International Journal of Research in Advent Technology, Vol.6, No.12, December 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

A Survey of Frequent Itemset Mining

U. Mohan Srinivas[1], E. Sreenivasa Reddy[2]
Research Scholar[1], Professor[2], Department of Computer Science and Engineering[1,2], ANU College of Engineering and Technology[1,2], Acharya Nagarjuna University[1,2]
Email: umohansrinivas@gmail.com[1],esreddy67@gmail.com[2]

Abstract-Frequent Itemset Mining (FIM) become an important subfield of data mining. FIM is a process of extracting or discovering hidden, unknown and interested patterns from a collection of transactions of a customer database. Initially, it was designed for Market Basket Analysis to discover a group of items which are exhibiting the same behavior means that appearing frequently together. Though it was designed for Market Basket Analysis, it became as popular and used as a general task for finding a group of attribute values that are occurring together frequently. Thus, it led to the many applications like web click, recommending products; e-learning, text mining, bio-informatics and stream analysis. This paper discusses survey of itemset mining that includes introduction, techniques, research advances and opportunities in the field. The problem statement of itemset mining is presented, and approaches for mining itemset mining is also provided, as well as characteristics and limitations are also presented. In addition to that, extensions to the itemset mining is also presented such are closed, Maximal, rare itemset and high-utility itemset mining. And also discusses the opportunities in the field of data mining.

Keywords - Frequent Itemset Mining, Apriori Algorithm, and Minimum threshold.

1. INTRODUCTION

Data mining is an important task in Knowledge Discovery in Database (KDD) as shown in Fig1. The main goal of the data mining is to predict the future by analyzing or understanding the past data. Several techniques used in data mining, which contains designing a model that can predict the future such as Neural networks, discovering patterns in the data which are useful and hidden that are understand by humans. Approaches for discovering patterns can be classified as the type of the patterns they discover. And some of the patterns are itemsets, clusters, outliers and trends. This paper focuses only on discovery of itemsets in transactional database.

The process of discovering itemsets in database is called Frequent Itemset Mining (FIM), which was introduced by Agrawal and Srikanth in 1993 [1]. It is defined as follows, for a given set of customer transactions, FIM derives group of items that are appeared frequently together known as large itemsets that are used in finding the association between items called as Association Rule Mining (ARM) [1,2, 16]. Association between itemsets helps the store manager to take strategic decisions for enhancing the sale.

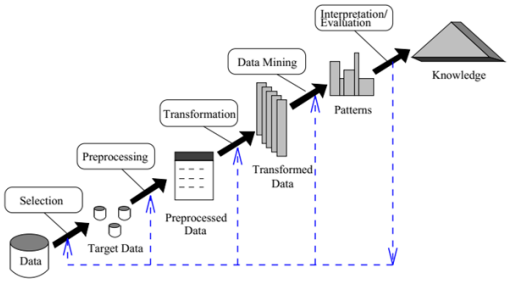

Fig 1: KDD [11]

Initially FIM was introduced to derive large itemsets in a database of customer transactions, which consists of set of transactions that are recorded with attribute values. Thus, FIM is a task is used to discover attribute values that are appearing frequently together in the database. It had led to the many domain applications where the attribute values are recorded as a transaction. For example, network traffic analysis, gene analysis in bio informatics, Analyzing customer reviews as Text Mining, Recommendation systems, malware detection, Activity monitoring. FIM has been extended to discover correlated patterns, ordered patterns in sequences and rare patterns.

In this paper, firstly, we address FIM problem statement, then main techniques employed, and then extensions that cause new problems.

2. FREQUENT ITEMSET MINING

The problem of FIM is formally defined as follows [Agarwal 94]. Let $I = \{i_1, i_2, \ldots, i_m\}$ be an itemset that contains a set of symbols. D be a transaction database {T1, T2, T3, … ,Tn} of transactions Tq such that each transaction $Tq \subseteq I (1 \leq q \leq m)$ is a set of distinct items, and each transaction is associated with a unique identifier called its Transaction Identifier (TID). For example, consider the transaction database TDB shown in Table 1. TDB contains five transactions, where each transaction is recorded with the item names I1, I2, I3, I4 and I5. An itemset X is a set of items such that $X \subseteq I$. The notation |X| denote the cardinality or the number of items in an itemset X. The notation k-itemset denotes the length of itemset whose length is k. The goal of itemset mining is to

derive interesting itemsets in a TDB, which is used to derive interesting associations between items.

Table 1. A sample Transaction Database (TDB)

Transaction ID	List of item IDs
1	I1, I2, I5
2	I2, I4
3	I2, I3
4	I1, I2, I4
5	I1, I3
6	I2, I3
7	I1, I3
8	I1, I2, I3, I5
9	I1, I2, I3

In general, in itemset mining, various measures can be used to find the interestingness of patterns. In FIM, support is the interestingness measure which is used widely. The support of an itemset X in a database TDB is denoted as sup(X) and defined as the number of transactions are containing X, that is sup(X) = |{Tk ⊆Tk ^TkϵTDB}|. For example, the support of the itemset {I1, I2} is 4 because this itemset appears in two transactions T1, T4, T8 and T9. Support of an itemset is also defined as a ratio. This definition called the relative support is rSup(X) = sup(X)=|TDB|. For example, the relative support of the itemset {I1, I2} is 0.44. The itemset X is said to be frequent if its support is no less than the minsup which is given by the user (user threshold value). The goal of FIM is to derive such kind of frequent itemsets in TDB w.r.t minsup. For example, if you consider the transactional database shown in Table 1 and minsup is 2, FIM task is to find the itemsets whose occurrence is at least two transactions and the result is {I1}:6, {I2}:7, {I3}:6, {I4}:2, {I5}:2, {I1,I2}:4, {I1,I3}:4, {I1,I5}:2, {I2,I3}:4, {I2,I4}:2, {I2,I5}:2, {I1,I2,I3}:2 and {I1,I2,I5}:2.

The general and Naïve approach of FIM task is to consider all the possible combinations and then result the itemsets whose support is greater than or equal to the minsup given by the user. It is required to enumerate all the patterns that lead to a huge search space. Hence such kinds of approaches are inefficient, because, 2m-1 possible combination are required for m items. If TDB contains 100 items, then the possible number of itemsets in search space is 2100-1. Hence it is difficult to handle many itemsets. It is necessary to design efficient approaches that explore less search space and less time for discovering itemsets in TDB.

Several algorithms have been proposed for FIM. Most of them are Apriori [1], FP-Growth [12], Eclat[38], LCM [34] and H-Mine [23]. All the mentioned algorithms have the same input and output, but they differ from the following. 1. Strategies: whether they use candidate generate and test based approach or Tree based approach 2. Type of the data structure that they used to keep the itemset information. 3. Counting mechanism for finding the support of itemset.

The rest of the paper is presented as follows, first we discuss Strategies in FIM, then discuss various techniques proposed for FIM. Strategies in FIM:

Most of the FIM techniques follow either Breadth-first or Depth-first search (BFS/DFS) for exploring frequent patterns with in the search space. Another name for breadth-first search is Level-wise algorithm, which follows the exploration of the itemsets at level by level. BFS explores the search space by considering 1-length then 2-length and K-length. It uses Hash diagram to represent search space. In addition to that, some optimizations are introduced in various algorithms such as Apriori and its extensions. Second one is, depth-first search, starts from 1-length and generates large itemset by appending item to the current itemset in a recursive manner.

From the above discussion, to reduce the search space, pruning strategies are used in FIM. One of the pruning technique is, for any itemsets X and Y such that $X \subset Y$, then it follows that $sup(X) \geq sup(Y)$. Thus, it is implied that if an itemset support is less than **minsup** means that it is infrequent, then all of its supersets also infrequent. This property is called down-ward closure property or anti-monotonicity or Apriori property.

3. APRIORI [1]

It is the basic algorithm in FIM, which takes input as minsup and database that is represented transactions in horizontally, output as frequent itemsets. Apriori follows level wise approach in which maintains all possible combinations and then result the itemsets whose support reaches minsup. Apriori scans database TDB once to find the cumulative support of 1-itemset. It uses the same information to identify the 1-length frequent itemset F1. Then apriori uses breadth first search to find next length large frequent itemsets. And it uses k-1 length frequent itemsets to generate the possible frequent itemsets length Ck. It uses the following procedure for generating CK, Let say {I1,I2} and {I1,I3} are two frequent 2-itemsets, it combines into a single possible 3-itemset iff they share common k-1 length items. For that example, it generates {I1,I2, I3} 3-itemset. To avoid unnecessary itemsets, it checks whether (k-1) subsets of K are frequent or not, if they are considered, otherwise are not considered. This kind of pruning strategy is called downward-closure property. Then it scans TDB to calculate the support of Ck and continues the same procedure to find Ck+1 until no candidates are generated.

Apriori result for the TDB of Table 1 is visualized in Fig 2.

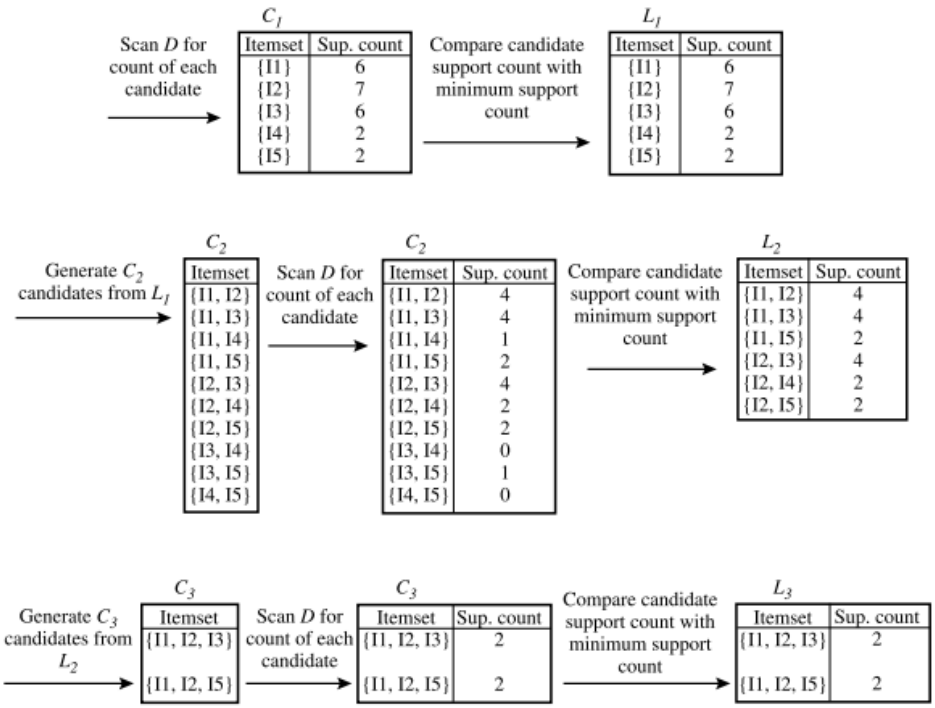

Fig. 2. Apriori demonstration of Table 1.

Apriori has inspired many algorithms. However, it suffers from the following limitations.

Limitations of Apriori [4]:

Apriori generates too many candidates while exploring the search space, which can take huge storage space and more time processing.

It scans TDB repeatedly to calculate the cumulative support of all itemsets, which is very costly.

Especially when the minsup is low, BFS search can be quietly costly in terms of storage space. Because, when minsup is low, too many k and k-1 length possible itemsets are to be maintained in search space. In other words we can say that time complexity of algorithm is $O(m^2n)$, when m is the number of distinct items and n is the number of transactions [13].

4. ECLAT: DEPTH-FIRST SEARCH ALGORITHM

Eclat [38] algorithm improves the performance of Apriori by using DFS to avoid keeping too many itemsets in search space. It uses vertical database representation; list of transactions where each item appears is maintained. For example, itemset I, the list of transactions where item I is appeared, it is called TID list and it is denoted as tid(i). tid list can be obtained by scanning the data base that is shown in figure. This vertical representation is very helpful to reduce number of scans over the database. For ex: Support count of {I1, I2} can be calculated from the tid(I1) ∩ tid(I2) which is { T4, T8, T9}. For calculating the above large itemset support, database scan is not required. It is also helpful to calculate the sup(X)=|tid(X)|. For the above example, sup({I1,I2})=3. Thus, using these properties, Eclat can discover all frequent itemsets by limiting the database scans to single.

Limitations of Eclat:

Though it is faster than Apriori, it is also employing candidate generation and test-based approach, but is also processing the itemsets which are not in database.

If the minsup is low, it can consume more space for TID-list.

Improvements of Eclat:

Diffset [39] enhances Eclat algorithm efficiency by introducing new structure. Bitvector [18, 39] approach is proposed to overcome the memory issue of Eclat by encoding TID-lists.

Table 2: Vertical representation-TID set

Item(i)	TID(i)
I1	{T1, T4, T5, T7, T8, T9}
I2	{T2,T3, T4, T6, T8, T9}
I3	{T3, T5, T6, T7, T8,T9}
I4	{T2, T4}
I5	{T1, T8, T9}

The 2-itemsets in vertical data format.

itemset	*TID_set*
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}

The 3-itemsets in vertical data format.

itemset	*TID_set*
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

Fig 3: ECLAT description of Table 1

5. PATTERN-GROWTH ALGORITHMS

To address the limitations of apriori, Eclat, Pattern-growth algorithms such as FP-growth [12], H-Mine [23], LCM [34]. The main idea of this approach is scanning the database once to find the 1-length itemsets, then use compact tree to keep itemsets to avoid candidate generation. To reduce number of scans, projected database is introduced and employs depth-first search for deriving larger itemsets.

The summary of FIM techniques have been presented in table 3

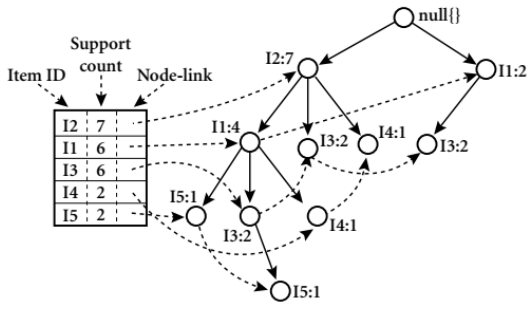

Fig 4: FP Tree structure of Table 1

Algorithm	**Type of search**	**Database representation**
Apriori[1]	breadth-first (candidate generation)	Horizontal
Apriori - TID[1]	breadth-first (candidate generation)	Vertical (TID-lists)
Eclat [38]	depth-first (candidate generation)	Vertical (TID-lists, diffsets)
FP – Growth [12]	depth-first (pattern-growth)	Horizontal (prefix-tree)
H – Mine [23]	depth-first (pattern-growth)	Horizontal (hyperlink structure)
LCM [34]	depth-first (pattern-growth)	Horizontal (with transaction merging)

Table 3: Summary of FIM Approaches

6. VARIATION OF FIM

This section discusses some of the limitations of FIM then alternative solutions and its methods.

Generally, the output of FIM that are frequent patterns are depends on the database and minsup. It may give many itemsets where human may feel difficult to analyze. And also, itemsets may contains redundancy which is difficult for decision makers. For ex: From the figure 2, it is observed that itemsets {I4}:2 and {I2, I4}:2 are frequent, but both of them are containing I2. To reduce the redundancy, researchers have proposed algorithms to extract concise representations. A concise representation is a set of frequent itemsets that are similar and summarize all the frequent itemsets. Discovering these kinds of patterns which can summarizes other helps to reduce the search space and faster than frequent itemsets. And lossless concise representations those are Maximal and Closed frequent itemsets. The concise representations of frequent itemsets are listed as follows.

6.1. *Maximal Itemsets [34]:*

Maximal itemsets are the frequent itemsets that do not have any supersets. These kinds of itemsets are also called as the largest frequent itemsets. It can be represented as MI $\subseteq$ L, where MI is the Maximal itemsets and L is the set of frequent itemsets. When MI are subsets of L, and X is a frequent itemset and all its subsets are also frequent. From the above two points, it can say that MI reduces the search space. Hence extensive research [34] has been carried to discover Maximal itemsets. However, MI cannot recover their original support.

6.2. *Closed Itemsets [3, 20, 35,40]:*

Closed itemsets are represented with a set of frequent itemsets that do not have supersets with the same support. Thus, search space can be reduced by discovering closed itemsets, which are lossless and concise representation of frequent itemsets. Generally closed itemsets are represented with CI and they are able to recover the whole frequent itemsets. In other words it can be said that CI ⊆ L. Hence, extensive research [40] has been carried out by researchers with many algorithms and data structures.

6.3. *Generator Itemsets [7, 25, 27]:*

Generator itemsets are the set of frequent itemsets that do not have any subsets with the same support, that is i.e. GI ={X|XϵL ∧∃XϵL} such that Y ⊂ X ∧ sup(X) = sup(Y)}. the support of the closed itemsets are always same or larger than closed and maximal itemsets. Since it is the main reason for moving frequent itemsets into closed itemsets, it had driven researchers into various algorithms and data structures. Similar to other, it is also suffers from an opinion that whether the itemsets are interested or not.

6.4. *Constraints on Itemsets:*

To reduce and avoid frequent itemsets that are less interesting patterns, constrains are introduced to filter such kind of itemsets. General and the naïve approach is applying constraints on FIM as post processing. But this kind of approaches may suffer from huge space and more time. Hence it is important to filter such king of itemsets while discovering itemsets. It has motivated researches to come with new algorithms and data structures. Some of the constraints are listed below.

Occupancy [30]: is a measure that is used to identify the itemsets that occupy the large portion of transactions. Various measures have been presented [4, 8, 26] to assess how itemsets are correlated each other. Examples are bond [8], Affinity [36], All-confidence [19], Coherence and Mean [4,26].

In FIM, constraints are categorized into several categories [10] such are monotone, anti-monotone, succinct [5, 21, 22] and convertible [21]. Anti-monotone property is widely used in FIM to reduce the search space. It is defined as, if X is infrequent then all of its supersets are infrequent.

6.5. *Rare Itemsets [17]:*

In real world applications, items are often different from each other with different frequency. One should not expect that they have the same frequency since one may be very common items and other may not be common items. Thus, it leads to the discovery of rare itemsets, are itemsets that are less likely to appear in frequent itemsets than others. Generally finding rare itemsets is more difficult than FIM since usually finds more rare itemsets. Hence, Researchers have proposed various algorithms for deriving rare itemsets [14, 28, 29].

Another important limitation of traditional FIM algorithms is the database format. As previously explained, FIM assumes that the input database only contains binary attributes (items). But in real-life this assumption does not always hold. Thus, several extensions of FIM have been proposed to handle richer database types. Some of the most important ones are the following.

6.6. *Weighted itemset mining:*

It is an extension of FIM, where items are associated with weights that indicate importance [31, 32, 37]. The primary task weighted itemset mining is to find the itemsets that have a minimum weight. It is also extended to mine infrequent weighted itemsets.

The above approaches in FIM consider item frequency rather than their profit. Some items like cars may carry less frequency but gives more profit to the organizations. It is not possible with the traditional FIM approaches. The following approach is proposed to address the same.

6.7. *High-utility itemset mining (HUIM)*

HUIM is an extension of FIM and weighted itemset mining where weights and purchase quantities are considered rather than only weights [17,37]. In HUIM, weights could indicate the profit/quality of the item and quantity indicates the number of units bought for each item. HUIM calculates the utility of item/itemset from the product of profit and quantity of item in a transaction. It results the itemsets as high if their utility is not less than the given minimum utility which is given by the user. The goal of HUIM is to find all itemsets that have a utility higher than a given threshold in a database. A major challenge in HUIM is that the utility measure is neither monotone nor anti-monotone [17].

Hence, the utility measure cannot be directly used to prune the search space. To solve this problem, the concept of upper-bound TWU is introduced in Two-Phase algorithm. The concept of tighter upper-bounds on the utility is introduced to prune a larger part of the search space, and improve the performance of HUIM algorithms [7, 9, 11, 37 and 41]. One of the fastest HUIM algorithms is EFIM [14]. Various extensions of the HUIM are proposed to find shelf-time periods of items [FV-15], discount strategies [15], and also to discover the top-k most profitable itemsets [6, 33].

7. RESEARCH OPPORTUNITIES

Although, FIM and its related mining techniques are so popular more than 20 years,still attracted

bymany applications to continue research. This section classifies research opportunities in the data mining field.

- Novel applications. Pattern mining algorithms are quite general, they can be applied in a multitude of domains. Some of the domains are, social network analysis, the Internet of Things, sensor networks. The easiest way is to carry research is apply existing pattern mining algorithms in new ways in terms of application domains.
- Enhancing the performance of pattern mining algorithms. Some of the pattern mining algorithms are quite time consuming on dense databases and efficient on sparse databases. Others are good in sparse and time consuming in dense databases. Hence, a lot of research is carried on developing more efficient algorithms. This is an important problem especially for new extensions ofthe pattern mining problem such as uncertain itemset mining or high-utility itemsetmining, which have been less explored. Many opportunities also in distributed,GPU, multi-core or parallel algorithm development to increase speed and scalability ofthe algorithms.
- Extending pattern mining to consider more complex data. Another research opportunity is to develop pattern mining algorithms on complex data. Researchers have focused on mining spatial patterns [24].

Also, another research opportunity is to work on novel interesting measures that can give more interesting and useful patterns.

8. CONCLUSION

Frequent Itemset Mining is an active filed in data mining. This paper has presented the problem statement of Frequent Itemset Mining, several techniques are discussed for exploring itemsets, and extensions are also discussed to improve the performance of FIM. Also, this paper has discussed, several extensions to the FIM to overcome limitations and to meet application need. This paper also discussed the various research opportunities in the area of data mining.

REFERENCES

[1] Agrawal, R, Srikant, R. Fast algorithms for mining association rules. In: Proc. 20th int. conf. very large data bases, VLDB 1994, Santiago de Chile, Chile, 12-15 September, 1994: 487-499.

[2] Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol. 22, no. 2, pp. 207–216. ACM, June 1993.

[3] Aliberti, G, Colantonio, A, Di Pietro, R, Mariani, R. EXPEDITE: EXPress closed ITemsetEnumeration. Expert Systems with Applications, 2015, 42(8):3933-3944.

[4] Barsky, M, Kim, S, Weninger, T, Han, J. Mining flipping correlations from large datasets with taxonomies. VLDB Endowment, 2011, 5(4):370-381.

[5] Bonchi F, Lucchese C. Pushing tougher constraints in frequent pattern mining. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Hanoi, Vietnam, 18-20 May, 2005:114-124.

[6] Duong, QH., Liao, B, Fournier-Viger, P, Dam, TL. An efficient algorithm for mining the top-k high utility itemsets, using novel threshold raising and pruning strategies. Knowledge-Based Systems, 2016.

[7] Fournier-Viger P, Wu CW, Tseng VS. Novel concise representations of high utility itemsets using generator patterns. In: Proc. Intern. Conf. International Conference on Advanced Data Mining and Applications, Guilin, China, 19-21 December, 2014:30-43.

[8] Fournier-Viger, P, Lin, JCW, Dinh, T, Le, HB. Mining Correlated High-Utility Itemsets using the Bond Measure. In: Proc. Intern. Conf. Hybrid Artificial Intelligence Systems. Seville, Spain, 18-20 April, 2016:53-65.

[9] Fournier-Viger, P, Zida, S. FOSHU: Faster On-Shelf High Utility Itemset Mining with or without negative unit profit. Proc. 30th Symposium on Applied Computing. Salamanca, Spain, 13-17 April, 2015:857-864.

[10] Geng L, Hamilton HJ. Interestingness measures for data mining: A survey. ACM Computing Surveys. 2006, 30;38(3):9.

[11] Han, J, Pei, J, Kamber, M. Data mining: concepts and techniques. Amsterdam:Elsevier; 2011.

[12] Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD Record, vol. 29, no. 2, pp. 1–12. ACM, May 2000

[13] Hegland M. The apriorialgorithma tutorial. Mathematics and computation in imaging science and information processing. 2005;11:209-62.

[14] Koh, YS, Ravana, SR. Unsupervised Rare Pattern Mining: A Survey. ACM Transactions on Knowledge Discovery from Data, 2016, 10(4): article no. 45

[15] Lin, JC. W, Gan, W, Fournier-Viger, P, Hong, TP, Tseng, VS. Fast Algorithms for Mining High-Utility Itemsets with Various Discount Strategies. Advanced Engineering Informatics, 2016.

[16] Liu, B., Hsu, W., Ma, Y. Mining Association Rules with Multiple Minimum Supports. In: Proc. ACM SIGKDD Intern. Conf. Knowledge Discovery and Data Mining, San Diego, USA, 15-18 August, 1999:337-341.

[17] Liu, Y., Liao, W.K. and Choudhary, AN. A two-phase algorithm for fast discovery of high utility itemsets. In: Pacific-Asia Conference on

Knowledge Discovery and Data Mining, Hanoi, Vietnam, 18-20 May, 2005:689-695.

[18] Lucchese, C., Orlando, S., Perego, R. Fast and Memory Efficient Mining of Frequent Closed Itemsets. IEEE Trans. Knowl. Data Eng., 2006, 18(1):21-36

[19] Omiecinski, E. Alternative Interest Measures for Mining Associations in Databases. IEEE Transactions on Knowledge Discovery and Data Engineering. 2003, 15(1):57-69.

[20] Pasquier N, Bastide Y, Taouil R, Lakhal L. Discovering frequent closed itemsets for association rules. In: Proc. Intern. Conf. Database Theory, Jerusalem, Israel, 10-12 January, 1999:398-416.

[21] Pei, J, Han, J, Lakshmanan, LV. 2004. Pushing convertible constraints in frequent itemset mining. Data Mining and Knowledge Discovery, 8(3):227-252.

[22] Pei, J, Han, J, Lakshmanan, LV. Mining frequent itemsets with convertible constraints. In: Proc. 17th Intern. Conf. Data Engineering, Heidelberg, Germany 2-6 April, 2001:433-442.

[23] Pei, J, Han, J, Lu, H, Nishio, S, Tang, S, Yang, D,H-mine: Hyper-structure mining of frequent patterns in large databases. In: Proc. 2001 IEEE Intern. Conf. Data Mining, San Jose, USA, 29 November - 2 December, 2001:441-448.

[24] Sengstock, C, Gertz, M. Spatial Itemset Mining: A Framework to Explore Itemsets in Geographic Space. In: Proc. East European Conference on Advances in Databases and Information Systems, Genoa, Italy, 1-4 September, 2013:148-161.

[25] Soulet A, Rioult F. Efficiently depth-first minimal pattern mining. In: Proc. 18th PacificAsia Conf. Knowledge Discovery and Data Mining. Tainan, Taiwan, 13-16 May, 2014:28-39.

[26] Soulet, A, Raissi, C, Plantevit, M, Cremilleux, B. Mining dominant patterns in the sky. In: Proc. 11th IEEE Int. Conf. on Data Mining, Vancouver, Canada, 11-14 December, 2011:655-664..

[27] Szathmary L, Valtchev P, Napoli A, Godin R, Boc A, Makarenkov V. A fast compound algorithm for mining generators, closed itemsets, and computing links between equivalence classes. Annals of Mathematics and Artificial Intelligence. 2014, 1;70(1-2):81-105.

[28] Szathmary, L, Napoli, A, Valtchev, P. Towards Rare Itemset Mining. In: Proc. 19th IEEE Intern. Conf. Tools with Artificial Intelligence, Patras, Greece, 29-31 October, 2007:305-312.

[29] Szathmary, L, Valtchev, P, Napoli, A, Godin, R. Efficient Vertical Mining of Minimal Rare Itemsets. In: Proc. 9th Intern. Conf. Concept Lattices and Their Applications, Fuengirola, Spain, 11-14 October, 2012: 269-280.

[30] Tang L, Zhang L, Luo P, Wang M. Incorporating occupancy into frequent pattern mining for high quality pattern recommendation. In: Proc. 21st ACM Intern. Conf. Information and knowledge management, In: Proc. 21st ACM Intern. Conf. Information and knowledge management. Maui, USA, 29 October - 2 November, 2012:75-84.

[31] Torres-Verdn C, Chiu KY, Vasudeva Murthy AS. WFIM: weighted frequent itemset mining with a weight range and a minimum weight. In: Proc. 2005 SIAM Intern. Conf. Data Mining, Newport Beach, USA, 21-23 April, 2005:636-640.

[32] Feng Tao, Weighted Association Rule Mining using Weighted Support and Significant framework. ACM SIGKDD, Aug 2003.

[33] Tseng, V, Wu, C, Fournier-Viger, P, Yu, PS. Efficient Algorithms for Mining Top-K High Utility Itemsets. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(1):54-67.

[34] Uno, T, Kiyomi, M, Arimura, H. LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets. Proc. ICDM'04 Workshop on Frequent Itemset Mining Implementations, CEUR, 2004.

[35] Vo B, Hong TP, Le B. DBV-Miner: A Dynamic Bit-Vector approach for fast mining frequent closed itemsets. Expert Systems with Applications. 2012, 39(8):7196{206.

[36] Xiong H, Tan PN, Kumar V. Mining strong affinity association patterns in data sets with skewed support distribution. In: Proc. 2003 IEEE Intern. Conf. Data Mining. Melbourne, USA, 19-22 December, 2003:387-394.

[37] Yun, U, Ryang, H, Ryu, KH. High utility itemset mining with techniques for reducing overestimated utilities and pruning candidates. Expert Syst. Appl. 2014, 41(8):3861-3878.

[38] Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: Third International Conference Knowledge Discovery and Data Mining (1997).

[39] Zaki, MJ, Gouda, K. Fast vertical mining using diffsets. In: Proc. 9th ACM SIGKDD Intern. Conf. Knowledge Discovery and Data Mining, Washington DC, USA, 24 – 27 August, 2003:326-335

[40] Zaki, MJ, Hsiao, CJ, CHARM: An efficient algorithm for closed itemset mining. In: Proc. 12th SIAM Intern. Conf. Data Mining, Anaheim, USA, 26-28 April, 2012:457-473.

[41] Zida, S., Fournier-Viger, P, Lin, JC.W, Wu, CW, Tseng, VS. EFIM: A Highly Efficient Algorithm for High-Utility Itemset Mining. In: Proc. 14th Mexican Intern. Conf. Artificial Intelligence, Cuernavaca, Mexico, 25-31 October, 2015:530-546.