International Journal on Applications in Engineering and Technology
Volume 2: Issue 5: May 2016, pp 6-10. www.aetsjournal.com ISSN (Online) : 2455 - 0523

An Efficient Algorithm For Knuth Morris Pratt Temporal Patterns In Time Interval Based Data

C. Indra Devi, V.Rajesh Kannan, D.Madhu Mitha

Abstract— Sequential mining applications using time interval based event data have attracted considerable efforts in discovering patterns from events. Here challenging issues are the relationship between two intervals is complex and how to effectively and efficiently mine interval based sequences. For solving complex relation, develop two novel representations such as end point and end time representation. Use algorithms Knuth Morris Pratt algorithm and Probabilistic Temporal Pattern Miner to discover the three types of interval based sequential patterns for reducing the computational process. Also they propose three types of pruning techniques in order to reduce the search space which are called scan pruning, point pruning, postfix pruning. The experimental results show the effective mining.

Key words: data mining, sequential pattern, temporal pattern, interval-based event.

I. INTRODUCTION

Data mining , the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

Data mining techniques are the result of a long process of research and product development. This evolution began when business data was first stored on computers, continued with improvements in data access, and more recently, generated technologies that allow users to navigate through their data in real time. Data mining takes this evolutionary process beyond retrospective data access and navigation to prospective and proactive information delivery. These include building efficient databases and indexes for sequence information, extracting the frequently occurring patterns, comparing sequences for similarity, and recovering missing sequence members. In general, sequence mining problems can be classified as *string mining* which is typically based on string processing algorithms and *item set mining* which is typically based on association rule learning.

There are several key traditional computational problems addressed within this field. These include building efficient databases and indexes for sequence information, extracting the frequently occurring patterns, comparing sequences for similarity, and recovering missing sequence members. In general, sequence mining problems can be classified as *string mining* which is typically based on string processing algorithms and *item set mining* which is typically based on association rule learning.

II. RELATED WORK

Allen's proposed a compact encoding method, for pattern named hierarchical representation, to efficiently express at the temporal relationships among intervals. However, event this method may suffer from two ambiguous problems. First, the same relationships among event intervals can be mapped to different temporal patterns. Second, a temporal pattern can represent different relationships among event intervals. Hoppner proposed an unambiguous representation, relation matrix, which exhaustively lists all binary relationships among event intervals in a pattern. Temporal representation utilizes the relationship among end time points to express the temporal pattern unambiguously only need to project the frequent finishing endpoints which have the corresponding starting endpoints in their prefixes .

Each Allen describer contains a counter that counts the number of relation occurrences. temporal considers the noise tolerance and expresses the of coincidence for interval patterns.

The pattern represented in This strategy is called point pruning, and can prune off non-qualified patterns before constructing the projected database however, this representation was based on closed sequential patterns and closed item sets; hence, the mining is time consuming. Coincidence Representation involves segmenting intervals into disjointed slices to In addition, several pattern-growth algorithms have been introduced to increase the efficiency of temporal pattern discovery.

Allen's was proposed to find frequent temporal patterns in a large database. This algorithm requires candidate generation to determine the relationship for growing patterns from local frequent intervals. T- Prefix Span entails generating all possible candidates and then discovering frequent events and

C. Indra Devi, Assistant Professor - Department of CSE, velammal College of Engineering and Technology, Madurai, India
Dr. V.Rajesh Kannan, H0D - Department of CSE, velammal College of Engineering and Technology, Madurai, India
D.Madhu Mitha , PG student-Department of CSE, velammal College of Engineering and Technology, Madurai, India.

scanning the projected databases recursively to discover all temporal patterns.

patta was developed to mine hybrid temporal pattern from event sequences, which contain both point based and interval based events. Sadasivam et al. modified the T-PrefixSpan algorithm and proposed a refinement method to reduce the number of database scans.

Actually, mining temporal patterns is more arduous than mining point-based patterns because of the complex relationships among intervals. For example, given a point-based database consisting of 2 frequent items, the first three levels of search space for mining sequential patterns the length of the longest sequential pattern is *h*, the size of the search space is 2+2x(2x2)1+2x(2x2)2+…+2□(2x2)*h*-1 =O(4*h*).

However, compared to mining sequential pattern, the search space for discovering temporal patterns from an interval-based database containing 2 frequent intervals is much larger. The Allen's 13 relationships can be reduced to relations . If the length of the longest temporal pattern is *h*, the size of the search space is 2+2x(71x2)+2x(71x2) +2x(72x2)x…x(7*h*-1x2) = O(7*h* 2*h* 2 the complex relationships among intervals create a huge search space and complicate the mining processes for temporal patterns. From the analysis above, we know that the complex relationship is a critical concern when designing an efficient

III. PROPOSED METHOD

Representation of data set is the first process in our approach. Currently, time interval-based mining problem is much more arduous than the time point-based mining problem.

Here two time intervals may overlap, and then the relationship among event intervals is more complex than that of the event points. In this paper, two new representations are developed called endpoint representation and end time representation, to effectively express temporal patterns. says that, the complex relationships among event intervals are the major bottleneck for mining temporal patterns. Endpoint representation utilizes the endpoint arrangements to express the relations among intervals in sequence unambiguously. The time information is so critical for numerous applications. end time representation, which not only expresses relations among intervals but also reveals the occurrence time.

Temporal database, and event intervals associated with the same sequence ID are grouped into an interval sequence. It is first transformed the temporal database into the endpoint representation, and then scans the database to calculate the count of each endpoint concurrently. Then it removes infrequent endpoints below the given support threshold. For each frequent starting endpoint, we build the projected database and it is processed recursively to discover sets of all temporal patterns. Finally, outputs all temporal patterns.

The output of Temporal pattern is transformed into end time representation, and then finds frequent endpoints and removes infrequent ones then includes the concept of probabilistic function calculation. Frequent endpoint can be appended to the original prefix to generate a new frequent sequence.

Time information in database to estimate the occurrence- and duration-probability functions. If all endpoints in a frequent endpoint sequence appear in pairs, i.e., every starting (finishing) endpoint has a corresponding finishing (starting) endpoint, we can output this frequent endpoint sequence, including its occurrence and duration probability function, as the occurrence- and duration-probabilistic temporal pattern, respectively.

The Pruning Techniques is important in our process. These are the properties of endpoints; we propose three pruning strategies called scan-pruning, point pruning, and postfix-pruning for efficiently and effectively reducing the searching space.

First, to calculate the support of all endpoints in the database, It is unnecessary that scanning each sequence from the beginning to the end.

Instead, we only need to scan from the start of each sequence and stop at the first finishing endpoint which has a corresponding starting endpoint in prefix. Because of endpoints which always appear in pairs in a pattern, a frequent sequence will never become a pattern if it has no chance of obtaining all pairs of endpoints in its subsequent growth.

This process is called scan pruning. Then the next process is, the starting and finishing endpoints definitely occur in pairs in a sequence.

The frequent finishing endpoints which have the corresponding starting endpoints in their prefixes. This process is called point pruning, for pruning non-qualified patterns before constructing the projected database when constructing a projected database, some endpoints in the postfix sequences need not be considered we use point pruning.

With respect to a prefix sequence, finishing endpoint in a projected postfix sequence is essential if it has corresponding starting items. When constructing the projected database, only the essential endpoints in the postfix sequences are collected.

All nonessential items are eliminated because they can be ignored during the discovery of temporal patterns. For eliminating the non essential items we use the last pruning method is called postfix pruning, for projecting a database.

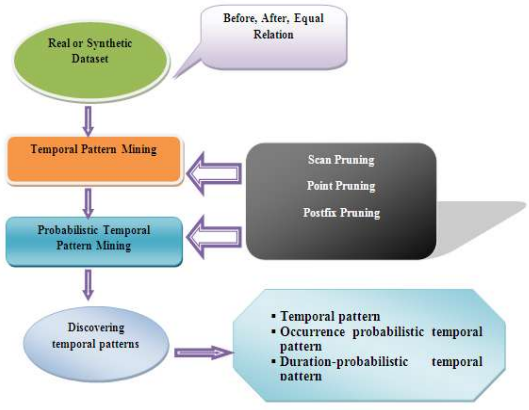


Figure 1 Temporal Process

In consideration of the properties of end points, propose three pruning strategies (scan-pruning, point pruning, and postfix-pruning) for efficiently and effectively reducing the searching space. First, in order to calculate the support of all endpoints in *DB* scanning each sequence from the of begin to the end is unnecessary.

Instead, only need to scan from the start of each sequence and stop at the first finishing endpoint which has a corresponding starting end point in prefix Because the endpoints always appear in pairs in a pattern, a frequent sequence will never become a pattern if it has no chance of obtaining all pairs of endpoints in its subsequent growth. This strategy is called scan pruning.

procedure *count_support*, for each sequence in *DB* first find the *stop_position*, and then accumulate the supports of all endpoints from the beginning of the mining sequence to the *stop_position stop_position* is not found, if set the *stop-position* as the last endpoint in the sequence. Let the database in with *min_sup* = 2 .length of the mining sets.

Set of maximal potentially large sequences first created to generate event sequences. The number of maximal potentially large sequences is *NS*. A maximal potentially large sequence is generated by first selecting the size from a Poisson distribution with mean equal to |*S*|

Randomly choose the event interval symbols in the maximal potentially large sequence from *N* events. Since the time interval in a sequence has duration, the data generator for temporal pattern mining algorithms requires an additional tuning for the experimental data generation. dopt the modification proposed by Wu et al. All the duration times of the event intervals are classified into three categories: long, medium and short. The long, medium and short interval events have an average length of 12, 8 and 4 respectively. For each event interval, we first randomly decide its category and then determine its length by drawing a value from a mining date for as more then time in normal distribution.

Finally, we randomly select the temporal relations between data mining is primarily used today by companies with a strong consumer focus - retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

Cluster characteristic or discriminate rules associate objects belonging to a cluster of some attributes with some probability. A -temporal clustering might discover that “The grid cells with similar values in meteorological satellite image at noon, can be clustered as a spot at high temperature, tending to be a fire spot”. Widely used spatial clustering techniques e.g., *K*-means and *K*-medoids and CLARANS may be extended for temporal clustering.

Basically temporal data mining is concerned with the analysis of temporal data and for finding temporal patterns and regularities in sets of temporal data. Also temporal data mining techniques allow for the possibility of computer driven, automatic exploration of the data because the time series is not steady, in transform the data to steady sequence according to difference method. After the data has been appropriately transformed according to the logarithmic method.

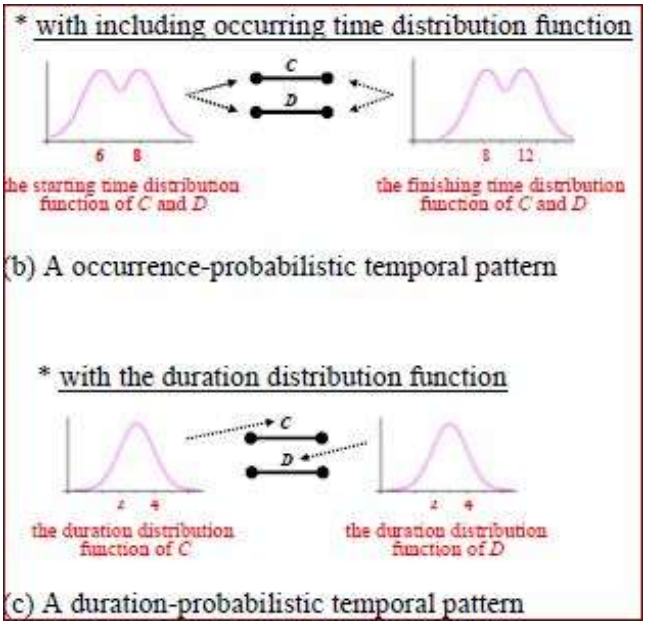


Figure 2 Interval Based Pattens

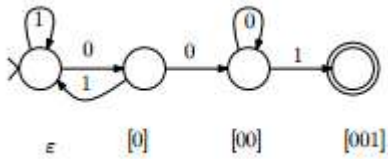
several appliances, occurring time information of intervals could provide the insights into the discovered temporal patterns. The following application describes how temporal pattern with time information are utilized to detect anomalies in appliance usage.

IV. KNUTH MORRIS PRATT

This algorithm can solve the classic text search problem in linear time in the length of the text string. (It can also be used for a variety of other string searching problems.) Formally, you have a pattern P of length p, and a text T of length t, and you want to find all locations i such that T[i . . . , i + p − 1] = P.

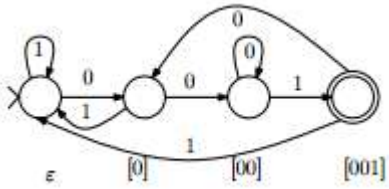
For example, if you have a pattern P = ana and T = banana, then you would want to output locations {1, 3}, since T[1 . . . 4] = T[3 . . . 6] = ana. 1 An easy solution to this problem takes time O(p·t). This is fine if p is small, but as p gets large this becomes bad. It seems that if once we have checked whether T[i . . . i + p − 1] matches P or not, we have a lot of information that we can use to determine if T[i + 1 . . . i + p] gives a match. One high-level solution to this problem is to build a deterministic finite automaton MP , which depends on the pattern. It consists of p = |P| states, which you should think of as arranged in a line. If feed in the text T to this automaton.

The properties of the DFA M ensure that we're in the j th state if and only if at the current location in the string already matched a sequence of j characters from the pattern. Now we compare the next two characters. If we get a match we move to the next state in the list (matching j + 1 characters). If we get a mismatch skip back to some previous state. The start state of the automaton that would be wrong. We go to the furthest back state that we know we can go to based on the information that we have. For example, if the pattern P = 001 and suppose we consider the DFA .



Mining would reach the final state (the one with the double circle) exactly when we have seen the pattern. This would allow us to find the first occurrence of pattern P in the text T. And in order to find all the occurrences of P, we could alter the DFA slightly to s get this one:

Output the current location in T every time we hit the final state, and then output all occurrences of P in the text T.



The features of time intervals and time points vary substantially; the pair wise relationship between two time interval-based events is intrinsically complex. This complex relationship is a critical problem in the endeavor to design an efficient and effective time interval based pattern (or temporal pattern) mining algorithm, since it may increase candidate generation and the workload for counting the support of candidate sequences.

V. EXPERIMENTAL RESULTS

The relation format in is a typical non-first-normal-form representation, in which a relation is thought of as recording information about some types of objects. The present relation records information about customers and thus holds one tuple for each customer in the example, with a tuple containing all information about a customer. In this way, a single tuple records multiple facts .

The statistical parameters were unique for each and every regions in the images and hence the process can be more accurate while recognizing based on the similarities. Mean and Standard deviation is the basic statistics for all the types of images and time frames. The statistical parameters were estimated for each and every patches divided.

Because the three relations are different, they can well be taken to mean different things. In temporal data models, however, the relations are typically taken to contain the same information because they all contain the same snapshots: the snapshots computed at times 2 through 8 contain two tuples with values "a" and "b," (denoting some facts) and the snapshots at all other times are empty.

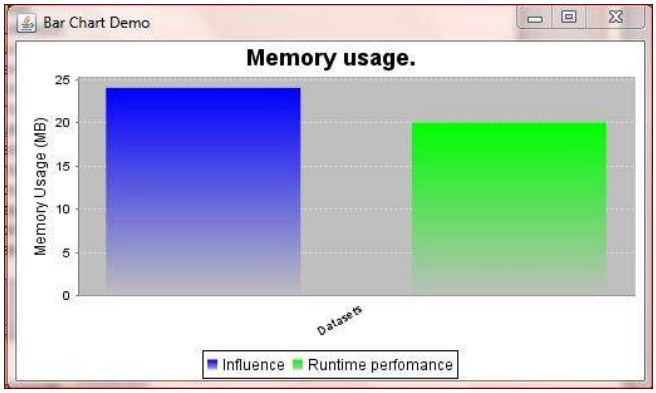


Figure 3 Memory Performance

The statistical parameters were unique for each and every regions in the images and hence the process can be more accurate while recognizing based on the similarities. Mean and Standard deviation is the basic statistics for all the types of images and video frames. The statistical parameters were estimated for each and every patches divided.

The mining process in the face reduced the mapping patterns and computational decrease memory space .

VI. CONCLUTION

Mining temporal patterns from time interval-based data is an important subfield in data mining. Here, the relationships are intrinsically complex among intervals and tends to increase the difficulty of designing an efficient algorithm. In this paper, for solving complex relations among the intervals we develop two new representations called endpoint representation and end time representation. we propose two algorithms TP Miner and P-TP Miner, are developed to efficiently discover three types of patterns: temporal pattern, occurrence- probabilistic temporal pattern, and duration probabilistic temporal pattern, based on the two representations and to describe the correlation among intervals and the probability of the occurring time and duration of each intervals. We also propose several pruning techniques to effectively reduce the search space. The experimental studies indicate that TP Miner and P-TP Miner are efficient and practical.

REFERENCES

[1] E. Winarko and J.F Roddick, "ARMADA-An algorithm for discovering richer relative temporal association rules from interval- based data," *Data and Knowledge Engineering*, vol. 63, issue 1, pp. 76-90, 2007.

[2] A. Wong, D. Zhuang, G. Li, and E. Lee, "Discovery of Closed Patterns and Noninduced Patterns from Sequences," *IEEE Transactions on Knowledge and Data Engineering*, vol.24, no. 8, pp. 1408-1421, 2012.

[3] S. Wu and Y. Chen, "Mining Nonambiguous Temporal Patterns for Interval-Based Events," *IEEE Transactions on Knowledge and Data Engineering*, vol.19, no. 6, pp. 742-758, 2007.

[4] S. Wu and Y. Chen, "Discovering hybrid temporal patterns from sequences consisting of point- and interval-based events," *Data & Knowledge Engineering*, vol.68, issue 11, pp.1309–1330, 2009.

[5] J. Yang, W. Wang, and P. Yu, "InfoMiner: Mining Surprising Periodic Patterns," *Data Mining and Knowledge Discovery*, vol. 9, no. 2, pp. 189-216, 2004.

[6] J. Yang, W. Wang, P. S. Yu, and J. Han, "Mining Long Sequential Patterns in a Noisy Environment," *The 2002 ACM SIGMOD international conference on Management of data (SIGMOD'02)*, pp. 406-417, 2002.

[7] M. Yoshida, T. lizuka, H. Shiohara and M. Ishiguro, "Mining Sequential Patterns Including Time Intervals," *Proceeding of SPIE - Data Mining and Knowledge Discovery: Theory, Tools, and Technology II*, vol. 4057, pp. 213-220, 2000.

[8] M. Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences," *Machine Learning*, vol. 42, no. 1-2, pp. 31-60, 2001. [35] A. Zakour, S. Maabout, M. Mosbah and M. Sistiaga, "Uncertainty Interval Temporal Sequences Extraction," *International Conference on Information Systems Technology and Management (ICISTM' 12)*, pp. 259-270, 2012.

[9] [36] Z. Zhao, D. Yan and W. Ng, "Mining Probabilistically Frequent Sequential Patterns in Large Uncertain Databases," *The 15th International Conference on Extending Database Technology (EDBT'12)*, pp. 74-85, 2012.

[10] R.Villafane, K. Hua and D. Tran, "Knowledge Discovery from Series of Interval Events," *Journal of Intelligent Information Systems*, vol.15, pp.71-89, 2000.