ORIGINAL ARTICLES

Mining Sequential Pattern With Synchronous And Asynchronous Periodic Time Stamp Using Hash Based Algorithm

1M. Parimala, 2S. Sathiyabama

1Senior Lecturer, Department of Computer Applications, M. Kumarasamy college of Engineering, Karur, Tamilnadu, India.
2Assistant Professor, Department of Computer Science, Thiruvalluvar Govt Arts and Science College, Rasipuram, Tamilnadu, India.

ABSTRACT

Sequential Pattern Mining (SPM) is considered as an interesting data mining problem conducted in research works. In SPM, the time-series pattern can be used as a tool for identifying the behavior of the patterns while mining sequential pattern types of data. Mining time series patterns in temporal dataset acts as an imperative function in data mining and knowledge detection mechanisms. Several authors have been focused on their research on mining time series pattern with synchronous and asynchronous periodic databases. But the downside of the researches is that due to the involvement of random noise and disruption, some periodic patterns were not recognized. So, to enhance the sequential pattern mining with both the synchronous and asynchronous periodic timestamp, in this work, we deploy a new technique termed hash based algorithm. The hash based algorithm finds all maximal complex patterns in a single step devoid of mining distinct event and various events patterns employing only one sequential dataset search. It introduces sequential pattern for varied periods and generate frequent pattern and candidate maximum pattern for an effective mining. An experimental evaluation is carried out with the datasets retrieved from the UCI repository used to evaluate the performance in terms of accuracy and scalability using hash-based algorithm to that of GSP for asynchronous periodic pattern mining for asynchronous periodic pattern mining.

Key words:

Introduction

Mining asynchronous periodic pattern in time series data find those patterns that occur frequently in some subsequences but their occurrences may be shifted with disturbance included in it which ranges to noise to disruption. Two parameters are used to specify the minimum number of occurrences that is required within each subsequences and the maximum disturbance between any two successive subsequences. Related research work involving mining sequential pattern in a noisy environment has been explored. Compatibility matrix is introduced to provide a probabilistic connection from observed values to the true values.

Periodical pattern analysis is the mining of dataset involving periodic patterns which aims to find all the recurring patterns in the time-series databases. Periodic pattern mining is following by sequential pattern mining which takes into consideration durations as a basis for partitioned sequences. The problem of mining periodic patterns is partitioned into three sections. They are 1) Mining full periodic patterns which consists of mining based on the cyclic behavior of a dataset conducted repeatedly. 2) Mining partial periodic patterns which includes mining a particular dataset which do not consider whole fields in it but considers only specific field. 3) Mining cyclic association rules, rules that associate a set of events that occur periodically.

Most of the full periodic pattern mining problems have been studied in signal analysis and statistics, or transformed into sequential pattern mining. The most useful and challenging part in sequential pattern mining focuses on mining of partial periodic patterns. The proposed work also concentrates on incremental mining which maintains the changes that occur in the discovered patterns over time as more number of items gets added into the database. In this paper, we propose a hash based algorithm for asynchronous periodic patterns from a series of symbol positions where an instance slot can enclose numerous events. The parameter min rep is employed here to identify the least amount of duplications desired for a suitable section of non-disrupted model occurrences.

The paper is arranged as follows. Section 2 briefly describes about the related works performed for sequential pattern mining with the help of researchers presented in the work hash-based algorithm. Section 3
provides with the problem description. Section 4 we present concepts related to asynchronous periodic pattern mining and a comparison is made for hash-based algorithm, PCD and GSP.

2 Literature Review:

The hash based algorithm is an extension of GSP. Pattern mining plays an essential role in data mining tasks. Based on these properties some of the tasks involved in data mining include frequent pattern, sequential pattern (Srikant, R. and R. Agrawal, 1996), inter transaction pattern (Tung, A.K.H., et al., 1999) and episode mining (Heikki Mannila, Hannu Toivonen, and A. Inkeri), etc. Periodic pattern mining is the problem that regards temporal regularity. For example in a transactional data, we may find that purchase pattern, Beer and Diaper, occurs at every Saturday night for 20 weeks continuously. This implies that the purchase of beer and diaper occurs periodically during Saturday night. The paper (Thodeti Srikanth, 2011) presents, devise and expansion of software for chronological pattern drawing out for asynchronous episodic patterns in chronological database. The discovery of pattern with periodicity has been studied in (Han, J., G. Dong, and Y. Yin, 1999; Ozden, B., et al, 1998). However, these activities considered only synchronous periodic patterns but did not perceive non-aligned Inclusion of patterns due to the intervention of noise in a random manner. Therefore, Ref (Yang, J., et al., 2000) extended the periodic pattern by introducing a concept from information theory to address noisy symbols.

Recent studies in periodic pattern mining, Ozden (1998) defined the problem of discovering cyclic association rules by way of finding cyclic relationships between items within transactions. In their research, the input data was a set of transactions, in which each transaction consisted of a set of items. In addition, each transaction included the involvement of execution time. By studying the interaction between association rules and time, the researchers applied three heuristics. They are cycle pruning, cycle skipping and cycle elimination to find cyclic association rules involved in transctional databases.

Han (1999) provided with different algorithms to efficiently mine partial periodic patterns, by considering interesting properties which were related to partial periodicity, such as the Apriori property and the max-sub pattern hit set property. In order to follow the restriction cyclic association rule, Han, used confidence metric to measure the significance of periodic pattern. The confidence involved in sequential pattern mining was defined as the number of count for a particular pattern over a given number of periods involved in the temporal database. In (Fahad Maqbool et al., 2006), the author propose a novel algorithm E-MAP (Efficient Mining of Asynchronous Periodic Patterns) for efficient mining of asynchronous periodic patterns in large temporal datasets.

Yang (2000) proposed the model to mine asynchronous periodic patterns that are significant using a subsequence of symbols which may contain a disturbance of length up to certain threshold. They propose three strategies involved to mine asynchronous periodic patterns. They are pruning based on distance factor, verification process based on single pattern and complex pattern. This model takes into consideration only sequences of symbols, and the longest subsequences involved in the process can only grab part of the system behavior. Most studies of Sequential pattern mining concentrate on categorical patterns (Jen-Wei Huang, et al., 2007).

3. Problem Domain of Partial Periodic Pattern Mining:

Periodic pattern mining involves the mining of periodic patterns that considers search related to recurring patterns in time-related data sets. Periodic pattern mining is applied over time-series data, which consists of sequences of or events which are measured regularly at constant time intervals ranging from hourly basis to monthly or yearly basis. The items to be analyzed for periodic pattern mining include data such as numerical or categorical data. Periodic pattern mining ranges from partial to full periodic mining.

Assume that a sequence of n time stamped datasets have been collected in a database. For each time instant i, let Di be a set of features derived from the dataset collected at that instant. The time series features are represented. The mining of frequent partial periodic sequential patterns in a time series is to find, possibly with some restriction, all the frequent patterns of the series for one period or a range of specified periods. For single-period pattern mining, it aims to find all partial periodic patterns for a given period p, support threshold minimal support and confidence threshold minimal confidence in time-series S. One approach is directly use the Apriori algorithm to the mining process after the sequence is divided into period segments.

The problem of finding frequent sequence is similar to finding the frequent itemsets in association rules mining. The Apriori property is used to prune the candidate of large sequences. The total number of scans in this algorithm is no more than the length of period p. Space needed in this method is 2F1-1 in the worst case, where F1 is the number of frequent 1-patterns. Another method is called max-subpattern hit set method. In this algorithm F1 is used to produce a Cmax the candidate (frequent) max-pattern. A subpattern of Cmax is hit set in a period segment Si of S if it is the maximal subpattern of Cmax in Si. The hit set, H, of a time series S is the set
of all hit subpatterns in Cmax in S. During the first scan, as in Apriori, the frequent 1-pattern F1 is generated. In the second scan, hit set of each period segment is generated and stored into the hit set buffer in a tree structure with its count. The frequent patterns are generated from the hit set by their counts. In this approach only 2 scans of the database are needed, the space needed at most is \( \min f m, 2 F_{1-1} \), where m is the total number of periods in S.

4. Asynchronous Periodic Pattern Mining:

Asynchronous partial periodic patterns in multi-event temporal database comprises of three parameters involved. They are min-rep (minimum repetition), global-rep (global repetition) and max-distortion (maximum distortion) are employed to segregate into valid patterns and the subsequence involved in it. The subsequence is in turn viewed as list consisting of valid segments. The valid segment involved in the subsequence is of at least min-rep and the distance between each piece of disturbance is assumed to be up to max dis. The number of repetitions in a sequence is the sum of the repetitions of its valid segments. A sequence is said to be trustworthy if the overall iteration of the pattern are higher than global rep value.

The working of mining Asynchronous Periodic Patterns in Multifaceted-Event (APPM-E) with partial periodic pattern mining for time-series database ranging from simplex to complex pattern is discussed and evaluated in this proposed work. Sequential Pattern Miner (SPM) discovers all valid segments for each specific-event. Multiple Pattern Miner (MPM) and complex pattern miner explores valid segments for multifaceted-event and patterns involving complex nature. In conclusion, all valid segments with respect to periodic pattern mining are combined to form an asynchronous sequence format by way of using asynchronous pattern miner.

The proposal provide procedural algorithm for mining asynchronous periodic pattern. The proposal work proceeds with a hash based validation mechanism which finds all relevant specific-event periodic patterns. In the same way multifaceted-events and asynchronous-events are evaluated using multifaceted event pattern validation and asynchronous pattern validation as shown in figure 1.

**Fig. 1: Architecture of APPM-E**

4.1 Generalized Sequential Pattern Mining Algorithm (GSMP):

GSPM Algorithm is used for sequence mining of datasets. The algorithm is mainly based on the apriori algorithm. This is also said to be level-wise algorithm. The algorithm is divided into two levels. The job of first level is to identify all the frequent item sets which in turn sum of all the frequent item sets in the database. The second level consists of removing the non-frequent item sets in the transactions involved. Finally the third level is the resultant modified transaction comprising of frequent item sets. The modified database obtained at the resultant of the third level is considered as input to the GSMP algorithm. The GSMP algorithm is presented in figure 2.

The working of GSMP Algorithm makes multiple passes over the database. The first pass consists of counting of all 1-sequence items. From the resultant value obtained as input to the next phase. The second pass consists of counting of all 2-sequence items. Using the 2-sequence items 3-sequence items are generated. In this similar manner the process is iteratively processed until no more frequent sequences are contained. The algorithm involves two steps. They are

- Generation of Candidate items : Given set of frequent sequences F(S-1), candidate sequences are generated by way of using eliminating process and the same process is repeated until there are no subsequences.
- Counting of Support items : Our proposal uses hash-based algorithm for efficient counting of support items.
4.2 Hash based Validation Mechanism (HVM):

In this section, we present a hash based validation scheme for mining asynchronous periodic pattern for large temporal databases. At first, a hash based validation mechanism is presented to determine all distinct periodic patterns.

Contrary to nearly all prior study on pattern mining, which presumes a parallel catalog outline, we employ perpendicular database design. Consider the straight up system for database \( VD \), where a time list is sustained for every event. By probing the difference of time lists, we develop two mining approaches for mining periodic sections of distinct events.

Potential Cycle Detection (PCD): A suitable pattern with time \( l \) valid entails that there subsists as a minimum \( \min rep \) matches. Consequently, we first employ an array \( \text{CheckSet}[l] \) to build up the tot ups for every time \( l \) (1 ≤ \( l \) ≤ \( L_{max} \)). If the \( \text{CheckSet}[l] \) is better than \( \min rep \), it is a possible cycle. Acquire event \( D \) for an instance. After scrutinizing the timeliest of incident \( C \), we obtain \( \text{CheckSet}(Ahmed, C.F., \textit{et al.}, 2009) = 3 \), \( \text{CheckSet}(Ding-An \ Chiang, 2009) = 8 \) and \( \text{CheckSet}(Ya-Han \ Hu, \textit{et al.}, 2009) = 4 \). With \( \min rep = 5 \), only 2 is a feasible time for event \( C \). This can be realized by scrutinizing the timelist for an incident once and continuing a sliding window of \( L_{max} \) newest time directs. At time period \( T_i \), if the disparity among \( T_j \) and \( T_i \) indicated by \( p \) is fewer than \( L_{max} \) for time period \( T_j \), \( j = i - 1, \ldots, i - L_{max} \), then \( \text{CheckSet}[p] \) is improved by one.

Hash-Based Validation (HBV): For every potential cycle \( p \) of an incident \( E \), this process scrutinizes the timelist one time and produces legitimate segments with time instant \( p \). Remind that segments symbolize synchronous periodic incidences and can be go beyond the certain limit. This is realized by maintaining paths of \( p \) autonomous (potentially overlapping) sections in a data structure termed \( C\text{Seg} \), where every \( C\text{Seg} \) proceeds the previous location where the event happens and the number of recurrences for present segment. For every time instant \( T_i \) in the timeliest of an incident, we calculate the modulus \( pos = T_i \% p \). The probable section is reserved in \( C\text{Seg}[pos] \). If \( T_i - C\text{Seg}[pos].last \) is precisely \( p \), it entails that this incident has happened at \( (T_i-p) \)-th time period. In this study, we enlarge \( C\text{Seg}[pos]\).rep by one and revise \( C\text{Seg}[pos]\.last \) by \( T_i \). If or else, \( T_i - C\text{Seg}[pos]\.last \) is not equal to \( p \), it entails the previous section has been broken up. In this folder, yield this section if \( C\text{Seg}[pos]\.rep \) is superior than \( \min rep \) and rearrange \( C\text{Seg}[pos]\.rep \) to 1 and \( C\text{Seg}[pos]\.last \) to \( T_i \). At last, scrutinize \( C\text{Seg} \) one time and output suitable segments if the recurrences are superior than \( \min rep \). Taking period 3 of incident \( D \) for instance, the procedure of scrutinizing \( D\.timelist \). Initialize every record of \( C\text{Seg} \) with \( rep = 1 \) and \( last = -Max \). Then the suitable section \( (D, p = 3, rep = 6, start = 3) \) is returned.

HBV specifies probable mixtures of legitimate segments of the similar time in depth-first order and ensure if a mixture forms a complex pattern. For two partly covering segments with diverse offsets, they can figure out a pattern if the recurrence of the overlying region is superior to \( \min rep \). To determine \( i \)-pattern, we can create it.
from an \((i-1)\)-model with 1-patterns. Consider a pattern \((A, *, B, C)\) be specified by \((B, C, A, *)\) or \((C, A, *, B)\), it is enviable to choose one demonstration to evade replication. The design is to choose the one with the prevalent duplication. Consequently, the primary constituent of the prototype is dogged by the section with the smallest amount end position. After that, every 1-pattern, \(S_i\) is positioned in the prototype with an counteract dogged by (the minimum end position).

The overall consumption time for processing the discovery of complex patterns for a specified incident \(e\) is \(2 \times ne\) (PCD + HBV), where \(ne\) is the amount of occurrences of incident \(e\). For a specified time length \(l\), the point to discover the remarkable periodic prototype for all actions is hence \(\sum \min n_e\) which is comparable to two database searches. Let \(D\) indicates the amount of time slots and \(T\) be the standard number of actions in every time slot. The database size can be symbolized by \(D \times T\). Accordingly, the time difficulty to determine all suitable segments for all time periods is \(O(D \times T \times Lmax)\). The data prearranged utilized for PCD and HBV when handing out an incident is CheckSet and CSeg, correspondingly. The dimension of the data structure is a manifold of \(Lmax\), which can be reclaimed for all measures. Consequently, the space intricacy is \(O(Lmax)\).

The proposed algorithm HVM utilizes a hash method for generation of candidate maximum pattern which in turn reduces the overall size of the transaction data set. The algorithm HVM uses the concept of hashing to eliminate unnecessary item sets for the next iteration which generates candidate item set. Iteratively following this step HVM collects the information about the candidate item set \((C + 1)\) in such a way that all \((C+1)\) datasets are pruned. The bucket in the hash represents how many item sets are mined and how many item sets has to undergo the mining process. Based on the resultant hash table obtained bit vector can be used to minimize the number of item sets in the hash table. A threshold value is fixed to be ‘\(T\)’ which is used as the basis for bit vector which sees to that resultant value is not greater than the threshold value ‘\(T\)’.

5. Experimental Results:

We implemented the GSMP, PCD and hash based validation of sequential pattern mining using Weka tool. The experimental evaluation of PCD, GSMP is conducted on the basis of time series manner using dense data sets in arff format. The impact of SP mining characteristics in the GSP, PCD and hash based algorithms are studied on Time series datasets of varied size (maximum of 1200 instances), considering all the enumerated characteristic attributes. The performance of PCD and GSMP only serves as a reference line to the performance of the hash algorithm, since the execution time are generally much larger than the execution times of hash algorithm. The experiments are conducted using Pentium dual core 2.5 GHz with 2GB of RAM with Windows XP operating using WEKA data mining tool. The time series dataset with synchronous and asynchronous time stamp were maintained in main memory during the algorithms processing.

To perform the analysis over a large range of different characteristics, we used real dense dataset of time series from Weka tool Machine Learning Repository. Figure 3 shows the relation between minimum support value (Min Sup) and number of sequential pattern generated (K). As shown in the figure 3 an increase in the support value leads to the decrease in the number of sequential pattern generated by way of using three algorithms GSMP, PCD and Hash-based algorithm.

![Fig. 3: Sequential pattern obtained with GSP and Hash algorithms](image-url)
Figure 4 shows the performance of processing time of three algorithms hash, PCD and GSMP with difference min_support value. When Min_Sup is 0.1 the running time is 75 seconds at the same time when Min_Sup is 0.2 the running time is decreased to 45 seconds and in this way an increase in Min_Sup value leads to a decrease in running time.

5.1 Performance of Hash-based GSMP for Synchronous and Asynchronous Time Stamp:

The behavior of both algorithms(GSMP and Hash-based) for synchronous and asynchronous differs on the basis of density. Hash algorithm achieves better results for dense dataset compared to that of GSMP in asynchronous pattern mining. The main behind this difference is that Hash-based algorithm does not waste much time generating infrequent candidates for dense datasets. In terms of memory consumptions, GSMP consumes more memory than hash, as it has to maintain multiple indexes for the same sequence. The results however show that both algorithms (GSMP and Hash-based) consume more memory while processing dense datasets. By way of varying the number of items in the dataset different values for density is achieved

5.2 Scalability:

The time consuming process is involved in scanning the database. The results differ based on the size of database. As the size of database grows larger scanning the database becomes a time consuming process. From the experiments conducted using Weka tool, in which worst behaviors which involves large datasets. Results also show that hash-based present significantly better performance for very large databases than GSMP and PCD. At the same time the hash-based consumes more memory than GSMP algorithm. This difference in memory consumption is already discussed in 5.1. The scalability performance of hash-based algorithm in terms of record size, for asynchronous db is compared with synchronous db to show improvement of sequential pattern mining is depicted in table 1 and figure 5 respectively.

<table>
<thead>
<tr>
<th>Recordsize</th>
<th>Time for Asynchronize db (seconds)</th>
<th>Time for synchronize db (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>628</td>
<td>0.0093</td>
<td>0.011</td>
</tr>
<tr>
<td>700</td>
<td>0.0098</td>
<td>0.016</td>
</tr>
<tr>
<td>802</td>
<td>0.011</td>
<td>0.021</td>
</tr>
<tr>
<td>910</td>
<td>0.014</td>
<td>0.028</td>
</tr>
<tr>
<td>1218</td>
<td>0.02</td>
<td>0.032</td>
</tr>
</tbody>
</table>

Another important factor in the performance of sequential pattern mining algorithms is the average length of sequences to be considered as in (Fahad Maqbool et al., 2006). In order to measure with different scenario, the generated datasets include sequences with different numbers of transactions. For long sequences which involve more than 25 item sets, the probability of supporting every element is very high. In this manner, hash-based is not able to reduce the search space and its candidate pruning does not eliminate a significant number of candidates.
The memory consumed in mining the sequential pattern for both asynchronous db and synchronous db are displayed in Table 2 and Figure 6 respectively. The performance of SPM shown in Figure 6 infers that for asynchronous db SPM consumes less memory compared to that of synchronous db. With the performance of execution time and memory consumption of SPM is good for both synchronous db and asynchronous db as compared to that of traditional pattern mining algorithm.

**Conclusion:**

The proposal our work presented a scheme for asynchronous periodic pattern mining. A sequential procedural algorithm which includes singular periodic pattern mining, complex periodic pattern mining and asynchronous sequence mining are devised to solve the problem. The overall contributions of the proposed asynchronous periodic pattern mining are presented below. The proposal of GSP, PCD and Hash based algorithm analyze sequential time based periodic pattern mining and comparison is made with the existing approaches. Real dense data sets have been used to compare and analyze the performance of GSP, PCD and hash based algorithm. As hash based algorithm is one of the optimization technique for sequential pattern mining every constraint used by GSP can be applied to our work without any change. The hash-based validation mechanism is constructed to discover the singular patterns using two-step process involving temporal database. Finally the time and memory space is analyzed and complexity proves the correctness of the hash and GSP algorithm. The experimental simulation shows that hash algorithm utilizes less consumption time in pruning than the PCD (nearly 5% to 6%). However in terms of scalability, memory consumption of hash algorithm is slightly higher (3% to 4%) than GSP for asynchronous periodic pattern mining.
References


Heikki Mannila, Hannu Toivonen, and A. Inkere Verkamo Discovering frequent episodes in sequences.


