International Journal of Advanced Research in

Computer Science Engineering and Information Technology

Volume: 5 Issue: 3 22-Mar-2015,ISSN_NO: 2321-3337

Infrequent Weighted Itemset Mining Using Frequent Pattern Growth

Namita Dilip Ganjewar

Namita Dilip Ganjewar, Department of Computer Engineering, Pune Institute of Computer Technology, India.

.

ABSTRACT— *Itemset mining has been an active area of research due to its successful application in various data mining scenarios including finding association rules. Frequent weighted itemsets represent correlations frequently holding in data in which items may weight differently. Infrequent Itemset mining is a variation of frequent itemset mining where it finds the uninteresting patterns i.e., it finds the data items which occurs very rarely. This seminar tackles the issue of discovering rare and weighted itemsets, i.e., the infrequent weighted itemset (IWI) mining problem. Two algorithms that perform IWI and Minimal IWI mining efficiently are presented. This new algorithm is based on the pattern-growth paradigm to find minimally infrequent itemsets. A minimally infrequent itemset has no subset which is also infrequent.*

Index Terms— Association rules, infrequent patterns, IWI support, FP Growth

1. INTRODUCTION

Itemset mining is an exploratory data mining technique widely used for discovering valuable correlations among data. The first attempt to perform itemset mining was focused on discovering frequent itemsets, i.e., patterns whose observed frequency of occurrence in the source data (the support) is above a given threshold [5]. Frequent itemsets find application in a number of real-life contexts (e.g., market basket analysis, medical image processing, biological data analysis). However, many traditional approaches ignore the influence or interest of each item or transaction within the analyzed data. To allow treating items or transactions differently based on their relevance in the frequent item set mining process, the notion of weighted item set has also been introduced [7].

A weight is associated with each data item and characterizes its local significance within each transaction. The significance of a weighted transaction, i.e., a set of weighted items, is commonly evaluated in terms of the corresponding item weights [10]. Furthermore, the main itemset quality measures (e.g., the support) have also been tailored to weighted data and used for driving the frequent weighted itemset mining process. In recent years, the attention of the research community has also been focused on the infrequent itemset mining problem, i.e., discovering itemsets whose frequency of occurrence in the analyzed data is less than or equal to a maximum threshold. Infrequent itemset discovery is applicable to data coming from different real-life application contexts such as statistical disclosure risk assessment from census data and fraud detection.

2. Literature Survey

2.1 The Apriori Algorithm

Apriori was the first proposed algorithm in association rule mining, to identify the frequent itemsets in the large transactional database [5]. Apriori works in two phases. During the first phase it generates all possible Item sets combinations. These combinations will act as possible candidates. The candidates will be used in subsequent phases. In Apriori algorithm, first the minimum support is applied to find all frequent itemsets in a database and Second, these frequent item sets and the minimum confidence constraint are used to form rules. The advantages of Apriori algorithm are as follows:

1. Easy to implement and can be parallelized easily
2. It uses large itemset property

The main drawback of Apriori is the generation of large number of candidate sets. The efficiency of apriori can be improved by Monotonicity property, hash based technique, Partioning methods and so on. The apriori needs (n+1) scans if n is length of longest pattern. So it increases computational time [6].

2.2 FP Growth Algorithm

The drawback of Apriori can be improved by frequent pattern Growth algorithm. This algorithm is implemented without generating the candidate sets. This algorithm proposes a tree structure called FP tree structure, going to collect information from the database and creates an optimized data structure as Conditional pattern [6]. Initially it Scans the transaction database DB once and Collects the set of frequent items F and their supports and then Sort the frequent itemsets in descending order as L, based on the support count. This algorithm reduces the number of candidate set generation, number of transactions, number of comparisons. Advantages of FP Growth include:

1. Only two passes over data set
2. No candidate generation
3. Faster than Apriori

But the major drawback of FP Growth was the FP tree was expensive to build and 80% of CPU was used for traversing the FP-trees.

2.3 FP-Growth* Algorithm

Grahne et al, found that 80% of CPU was used for traversing the FP-trees. Fp-growth* algorithm uses FP-tree data structure in combination with the array-based and incorporates various optimization techniques [8]. Array-based technique is used to reduce the traversal time of FP-tree. It reduces the memory consumption compared to FP-growth algorithm.

.

2.4 DynFP-growth Algorithm

The main drawback of the Apriori-like methods is at the candidate set generation and test. This problem was taken into consideration by introducing a novel, compact data structure, named frequent pattern tree, or FP-tree, is not unique for the same logical database. This approach can provide a very quick response to any queries even on databases that are being continuously updated. Because the dynamic reordering process, Gyorodi C et al [9] proposed a modification of the original structures, by replacing the single linked list with a doubly linked list for linking the tree nodes to the header and adding a master table to the same header.

2.5 Enhanced FP-Growth Algorithm

Enhanced-FP, it does its work without any prefix tree and any other complex data structure. It processes the transactions directly, so its main strength is its simplicity. It initially scans the supports of the items and is calculated. The items whose support count is less than minimum support are discarded and specified as infrequent items. Then the items in the database are sorted in ascending order with respect to their support [10]. And the initial transaction database is converted in to a set of transaction list, with one list for each item. These lists are stored in array, each of which contains a pointer to the head of the list. And the Transaction lists are traversed from left to right for finding all the frequent item set that contain the item the list corresponds to. Before a transaction list is processed, its support count is checked, if it exceeds than minimum support count than there must be a frequent item set.

2.6 IFP-min Algorithm

IFP min algorithm that uses a recursive approach to mine minimally infrequent item sets (MIIs). The infrequent item sets are then reported and it gets pruned from the database. The items presented in the modified database are individually frequent. This algorithm then selects the MIIs and it divides into two non-disjoint sets as residual database and projected database [4]. First the IFP-min algorithm is applied to residual database, where the MIIs are reported, if the database has the single item then it is considered to be the item itself or as an empty set. Then IFP-min algorithm is applied to projected database. The itemsets in the projected database share the lf-item as a prefix. The MIIs obtained from the projected database by recursively applying the algorithm are compared with those obtained from residual database. If an itemset is found to occur in the second set, it is not reported; otherwise, the lf-item is included in the itemset and is reported as an MII of the original database [4]. The use of residual tree is to reduce the computational time.

- If support threshold is too high, less no. of frequent itemsets will be generated resulting in loss of valuable associations.
- If support threshold is too low, large no. of frequent itemsets will be generated resulting in large no. of association rules thereby making it difficult to choose important association rules.

- To alleviate this problem, Multiple Level Minimum Support (MLMS) model was proposed where separate thresholds were assigned to itemsets of different sizes [4].
- This provided the advantage of Optimization of no. of association rules generated and we can constrain the no. of frequent and infrequent itemsets generated.

Paper	Features	Limitation
J. Han, J. Pei, and Y. Yin Mining frequent patterns without candidate generation, 2000	• Depth first search • Only two passes over data set • No candidate generation • Faster than Apriori	1. FP tree is expensive to build 2. 80% of CPU involved in tree traversals
Grahne O. and Zhu J Efficiently Using Prefix-trees in Mining Frequent Itemsets, 2007	• FP-Growth* algorithm uses FP-tree data structure in combination with the array-based data structure • Reduced traversal time, Reduced memory requirement	1. Rare combinations cannot be discovered
C.K.-S. Leung, C.L. Carmichael, and B. Hao Efficient Mining of Frequent Patterns from Uncertain Data, 2008	• Probabilistic frequent itemset mining integrated with Apriori based algorithms.	1. Weight assignment 2. Computational time is more
Cornelia Gyorodi, Robert Gyorodi, T. Cofeey& S. Holban Mining association rules using Dynamic FP-trees, 2007	• Doubly linked list for linking the tree nodes to the header and adding a master table to the same header	1. Additional overhead of master table 2. Not efficient for large databases 3. Tree traversal time
A. Gupta, A. Mittal, A. Bhattacharya Mining Infrequent Itemset Mining Using Pattern-Growth Paradigm and Residual Trees, 2011	• IFP min algorithm with Multiple Level Minimum Support (MLMS) • Optimization of no. of association rules generated	1. Residual Trees are expensive to build. 2. Memory requirement is more as in MLMS separate thresholds are defined at each level

Islam A.B.M.R., Suwon, Tae-Sun Chung An Improved Frequent Pattern Tree Based Association Rule Mining Technique, 2012	• New and improved FP tree with a table and a new algorithm for mining association rules. • Mines all possible frequent item set without generating the conditional FP tree and provides the frequency of frequent items, which is used to estimate the desired association rules.	1. Additional overhead of maintaining a table
Ahmad, T, Jamia Millia Islamia, Doja, M.N. Opinion Mining Using Frequent Pattern Growth Method from Unstructured Text, 2013	• The FP-growth method for frequent pattern mining from review documents which act as a backbone for mining the opinion words along with their relevant features	1. Massive no. of conditional FP trees generated.
Hao Yan, Bo Zhang, Yibo Zhang, Fang Liu, Zhenming Lei Discovery of Frequent Patterns from Web Log Data by using FP-Growth algorithm for Web Usage Mining, 2013	• The automatic discovery and analysis of patterns in click stream and providing valuable information about the user's interest • Most frequently access pattern is generated	1. Some wrong patterns generated because of semantics problem
Kostyantyn Demchuk, Douglas J. Leith A Fast Minimal Infrequent Itemset Mining Algorithm, 2014	• New algorithm Kyiv, for finding all minimal infrequent attribute combinations in both sequential and parallel form • Breadth first search of the prefix tree • Use of a recursive data structure called Graph to hold the prefix tree levels • Fast access to the children is achieved by use of a hash table	1. More memory required but most efficient in all existing algorithms

Baralis, E, Cagliero, L, Chiusano, S. Frequent weighted itemset mining from gene expression data, 2013	• Gene Expression Datasets (GEDs) usually consist of the expression values of thousands of genes within hundreds of samples. • This paper presents a novel approach to discovering gene correlations from GEDs which does not require data discretization.	1. Complex structure and feature data of genes

3. PROPOSED SYSTEM

3.1 MATHEMATICAL MODEL

3.1.1 Definition of the System

Let S be the system that gives the infrequent patterns in the database such that,

$S = \{s, e, X, Y, f_{main}, DD, NDD, f_{friend}, \phi\}$

Where,

s- initial state

e - End state

X = Input to system = $\{ I, T, S_t \}$ such that,
I = Set of data items = $\{ i_1, i_2, \ldots, i_n \}$
T = Transactional data set = $\{ t_1, t_2, \ldots, t_n \}$
Where $t_q = \{ (i_1,w_1), (i_2,w_2), \ldots, (i_n,w_n) \}$
S_t = Support threshold
Y= Set of IWI satisfying S_t from T
F_{main}- Main algorithm resulting into outcome Y

DD- Deterministic data

1. Set of data items I
2. Transactional data set T
3. Support Threshold S_t

NDD- Non-Deterministic data

1. Set of IWI satisfying S_t from T

ϕ- constraints

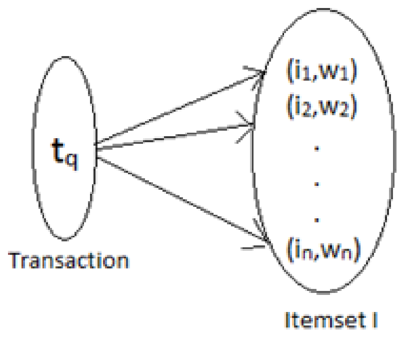

FIGURE 1. Mapping Diagram

3.1.2 Functions in the system

For Infrequent Weighted Itemset Miner Algorithm that is IWI_Miner which addresses the task of discovering all IWIs that satisfy S_t in T. The main function is IWI_Miner returning Y, the set of IWIs satisfying $S_{t.}$

$Y \longleftarrow$ IWI_Miner (Tree, S_t)
$f_{friend1} \rightarrow$ countItemIWI_support (T)

Let I be an itemset, T a weighted transactional data set, IS (t_q) the set of items in tq Ɛ T, and W_f a minimum or maximum weighting function. The IWI-support of I in T is defined as follows:

$$\text{IWI-support}(I,T) = \sum_{t_q \in T |\ I \subseteq IS(t_q)} W_f(I,t_q).$$

If $W_f = W_{min}$ then the IWI-support measure is denoted as IWI-support-min. Otherwise in case $W_f = W_{max}$, it is denoted as IWI-support-max.

For Minimal Infrequent Weighted Itemset Miner Algorithm that is IWI_Mining which addresses the task of discovering only minimal IWIs that satisfy S_t in T. The main function is IWI_Mining returning Y, the set of IWIs satisfying $S_{t.}$

$Y \longleftarrow$ IWI_Mining (Tree, $S_{t,}$ prefix)
condPatterns $\longleftarrow$ generateConditionalPatterns (Tree, I)
$Tree_I$ = createFPTree (condPatterns)
prunableItems $\longleftarrow$ identifyPrunableItems ($Tree_{I,}$ S_t)
$Tree_I \longleftarrow$ pruneItems ($Tree_{I,}$ prunableItems

3.1.3 Success State

1. Set of infrequent patterns i.e Y correctly discovered satisfying S_t in T
2. IWI Support measures calculated correctly whether it is IWI-support-min measure or IWI-support-max measure

3.1.4 Failure State

1. All infrequent patterns not included in output Y so some of the important patterns missed out.
2. IWI Support measures not calculated correctly.

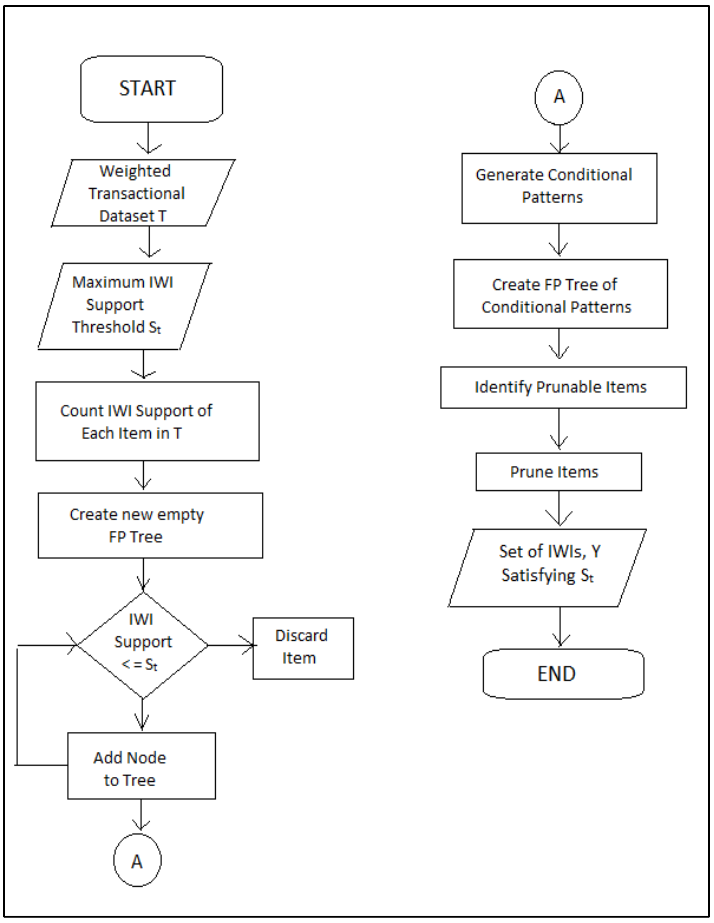
START
Weighted Transactional Dataset T
Maximum IWI Support Threshold S_t
Count IWI Support of Each Item in T
Create new empty FP Tree
IWI Support < = S_t
Discard Item
Add Node to Tree
A
A
Generate Conditional Patterns
Create FP Tree of Conditional Patterns
Identify Prunable Items
Prune Items
Set of IWIs, Y Satisfying S_t
END

FIGURE 2. Flowchart of the System

3.2 Algorithmic Solution

This section presents two algorithms, namely Infrequent Weighted Itemset Miner and Minimal Infrequent Weighted Itemset Miner, which address tasks (A) and (B), that is,

1. A. Discovering all IWIs that satisfy S_t in T,
2. B. Discovering all MIWIs that satisfy S_t in T.

Task (A) entails discovering all IWIs (minimal and not), task (B) selects only minimal IWIs, which represent the smallest infrequent item combinations satisfying the constraints [1].

3.2.1 Algorithm IWI – Miner (T, Ɛ)

Input: T, a weighted transactional dataset

Input: Ɛ, a maximum IWI – support threshold

Output: F, the set of IWIs satisfying Ɛ

```
1.  F = 0   /* Initialization */
       /* Scan T and count the IWI – support of each item */
2.  countItemIWI-support(T)
3.  Tree ← a new empty FP – tree;  /* Create the initial FP – tree from T */
4.  for all weighted transaction $t_q$ in T do
5.  $TE_q$ ← equivalentTransactionSet ($t_q$)
6.  for all transaction $te_j$ in $TE_q$ do
7.  insert $te_j$ in Tree
8.  end for
9.  end for
10. F ← IWIMining ( Tree, Ɛ, null )
11. return F
```

3.2.2 Algorithm IWIMining (Tree, Ɛ, prefix)

Input: Tree, a FP – tree

Input: Ɛ, a maximum IWI – support threshold

Input: prefix, the set of items / projection patterns with respect to which Tree has been generated.

Output: F, the set of IWIs extending prefix

```
1.  F = 0
2.  for all item i in the header table of Tree do
3.  I = prefix U { i }   /* Generate a new itemset I by joining prefix and i with
    IWI – support set to the IWI – support of item i */
    /* If I is infrequent store it */
4.  if IWI - support ( I ) < = Ɛ then
5.  F ← F U { I }
6.  end if
       /* Build I's conditional pattern base and I's conditional FP – tree */
7.  condPatterns ← generateConditionalPatterns( Tree, I )
8.  $Tree_I$ = createFP-tree ( condPatterns )
       /* Select the items that will never be part of any infrequent itemset */
9.  prunableItems ← identifyPrunableItems ( $Tree_I$, Ɛ )
       /* Remove from $Tree_I$ the nodes associated with prunable items */
```

```
10.     $Tree_I \leftarrow$ pruneItems ( $Tree_I$, prunableItems )
11. if $Tree_I \neq \emptyset$ then
12.     F $\leftarrow$ F U IWIMining ( $Tree_I$, Ɛ, I )   /* Recursive Mining */
13. end if
14. end for
15. return F
```

4. Summary And Conclusion

This seminar focuses the issue of discovering infrequent itemsets by using weights for differentiating between relevant items and irrelevant items within each transaction. Two FP - Growth like algorithms that accomplish IWI and MIWI mining efficiently are also proposed. Analysis and monitoring of multi-core system usage is commonly devoted to (i) detecting system malfunctioning, (ii) optimizing computational load balancing and resource sharing, and (iii) performing system resizing. The patterns extracted by IWI Miner and MIWI Miner from the real-life data sets are used to analyze and validate the multicore system usage. The other applications of the discovered infrequent patterns are in decision making, error detection and recovery etc.

5. Future Enhancement

As future work, we can integrate the proposed approach in an advanced decision-making system that supports domain expert's targeted actions based on the characteristics of the discovered IWIs. We can use the discovered infrequent patterns in the applications like error detection and recovery, fraud detection, finding outliers in the database. Furthermore, the application of different aggregation functions besides minimum and maximum is to be studied.

REFERENCES

[1] Luca Cagliero, Paulo Garza, “Infrequent Weighted Itemset Mining Using Frequent Pattern Growth”, IEEE Transactions on Knowledge and Data Engineering, Vol.26, No.4, April 2014.

[2] Kostyantyn Demchuk, Douglas J. Leith, “A Fast Minimal Infrequent Itemset Mining Algorithm”, ACM Transactions on Knowledge Discovery from Data, Vol.V, No.N, March 2014.

[3] Ahmad, T, Jamia Millia Islamia, Doja M.N, “Opinion Mining Using Frequent Pattern Growth Method from Unstructured Text”, IEEE Transactions on Knowledge and Data Engineering, Vol.24, No.3, Aug 2013.

[4] A. Gupta, A. Mittal, A. Bhattacharya, “Minimally Infrequent Itemset Mining Using Pattern-Growth Paradigm and Residual Trees,” Proc. Int’l Conf. Management of Data (COMAD), pp. 57- 68, 2011.

[5] Agrawal R, Imielinski T, Swami A, “Mining association rules between sets of items in large databases”, In proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, 1993.

[6] J. Han, J. Pei, Y. Yin, “Mining frequent patterns without candidate generation,” Proceedings of ACM SIGMOD International Conference on Management of Data, ACM Press, Dallas, Texas, pp. 1-12, May 2000.

[7] W. Wang, J. Yang, P.S. Yu, “Efficient Mining of Weighted Association Rules (WAR),” Proc. Sixth ACM SIGKDD Int’l Conf. Knowledge Discovery and data Mining (KDD ’00), pp. 270-274, 2000.

[8] Grahne O, Zhu J, “Efficiently Using Prefix-trees in Mining Frequent Itemsets”, In Proc. Of the IEEE ICDM Workshop on Frequent Itemset Mining, 2004.

[9] Cornelia Gyorodi, Robert Gyorodi, T. Cofeey, S. Holban, “Mining association rules using Dynamic FP-trees”, Proceedings of The Irish Signal and Systems Conference, University of Limerick, Limerick, Ireland, 30th June-2nd July 2007, ISBN 0-9542973-1-8, pag. 76-82.

[10] K. Sun, F. Bai, “Mining Weighted Association Rules without Preassigned Weights,” IEEE Trans. Knowledge and Data Eng., vol. 20, no. 4, pp. 489-495, Apr. 2008.

[11] Islam, A.B.M.R, Tae-Sun Chung, “An Improved Frequent Pattern Tree Based Association Rule Mining Technique”, Information Science and Applications (ICISA), 2011.