Discovering Erasable Closed Patterns

Giang Nguyen[1], Tuong Le[2(✉)], Bay Vo[1], and Bac Le[3]

[1] Faculty of Information Technology, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam
{nh.giang,vd.bay}@hutech.edu.vn
[2] Division of Data Science and Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
lecungtuong@tdt.edu.vn
[3] Faculty of Information Technology, University of Science, VNU Ho Chi Minh City, Vietnam
lhbac@fit.hcmus.edu.vn

Abstract. Data mining that discovers knowledge from large datasets is more and more popular in artificial intelligence. In recent years, the problem of mining erasable patterns (EPs) has been proposed as an interesting variant of frequent pattern mining. There are many algorithms for solving effectively the problem of mining EPs. However, for very big datasets, the large number of EPs takes the large memory usage of the system, and then obstructs users' using the system. Therefore, it is necessary to mine a condensed representation of EPs. In this paper, we present the erasable closed patterns (ECPs) concept and an effective algorithm for mining ECPs (MECP algorithm). The experimental results show that the number of ECPs is much less than that of EPs. Besides, the runtime of MECP is better than the naïve approach for mining ECPs.

Keywords: Data mining · Erasable closed patterns · Erasable patterns

1 Introduction

Data mining is the process of discovering interesting patterns in large dataset. These patterns will be used as the knowledge in the some intelligent systems such as expert systems, recommendation systems etc. Many problems in data mining have attracted research attention such as association rule mining [1-2, 18-19] and classification [6, 13-14]. Pattern mining including frequent pattern mining [7, 11, 16-17, 20], frequent closed pattern mining [15] etc. is an essential task in association rule mining. Recently, the problem of mining erasable patterns (EPs) [3-4, 5, 8-10, 12] proposed by Deng et al. [4] is an interesting variation of pattern mining. For details, a factory produces many products created from a number of items. Each product brings an income to the factory. A financial resource is required to buy and store all items. However, in a financial crisis situation, this factory has not enough money to purchase all necessary items as usual. This problem of mining erasable patterns is defined as the following: find the patterns which can best be erased so as to minimize the loss to factory's gain. The managers can then utilize the knowledge of these erasable patterns to make a new production plan. Currently, there are many algorithms for solving this problem such as

N.T. Nguyen et al. (Eds.): ACIIDS 2015, Part I, LNAI 9011, pp. 368–376, 2015.
DOI: 10.1007/978-3-319-15702-3_36

META [4], MERIT [3] and MEI [8] algorithms. Deng et al. (2009) proposed META, an Apriori-based algorithm, to solve the problem of mining erasable patterns. However, the execution time of META is slow because it uses a generate candidate approach which is a naïve strategy. MERIT [3] uses the concept of NC_Sets to reduce memory usage. Although the use of NC_Sets gives MERIT some advantages over META, there are still some disadvantages. First, the weight value of each node code (NC) is stored individually even though it can appear in many erasable patterns' NC_Sets, leading to a lot of duplication. Second, it uses a strategy whereby pattern sX's NC_Set is assumed to be a subset of pattern Y's NC_Set if $X \subset Y$. This leads to high memory consumption and high run time when patterns are combined to create new nodes. MEI [8] uses the dPidset structure to quickly determine the information of erasable patterns. Although mining time and memory usage are better than those of the above algorithms, MEI's performance for mining erasable patterns can be improved.

However, in the problem of mining EPs, the number of obtained EPs from these algorithms is so large. Therefore, intelligent systems using EPs as knowledge will be difficult situation with the large number of EPs. In reality, the set of erasable closed patterns (ECPs) can represent the set of EPs without losing information. This set does not have two or more EPs which have the same gain. They can be used to mine the non-redundant rules without pruning the redundant rules. Therefore, this paper presents ECPs concept and proposes an effective algorithm (MECP algorithm) for mining ECPs.

The rest of the paper is organized as follows. Section 2 presents basic concepts. The definition of ECPs and some theorems for fast mining ECPs are presented in Section 3. Section 4 introduces MECP algorithm for mining ECPs and an illustrated example of the MECP algorithm's process. Experimental result is presented in section 4 to show the effectiveness of MECP algorithm. The paper concluded in section 5.

2 Basic Concepts

2.1 Erasable Patterns

Deng et al. [4] proposed the problem of erasable pattern mining as follows. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of all items and DB is a product dataset. Each product presented in the form of $\langle Items, Val \rangle$, where $Items$ are the items to conduct this product and Val is the profit that the factory obtains by selling this product. Table 1 presents an example dataset (DB_E) which will be used throughout this article.

Table 1. An example dataset (DB_E)

Product	Items	Val ($)
P_1	a, b	1,000
P_2	a, b, c	200
P_3	c, e	150
P_4	b, d, e, f	50
P_5	d, e	100
P_6	d, e, f, h	200

Definition 1. Given a threshold ξ and a product dataset DB. A pattern X is erasable if:

$$g(X) \leq T \times \xi \tag{1}$$

where:
- $g(X) = \sum_{\{P_k | X \cap P_k.Items \neq \emptyset\}} P_k.Val$ is gain of patterns X;
- $T = \sum_{P_k \in DB} P_k.Val$ is total gain of the factory.

Based on Definition 1, the problem of mining EPs is to find all EPs which have gain $g(X)$ less than $T \times \xi$ in dataset.

Example 1. We have $g(e) = P_3.Val + P_4.Val + P_5.Val + P_6.Val = 150 + 50 + 100 + 200 = 500$ dollars and $T = 1{,}700$ dollars. With $\xi = 30\%$, e is a EPs because $g(e) = 500 \leq T \times \xi = 510$ dollars.

2.2 dPidset Structure

Le and Vo [8] proposed the dPidset structure for effectively mining erasable pattern as follows.

Definition 2 (pidset). The pidset of a pattern X is denoted as follows:

$$p(X) = \bigcup_{A \in X} p(A) \tag{2}$$

where A is an item in pattern X and $p(A)$ is the set of product identifiers which includes A.

Definition 3 (dPidset). The dPidset of pidsets $p(XA)$ and $p(XB)$, denoted as $dP(XAB)$, is defined as follows:

$$dP(XAB) = p(XB) \setminus p(XA) \tag{3}$$

According to Definition 3, the dPidset of $p(XA)$ and $p(XB)$ is the product identifiers which only exist on $p(XB)$.

Theorem 1. Let XA and XB be two patterns and $dP(XA)$ and $dP(XB)$ be the dPidsets of XA and XB, respectively. The dPidset of XAB is computed as follows:

$$dP(XAB) = dP(XB) \setminus dP(XA) \tag{4}$$

Theorem 2. The gain of XAB is determined based on that of XA as follows:

$$g(XAB) = g(XA) + \sum_{P_k \in dP(XAB)} P_k.Val \tag{5}$$

where $g(XA)$ is the gain of XA and $P_k.Val$ is the gain of the product P_k.

Example 2. We have $p(e) = \{3, 4, 5, 6\}$ and $p(c) = \{2, 3\}$, so $g(e) = 500$ and $g(c) = 350$. We have $dP(ec) = \{2\}$ so $g(ec) = g(e) + \sum_{P_k \in dP(ec)} P_k.Val = 500 + 200 = 700$ dollars.

3 Erasable Closed Pattern Mining

3.1 Erasable Closed Patterns

Similar to the definition of frequent closed patterns [15], an erasable pattern is called an erasable closed pattern if none of its supersets has the same gain. For example, consider DB_E with $\xi = 30\%$, e and $edfh$ are two erasable patterns because $g(e) = g(edfh) = 500 \leq 1700 \times 30\% = 510$ dollars. However, e is not an erasable closed pattern because $edfh$, one of its supersets, has the same gain as e.

When combining two elements X and Y in the same equivalence class, the algorithm will check their *dPidsets*. There are four cases as follows. If $dP(XA) = dP(XB)$ then remove XB and replace XA by $XA \cup XB$. If $dP(XA) \subset dP(YB)$, the algorithm will replace XB by $XA \cup YB$. Conversely, no element XA or XB can be removed.

3.2 MECP Algorithm

Using *dPidset* concept and its theorems for ECPs, we propose the MECP (Mining Erasable Closed Patterns) algorithm for mining ECPs in Fig. 1.

Algorithm 1. MECP algorithm

Input: product database *DB* and threshold ξ
Output: E_{result}, the set of all ECPs

```
1  Scan DB to determine the total profit of DB (T),
   the index of gain (G), and erasable 1-patterns with
   their pidsets (E1)
2  Sort E1 by the length of their pidsets in decreasing
   order
3  If E1 has more than one element, the algorithm will
   call Expand_E(E1)

1  Procedure Expand_E(Ev)
2  For i ← 0 to |Ev| do
3  Begin for
4     Enext ← ∅
5     For j ← i+1 to |Ev| do
6        dP(ECP) = dP(Ev[j]) \ dP(Ev[i])
7        If ECP.val <= ξ × T then
8          If dP(Ev[i]) = dP(Ev[j]) then
9            Ev[i] = Ev[i] ∪ Ev[j]
10           Update Enext
```

```
11          Remove E_v[j]
12          j--
13        Else if dP(E_v[i]) ⊂ dP(E_v[j]) then
14          E_v[i] = E_v[i] ∪ E_v[j]
15          Update E_next
16        Else
17          ECP = E_v[i] ∪ E_v[j]
18          E_next ← ECP
19    End for
20    If Check_Closed_Property(E_v[i]) == true then
21      E_result ← E_v[i]
22      Add E_v[i] to Hashtable with E_v[i].val as a key
23    If |E_next| > 1 then
24      Sort E_next by the length of their dPidsets in de-
        creasing order
25      Expand_E(E_next)
26  End for

1  Function Check_Closed_Property(EI)
2  Let ECPs ← Hashtable[EI.val]
3  If ECPs is not null then
4    For each ECP in ECPs do
5      If EI ⊂ ECP then
6        Return false
7  Return true
```

Fig. 1. MECP algorithm

4 Experimental Results

All experiments presented in this section were performed on a laptop with an Intel Core i3-3110M 2.4-GHz CPU and 4 GB of RAM. MEI and MECP algorithms were coded in C# and .Net Framework Version 4.5.50709.

The experiments are conducted on Chess, Mushroom and Connect datasets[1]. To make these datasets look like product datasets, a column was added to store the profit of products. To generate values for this column, a function denoted by $N(100, 50)$, for which the mean value is 100 and the variance is 50, was created. The features of these datasets are shown in Table 2.

[1] Downloaded from http://fimi.cs.helsinki.fi/data/

Table 2. Features of datasets used in experiments

Dataset[2]	# of Products	# of Items
Chess	3,196	76
Mushroom	8,124	120
Connect	67,557	130

4.1 The Number of ECPs and EPs

Table 3 shows the number of ECPs and EPs on Chess, Mushroom and Connect datasets. The number of ECPs is clearly smaller than that of EPs. Therefore, the required resource in the intelligent systems is also reduced which make these systems better.

Table 3. The number of ECPs and EPs on experimental datasets

Dataset	Threshold ξ (%)	Number of EPs	Number of ECPs
Chess	10	665	523
	15	3,083	2,082
	20	10,913	6,260
	25	30,815	15,637
Mushroom	0.75	1,830	209
	1.0	8,368	549
	1.25	24,537	1,262
	1.5	63,033	2,716
Connect	0.75	1,677	1,644
	1.0	5,185	4,892
	1.25	13,625	12,220
	1.5	30,540	25,817

4.2 The Runtime

Currently, the problem of mining ECPs is unsolved. To evaluate MECP algorithm, we conduct a naïve approach which mines all EPs and then finds ECPs from the obtained EPs. This section reports the mining time of MECP algorithm and the naïve approach which show in Figs. 2-4. The experimental results show that the mining time of MECP algorithm outperforms the naïve approach.

[2] These datasets are available at http://sdrv.ms/14eshVm

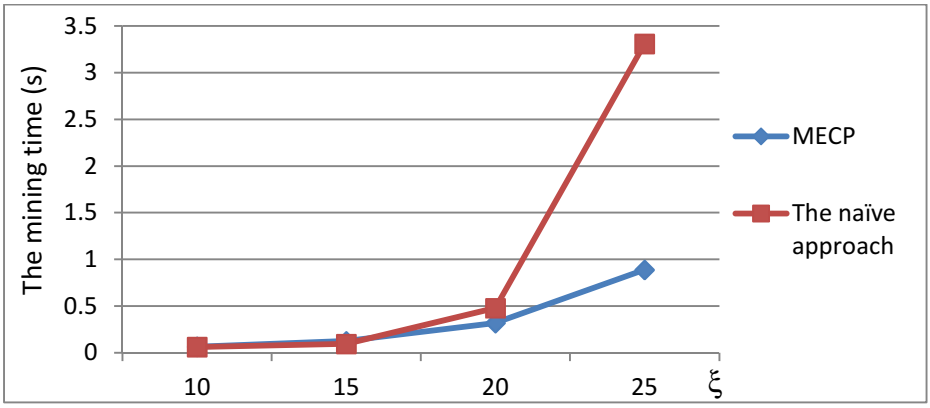

Fig. 2. The mining time of MECP and the naïve approach on Chess dataset

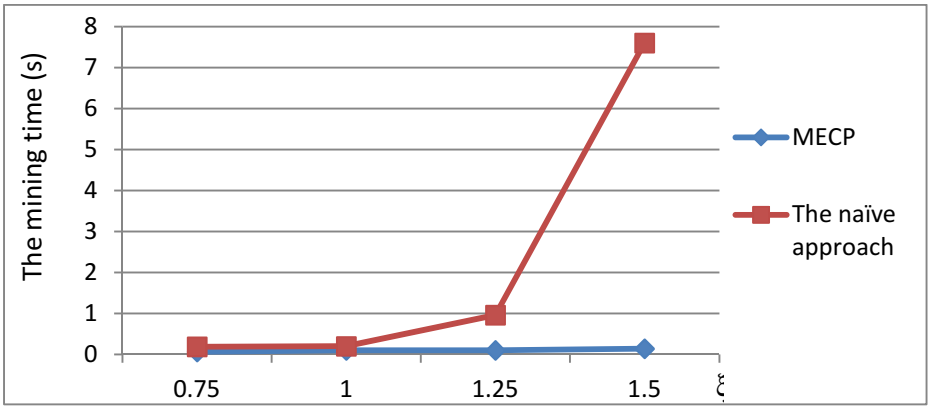

Fig. 3. The mining time of MECP and the naïve approach on Mushroom dataset

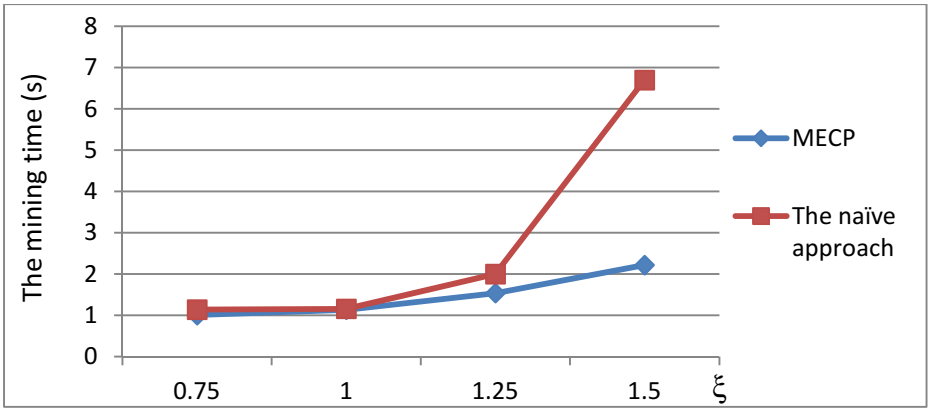

Fig. 4. The mining time of MECP and the naïve approach on Connect dataset

5 Conclusion and Future Work

Erasable pattern mining is an interesting problem is introduced in 2009. Up to now, there are many algorithms for solving effectively this problem such as META, MERIT, dMERIT+ and MEI. However, a small number of EPs is used in intelligent systems. Therefore, there is necessary to remove redundant EPs. In this paper, we present the erasable closed patterns (ECPs) concept and MECP for mining ECPs. The experiment was conducted to compare the numbers of patterns and the mining time. The results show that the number of ECPs is much smaller than the number of EPs and the mining time of ECPs is better than the naïve approach.

In future work, some issues related to EPs will be studied, such as mining EPs from huge datasets, mining top-rank-*k* EPs, mining maximal EPs, and mining EPs from incremental datasets.

Acknowledgments. This research was funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2012.17.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB 1994, pp. 487–499 (1994)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between set of items in large databases. In: SIGMOD 1993, pp. 207–216 (1993)
3. Deng, Z.H., Xu, X.R.: Fast mining erasable itemsets using NC_sets. Expert Systems with Applications **39**(4), 4453–4463 (2012)
4. Deng, Z.H., Fang, G., Wang, Z., Xu, X.: Mining erasable itemsets. In: ICMLC 2009, pp. 67–73 (2009)
5. Deng, Z., Xu, X.: An efficient algorithm for mining erasable itemsets. In: Cao, L., Feng, Y., Zhong, J. (eds.) ADMA 2010, Part I. LNCS, vol. 6440, pp. 214–225. Springer, Heidelberg (2010)
6. Do, T.N., Lenca, P., Lallich, S.: Classifying many-class high-dimensional fingerprint datasets using random forest of oblique decision trees. Vietnam Journal of Computer Science, DOI:10.1007/s40595-014-0024-7 (in press)
7. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: SIGMOD 2000, pp. 1–12 (2000)
8. Le, T., Vo, B.: MEI: an efficient algorithm for mining erasable itemsets. Engineering Applications of Artificial Intelligence **27**, 155–166 (2014)
9. Le, T., Vo, B., Nguyen, G.: A survey of erasable itemset mining algorithms. WIREs Data Mining Knowl. Discov. **4**, 356–379 (2014)
10. Lee, G., Yun, U., Ryang, H.: Mining weighted erasable patterns by using underestimated constraint-based pruning technique. Journal of Intelligent and Fuzzy Systems (2014, in press)
11. Huynh, T.L.Q., Vo, B., Le, B.: An efficient and effective algorithm for mining top-rank-k frequent patterns. Expert Syst. Appl. **42**(1), 156–164 (2015)

12. Nguyen, G., Le, T., Vo, B., Le, B.: A New Approach for Mining Top-Rank-*k* Erasable Itemsets. In: Nguyen, N.T., Attachoo, B., Trawiński, B., Somboonviwat, K. (eds.) ACIIDS 2014, Part I. LNCS, vol. 8397, pp. 73–82. Springer, Heidelberg (2014)
13. Nguyen, D., Vo, B., Le, B.: Efficient strategies for parallel mining class association rules. Expert Syst. Appl. **41**(10), 4716–4729 (2014)
14. Nguyen, L.T.T.: Mining class association rules with the difference of obidsets. In: Nguyen, N.T., Attachoo, B., Trawiński, B., Somboonviwat, K. (eds.) ACIIDS 2014, Part II. LNCS, vol. 8398, pp. 72–81. Springer, Heidelberg (2014)
15. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Beeri, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 398–416. Springer, Heidelberg (1998)
16. Song, W., Yang, B., Xu, Z.: Index-BitTableFI: An improved algorithm for mining frequent itemsets. Knowledge-Based Systems **21**, 507–513 (2008)
17. Vo, B., Coenen, F., Le, T., Hong, T.-P.: Mining frequent itemsets using the N-list and subsume concepts. International Journal of Machine Learning and Cybernetics DOI:10.1007/s13042-014-0252-2 (in press)
18. Vo, B., Hong, T.-P., Le, B.: A lattice-based approach for mining most generalization association rules. Knowledge-Based Systems **45**, 20–30 (2013)
19. Zaki, M.J.: Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering **12**(3), 372–390 (2000)
20. Zaki, M.J., Gouda, K.: Fast vertical mining using diffsets. In: SIGKDD 2003, pp. 326–335 (2003)