AN EFFICIENT FUZZY WEIGHTED ASSOCIATION RULE MINING WITH ENHANCED HITS ALGORITHM

V. Vidya
Research and Development Centre, Bharathiar University, Coimbatore, Tamil Nadu, India
E-Mail: pondymiraalfssa@gmail.com

ABSTRACT
Association rule mainly focuses on large transactional databases. In association rule mining all items are considered with equal weightage. But it is not suitable for all datasets. The weight should be considered based on the importance of the item. In our previous work HITS algorithm (Hyperlink Induced Topic Search) is used to find the weight of an item w-support is calculated for generating frequent item sets. In this work enhanced HITS is used to calculate the weightage of the items. The enhanced HITS update the weight value in online manner. Fuzzy logic approach is applied to improve the association rule mining. So the proposed fuzzy weighted association rule mining with enhanced HITS satisfies downward closure property which decreases computation time; uninteresting rules can be pruned because of assigning weights to items, which also reduce the execution time. This paper introduces enhanced HITS algorithm and compute weights to describe the importance of attribute with respect to users intuition and integrate the options into mining weighted fuzzy association rule algorithm. Most weighted association rule mining eliminates extra steps during rules generation. Experiments show that the proposed algorithm is capable of discovering new rules in an effective manner by obtaining high confidence results. The comparison between fuzzy weighted association rule mining with enhanced HITS and weighted association rule mining with enhanced HITS is experimentally evaluated with food mart dataset as shown in enhanced version outperforms the weighted association rule mining with enhanced HITS.

Keywords: association rule mining, enhanced HITS algorithm, fuzzy weighted association rule mining, weighted association rule mining.

1. INTRODUCTION
Data mining is an emerging technique to address the problem of reconstructing data into useful knowledge of information from the user who can mine the results which they really need. The rules are generated according to the knowledge by data mining algorithms in which one of most problematic steps in association rule is to discover process of knowledge validation. To solve this problem association rules have been widely used in many applications domains for finding pattern in data and to generate the association rules. The pattern reveals the combinations of events that occur at the same time based on the interesting associations and/or correlate the relationships among large set of data items. The attributes value conditions shown in association rule that occur frequently in a given dataset.

Apart from the antecedent (the "if" part) and the consequent (the "then" part), an association rule has two numbers that express the degree of uncertainty about the rule. First, Support is simply the number of transactions that include all items in the antecedent and consequent parts of the rule. Second, Confidence is the ratio of the number of transactions that include all items in the consequent as well as the antecedent to the number of transactions that include all items in the antecedent. Lift: Lift is nothing but the ratio of confidence to expected confidence. Lift is a value that gives us information about the increase in probability of the "then" (consequent) given "if" (antecedent) part.

In Association Rule Mining (ARM) model assumes all items that have the same significance without taking their weight into account which also ignores the difference between the transactions and importance of each and every item sets. Whereas Weighted Association Rule Mining (WARM) work only based on binary attributes and not on databases which make use of the importance of each item set and transaction. WARM needs each item to be given with weight that may correspond to special promotions or profitability to reflect their importance to the user. This research work is based on a weight assignment based on directed graph where nodes denote items and links to represent association rules, and a generalized version of HITS is used to rank items in a direct graph, whereas all nodes and links are allowed to have weights. This research uses enhanced HITS algorithm by developing an online eigenvector calculation method that can compute the results of mutual reinforcement voting in case of frequent updates of the nodes.

1.1. Analysis and practical examples
Let Given a set of items I= {i1,i2,i3,…,im} and a database of transactions DB= {t1,t2…tn} where 

ti={Ii1,Ii2,…,Iip}, if A ⊆ I with K = |A| is called a k-itemset or simply an itemset. Let a database D be a multi-
set of subsets of I as shown. Each $T \in DB$ supports an item set $A \subseteq I$ if $A \subseteq T$ that holds. An association rule is an expression $A \Rightarrow B$, where $A$, $B$ are item sets and $A \cap B = \emptyset$ holds. Number of transactions $T$ supporting an item $A$ w.r.t DB is called support of $A$, $\text{Supp}(A) = |T \cap DB|$. The strength or confidence ($c$) for an association rule $A \Rightarrow B$ is the ratio of the number of transactions that contain $A \cup B$ to the number of transactions that contain $A$, $\text{Conf}(A \Rightarrow B) = \frac{\text{Supp}(A \cup B)}{\text{Supp}(A)}$.

For non-Boolean items fuzzy association rule mining was proposed by using fuzzy sets such that quantitative and categorical attributes can be handled. A fuzzy quantitative rule represents each item as (item, value) a pair. Fuzzy association rules are articulated in the subsequent form: if $A$ is $X$ which satisfies $B$ is $Y$. For instance, if $(\text{age is young}) \Rightarrow (\text{salary is low})$ were given a database $T$, attributes $I$ with item sets $A \subseteq I$, $B \subseteq I$ and $A = \{a_1, a_2, a_3 \ldots a_n\}$ in addition to $B = \{b_1, b_2, b_3 \ldots b_n\}$ and $A \cap B = \emptyset$. Fuzzy sets can be described as $X = \{f_{x_1}, f_{x_2}, \ldots, f_{x_i}\}$ and $Y = \{f_{x_1}, f_{x_2}, \ldots, f_{x_i}\}$ associated to $A$ and $B$ respectively. For example; consider $(A, B)$ possibly will be $(\text{age, young})$, $(\text{age, old})$, $(\text{salary, high})$ etc. The semantics of the rule is that when the antecedent "$A$ is $X$" is satisfied, which means that "$B$ is $Y$" also fulfilled, their votes to the attribute fuzzy set pairs and the sum of these votes is larger than the user specified threshold which shows the means of sufficient records that fix., the above ARM framework that assumes all items have the same weight contained by a transaction or record is the same (weight=1) which is not constant.

### Table-1. Weighted items database.

<table>
<thead>
<tr>
<th>ID</th>
<th>Item</th>
<th>Profit</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Butter</td>
<td>25</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>jam</td>
<td>45</td>
<td>0.45</td>
</tr>
<tr>
<td>3</td>
<td>Bread</td>
<td>75</td>
<td>0.75</td>
</tr>
</tbody>
</table>

The conception of association rule mining offers the support-confidence framework and condensed association rule mining to invent frequent item sets. The WARM generalizes the traditional model to generate the frequent item sets where items have weight that match to the effectiveness of diverse items. Since more data is gathered, which are frequently updated and the construction of the graph should be dynamic instead of static. The grid can be marked dynamically and the cost can be condensed by using enhanced HITS algorithm, during postponing informs whenever possible. By enforcing the association among the items by calculating eigen values. The existing HITS algorithm is suitable only for static content. 1) There is no lively updating of weights to the items is feasible. 2) They also fail to capture the rich information in the structure that can be defined by user group implicitly.

Weighted ARM deals with the significance of individual items in a database [1, 2, 3]. For example, some products are additional gainful or not to be, so the rules in relation to them are of greater value consequently and more interesting as compared to others is to be completed. Items are allocated with weights ($w$) dealing to their importance as shown in Table-1. These weights may be situated according to an item’s profit boundary. This generalized description of ARM is called Weighted Association Rule Mining (WARM). As Table-1, from the rule Bread→ Jam is more attention-grabbing than Bread→ Butter for the reason that the profit of a Jam is greater than that of Butter. The main face up in weighted ARM is that "downward closure property" which is to well-organized iterative procedure for generating and shortening frequent item sets from subsets.

In this proposed system weights of HITS algorithm are stored and calculate the weight in enhanced HITS algorithm with WARM, which decreases the cost by postponing the updates whenever possible and makes it more suitable for dynamic environments. HITS algorithm can be normally used for web pages, but it can also be used for transactional datasets from which the hub and authorities are calculated, based on the graph constructed but HITS is not suitable for dynamic environment. HITS algorithm is used as outlay to run each update. When the updates are collected by running enhance HITS the way of cost is concentrated high. An additional advantage of enhance HITS is to inspect the user queries, that update
the Eigen vector with specified Matrix $A$. By comparing the investigational results with the enhanced HITS and fuzzy weighted association rule mining with the enhanced HITS and weighted association rule mining.

2. PREVIOUS WORK

Previously, the most two well-known link investigation algorithms were PageRank algorithm planned [3] by Sergey Brin and Lawrence Page and the HITS algorithm [4] by J.M Kleinberg. They do not consider the content of the page which are entirely link-based algorithms. Therefore, the outcomes of the algorithms contain often some non-relevant pages with tightly interconnected density. In order to avoid the topic drift, page ranking algorithms sanded on hyperlinks and content had been planned, such as the ARC (Automatic Resource Compilation) algorithm [5] and the Average algorithm [6]. HITS algorithm cannot treat relations differently, but its definition makes the qualities of hubs are determined by the quantity of authority page.

Mining Association rule is extremely a vital field to investigate in data mining. The problem of mining Association rule is put forward by R.S Agarwal first in 1993. Currently they survive many mining methods for finding the frequent item set such as Apriori algorithm, Frequent Pattern-Tree algorithm etc. Apriori algorithm’s weakness is to produce lots of candidate item sets and scans database every time. It will be too costly to generate the frequent item set applicants. If the database contains enormous number of transactions then scanning the database for finding the frequent item set will take long time to process.

Later, Allan Borodin and others proposed the Authority-Threshold algorithm [2] which sets the hub weight of node $n$ to be the sum of the $j$ major authority weights of the authorities pointed by node $n$. This communicates to say that a node is a high-quality focal point if it spots at least $j$ fine authorities. Moreover, Lempel and Moran proposed the SALSA algorithm [7] supported on the Markov chain. Cohn and Chang proposed the PHITS algorithm [8] based on the probability model, and so on.

FP-Tree algorithm does not generate any candidate items but it checks database two times in the memory allowed. But there is a need of high memory requirement if not it becomes a complex process. As the database scans more than two times its I/O operating cost will increase. That is why there is a requirement in designing an efficient algorithm which informs, defends and controls the association rule in large transactional database. Many examiners made analysis and research to intend how efficiently to renew the association rules and place forward parallel algorithm. The two most cases in the association rule updation.

The first case is when the database is altered and by discovering frequent item sets. FUFIA algorithm is the emblematic updating method for this problem in earlier phase. The second case is when the lowest hold up is altered then to find frequent items sets. IUA algorithm is the figurative updating method for this problem. These updating algorithms have both advantages and disadvantages. This manuscript suggests a dynamic algorithm of frequent mining supported by undirected item set Table which scans the database merely once and then put aside the information of original database in undirected item set graph and finds the frequent item sets directly from the graph. It does not produce any candidate items. When database and minimum support is distorted, the algorithm rescans the undirected item set graph to obtain the new frequent item sets [1].

3. RELATED WORK

Typical ARM data items are analyzed by having equivalent significance but newly several approaches generalize the items which are given weights to reflect their significance to the user [6]. The weights may correspond to particular supports on some artifacts or the productivity of different items etc. Currently, two approaches exist: pre- and post-processing. Post processing solves the non-weighted problem (weights=1 per item) and then prunes the rules later. Pre-processing prunes the non-frequent item sets were using weights after the every iteration previously. The issue post-processed weighted ARM is scans the items without bearing in mind their weights. Finally, the rule support is checked for frequent weighted ARs. This gives us a very incomplete item set pool to check weighted ARs and may miss a lot of prospective item sets.

Pre-processing, traditional ARM reduces item sets by checking frequent item once and weighted support subsequent to every scan. In pre-processing, smaller amount rules are attained as compared to post processing because many potential frequent super sets are missed. In [4] post-processing model, two algorithms were suggested to mine item sets with regularized and un-normalized burdens. The K-support apparent metric was used to make certain validity of the downward closure property. It is not guarantee for every subset of a frequent set being frequent unless the k-support bound value of (K-1) subset was higher than (K). Efficient mining method for Weighted Association Rules (WAR) is proposed in [6].

An algebraic attribute is allocated for each item where the weight of item is defined as part of a particular weight domain. WAR uses a post-processing approach by deriving the maximum weighted rules from frequent item sets. With the process of generating frequent item sets Post WAR doesn’t interfere it focus on how weighted association rules can be produced by grouping the weighting issues of the items incorporated in generating frequent item sets. Comparable techniques for weighted fuzzy quantitative association rule mining [2, 8]. In [7], a dual pre processing approach is used where quantitative attributes are discretized into different fuzzy linguistic periods and weights allocated to each linguistic label.

A mining algorithm is applying two support measures for normalized and un-normalized cases then it give rise to dataset. The closure property deals with z-potential frequent subset for each candidate set. An
arithmetic mean is used to find the chance of frequent k+1 itemset, which is not guaranteed to authenticate the valid downward closure property, that handles the DCP problem, another impact framework, were WARM is proposed. Weighting spaces were introduced as inner-transaction space, item space and transaction space, in which items can be weighted depending on different scenarios and mining focus. However, support is calculated only by considering the transactions that contribute to the item set.

Additionally, no negotiations were prepared on the interestingness issue of the rules created. In this paper, fuzzy weighted support and confidence framework to mine weighted boolean and quantitative data (by fuzzy means) is to address the issue of invalidation of downward closure property. In the proposed framework, rules can be generated professionally with a valid downward closure property with no favoritisms made by pre- or post-processing approaches. Most of the present work on the customary Apriori algorithm [4] make use of the “large - support” metric framework. However these works still view the items having equal weights and try to distinguish them using various methods. Wei Wang et al. proposed an efficient mining methodology for Weighted Association Rules (WAR) [9].

Han et al. projected a solution where a concept hierarchy order is used and association rules were classified into manifold conceptual levels of granularity. This idea motivates the work [1] where the existing association rule model is extended that allows users to specify multiple threshold supports. In the extended model, the threshold support is expressed in terms of Minimum Item Supports (MIS) to the items that appear in the rule. The main feature of this method is that the user can identify a different threshold item support for each item, similar to the scenario of transmission weights to items. This technique can find out rare item rules without causing frequent items that generate many unnecessary rules. Liu’s model also breaks the “downward closure property”. The problem is solved by using a “sorted closure property” where the items in the item break are typed in ascending order of their MIS values.

4. ENHANCED HITS ALGORITHM

HITS algorithm calculates weight values statically but enhanced HITS algorithm updates the changes of values dynamically in order to reduce the execution time. In the proposed work matrix $\mathbf{M}$, this is based on the transactions and the items contained by each of the transactions having the values of the regulation that are to be considered for mining. The positions $\mathbf{x}$ and $\mathbf{y}$, correspond to the major eigenvectors of the matrix $\mathbf{M}^T\mathbf{M}$ and $\mathbf{MM}^T$, respectively. Still a single fill in to the user access will communicate to a perturbation of the matrix $\mathbf{M}$. Depending on the weight function selected it can change the behavior of a single element or a row of elements of the matrix $\mathbf{M}$. The alterations in behavior may cause variations to the primary eigenvector of $\mathbf{M}^T\mathbf{M}$ (and $\mathbf{MM}^T$). By checking each update whether it causes too many changes to $\mathbf{x}$ and $\mathbf{y}$ and finds the relationship between the variation of $\mathbf{x}$ and $\mathbf{y}$ and the behavioral changes to the matrix $\mathbf{M}$, updation can be applied, if the change is within acceptable precision, it avoid running the enhance HITS through which the charge of running algorithm repeatedly can be reduced.

An additional advantage of this approach is to facilitate examine the user queries and updating the matrix $\mathbf{M}$ and organization of enhance HITS may break up. The system can revise the matrix $\mathbf{M}$ and run Enhance HITS in background, and carry on servicing user queries with previous results that are convinced to be within positive range from the newest ones were users can benefit from the service without any troubles. To experience theoretical change it is too excellent for the results, and techniques which reflect modernize of the latest trends. In addition, the conclusions are considered only to apply unweighted graphs signified by adjacency matrices. Enhanced HITS algorithm utilizes two focal concepts, that exact the computations of eigen value and perturbation. They have to be completely proficient or else the cost of computing will influence the saving by not running HITS. They will be addressed in the following subsections.

4.1. Computation of Eigen value

Eigen value is the variation between the largest and second largest Eigen values specifically $\lambda_1$ and $\lambda_2$. The unique HITS algorithm is fundamentally a power method to calculate the principal eigenvector of S. It can be enhanced easily without count complexity to produce $\lambda_1$ and $\lambda_2$ as byproducts. Two alterations to the original HITS algorithm are launched:

a) Find the two eigenvectors $\lambda_1$ and $\lambda_2$, as an alternative of finding only the prime eigenvector. Initially, start with two orthogonal vectors, multiply all by S and apply Gram- Schmidt to orthogonalize them. This can be finished by using the block power method. This is a single step. Iterate in anticipation of the congregate. In the HITS algorithm put back this step with the step that calculates principle Eigen vector.

b) HITS ensure convergence by normalizing the vector at each step to unit length. By separating them, their first non-zero element instead, normalizing each vector. They still converge the two eigenvectors and the scaling factors converge to $\lambda_1$ and $\lambda_2$.

4.2. Estimation and analysis of enhance hits in dynamic environments

The Enhance HITS algorithm is the process of constructing an access based graph which is based on the single update of a user or every change of the item is taken corresponding to a perturbation to a matrix $\mathbf{M}$. The weight meeting chosen can perturbation in any of these cases which can be always limited a single element of the matrix M or row of elements in the matrix $\mathbf{M}$. This will
have an effect on the Eigen vector of the matrix $M \cdot M$ and $M \cdot M^T$. If we can find the association between the changes of the ranking to principle Eigen vector of $M \cdot M$ and $M \cdot M^T$ which are called as $\lambda$ and $\gamma$ values, if there is too many changes to $\lambda$ and $\gamma$, check each update. Validating that level of correctness, by applying the update informed thus avoid running Online HITS. The Online HITS takes into account the current variations that happen dynamically, and the algorithm reflects a true dramatically changed, updated system knowledge of the current world. Online HITS continuously checks the changes and makes operation.

4.3. pseudo code for implementing enhanced hits algorithm

**STEP 1:** Apply two priority queues sorted in decreasing of their weight values.

**STEP 2:** Initialize, the priority queues with the seed sites (with each one of them allocated a weight of 1).
STEP 3: Consider the top N items from the hub priority queue. During different iterations this item can be occurred already. Once the item added into the queues are never removed, their weights get updated accordingly (as mentioned in Steps 4 and 5).

STEP 4: The weights of the items (retrieved from the hub queue) are evenly dispersed to their outgoing relations in the authority queue. The weights of the items are then located to 0 in the hub queue. The items (present in the set of weight intended items) which are not present in both the priority queues are scored as original. These innovative items are added to both the queues. The weights of all these newly inserted items are set to 0 in both the priority queues.

STEP 5: The item by maximum weight from the authority queue is selected. This weight is equally distributed to its incoming links in the hub queue. The weights of all the chosen items from the authority queue are then set 0.

STEP 6: The hub and the authority priority queues are normalized.

STEP 7: Repeat steps 3 to 6 until finished by the user

STEP 8: The values present in the authority and hub queues after step 7 represent the influence and hub ranks individually.

4.4. Ranking transaction with hits

Let consider DB= \{T_1, T_2 ....T_m\} be a list of transactions and I= \{i_1, i_2 ....i_n\} be the matching set of items DB is equivalent to the bipartite graph. By considering the transaction of pure hubs and pure authorities the correlation between transaction and item is just like the relationship between hubs and authorities which shows the statement or expression are equal through iteration process such as auth (i) = \sum_{T:i \in T} hub (T), hub (T) = \sum_{i:i \in T} auth (i)

Table-3. Bipartite representation of a database.

<table>
<thead>
<tr>
<th>TID</th>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>{A,B,C,D,E,F}</td>
</tr>
<tr>
<td>200</td>
<td>{A,C,F,G}</td>
</tr>
<tr>
<td>300</td>
<td>{A,B,C,I}</td>
</tr>
<tr>
<td>400</td>
<td>{A,B,D,E}</td>
</tr>
<tr>
<td>500</td>
<td>{C,F,G,H,I}</td>
</tr>
<tr>
<td>600</td>
<td>{A,G,F,H,I}</td>
</tr>
</tbody>
</table>

Figure-2. Bipartite graph.

When HITS model is applied all contracts are obtained, several weights represent high-value item, transaction with a small number of items have good hub. If all sub items are top-ranked transactions with more common item that have low hub weight.

4.5. Weighted association rule mining

The objective of using weighted support is to construct the use of weight in the mining process and prioritize the collection of object item sets dealing to their implication in the dataset, than their frequency alone. Items can be weighted inside different weighting breaks depending on different situations and mining focus.

Weighting space WS is the context within which the weights are appraised:

a) **Internal-transaction space WST:** This space submits to the host transaction that an item is weighted in.

b) **Item space WSI:** This space submits to the space of the item collection that covers all the items appears in the transactions.

c) **Transaction space WST:** This space is defined for transactions rather than for items.

An item set is indicated large it carries above a predefined minimum support threshold. In the WARM context, when an item set is significant its weighted support is over a pre-defined minimum weighted support threshold. In fact, the threshold values precised by the user are from the margin of significance of charge point of view. This method may be more significant than identifying comparatively random support threshold. For illustration, in the superstore situation, allocating weight to each of the items according to their earnings it produces to the store, rather than simply counting and calculating the percentage of transactions that contain item set.
According to the weighted support WSP of an item set. A set of transactions T respects a rule R in the form $A \rightarrow B$ where $A$ and $B$ are non-empty sub-item sets of the item space $I$ and they share no item in common. Its weighted support is the fraction of weight of the transactions that contains both $A$ and $B$ relative to the weight of all transactions. The popular methods in judging frequent item sets are proposed for usual databases. On the other hand, the occurrence of an item set may not be satisfactory pointer of significance, because frequency reproduces only the number of transactions in the database that enclose to item set. This can be quantified in terms of cost, profit, or other expressions of user preference that does not reveal the utility of an item set. Conversely, only frequent item sets possibly add a small portion of the overall profit, while non-frequent item sets contain a large part of the profit. In this the weight support are referred as utility weight support (UWS). This can be formulated as:

$$UWS(AB) = \frac{\sum_{T_i \in DB \times X} T_i \cdot uw(A \rightarrow T_i)}{\sum_{T_i \in DB \times X} T_i \cdot uw(A \rightarrow I)}$$  \hspace{1cm} (1)$$

Thus, utility based weighted support is formed to measure the actual allocation of an item set in the transaction space in weighted association rule mining development. That approximation utility weighted support of an item set; a technique is used to evaluate transaction weight. In which the amount of item set referred as $T_i$. The transaction utility based on weight ($t_{uw}$) can be derivative from utility weights of the items close to the transaction. Individual may make it easily as follows:

$$uw(A) = \sum_{T_i \in DB \times X} T_i \cdot t_{uw}(A \rightarrow T_i)$$  \hspace{1cm} (2)$$

This value is used to work out the weighted support of a potentially considerable item set illustrated above. If its weighted support is above the pre-defined minimum weighted support the item set is then corroborated as significant.

### Table-4. Hubs and W-support for database.

<table>
<thead>
<tr>
<th>TID</th>
<th>Transaction</th>
<th>Hub Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>${A,B,C,D,E,F}$</td>
<td>0.618</td>
</tr>
<tr>
<td>200</td>
<td>${A,C,F,G}$</td>
<td>0.546</td>
</tr>
<tr>
<td>300</td>
<td>${A,B,C,I}$</td>
<td>0.343</td>
</tr>
<tr>
<td>400</td>
<td>${A,B,D,E}$</td>
<td>0.258</td>
</tr>
<tr>
<td>500</td>
<td>${C,F,G,H,I}$</td>
<td>0.645</td>
</tr>
<tr>
<td>600</td>
<td>${A,G,F,H,I}$</td>
<td>0.514</td>
</tr>
</tbody>
</table>

### Table-5. Utility weight table.

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Utility value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>-3</td>
</tr>
<tr>
<td>E</td>
<td>-2</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>3</td>
</tr>
<tr>
<td>H</td>
<td>-4</td>
</tr>
<tr>
<td>I</td>
<td>6</td>
</tr>
</tbody>
</table>

For example:

$$UW(item) = uw(T_1) + uw(T_2) + ... + uw(T_n)$$

$$UW(C) = uw(T_1) + uw(T_2) + uw(T_3) + uw(T_4)$$

$$= (0.618 \times 2) + (0.546 \times 2) + (0.343 \times 2) + (0.645 \times 2)$$

$$= 4.304$$

$$UWS(A) = uw(A \rightarrow I, T_3) + uw(A \rightarrow I, T_6)$$

$$= (0.343 \times 5) + (0.343 \times 6) + (0.514 \times 5) + (0.514 \times 6) = 9.827$$

As may be noted down, if an item set is spotted with specked background, then any of its supersets in the upper layer of the lattice cannot be significant. This property, denoted as “weighted downward closure property”, is applicable less than the “weighted support - significant” framework. It justifies the efficient mechanism of generating and pruning significance iteratively.

initialize auth(i) to 1 for each item i

for $l = 0 \leq t \leq \text{num}$ do begin
  auth(i) = 0 for each item i
  for all transactions $T \in DB$ do begin
    $\sum_{i \in \text{item}} \text{hub6} = \text{hub6} + \text{auth}(i)$ for each item i $\in T$
  end
end
Frequent is the set of frequent item sets, \( m \) is the set of weights associated to items.

The algorithm is given in Table-6. In the table:

- **FWARM** belongs to the breadth first traversal family in ARM algorithms, some adaptations.
- The proposed Fuzzy Weighted Association Rule Mining (FWARM) algorithm belongs to a fashion similar to the Apriori algorithm.

5. FUZZYWEIGHTED ASSOCIATION RULE MINING

Fuzzy Weighted Association Rule Mining (FWARM) algorithm can be used or accepted after some adaptations. The proposed Fuzzy Weighted Association Rule Mining (FWARM) algorithm belongs to the breadth first traversal family in ARM algorithms, expanded using tree data arrangements and works in a fashion similar to the Apriori algorithm. The FWARM algorithm is given in Table-6. In the table:

- **Candidate** is the set of candidate item sets in cardinality; \( iw \) is the set of weights associated to items.
- \( F \) is the set of frequent item sets, **WAR** is the set of potential rules and **WAR** is the final set of generated fuzzy weighted ARs.

An efficient Fuzzy Weighted Association Rule Mining with enhanced HITS algorithm steps:

**STEP 1:** Starting with algorithm having values for hub and authority as 1.

**STEP 2:** Run both hub and authority using update rule.

**STEP 3:** Normalize values by dividing every hub value in sum of squares of all hub values and divide each authority value by sum of squares of all authority values and store in **auth(i)** = \( m \).

**STEP 4:** Repeat step 2 as necessary.

**STEP 5:** After weight calculation searching database searches and returns to the complete set containing all attribute of the database.

**STEP 6:** From step 5 a transformed fuzzy database is created from the original one.

**STEP 7:** User defined items in the original database will be mapped from the weight assigned database that is after generating candidate item sets from the transformed database, it is scanned in order to evaluate support and after comparing the support to the predefined minimum support of the item with a low support is deleted. Whereas the enhanced HITS algorithm for weight calculation.

**STEP 8:** Frequent item sets **Frequentm** will be created from candidate item sets **Cm**.

**STEP 9:** New candidate sets are generated with the weights assigned as \( m \). **Candidatem** from the old ones in a subsequent step.

**STEP 10:** **Candidatem** is generated from \( c'_m \).

**Input:**
- **D** = Data set
- **iw** = Item set weights
- **wsup** = Weighted support
- **wconf** = Weighted confidence
- **Candidatem** = Candidate item set
- **Frequentm** = Frequent item set
- **m** = Item set
- **c'_m** = Number of candidate item set in **Cm**.
- **Fz** = Fuzzy Association rule
- **fs** = Fuzzy item set in fuzzy association rule
- **rs** = Rules generated from **Candidatem**.
- **WAR** = Rules
- **min_wsup** = minimum weighted support
- **min_wconf** = minimum weighted confidence

**Output:**
- **WAR** = **Set of Weighted Association Rules**
- **m** = 0; **Candidatem** = \( \emptyset \); **Frequentm** = \( \emptyset \)
- **Candidatem** = Set of 1 item sets
- **m** = \( m + 1 \)

**Loop**
- if **Candidatem** = \( \emptyset \) break
- \( \forall \ c'_m \) **Candidatem**
- \( c'_m \).weighted support \( \rightarrow \) weighted support count
- if \( c'_m \). weighted support > \( \text{min}_w\sup \)
- \( Fz \rightarrow Fz \cup c'_m \)
- \( m \rightarrow m + 1 \)
- **Candidatem** = generate candidates (**Fz**)

**end loop**
- \( \forall \ fs \in Fz \)
- generate set of candidate rules \( \{ rs_1, \ldots, rs_{n} \} \)
- \( \{ rs_1, \ldots, rs_{n} \} \rightarrow \text{WAR} \rightarrow \text{WAR} \cup rs \)
rs. weighted confidence \rightarrow \text{weighted confidence value}

if rs. weighted confidence > \text{min\_wconf}

\textbf{Output}: \quad \text{\textit{WAR}} = \text{\it{WA}} \quad \text{\it{R}}^\top \quad \text{U} \quad \text{rs}

\textbf{Figure-4}. An algorithm for fuzzy weighted association rule mining.

\section{RESULTS AND DISCUSSIONS}

\subsection{Dataset description}

In regard to the experimental testing database, its source is Food Mart 2000 retail transaction database embedded in a Microsoft SQL Server 2000. It provides information about the sale of the product and how the customers are satisfied with the variety of the product. Since there are different kinds of transaction databases in FoodMart2000, by selecting only sales fact 1997 data table for assessment. The number of product items in this data table is 1560. In order to mine effectively meaningful association rules, this experiment categorizes the products into groups according to the product category provided by the data table. Thus, products are classified into 34 categories, each with a corresponding product category id. In regard to data selection, 6000 customers are randomly selected along with their corresponding transaction data at different times. After this arrangement, there is a total of 12,100 transaction records for these 6000 customers.

\subsection{Number of rules vs minimum weighted confidence}

The performance of the technique is offered by weighted association rule with enhanced HITS and fuzzy weighted association rule with enhanced HITS is compared and based on two parameters are number of rules and weighted confidence. Here if the weighted confidence is increased the number of rules is decreased linearly. But the number of rules generated by proposed system is high when compared with existing system. Based on the comparison the results from the experiment show the proposed approach works better than the other existing systems.

\subsection{Frequent item set vs support}

By analyzing and comparing the performance offered by weighted association rule with enhanced HITS and fuzzy weighted association rule with enhanced HITS. The technique based on two parameters is frequent item set and support is compared. Here if the Support value is increased the frequent item sets is decreased linearly. But the frequent item sets of proposed system are producing the high frequent item set when compared with existing system. Based on the comparison and results from the experiment show the proposed approach works better than the other existing systems.

\section{CONCLUSION AND FUTUREWORK}

In this proposed work, the enhanced HITS algorithm with weighted association rule mining and with fuzzy weighted association rule mining is developed. The enhanced HITS algorithm assigns the weights in online manner where the weight value is calculated in dynamic manner for the item set. In this the WARM is used for rule generation based on the user threshold value and ranking them based on that value. WARM rule is to calculate frequent item set with large number of iteration to produce an outcome to be generated. Whereas the FWARM is ordering the item set based on the fuzzy weighted value and confidence value have chance of computation complexity. From the experimental result FWARM with enhanced HITS is an efficient algorithm to generate the rules. As a future work, genetic algorithm is used to tune the membership value and find optimal membership value to bring more appropriate association rules. In this work new genetic improved fuzzy weighted association rule mining were proposed using enhanced HITS algorithms. The experimental results were measured with real-time dataset food mart between the WARM with enhanced HITS and FWARM with enhanced HITS. Proposed FWARM with enhanced HITS system performs better than the existing WARM with enhanced HITS in real-time dataset food mart.
REFERENCES


