Heuristic-Based Clustering Approach for Discovering Colossal Patterns from High-Dimensional Databases

T. Sreenivasula Reddy, R. Sathya, and Mallikharjuna Rao Nuka

Abstract Since its introduction, frequent pattern mining in data mining has been extensively studied to discover relationships among variables in large databases. Majority of the frequent pattern mining algorithms have been developed based on the Apriori property which traverse iteratively over the item lattice in a level-wise approach. This phenomenon works well with small- and medium-size pattern sequences from the databases and does not contain any useful information for data analysts. However, for large patterns the runtime complexity will increase exponentially as the pattern sequence length increases. In this paper, a new Heuristic Clustering based Colossal Pattern mining from high dimensional datasets (HCCP). This paper proposes a strategy which avoids exhaustive level-wise pattern tree traversal and quickly mines colossal patterns. Our method performs binary clustering over sub-pattern**.** In our method, we used a lattice array to construct sub-pattern sequences along with support values. Further, these sub-patterns are explored as conditional patterns by estimating core patterns. Finally, colossal cluster is constructed and colossal patterns are discovered. Our experimental result on various input datasets shows that our HCCP achieves very high mining results. In fact, our analysis of results reveals that this algorithm outperforms with core fusion and colossal pattern miner (CPM) in diverse aspects.

Keywords Frequent pattern mining · Colossal patterns · Core fusion · Colossal pattern miner

T. Sreenivasula Reddy (✉)
Department of Computer Science and Engineering, SVPCET, Puttur, Andhra Pradesh 517518, India
e-mail: seenu4linux@gmail.com

R. Sathya
Department of Computer Science and Engineering, Annamalai University, Annamalai Nagar, Chidambaram, Tamil Nadu 608002, India

M. Rao Nuka
Department of Computer Applications, AITS, Rajampet, Andhra Pradesh 516115, India

V. Komanapalli et al. (eds.), *Advances in Automation, Signal Processing, Instrumentation, and Control*, Lecture Notes in Electrical Engineering 700,
https://doi.org/10.1007/978-981-15-8221-9_165

1 Introduction

In data mining, frequent pattern mining (FPM) has emerged to be one of the significant and vital mining methods after it is introduced in the research community. Frequent pattern mining is an important problem in association analysis [1]. The majority pattern kinds are association rule mining [2], correlations [3], closed patterns [4], classification [5], episodic [6], maximal patterns [7], clustering [8] and sequential patterns [9] .

The problem of colossal pattern mining is first identified and proposed an algorithm for effectively mining colossal patterns [10]. Algorithm is based on his observation that, every colossal pattern has enormous number of sub-patterns and he randomly selected sub-patterns and merged them to form candidate colossal patterns whose support is counted by individual database scans which are expensive. Since only large-scale patterns provide relevant and sufficient data in many implementations, it is better to develop a new kind of mining method that can discover only long-size patterns without mining small- and midsized patterns. Colossal patterns' mining is the new approach to discover long-size patterns without mining smaller patterns.

In this paper, a heuristic clustering approach for mining colossal patterns (HCCP) is presented. It uses a data structure called lattice array to identify long-size patterns known as colossal patterns by depth-wise item enumeration in each row and minimizes the search time on the dataset effectively. HCCP will be performed in the phases: first, the construction of a sub-pattern using a lattice array that fits into the memory; secondly, HCCP utilizes heuristic measures to find core patterns for each conditional sub-pattern which effectively reduce search time and database scans. Finally, colossal clusters are constructed from which un-rooted colossal patterns are discovered. We propose HCCP to build clusters out of itemsets, in such a way that the clusters are both common and interesting. The experimental results demonstrate better results of the approach when mining is applied on dense datasets and it is ascertained that HCCP significantly shows better performance with colossal pattern miner and core fusion on different settings.

The remaining of the paper is structured as follows: In Sect. 2, a related research work is presented. The HCCP heuristic clustering approach for mining colossal patterns is explained in Sect. 3. Experimental study is presented in Sect. 4, and finally we conclude the study in Sect. 5.

2 Related Research Work

In the literature, many algorithms are developed to identify both frequent and closed patterns under pattern growth approach [7]. This utilizes enumeration-based approaches [11], [12] to look for frequent colossal patterns in item combinations. With this in view, it increases its execution time exponentially with an increase in

the average record length which makes at least two searches over the database. It uses memory in large amount and also takes more execution time predictably in the case of memory constraints. The mining process starts with small patterns in these approaches and continues on to the larger one. Mid-size as well as small-size pattern doesn't contain any useful information and in some applications only large-size patterns will carry useful information and these large-size pattern are called Colossal patterns.

2.1 Mining Colossal Patterns

A colossal pattern is lengthy by nature; most of the colossal pattern's sub-patterns are expected to occur with almost the same frequency as the colossal pattern, and thus most of the colossal pattern's sub-patterns are identifiable based on their support counts. Form the literature, it can be found that for midsized patterns, the search space is exponentially booming. So we need to investigate a large number of smaller patterns to hit the colossal patterns. In order to accomplish colossal patterns, a methodology is suggested to traverse the search space to leap bypassing most of the midsized patterns. The main core fusion algorithm was the first serious algorithm for mining colossal patterns and was described in 2007 [10]. A central pattern-fusion approach which could give a good approximation is expressed in this article. The central idea of the current method is to merge small sub-patterns of large patterns as one phase.

Some more effectively colossal model mining approaches like colossal pattern miner (CPM) in 2010 preceded this work [12]. CPM recommends picking up the core pattern smartly rather than picking up a core pattern blindly. This approach provided a way to separate sub-patterns with interacting gigantic patterns depending on their frequencies, making it easier to jump through large quantity of midsize patterns. Sohrabi [13] suggested the BVBUC algorithm to mine all colossal patterns, which makes used of bit vectors, presenting patterns in every transaction and bottom-up traversing in vertical form is used to mine colossal patterns is suggested in [13]. The search space of BVBUC often decreases if *minSup* rises or the amount of transactions increases. An improved algorithm to mine colossal pattern sequences using doubleton pattern mining is presented in [14]. Also, the CP-Miner algorithm utilizes bottom-up approaches such as BVBUC. However to reduce runtime and memory utilization, it uses efficient pruning techniques.

3 HCCP: A Heuristic Clustering Approach for Colossal Patterns

In this chapter, we are studying efficient mining of colossal pattern from high-dimensional dataset. Table 1 provides an instance of the dataset where the attributes are represented from A to K. Let $C = \{1,2,3,4,5,6\}$ be a collection of experimental condition rows where each TID is a set of n subsets called attributes.

Definition 1: Itemset Support The number of transactions exactly contains set of items or attributes. It is denoted as IS(X).

Definition 2: Frequent Pattern In a given transaction database, a pattern(X) is frequent if sup(X) $\geq$ *minSup*.

Definition 3: Core Pattern In a given pattern Y, a pattern $X \subseteq Y$ is said to be a τ-core pattern of Y if sup(Y)/sup(X) $\geq \tau$, $0 < \tau \leq 1$(where τ is called core ratio).

Definition 4: Colossal Pattern A pattern X is called a colossal pattern in a frequent pattern if and only if there does not exist an itemset Y such that $X \subset Y$ and X is a τ-core pattern of Y.

Definition 5: Sub-pattern Support The number of times X appears as a strict sub-pattern of a frequent pattern. It is denoted as bs(X).

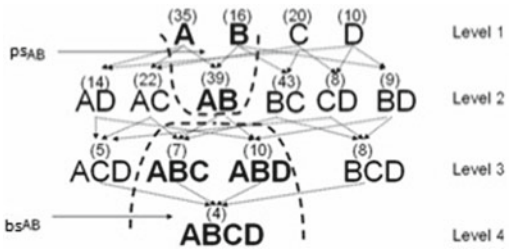
Definition 6: Super-pattern Support The number of times X is a super-pattern of a frequent pattern. It is denoted as ps(X).

3.1 Sub-pattern Construction Using Lattice Array

By constructing a sub-pattern lattice with vertical search techniques, this distinctive feature can reduce the amount of the mining process. Horizontal search strategy cannot carry out effective pattern mining since it is possible to detect exponential

Table 1 Sample transaction database DB

TID	Pattern attributes
1	A, B, C, E, G, H, I
2	A, C, D, E, F, H, J
3	B, E, F, G, H
4	A,B,C,D,E,F,G,K
5	A, B, D, F, G
6	B, E, G, H, I, J, K



.	
ABC	2
.	.
AD	3
AC	3
AB	3
.	.
D	3
C	3
B	5
A	4

Fig. 1 Lattice array construction for the sub-patterns

sequence of objects. We used vertical search strategy to produce sub-patterns from EDM. The difficulty in finding sub-patterns is that they cannot be identified explicitly, which reflects attributes over a finite time span, which may be less than the attributes. The attributes are included as Level 1 in the lattice array of one sub-pattern, and then the array of one sub-pattern can include many attributes not expressed as Level 2. Ultimately, the remaining attributes in one's sub-pattern array can be derived from other row id's transactional records with sub-pattern identical to Level *i* as shown in Fig. 1.

The mining algorithms of the association rule establish different frequency counts for items throughout a database scan, and it is difficult to load the page into the main memory. To optimize the main memory, a pattern in the dataset should be counted at one location. Lattice array (LA) is a more effective way of storing sub-pattern in memory [14]. A lattice array will store a count for the pair as LA [k], and K is given as

$$K = (i-1)\left(n - \frac{i}{2}\right) + (j-i)$$

We can carry out an efficient investigation on pattern sequences by imposing backtracking search order on column sets.

3.2 Estimating Core Patterns for Conditional Sub-patterns

We describe a colossal pattern contains the set of attributes that contain only core pattern sets. The colossal pattern sets are those which contain super-pattern support that is greater than the core pattern and should have sub-pattern support that is smaller than the core pattern which is less than the estimated support. We now compute core pattern for each conditional sub-pattern. The estimated bs and estimated ps are calculated using the IS_y. Given N transaction, the estimated item probability i is estimated as $p_i = (is_i + ps_i)/N$. The estimates are given as follows.

$$\text{Estimate } (\text{IS}_x) = N \prod_{i \in x} P_{ij \notin x}\left(1 - P_j\right)$$

$$\text{Estimate } (\text{bS}_x) = N \prod_{i \in x} P_i \; - \; \text{Estimate}(\text{IS}_x)$$

$$\text{Estimate}(\text{PS}_x) = N \prod_{j \in x}\left(1 - P_j\right)$$

A colossal cluster center is considered for each node in the lattice vector. A center must satisfy two important selection criteria. First, the sub-patterns in a set should appear very often with expectation; i.e., the ps_*ratio* = ps_x/*Estimate*(ps_x) should be maximum. Second, its strict super-pattern is less than the expected; i.e., the bs_*ratio* = bs_x/*Estimate*(bs_x) should be minimum. Now the core ratio for a sub-pattern set is core _*ratiox* (r) = ps_*ratio*–bs_*ratio*. The subsets with higher core ratio values are those pattern sets are larger than expected, i.e., the edge of the lattice and super-sets smaller than expected which is the bottom of the lattice.

3.3 Colossal Cluster Construction

Once the core patterns are calculated, we begin from the top level of the lattice and order nodes on the basis of their ratio values and conditioned sub-patterns within each level. Colossal clusters are created to classify sub-pattern sets after ordering the candidate centers. A ratio threshold, RT, is used only if the clusters include an amply huge number of data points. For each subset x, if $\text{ps}_x \geq RT$, x is marked as a confirmed sub-pattern set, and the subsets of x are allocated with support values. Each level in lattice array performs search reduction, if necessary sort sub-patterns on ratio. The threshold is experimentally determined by selecting the array that provides good value for cluster analysis on simulated results. For each sub-pattern on level i, all the sub-patterns whose threshold is above the min *RT* are clustered together as colossal cluster.

3.4 Extracting Colossal Patterns

We may extend each non-overlap cluster into un-rooted trees from the colossal clusters that have been discovered. Today, A can be enumerated, so a non-overlapping subset corresponding to pattern A can be enumerated. On the other side, for every A, if A belongs to all rows of Ci, that condition A will create the colossal pattern of A. So if we only expand the tree to the point of minsup, it will be able to discover all the colossal patterns The un-rooted tree searches only for minsup tree level that is

Table 2 Colossal pattern discovered from sub-pattern trees

S. no.	Colossal pattern sequence
1	{A,B,C,E,G}
2	{A,B,D,F,G}
3	{A,C,D,E,F}
4	{B,E,F,G}
5	{B,E,G,H}
6	{A,C,E,H}

explicitly conditioned on non-overlapping set and prunes its children. The colossal patterns extracted from the un-rooted sub-pattern trees are shown in Table 2.

4 Performance Evaluations and Experimental Analysis

The output of our three algorithms is compared and analyzed with the existing core fusion and CPM methods along our HCCP method in performance evaluation. Evaluation of the algorithm is done in terms of runtime. Four real datasets from [15] are taken for evaluating and comparing the performance our algorithms as shown in Table 3.

Figures 2, 3, 4 and 5 show the response time of HCCP with CPM and core fusion algorithms on datasets defined in Table 3. From Figs. 2, 3, 4 and 5, it is noticed that all the algorithm performances in all datasets will decrease with increasing minimal support. Nevertheless in HCCP, the amount of tree level created will also decrease when the minsup decreases. Therefore, if the minsup is small we have an improvement in HCCP efficiency. Figures 2, 3, 4 and 5 indicate that when the minsup is high, the gap in output of HCCP with CPM and core fusion is important. The percentage of improved results, however, often depends on each form of dataset, such as up to 25% improved on accident, 20% on pumsb*, 15% on retail and 30% on yeast, respectively. Compared to the decreased minimum support values, the more frequent1-itemsets and more time needed to process, the response time in both

Table 3 Datasets used

Dataset	Items	Transactions
Accident	468	340,183
Pumsb*	7117	49,046
Retail	16,469	88,162
Yeast	79	2467

*A well preprocessed dataset

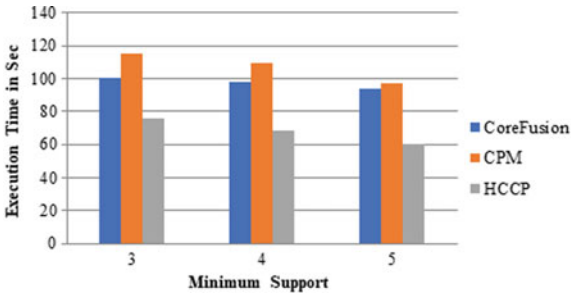


Fig. 2 Comparison of our HCCP with core fusion and CPM on accident dataset with different minimum supports

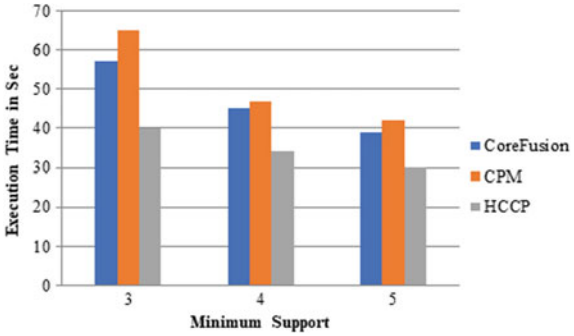


Fig. 3 Comparison of our HCCP with core fusion and CPM on pumsb* dataset with different minimum supports

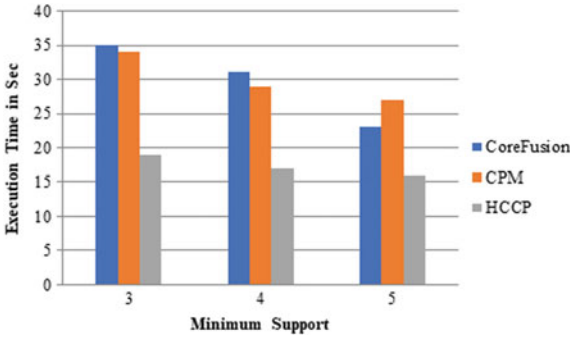


Fig. 4 Comparison of our HCCP with core fusion and CPM on retail dataset with different minimum supports

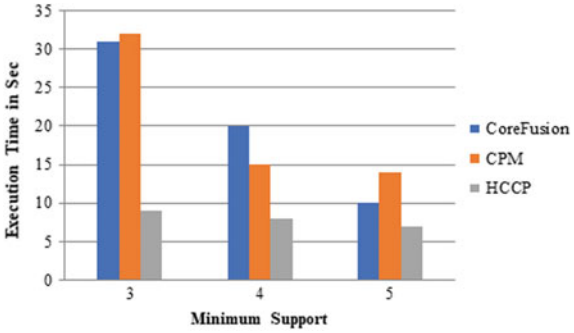


Fig. 5 Comparison of our HCCP with core fusion and CPM on yeast dataset with different minimum supports

approaches increased in all datasets when the minimum support values are decreased. Our HCCP outperformed the CPM in four of four datasets.

5 Conclusion

In this paper, we extensively studied our heuristic clustering-based colossal pattern mining from high-dimensional datasets (HCCP). Core patterns are selected by estimating its core ratio using statistical heuristics rather than random selection. Lattice array is used to separate sub-patterns which overlap colossal patterns depending on their ration threshold. During the phases of HCCP, sub-pattern at the top of the lattice satisfying the minimum threshold ratio is clustered together and colossal patterns are discovered. The empirical analysis reveals that high-dimensional databases, HCCP, outperform other algorithms. The results showed that our approach to HCCP performed very well on four datasets, which surpassed those of the approach to CPM and core fusion. Compared to the low support values, the more frequent1-itemsets and more time needed to process, the response time in both approaches increased in all datasets when the minimum support values are decreased. This research work can be further extended on gene expression data analysis.

References

1. Prasanna K, Seetha M, Kumar APS (2014) CApriori: conviction based apriori algorithm for discovering frequent determinant patterns from high dimensional datasets. In: 2014 IEEE International Conference on Science Engineering and Management Research (ICSEMR-14), pp 1–6
2. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: VLDB'94, pp 487–499

3. Brin S, Motwani R, Silverstein C (1997) Beyond market basket: generalizing association rules to correlations. In: Proceedings of the ACM-SIGMOD international conference on management of data, pp 265–276
4. Bayardo RJ (1998) Efficiently mining long patterns from databases, In: SIGMOD'98, pp 85–93
5. Cheng Y, Church GM (2000) Biclustering of expression data. In: Proceedings of the 8th international conference on intelligent systems for molecular biology
6. Manila H, Toivonen H, Verkamo AI (1997) Discovery of frequent episodes in event sequences. Data Min Knowl Disc 259–289
7. Pei J, Han J, Mao R (2000) CLOSET: an efficient algorithm for mining frequent closed itemsets. In: Proceedings 2000 ACM-SIGMOD on DMKD'00, pp 11–20
8. Prasanna K., Seetha M (2012) Mining high dimensional association rules by generating large frequent k-dimension set. In: 2012 IEEE International Conference on Data Science and Engineering (ICDSE-2012). pp 58–63
9. Pei J, Han J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu M-C (2001) PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: ICDE'01, pp 215–224
10. Zhu F et al Mining colossal frequent patterns by core pattern fusion. In: Proceedings of the 2007 international conference on data engineering, Istanbul, Turkey
11. Pan F, Cong G, Tung AKH, Yang J, Zaki MJ (2003) CARPENTER: finding closed patterns in long biological datasets In: Proceedings ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD)
12. Dabbiru M, Shashi M (2010) An efficient approach to colossal patternmining. Int J Comput Sci Netw Secur. (IJCSNS) 6:304–312
13. Sohrabi MK, Barforoush AA (2012) Efficient colossal pattern mining in high dimensional datasets. Proc J Knowl Based Syst 33: 41–52
14. Prasanna K, Seetha M (2015) Efficient and accurate discovery of colossal pattern sequences from biological datasets: a doubleton pattern mining strategy (DPmine). Proce Comput Sci 54:412–421
15. Frequent Itemset Mining Implementations Repository: https://fimi.cs.helsinki.fi/