International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020

Data Mining Association Rules using Probabilistic Functions without Predefined Weights

Amit Rehalia, Samar Wazir, Tabrez Nafis

Abstract*: Data mining is the procedure of identifying the important and relevant data from large heterogeneous databases. Data mining plays an important role because of its usage in various domains. The transaction in the data mining defines the profit of the items associated with it. Earlier algorithms were proposed to measure the w-support without assigning predefined weights to determine the important transactions using the HITS model. Significant items are extracted from the databases using the quality of the transactions. However, there is considerable overhead in computing the w-support, as it requires four to five iterations. In this paper, two algorithms are proposed which uses the Poisson distribution and Normal distribution while computing the w-support without using the pre-assigned weights. The Poisson distribution uses the probability mass functions whereas the Normal distribution uses the probability density function to compute the w-support. The experiments were executed on various standard datasets. The results of our proposed algorithms show a considerable decrease in normalization time to compute the w-support as compared to the HITS model. Hence our algorithms provide better performance with respect to execution time and a number of significant items.

Keywords*: Association Rule Mining, Data mining, Poisson distribution, Normal distribution, Weighted-Support.

I. INTRODUCTION

Data mining has become a noticeably and most desired technology nowadays to extract vital information from various application domains like E-commerce, Web mining, text mining, etc. to retrieve the relevant information from these domains [1], [5], [6], [7], [8]. The growth of data increased manifolds due to the penetration of the digital information to everyone's use, and the existing mining concepts are unable to handle such large data [2]. The existing data mining methods have been optimized to support big data. Data mining is the most important step of the KDD and provides interesting and useful patterns from big data and is being used in a variety of domains and applications like information technology, security, marketing, infrastructure, health etc.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
Amit Rehalia*, Department of Computer Science & Engineering. Jamia Hamdard, New Delhi-110062, India .Email:amit.reh@gmail.com
Samar Wazir, Department of Computer Science & Engineering. Jamia Hamdard, New Delhi-110062, India. Email: samar.wazir786@gmail.com
Md. Tabrez Nafis, Department of Computer Science & Engineering. JamiaHamdard, New Delhi-110062, India. Email: tabrez.nafis@gmail.com

The association rule uses the support and confidence measures, where the weights of the transactions are not being considered [9], and is used to find out the frequent item set. *Frequent Itemset Mining (FIM)* [3] is a method in which frequent itemsets are generated from datasets using minimum support value and the generated candidate which is greater than minimum support is known as frequent itemsets. The purchasing habits of the customers are analyzed by applying the FIM concepts. Take an example, if a passenger books an air ticket, how likely he also book hotel and taxi in that destination [4].

There are many association mining algorithms associated with the pre-assigned weight [10], [11], [12], [13] to determine important items of transaction in terms of some profit. However, there is considerable overhead and maintenance to keep these pre-assigned weights with the transactions. Also, the existing pre-assigned weights algorithms emphasize the importance of the transactions on the basis of the number of items and size i.e. if the transactions have more items then it will be considered more important than the transactions with the few items. Most datasets do not support the pre-assigned weight like on-click data.

In this paper, weighted-support (w-support) is calculated which differs from classical support of the transactions [14], which is being computed on the basis of the internal relationship of the dataset. It uses the concepts Kleinberg's HITS algorithm [15] and is different from the WARM (weighted association rule mining) [11], [12] in which the profits of the items are provided for each transaction. The initial weights are computed using the binary attributes through the internal structure of the database. To compute the w-support faster the proposed algorithm extends the assumptions of Poisson and normal distribution [16], hence it is named as mining weighted association rules using Poisson/normal distribution without preassigned weight. The experimental result depicts the efficiency and effectiveness of our algorithm with respect to execution time and numbers of significant items discovered. Following steps are performed for the execution of the proposed algorithm:

1) Execution of data mining association rule without predefined weights [14] using various datasets.
2) Execution of data mining association rule using Poisson distribution without predefined weights on various datasets.
3) Execution of data mining association rule using Normal distribution without predefined weights on various datasets.

4) Comparison with respect to initialization time, execution time and a number of significant item sets generated for the above algorithms.

II. RELATED WORK

Data mining retrieves the relevant and meaningful data from a large-scale database and was proposed in [9] for the association rule. It uses the support and confidence methodology to prune the frequent itemsets. Apriori algorithm [17] was used to find out the frequent itemsets from data in an iterative fashion. Apriori works on a pruning principle that if an itemset is infrequent its superset is not generated. To generate a large number of itemsets efficiently, DHP for large itemset generation is proposed [18], Jiawei Han introduces a Frequent Pattern Growth where frequent patterns are discovered using pattern fragment growth [19] and are also called FP-Tree to store vital information in compressed format [20]. All the above algorithms also called the breadth-first search algorithm. The Eclat is another algorithm [21], [22], [23] for the pruning of frequent itemsets on a vertical database and called a depth-first search technique. All these algorithms treated all the transactions uniformly and there are no profits associated with the items.

Weighted association rule mining introduces the concept of weight to the items while discovering the frequent itemsets [10], [11], [12], [13]. It focuses on the amount associated with the items in the transaction e.g. purchase of costly items like buying a BMW car is more important than buying lots of toys, toffees, biscuits, etc. Costs are assigned to both the items and transactions, and association rules are determined based on the weight support concepts [10]. WIS algorithm was presented which determines the itemsets on the basis of weighted support and the user-defined threshold. Instead of using the weights of both items and transactions, Cai [11] proposed two new algorithms named MINWAL(O) and MINWAL(W) where the weight of the items is considered. The weight associated with the items determines the significance and profitability to the end-user. However, the downward closure property is being broken, hence increasing the algorithm's execution time and complexity. To overcome the same, WARM algorithm (Weighted Association Rule Mining) was proposed by Tao [13], which uses the "weighted downward closure property" through the weighted support.

The item ranking approach was proposed by Wang and Su [24], where the "cross-selling" effect of the items determines the profitability and importance of the items. The direct graph is constructed on the basis of Items and association rules. The nodes in the graph are treated as items and links are represented by association rules. It provides the analogy that is being supported through the HITS model [15] to provide a relationship between transactions and items as it is being provided between hubs and authorities. A profitable transaction consists of good items and vice-versa. Ke Sun and F Bai [14] proposed such a model to compute the w-support using the link-based model to compute the profitable transactions. The transactions are treated as "hubs" and items are represented as "authorities", and computed using (1) and (2):

$$auth(i) = \sum_{T:i\in T} hub(T) \quad (1)$$

$$hub(T) = \sum_{i:i\in T} auth(i) \quad (2)$$

Here, the Database D consists of $\{T_1, T_2, \ldots\ldots T_m\}$ transactions and i= $\{I_1, I_2, \ldots..I_n\}$ be the set of items.

The hubs weights of all the transactions in a database are obtained when the values are the same through each iteration. The weighed support of item Y can be computed as:

$$weightedSupport(Y) = \frac{\sum_{T:X\subset T\wedge T\in D} hub(T)}{\sum_{T:T\in D} hub(T)} \quad (3)$$

The algorithm for mining of important itemsets requires at least four to five iterations as defined in (1) and (2) to compute the weights of the hub for all the transactions in the database. S. Wazir et al. [25] uses the Poisson and Normal distribution-based model to compute the frequent itemsets for an uncertain database.

III. PROPOSED SOLUTION

After finding the Authority and hub weights at 0^{th} iteration using (1) and (2), our proposed solution computes the hub weights of all the transactions using the Poisson and Normal distribution function. Instead of repeating the steps four to five times to evaluate the hub weights, the proposed algorithm is more efficient and optimized to compute the hub weights using the probabilistic functions as defined in (4) and (5). The final weight support is computed using (3) and the algorithm is extended to the Apriori algorithm [17] to obtain the results.

A. Computing Weighted Support using Poisson distribution

As per the Ke Sun and F Bai [14], the hub weight of the transactions is being computed using four to five iterations to provide the normalization of the data. These iterations were necessary to converge the values of the hubs and authorities. Weighted support using Poisson distribution is proposed to obtain the hub weights of the transactions. In this model, if μ is the authority value that is being calculated using (1) and (2), then the finalAuthPoisson(i) can be calculated as defined in (4) using the Poisson distribution's Cumulative Distribution Function(CDF) [26][27]

$$finalAuthPoisson(i) \approx 1 - e^{-\mu} \sum_{i=0}^{I_n} \frac{(\mu_i)}{i!} \quad (4)$$

In other words, a Poisson distribution with a mean of the authority values obtained as per (1) and (2) is being calculated. A *finalAuthPoisson* value is being computed on the basis of the probability mass function (PMF) for the distribution.

B. Computing Weighted Support using Normal distribution

Computing the weighted support based on the *Normal distribution* is used for dense large certain databases. In this approach,

if μ is the authority value that is being calculated using (1) and (2), and σ^2 defines variance for the authority values of itemsets with T_m number of transactions, the finalAuthNormal(i) can be calculated as defined in (5) and (6) using the *Normal distribution's Cumulative Distribution Function*(CDF) [28][29] with mean μ and variance σ^2

$$finalAuthNormal(i) \approx 1 - \sum_{i=0}^{I_n} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(i-\mu)^2}{2\sigma^2}} \quad (5)$$

Here σ^2 can be calculated as,

$$\sigma^2 = \sum_{t_j \in T_U} P(Item \subseteq t_j) \times \left(1 - P(Item \subseteq t_j)\right) \quad (6)$$

P is the probability for Item in transaction t_j.

Here, a Normal distribution with a mean and standard deviation of the authority values obtained as per (1) and (2) is being calculated. A *finalAuthNormal* value is being computed on the basis of the probability density function (PDF) for the distribution.

IV. EXPLANATION WITH EXAMPLE

Consider the example database as defined in Table 1

Table-I: Database

TID	Transaction
T1	1,2,3,4,5
T2	3,6,7
T3	1,2
T4	1
T5	3,6,7,8
T6	1,7,8

The value of items (i.e authority) using the HITS model at 0^{th} iterations is defined in table2. The Poisson distribution and Normal distribution is applied to these values itself, rather than making iterations for the values to converge.

Table-II: Values of Poisson and Normal distribution on the basis of the 0^{th} iteration of HITS model

Authority	Value	Poisson Distribution	Normal Distribution
Auth (1)	11	0.0722	0.080
Auth (2)	7	0.1396	0.139
Auth (3)	12	0.0481	0.049
Auth (4)	5	0.0916	0.080
Auth (5)	5	0.0916	0.080
Auth (6)	7	0.1396	0.139
Auth (7)	10	0.0993	0.113
Auth (8)	7	0.1396	0.139

Table 3 defines the hub weights computed using the example database

Table-III: Hub Weights of the example database

TID	Transaction	Hub Weights using Poisson Distribution	Hub Weights using Normal Distribution	Hub Weights using Ke Sun and F Bai [14] model
T1	1,2,3,4,5	0.569	0.537	0.520
T2	3,6,7	0.368	0.379	0.435
T3	1,2	0.272	0.276	0.233
T4	1	0.093	0.100	0.148
T5	3,6,7,8	0.547	0.554	0.542
T6	1,7,8	0.399	0,418	0.412

Table 4 illustrates the weighted support achieved through Poisson distribution, Normal Distribution and Ke Sun and F Bai [14] model for the example database.

Table-IV: Values of Weighted support using Poisson, Normal distribution and Ke Sun model model

ItemSet	W-Support using Poisson Distribution	W-Support using Normal Distribution	W-support using Ke Sun and F Bai [14] model
1	0.59	0.59	0.57
2	0.37	0.36	0.33
3	0.66	0.65	0.65
4	0.25	0.24	0.23
5	0.25	0.24	0.23
6	0.41	0.41	0.43
7	0.58	0.60	0.61
8	0.42	0.43	0.42

V. WEIGHTED SUPPORT MINING ALGORITHM

The downward closure property is being applied while proposing the mining algorithm using weighted support. It says that if the items are greater than the minimum w-support, all the subsets of the items also satisfies the conditions. The given threshold is being applied to the weighted support for the mining rules of w-support and w-confidence. Hence, our proposed model extends the Apriori algorithm [17] and consists of two algorithms.

A. Weighted Support Mining Algorithm using Poisson distribution

The algorithm uses the HITS models to compute authority (items) and hubs (transactions) for the whole database. To normalize the authority values, it uses the Poisson distribution function as defined in (4). After that, the Apriori algorithm prunes the relevant item-sets in a level-wise manner using the join and prune stages. The itemsets which are greater than the user-defined minwsupp are said to be relevant itemsets.

Step 1: Initialize the each item i in *auth(i)* to 1
Step 2: for(*k=0; k< num_item; k++*) **do begin**
Step 3: *auth'(i)* = 0 for each item *i*
Step 4: **for all** transactions, $t \in D$ **do begin**
Step 5: $hub(t) = \sum_{i:i \in T} auth(i)$
Step 6: *auth'(i)+= hub(t) for each item i* ∈ *t*
Step 7: **end** //end of inner-for loop
Step 8: $finalAuthPoisson(i) \approx 1 - e^{-\mu} \sum_{i=0}^{I_n} \frac{(\mu_i)}{i!}$
Step 9: end //end of outer-for loop
Step 10: $L_1 = \{ \{i\}: wsupp(i) \geq minwsupp \}$
Step 11: for ($k=2; L_{k-1} \neq \phi$; k++) **do begin**
Step 12: C_K = apriori-gen(L_{K-1})
Step 13: **for all** transactions $t \in D$ **do**
Step 14: C_t = subset (C_k,t)
Step 15: **for all** candidate c ∈ C_t **do**
Step 16: c.wsupp += hub(t)

Step 17: H += hub(t)
Step 18: **end**
Step 19: $L_K = \{ c \in C_K \mid c.wsupp/H \geq minwsupp$
Step 20: end
Step 21: Answer L =$\cup_k L_k$

Weighted Support Mining Algorithm using Normal distribution

The Normal distribution functions as defined in (5) and (6) are used to normalize the authority values. Apriori algorithm is used to prune the relevant itemsets in a level-wise manner using the join and prune stages. The itemsets which are greater than the user-defined minwsupp are said to be relevant itemsets.

Step 1: Initialize the each item i in *auth(i)* to 1
Step 2: for(*k=0; k< num_item; k++*) **do begin**
Step 3: *auth'(i)* = 0 for each item *i*
Step 4: **for all** transactions, $t \in D$ **do begin**
Step 5: $hub(t) = \sum_{i:i \in T} auth(i)$
Step 6: *auth'(i)+= hub(t) for each item* $i \in t$
Step 7: **end**
Step 8: $finalAuthNormal(i) \approx 1 - \sum_{i=0}^{I_n} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(i-\mu)^2}{2\sigma^2}}$
Step 9: $\sigma^2 = \sum_{t_j \in T_U} P(Item \subseteq t_j) \times \left(1 - P(Item \subseteq t_j)\right)$
Step 10: end
Step 11: $L_1 = \{ \{i\}: wsupp(i) \geq minwsupp \}$
Step 12: for ($k=2; L_{k-1} \neq \phi$; k++) **do begin**
Step 13: C_K = apriori-gen(L_{K-1})
Step 14: **for all** transactions $t \in D$ **do**
Step 15: C_t = subset (C_k,t)
Step 16: **for all** candidate c $\in C_t$ **do**
Step 17: c.wsupp += hub(t)
Step 18: H += hub(t)
Step 19: **end**
Step 20: $L_K = \{ c \in C_K \mid c.wsupp/H \geq minwsupp$
Step 21: end
Step 22: Answer L =$\cup_k L_k$

VI. EXPERIMENTS AND RESULT ANALYSIS

This section details the analysis of the experiments done on six datasets. The dataset is Kosarak [33] and the remaining five datasets are chess.dat, retail.dat, T40I10D100K.dat, mushroom.dat, connect.dat [30]. Table 5 outlines the total transactions and distinct items available in these datasets.

Table-V: No of Transactions and distinct items

DataSet	No of Transactions	Distinct Items
chess.dat	3196	75
retail.dat	88162	16470
T40I10D100K.dat	100000	942
mushroom.dat	8124	119
connect.dat	67557	129
Kosarak	990002	41270

The above experiments are performed on the proposed two algorithms which are an extension of the Apriori algorithm. The results with respect to normalization time (calculating the weight), execution time and a number of significant items extracted are compared with our proposed algorithm and the Ke Sun and F Bai [14] model (HITS model). The algorithms are written in JAVA version "1.8.0_151" using the Apriori code [31] and utilizing the jar file of Apache [32] for mathematical functions. The machine configuration where these algorithms are executed comprised of Intel(R) Core(TM) i7-4790 CPU @ 3.6GHz processor with 8GB of RAM, Ubuntu 18.04 operating system. The results between the proposed mining algorithm using Poisson distribution, Normal distribution and HITS model (using Ke Sun and F Bai [14]) are executed on various datasets [34], [35], [36], [37].

Fig. 1, Fig. 2, Fig. 3 depicts the graphs of Initialization time (Normalization time to achieve weights) in milliseconds, execution time in milliseconds and number of significant items generated on chess.dat respectively.

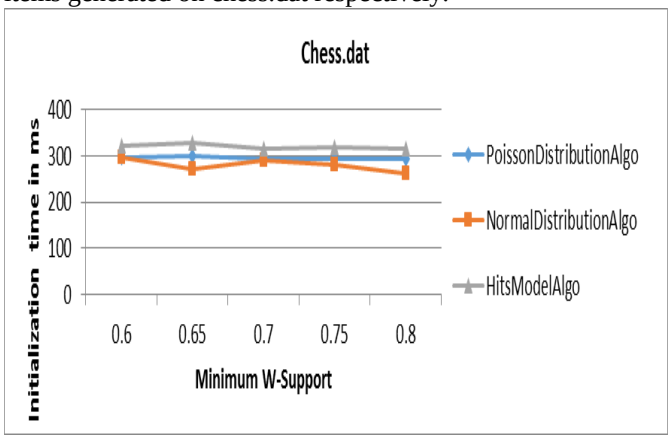

Fig. 1. Initialization time for Poisson distribution, Normal distribution and HitsModel Algo for Chess.dat

It is being clear from Fig. 1 that the initialization time i.e obtaining the final weights for items using Poisson distribution and Normal Distribution is far less than the Hits Model

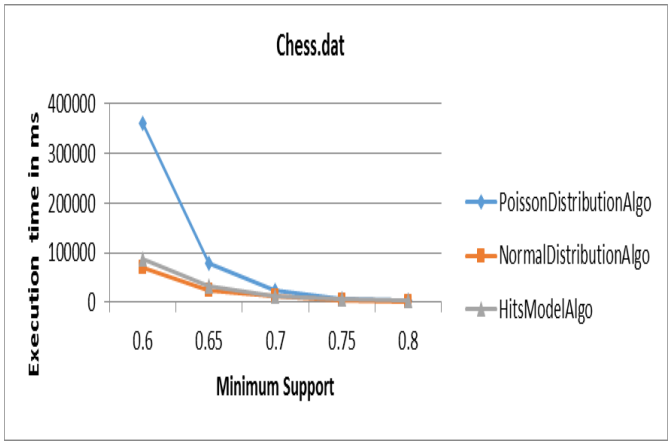

Fig. 2. Execution time for Poisson distribution, Normal distribution and HitsModel Algo for Chess.dat

Fig. 2 shows the execution time by our proposed algorithm is higher than the HITS model because it generates the number of significant items as shown in Fig. 3.

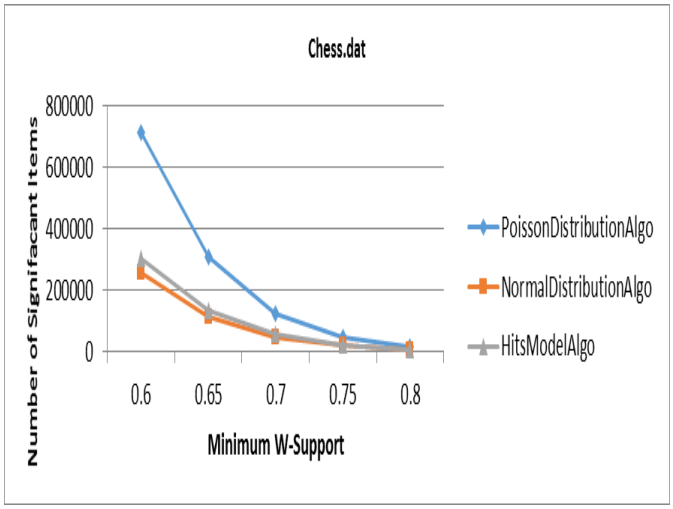

Fig. 3. Number of significant Items for Poisson distribution, Normal distribution and HitsModel Algo for Chess.dat

Fig. 3 depicts that the number of significant items generated using the proposed Poisson distribution is far higher than the HITS model. Also, the results using Normal distribution is better than the HitsModel algorithm. Similar scenarios are depicted for retail.dat dataset in Fig. 4, Fig. 5 and Fig. 6 for Initialization time taken, execution time and the number of significant items extracted respectively.

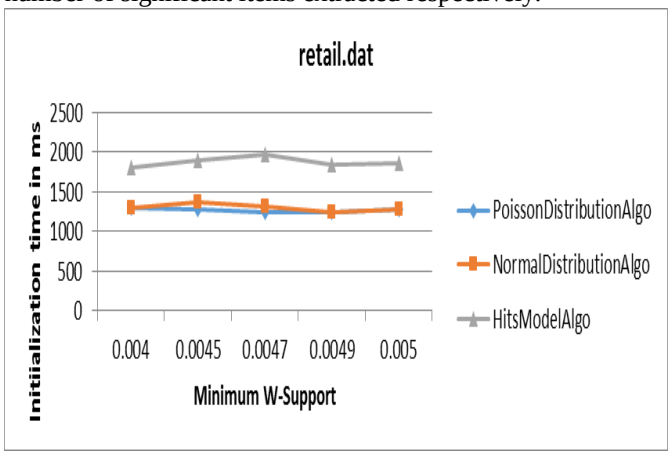

Fig. 4. Initialization time for Poisson distribution, Normal distribution and HitsModel Algo for retail.dat

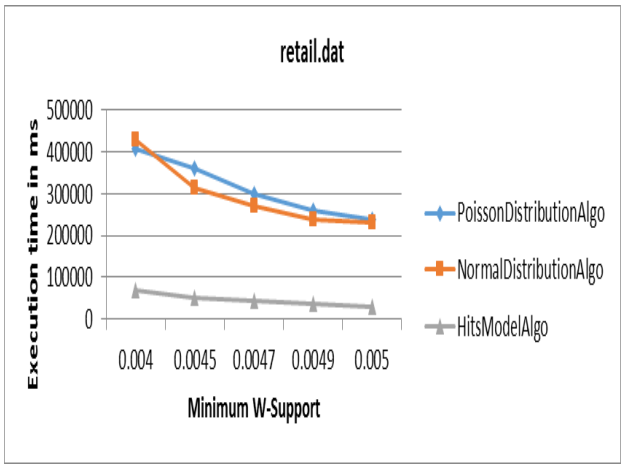

Fig. 5. Execution time for Poisson distribution, Normal Distribution and HitsModel Algo for retail.dat

The execution for the "retail.dat" dataset is depicted in Fig. 4(Initialization time), Fig. 5 (Execution time), and Fig. 6 (Number of significant items).

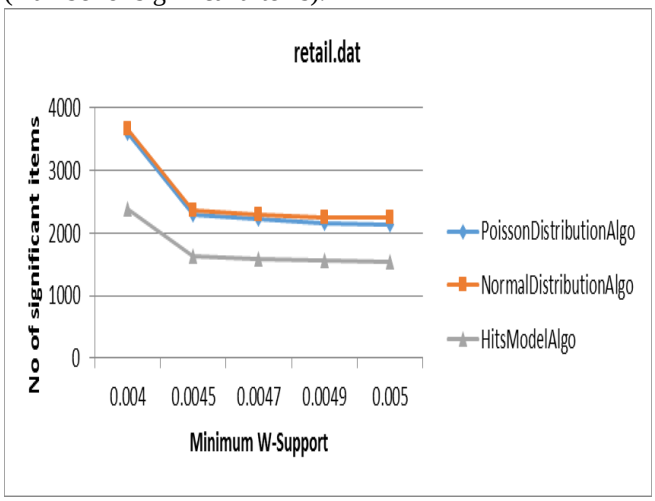

Fig. 6. Number of significant Items for Poisson distribution, Normal distribution and HitsModel Algo for retail.data

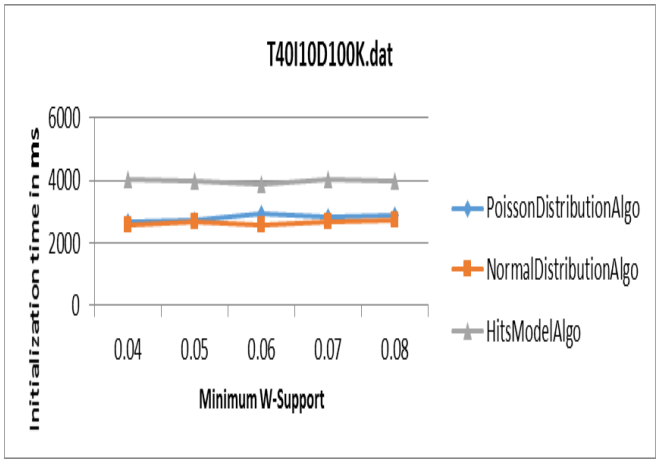

Fig. 7. Initialization time for Poisson distribution, Normal distribution and HitsModel Algo for T40I10D100K.dat

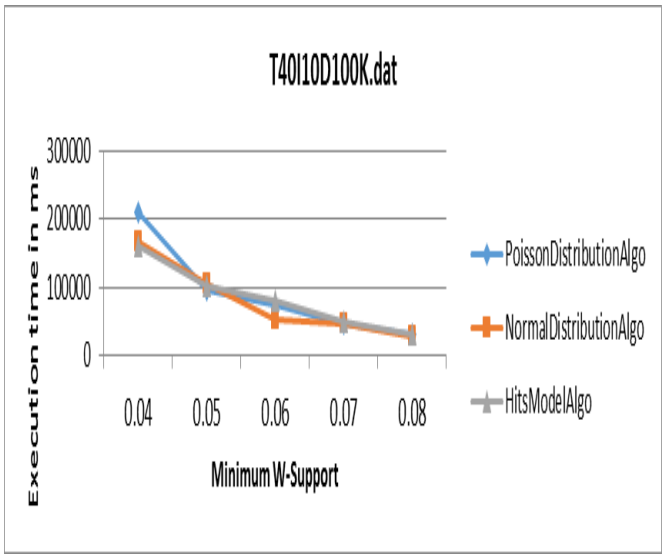

Fig. 8. Execution time for Poisson distribution, Normal distribution and HitsModel Algo for T40I10D100K.dat

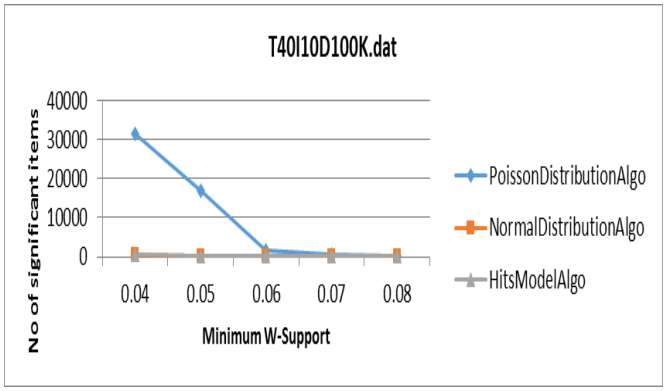

Fig. 9. Number of significant Items for Poisson distribution, Normal distribution and HitsModel Algo for T40I10D100K.dat

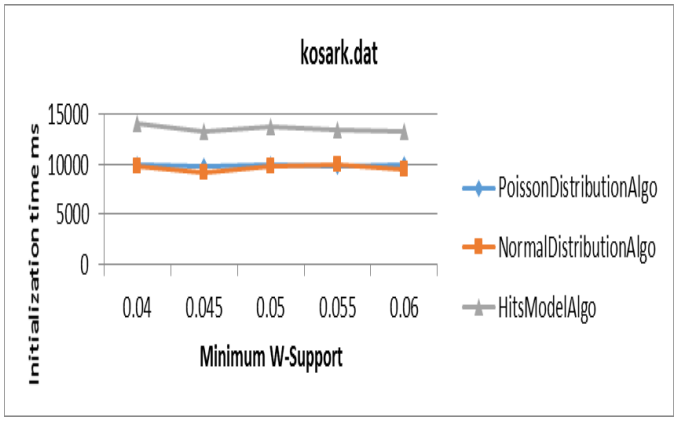

Fig. 10. Initialization time for Poisson distribution, Normal distribution and HitsModel Algo using koasrk.dat

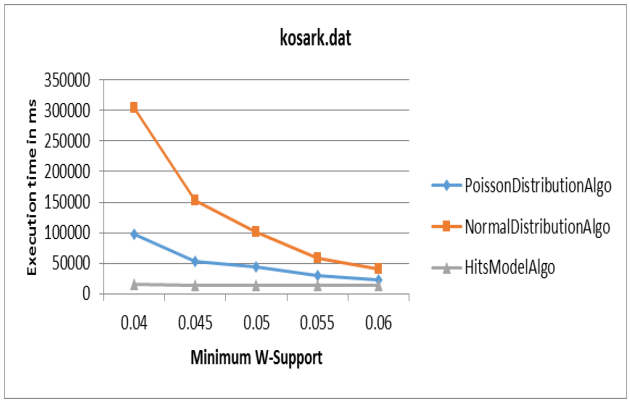

Fig. 11. Execution time for Poisson distribution, Normal distribution, and HitsModel Algo using koasrk.dat

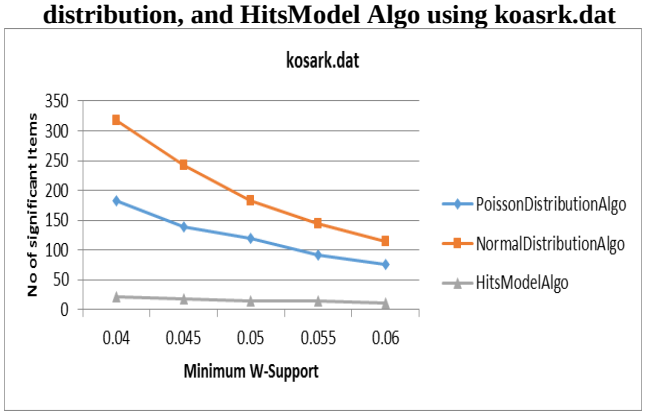

Fig. 12. Significant Items generated for Poisson distribution, Normal distribution and HitsModel Algo using koasrk.dat

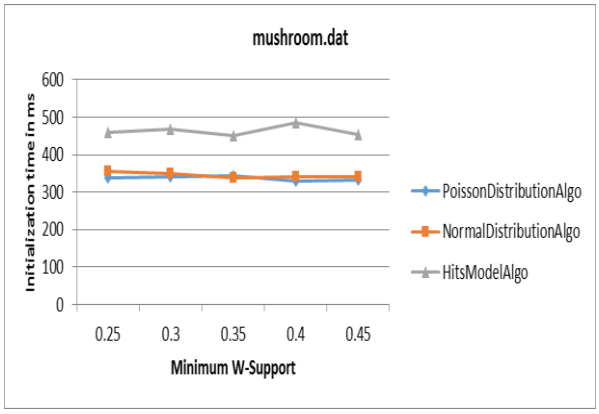

Fig. 13. Initialization time for Poisson distribution, Normal distribution, and HitsModel Algo using mushroom.dat

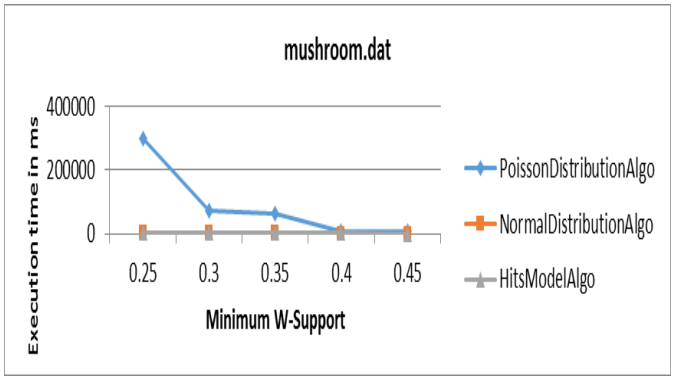

Fig. 14. Execution time for Poisson distribution, Normal distribution, and HitsModel Algo using mushroom.dat

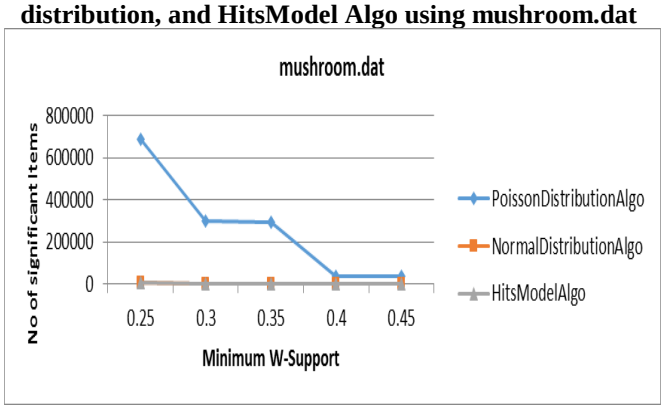

Fig. 15. Significant Items generated for Poisson distribution, Normal distribution and HitsModel Algo using mushroom.dat

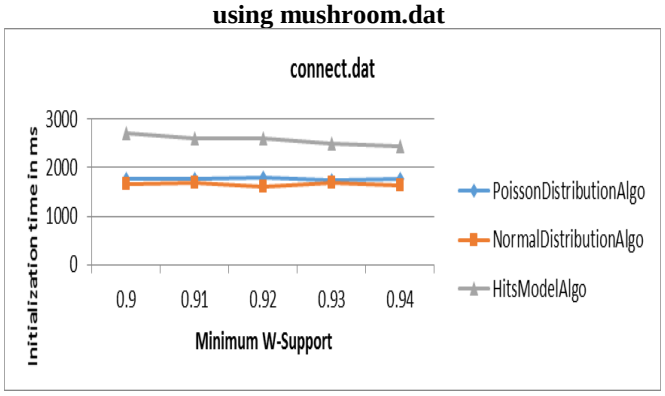

Fig. 16. Initialization time for Poisson distribution, Normal distribution and HitsModel Algo using connect.dat

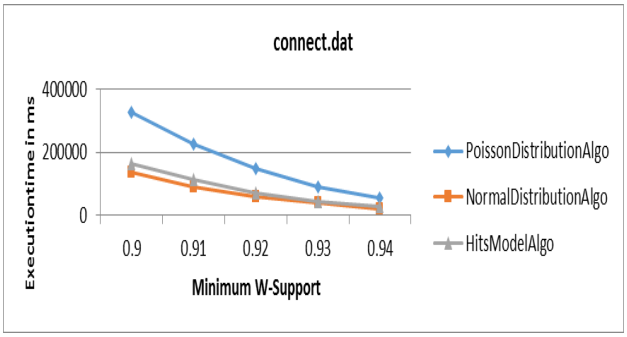

Fig. 17. Execution time for Poisson distribution, Normal distribution, and HitsModel Algo using connect.dat

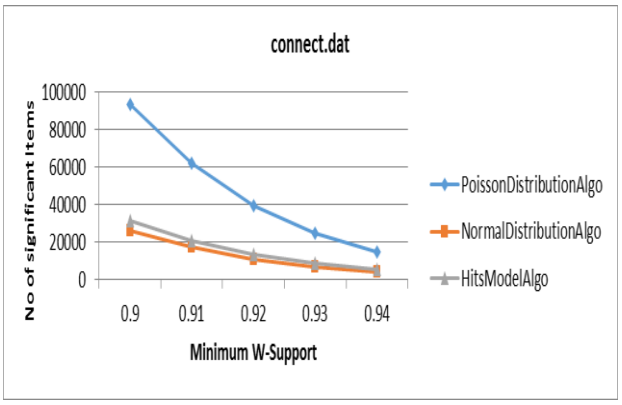

Fig. 18. Significant Items generated for Poisson distribution, Normal distribution and HitsModel Algo using connect.dat

Similar observations for other datasets for kosark.dat (Fig. 10, Fig. 11, Fig. 12), mushroom.dat (Fig. 13, Fig. 14, Fig. 15) and connect.dat (Fig. 16, Fig. 17, Fig.18) has been observed in terms of initialization time (Normalization time), execution time and number of significant items generated.

Table 6 outlines the initialization time taken by Poission distribution, Normal distribution and Hits Model Algo using various datasets.

Table-VI: Initialization time for Poission distribution, Normal Distribution and Hits Model Algo using various datasets

Data Set	Min. Supp	Initialization Time(in ms) using Poission Distribution	Initialization Time(in ms) using Normal Distribution	Initialization Time(in ms) using HitsModel Algo
Chess.dat	0.6	296	298	322
	0.65	299	271	330
	0.7	295	290	315
	0.75	293	280	320
	0.8	295	262	315
Retail.dat	0.004	1291	1300	1801
	0.0045	1277	1366	1906
	0.0047	1250	1325	1970
	0.0049	1240	1247	1852
	0.005	1280	1283	1868
T10I4D100K.dat	0.005	1211	1155	1543
	0.0075	1241	1186	1500
	0.01	1224	1280	1556
	0.015	1255	1191	1573
	0.02	1240	1243	1512
kosark.dat	0.04	9950	9771	14070
	0.045	9820	9197	13389
	0.05	9998	9890	13770
	0.055	9820	10000	13421
	0.06	10000	9461	13391
Mushroom.dat	0.25	339	355	460
	0.3	342	350	467
	035	345	338	451
	0.4	330	340	485
	0.45	331	342	454
connect.dat	0.9	1772	1664	2722
	0.91	1763	1683	2600
	0.92	1790	1614	2610
	0.93	1754	1688	2500
	0.94	1773	1642	2450

Table 7 defines the execution time taken by Poission distribution, Normal distribution and Hits Model Algo using various datasets.

Table-VII: Execution time for Poission distribution, Normal Distribution and Hits Model Algo using various datasets

Data Set	Min. Supp	Execution Time(in ms) using Poission Distribution	Execution Time(in ms) using Normal Distribution	Execution Time(in ms) using HitsModel Algo
Chess.dat	0.6	360706	69860	87163
	0.65	79172	25279	33127
	0.7	23109	11879	13258
	0.75	8169	5377	5884
	0.8	3468	2359	2813
Retail.dat	0.004	408254	429199	70218
	0.0045	359858	312494	49857
	0.0047	299877	269987	42120
	0.0049	258277	239730	38084
	0.005	238263	229371	30724
T10I4D100K.dat	0.005	116375	203263	152380
	0.0075	69192	131427	94376
	0.01	47027	82320	61637
	0.015	27388	41981	28993
	0.02	15809	17744	15802
kosark.dat	0.04	98001	303967	15031
	0.045	53659	151901	13980
	0.05	43804	100508	14402
	0.055	29075	58648	13855
	0.06	23184	39959	13726
Mushroom.dat	0.25	300398	1684	2781
	0.3	71812	1095	1512
	035	63355	718	979
	0.4	7407	551	751
	0.45	7035	502	608
connect.dat	0.9	329114	136289	162136
	0.91	227360	91547	114002
	0.92	149713	58697	69788
	0.93	90288	37886	44005
	0.94	55796	21228	28347

Table 8 defines the number of significant items generated by Poission distribution, Normal distribution and Hits Model Algo using various datasets.

Table-VIII: Significant items generated for Poission distribution, Normal Distribution and Hits Model Algo using various datasets

Data Set	Min. Supp	Number of Significant items using Poission Distribution	Number of Significant items using Normal Distribution	Number of Significant items using HitsModel Algo
Chess.dat	0.6	716083	258100	306238
	0.65	308475	113017	134393
	0.7	124123	49515	58178
	0.75	46427	21250	24972
	0.8	15315	8345	9776
Retail.dat	0.004	3607	3661	2384
	0.0045	2290	2353	1628
	0.0047	2220	2303	1593
	0.0049	2168	2256	1565
	0.005	2141	2238	1547
T10I4D100K.dat	0.005	3751	3208	3997
	0.0075	2230	707	827
	0.01	1759	477	497
	0.015	1272	305	275
	0.02	961	193	182
kosark.dat	0.04	182	318	21
	0.045	139	243	18
	0.05	119	183	15
	0.055	92	145	14
	0.06	76	115	12
Mushroom.dat	0.25	688255	4585	9595
	0.3	297983	2001	3617
	035	295935	901	1639
	0.4	37951	477	757
	0.45	36863	279	359
connect.dat	0.9	93887	25899	31187
	0.91	62047	17219	21027
	0.92	39647	10919	13303
	0.93	24643	6895	8363
	0.94	14783	4093	5103

The results show that our proposed algorithm requires less initialization time to compute weighted support and generates more significant items.

VII. CONCLUSION

The proposed algorithm provides quality transactions where the weights of the itemsets are not provided. It provides an optimized and efficient algorithm using the probability density function and probability mass function. The results obtained by executing different results sets clearly show that our proposed algorithm generates more significant items with reduced normalization time. It uses the framework provided by the HITS model and extending the mathematically defined probability functions (Poisson and Normal distribution). Hence it is a novel algorithm to determine the important transactions in the database by considering the presence of important items without assigning the preassigned weights. Also computing the weight through our proposed algorithm instead of the assigned weights to the item ensures reduction of complexity and magnitude of the database.

REFERENCES

1. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" (2009)
2. R. Agrawal, and J. C. Shafer, "Parallel Mining of Association Rules", IEEE Transactions on Knowledge and Data Engineering, vol. 8, no. 6, pp. 962-969 (1996)
3. Bernecker T, Cheng R, Kriegel H-P, Renz M, Verhein F, Züfle A, Cheung D W, Lee S D, Wang Liang, "Model-based probabilistic frequent itemset mining", Knowledge information system 37 pp. 181-217, Springer (2013)
4. J. Han, and M. Kamber, "Data Mining: Concepts and Techniques," Morgan Kaufmann Publishers (2016)
5. B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," Proc. ACM SIGKDD '98, pp. 80-86, 1998.
6. H. Cherfi, A. Napoli, and Y. Toussaint, "Towards a Text Mining Methodology Using Frequent Itemsets and Association Rule Extraction," Proc. Fourth Int'l Conf. Journe´es de l'Informatique Messine (JIM '03) on Knowledge Discovery and Discrete Math., pp. 285-294, 2003.
7. S. Madria, S. Bhowmick, W. Ng, and E. Lim, "Research Issues in Web Data Mining," Proc. First Int'l Conf. DataWarehousing and Knowledge Discovery (DaWaK '99), pp. 303-312, 1999.
8. J. Li, B. Tang, and N. Cercone, "Applying Association Rules for Interesting Recommendations Using Rule Templates," Proc. Eighth Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD'04), pp. 166-170, 2004.
9. R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Datasets," Proc. ACM SIGMOD '93, pp. 207-216, 1993.
10. G.D. Ramkumar, S. Ranka, and S. Tsur, "Weighted Association Rules: Model and Algorithm," Proc. ACM SIGKDD, 1998.
11. C.H. Cai, A.W.C. Fu, C.H. Cheng, and W.W. Kwong, "Mining Association Rules with Weighted Items," Proc. IEEE Int'l Database Eng. and Applications Symp. (IDEAS '98), pp. 68-77, 1998.
12. W. Wang, J. Yang, and P.S. Yu, "Efficient Mining of Weighted Association Rules (WAR)," Proc. ACM SIGKDD '00, pp. 270-274, 2000.
13. F. Tao, F. Murtagh, and M. Farid, "Weighted Association Rule Mining Using Weighted Support and Significance Framework," Proc. ACM SIGKDD '03, pp. 661-"666, 2003.
14. K. Sun and F. Bai, "Mining Weighted Association Rules without Preassigned Weights," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 4, pp. 489-495, Apr. 2008.
15. J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," J. ACM, vol. 46, no. 5, pp. 604-632, 1999.
16. Bernecker, T., Cheng, R., Cheung, D.W. et al. "Model-based probabilistic frequent itemset mining" Knowl Inf Syst (2013) 37
17. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94), pp. 487-499, 1994.
18. J.S. Park, M. Chen, and P.S. Yu, "An Effective Hash Based Algorithm for Mining Association Rules," Proc. ACM SIGMOD,1995.
19. Han, J., Pei, J., Yin, Y., Mao, R. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Min. Knowl. Discov. 8, 53–87., 2004.
20. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA 2000
21. Scalable Algorithms for Association Mining. IEEE Trans. Knowl. Data Eng. 12(3): 372-390 (2000)
22. M. Zaki, and K. Gouda. Fast vertical mining using diffsets. ACM KDD Conference, 2003.
23. M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New Algorithms for Fast Discovery of Association Rules. KDD Conference, pp. 283–286, 1997.
24. K. Wang and M.-Y. Su, "Item Selection by "Hub-Authority" Profit Ranking," Proc. ACM SIGKDD, 2002.
25. Samar Wazir, M. M. Sufyan Beg & Tanvir Ahmad. Comprehensive mining of frequent itemsets for a combination of certain and uncertain databases. International Journal of Information Technology (2019)
26. Le Cam L (1960) An approximation theorem for the Poisson binomial distribution. Pac J Math 10:1181–1197
27. Hodges JL, Cam Le (1959) The poisson approximation to the poisson binomial distribution. Ann Math Stat Inst Math Stat Probab Lett 31:737–740. https://doi.org/10.1016/0167-7152(91)90170-v
28. Feller W (1945) The fundamental limit theorems in probability. Bull Am Math Soc 51:800–832. https://doi.org/10.1090/S0002-9904-1945-08448-1
29. Feller W (1968) An introduction to probability theory and its applications, vol I. xviii ? 509. Wiley, Amsterdam
30. Frequent itemset mining Dataset repository http://fimi.uantwerpen.be/data/

31. Prof. Fournier-Viger. SPMF: A Java Open-Source Data Mining Library. http://www.philippe-fournier-viger.com/spmf/index.php?link=download.php. Accessed 31 Jan 2020
32. Apache Commons Math 3.6.1 API https://commons.apache.org/proper/commons-math/download_math.cgi. Accessed 31 Jan 2020
33. Fournier-Viger SPMF (2018) A Java open-source data mining library. http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php. Accessed 20 Nov 2019
34. F. Bodon, "A Survey on Frequent Itemset Mining," technical report, Budapest Univ. of Technology and Economics, 2006
35. The IBM Synthetic Data Generator, http://www.almaden.ibm.com/software/projects/iis/hdb/Projects/data_mining/datasets/syndata.html, 2007.
36. R.J. Bayardo Jr. and R. Agrawal, "Mining the Most Interesting Rules," Proc. ACM SIGKDD '99, pp. 145-154, 1999.
37. T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets, "The Use of Association Rules for Product Assortment Decisions: A Case Study," Proc. ACM SIGKDD '99, pp. 254-260, 1999.