Chapter 33
Multiple-Threshold Erasable Mining Under the Tightest Constraint

Tzung-Pei Hong, Yi-Chen Chang, Wei-Ming Huang, and Wen-Yang Lin

Abstract Erasable data mining has become increasingly important in recent years because it can provide helpful suggestions for material procurement in factory manufacturing. In traditional erasable data mining, only a single threshold is set to judge whether an itemset is erasable. In this paper, we extend the problem to multiple thresholds and propose an algorithm to solve it by considering the tightest constraint. We prove the problem under the tightest constraint has the downward-closure property, and thus, we can search the solution space efficiently. Experimental results also show the effectiveness and efficiency of the proposed approach.

T.-P. Hong (✉) · W.-Y. Lin
National University of Kaohsiung, Kaohsiung 811, Taiwan
e-mail: tphong@nuk.edu.tw

W.-Y. Lin
e-mail: wylin@nuk.edu.tw

T.-P. Hong · Y.-C. Chang
National Sun Yat-Sen University, Kaohsiung 804, Taiwan
e-mail: 4a0g0902@stust.edu.tw

W.-M. Huang
Department of Electrical and Control, China Steel, Inc, Kaohsiung 806, Taiwan

K. Kondo et al. (eds.), *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, Smart Innovation, Systems and Technologies 339,
https://doi.org/10.1007/978-981-99-0105-0_33

33.1 Introduction

Data mining technology has been an important research field in analyzing big data [1]. It can help obtain valuable information or knowledge that people cannot easily discover at first glance. With the vigorous development of the industry, more and more factories are adopting data mining technology, expecting to solve various problems happening on the production side. The erasable itemset mining was proposed for achieving the above purpose [2]. It is used to deal with the problem that when all raw materials cannot be purchased, we must decide which materials to be erased, but it does not affect the profit much. It may happen, for example, when the factory encounters insufficient funds, limited logistics cargo capacity, insufficient storage space, etc.

Traditional erasable itemset mining uses a single threshold to decide erasable materials (items), that is, an itemset with a profit loss less than a given threshold will be regarded as erasable. However, different items usually hold different characteristics and considerations, generating different decision criteria. Therefore, in this paper, we consider the multiple threshold erasable itemset mining problem. We use the tightest constraint to set the threshold for an itemset with more than two items and show it has the property of downward closure. We then modified the original erasable-itemset mining algorithm to handle the problem. At last, we conduct experiments to show the performance of the proposed approach.

33.2 Related Works

In this section, we review some related works to the paper. They include erasable data mining and multiple threshold mining.

33.2.1 Erasable Itemset Mining

Erasable itemset mining is often used in factory production management [3]. It was first introduced by Deng et al. in 2009 [2]. They also proposed an algorithm called META to solve it [2]. In 2010, Deng and Xu then proposed the Vertical format-based algorithm for Mining Erasable Itemsets (VME) approach with a list structure called PID_List [4]. In 2012, Deng and Xu designed the MERIT algorithm, which uses a tree structure [5]. After that, Le et al. proposed the MERIT + algorithm that was based on MERIT to improve the mining performance [6]. In 2014, Le and Vo proposed an effective itemset screening algorithm called MEI [7]. It adopts the concept of divide-and-conquer. Hong et al. then adopted the bitmap approach to speed up the mining process [8]. Hong et al. considered the incremental processing for erasable itemset mining [9]. Besides, Hong et al. proposed modified erasable

itemset mining for processing quantitative product databases [10]. Vo et al. proposed efficient algorithms for mining erasable closed patterns from product datasets [11]. Hong et al. then considered the temporal issues of erasable mining [12].

33.2.2 Multiple Threshold Mining

The multiple threshold concept was originally introduced to solve the problem in frequent mining [1]. B, Liu et al. [13] discovered that rare items were never found in the database and thus proposed multiple thresholds for different itemsets. Lee et al. then proposed a strict constraint of multiple thresholds for frequent itemset mining [14], which picks the maximum support threshold among different items. To express general constraint, Wang et al. introduced a new mechanism that used bins and an enumeration tree structure [15]. They allowed users to assign arbitrary aggregation functions of multiple thresholds. Yang et al. mined partial periodic patterns with individual event support thresholds [16]. Lin et al. handled utility mining using multiple minimum utility thresholds [17]. Huang used multiple thresholds in temporal fuzzy utility mining [18].

33.3 Problem Description

In the erasable mining problem, a product database is given. Each tuple in the database includes the product name, the items (materials) to produce the product, and the profit the product can earn. An example of a product database is given in Table 33.1.

The total gain of a product database represents the sum of all product profits. For example, in Table 33.1, the total gain value is 200 + 200 + 100 + 100 + 300 + 100, which equals 1000. When a particular item (material) cannot be purchased or stocked, it will cause the products that need to be produced with this material to be unable to be manufactured. The total loss caused by these products that cannot be manufactured is called the gain.

Table 33.1 Example of a product database

Product database		
PID	Items	Profit
$Product_1$	ABE	200
$Product_2$	DEF	200
$Product_3$	BCE	100
$Product_4$	ADF	100
$Product_5$	BF	300
$Product_6$	ACDF	100

Table 33.2 Maximum thresholds of the items in the above example

Item	*A*	*B*	*C*	*D*	*E*	*F*
λ	0.5	0.4	0.3	0.6	0.2	0.7

For example, in Table 33.1, there are three products containing item $\{A\}$, which are $Product_1$, $Product_4$, and $Product_6$. Adding up the profits of the three products, we may get $\text{Gain}(A) = 200 + 100 + 100 = 400$.

In the original erasable itemset mining problem, an itemset is called erasable if its gain ratio is larger than a single maximum threshold, which ranges from 0 to 1 and is preset by users. This paper will consider the multiple thresholds because different items usually hold different characteristics. For example, the maximum threshold values of the items for the above product database may be given, as given in Table 33.2, where λ represents the maximum threshold mapping for each item. For example, in Table 33.2, $\lambda(A) = 0.5$ and $\lambda(B) = 0.4$.

After the maximum thresholds are set for the items, it is easy to judge whether a 1-itemset is erasable by comparing its gain ratio with its own maximum threshold. However, different constraints may be given for judging an itemset with two or more items. In this paper, we adopt the tightest constraint because it possesses the property of downward closure and can easily be used. By the constraint, the minimum of the maximum thresholds of the items in an itemset will be used to judge whether the itemset is erasable. Formally, the formula of the tightest constraint for an itemset X is defined as follows:

$$\lambda(X) = \min(\lambda(i) | i \in \text{X}).$$

For example, the 2-itemset $\{A, B\}$ contains both items A and B. Its maximum threshold is then set as $\min(\lambda(A), \lambda(B))$, which is $\min(0.5, 0.4) = 0.4$. Since the gain ratio is obtained by dividing the gain of an itemset by the total gain, we may derive the maximum gain threshold (MGT) as the total gain multiplied by the maximum threshold (ratio).

For the multiple threshold erasable mining under the tightest constraint, we prove the downward-closure property holds. That is, if an itemset is erasable under the tightest constraint, then all of its sub-itemsets are erasable as well. This property will be used in the algorithm below to increase search efficiency.

33.4 The Proposed Algorithm Under the Tightest Constraint

In this section, we will introduce in detail how the algorithm works. The corresponding pseudo-code is shown below.

The algorithm

1. **Input:** A product database *PD* and a set of thresholds for the items
2. **Output:** The set of erasable itemsets under the tightest constraint
3. $k = 1$ // the amount of items in the itemset
4. *total_profit* = 0
5. **For** each item $i \in$ the product database *PD* **do**
6. *Gain(i)* = 0
7. **End For**
8. **For** each product p in *PD* **do**
9. *total_profit* = *total_profit* + p.profit
10. **For** each item $i \in$ product p **do**
11. *Gain(i)* = *Gain(i)* + p.profit
12. **End For**
13. **End For**
14. $EI_1 = \varnothing$
15. **For** each item i in *PD* **do**
16. *MGT(i)* = *total_profit* × $\lambda(i)$ // Maximum gain threshold
17. **If (**Gain(i) ≤ MGT(i)) **then**
18. $EI_1 = EI_1 \cup i$ // Erasable 1-itemset
19. **End If**
20. **End For**
21. $EI = EI_1$
22. **While** ($EI_k \neq$ **NULL**) **do**
23. k++
24. $EI_k = \varnothing$
25. CI_k = generate_candidate_k-itemset (EI_{k-1}) // Candidate k-itemset
26. **For** each candidate itemset $c \in CI_k$ **do**
27. *Gain(c)* = 0
28. **For** each product p in *PD* **do**
29. **If** ($c \cap p.items \neq$ **NULL**) **then**
30. *Gain(c)* = *Gain(c)* + p.profit
31. **End If**
32. **End For**
33. **If** ($Gain(c) \leq (min(MGT(i)) \mid \forall\, i \in c$)) **then**
34. $EI_k = EI_k \cup c$
35. **End If**
36. **End For**
37. $EI = EI \cup EI_k$ // Erasable itemset
38. **End While**
39. **Return** *EI*

In the above algorithm, the total profit and the actual gains of all the 1-itemset are calculated in Lines 7–12. The maximum gain threshold for each item with the user-presetting threshold is calculated, and each 1-itemset is judged to be erasable or not in Lines 15–20. The candidate k-itemsets are generated from the erasable $(k-1)$-itemsets in a way similar to the Apriori algorithm in Line 25. The step is based on

the proven downward closure for the tightest constraint. Next, the candidate itemsets are judged to be erasable or not in Lines 26–36. All the erasable itemsets found at each level are output as the final mining result in Line 39.

33.5 Experiments

Experiments were performed on a synthetic dataset, T10I4N0.03KD100K, by the IBM data generator to evaluate the proposed approach for the multiple threshold erasable mining under the tightest constraint. The parameter T denotes the average number of items in each product, I denotes the size of a maximal potentially erasable itemset, N denotes the number of items, and D denotes the number of products in a dataset. Each tuple was thought of as a product. We also modified the tuples to fit the problem of erasable itemset mining by generating product profits randomly. We compared the proposed approach for multiple thresholds and fixed single thresholds. The multiple thresholds are randomly generated within an interval of 0.3. Two single fixed thresholds are used, one is the minimum of the interval, and the other is the maximum.

The numbers of erasable itemsets mined for different threshold intervals are shown in Fig. 33.1. As expected, the line of the numbers of erasable itemsets mined for multiple thresholds lies in the middle of the other two lines for single thresholds. But it is much closer to the line with the single threshold set as the minimum of a threshold interval than to the one with the single threshold set as the maximum of a threshold interval.

The execution times of the proposed algorithm for different threshold intervals are shown in Fig. 33.2. The results are very consistent with those in Fig. 33.1.

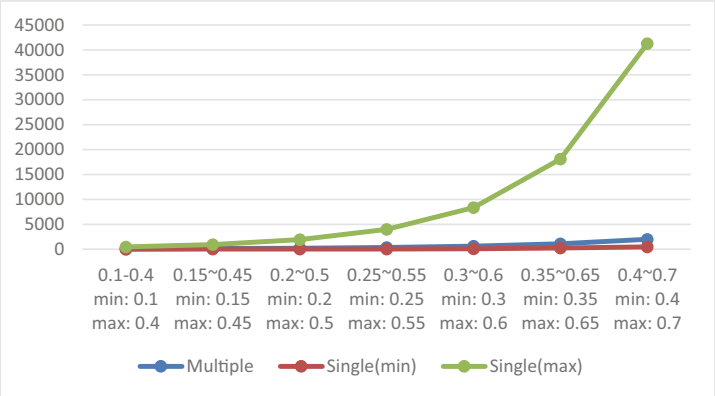

Fig. 33.1 Numbers of erasable itemsets for different threshold intervals

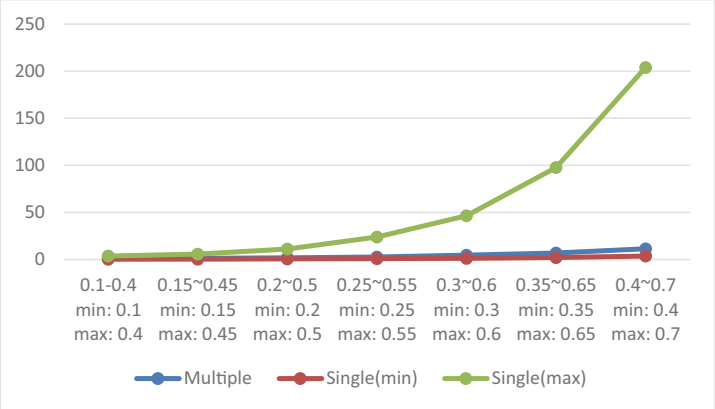

Fig. 33.2 Execution time for different threshold intervals

Since the execution time is significantly related to the number of candidate itemsets, we record the number of candidate itemsets in the mining process. The results are shown in Fig. 33.3, where the unit is a second. Again, the trend of the lines is similar to that in Fig. 33.2, which explains why the execution time in Fig. 33.1 has such behavior.

Finally, we measure the memory usage of the proposed approach with different threshold intervals. The results are shown in Fig. 33.4, where the unit is a kb. Again, the results are very consistent with those in Fig. 33.3 because the most memory usage is for storing and processing candidate itemsets.

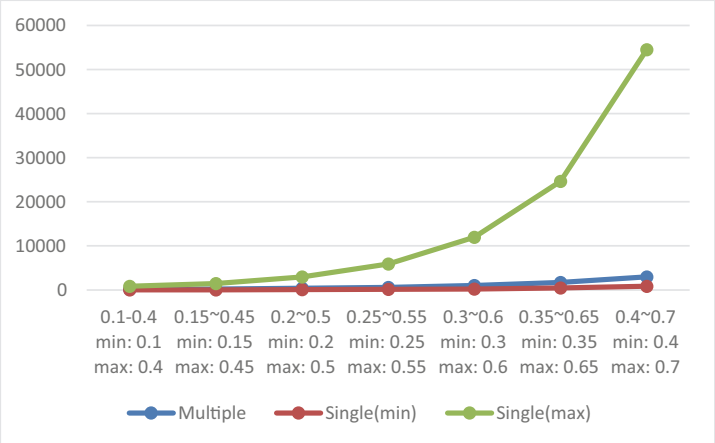

Fig. 33.3 Number of candidate itemsets for different threshold intervals

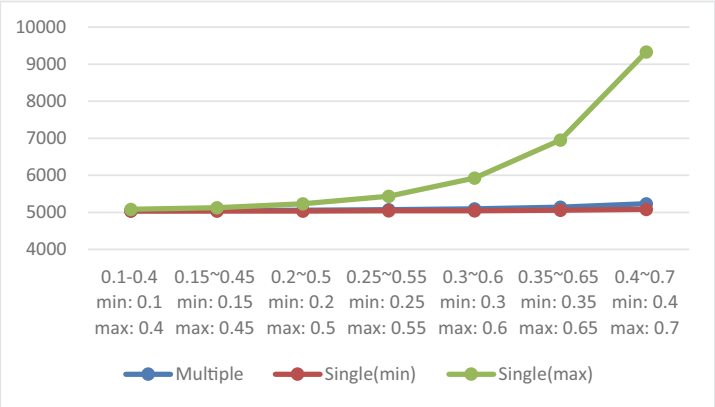

Fig. 33.4 Memory usage for different threshold intervals

33.6 Conclusion and Future Work

In this paper, we have used the tightest constraint to deal with the erasable mining problem with multiple thresholds. We have proved the downward-closure property holds under the tightest constraint, and thus, we can efficiently handle the problem in an Apriori way to prune the candidate search space fast. We experimentally compared the proposed approach for multiple thresholds and fixed single thresholds. Experimental results show that the line of the numbers of erasable itemsets mined for multiple thresholds lies in the middle of the other two. But it is much close to the line with the single threshold set as the minimum of a threshold interval because of the tightest constraint. Besides, all the measures, including the number of erasable itemsets, execution time, number of candidate itemsets, and memory usage, show the same trend. Thus, considering the tightest constraint for multiple thresholds has the advantages of simplicity and efficiency. In the future, we will conduct more experiments to verify the proposed approach and generalize it to other constraints.

Acknowledgements This work was supported by the grant NSTC 109-2221-E-390-013-MY2, the National Science and Technology Council, Taiwan.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: The International Conference on Very Large Data Bases, vol. 1215, pp. 487–499 (1994)
2. Deng, Z.H., Fang, G.D., Wang, Z.H., Xu, X.R.: Mining erasable itemsets. In: The 2009 International Conference on Machine Learning and Cybernetics, vol. 1, pp. 67–73 (2009)
3. Le, T., Vo, B., Nguyen, G.: A survey of erasable itemset mining algorithms. Data Min. Knowl. Disc. **4**(5), 356–379 (2014)

4. Deng, Z.H., Xu, X.R.: An efficient algorithm for mining erasable itemsets. In: The 6-th International Conference on Advanced Data Mining and Applications, pp. 214–225 (2010)
5. Deng, Z.H., Xu, X.R.: Fast mining erasable itemsets using NC_sets. Expert Syst. Appl. **39**(4), 4453–4463 (2012)
6. Le, T., Vo, B., Coenen, F.: An efficient algorithm for mining erasable itemsets using the difference of NC-Sets. In: The IEEE International Conference on Systems, Man, and Cybernetics Manchester, pp. 2270–2274 (2013)
7. Le, T., Vo, B.: MEI: An efficient algorithm for mining erasable itemsets. Eng. Appl. Artif. Intell. **27**, 155–166 (2014)
8. Hong, T.P., Huang, W.M., Lan, G.C., Chiang, M.C., Lin, C.W.: A bitmap approach for mining erasable itemsets. IEEE Access **9**, 106029–106038 (2021)
9. Hong, T.P., Lin, K.Y., Lin, C.W., Vo, B.: An incremental mining algorithm for erasable itemsets. In: The 2017 IEEE International Conference on Innovations in Intelligent Systems and applications (INISTA), pp. 286–289, Poland (2017)
10. Hong, T.P., Chen, H.W., Huang, W.M., Chen, C.H.: Erasable pattern mining with quantitative information. In: The 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taiwan (2019)
11. Vo, B., Le, T., Nguyen, G., Hong, T.P.: Efficient algorithms for mining erasable closed patterns from product datasets. IEEE Access **5**, 3111–3120 (2017)
12. Hong, T.P., Chang, H., Li, S.M., Tsai, Y.C.: A dedicated temporal erasable-itemset mining algorithm. In: The 21st International Conference on Intelligent Systems Design and Applications (ISDA), pp. 977–985. World Wide Web (2021)
13. Liu, B., Hsu, W., Ma, Y.: Mining association rules with multiple minimum supports. In: The 1999 International Conference on Knowledge Discovery and Date Mining, vol. 99, pp. 337–341 (1999)
14. Lee, Y.C., Hong, T.P., Lin, W.Y.: Mining association rules with multiple minimum supports using maximum constraints. Int. J. Approximate Reasoning **40**(1–2), 44–54 (2005)
15. Wang, K., He, Y., Han, J.: Pushing support constraints into association rules mining. IEEE Trans. Knowl. Data Eng. **15**(3), 642–658 (2003)
16. Yang, K.J., Hong, T.P., Lan, G.C., Chen, Y.M.: Efficient mining of partial periodic patterns with individual event support thresholds using minimum constraints. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **22**(6), 793–814 (2014)
17. Lin, C.W., Gan, W., Fournier-Viger, P., Hong, T.P., Zhan, J.: Efficient mining of high-utility itemsets using multiple minimum utility thresholds. Knowl. Based Syst. **113**, 100–115 (2016)
18. Huang, W.M., Hong, T.P., Chiang, M.C., Lin, C.W.: Using multi-conditional minimum thresholds in temporal fuzzy utility mining. Int. J. Comput. Intell. Syst. **12**(2), 613–626 (2019)