Asian Journal of Computer Science and Technology
ISSN: 2249-0701 Vol.8 No.S2, 2019, pp. 57-60

Mining of High Average-Utility Pattern Using Multiple Minimum Thresholds in Big Data

R. Vasumathi[1] and S. Murugan[2]
[1]Research Scholar, [2]Associate Professor
[1&2]PG & Research Department of Computer Science, Nehru Memorial College, Tamil Nadu, India
E-Mail: rsvasumathi.msc@gmail.com

***Abstract*– In the past years most of the research have been conducted on high average-utility itemset mining (HAUIM) with wide applications. However, most of the methods are used for centralized databases with a single machine performing the mining job. Existing algorithms cannot be applied for big data. We try to solve this issue, by developing a new method for mining high average-utility itemset mining in big data. Map Reduce also used in this paper. Many algorithms were proposed only mine HAUIs using a single minimum high average-utility threshold. In this paper we also try solve this by mining HAUIs multiple minimum high average-utility thresholds. We have developed two pruning methods namely Reduction of utility co-occurrence pruning Method (RUCPM) and Pruning without Scanning Database (PWSD).**
Keywords: **Data Mining, Frequent Itemset Mining, High Average Utility Mining, Big Data, Map Reduce**

I.INTRODUCTION

The knowledge discovery in database (KDD) is used to discover implicit and useful information in a large data. Association-rule mining (ARM) or frequent itemset mining (FIM) is an important data mining tasks in KDD, which is used by many researchers and should be broadly studied [1, 13]. Association rule mining or frequent itemset mining is focused in a binary database, and treats all items as having the same importance without considering factors, it is a major limitation. To solve this limitation, the problem of high utility itemset mining (HUIM) [14, 2] was introduced.An important limitation of conventional HUIM is that the utility of an itemset is generally smaller than the utility of its supersets. Hence, traditional HUIM tends to be biased toward finding itemsets of greater length (containing many items), more likely to be high utility itemsets.To highlight the influence of an itemset's length on its utility, and find more useful high utility itemset, Hong et al. [3] proposed the average utility measures, and the problem of high average utility item set mining (HAUIM). The average-utility itemset is determined as the total utility of its items in transactions divided by the number of items in the item set. Various algorithms have been used to more efficiently mine high average-utility item sets[4, 5].But most of the algorithm depends on single minimum average-utility threshold to mine HAUIs.In reality every item must be important to the user. Therefore, it is not correct to measure, the utility of all items in a database using the same minimum utility threshold.

To solve this problem, we proposed a method for HAUIM with MMAU thresholds in this paper. Two efficient pruning strategies used one is RUCPM and another one is PWSD which discovered the prune unpromising itemsets and reducing the search space of HAUIs.

II.RELATED WORK

Shifeng Ren, *et al.,* [7] proposed two novel tighter upper-bound models. One is looser upper bound model and another model is second upper bound model. The first model considers the remaining maximum utility in a transaction to reduce the upper bound on the utilities of itemsets. The second model ignores the unnecessary items in transactions to further tighter upper bound. The upper-bound auub model [6] was proposed to prune unpromising itemsets. The goal is reduce the number of unpromising itemsets considered for HAUIM. Thus the computational time and search space can be greatly reduced and developed pruning strategies are efficient to speed up the mining performance. It is used to the traditional auub model.

In early stages like HAUIM-MMAU [8] algorithm was proposed to discover all HAUIs with multiple minimum high average-utility counts. To improve the mining performanceJerry Chun-Wei Lin, *et al.*,[10] proposed an efficient algorithm with multiple high average utility counts(MEMU).A sorted enumeration tree and a tight upper-bound model were presented to speed up the mining performance and reduce the search space for mining HAUIs. Previous studies ([11], [9]-[12]) mainly focused on finding HAUIs with a single minimum average-utility threshold based on the auub model. Most of the authors used the average utility upper bound to reduce the number of candidates and the time for generating them. In Hong *et al.*,[14] used average utility upper-bound to prune generated candidates and create the set of item sets having upper-bound value are greater than or equal threshold. It is used auub value to reduce the number of candidates by using items which have suitable up. They used ub to prune two times. Firstly, they removed the unsuitable item and then calculated again without an unsuitable item. However, this method took much time.

III.HADOOP

Hadoop is an open-source software framework. It is scalable

fault-tolerant Virtual Grid operating system architecture for data storage which is fault-tolerant highly-bandwidth clustered storage architecture. Hadoop Distributed File System and Map Reduce are taken from the Hadoop software. It runs the Map Reduce for distributed data processing and is works with structured and unstructured data.

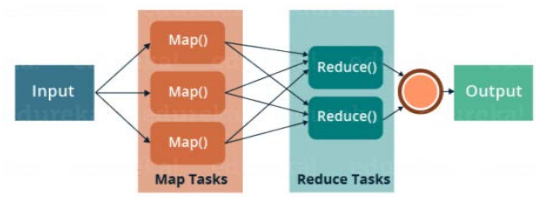

Fig.1 Map Reduce

Map Reduce used to three phases. (i) Counting Phase (ii) Transformation phases (iii) Mining phase.

A. *Counting Phase:*In the first phase, takes the one map reduces pass to parallel count in distributed database. This phase is divided into two stages one is Map stage and another one is Reduce stage.

1. *Map Stage:* In the map stage, each mapper takes the transactions Tc= $\{i_1,i_2,i_3,\ldots..i_l\}$ in distributed database. Every mapper outputs a key-value pairs.
2. *Reduce Stage:* In reducing stage the output of the key-value pair of the mappers is taking the reducer. The key-value pairs having the same key are collected into the same reducer.

B. Transformation Phase: In transformation phase, it removes the all low 1-itemsets from distributed database and sorts remaining items in an ascending order.

C. Mining Phase: In mining phases, we have to find HAUIs using various iterations. Initially, a variable k is set to 0. It generates HAUIM have a length greater than k. In the K-th iteration, all the HAUIs of length K are discovered by performing a Map Reduce Pass. It can be divided into two cases. (i) K=1, and (ii) K≥2.

IV.THE PROPOSED HAUI-MMAU ALGORITHM

In this section HAUI-MMAU consists of two phases. The first phase performs a breath-first search to mine HAUUBIs. Second Phase, performed to identify the actual HAUI from the HAUUBIs. It is discovered in the first phase. It is scan in the additional database.

A. Reduction of Utility Co-Occurrence Pruning Method (RUCPM)

Reduction of utility co-occurrence pruning Method (RUCPM). It is designed to reduce the search space, reduce the number of join operations, and pruning the unpromising item sets. The RUCPMstrategies are applied for the generations of K-item sets. The correctness and the completeness of the HAUI-MMAU with the RUCPM strategy for discovered HAUIs.Let there be an itemset Y^{k-1} and an itemsetY^k that is an extension of Y^{k-1}. The itemsets are sorted by the ascending order.

B. Pruning without Scanning Database, PWSD

Also, developed the RUCPM strategy is reduce the number of unpromising candidates in the first phase, calculating the set of actual HAUIs in the second phase a very time-consuming process. To solve this issue, further proposes an efficient pruning before calculation strategy (PWSD) to pruning itemsets without performs a database scan.

C. Basic Concepts and Problem Statement

Let a finite itemset I=$\{i_1,i_2,\ldots..,i_m\}$ with in m distinct items and set of transaction dataset D=$\{T_1,T_2,\ldots..,T_n\}$.Every transaction $T_q \epsilon$ D ($1 \leq q \leq m$) is the subset of I and have a unique identifier q, is called TID. Also, every item i_1 in a transaction Tq have a purchase quantity (a not negative integer) indicate that q (i_1, Tq). A profit table P = {P (i_1), P (i_2)… P (i_m)} specifies the unit profit of every items i_1. Y= $\{i_1,i_2,\ldots..,i_k\}$ is a set of K distinct item set where K is the length of the item set. An itemset Y is said to contained in a transaction Tqif$Y \subseteq Tq$.

TABLE I TRANSACTION DATABASE

TID	Transaction Items
T1	A(1),B(2),C(4),D(3),E(8),F(2)
T2	A(3),B(3),C(8)
T3	A(2),B(5),D(5),E(7)
T4	A(4),C(4),F(2),G(12)
T5	A(5),B(2),C(3),D(5),F(8)
T6	E(1),F(1),A(3)

The quantitative database items with their quantities are shown in Table I. This will be used to describe the HAUPM. It is assumed Five transactions are used in the Seven items, i.e. (a to g). The external utility value of every item is shown Table II.

TABLE II UTILITY TABLE

Item	Profit
A	2
B	6
C	5
D	2
E	7
F	4
G	2

It is assumed that the minimum average-utility thresholds of all items for the running example are defined as :{maua:(65), mau b:(70), mau c:(66), mau d:(67), mau e:(65), mau f:(68),mau g:(69)}

TABLE III MAP REDUCE IN FULL TRANSACTIONS

Distributed Database			Mapper Stage			Reducer Stage		
T1	ABCDEF	→	Mapper1	<a,104><b,104> <c,104><d,104> <e,104><f,104>	→	Reducer 1	A	<a,104><a,64> <a,93><a,60> <a,79><a,17>
T2	ABC	→	Mapper 2	<a,64><b,64> <c,64>	→		B	<b,104><b,64> <b,93><b,79>
T3	ABDE	→	Mapper 3	<a,93><b<93> <d,93><e,93>	→	Reducer 2	C	<c,104><c,64> <c,60><c,79>
T4	ACFG	→	Mapper 4	<a,60><c,60> <f,60><g,60>	→		D	<d,104><d,93> <d,79>
T5	ABCDF	→	Mapper 5	<a,79><b,79> <c,79><d,79> <f,79>	→	Reducer 3	E	<e,104><e,93> <e,17>
T6	EFA	→	Mapper 6	<e,17><f,17> <a,17>	→		F	<f,104><f,17>

For the above example, we take six transactions. First, to find the high average utility pattern mining from the transaction database. Second, to find the minimum threshold value through the high average utility pattern mining. Find the minimum threshold value.Minimum threshold value is calculated as MMAU = {maua:(65), mau b:(70), mau c:(66), mau d:(67), mau e:(65),mau f:(68),mau g:(69)}= 65.The transactions are reduced using the minimum threshold value (65). The resulting transactions are used as inputs for the map reduce process. After map reducing we will get only 3 transactions shown in the table IV.

TABLE IV MAP REDUCE

Distributed Database			Mapper			Reducer		
T1	ABC DEF	→	M1	<a,104> <b,104> <c,104> <d,104> <e,104> <f,104>	→	R1	A	<a,104> <a,93> <a,79>
							B	<b,104> <b,93> <b,79>
T3	ABDE	→	M2	<a,93> <b,93> <d,93> <e,93>	→	R2	C	<c,104> <c,79>
							D	<d,104> <d,93>
T5	ABC DF	→	M3	<a,79> <b,79> <c,79> <d,79> <f,79>	→	R3	E	<e,104> <e,93>
							F	<f,104> <f,79>

TABLE V MAPPER FOR K=1

M1	A(1),B(2),C(4) D(3),E(8),F(2)	<{a},1,{b(2),c(4),d(3),e(8),f(2)}> <{b},2,{c(4),d(3),e(8),f(2)}> <{c},4,{d(3),e(8),f(2)}> <{d},3,{e(8),f(2)}> <{e},8,{f(2)}> <{f},2,{ϕ}>
M2	A(2),B(5),D(5),E(7)	<{a},2,{b(5),d(5),e(7)}> <{b},5,{d(5),e(7)}> <{d}5,{e(7)}> <{e},7,{ϕ}>
M3	A(5),B(2),C(3) ,D(5),F(8)	<{a},5,{b(2),c(3),d(5),f(8)}> <{b},2,{c(3),d(5),f(8)}> <{c},3,{d(5),f(8)}> <{d},5,{f(8)}> <{f},8,{ϕ}>

In mining phase is k=1, the reduced transaction is shown in table V and the reducer table is shown in table VI.

TABLE VI REDUCER FOR K=1

Reducer 1	A	<{a},1,{b(2),c(4),d(3),e(8),f(2)}> <{a},2,{b(5),d(5),e(7)}> <{a},5,{b(2),c(3),d(5),f(8)}>
	B	<{b},2,{c(4),d(3),e(8),f(2)}> <{b},5,{d(5),e(7)}> <{b},2,{c(3),d(5),f(8)}>
Reducer2	C	<{c},4,{d(3),e(8),f(2)}> <{c},3,{d(5),f(8)}>
	D	<{d},3,{e(8),f(2)}> <{d}5,{e(7)}> <{d},5,{f(8)}>
Reducer 3	E	<{e},8,{f(2)}> <{e},7,{ϕ}>
	F	<{f},2,{ϕ}> <{f},8,{ϕ}>

In mining phase is k=2, the reduced transactionis shown in table VII and the reducer table isShown in table VIII.

TABLE VII MAPPER FOR K=2

M1	<{a},1,{b(2),c(4) ,d(3),e(8),f(2)}>	<{ab},3,{c(4),d(3), e(8),f(2)}> <{ac},5,{d(3),e(8),f(2)}> <{ad},4,{e(8),f(2)}> <{ae},9,{f(2)}> <{af},2,{ϕ}>
	<{a},2,{b(5), d(5),e(7)}>	<{ab},7,{d(5),e(7)}> <{ad},7,{e(7)}> <{ae},9,{ϕ}>
	<{a},5,{b(2),c(3) ,d(5),f(8)}>	<{ab},7,{c(3),d(5),f(8)}> <{ac},8,{d(5),f(8)}> <{ad},10,{f(8)}> <{af},13,{ϕ}>
M2	<{b},2,{c(4),d(3), e(8),f(2)}>	<{bc},6,{d(3),e(8),f(2)}> <{bd},5{e(8),f(2)}> <{be},10,{f(2)}> <{bf},4, {ϕ}>
	<{b},5,{d(5),e(7)}>	<{bd},12,{e(7)}> <{be},14,{ϕ}>

	<{b},2,{c(3), d(5),f(8)}>	<{bc},5,{d(5),f(8)}> <{bd},7,f(8)}> <{bf},10, {ϕ}>
M3	<{c},4,{d(3), e(8),f(2)}>	<{cd},7,{e(8),f(2)}> <{ce},8,{f(2)}> <{cf},6,{ϕ }>
	<{c},3,{d(5),f(8)}>	<{cd},8,{f(8)}> <{cf},11, {ϕ }>
M4	<{d},3,{e(8),f(2)}>	<{de},11,{f(2)}> <{df},5, {ϕ }>
	<{d}5,{e(7)}>	<{de},12,{ϕ }>
	<{d},5,{f(8)}>	<{df},12,{ϕ }>
M5	<{e},8,{f(2)}>	<{ef},10, {ϕ }>

TABLE VIII REDUCER FOR K=2

Reducer 1	AB	<{ab},3,{c(4),d(3),e(8),f(2)}> <{ab},7,{d(5),e(7)}> <{ab},7,{c(3),d(5),f(8)}>
	AC	<{ac},5,{d(3),e(8),f(2)}> <{ac},8,{d(5),f(8)}>
	AD	<{ad},4,{e(8),f(2)}> <{ad},7,{e(7)}> <{ad},10,{f(8)}>
	AE	<{ae},9,{f(2)}> <{ae},9,{ϕ}>
	AF	<{af},2,{ϕ}> <{af},13,{ϕ}>
Reducer 2	BC	<{bc},6,{d(3),e(8),f(2)}> <{bc},5,{d(5),f(8)}>
	BD	<{bd},5{e(8),f(2)}> <{bd},12,{e(7)}> <{bd},7,f(8)}>
	BE	<{be},10,{f(2)}> <{be},14,{ϕ}>
	BF	<{bf},10, {ϕ}>
Reducer 3	CD	<{cd},7,{e(8),f(2)}> <{cd},8,{f(8)}>
	CE	<{ce},8,{f(2)}>
	CF	<{cf},6,{ϕ }> <{cf},11, {ϕ }>
Reducer 4	DE	<{de},11,{f(2)}> <{de},12,{ϕ }>
	DF	<{df},5, {ϕ }> <{df},12,{ϕ }>
	EF	<{ef},10, {ϕ }>

V. CONCLUSION

In this paper the high average utility pattern mining with multiple minimum thresholds, method was discussed to mine high-average utility itemsets. This method consists two phases (i) RUCPM (ii) PWSD.The RUCPM is designed to reduce the search space. The Second method PWSD is used to reduce the number of HAUUBIs at the initial stage of the second phase. Map reduce methods are used in ordered to mining in Bigdata. Results show that these methods can efficiently find the HAUIs and reduce the number of transactions.

REFERENCES

[1] R. Agarwal, T. Imielinski, and A. Swami, "Database mining: a performance perspective", *IEEE Trans. Knowl. Data Eng*, Vol. 5, No.6, pp.914–925 1993.

[2] Y.Liu, W. Liao, and A.K Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets", *Springer*,Vol. 3518, pp. 689–695, 2005

[3] T.P Hong, C.H. Lee and S.L. Wang, "Effective utility mining with the measure of average utility", *Expert Syst. Appl.* Vol. 38, No. 7, pp. 8259–8265, 2011.

[4] C.W.Lin, T.P.Hong, and W.H,Lu," Efficiently mining high average utility itemsets with a tree structure", *Springer*,Vol. 5990, pp. 131–139, 2010.

[5] G.C. Lan, T.P. Hong, and V.S. Tseng, "A projection-based approach for discovering high average-utility itemsets" , *J. Inf. Sci. Eng.*Vol. 28, No.1, pp. 193–209, 2012.

[6] T.P. Hong, C.H. Lee, and S.L. Wang, "Effective utility mining with the measure of average utility", *Expert Syst. Appl.*, Vol. 38, No. 7, pp. 8259_8265, 2011.

[7] J.C.W. Lin, S. Ren, and P. Fournier-Viger, "EHAUPM: Efficient high average-utility pattern mining with tighter upper bounds", *IEEE Access*, Vol. 5, pp. 12927_12940, 2017.

[8] T. Lu, B. Vo, H. T. Nguyen, and T.P. Hong, "A new method for mining high average utility itemsets", *J.Computer Information Systems and Industrial Management. Springer*, pp. 33-42, 2014.

[9] J. C.W. Lin, T. Li, P. Fournier-Viger, T.P. Hong, J. Zhan, and M. Voznak, "An efficient algorithm to mine high average-utility itemsets", *Adv. Eng. Informat.*, Vol. 30, No. 2, pp. 233 - 243, 2016.

[10] Jerry Chun-Wei Lin, Shifeng Ren, and Philippe Fournier-Viger "MEMU: More Efficient Algorithm to Mine High Average-Utility Patterns with Multiple Minimum Average-Utility Thresholds", *IEEE Access*, Vol. 6, 2018.

[11] J. C.W. Lin, T. Li, P. Fournier-Viger, T.P. Hong and J.-H. Su, ``Efficient mining of high average-utility itemsets with multiple minimum thresholds", *Proc. Ind. Conf. Data Mining*, pp. 14 - 28, 2016.

[12] C.W. Lin, T.P. Hong, and W. H. Lu, "Efficiently mining high average utility item sets with a tree structure", *Proc. Int. Conf. Intell. Inf. Database Syst.*, pp. 131 - 139, 2010.

[13] R. Agrawal, and R. Srikant, "Fast algorithms for mining association rules in large databases", *In International Conference on Very Large Data Bases*, pp. 487–499 1994.

[14] R. Chan, Q. Yang, and Y.D. Shen, "Mining high utility itemsets", *In IEEE International Conference on Data Mining*, pp. 19–26, 2003.