ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 6.047

Volume 5, Issue 2, February 2017

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Mining Huge Data with Closed Sequential Pattern Model

Dr. K. Subramanian[1]
Assistant Professor,
H.H The Rajah's College,
Pudukkottai,
Tamil Nadu – India

S. Surya[2]
Ph.D. Research Scholar,
J.J. College of Arts and Science,
Pudukkottai,
Tamil Nadu – India

Abstract: ***Pattern Mining is the essential task in the data mining industry. Several pattern mining models are available to process such as: Distributed Pattern Sequence Modelling, Structural Pattern Mining and so on. But there is a requirement for preserving the information about set of patterns and require compactness in mining methodology. A new pattern mining scheme is introduced to achieve this task called "Closed Sequential Pattern Mining", which maintains the information regarding pattern-set more compact and privacy oriented manner. There are lots of existing algorithms available to mine the closed patterns such as CSpan, CLOSET, MAFIA, CHARM and so on. All these algorithms are good in working and perform mining well but the complexity occurs when these algorithms deals huge data without entity relationships. In this case we need a improvised process model which mine the patterns using closed sequential model along with more perfect mean. In this paper a new pattern mining algorithm is introduced called "Subsequent CloSpan Algorithm [SCloSpan]", which is abbreviated as Subsequent Closed Sequential Pattern Mining. This SCloSpan algorithm provides an efficient decision tree reduction model with power duplicate content checking along with data stemming reduction principles, which provides earlier prediction of closed sequential patterns over the mining procedure. Our procedural analysis shows that the sequence is comparatively reduced than the existing procedures as well as preserving the processing time with complete set of subsequent patterns.***

Keywords: Closed Pattern Mining, Sequential Pattern, Subsequent Pattern Mining, SCloSpan, Frequent Patterns.

I. INTRODUCTION

Pattern-Mining with SubSequent Sequence Analysis are the most important mining as well as research task now-a-days. Several Sequential Analysis tasks are introduced as well as analysed for past several years to identify the sequential mining patterns efficiently. Consecutive patterns can be connected to a few business as well as logical applications including organic arrangement investigation, advertise examination, finding web get to patterns, mining client shopping arrangements, web click streams, XML inquiry get to designs for reserving, piece connections away frameworks, groupings of document piece references in working frameworks, target showcasing, client maintenance, highlight choice for arrangement grouping, client conduct investigation, web recommender frameworks, organize interruption recognition, personalization frameworks as well as the investigation of logical or therapeutic procedures.

There is an expanded and well arranged pattern of using successive pattern mining in source code mining as well as programming detail mining. These devices change over the source code into a grouping database representation, as well as locate the successive patterns keeping in mind the end goal to recover distinctive data, for pattern, duplicate stuck code fragments, API utilization, programming rules as well as so on.

Consider a book shop, for pattern, Amazon can find that the clients who have first purchased a book 'Prologue to Data Mining" will frequently buy another book - Data Science for Business' in a later exchange. Store directors can utilize this data to

do proposal when a client peruses the information mining book. Another valuable application is to recognize piece get to patterns of plate frameworks. The piece get to patterns are helpful to anticipate the hinder that are gotten to next, as well as these pieces can be pre-fetched into reserve to decrease the plate I/O idleness. Consecutive pattern mining produces an exponential number of patterns when the database contains long groupings which are costly in both time as well as space. A similar issue likewise happens in itemset as well as chart mining when the patterns are long. A few sub-sequences have the bolster which is proportionate to the support of the long succession, which are fundamentally excess patterns. Consequently, rather than mining the entire arrangement of consecutive designs, it is ideal to mine shut successive patterns as it were.

A shut consecutive pattern is a consecutive pattern which has no super sequence with a similar support. Shut successive pattern mining widely diminishes the quantity of patterns created as well as it can be used to get the total arrangement of successive patterns. Furthermore, shut successive pattern mining calculations make utilization of pursuit space pruning strategies as well as beat successive design mining calculations. Shut consecutive pattern mining helps clients to discover all the more fascinating designs as well as decreases the weight of clients to investigate an excessive number of patterns.

II. LITERATURE REVIEW

In the year of December 2016, the authors "David Savage, Xiuzhen Zhang, Pauline Chou, Xinghuo Yu and Qingmai Wang", described in their paper titled "Distributed Mining of Contrast Patterns" such as a novel calculation is proposed for mining contrast designs utilizing an appropriated, outline like system. Differentiate designs portray contrasts between differentiated information sets and have beforehand been utilized for building profoundly precise classifiers. Be that as it may, digging for complexity examples is a computationally costly undertaking and existing calculations are intended to keep running in a successive way on a solitary machine. Therefore, existing methodologies can't deal with thick, high volume and high dimensional databases. Our calculation addresses this issue by dividing the scan space for complexity designs into little, autonomous units. These units can be mined in parallel, giving an adaptable answer for mining vast information sets. Utilizing three distinctive certifiable information sets we test an execution of our calculation on a Spark bunch. Aftereffects of these tests demonstrate that our calculation accomplishes a high-level of parallelism and versatility.

In the year of August 2016, the authors "Zhicheng Liu, Yang Wang, Mira Dontcheva, Matthew Hoffman, Seth Walker and Alan Wilson", described in their paper titled "Patterns and Sequences: Interactive Exploration of Clickstreams to Understand Common Visitor Paths" Present day web clickstream information comprises of long, high-dimensional successions of multivariate occasions, making it hard to examine. Taking after the general rule that the visual interface ought to give data about the dataset at various levels of granularity and permit clients to effortlessly explore over these levels, we distinguish four levels of granularity in clickstream investigation: designs, portions, groupings and occasions. We exhibit a scientific pipeline comprising of three phases: design mining, design pruning and facilitated investigation amongst examples and successions. In view of this approach, we talk about properties of maximal successive examples, propose techniques to decrease the quantity of examples and portray outline contemplations for envisioning the removed consecutive examples and the relating crude arrangements. We show the suitability of our approach through an investigation situation and talk about the qualities and impediments of the strategies in light of client criticism.

In the year of July 2016, the authors "Marwan Hassani, Yifeng Lu, Jens Wischnewsky and Thomas Seidl" described in their paper titled "A geometric approach for mining sequential patterns in interval-based data streams" such as all exercises saw in these days applications are connected with a planning succession. Clients are principally searching for fascinating groupings out of such information. Successive example mining calculations go for finding continuous groupings. For the most part, the mined exercises have timing lengths that speak to time interims between their beginning and closure focuses. The lion's share of consecutive example mining approaches managed such exercises as a solitary point occasion and along these lines lost important data in the gathered examples. As of late, some methodologies have painstakingly considered this interim based nature of the occasions, however they have significant impediments. They focus just on the request of occasions without

considering the spans of the holes amongst them and more often than not utilize a double representation to depict designs. To determine these issues, we propose the PIVOTMiner, an interim based information mining calculation utilizing a geometric representation approach of interims. Loud occasions can be presented with the geometric representation and a fluffy set can be recovered from the geometric examples. PIVOTMiner can adaptably chip away at information displayed as any number of not really adjusted interim groupings and specifically can use information exhibited as single interim arrangement stream without the need to make tests. Our test comes about on both manufactured and true shrewd home datasets demonstrate that the data introduced in our mined examples are wealthier than those of most best in class calculations while spending extensively littler running circumstances.

III. Problem Analysis

Following analysis, we first present some preparatory ideas, and afterward formalize the shut consecutive example mining issue. Let X ={s1,s2,… .,sm} be an arrangement of all things.

A subset of X is called an itemset. An arrangement Y=(h1,h2, … , hn) (hi ⊆ X) is a requested rundown of itemsets. The things in each itemset are sorted in alphabetic request. The length of the grouping is the aggregate number of things in the arrangement. An arrangement W1 =(q1,q2,… ..,qm) is a subsequence of another grouping W2 =(e1,e2,… .,en), indicated as W1 ⊑ E2, if there exists whole numbers 1 is x1 < x2 < . . . < hm & xm and s1 ⊆ si1 , s2 ⊆ Si2 , . . . , and Xm ⊆ Xim. We call W2 as a super-grouping of W1 and W2 contains X1.

IV. Proposed Method

For all many researchers found lots of pattern-mining procedures to analyze and mine the sequential patterns intelligently, however these algorithms are strucked up with certain levels and those cannot be proceeding further for long term sequences and huge datasets. So that a new pattern mining scheme is required to perform intelligent mining principles. In this paper we form an efficient pattern mining algorithm called "Subsequent CloSpan Algorithm [SCloSpan]", in which it provides an efficient decision tree reduction model with power duplicate content checking along with data stemming reduction principles, which provides earlier prediction of closed sequential patterns over the mining procedure.

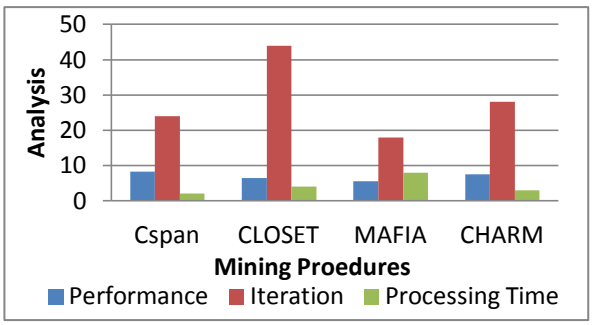

Figure.1 Evaluation of Various Pattern Mining Procedure

V. Dataset Manipulation

This approach is manipulated with powerful multi relational/social dataset. We display a system intended to mine successive fleeting examples from multi-social databases. Keeping in mind the end goal to endeavor rationale social data without utilizing accumulation philosophies, we change over the multi-social dataset into what we name a multi-grouping database. Every case in a multi-social target table is coded into a succession that joins intra-table and between table social fleeting data. This permits us to discover heterogeneous worldly examples through standard arrangement diggers. Our system is grounded in the fantastic outcomes accomplished by past propositionalization methodologies. We take after a pipelined approach, where we first utilize a grouping mineworker to discover visit successions in the multi-arrangement database.

SEQUENCE_ID	DATA SEQUENCE
1	[a][b,c]
2	[a, b][c]
3	[a, c][b]
4	[c][a, b]

Table-1. Example for Multi-Relational Sequence Data

Next, we select the most fascinating discoveries to enlarge the representational space of the illustrations.

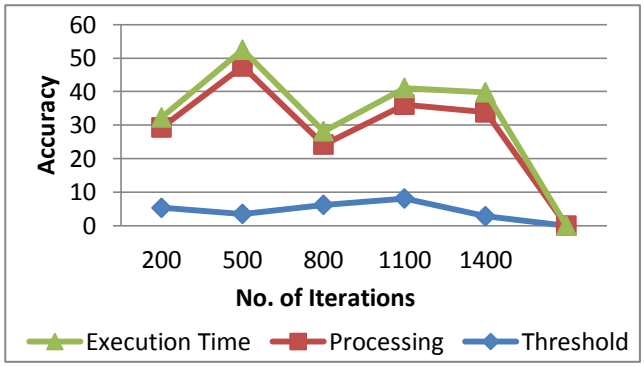

Figure.2 Multi-Relational Dataset Processing

The most intriguing succession examples are discriminative and class corresponded. In the last stride we fabricate a classifier demonstrate by taking a developed target table as contribution to a classifier calculation. We assess the execution of this work through a propelling application, the hepatitis multi-social dataset. We demonstrate the adequacy of our procedure by tending to two issues of the hepatitis dataset.

An arrangement database, FD={x1,x2,… ,xn}, is an arrangement of groupings and every succession has an id. The size, [FD], of the arrangement database FD is the aggregate number of groupings in the FD. The support of an arrangement W in a succession database FD is the no of groupings in FD which contain Q.

Definition-1 [Sequential Instance]: Given a base bolster limit m_sup, a grouping a will be a consecutive example on FD if bolster [an] is more noteworthy than m_sup.

Definition-2 [Closed consecutive Instance]: A successive example a will be a shut successive example if there does not exist a consecutive example b, with the end goal that support [a] = bolster [b] and a, b.

The issue of shut successive example mining is to locate the entire arrangement of shut consecutive examples over a base bolster edge m_sup for an info grouping database FD.

Table 1 demonstrates a specimen succession database. The things in each itemset are sorted in alphabetic request. In the event that m_sup=2 , the shut successive example set contains 3 arrangements and the relating consecutive example set contains 9 groupings. It demonstrates that shut successive example set contains less no of arrangements than consecutive example set.

VI. Subsequent Clospan Algorithm

In this system Subsequent CloSpan Algorithm [SCloSpan] is proposed to efficiently mining the Subsequent Closed Sequential Patterns over long datasets and huge processing units. This SCloSpan algorithm efficiently perform and providing decision tree reduction model with power duplicate content checking along with data stemming reduction principles, which provides earlier prediction of closed sequential patterns over the mining procedure.

This SCloSpan algorithm uses both depth and breadth first searching schemes to producing the sequential pattern with closed mining rules. Till now numerous modern methodologies are developed and they all portraited that these depth and

breadth first search schemes are more efficient than the classical searching techniques in mining area for large data pattern.

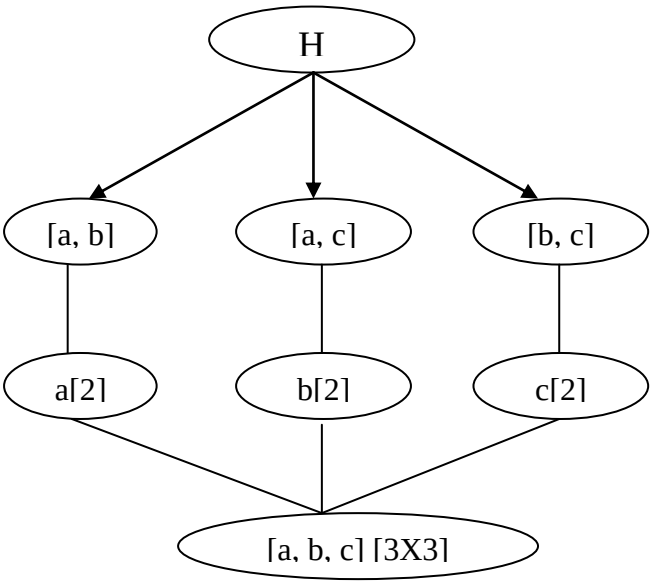

Figure.3 Subsequent CloSpan Algorithm Manipulation Structure

The SCloSpan algorithm uses a pattern structure with closed-sequence principles for producing the sequential pattern with closed structure. The sequential pattern structure is constructed by using the following way. The parent node of the structure is named as G. Subsequent levels of the structure are frequently named as 1st sequence, 2nd sequence and so on into the database, which are recogonized as well as created to next step of the sequential pattern structure.

Algorithm: SCloSpan

Input: Multi-Relational or Social Dataset
Output: Closed sequential patterns with Subsequent Strategies.
a. X1 = Subsequent Frequencies 1 to n
b. H2 = X1
c. for each g in X1 do
d. GH = Portraited dataset of g
e. H2 = GenerateSubsequentpatterns(GH, g, X1)
f. H1 = GH H2
g. close for
h. GH = g
i. K1 = Subsequent sets in GH
j. if Subsequent[g] H2 and X1 g then
k. if [g doesn't flow the apperance testing) then
l. if [g is closed with respect to GH] then
m. HD = H2 g
n. close if
o. close if
p. for each x in T1 do
q. XD = Portraited dataset of g s x
r. GD = GH GenerateSubsequentpatterns[HF, g s x, X1]
s. GH = Portraited databset of g i x
t. H2 = GD GenerateSubsequentPatterns [GH, g x T1, X1]
u. close for
v. close if
w. Pass H2

After that this sequence H at process step 1 will be enhanced with respect to adding a frequent-object in H portraited database for obtaining its subsequent pattern with fine principles at process step 2. A grouping can be reached out in either one of the following ways. They are: grouping augmentation and itemset expansion. If there should arise an occurrence of

succession expansion, the thing is included as a new itemset to the grouping. If there should arise an occurrence of itemset expansion, the thing is attached to the last itemset in the grouping. This procedure is rehashed for the successions at the level 2 or more until there are no super-groupings to produce.

VII. Experimental Analysis

The proposed SCloSpan algorithm runs on both real and synthetic data sets with various kinds of sizes and data distributions, and we compare SCloSpan with clasp, clospan and cspan. All experiments were conducted on a 2GHz Intel Core2 Duo processor PC with 1GB main memory running Microsoft Windows XP. The algorithms were implemented in Java and were executed using different support values.

In our experiments we used a real world dataset BMS WebView of KDD CUP. BMS WebView is a click stream data from an e-commerce web store named Gazelle and it has been used widely to assess the performance of frequent pattern mining. This dataset contains sequences of 59601 customers with a total of 146000 purchases in 497 distinct product categories. The average length of sequences is 2.42 items with a standard deviation of 3.22.

S. No.	Characteristic	Value
1	No of sequences	59601
2	No of distinct items	497
3	Average length of sequences	2.42

Table 2. Characteristics of the BMS WebView dataset

The experiments were conducted to evaluate the performance of the SCloSpan algorithm. The first set compares the runtime performance of Clospan,Clasp,Cspan and SClospan using BMS WebView for different support values.

COMPARISON WITH RESPECT TO RUN TIME(S)						
	Minimum Support values					
Algorithm	0.1	0.2	0.3	0.4	0.5	0.6
Clospan	17	15	6	4	3	2
Clasp	14	8	4	3	2	2
Cspan	12	6	3	2	1	1
SClospan	11	5	3	1	1	1

Table 3: Evaluation results respect to runtime

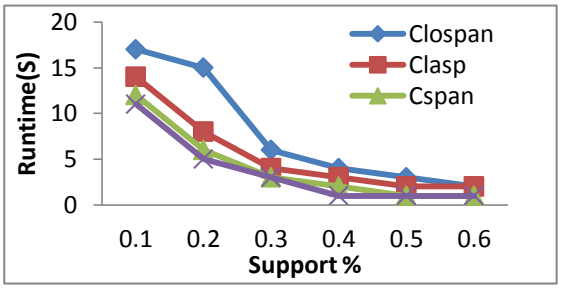

Figure 4: Performance comparison using runtime

Figure 4 show the results of runtime performance using the BMS WebView datasets. The X-axis is the minimum support, while the Y-axis is the algorithms runtime. The support values are set from 0.1 to 0.6. Our proposed algorithm SCloSpan outperforms CloSpan, ClaSP and cspan both the synthetic datasets.

COMPARISON WITH RESPECT TO SEARCH SPACE MEMORY(MB)						
	Minimum Support values					
Algorithm	0.1	0.2	0.3	0.4	0.5	0.6
Clospan	14	12	11	9	3	2
Clasp	12	9	7	4	2	2
Cspan	9	7	5	2	1	1
SClospan	7	5	4	1	1	1

Table 4: Evaluation results respect to memory

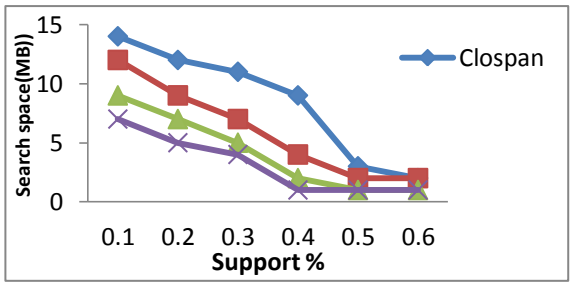

Figure 5: Performance comparison using memory

Figure 5 show the results of memory space performance using the BMS Web viewer datasets. The X-axis is the minimum support, while the Y-axis is the algorithms runtime. The support values are set from 0.1 to 0.6.

All the above experiments confirm that the proposed algorithm SCloSpan is efficient and takes less memory usage comparing to other algorithms. Because SCloSpan uses event watching that permits the early location of shut successive patterns amid the mining process and stores a result set that contains only closed sequential patterns, where as CloSpan, ClaSP and cspan keep a larger number of closed sequential pattern candidates and remove the non-closed ones at the end of the mining process.

VIII. CONCLUSION

A few researchers/scientists concentrated on the consecutive pattern mining issue and numerous calculations were created to mine successive patterns. Shut successive pattern mining is a variation of consecutive design mining and accomplishes expansive consideration in the late years since it has a similar expression capacity of the consecutive pattern mining and more minimal than the successive pattern mining. In this research work, a proficient calculation SCloSpan is proposed, which makes utilization of another pruning strategy called event watching that permits the early location of shut successive patterns amid the mining process. Our broad execution examine on different genuine and manufactured datasets appears that the proposed calculation SCloSpan outflanks the past works and an as of late proposed calculation Catch by a request of greatness. In further scenarios we will stretch out SCloSpan to consolidate client indicated requirements. Other fascinating inquire about issues that can be sought after incorporate parallel mining of shut successive patterns what's more, mining of organized patterns.

References

1. David Savage, Xiuzhen Zhang, Pauline Chou, Xinghuo Yu and Qingmai Wang, “Distributed Mining of Contrast Patterns” December 2016.
2. Zhicheng Liu, Yang Wang, Mira Dontcheva, Matthew Hoffman, Seth Walker and Alan Wilson, “Patterns and Sequences: Interactive Exploration of Clickstreams to Understand Common Visitor Paths” August 2016.
3. Marwan Hassani, Yifeng Lu, Jens Wischnewsky and Thomas Seidl, “A geometric approach for mining sequential patterns in interval-based data streams” IEEE Trans. Knowledge and Data Eng., July 2016.
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 patternprojected 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-fournierviger. 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.

