Mining Closed Sequential Patterns in Large Sequence Databases

Lokendra Shah*, Mrs. Priyanka Chouhan** and Dr. N. K.Tiwari***
* Research Scholar, Dept. of CSE, Bansal Institute of Science and Technology
Bhopal, M.P., India
** Assistant Professor, Dept. of CSE, Bansal Institute of Science and Technology
Bhopal, M.P., India
***Bansal Institute of Science and Technology, Bhopal

Abstract: Sequential pattern mining is studied widely in the data mining community. Finding sequential patterns is a basic data mining method with broad applications. Closed sequential pattern mining is an important technique among the different types of sequential pattern mining, since it preserves the details of the full pattern set and it is more compact than sequential pattern mining. An important goal of knowledge discovery is the search for patterns in the data that can help explaining its underlying structure. To be practically useful, the discovered patterns should be novel (unexpected) and easy to understand by humans. In this thesis, we study the problem of mining patterns (defining subpopulations of data instances) that are important for predicting and explaining a specific outcome variable. In this paper, we propose an efficient algorithm Enhanced CSpan for mining closed sequential patterns. Like CSpan, we uses a pruning method called occurrence checking that allows the early detection of closed sequential patterns during the mining process. Our extensive performance study on various real and synthetic datasets shows that the proposed algorithm Enhanced CSpan outperforms the CSpan and previously proposed algorithm by an order of magnitude. Our pattern mining method works on complex Large sequence database , such as electronic health records, for the event detection task.

Keywords: Data mining, sequential pattern mining, closed sequential pattern mining, sequence database.

Introduction

If we consider an another example, we can say that a Store managers can provide the recommendation to the user with this information when a user browses the data mining book. We can also identify block access patterns of disk systems with the help of this information. The block access patterns can easily predict the blocks that are accessed next, and these blocks can be perfected into cache to reduce the disk I/O latency. Sequential pattern mining is an important and active research topic in data mining From more than 25 yrs..There are many algorithms which are proposed for sequential pattern mining since the first it introduced.

Databases contain large amount of data, information and sequences inside it. there for whenever we want to generate patters from the databases, Sequential pattern mining produces an exponential number of patterns when the database contains long sequences which are expensive in both time and space. The same problem also occurs in item set and graph mining when the patterns are long. For example, assume the database contains a long sequence $\{(x_1)(x_2)\ldots\ldots(x_{200})\}$ it will generate $2^{200} - 1$ frequent subsequence's. That's why we generally used Closed sequential pattern mining for finding the patterns from long sequences.

Closed sequential pattern mining extensively reduces the number of patterns produced and it can be utilized to obtain the complete set of sequential patterns. Some subsequence has the support which is equivalent to the support of the long sequence, which are basically redundant patterns. Therefore, instead of mining the complete set of sequential patterns, it is better to mine closed sequential patterns only.

A closed sequential pattern is a sequential pattern which has no subsequence with the same support. In addition, closed sequential pattern mining algorithms make use of search space pruning techniques and outperform sequential pattern mining algorithms.

As an example, if we consider a sequence database which has thousands of sequences and we want to generate patterns, there are thousands of sequential patterns generates, where as only hundreds of closed sequential patterns are generate.

There are various popular algorithms for closed sequential pattern mining: CloSpan[2] and BIDE[3]. CloSpan performs the mining in two steps. In the first stage it produces a closed sequential pattern result set candidate set and stores it in a prefix sequence lattice. In the second steps it performs post pruning to eliminate non closed sequential patterns. CloSpan works under candidate maintenance-and-test paradigm, therefore it is not scalable because a large number of closed sequential pattern candidates will occupy more memory space and lead to huge search space for checking the closure of new patterns,

particularly for low support threshold values or long patterns. BIDE mines closed sequential patterns without candidate maintenance.
An another algorithm CSpan is also provided recently, which is more compact and fast than the other previously provided algorithms. CSpan uses a pruning method called occurrence checking that allows the early detection of closed sequential patterns during mining process.
In this paper, we propose a effective algorithm called Enhanced CSpan, which is the extensive version of CSpan. which is more efficient and less time consuming for generation of closed sequential patterns. Enhanced CSpan outperforms the CSpan and previously proposed algorithm by an order of magnitude. Our pattern mining method works on complex Large sequence database and synthetic datasets.

Related Work

Agrawal and Srikant[1] first introduced the sequential pattern mining problem. Later a number of efficient algorithms have been developed for sequential pattern mining. These algorithms are categorized into three groups such as Apriori-based methods, pattern-growth methods and vertical format based methods.
Apriori-based methods implement a candidate-generation-and-test strategy. The data also provide information about the incidence of several adverse medical events, such as diseases or drug toxicities. Our objective is to mine patterns that can accurately predict adverse medical events and apply them to monitor future patients. This task is extremely used for intelligent patient monitoring, outcome prediction and decision support.
The representative algorithm of these methods is GSP [5], which is an enhancement of AprioriAll algorithm developed by the same authors. mining in closed patterns in abstract time-interval data is very challenging mainly because the search space that the algorithm has to explore is extremely large and complex. All existing methods in this area have been applied in an unsupervised setting for mining association rules Algorithms implementing a candidate-generation-and-test strategy produce a large number of candidate sequences. Counting and pruning such a large set is more expensive. In addition, Apriori-based methods require more number of database scans when there are long patterns.
Pattern-growth methods generally start with a frequent pattern, and grow the frequent pattern when traversing the pattern search space using depth-first search. FreeSpan[6], PrefixSpan [7], CloSpan, BIDE and etc belonging to this type. PrefixSpan finds the frequent-1 sequences in the database and builds projected databases for each sequence. In contrast to the existing methods, our work applies pattern mining in the supervised setting to find patterns that are important for the event detection task. To ef-ficiently mine such patterns, we propose the Recent Patterns (RTP) framework. It then searches the projected database to uncover locally frequent items and recursively locates the frequent sequences that contain the item as a prefix. This process is repeated until all frequent sequences are discovered. We present an efficient algorithm that mines time-interval patterns backward in time, starting from patterns related to the most recent observations. Finally, we extend the minimal predictive patterns framework to the domain for mining predictive and non-spurious RTPs.
Closed itemset mining is used to find closed itemsets in a transaction database. Several algorithms were developed for closed item set mining. A-Close[10] is the first algorithm developed for mining closed itemsets. It first finds level-wise frequent itemsets using Apriori strategy, and mines all minimal generators. In the second step, it computes the closure of all minimal generators. The performance of A-Close degrades due to the huge cost of the off-line closure calculation because both the itemsets belong to the same equivalence class. Charm uses diff-set technique for sorting itemset tid-lists in each node of the tree.
It is not feasible to adopt the techniques used in closed itemset mining for closed sequential pattern mining, because subsequence testing needs order matching and the search space of sequences is bigger than that of itemsets. Also, sequential pattern mining is generally less efficient than closed sequential pattern mining, particularly in mining long patterns and with low support threshold.
X.Yan et al. developed the CloSpan algorithm for mining closed sequential patterns. CloSpan produces less number of discovered sequences than the traditional methods while preserving the same expressive power. CloSpan can mine long sequences and runs faster than PrefixSpan. CloSpan divides the mining process into two phases. In first phase it generates a candidate set for closed sequential patterns and stores it in a prefix sequence lattice. In second phase it conducts post pruning to eliminate non closed sequences.
CloSpan prunes the nonclosed sequential patterns by using an efficient search space pruning method, known as equivalence of projected databases. Unfortunately, a closed sequential pattern mining algorithm under candidate maintenance-and-test paradigm has rather poor scalability because a large number of closed sequential patterns candidates require more memory and lead to huge search space for the closure checking of new patterns, which is usually the case when the support threshold is low or the patterns are long.
Wang et al. developed the BIDE algorithm. It mines closed sequential patterns without candidate maintenance. It prunes the search space more deeply and performs closure checking of patterns in an efficient manner. BIDE follows a strict depth-first search order to produce the closed sequential patterns. It uses a novel closure checking scheme called BI-Directional Extension. The forward directional extension is used to grow the prefix patterns and also for closure checking of prefix

patterns, whereas the backward directional extension is used for both closure checking of a prefix pattern and pruning the search space. It prunes the search space by using the BackScan pruning method. BIDE first applies the BackScan pruning method to check if a prefix sequence can be pruned, if not, computes the number of backward-extension items. Later it finds the number of forward extension items, if there is no backward-extension item or forward-extension item then it outputs closed sequential patterns. Although BIDE does not keep track of any historical closed sequential patterns candidates for a new pattern's closure checking, it is a computational consuming approach since it needs multiple database scans for the bi-direction closure checking and the BackScan pruning.

An another algorithm CSpan is also provided recently, which is more compact and fast than the other previously provided algorithms. CSpan uses a pruning method called occurrence checking that allows the early detection of closed sequential patterns during mining process. algorithm CSpan. uses depth-first search for generating the closed sequential patterns. Since many previously developed algorithms proved that depth-first search based algorithm is more efficient than the breadth-first search based algorithm in mining long patterns.

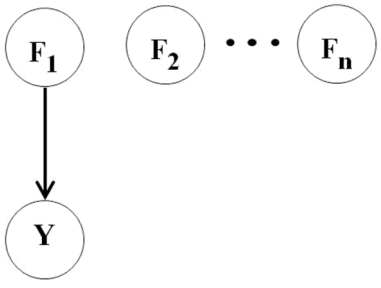
Problem Definition

In this section, we first introduce some preliminary concepts, and then formalize the closed sequential pattern mining problem.

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of all items. A subset of I is called an itemset. A sequence $S=(k_1, k_2, \ldots, k_n)$ ($k_i \subseteq I$) is an ordered list of itemsets. The items in each itemset are sorted in alphabetic order. The length of the sequence is the total number of items in the sequence. A sequence $S_1=(a_1, a_2, \ldots, a_m)$ is a subsequence of another sequence $S_2 =(b_1, b_2, \ldots, b_n)$, denoted as $S_1 \sqsubseteq S_2$, if there exists integers $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ and $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i2}$, . . . , and $a_m \subseteq b_{im.}$ We $\sqsubseteq$ call S_2 as a super-sequence of S_1 and S_2 contains S_1.

Definition 1. A pattern P is a spurious pattern if P is predictive when evaluated by itself, but it is redundant given one of its subpatterns. Spurious patterns are formed by adding irrelevant items to other simpler predictive patterns.

Definition 2 (Closed sequential pattern): A sequential pattern α is a closed sequential pattern if there does not exist a sequential pattern β , such that support (α) = support (β) and $\alpha \sqsubset \beta$.



The problem of closed sequential pattern mining is to find the complete set of closed sequential patterns above a minimum support threshold m_sup for an input sequence database SD. Table 1 shows a sample sequence database. The items in each itemset are sorted in alphabetic order.

Table 1. A sample sequence database

Sid	Sequence
1	(ab)(bc)
2	(bc)(d)(e)
3	(ac)(bcd)

Proposed Method

In this section, we describe our proposed algorithm Enhanced CSpan. It uses depth-first search for generating the closed sequential patterns. Since many previously developed algorithms proved that depth-first search based algorithm is more efficient than the breadth-first search based algorithm in mining long patterns.

Like the CSpan algorithm, Enhanced CSpan uses a sequential pattern tree to generate the closed sequential n patterns. The sequential pattern tree is constructed as follows. The root of the tree is labelled with φ. Next, all frequent 1-sequences in the database are identified and added to level 1 of the sequential pattern tree. Then, a sequence k at level 1 is extended by appending a frequent item in k's projected database to get its Frequent super-pattern at level 2. A sequence can be extended in two ways: sequence extension and itemset.

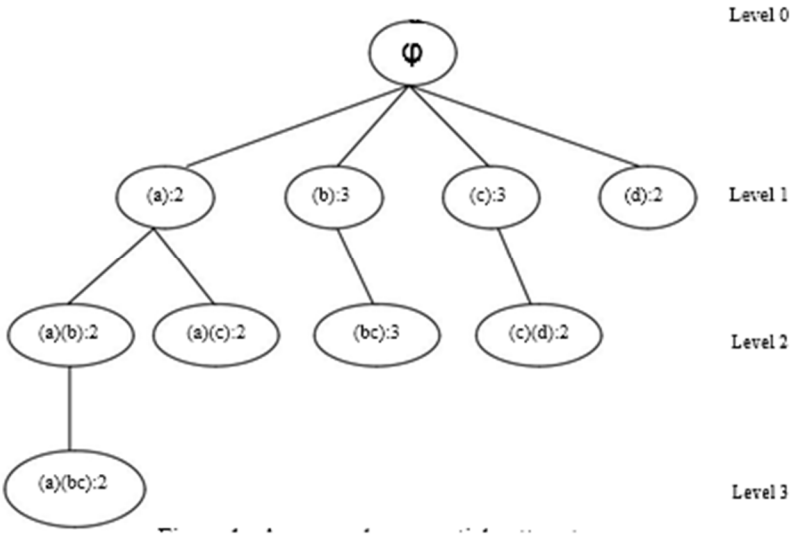


Figure No.1 An example of sequential Pattern tree

Extension. In case of sequence extension, the item is added as a new itemset to the sequence. In case of itemset extension, the item is appended to the last itemset in the sequence.

A sequence in the sequential pattern tree is considered as a sub-sequence of its child. To locally mine the frequent super-sequences of a certain sequence, we construct the projected database for the sequence and grow the sequence to obtain its frequent super-sequences. We use the pseudo based projection approach of Prefix Span to create the projected database.

Occurrence checking

As our Enhanced CSpan algorithm is extension and improved form of CSpan, thats why We also used the pruning method called occurrence checking that allows the early detection of closed sequential patterns in pattern mining.

Lemma 1. Occurrence checking: A sequential pattern X is not closed if a frequent item y exists such that (1) y appears in every sequence of X's projected database and (2) the distance between X and y is identical in every sequence of X's projected database.

Proof. If a frequent item y appears in every sequence of X's projected database and the distance between X and y is identical in every sequence of X's projected database, then we can always discover another frequent sequence containing X and y whose support is equivalent to X's support. Therefore, X cannot be closed.

We demonstrate the occurrence checking scheme as follows. Assume *a* is a frequent sequence. If we locate an item *b* after *a* in every sequence of *a*'s projected database, then we can declare that *a* is not closed since *ab* is its super-sequence with the same support.

Enhanced CSpan Algorithm

Like CSpan, The Enhanced CSpan algorithm also has two phases. First, it scans the database to determine all frequent 1-sequences. Second, for each frequent k-sequence, it constructs the projected database and locates all frequent items in the projected database to produce its frequent super-sequences at the next level in the sequential pattern tree, where k≥1. During the mining process, it employs occurrence checking for early detection of closed sequential patterns. So the proposed algorithm for Enhanced CSpan is provided here.

Algorithm 1: Enhanced CSpan

Input: A sequence database SD.

Output: The complete set of closed sequential patterns.

1. S1 = sequential database
2. CSP = φ
3. SP = φ
4. while(s<S1)
5. SD = Sequential database of s
6. SP = seq_patterns (SD, s)
7. CSP = CSP ∪ SP
8. end while

Algorithm 2: seq_patterns(SD, s)
Input: A sequence database SD, sequence s
Output: Closed sequence patterns with prefix s.
1. SP = φ
2. P1 = frequent items in SD
3. If CSP =null and P1≠ φ then
4. Exit // to save the number of comparison
7. else
8. while (p<=P1)
10. SD=Sequence database of s Δs p
11. SP=SP ⊔ seq_patterns(SD, s Δi p)
14. end while
15. end if
16. return SP

Performance Evaluation

In this section, we perform a thorough performance evaluation of our proposed Enhanced CSpan algorithm on both real and synthetic data sets with various kinds of sizes and data distributions, and we compare Enhanced CSpan with CSpan and a priviously proposed algorithms.
A set of experiment were conducted to evaluate the performance of the Enhanced CSpan algorithm. the set compares the runtime performance of Enhanced CSpan with CSpan and previous old algorithms using synthetic datasets for different values.

Table.2 Characteristics of the Synthetic Dataset

S.No.	Characteristics	Dataset 1Value
1	No. Of sequence	1000
2	No of distinct items	6
3	No of items per itemset	3
4	No of itemset per sequence	3

Figure2 shows the results of runtime performance using the synthetic dataset. the X-axis is the support/duration, while the Y-axis is the algorithms runtime.figure3 is also showing the result of performance evaluation of dataset.

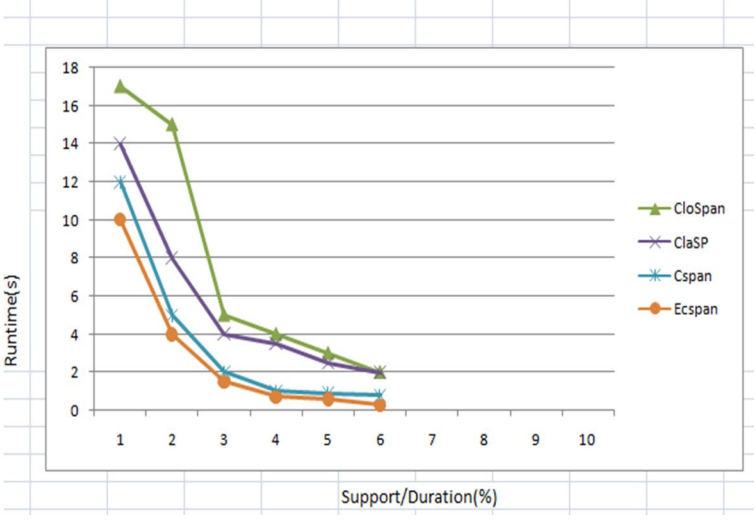


Figure .2 Performance comparison using Synthetic Dataset

Graph clearly show that Enhanced CSpan outperforms the other previous algorithms by some magnitude. We applied the both CSpan and Enhanced CSpan on a synthetic dataset and the pattern generation be new algorithm takes less time than the older algorithm. Result shows that Enhanced CSpan is more effective and faster.

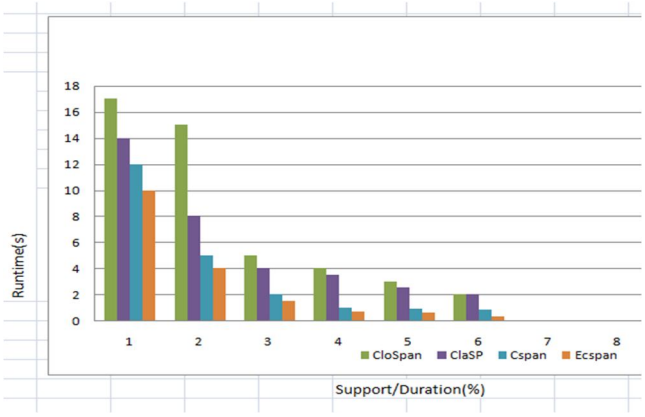


Figure.3 Performance comparison using Synthetic Dataset

Conclusion

Several researchers focused on the sequential pattern mining problem and many algorithms were developed to mine sequential patterns. Closed sequential pattern mining is a variant of sequential pattern mining and attains broad attention in the recent years because it has the same expression ability of the sequential pattern mining and more compact than the sequential pattern mining. It extensively reduced the no of patterns generation.

In this paper, we propose an efficient algorithm Enhanced CSpan which is improved form of algorithm CSpan. which makes use of a method called occurrence checking that allows the early detection of closed sequential patterns during the mining process. Our extensive performance study on synthetic datasets shows that the proposed algorithm Enhanced CSpan outperforms the CSpan and other proposed algorithm by an order of magnitude.

In future, we will extend Enhanced CSpan to incorporate user specified constraints. There are many areas where closed sequential pattern mining will be used for the better information generation.

References

[1] R. Agrawal and R. Srikant, “Mining sequential patterns,” Proc. Int’l Conf. Data Engineering (ICDE ’95), pp. 3-14, Mar. 1995.

[2] X. Yan, J. Han, and R. Afshar, “CloSpan: Mining closed sequential patterns in large databases,” Proc. SIAM Int’l Conf. Data Mining (SDM ’03), pp. 166-177, May 2003.

[3] J. Wang, J. Han, and Chun Li, “Frequent closed sequence mining without candidate maintenance,” IEEE Trans. Knowledge and Data Eng., vol. 19, no. 8, pp. 1042-1056, Aug. 2007.

[4] Antonio Gomariz, Manuel Campos, Roque Marin, and Bart Goethals, “ClaSP: An efficient algorithm for mining frequent closed sequences,” PAKDD 2013, LNAI 7818, Part I, pp. 50–61, 2013.

[5] R. Srikant and R. Agrawal, “Mining sequential patterns: Generalizations and performance improvements,” Proc. Int’l Conf. Extending Database Technology (EDBT ’96), pp. 3-17, Mar. 1996.

[6] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.C. Hsu, “FreeSpan: Frequent pattern-projected sequential pattern mining,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD’00), pp. 355-359, Aug. 2000.

[7] J . Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, “PrefixSpan : Mining sequential patterns efficiently by prefix-projected pattern growth,” Proc. Int’l Conf. Data Engineering (ICDE ’01), pp. 215-224, Apr. 2001.

[8] M. Zaki, “SPADE: An efficient algorithm for mining frequent sequences,” Machine Learning, vol. 42, pp. 31-60, 2001.

[9] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, “Sequential pattern mining using a bitmap representation,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD’ 02), pp. 429-435, July 2002.

[10] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lot Lakhal, “Discovering frequent closed itemsets for association rules,” Proceedings of the 7th International Conference on Database Theory (ICDT '99), pp. 398-416,1999.

[11] J. Pei, J. Han, and R. Mao, “CLOSET: An efficient algorithm for mining frequent closed itemsets,” Proc. ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD ’00), pp. 21-30, May 2000.

[12] J. Wang, J. Han, and J. Pei, “CLOSET+: Searching for the best strategies for mining frequent closed itemsets,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’03), pp. 236-245, Aug. 2003.

[13] M. Zaki and C. Hsiao, “CHARM: An efficient algorithm for closed itemset mining,” Proc. SIAM Int’l Conf. Data Mining (SDM ’02), pp. 457-473, Apr. 2002.

[14] Kuo-Yu Huang, Chia-Hui Chang, Jiun-Hung Tung, and Cheng-Tao Ho, “COBRA: Closed sequential pattern mining using bi-phase reduction approach,” Proceedings of 8th International Conference, DaWaK, Springer LNCS 4081, pp. 280-291, 2006.

[15] Ron Kohavi, Carla E. Brodley, Brian Frasca, Llew Mason, and Zijian Zheng, “KDD-Cup 2000 organizers' report: Peeling the onion,” SIGKDD Explorations, vol. 2, no. 2, pp. 86-93, Dec. 2000.

[16] Fournier-Viger P., An Open-Source Data Mining Library, http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php, 2008, Accessed 20 July 2014.

[17] S. Cong, J. Han, and D.A. Padua, “Parallel mining of closed sequential patterns,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (SIGKDD ’05), pp. 562-567, Aug. 2005.