pattern that contains a particular set of patterns as sub patterns.	An analyst may like to find the sequential patterns that contain first buying a PC and then a digital camera
Aggregate Constraint	The constraint on an aggregate (sum, avg, max, min, standard deviation) of items in a pattern.	A marketing analyst may like sequential patterns where the average price of all the item in each is over $100
Regular expression constraint	The constraint specified as a regular expression over the set of items using the established set of regular expression operators such as disjunction and Kleene closure.	To find sequential patterns about a Web click stream starting from Yahoo’s home page and reaching hotels in New York city, one may use the regular expression constraint *Travel(New York|New York city)(Hotels|Hotels and Motels|Lodging)*
Duration constraint	A duration constraint on an event specifies the time interval over which the event acts. It is defined only in sequence databases where each transaction in every sequence has a time-stamp.	If you press the button of the handset for less than 15 seconds, it connects to the local operator. If you press the button for any duration lasting between 15 to 30 seconds, it connects to the international operator. If you keep the button pressed for more than 30 seconds, then on releasing it would produce the dial tone.
Gap constraint[18]	Gap constraint is specified by a positive integer g. Given a sequence S = e1e2…en and an occurrence o_s = i1i1…im of a subsequence S' if i_{k+1}-i_k ≤ g+1 ∀k ϵ {1..m-1}, then the occurrences o_s fulfills the gap constraint. Otherwise o_s fails the gap constraint.	Comparing the first and last books from the Bible, the subsequences "having horns", "faces worship", "stones price", and "ornaments price" appear multiple times in sentences in the Book of Revelation, but never in the Book of Genesis. (The gap between the two words of each pair is ≤6 non trivial words.) Such pairs might be seen as a finger print associated with the Book of Revelation and may be of interest to Biblical scholars.

4.2 Constraint Properties

Property	Definition	Example/ Description
Anti-monotonicity	A constraint is anti-monotonic if and only if, whenever an itemset X violates it, so does any superset of X. [16]	The best known and simple example of an AM constraint is the minimum support threshold. If the support of an itemset is great or equal to the threshold, then itemset is frequent. If an itemset is less than threshold, so any of the supersests also will be infrequent.
Monotonicity	A constraint is monotonic if and only if, whenever an itemset X satisfies it, so does any superset of X. [16]	If an itemset satisfies the constraint, all supersets will also satisfy it. However, if an itemset violates the constraint, a superset can satisfy it, by introducing the missing items.[16]
Succinctness	A constraint is succinct if it is possible to enumerate all possible patterns, based only on items from the alphabet I. [16]	A simple example is the value constraint C(X) = X:price _ e100. We can select from the alphabet all items X1 with price _ e100, and the itemsets that satisfy C are exactly only those in the strict powerset of X1. This is a succinct anti-monotonic constraint (SAM), as supersets of itemsets with some item with price > e100 will never satisfy it.
Prefix Monotonicity	A constraint is prefix-monotone if there is an order of items that allows the algorithms to treat it as anti-monotonic or monotonic. [16]	Prefix monotonicity reflects a basic property of communicating systems: assume we have observed a finite sequence of output messages for a corresponding finite sequence of input messages. Then if we observe additional input (thus the old input sequence is a prefix of the extended one) we may just observe additional output (thus the old output sequence is a prefix of the extended one). Prefix monotonicity provides a notion of causality between input and output. It reflects the stepwise consumption of input and production of output and guarantees the existence of least fixed points, which is mandatory for giving meaning to communication feedback loops
Mixed Monotonicity	A constraint is mixed-monotone if it can be considered both anti-monotonic and monotonic, at the same time, for different groups of possible values (positive and negative). [16]	let the set of items I be divided into two disjoint groups based on their monotonicity relating to a constraint C: let IAM be the set of anti-monotonic items, and IM, the set of monotonic items. Then, a constraint is mixed monotone if, for any itemset X: (a) whenever X satis_es C, all supersets of X formed by adding items from the IM group, also satis_es C; and (b) whenever X violates C, all supersets of X formed by adding items from the IAM group, also violates C
Convertible Constraints	A constraint C is convertible anti-monotone provided there is an order R on items such that whenever an itemset S satisfies C, so does any prefix of S. It is convertible monotone provided there is an order R on items such that whenever an itemset S violates C, so does any prefix of S. A constraint is convertible whenever it is convertible anti-monotone or monotone. [5]	For example, *avg*(*S*)*θv*, *median*(*S*)*θv*, *sum*(*S*)*θv* (*S* can contain items of arbitrary values, $\theta \in \{>, <, \leq, \geq\}$ and *v* is a real number.)
Tougher Constraints	Given an itemsetX withIXI>2, a constraint is loose anti-monotone (denoted C_{LAM}) if: $C_{LAM}(X))$ $\exists i \in X : C_{LAM}(X \setminus \{i\})$	Namely *Loose Anti-monotone* constraints, they are a super-class of convertible anti-monotone constraints (e.g. constraints on average or median) and that they model tougher constraints (e.g. constraints on variance or standard deviation). [9]

4.3 Constraint based Approaches

i. Monotonicity is the most commonly used constraint property in many applications. With the help of monotonicity generating classification trees that satisfies the constraint property, both when training data which satisfies monotone and when it is not.[1]

ii. Constraint frequent pattern mining with a patter growth view finds all frequent itemset that satisfy the constraint and then the pattern growth mining method generates and test only a few among them. [2]

iii. Another classical framework for mining sequential pattern with prefix monotone constraints. This framework is extendible to aggregate constraints also.[3]

iv. SQL style constraints can be used for mining process to find the user specified patterns. UFPS algorithm retrieves frequent patterns that satisfy succinct constraints from uncertain data. [4]

v. Convertible constraints cannot be pushed into algorithms like Apriori. Algorithms like FLC^A can be used for pushing convertible anti-monotone constraints in frequent itemset mining and FLC^M can used to push convertible monotone constraint in frequent itemset mining. [5]

vi. Global patterns can be mined and model by combing several local patterns. ECL^iPS^e is a mathematical programming tool that can be used to search local patterns and combine them.[8]

vii. Algorithms like ExAMiner^{LAM} aims at solving problems like conjunction of anti-monotonicity and monotonicity which reduces the problem dimensions at all levels of a level wise Apriori like computation. [9]

viii. Another constraint approach, initializes the data structure for pruning and computes the closed sets, then updates the data structure and then prunes the current closed set which cannot satisfy the constraint. [10]

ix. Using constraint Frequent Pattern Growth (CFG) method convertible constraints can be easily pushed into frequent pattern mining methods for obtaining optimal results. [12]

x. Another approach is initially run an unconstraint algorithm, then with the result of the unconstraint run apply the constraints/ rules and get the required result. This approach gives a faster response but a truly interactive KDD process. [14]

xi. With CoPT4Streams algorithm an efficient constraint based mining can be performed, in which dynamically pushes constraint into pattern-tree with constraint properties. But this a post-processing approach, it first finds all the pattern then applies the constraint to the patterns found.[16]

5. CONCLUSION

This paper deals with the study of different data stream mining techniques and found that most of the techniques retrieve a huge number of results which is hard to analyze and get the relevant information for the user. In order to filter results/patterns obtained from those techniques, constraint based mining approach, in which user can give the constraints and get the required and precise information from the data stream. This paper also describes set strategies for constrained based stream mining. And also examine different types of constraints and constraint properties which can be pushed to data stream mining algorithms to retrieve more optimized and relevant pattern/dataset from the real time continuously arriving data set.

Most of the constraint based mining algorithms proposed are post-processing approaches, in which finds the possible patterns from the given data set and then applies the constraints to the result set and obtain the required patterns. For dynamically arriving data streams, to find the relevant patterns, every time need to do the process from the beginning as find the patterns from the available data set, apply the constraints and then get the relevant patterns for the user. As a future scope, store the relevant patterns once it is generated after applying the constraints and then for every data set/ stream check with the stored useful patterns that whether the newly arrived data set matches with the patterns stored or not.

6. REFERENCES

[1] R.Pothrast, AJ Feelders, “Classification trees for problems with monotonicity constraints”,

[2] Jian Pei, JiaweiHan,“Constraint Frequent Pattern Mining: A pattern- growth view”,

[3] Jian Pei, JiaweiHai, Wei Wang, “Mining Sequential Patterns with Constraint in Large Database”,

[4] Carson Kai-Sang Leung, Dale A Brajczuk,“Efficient Algorithm for the Mining of Constrained Frequent Patterns from Uncertain Data”

[5] Jian Pei, Jiawei Han, Laks VS Lakshmanan “Pushing convertible constraints in Frequent Itemset Mining”, Data Mining and knowledge Discovery 8, 277-252,2004, 2004 Kluwer Academic Publishers.

[6] Graham Cormode, MariosHadjcelefterion, “Methods for finding Frequent Items in Data Streams”

[7] Sandy Moens, EminAksehirli, Bart Goethals, “Frequent Itemset Mining for Big Data”

[8] Mehdi Khiari, PariceBoizumault, BrumoCremilleux, “Local Constraint-Based Mining and set Constraint Programming for Pattern Discovery”

[9] Francesco Bonchi, Claudio Lucchese, “Pushing Tougher Constraint in Frequent Pattern Mining”

[10] TaneliMielikainen, “Intersecting Data to closed sets with constraints”

[11] HetalThakkar, BarzanMozafari, Carlo Zaniolo, “Continuous Post-Mining of Association rules in a data stream Management System ”

[12] Jian Pei, Jiawei Han, “Can we push more constraints into Frequent pattern mining”

[13] A Mala, F Ramesh Dhanaseelan, “Data Stream Mining algorithms- A Review of issues and existing approaches” IJCSE

[14] JochenHipp, Ulrich Guntzer, “Is Pushing constraints Deeply into the mining algorithms Really what we want - An alternative approach for association rule mining” Issues in data stream mining

[15] Andreia Silva and CláudiaAntunes “Pushing constraints into data streams”, BigMine '13 Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications

[16] Ms. S.RanjithaKumari, Dr. P.KrishnaKumari, S.Shylaja “Handling real time data sets using stream mining techniques”,IJCSMC, Vol. 3, Issue. 10, October 2014, pg.62 – 69

[17] Xiaonan Ji, James Bailey, Guozhu Dong “Mining Minimal Distinguishing Subsequence Patterns with Gap Constraints”

[18]