International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 22
November 2018

Efficient Algorithms for Mining Top-K High Utility Item sets

Arveti Anusha[1], M.Subbareddy[2]
[1]P.G. Scholar, [2]Assistant Professor
[1,2] Branch: Computer Science & Engineering
[1,2] S V R Engineering College, Nandyal, Kurnool (Dist.).
Email : [1]anusha.arveti@gmail.com, [2]subbareddy.meruva@gmail.com

Abstract

High utility itemsets (HUIs) mining is a developing point in data mining, which alludes to finding all itemsets having an utility gathering a user-indicated least utility limit min_util. Nonetheless, setting min_util fittingly is a troublesome issue for users. As a rule, finding a suitable least utility limit by experimentation is a repetitive procedure for users. On the off chance that min_util is set too low, an excessive number of HUIs will be created, which may cause the mining procedure to be extremely inefficient. Then again, if min_util is set too high, it is likely that no HUIs will be found. In this paper, we address the above issues by proposing another system for best k high utility itemset mining, where k is the coveted number of HUIs to be mined. Two types of efficient algorithms named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One stage) are proposed for mining such itemsets without the need to set min_util. We give a basic examination of the two algorithms with exchanges on their points of interest and constraints. Exact assessments on both genuine and engineered datasets demonstrate that the execution of the proposed algorithms is near that of the ideal instance of best in class utility mining algorithms.

Index Terms—Utility mining, high utility itemset mining, top-k design mining, top-k high utility itemset mining

INTRODUCTION

Frequent thing set mining (FIM) is a crucial research subject in data mining. In any case, the conventional FIM may find a lot of frequent however low-esteem thing sets and lose the information on significant thing sets having low offering frequencies. Thus, it can't fulfill the prerequisite of users who want to find thing sets with high utilities, for example, high benefits.

To address these issues, utility mining develops as an essential subject in data mining and has gotten extensive consideration lately. In utility mining, every thing is related with an utility (e.g. unit benefit) and an event check in every exchange (e.g. amount). The utility of a thing set speaks to its significance, which can be estimated in terms of weight, esteem, amount or other information relying upon the user detail.

A thing set is called high utility thing set (HUI) if its utility is no not exactly a user-indicated least utility limit min_ util. HUI mining is fundamental to numerous applications, for example, gushing analysis, advertise analysis, versatile figuring and biomedicine However, efficiently mining HUIs in databases isn't a simple task on the grounds that the descending conclusion property utilized in FIM does not hold for the utility of thing sets. As it were, pruning look space for HUI mining is troublesome in light of the fact that a superset of a low utility thing set can be high utility. To handle this issue, the idea of exchange weighted use (TWU) show was acquainted with encourage the execution of the mining task. In this model, a thing set is called high exchange weighted usage thing set (HTWUI) if its TWU is no not exactly min_ util, where the TWU of a thing set speaks to an upper bound on its utility. In this manner, a HUI must be a HTWUI and all the HUIs must be incorporated into the entire arrangement of HTWUIs. An established TWU show based calculation comprises of two stages.

In the principal stage, called stage I, the entire arrangement of HTWUIs are found. In the second stage, called stage II, all HUIs are acquired by computing the correct utilities of HTWUIs with one database filter. Albeit numerous investigations have been committed to HUI mining, it is troublesome for users to pick a fitting least utility limit by and by. Contingent upon the edge, the yield size can be little or huge. In addition, the decision of the edge incredibly impacts the execution of the algorithms. In the event that the edge is set too low, such a large number of HUIs will be introduced to the users and it is troublesome for the users to understand the outcomes.

An expansive number of HUIs additionally makes the mining algorithms wind up inefficient or even come up short on memory, in light of the fact that the more HUIs the algorithms create, the more assets they devour. Despite what might be expected, if the edge is set too high, no HUI will be found. This procedure is both badly designed and tedious. To exactly control the yield measure and find the thing sets with the highest utilities without setting the edges, a promising arrangement is to reclassify the task of mining HUIs as mining top-k high utility thing sets (top-k HUIs). The thought is to give the users a chance to indicate k, i.e., the number of wanted thing sets, rather than determining the base utility limit. Setting k is more natural than setting the limit since k speaks to the number of thing sets that the users need to discover though picking the edge depends essentially on database qualities, which are frequently obscure to users. Utilizing a parameter k rather than the min_util limit is extremely alluring for some applications. For instance, to break down client buy conduct, top-k HUI mining fills in as a promising answer for users who want to know "What are the best k sets of items (i.e., thing sets) that contribute the highest benefits to the organization?" and "How to efficiently discover these thing sets without setting the min_util edge?". Albeit top-k HUI mining is fundamental to numerous applications, creating efficient algorithms for mining such examples isn't a simple task.

High Utility Itemset Mining:

High utility itemset mining has become piles of thought and various efficient algorithms have been proposed, for instance, Two Phase, IHUP, IIDS , UPGrowth [25], d2HUP [15] and HUI-Miner [14]. These algorithms can be all things considered arranged into two types: twophase and one-organize algorithms. The essential typical for two-arrange algorithms is that they include two phases. In the central stage, they deliver a game plan of hopefuls that are potential high utility itemsets. In the second stage, they figure the right utility of each candidate found in the principle stage to recognize high utility itemsets. Two-Phase, IHUP,IIDS and UP-Growth are two-arrange based algorithms. Advancement is one of the bleeding edge two-organize algorithms, which solidifies four convincing methods DGU, DGN,DLU and DLN for pruning hopefuls in the essential stage. One the inverse, the rule typical for one-organize algorithms is that they discover high utility itemsets using only a solitary stage and make no candidates. d2HUP and HUI-Miner are one-organize algorithms. d2HUP changes an even database into a tree-based structure called CAUL [15] and gets a model advancement system to explicitly discover high utility itemsets in databases. HUI-Miner considers a database of vertical game plan and changes it into utility-records [14]. The utility-list structure used in HUI-Miner allows clearly enrolling the utility of made itemsets on a fundamental level memory without checking the principal database. Despite the way that the above examinations may perform well in a couple of uses, they are not delivered for best k high utility itemset mining and still experience the evil impacts of the unnoticeable issue of setting reasonable edges.

Top-k Pattern Mining

Various examinations have been proposed to mine different types of best k designs, for instance, top-k frequent itemsets ,top-k frequent losed itemsets , top-k close back to back precedents, top-k association rules, top-k progressive rules, top-k association models and best k cosine likeness intriguing sets. What perceives each best k design mining figuring is the type of models found, and furthermore the data structures and chase systems that are used. For example, a couple of algorithms use a standard augmentation strategy for finding designs, while others rely upon a model improvement look using structures, for instance, FP-Tree. The choice of data structures and interest procedure impact the profitability of a best k design mining computation in terms of both memory and execution time. In any case, the above algorithms discover top-k designs according to ordinary measures instead of the utility measure. As a result, they may miss designs yielding high utility.

Top-k High Utility Pattern Mining

The task of best k high utility model mining was introduced by Chan et al. [4]. In any case, the significance of high utility itemset used in their examination is one of a kind in connection to the one used in this work. Chan et al's. contemplate has considered utilities of various tems, yet quantitative estimations of things in trades were not pondered. In [30], I have portrayed the task of best k high utility itemset mining by contemplating the two sums and advantages of things. This work has inspired several concentrates for mining top-k high utility models. Zihayat and A [37] have proposed an efficient count T-HUDS for mining top-k HUIs over data streams. Yin et al have proposed another system for mining top-k high utility back to back models. Starting late, Ryang and Yun extended [30] to propose the REPT computation [21] with four systems PUD, RIU, RSD and SEP for best k HUI mining. In REPT, other than the parameter k, users need to set another parameter N to control the sufficiency of RSD [21]. In any case, it is troublesome for users to pick a reasonable N regard and the choice of N fundamentally impacts the execution of REPT

EXISTING SYSTEM:

The standard FIM (Frequent thing set mining) may discover a considerable measure of frequent yet low-regard itemsets and lose the information on critical itemsets having low offering frequencies. In this way, it can't satisfy the need of users who need to discover itemsets with high utilities, for instance, high advantages. To address these issues, utility mining ascends as a basic subject in data mining and has become extensive thought starting late. In utility mining, everything is connected with an utility (e.g. unit advantage) and an occasion count in each trade (e.g. sum). The utility of a thing set addresses its centrality, which can be evaluated in terms of weight, regard, sum or other

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 22
November 2018

information depending upon the user detail. A thing set is called high utility itemset (HUI) if its utility is no not actually a user-shown slightest utility farthest point min_util. Starting late, high utility itemset mining has gotten groups of thought and various efficient algorithms have been proposed, for instance, Two-Phase, IHUP, IIDS, UP Growth, d2HUP and HUI-Miner. These algorithms can be all around arranged into two types: two-phase and one-organize algorithms

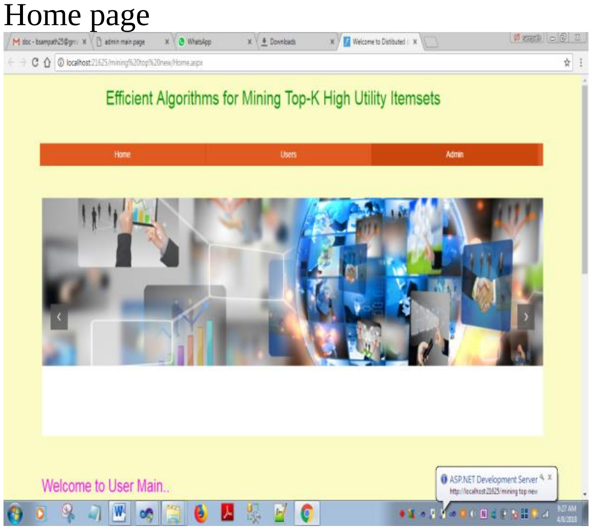
PROPOSED SYSTEM

In this endeavor, I address most of the above challenges by proposing a novel structure for best k high utility itemset mining, where k is the pined for number of HUIs to be mined.

Critical duties of this work are dense as seeks after:

• First, two efficient algorithms named TKU (mining Top-K Utility itemsets) and TKO (mining Top-K utility itemsets in One

Home page



stage) are proposed for mining the entire arrangement of best k HUIs in databases without the need to determine the min_util limit.

The TKU calculation receives a minimized tree-based structure named UP-Tree to keep up the information of exchanges and utilities of itemsets. TKU acquires valuable properties from the TWU model and comprises of two stages.

• In stage I, potential best k high utility itemsets (PKHUIs) are created. In stage II, top-k HUIs are recognized from the arrangement of PKHUIs found in stage I. Then again, the TKO calculation utilizes a rundown based structure named utility-rundown to store the utility information of itemsets in the database.

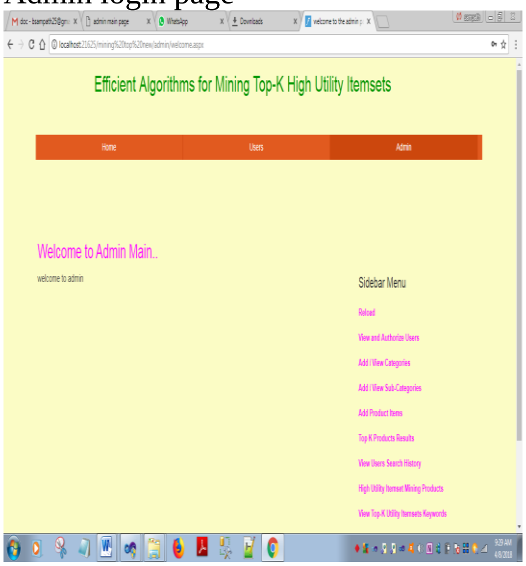
It uses vertical data representation techniques to discover top-k HUIs in only one phase

Admin login

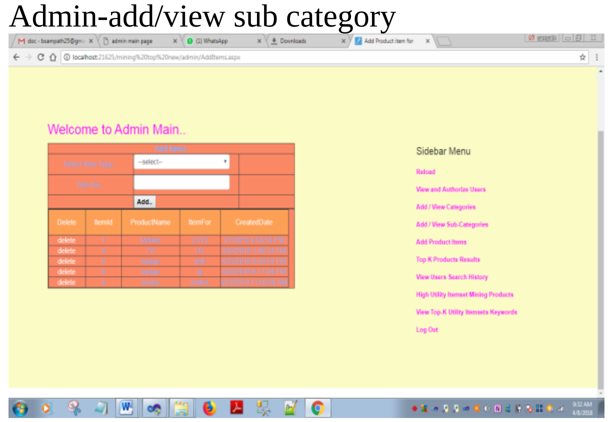
International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 22
November 2018

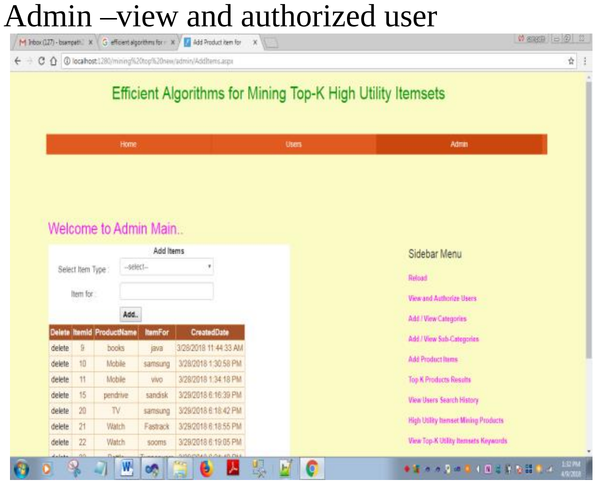
Admin login page



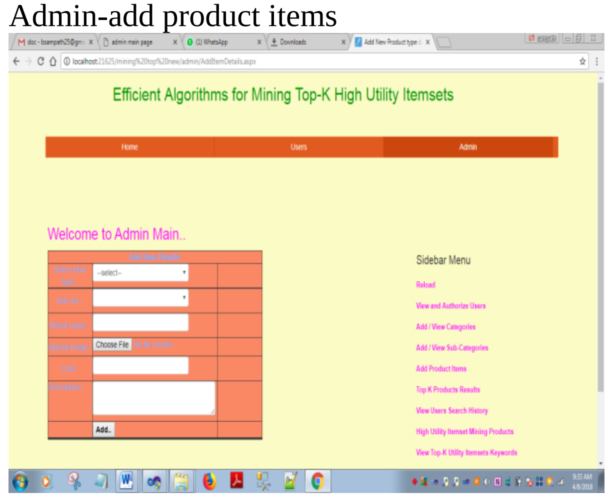
Admin-add/view sub category



Admin –view and authorized user

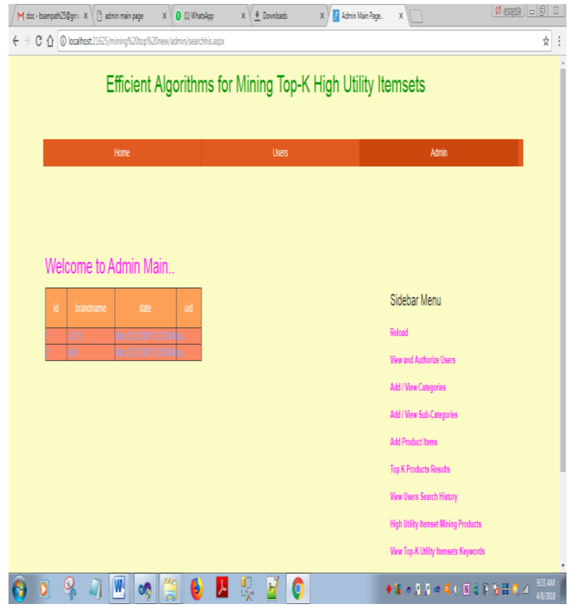


Admin-add product items



Admin-Add/ view category

Admin-top k product results



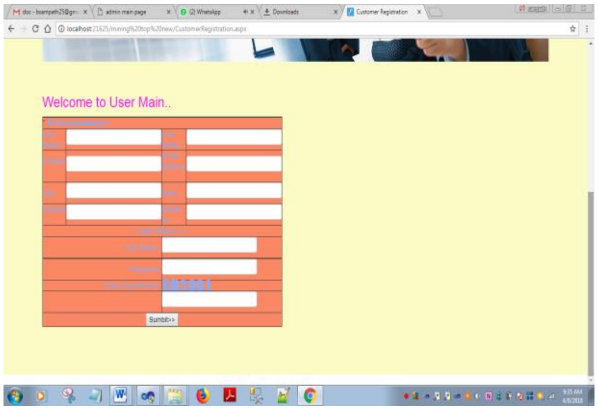
Top k Product Results

Admin-view user history search

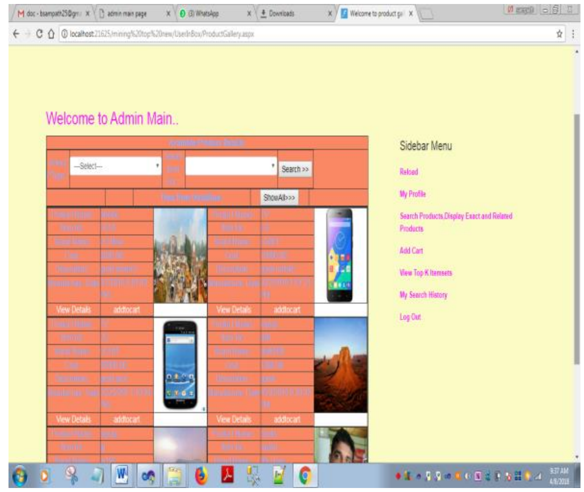
User login page

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 22
November 2018

New user

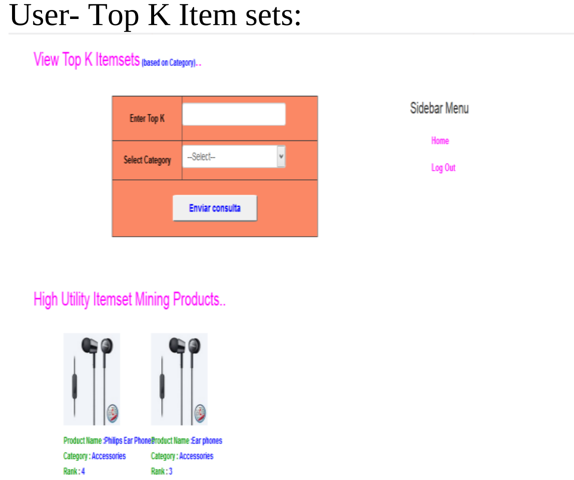


Users profile



User- search products, display exactly and related products

User- Top K Item sets:



Data Dictionary

Admin

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	name	text	latin1_swedish_ci		Yes	NULL		
	pass	text	latin1_swedish_ci		Yes	NULL		

Category

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 22
November 2018

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(11)			No	None	auto_increment	
category_name	varchar(45)	latin1_swedish_ci		Yes	NULL		

Product found

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(10)			No	None	auto_increment	
user	text	latin1_swedish_ci		Yes	NULL		
keyword	text	latin1_swedish_ci		Yes	NULL		
title	text	latin1_swedish_ci		Yes	NULL		
category	text	latin1_swedish_ci		Yes	NULL		
sub_category	text	latin1_swedish_ci		Yes	NULL		
date	text	latin1_swedish_ci		Yes	NULL		
type	text	latin1_swedish_ci		Yes	NULL		

Product search

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(10)			No	None	auto_increment	
user	text	latin1_swedish_ci		Yes	NULL		
keyword	text	latin1_swedish_ci		Yes	NULL		
date	text	latin1_swedish_ci		Yes	NULL		

Products

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(11)			No	None	auto_increment	
title	text	latin1_swedish_ci		Yes	NULL		
category	text	latin1_swedish_ci		Yes	NULL		
sub_category	text	latin1_swedish_ci		Yes	NULL		
price	text	latin1_swedish_ci		Yes	NULL		
description	text	latin1_swedish_ci		Yes	NULL		
image	longblob		BINARY	Yes	NULL		
rank	text	latin1_swedish_ci		Yes	NULL		

Request

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(11)			No	None	auto_increment	
user	text	latin1_swedish_ci		Yes	NULL		
search	text	latin1_swedish_ci		Yes	NULL		
download	text	latin1_swedish_ci		Yes	NULL		
ownerper	text	latin1_swedish_ci		Yes	NULL		

Reviews

International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 22
November 2018

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(10)			No	*None*	auto_increment	
title	text	latin1_swedish_ci		Yes	*NULL*		
category	text	latin1_swedish_ci		Yes	*NULL*		
sub_category	text	latin1_swedish_ci		Yes	*NULL*		
user	text	latin1_swedish_ci		Yes	*NULL*		
review	longtext	latin1_swedish_ci		Yes	*NULL*		
date	text	latin1_swedish_ci		Yes	*NULL*		

Search

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(11)			No	*None*	auto_increment	
user	text	latin1_swedish_ci		Yes	*NULL*		
keyword	text	latin1_swedish_ci		Yes	*NULL*		
permission	text	latin1_swedish_ci		Yes	*NULL*		
dt	text	latin1_swedish_ci		Yes	*NULL*		

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 22
November 2018

Sub category

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(11)			No	None	auto_increment	
category	varchar(45)	latin1_swedish_ci		Yes	NULL		
sub_category	varchar(45)	latin1_swedish_ci		Yes	NULL		

user

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(11)			No	None	auto_increment	
username	varchar(45)	latin1_swedish_ci		Yes	NULL		
password	varchar(45)	latin1_swedish_ci		Yes	NULL		
email	varchar(45)	latin1_swedish_ci		Yes	NULL		
mobile	varchar(45)	latin1_swedish_ci		Yes	NULL		
address	varchar(45)	latin1_swedish_ci		Yes	NULL		
dob	varchar(45)	latin1_swedish_ci		Yes	NULL		
gender	varchar(45)	latin1_swedish_ci		Yes	NULL		
status	varchar(45)	latin1_swedish_ci		Yes	NULL		
image	longblob		BINARY	Yes	NULL		

CONCLUSION

In this paper, we have studied the problem of top-k high utility item sets mining, where k is the desired number of high utility item sets to be mined. Two efficient algorithms TKU (mining Top-K Utility item sets) and TKO (mining Top-K utility item sets in One phase) are proposed for mining such item sets without setting minimum utility thresholds. TKU is the first two-phase algorithm for mining top-k high utility item sets, which incorporates five strategies PE, NU, MD, MC and SE to effectively raise the border minimum utility thresholds and further prune the search space. On the other hand, TKO is the first one-phase algorithm developed for top-k HUI mining, which integrates the novel strategies RUC, RUZ and EPB to greatly improve its performance. Empirical evaluations on different types of real and synthetic datasets show that the proposed algorithms have good scalability on large datasets and the performance of the proposed algorithms is close to the optimal case of the state-of-the art two-phase and one-phase utility mining algorithms [14], [25]. Although We have proposed a new framework for top-k HUI mining, it has not yet been incorporated with other utility mining tasks to discover different types of top-k high utility patterns such as top-k high utility episodes, top-k closed high utility item sets, top-k high utility web access patterns and top-k mobile high utility sequential patterns. These leave wide rooms for exploration as future work.

BIBLIOGRAPHY

[1] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proc. Int. Conf. Very Large Data Bases, 1994, pp. 487– 499.

[2] C. Ahmed, S. Tanbeer, B. Jeong, and Y. Lee, “Efficient tree structures for high-utility pattern mining in incremental databases,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 12, pp. 1708–1721, Dec. 2009.

[3] K. Chuang, J. Huang, and M. Chen, “Mining top-k frequent patterns in the presence of the memory constraint,”

VLDB J., vol. 17, pp. 1321–1344, 2008.
[4] R. Chan, Q. Yang, and Y. Shen, "Mining high-utility itemsets," in Proc. IEEE Int. Conf. Data Mining, 2003, pp. 19–26.
[5] P. Fournier-Viger and V. S. Tseng, "Mining top-k sequential rules," in Proc. Int. Conf. Adv. Data Mining Appl., 2011, pp. 180–194.
[6] P. Fournier-Viger, C.Wu, and V. S. Tseng, "Mining top-k association rules," in Proc. Int. Conf. Can. Conf. Adv. Artif. Intell., 2012, pp. 61–73.
[7] P. Fournier-Viger, C. Wu, and V. S. Tseng, "Novel concise representations of high utility itemsets using generator patterns," in Proc. Int. Conf. Adv. Data Mining Appl. Lecture Notes Comput. Sci., 2014, vol. 8933, pp. 30–43.
[8] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2000, pp. 1–12.
[9] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining top-k frequent closed patterns without minimum support," in Proc. IEEE Int. Conf. Data Mining, 2002, pp. 211–218.
[10] S. Krishnamoorthy, "Pruning strategies for mining high utility itemsets," Expert Syst. Appl., vol. 42, no. 5, pp. 2371–2381, 2015.
[11] C. Lin, T. Hong, G. Lan, J. Wong, and W. Lin, "Efficient updating of discovered high-utility itemsets for transaction deletion in dynamic databases," Adv. Eng. Informat., vol. 29, no. 1, pp. 16–27, 2015.
[12] G. Lan, T. Hong, V. S. Tseng, and S. Wang, "Applying the maximum utility measure in high utility sequential pattern mining," Expert Syst. Appl., vol. 41, no. 11, pp. 5071–5081, 2014.
[13] Y. Liu, W. Liao, and A. Choudhary, "A fast high utility item sets mining algorithm," in Proc. Utility-Based Data Mining Workshop, 2005, pp. 90–99.
[14] M. Liu and J. Qu, "Mining high utility item sets without candidate generation," in Proc. ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 55–64.
[15] J. Liu, K. Wang, and B. Fung, "Direct discovery of high utility item sets without candidate generation," in Proc. IEEE Int. Conf. Data Mining, 2012, pp. 984–989.
[16] Y. Lin, C. Wu, and V. S. Tseng, "Mining high utility itemsets in big data," in Proc. Int. Conf. Pacific-Asia Conf. Knowl. Discovery Data Mining, 2015, pp. 649–661.
[17] Y. Li, J. Yeh, and C. Chang, "Isolated items discarding strategy for discovering high-utility itemsets," Data Knowl. Eng., vol. 64, no. 1, pp. 198–217, 2008.
[18] J. Pisharath, Y. Liu, B. Ozisikyilmaz, R. Narayanan, W. K. Liao, A. Choudhary, and G. Memik, NU-MineBench version 2.0 dataset and technical report [Online]. Available: http://cucis.ece.northwestern. edu/projects/DMS/MineBench.html, 2005.
[19] G. Pyun and U. Yun, "Mining top-k frequent patterns with combination reducing techniques," Appl. Intell., vol. 41, no. 1, pp. 76–98, 2014.
[20] T. Quang, S. Oyanagi, and K. Yamazaki, "ExMiner: An efficient algorithm for mining top-k frequent patterns," in Proc. Int. Conf. Adv. Data Mining Appl., 2006, pp. 436 – 447.
[21] H. Ryang and U. Yun, "Top-k high utility pattern mining with effective threshold raising Strategies," Knowl.-Based Syst., vol. 76, pp. 109–126, 2015.
[22] H. Ryang, U. Yun, and K. Ryu, "Discovering high utility itemsets with multiple minimum supports," Intell.

Data Anal., vol. 18, no. 6, pp. 1027–1047, 2014.

[23] B. Shie, H. Hsiao, V. S. Tseng, and P. S. Yu, "Mining high utility mobile sequential patterns in mobile commerce environments," in Proc. Int. Conf. Database Syst. Adv. Appl. Lecture Notes Comput. Sci., 2011, vol. 6587, pp. 224–238.

[24] P. Tzvetkov, X. Yan, and J. Han, "TSP: Mining top-k closed sequential patterns," Knowl. Inf. Syst., vol. 7, no. 4, pp. 438–457, 2005.

[25] V. S. Tseng, C. Wu, B. Shie, and P. S. Yu, "UP-Growth: An efficient algorithm for high utility itemset mining," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 253–262.

[26] V. S. Tseng, C. Wu, P. Fournier-Viger, and P. S. Yu, "Efficient algorithms for mining the concise and lossless representation of high utility itemsets," IEEE Trans. Knowl. Data Eng., vol. 27, no. 3, pp. 726–739, Mar. 1, 2015.

[27] C. Wu, P. Fournier-Viger, P. S. Yu, and V. S. Tseng, "Efficient mining of a concise and lossless representation of high utility itemsets," in Proc. IEEE Int. Conf. Data Mining, 2011, pp. 824–833.

[28] J. Wang and J. Han, "TFP: An efficient algorithm for mining top-k frequent closed itemsets," IEEE Trans. Knowl. Data Eng., vol. 17, no. 5, pp. 652–663, May 2005.

[29] C. Wu, Y. Lin, P. S. Yu, and V. S. Tseng, "Mining high utility episodes in complex event sequences," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2013, pp. 536–544.

[30] C. Wu, B. Shie, V. S. Tseng, and P. S. Yu, "Mining top-k high utility itemsets," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 78–86.