IOP Conf. Series: Materials Science and Engineering **434** (2018) 012043 doi:10.1088/1757-899X/434/1/012043

The concept of frequent itemset mining for text

D S Maylawati*

Departement of Informatics, Sekolah Tinggi Teknologi Garut, Jalan Mayor Syamsu No 1 Tarogong Kidul Kabupaten Garut 44151, Indonesia

*dsaadillah@sttgarut.ac.id

Abstract. Frequent itemset mining is one of popular data mining technique with frequent pattern or itemset as representation of data. However, most of frequent itemset mining research was conducted for structured data. In this paper, we did literature review of the frequent itemset mining algorithm that suitable for unstructured data such as text data. We reviewed several frequent itemset mining algorithm that had already used in text mining research, among others Apriori algorithm; Pattern-growth algorithm; and various algorithm for itemset mining problem such as based on representation, database changes, and richer database type. The result showed that from year to year research on text data using frequent itemset mining had increased, including the development of frequent itemset mining algorithms. Although, still rarely new algorithms were implemented in text data

1. Introduction

Text are one of the unstructured data which need special treatment prior to further processes [1], [2] such as text mining, information retrieval, and natural language processing. In the digital and social media era, text running everyday can be utilized for important information or even knowledge. To find important aspects or unknown information automatically, text mining is the right technique since it extracts data to finally acquire knowledge [3]–[5]. Text mining, or sometimes known as text data mining, is a part of data mining [6], [7]. The difference between both is that in data mining, the data are structured while in text mining, the data analyzed are text which are unstructured or semi-structured [2], [8], [9]. Therefore, the text need to be represented in structured data to enable data mining process.

Structured representation of a text is generally divided into two types: single word (bag of words) and multiple words. Bag of words is a structured representation form which collect all the words in the document without seeing the relationships among the words [10]–[12], while multiple word representation collects words in the text document by selecting the relationships among the words so that the semantic meaning of the text is maintained [13]. Frequent pattern is a form of multiple word representation so that the structured representation of the text keep the meanings of the text [14]–[17]. Frequent pattern mining or frequent itemset mining (FIM) is one of the data mining techniques resulting in a pattern of frequent itemset [2], [17]–[19]. Since ealy 1993 to 2018, there have been at least 57 FIM algorithms [20]. Basically all the FIM algorithms implement mining towards structured data. However, it is possible to implement the algorithms in the unstructured data like text. In this study, we investigate literature on FIM algorithm and survey the trends of their use in text.

IOP Conf. Series: Materials Science and Engineering **434** (2018) 012043 doi:10.1088/1757-899X/434/1/012043

2. Frequent itemset mining

Data mining is a technique to find knowledge from data history which aims to predict the future. FIM, which was previously known as large itemset mining [18], [21], works to find frequent itemset from the database transaction [20], [22], [23]. Items which are frequent are those meeting the threshold value or minimum support. Minimum support indicates the number of itemset to meet from the whole transaction of the database. Different from sequential pattern mining, FIM creates patterns with items emerging simultaneously without paying attention to the order.

Table 1. The example of transaction database.

Id Transaction	Transaction
1	milk, diapers, tissue
2	soap, diapers, snack, coffee
3	tissue, milk, coffee
4	diapers, milk, soap

Table 1 is an example of database transaction of frequent itemset with minimum support of 50%. Each item arises at least two times, they are {(milk)}; {(diapers)}; {(tissue)}; {(soap)}; {(coffee)}; {(milk, diapers)}; {(milk, tissue)}; and {(soap, diapers)}. The frequent itemset of {(milk, diapers)} is considered equal with that of {(diapers, milk)}, while "snack" does not belong to frequent itemset since it does not meet the value of minimum support.

2.1. Apriori algorithm and its variants

Apriori algorithm is a basic as well as first algorithm for FIM. The algorithm takes transactions in the database which fulfill the minimum support or the threshold value using breadth-first search to search all the frequent itemset [18], [20]. Since the algorithm raises up the feature candidates prior to finding the frequent itemset, it usually scans repeatedly. To cope with it (and with big data), AprioriTID and AprioriHybrid algorithms, which are a combination of Apriori and AprioriTID, are developed [21]. Following that, there are several newer algorithms, one of which is Eclat algorithm which develops the transaction searching on database promoting depth-first search [24]. Éclat is further developed into dEclat which results in more efficient frequent itemset [25]. There is also SS-FIM algorithm, a development of Apriori, which only scans the database once.

2.2. Pattern-growth algorithm and its variants

Pattern-growth algorithm is designed to cope with the limitations of Apriori and Eclat algorithms that tend to scan database. Algorithms belonging to pattern-growth are FP-Growth [26], [27], H-Mine [28], and LCM [29]. Those three algorithms are FIM algorithms that do not raise up the feature candidates. There is also PrePost algorithm, an algorithm adopting FP-Growth, which has different structure [30]. It is later developed into FIN algorithm [31], and Pre-Post+ algorithm [32]. The other algorithm of FIM is Relim which eliminates the recursive. The algorithm has simpler structure inspired by FP-Growth yet similar to H-Mine [33].

2.3. Frequent itemset mining algorithm based on representation problem

There are three approaches for frequent itemset representation by selecting the features so that the frequent itemset is more efficient. The approaches are maximal itemset, close itemset, and generator itemset (key itemset). And i itemset is called maximum if there is no longer i itemset which is a sub-itemset of the itemset [14], [15], [18], [21]. For instance, an i itemset has several items (a, b, c, d, e) and and i' itemset has (b, d, e), and both are in a collection of documents. Thus, itemset i' is a sub-itemset of itemset i; meaning that itemset i is maximum and itemset i' will be removed. Close approach, in the meantime, selects features to be more efficient. An i itemset is considered close if there is no more itemset i' which is the sub-itemset of itemset i, where itemset i and i' have the same frequency [29], [34]. For example, itemset i has (a, be, c, d, e) and the frequency is 3, while itemset i' has (b, d, e) and the frequency is also 3. Therefore, itemset i is considered close and itemset i' will be removed. However, if itemset i' different frequency and itemset i is the super itemset, so itemset i' will not be removed since it is a close itemset. The last itemset, generator itemset, is the opposite of close itemset. Thus, if there is no more itemset i which is the super itemset of the sub itemset i', where itemset i and itemset i' have the same frequency.

dEclat algorithm is actually one of the algorithm using maximal approach. Other maximal approach frequent algorithms are FPMax [35], Charm-MFI [36], Mafia [37], and GenMax [38]. LCM algorithm is an algorithm using close itemset approach and later developed into LCM ver 2 [39] and LCM ver 3 [40]. Other FIM algorithms using close itemset approach are FPClose [41], Charm [42], dCharm [43], Closet [44], Closet+ [34], DCI_Close [45][46], and AprioriClose [45]. Algorithms using generator itemset approach are PASCAL [47], DefMe [48], ZART [49], and VGEN [50].

2.4. Frequent itemset mining algorithm based on database changes and richer database type

FIM algorithm is also developing since problems arise in database; one of which is the huge size of the database, the changing database, the uncertain database, and the streaming database. Based on those problems, new FIM algorithms emerge. CP-Tree (Compact Pattern Tree) algorithm, which is a development of FP-Growth algorithm, is designed for changing database due to additional transaction [51], [52], MEIT [53]. There is also U-Apriori algorithm [54], a FIM algorithm for uncertain data. For streaming database, there are CPS-Tree [55], estDec [56], estDec+ [57], CloStream [58], and CFI-Stream [59] algorithms. Algorithms categorized into new ones for quantitative transaction database using fuzzy frequent itemset approach are FFI-Miner [60] and MFFI-Miner [61]. Sometimes there are inefficient itemsets due to irrelevant data. Thus, VME [62] and MEI [63] FIM algorithms are present to remove itemsets from the irrelevant data.

3. Frequent itemset mining for text

FIM on a text is also known as frequent word itemset (FWI) [2], [17], as one of the structured text representations. FWI perceives documents or a series of text as an itemset pattern. The FWI structure is illustrated with {(w1,w2), (w3,w4), …} where (w1,w2) is FWIi, (w3,w4) and FWIi+1, etc. The order of FWI is according to the order of the data in the document or the text, yet elements in the FWI do not have to follow the order. This means that in the collection of FWIi document, it is usually followed by FWIi+1 and so on. Elements or items in FWIi, w1 usually emerge with w2 and do not have to be in order with w1 followed by w2; however, if w2 comes earlier, then w1 will be categorized as the same FWI, so as the emergence of FWSi+1 and so on.

Table 2. The example of document collection (presented in Indonesian Slang language).

No.	Content of document
1	*Gue kalo nonton drama korea tuh berasa ngehipnotis gue. Secara ceritanya seru, episodenya dikit ga sampe ratusan episode. Udah gitu pemainnya enak diliat, hehe.*
2	*Gue lagi terhipnotis sama yang namanya drama korea. Ga bisa berhenti nonton sampe abis episodenya. Secara cuma dikit gitu loh episodenya, paling 2-3 hari kelar nontonya.*
3	*Temen gue bilang sekali nonton drama korea bakal ngehipnotis pengen nonton terus. Terus gue coba, eeh ternyata seru juga ceritanya, episodenya cuma dikit paling banyak 20-an, jadi ga lebay en ngebosenin.*

From the document collection in table 2, FWI representation with minimum support 50%, such as (*gue, nonton*) as FWI_1 in documents 1 and 3; (*gue, nonton, drama, korea*) as FWI_2 in documents 1 and 3; (*gue, drama, korea, hipnotis*) as FWI_3 in documents 1, 2, and 3 is equal to (*drama, korea, hipnotis, gue*) in document 1; (*drama, korea, hipnotis*) as FWI_4 in documents 1, 2, and 3; (*seru, cerita*) as FWI_5 in document 1 is equal to (*cerita, seru*) in documents 1 and 2; (*secara, episode, dikit*) as FWI_6 in document 1 is equal to (*secara, dikit, episode*) in documents 1 and 2; and (*episode, dikit*) as FWI_7 in documents 1 and 3 is equal to (*dikit, episode*) in documents 1 and 2. Of seven FWI shaped from the example textx in the table 2, the set of FWI are {(*gue, nonton*)} as set of FWI_1; {(*gue, nonton*), (*drama, korea, hipnotis*), (*episode, dikit*)} as set of FWI_2; {(*drama,korea, hipnotis*), (*cerita, seru*), (*episode, dikit*)} as set of FWI_3; {(*gue, drama, korea, hipnotis*), (*cerita, seru*)} as set of FWI_4; and {(*gue,drama,korea,hipnotis*), (*secara,episode, dikit*)} as set of FWI_5.

4. Results and discussion

From all the FIM algorithms that keep developing, we do a survey on each algorithm to see the trends of the FIM algorithms for text data. We collected the data from Mendeley and Google Scholar since the indexing of both is complete and quite representative for publications of several resources. Table 3 shows that from 38 FIM algorithms, more than five research studies with text data use Apriori and FP-Growth algorithms, and seven FIM algorithms implemented in the research studies with text data such as AprioriTD, LCMFreq, LCM, AprioriClose, AprioriTID Close, U-Apriori, and CP-Tree. Whereas, 29 other FIM algorithms have not been found in research studies using text data. This indicates that FIM algorithms have been used to search frequent itemset from unstructed data such as texs, either in text mining, information retrieval, and natural language processing data mining. FIM basic algorithms like Apriori and FP-Growth and the most used ones. However, there are several FIM algorithms which have not been implemented in studies with text data.

Table 3. FIM algoithm for research with text data.

Algorithm	**How many used for research with text data**		
	0	**> 0 & < 5**	**≥ 5**
Apriori			√
AprioriTID		√	
FP-Growth			√
Eclat	√		
dEclat	√		
Relim	√		
H-Mine	√		
LCMFreq		√	
PrePost	√		
PrePost+	√		
FIN	√		
SSFIM	√		
FPClose	√		
Charm	√		
DCI_Closed	√		
LCM		√	
AprioriClose		√	
AprioriTID Close		√	
FPMax	√		
Charm-MFI	√		

Table 3. Cont.

DefMe	√	
PASCAL	√	
ZART	√	
Itemset-Tree	√	
MEIT	√	
estDec	√	
estDec+	√	
CloStream	√	
U-Apriori		√
VME	√	
FFI-Miner	√	
MFFI-Miner	√	
CP-Tree		√
VGEN	√	
GenMax	√	
Mafia	√	
CPS-Tree	√	
MEI	√	

5. Conlusion

FIM is a data mining technique which searches frequent itemset from transaction database. Basically FIM is used to do mining for structured data. However, FIM can also be used for unstructured data such as text which create FWI as structured representation from text. From several FIM algorithms which keep developing, only 2 out of 28 (5.26%) which are used in research studies with text data and 7 out of 38 (18.42%) which are used in research studies with text data. Whereas, 29 out of 38 (76.32%) have not been implemented in text. This becomes a possibility for future studies to implement and research FIM algorithms for text, either in text mining, information retrieval, or natural language processing.

Acknowledgement

We would like to thank Sekolah Tinggi Teknologi Garut for the full support for this publication.

References

[1] H Mahgoub, D Rösner, N Ismail and F Torkey 2008 A Text Mining Technique Using Association Rules Extraction *Int. J. Comput. Intell.* **4**(1) pp. 21–28

[2] D S A Maylawati 2015 *Pembangunan library pre-processing untuk text mining dengan representasi himpunan frequent word itemset (hfwi) studi kasus: bahasa gaul Indonesia* (Bandung)

[3] V Gupta and G S Lehal 2009 A survey of text mining techniques and applications *Journal of Emerging Technologies in Web Intelligence* **1**(1) pp. 60–76

[4] V Gupta and G SLehal 2010 A Survey of Text Summarization Extractive techniques in *Journal of Emerging Technologies in Web Intelligence* **2**(3) pp. 258–268

[5] C J Torre, M J Martin Bautista, D Sanchez and I Blanco 2008 Text Knowledge Mining: And Approach To Text Mining *ESTYLF08*

[6] A H Tan 1999 Text Mining: The state of the art and the challenges in *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases 1999* **8** pp. 65–70

[7] H Jiawei, M Kamber, J Han, M Kamber and J Pei 2006 Data Mining: Concepts and Techniques Elsiver)

[8] H Jiawei, M Kamber, J Han, M Kamber and J Pei 2012 Data Mining: Concepts and Techniques

[9] S M Weiss, N Indurkhya, T Zhang and F J Damerau 2010 *Information Retrieval and Text Mining* (Springer Berlin Heidelb) Fundamentals of Predictive Text Mining pp. 75–90

[10] H M Wallach 2006 Topic Modeling: Beyond Bag-of-Words *ICML* **1** pp. 977–984
[11] A Sethy and B Ramabhadran 2008 Bag-of-word normalized n-gram models in *Proceedings of the Annual Conference of the International Speech Communication Association INTERSPEECH 2008* pp. 1594–1597
[12] W Pu, N Liu, S Yan, J Yan, K Xie and Z Chen 2007 Local word bag model for text categorization in *Proceedings - IEEE International Conference on Data Mining ICDM 2007*pp. 625–630
[13] A Doucet and H Ahonen-Myka 2010 An efficient any language approach for the integration of phrases in document retrieval *Lang. Resour. Eval.* **44**(1–2) pp. 159–180
[14] A Doucet and H Ahonen Myka 2004 Non-contiguous word sequences for information retrieval *MWE '04 Proc. Work. Multiword Expressions* **26** pp. 88–95
[15] H Ahonen Myka 2002 Discovery of Frequent Word Sequences in Text *Proc. ESF Explor. Work. Pattern Detect. Discov.* **24** (Teollisuuskatu) pp. 180–189
[16] H Ahonen Myka 1999 Finding All Maximal Frequent Sequences in Text *Proc. ICML Work. Mach. Learn. Text Data Anal.* pp. 11–17
[17] D Sa'Adillah Maylawati and G A Putri Saptawati Set of Frequent Word Item sets as Feature Representation for Text with Indonesian Slang in *Journal of Physics: Conference Series* **801(**1)
[18] R Agrawal and R Srikant 1994 Fast Algorithms for Mining Association Rules in Large Databases *J. Comput. Sci. Technol.***15**(6) pp. 487–499
[19] J Han, H Cheng, D Xin and X Yan 2007 Frequent pattern mining: Current status and future directions *Data Min. Knowl. Discov.***15**(1) pp. 55–86
[20] P Fournier Viger, J C W Lin, B Vo, T T Chi, J Zhang and H B Le 2017 A survey of itemset mining *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **7**(4)
[21] R Agrawal, H Mannila, R Srikant, H Toivonen and a I Verkamo 1996 Fast discovery of association rules *Advances in knowledge discovery and data mining* **12** pp. 307–328
[22] F Kovács and J Illés 2013 Frequent itemset mining on hadoop *Comput. Cybern. (ICCC) 2013 IEEE 9th Int. Conf.* pp. 241–245
[23] S Moens, E Aksehirli and B Goethals 2012 Frequent Itemset Mining for Big Data in *2013 IEEE International Conference on Big Data* pp. 111–118
[24] M J Zaki 2000 Scalable algorithms for association mining *IEEE Trans. Knowl. Data Eng.* **12(**3) pp. 372–390
[25] M J Zaki and K Gouda 2003 Fast vertical mining using diffsets in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining KDD'03* p. 326
[26] J Han, J Pei and Y Yin 2000 Mining frequent patterns without candidate generation in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data SIGMOD 2000* pp. 1–12
[27] J Han, J Pei, Y Yin and R Mao 2004 Mining frequent patterns without candidate generation: A frequent-pattern tree approach *Data Min. Knowl. Discov.***8**(1) pp. 53–87
[28] J Pei, J Han, H Lu, S Nishio, S Tang and D Yang 2001 H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases *IEEE Int. Conf. Data Min.* pp. 441–448
[29] T Uno, T Asai, Y Uchida and H Arimura 2003 LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets. *Fimi* **90**
[30] Z Deng, Z Wang and J Jiang 2012 A new algorithm for fast mining frequent itemsets using N-lists *Sci. China Inf. Sci.* **55**(9) pp. 2008–2030
[31] Z H Deng and S L Lv 2014 Fast mining frequent itemsets using Nodesets *Expert Syst. Appl.* **41**(10) pp. 4505–4512
[32] Z H Deng and S L Lv 2015 PrePost+: An efficient N lists based algorithm for mining frequent itemsets via Children-Parent Equivalence pruning *Expert Syst. Appl.* **42**(13) pp. 5424–5432
[33] C Borgelt 2005 Keeping things simple in *Proceedings of the 1st international workshop on open source data mining frequent pattern mining implementations OSDM 2005* pp. 66–70
[34] J Wang, J Han and J Pei 2003 Closet+: Searching for the best strategies for mining frequent closed

itemsets *Proc. ninth ACM SIGKDD Int. Conf. Knowl. Discov. data Min.* pp. 236–245
[35] G Grahne and J Zhu Efficiently Using Prefix-trees in Mining Frequent Itemsets *Proc. 1st IEEE ICDM Work. Freq. Itemset Min. Implementations* pp. 236-245
[36] M J Zaki and C J Hsiao 2005 Efficient algorithms for mining closed itemsets and their lattice structure *IEEE Trans. Knowl. Data Eng.* **17**(4) pp. 462–478
[37] D Burdick, M Calimlim, J Flannick, J Gehrke and T Yiu 2005 MAFIA: A maximal frequent itemset algorithm *IEEE Trans. Knowl. Data Eng.***17**(11) pp. 1490–1504
[38] K Gouda and M J Zaki 2005 GenMax: An efficient algorithm for mining maximal frequent itemsets *Data Min. Knowl. Discov.* **11**(3) pp. 223–242
[39] T Uno, M Kiyomi and H Arimura 2004 LCM ver 2 : Efficient Mining Algorithms for Frequent / Closed/Maximal Itemsets Algorithms for Efficient Enu- meration in *International Workshop on Open Source Data Minig* pp. 1–11
[40] T Uno, M Kiyomi and H Arimura 2005 LCM ver.3: Collaboration of Array, Bitmap and Prefix Tree for Frequent Itemset Mining *Proc. 1st Int. Work. open source data Min. Freq. pattern Min. implementations OSDM'05*, pp. 77–86
[41] G Grahne and J Zhu 2005 Fast algorithms for frequent itemset mining using FP-trees *IEEE Trans. Knowl. Data Eng.* **17**(10) pp. 1347–1362
[42] M J Zaki and C J Hsiao 2001 CHARM : An Efficient Algorithm for Closed Itemset Mining *Data Min. Knowl. Discov.* **15** pp. 457–473
[43] M J Zaki and Ching Jui Hsiao 2002 An Efficient Algorithm for Closed Itemset Mining in *SIAM International Conference on Data Mining SDM'02 2002* pp. 33–43
[44] J Pei, J Han and R Mao 2000 CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets *ACM SIGMOD Work. Res. issues data Min. Knowl. Discov.* **4**(2) pp. 21–30
[45] N Pasquier 2009 Frequent Closed Itemsets Based Condensed Representations for Association Rules *Post-Mining Assoc. Rules Tech. Eff. Knowl. Extr.* pp. 248–273
[46] C Lucchese, S Orlando and R Perego 2006 Fast and memory efficient mining of frequent closed itemsets *IEEE Trans. Knowl. Data Eng.* **18**(1) pp. 21–36
[47] Y Bastide, R Taouil, N Pasquier, G Stumme and L Lakhal 2000 Mining frequent patterns with counting inference *ACM SIGKDD Explor. Newsl.* **2**(2) pp. 66–75
[48] Soulet, A., & Rioult, F. (2014, May). Efficiently depth-first minimal pattern mining. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (Cham: Springer) pp. 28-39
[49] L Szathmary, A Napoli and S O Kuznetsov 2007 ZART: A multifunctional itemset mining algorithm in *CEUR Workshop Proceedings* **331** pp. 22–33
[50] Fournier Viger P, Gomariz A, Šebek M and Hlosta M 2014 VGEN: fast vertical mining of sequential generator patterns In *International Conference on Data Warehousing and Knowledge Discovery* (Cham: Springer) pp. 476-488
[51] Ahmed C F, Tanbeer S K, Jeong B S and Lee Y K 2008 Mining weighted frequent patterns in incremental databases In *Pacific Rim International Conference on Artificial Intelligence* (Berlin Heidelberg: Springer) pp. 933-938
[52] D S A Maylawati, M A Ramdhani, A Rahman and W Darmalaksana 2017 Incremental technique with set of frequent word item sets for mining large Indonesian text data in *2017 5th International Conference on Cyber and IT Service Management CITSM 2017*
[53] Fournier Viger P, Mwamikazi E, Gueniche T and Faghihi U 2013 MEIT: Memory Efficient Itemset Tree for targeted association rule mining In *International Conference on Advanced Data Mining and Applications* (Berlin Heidelberg: Springer) pp. 95-106
[54] C K Chui, B Kao and E Hung 2007 Mining Frequent Itemsets from Uncertain Data *Proc. 11th Pacific-Asia Conf. Adv. Knowl. Discov. data Min.* pp. 47–58
[55] S K Tanbeer, C F Ahmed, B S Jeong and Y K Lee 2008 Efficient frequent pattern mining over data streams in *Proceeding of the 17th ACM conference on Information and knowledge mining CIKM'08* p. 1447.
[56] J H Chang and W S Lee 2003 Finding recent frequent itemsets adaptively over online data streams

in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining-KDD'03* p. 487

[57] S J Shin, D S Lee and W S Lee 2014 CP-tree: An adaptive synopsis structure for compressing frequent itemsets over online data streams *Inf. Sci. (Ny).* **278** pp. 559–576

[58] Yen S J, Lee Y S, Wu C W and Lin C L 2009 An efficient algorithm for maintaining frequent closed itemsets over data stream In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (Berlin Heidelberg: Springer) pp. 767-776

[59] N Jiang and L Gruenwald 2006 CFI-stream: Mining closed frequent itemsets in data streams *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* pp. 592–597

[60] J C W Lin, T Li, P Fournier Viger and T P Hong 2015 A fast Algorithm for mining fuzzy frequent itemsets *J. Intell. Fuzzy Syst.* **29**(6) pp. 2373–2379

[61] J C W Lin, T Li, P Fournier Viger, T P Hong, J M T Wu and J Zhan 2017 Efficient Mining of Multiple Fuzzy Frequent Itemsets *Int. J. Fuzzy Syst.***19**(4) pp. 1032–1040

[62] Deng Z and Xu X 2010 An efficient algorithm for mining erasable itemsets In *International Conference on Advanced Data Mining and Applications* (Berlin Heidelberg: Springer) pp. 214-225

[63] T Le and B Vo 2014 MEI: An efficient algorithm for mining erasable itemsets *Eng. Appl. Artif. Intell.* **27** pp. 155–166