An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining

Ph.D. Synopsis

Submitted To

Gujarat Technological University

For the Degree

of

Doctor of Philosophy

in

Computer / IT Engineering

By

Patel Mahendrakumar Narottamdas

Enrollment No: 189999913028

Supervisor:

Dr. Sanjay M. Shah

Professor and Head CE Department,

Government Engineering College Rajkot, Gujarat, India

Table of Contents

1. Title of the Thesis and Abstract

1.1. Title of the Thesis

An Adjacency Matrix based Multiple Fuzzy Frequent Itemsets Mining.

1.2. Abstract

Discovering helpful information from transactions is becoming an important research issue. Several frequent itemsets mining algorithms are proposed for association rule mining which handle only binary datasets. These methods concentrate on an item's presence or absence in a dataset. However, in some situations in real life, it is crucial to consider the quantity of items. A fuzzy technique is used to handle quantitative datasets and to generate meaningful representations of the dataset. Thus several algorithms were developed to discover fuzzy frequent itemsets from quantitative transactions. Most of them merely take the linguistics term with the highest cardinality into account. As a result, the number of original elements and fuzzy regions processed is equal. On the other hand, decision-making can be made more successfully when an item has several fuzzy zones. Existing approaches scan the database more than once, and the high number of join counts (candidate itemsets) required thus degrade the algorithm's performance by increasing execution time.

In this research, we proposed AMFFI (Adjacency matrix-based multiple fuzzy frequent itemsets mining) and MFFPA-2 (multiple fuzzy frequent itemsets mining using adjacency matrix with type-2 membership function) using an Adjacency matrix and Fuzzy-Tid-list structures to discover multiple fuzzy frequent itemsets (MFFI) that scan the database only once. AMFFI is proposed for mining MFFI from quantitative transactions. AMFFI technique uses a type-1 membership function to transform quantitative datasets into fuzzy linguistics terms. An efficient search space exploration strategy is proposed to find the occurrence of two fuzzy linguistic terms together immediately from the adjacency matrix to minimize the join counts and speed up discovery MFFI. The proposed MFFPA-2 uses type-2 membership function to transform quantitative database into fuzzy linguistics terms. The type 2 Fuzzy Set could be useful for providing more reliable and agile decision-making by considering many uncertainty possibilities and considering more complex relationships between variables. Extensive experiments have been conducted to verify efficiency regarding runtime, memory usage, and join counts with different min support thresholds. Experimental results demonstrate that the designed approaches AMFFI and MFFPA-2 achieved superior performance compared to cutting-edge techniques. The AMFFI improves execution time by 8% to 81% and node join count by 93% to 99%. The MFFPA-2 improves execution time by 38% to 75% and node join count by 93% to 99%.

2. A brief description of the problem and State-of-art methods

In recent decades, online and mall shopping have been drastically increasing. For increasing the business, discovering valuable information from datasets is very important. Data mining techniques require finding knowledge from a large volume of the dataset. AR mining [1][2], clustering [3], and classification [4][5] are the three primary categories of Knowledge Discovery from Dataset (KDD) methods [1]. In FIs (frequent itemsets) mining of association rule is frequently employed. The Apriori algorithm is the first and fundamental data mining algorithm for association rule mining to mine common itemsets in a level-by-level (level-wise) methodology presented by Agrawal et al. [2]. Before finding FIs level by level, it generates candidate itemsets and prunes them. This method requires a time-consuming computation involving repeated database scanning and creating numerous candidate itemsets. Since Apriori requires multiple-time database scanning and generates more candidate sets, Han et al. [6] presented an FP-growth mining method to create frequent itemsets (FIs) without creating candidate itemsets and scanning the database only two times.

Quantitative databases are used in frequent itemsets mining for decision-making in real-world scenarios. It is challenging to manage the quantitative database. Mostly all authors used fuzzy set theory to manage quantitative databases. In fuzzy set theory, quantitative values of an item in the transaction are transformed into linguistic terms using a pre-defined membership function [7]. In [8], the authors used the max cardinality value in a level-by-level approach to mining fuzzy frequent itemsets (FFIs). Using maximum cardinality generating FFIs cost is minimized, but not generate all possible FFIs. Many authors present multiple fuzzy frequent itemsets (MFFIs) techniques for generating complete fuzzy frequent itemsets.

Mining fuzzy-frequent itemsets

Itemset is a set of items. Several 'm' unique items are called the itemset-I (i_1, i_2,..., i_m). The quantitative dataset D has 'n' transactions made of items from itemset-I, where D=T_1, T_2, T_3,..., $T_{n.}$ Every transaction contains the notation $T_q \in D$. Every transaction also includes a TID, which stands for a unique identifier. Each transaction T_q consists of an item and the value of the buy quantity; let us call it w_{iq}. "k-itemset" refers to an itemset of length K=i_1, i_2... i_k.

In the example below, Table 1 displays a sample quantitative dataset-D of seven transactions. The minimum support Ø=1. The Type-1 membership function £1 and Type-2 membership function £2 are demonstrated in Figure 1 and Figure 2, respectively.

Table 1. Quantitative dataset

TID	Item with Quantity
1	A-4, B-3, C-2, D-2
2	B-3, C-2, E-3
3	A-5, B-3, C-4, E-4
4	A-2, C-1, D-3
5	A-4, B-2, C-5
6	B-3, C-3, D-2, E-2
7	C-3, E-2

Mining fuzzy-frequent itemsets typically involves the following three steps.

Step 1: Determine the item's (Linguistic variable) fuzzy terms.

Consider the dataset D and item i (i⊆I), and the value of i is the collection of fuzzy terms. The built-in type-1 and type-2 membership functions produce £1 and £2, seen in Figure 1 and Figure 2, respectively.

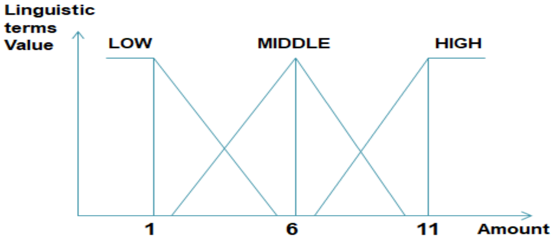

Fig. 1. Type-1 Membership Function

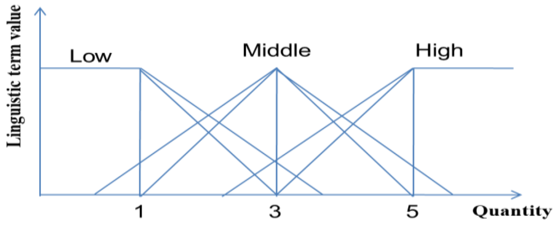

Fig. 2. Type-2 Membership Function

Fuzzy terms are represented as l_{i1}, l_{i2},…, l_{ih}, where h is the membership degree. Figure 1 and figure 2 show the 3-term membership function means here h=3. It may differ as 4-term or 5-term as per requirements. Three linguistic concepts are employed in this example: High-H, Middle-M, and Low-L. Term V_{iq} for transaction T_q is the quantitative value of i (item). Term F_{iq} is the linguistic term of item i. F_{iq} generated from item i quantity value V_{iq} using membership function £1 or £2. Fiq for membership function £1 of V_{iq} for item i is shown below.

$$f_{iq}(v_{iq}) = \frac{f_{iq1}}{l_{i1}} + \frac{f_{iq2}}{l_{i2}} \ldots + \frac{f_{iqh}}{l_{ih}} \tag{1}$$

F_{iqk} is the fuzzy value of k-th linguistic terms of l_{ik}, $1 \leq k \leq h$, and $f_{iqk} \subseteq [0, 1]$. For instance, the 3-term membership function £1 used in the example above represents item A with the quantity five in linguistic terms (0.2/AL, 0.8/AM, 0.0/AH). The quantitative dataset should first be transformed into a fuzzy set, say D', with several linguistic terms for each item in each transaction illustrated in Table 2 by using the membership function £1.

Table 2: Fuzzy dataset generated by Type1 Member function

1	A B C D::4 3 2 2	0.5/AM + 0.5/AH, 1/BM, 0.5/CL + 0.5/CM, 0.5/DL + 0.5/DM
2	B C E::3 2 3	1/BM, 0.5/CL + 0.5/CM, 1/EM
3	A B C E::5 3 4 4	1/AH, 1/BM, 0.5/CM + 0.5/CH, 0.5/EM + 0.5/EH
4	A C D::2 1 3	0.5/AL + 0.5/AM, 1/CL, 1/DM
5	A B C::4 2 5	0.5/AM + 0.5/AH, 0.5/BL + 0.5/BM, 1/CH
6	B C D E::3 3 2 2	1/BM, 1/CM, 0.5/DL + 0.5/DM, 0.5/EL + 0.5/EM
7	C E::3 2	1/CM, 0.5/EL + 0.5/EM

F_{iq} for £2 is a set of three linguistic terms f_{iq1}^{lower}, f_{iq1}^{upper}/l_{i1} for membership value low, f_{iq2}^{lower}, f_{iq2}^{upper}/l_{i2} for membership value middle, and f_{iq3}^{lower}, f_{iq3}^{upper}/l_{i3} for membership value high as shown in following equation 2. Where l_{il} shows l-th linguistic (fuzzy) terms, f_{iql}^{lower} and f_{iql}^{upper} show lower and upper membership values of V_{iq} for item i.

$$f_{iq}(v_{iq}) = \frac{f_{iq1}^{lower}, f_{iq1}^{upper}}{l_{i1}} + \frac{f_{iq2}^{lower}, f_{iq2}^{upper}}{l_{i2}} \ldots + \frac{f_{iqh}^{lower}, f_{iqh}^{upper}}{l_{ih}} \tag{2}$$

Table 3 shows the resultant fuzzy dataset say D', after applying the membership function £2 on given an example.

Table 3. Fuzzy Dataset generated by Type2 member function

TID	Original Dataset	Fuzzy dataset
1	A-4, B-3, C-2, D-2	$\frac{0.5,0.62}{AM}+\frac{0.5,0.62}{AH}, \frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CL}+\frac{0.5,0.62}{CM}, \frac{0.5,0.62}{DL}+\frac{0.5,0.62}{DM}$
2	B-3, C-2, E-3	$\frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CL}+\frac{0.5,0.62}{CM}, \frac{0,0.25}{EL}+\frac{1,1}{EM}+\frac{0,0.25}{EH}$
3	A-5, B-3, C-4, E-4	$\frac{0,0.25}{AM}+\frac{1,1}{AH}, \frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0.5,0.62}{CM}+\frac{0.5,0.62}{CH}, \frac{0.5,0.62}{EM}+\frac{0.5,0.62}{EH}$
4	A-2, C-1, D-3	$\frac{0.5,0.62}{AL}+\frac{0.5,0.62}{AM}, \frac{1,1}{CL}+\frac{0,0.25}{CM}, \frac{0,0.25}{DL}+\frac{1,1}{DM}+\frac{0,0.25}{DH}$
5	A-4, B-2, C-5	$\frac{0.5,0.62}{AM}+\frac{0.5,0.62}{AH}, \frac{0.5,0.62}{BL}+\frac{0.5,0.62}{BM}, \frac{0,0.25}{CM}+\frac{1,1}{CH}$
6	B-3, C-3, D-2	$\frac{0,0.25}{BL}+\frac{1,1}{BM}+\frac{0,0.25}{BH}, \frac{0,0.25}{CL}+\frac{1,1}{CM}+\frac{0,0.25}{CH}+\frac{0.5,0.62}{DL}+\frac{0.5,0.62}{DM}$
7	C-3, E-2	$\frac{0,0.25}{CL}+\frac{1,1}{CM}+\frac{0,0.25}{CH}, \frac{0.5,0.62}{EL}+\frac{0.5,0.62}{EM}$

Step 2: Find the support count of each fuzzy item.

Fuzzy itemsets L_{ik}'s support count (scalar cardinality) is shown by the symbol sup (L_{ik}). Find each fuzzy itemset's support in this stage. According to this definition,

$$\text{Sup}\ (L_{ik}) = \sum_{q=0,\ L_{ik} \subseteq T_q\ ^\wedge\ T_q \in D'}^{n} f_{iqk} \tag{3}$$

In fuzzy dataset D', fuzzy item L_{ik}'s fuzzy value is f_{iqk}. Check each fuzzy item's sup (L_{ik}); if the minimum support requirement is satisfied, place the item in FL_1.

$FL_1 = FL_1 \cup (\text{sup}\ (L_{ik}) >= \emptyset)$. Where FL_1 is fuzzy 1-frequent itemsets, and Ø is the min-support threshold.

Step 3: Finding the Sup of each frequent fuzzy itemsets:

The following-level frequent itemsets are fuzzy k-itemsets with k=2, produced by fuzzy 1-frequent itemsets (FL_1). Fuzzy items from FL_1 are combined using the join procedure to create a candidate set, such as FC_2 (fuzzy 2-candidate itemsets). Consider the itemset X that was produced by merging the FL_1 itemsets A and B. Sup(X) = support of itemset: X, where $X \subseteq T_q$ and $T_q \in D'$, is consider the lowest fuzzy values of fuzzy itemsets A and B from truncation T_q. According to this definition,

$$\text{Sup (X)} = X \in L_i\ /\ \sum_{q=0,\ X \subseteq T_q\ ^\wedge\ T_q \in D'}^{n} \min(\text{faq}, \text{fbq}) \tag{4}$$

If they satisfy the minimal support, store in fuzzy 2-frequent itemsets (FL2) from FC2 itemsets. Similarly, fuzzy k-frequent itemsets are discovered later.

2.1 The state-of-the-art methods using the type-1 membership function

In [20], the authors tried to find a method to achieve the fuzzy association rules useful in quantitative data and relational databases. In [21], the author presented a novel mining method to extract common patterns for building itemsets from quantitative databases, use the Apriori Tid data structure. In [24], the authors used an FP-tree structure called FUFP-tree to reduce the execution time when new data is inserted or arrives. The same FP-tree-like structure, referred to as the FFP-tree (fuzzy frequent pattern) in [22], was employed by the authors to find FFIs in quantitative databases. There are several limitations but they are addressed in [9, 23]. A compressed fuzzy frequent pattern (CFFP-tree) structure was presented by the authors in [23], and in [9], they employ the upper bound fuzzy frequent pattern (UBFFP-tree) structure to discover FFIs. The CFFP-tree [23] and UBFFP-tree [9] structures employ a global sorting strategy to minimize the number of tree nodes. The author of [27] suggested the FC-Tree structure and FCFI-miner (Fuzzy closed frequent itemsets miner) for the aim of finding FFIS. In this method, the author used a superset pruning mechanism to speed up mining. Authors come upon MFFIs, which offer thorough details on all linguistic expressions in the fuzzy set. In [10], the author suggested using an MFFP-tree structure and mining MFFP growth to find MFFIs. Similar to this, authors created the CMFFP-tree [11] and UBMFFP-tree [12] ways to create MFFIs based on the CFFP-tree [23] and the UBFFP-tree [9], respectively. The authors of [14] developed an MFFI-miner technique and a fuzzy-list structure to find MFFIs. To decrease the search space, shorten the running time, and shrink the running space, the author of this method employed two pruning strategies. Several algorithms based on the fuzzy-set theory for discovering the required information were developed in progress [28, 29, and 30] for different applications and domains.

2.2 The state-of-the-art methods using the type-2 membership function

The solution, as mentioned above, solely counters type-1 fuzzy-set theory, which ignores uncertainty. The fuzzy-set idea with type-2 membership function [16],[25], and[26] was then put out and improved to more effectively present the acquired information with uncertainty. To merge pattern mining and type-2 fuzzy sets, Chen et al. [17] applied the standard level-wise like-Apriori method for mining level-wise fuzzy type-2 frequent patterns. However, the procedure necessitates generating large numbers of candidates, which is ineffective for the mining task. To store the information for the mining process, a list-based approach proposed by Lin et al., the strategy still needs to investigate many candidates for determining the true FFIs due to inefficient pruning algorithm and loose upper bound value on the pattern, which are not frequent. After that, Lin et al. provided a list-based approach to keep the necessary information for discovering frequent items [18]. This approach still needs to investigate many candidates to get the actual FFIs because it requires more effective search space trimming strategies and a flexible upper bound measure on the patterns that could be more promising. The authors of [19] employed a complex fuzzy list (CFL)-structure to find MFFIs, and that was similar to the fuzzy list structure from [14].

2.3 Open Issue

Existing approaches scan the database more than one time and generate a large number of candidates itemsets for mining MFFIs, which requires more join counts and increased running time. Research work is required to optimize running time, memory usages and node join counts.

3. Objective, Scope of the work, and problem statement

3.1 Aim and Research Objectives

This research aims to optimize running time, memory usage, and the number of node traversals required in fuzzy frequent itemsets mining. This research work proposes to achieve the following objectives:

- To study and investigate existing methods for fuzzy frequent itemsets mining.
- To identify the challenges for the fuzzy frequent itemsets mining.
- To identify the scope to improve the performance of the fuzzy frequent itemsets mining methods.
- To develop and investigate the efficient search space exploration technique to reduce the cost of fuzzy list join operations by reducing the number of comparisons required to join fuzzy lists.
- To design a novel structure to store the fuzzy value of the itemsets that can be used to develop an efficient pruning mechanism.
- To develop and investigate an efficient pruning mechanism to reduce the number of join operations by eliminating unnecessary join operations of fuzzy lists.
- To evaluate performance and compare the results with existing state-of-the-art methods.

3.2 Scope of the research

The research is performed with the following scope of work:

Research is focused on transaction datasets.

Problem statement

The focused problem statement of this research is:

"Design an efficient and accurate method to generate fuzzy frequent itemsets using Adjacency matrix."

4. Research contribution (Research Methodology)

The main contribution of this research is to design an efficient multiple fuzzy frequent itemsets mining method using an adjacency matrix. Proposed methods AMFFI and MFFPA-2 scan database only once and reduce the number of node join counts (candidate itemsets) by pruning un-frequent itemsets extracted from the adjacency matrix.

The research methodology comprises developing a novel approach to mining MFFIs using an adjacency matrix and fuzzy-tid-list structures. Section 5.1 discuss AMFFI, and 5.2 discuss the MFFPA-2 approach. Performance of the proposed approaches evaluated with state-of-the-art methods on standard real datasets.

4.1 An Adjacency matrix based Multiple Fuzzy Frequent Itemsets mining (AMFFI) technique

This section proposes a two-phase method for producing MFFIs. Create an adjacency matrix and fuzzy-tid list from the quantitative dataset D in phase 1. The next step is to effectively use the AMFFI method to find MFFIs from the adjacency matrix and fuzzy-tid-list. The suggested method efficiently creates full MFFIs by performing a single database scan.

4.1.1 Adjacency matrix and fuzzy-tid-list construction

During the first phase, the algorithm transforms the quantitative transaction values into a fuzzy set using the given member function with several linguistic terms.

Think about the membership function $£_1$ for three terms. AdjMat (M) of size (m*3) X (m*3) should first be constructed, where 'm' is the total number of items in the original dataset D. Three times as many items (m) are required as matrix space in this case. Membership function determines the size of the matrix.

$$AdjMat\ (M) = (m * t)\ \times (m * t) \tag{5}$$

Where 'm' is the number of items used in the original quantitative dataset, and 't' is the number of the fuzzy region used in the t-term membership function.

The quantitative dataset of the transaction T_q with the TID q is transformed into a fuzzy dataset by applying the membership function £1. Create a pair of converted fuzzy itemsets from transaction T_q for various fuzzy variables. The correspondence cell value of the adjacency matrix should be updated by adding the minimal fuzzy value of each pair and entering into corresponding fuzzy-tid-list.

AdjMat (L_i, L_j) =AdjMat (L_i, L_j) +min (f_{wiq}, f_{wjq}), Where L_i and L_j are fuzzy items whose fuzzy values f_{wiq} and f_{wjq}, respectively. If L_i and L_j don't already have fuzzy-tid-lists, create them. The transaction id q (TID of T_q) and minimum fuzzy value of the pair as min (f_{wiq}, f_{wjq}) were added to this fuzzy-Tid-list.

Let us consider an example of the quantitative dataset in Table 1 and the membership function in Fig 1. There are five items (A, B, C, D, E); accordingly, construct the adjacency matrix shown in Fig 3.

	B.L	B.M	B.H	C.L	C.M	C.H	D.L	D.M	D.H	E.L	E.M	E.H
A.L												
A.M												
A.H												
B.L												
B.M												
B.H												
C.L												
C.M												
C.H												
D.L												
D.M												
D.H												

Fig. 3 Adjacency Matrix

	B.L	B.M	B.H	C.L	C.M	C.H	D.L	D.M	D.H	E.L	E.M	E.H
A.L												
A.M		0.5		0.5	0.5		0.5	0.5				
A.H		0.5		0.5	0.5		0.5	0.5				
B.L												
B.M				0.5	0.5		0.5	0.5				
B.H												
C.L							0.5	0.5				
C.M							0.5	0.5				
C.H												
D.L												
D.M												
D.H												

Fig. 4 Adjacency Matrix after a 1st-row scan

Scan the first transaction from dataset D and apply the membership function £1 to create a fuzzy set as shown in table 2. So created pair of transformed fuzzy itemsets are "AM-BM, AM-CL, AM-CM, AM-DL, AM-DM, AH-BM, AH-CL, AH-CM, AH-DL, AH-DM, BM-CL, BM-CM, BM-DL, BM-DM, CL-DL, CL-DM, CM-DL, and CM-DM". A minimum fuzzy value adjacency matrix should be updated with all pair cooccurrences, as shown in Fig 4. At first, no fuzzy-tid-list is formed, generate a fuzzy-tid-list with TID=1 and the minimal fuzzy value for each pair as a result. Figure 5 shows generated Fuzzy-Tid-list after scanning the first row. For the subsequent transaction, the same steps are taken. Following the reading of every transaction, the resulting Adjacency matrix (M) and Fuzzy-Tid-list is illustrated in Figures 6 and 7, respectively.

AM-BM		AM-CL		AM-CM		AM-DL		AM-DM		AH-BM		AH-CL		AH-CM		AH-DL		AH-DM	
TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE
1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5

BM-CL		BM-CM		BM-DL		BM-DM		CL-DL		CL-DM		CM-DL		CM-DM	
TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE
1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5	1	0.5

Fig. 5 Fuzzy-Tid-list after a 1st-row scan

	B.L	B.M	B.H	C.L	C.M	C.H	D.L	D.M	D.H	E.L	E.M	E.H
A.L				0.5				0.5				
A.M	0.5	1		1	0.5	0.5	0.5	1				
A.H	0.5	2		0.5	1	1	0.5	0.5			0.5	
B.L						0.5						
B.M				1	2.5	1	1	1		0.5	2	0.5
B.H												
C.L							0.5	0.5			0.5	
C.M							1	1		1	2	0.5
C.H											0.5	0.5
D.L										0.5	0.5	
D.M										0.5	0.5	
D.H												

Fig. 6 Adjacency Matrix after all row scans

Fuzzy list	TID	FUZZY VALUE
AM-CM	1	0.5
AM-DL	1	0.5
AH-CL	1	0.5
AH-DL	1	0.5
AH-DM	1	0.5
CL-DL	1	0.5
CL-DM	1	0.5
CL-EM	2	0.5
AH-EM	3	0.5
AH-EH	3	0.5
BM-EH	3	0.5
CM-EH	3	0.5
CH-EM	3	0.5
CH-EH	3	0.5
AL-CL	4	0.5
AL-DM	4	0.5
AM-BL	5	0.5
AM-CH	5	0.5
AH-BL	5	0.5
BL-CH	5	0.5
BM-EL	6	0.5
DL-EL	6	0.5
DL-EM	6	0.5
DM-EL	6	0.5
DM-EM	6	0.5
AM-BM	1	0.5
	5	0.5
AM-CL	1	0.5
	4	0.5
AM-DM	1	0.5
	4	0.5
AH-CM	1	0.5
	3	0.5
BM-CL	1	0.5
	2	0.5
BM-DL	1	0.5
	6	0.5
BM-DM	1	0.5
	6	0.5
CM-DL	1	0.5
	6	0.5
CM-DM	1	0.5
	6	0.5
AH-CH	3	0.5
	5	0.5
BM-CH	3	0.5
	5	0.5
AH-BM	1	0.5
	3	1
	5	0.5
BM-EM	2	1
	3	0.5
	6	0.5
BM-CM	1	0.5
	2	0.5
	3	0.5
	6	1
CM-EM	2	0.5
	3	0.5
	6	0.5
	7	0.5
CM-EL	6	0.5
	7	0.5

Fig. 7 Fuzzy-Tid-list after all row scan

4.1.2 Phase 2 (AMFFI-miner to mine MFFIs)

In AMFFI approach Adjacency matrix, say M, an upper triangular matrix, is used to cut down on the enormous number of candidate generations used while creating frequent itemsets. In this phase, fuzzy-tid-lists created in phase 1 are used to extract MFFIs row by row from the adjacency matrix (M). The cell whose value is greater than or equal to the minimum support criterion (Ø) should be located by scanning the row starting from M. Fuzzy lists containing Row Number-Column Number of a known cell are retrieved from fuzzy-tid-lists and designated as fuzzy 2-frequent itemsets, such as FL_2 of this row. Create fuzzy k-frequent itemsets, such as FL_k (K>2), in a subsequent step by intersection-operating TIDs on FL_{k-1}. To quickly locate merged fuzzy lists, use the binary search technique.

Avoid joining fuzzy itemsets that can't create their superset knowing from the adjacency matrix M in order to limit the search space and candidate set. Take the second row from M in the case of a running example when Ø = 1. The cells BM, CL, and DM of this row number AM, met the minimum support requirement. FL_2 in this row is therefore AM-BM, AM-CL, and AM-DM. This potential superset is formed by joining AM-BM-CL, AM-BM-DM, and AM-CL-DM from FL_2. The proposed technique, however, does not join AM-CL-DM because it is aware that the created superset does not meet the required minimum support level. How? Fuzzy frequent itemsets AM-CL and AM-DM are present here. The CL row and DM column cell value check whether its superset is possible. If it is larger than or equal to the minimum support value, it might be achievable; if not, it is impossible. This value in our example, 0.5, does not meet the min support threshold, making its superset impossible. Extensions are discarded before joining since they are not fuzzy frequent itemsets. This way, the join operation minimizes the candidate set, improving the running time performance.

4.1.3 Experimental Study and Performance Evaluation of AMFFI

This section describes how the performance of the proposed method says AMFFI concerning the state-of-the-art MFFI-Miner [14] method. In [14], the authors make a comparison of their method MFFI-miner concerning the state-of-the-art GDF [31] and the UBMFFP tree [9] methods. The proposed AMFFI and MFFI-miner methods were implemented in Java. Experiments were evaluated to check performance on two real-life datasets, chess [32] and mushroom [32], as well as one synthetic dataset, T10I4D100k [32]. In the datasets, the item quantities were arbitrarily distributed in intervals of between 1 and 7. The runtime, memory utilization, and node join counts of the experiment are assessed for comparison against the planned approaches.

Runtime Analysis:

The implemented 3-term fuzzy linguistic AMFFI and MFFI-miner [14] were evaluated with different min-support thresholds to compare execution running time. The output of execution running time evaluated on chess, mushroom, and T10I4D100k dataset are shown in Fig 8, Fig 9, and Fig 10, respectively.

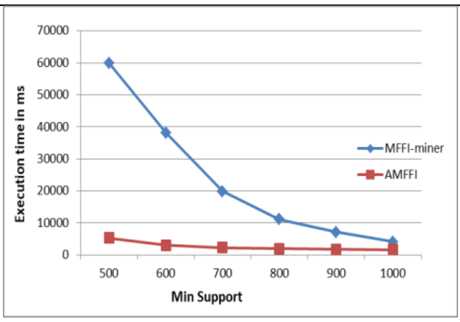

Fig. 8 Performance Evaluation of Chess Dataset

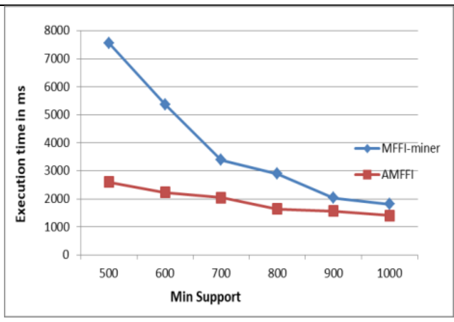

Fig. 9 Performance Evaluation of Mushroom Dataset

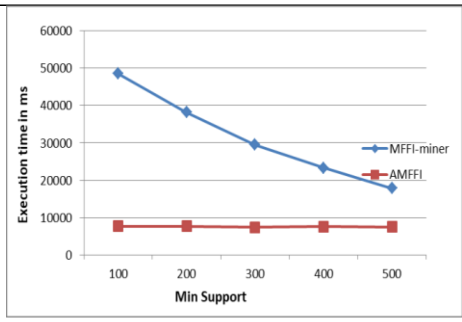

Fig. 10 Performance Evaluation of T10I4D100k Dataset

MFFI-miner [14] authors show that the running time performance of its proposed method is good concerning GDF [31] and the UBMFFP tree [9]. The result shows that the performance of the proposed AMFFI method is faster than the existing MFFI-miner method. The result also shows that the AMFFI method outperformed the existing method when taking a lower min-support threshold.

Join counts Analysis:

This section evaluates performance for the number of join count that occurs when generating MFFIs. The output of the number of join counts generated while evaluating the chess, mushroom, and T10I4D100k dataset are shown in Fig 11, Fig 12, and Fig 13, respectively.

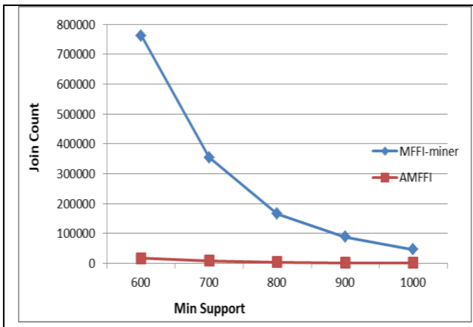

Fig. 11 Performance Evaluation of Chess Dataset

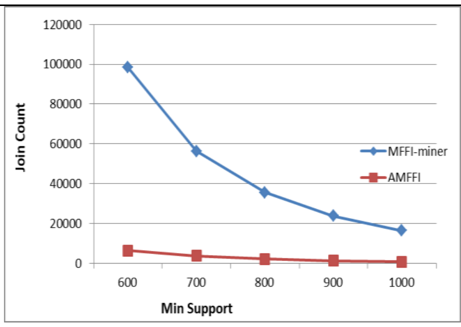

Fig. 12 Performance Evaluation of Mushroom Dataset

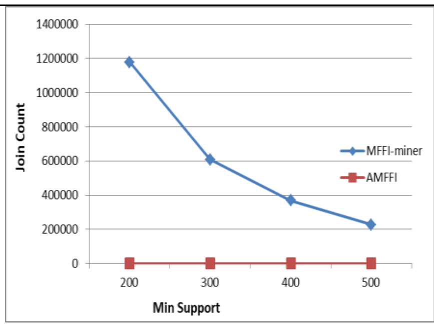

Fig. 13 Performance Evaluation of T10I4D100k Dataset

The result shows that the AMFFI method generates fewer join counts (candidate itemsets). Additionally, it has been noted that the AMFFI method's join count performance is the most impressive. When compared to cutting-edge approaches, the suggested AMFFI method produces less candidate itemsets.

Memory Usage Analysis:

In this section, performance is evaluated concerning memory utilization when evaluating experiments. The output of memory usage while evaluating the experiment on the chess, mushroom, and T10I4D100k datasets are shown in Fig 14, Fig 15, and Fig 16, respectively.

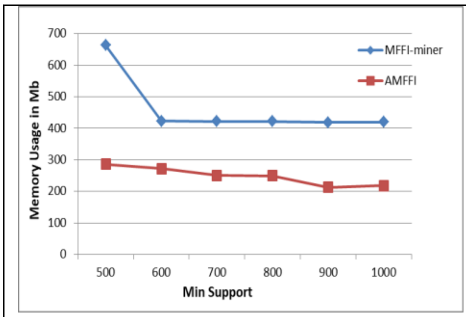

Fig. 14 Performance Evaluation of Chess Dataset

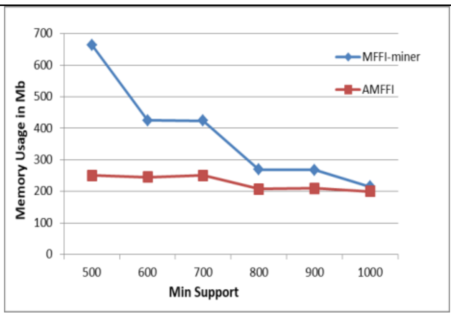

Fig. 15 Performance Evaluation of Mushroom Dataset

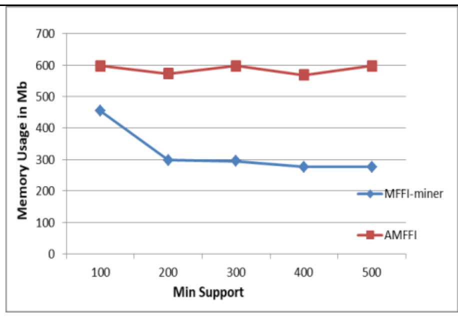

Fig. 16 Performance Evaluation of T10I4D100k Dataset

The outcome demonstrates that compared to the current MFFI-miner approach, the chess and mushroom dataset AMFFI method uses less memory. Furthermore, it has been found that the synthetic T10I4D100k dataset AMFFI method uses more memory than the MFFI-miner approach. Other experiments with different datasets show that the proposed method requires more memory when the number of items exceeds 1000.

4.2 An Efficient (MFFPA-2) Multiple Fuzzy Frequent Patterns Mining with Adjacency matrix and Type-2 Member function

This section suggests a two-step process for creating many fuzzy frequent itemsets. Using the quantitative dataset D, generate an adjacency matrix and Fuzzy-tid-list in phase 1. Using approach MFFPA-2, quickly extract numerous fuzzy frequent itemsets from the adjacency matrix, and Fuzzy-tid-list discussed in phase 2. The suggested technique effectively creates entire MFFIs from a single database scan.

4.2.1 Adjacency matrix and Fuzzy-tid-list construction

In this phase, first construct adjacency matrix M, same as previously. The size of matrix M is (I*h) × (I*h). In the membership function, h is the number of linguistic terms, and I is the number of items in D. Total items in D' (fuzzy dataset) is a product of I in D, and h say m. The corresponding required adjacency matrix is shown in Figure 3.

Next, in this phase, scan transaction T_q from quantitative database D and apply a pre-defined type-2 membership function $£_2$, which generates fuzzy linguistics terms, as shown in Table 3. Here are two values of each fuzzy linguistic term: the first is associated with a lower boundary, and the second is associated with a higher boundary of membership function, as shown in Figure 2. For example, the first transaction item A with quantity four generates two fuzzy linguistics terms, AM (A-middle) and AH (A-high), each linguistic term with a fuzzy value of 0.5 and 0.62 as lower and higher values, respectively, as shown in Table 3. Thus it is a complex task to mine MFFIs from two values of each fuzzy linguistic term. So, take the fuzzy interval value by taking an average of it, and use the centroid type-reduction approach [17] to reduce the complexity. Getting the interval value using the following formula

$$f_{iq1} = \frac{f_{iq1}^{lower} + f_{iq1}^{upper}}{2} \tag{6}$$

So, get 0.56 internal fuzzy values of linguistic terms AM and AH according to the given formula. This final transformed first transaction from D is displayed in Table 4.

Table 4. Final first fuzzy transaction

TID	Final Fuzzy dataset
1	$\frac{0.56}{AM}+\frac{0.56}{AH}, \frac{0.13}{BL}+\frac{1}{BM}+\frac{0.13}{BH}, \frac{0.56}{CL}+\frac{0.56}{CM}, \frac{0.56}{DL}+\frac{0.56}{DM}$

Add value to the adjacency matrix, the value of the corresponding cell in the adjacency matrix can be changed by entering the least fuzzy value of each pair.

$$\mathrm{AM(Li, Lj) = AM\ (Li, Lj) + \ min\ (fwiq, fwjq)} \tag{7}$$

fwiq and fwjq are the fuzzy value of the fuzzy items Li and Lj, respectively. Create a Fuzzy-tid-list for Li and Lj if it does not exist. A minimum of fwiq and fwjq was added with transaction ID q to the Fuzzy-tid-list. Completing the first-row adjacency matrix and Fuzzy-tid-list is shown in Figure 17 and Figure 18, respectively. After reading all rows final adjacency matrix and fuzzy-tid-list list are shown in Figures 19 and 20, respectively.

	BL	BM	BH	CL	CM	CH	DL	DM	DH	EL	EM	EH
AL												
AM	0.13	0.56	0.13	0.56	0.56		0.56	0.56				
AH	0.13	0.56	0.13	0.56	0.56		0.56	0.56				
BL				0.13	0.13		0.13	0.13				
BM				0.56	0.56		0.56	0.56				
BH				0.13	0.13		0.13	0.13				
CL							0.56	0.56				
CM							0.56	0.56				
CH												
DL												
DM												
DH												

Fig. 17. Adjacency Matrix after the first-row scan

Pair	TID	FUZZY VALUE
AM-BL	1	0.13
AM-BM	1	0.56
AM-BH	1	0.13
AH-BL	1	0.13
AH-BM	1	0.56
AH-BH	1	0.13
AM-CL	1	0.56
AM-CM	1	0.56
AH-CL	1	0.56
AH-CM	1	0.56
AM-DL	1	0.56
AM-DM	1	0.56
AH-DL	1	0.56
AH-DM	1	0.56
BL-CL	1	0.13
BL-CM	1	0.13
BM-CL	1	0.56
BM-CM	1	0.56
BH-CL	1	0.13
BH-CM	1	0.13
BL-DL	1	0.13
BL-DM	1	0.13
BM-DL	1	0.56
BM-DM	1	0.56
BH-DL	1	0.13
BH-DM	1	0.13
CL-DL	1	0.56
CL-DM	1	0.56
CM-DL	1	0.56
CM-DM	1	0.56

Fig. 18. Fuzzy-tid-list after the 1st-row scan

	BL	BM	BH	CL	CM	CH	DL	DM	DH	EL	EM	EH
AL				0.56	0.13		0.13	0.56	0.13			
AM	0.82	1.25	0.26	0.69	0.95	0.69	0.69	1.12	0.13		0.13	0.13
AH	0.82	2.12	0.26	0.56	1.25	1.12	0.56	0.56			0.56	0.56
BL				0.39	0.65	0.82	0.26	0.26		0.13	0.26	0.26
BM				1.25	2.81	1.25	1.12	1.12		0.13	1.56	0.69
BH				0.39	0.52	0.26	0.26	0.26		0.13	0.26	0.26
CL							0.82	1.69	0.13	0.26	0.69	0.13
CM							1.25	1.25	0.13	0.69	1.68	0.69
CH							0.13	0.13		0.13	0.69	0.56
DL												
DM												
DH												

Fig. 19. Adjacency Matrix after all row scan

AH-CL	
TID	FUZZY VALUE
1	0.56

AH-DL	
TID	FUZZY VALUE
1	0.56

AH-DM	
TID	FUZZY VALUE
1	0.56

BL-EL	
TID	FUZZY VALUE
2	0.13

BM-EL	
TID	FUZZY VALUE
2	0.13

BH-EL	
TID	FUZZY VALUE
2	0.13

CL-EH	
TID	FUZZY VALUE
2	0.13

AM-EM	
TID	FUZZY VALUE
3	0.13

AM-EH	
TID	FUZZY VALUE
3	0.13

AH-EM	
TID	FUZZY VALUE
3	0.56

AH-EH	
TID	FUZZY VALUE
3	0.56

CH-EH	
TID	FUZZY VALUE
3	0.56

AL-CL	
TID	FUZZY VALUE
4	0.56

AL-CM	
TID	FUZZY VALUE
4	0.13

AL-DL	
TID	FUZZY VALUE
4	0.13

AL-DM	
TID	FUZZY VALUE
4	0.56

AL-DH	
TID	FUZZY VALUE
4	0.13

AM-DH	
TID	FUZZY VALUE
4	0.13

CL-DH	
TID	FUZZY VALUE
4	0.13

CM-DH	
TID	FUZZY VALUE
4	0.13

CH-DL	
TID	FUZZY VALUE
6	0.13

CH-DM	
TID	FUZZY VALUE
6	0.13

CH-EL	
TID	FUZZY VALUE
7	0.13

AM-BH	
TID	FUZZY VALUE
1	0.13
3	0.13

AH-BH	
TID	FUZZY VALUE
1	0.13
3	0.13

AM-CL	
TID	FUZZY VALUE
1	0.56
4	0.13

AM-DL	
TID	FUZZY VALUE
1	0.56
4	0.13

AM-DM	
TID	FUZZY VALUE
1	0.56
4	0.56

BL-DL	
TID	FUZZY VALUE
1	0.13
6	0.13

BL-DM	
TID	FUZZY VALUE
1	0.13
6	0.13

BM-DL	
TID	FUZZY VALUE
1	0.56
6	0.56

BM-DM	
TID	FUZZY VALUE
1	0.56
6	0.56

BH-DL	
TID	FUZZY VALUE
1	0.13
6	0.13

BH-DM	
TID	FUZZY VALUE
1	0.13
6	0.13

BL-EM	
TID	FUZZY VALUE
2	0.13
3	0.13

BL-EH	
TID	FUZZY VALUE
2	0.13
3	0.13

BM-EM	
TID	FUZZY VALUE
2	1
3	0.56

BM-EH	
TID	FUZZY VALUE
2	0.13
3	0.56

BH-EM	
TID	FUZZY VALUE
2	0.13
3	0.13

BH-EH	
TID	FUZZY VALUE
2	0.13
3	0.13

CL-EL	
TID	FUZZY VALUE
2	0.13
7	0.13

CL-EM	
TID	FUZZY VALUE
2	0.56
7	0.13

CM-EL	
TID	FUZZY VALUE
2	0.13
7	0.56

CM-EH	
TID	FUZZY VALUE
2	0.13
3	0.56

AM-CH	
TID	FUZZY VALUE
3	0.13
5	0.56

AH-CH	
TID	FUZZY VALUE
3	0.56
5	0.56

BH-CH	
TID	FUZZY VALUE
3	0.13
6	0.13

CH-EM	
TID	FUZZY VALUE
3	0.56
7	0.13

AM-BL	
TID	FUZZY VALUE
1	0.13
3	0.13
5	0.56

AM-BM	
TID	FUZZY VALUE
1	0.56
3	0.13
5	0.56

AH-BL	
TID	FUZZY VALUE
1	0.13
3	0.13
5	0.56

AH-BM	
TID	FUZZY VALUE
1	0.56
3	1
5	0.56

AH-CM	
TID	FUZZY VALUE
1	0.56
3	0.56
5	0.13

BL-CL	
TID	FUZZY VALUE
1	0.13
2	0.13
6	0.13

BM-CL	
TID	FUZZY VALUE
1	0.56
2	0.56
6	0.13

BH-CL	
TID	FUZZY VALUE
1	0.13
2	0.13
6	0.13

CL-DL	
TID	FUZZY VALUE
1	0.56
4	0.13
6	0.13

CL-DM	
TID	FUZZY VALUE
1	0.56
4	1
6	0.13

CM-DL	
TID	FUZZY VALUE
1	0.56
4	0.13
6	0.56

CM-DM	
TID	FUZZY VALUE
1	0.56
4	0.13
6	0.56

CM-EM	
TID	FUZZY VALUE
2	0.56
3	0.56
7	0.56

BL-CH	
TID	FUZZY VALUE
3	0.13
5	0.56
6	0.13

BM-CH	
TID	FUZZY VALUE
3	0.56
5	0.56
6	0.13

AM-CM	
TID	FUZZY VALUE
1	0.56
3	0.13
4	0.13
5	0.13

BH-CM	
TID	FUZZY VALUE
1	0.13
2	0.13
3	0.13
6	0.13

BM-CM	
TID	FUZZY VALUE
1	0.56
2	0.56
3	0.56
5	0.13
6	1

BL-CM	
TID	FUZZY VALUE
1	0.13
2	0.13
3	0.13
5	0.13
6	0.13

Fig. 20. Fuzzy-tid-list after all row scans from dataset D

4.2.2 From Adjacency matrix Mining MFFIs using MFFPA-2 method

Using Fuzzy-tid-list, row-by-row extraction of MFFIs from the adjacency matrix (M) produced results above 4.2.1 points in this phase. Select the cell with a value greater than or equal to in the row of Adjacency matrix M. (min-support). They declared identified cell row-column combination as fuzzy 2-frequent itemsets (FL_2) and fetched Fuzzy-tid-list for identified cell. For the example in the first row in Fig 19 with headed AL, there is no cell with value >= Ø; here, min-support Ø=1. Scan the second row in which two cells, namely AM-BM and AM-DM, find which satisfies min-support threshold Ø so fetched Fuzzy-tid-list of AM-BM and AM-DM. Next, recursively create fuzzy k-frequent itemsets, such as FL_k (K>2), in a subsequent step by intersection-operating TIDs on FL_{k-1}. The binary search method can be used to find combined fast fuzzy lists. To create the Fuzzy-tid-list for k-frequent itemsets (k>2), existing FL_{k-1} Fuzzy-tid-list are combined. Elements in a newly created Fuzzy-tid-list are those with a common Tid in an existing Fuzzy-tid-list.

Only joining fuzzy itemsets that can create their superset directly from the adjacency matrix M will reduce the search space and candidate set. As found, FL_2 is from the second row AM-BM and AM-DM, so the subsequent possible superset is AM-BM-DM. If the BM-DM value from the BM row and DM column cell value satisfy the min-support threshold, generate the AM-BM-DM Fuzzy-tid-list by joining the AM-BM Fuzzy-tid-list and AM-DM Fuzzy-tid-list using the intersection operation on it. In the fifth row headed by BM, five cells satisfy Ø, so generated FL_2 from this row is BM-CL, BM-CM, BM-CH, BM-DL, and BM-DM. Next subsequent possible FL_3 are BM-CL-DL, BM-CL-DM, BM-CM-DL, BM-CM-DM, BM-CH-DL and BM-CH-DM. BM-CL-DL not join because of CL-DL value, which does not satisfy Ø, so ignore this set directly without generating its candidate itemsets or not join BM-CL-DL. This way drastically joins operation minimize or candidate itemsets, improving running time efficiency.

After reading all rows according to the algorithm-generated candidate Fuzzy-tid-list shown in Figure 21, many candidate sets are possible, but this approach generates only seven Tid-fuzzy sets out of these five fuzzy frequent itemsets of length 3. Next, it does not generate a candidate set for length 4, which know directly from the adjacency matrix. So using the MFPPA-2 method and Adjacency matrix generates fewer candidate sets than the state-of-art method.

AM-BM-DM		AH-BM-CM		AH-BM-CH		BM-CL-DM		BM-CM-DL		BM-CM-DM		BM-CM-EM	
TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE	TID	FUZZY VALUE
1	0.56	1	0.56	3	0.56	1	0.56	1	0.56	1	0.56	2	0.56
		3	0.56	5	0.56	6	0.13	6	0.56	6	0.56	3	0.56
		5	0.13										

Fig. 21. Fuzzy-tid-list after all row scans from Matrix M

4.2.3 Experimental Study and Performance Evaluation of MFFPA-2

Here, we contrast the MFFPA-2 performance of the recommended method with that of the list-based techniques put forward by Lin et al. [18] and EFM [19]. We Implement the proposed MFFPA-2, EFM, and Lin's method in Java. The outcomes are examined using two real datasets, chess and mushroom [32] and one artificial T10I4D100k dataset [32]. The quantities of objects in the datasets provide at random intervals between 1 and 7. The outcome of the experiment runtime, join count, and memory usage were all examined.

Runtime Analysis:

We implemented MFFPA-2, EFM [19], and Lin's [18] methods using the type-2 member function with 3-term fuzzy linguistic terms. To compare the execution time of the implemented methods, we have used different minimum support threshold values. Figures 22 through 24 show the findings of the execution running time evaluation on the chess dataset, mushroom dataset, and T10I4D100k dataset.

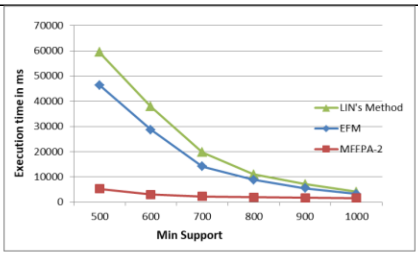

Fig. 22. Comparisons of execution times: Chess dataset

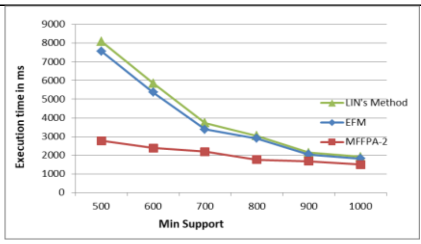

Fig. 23. Comparisons of execution times: Mushroom dataset

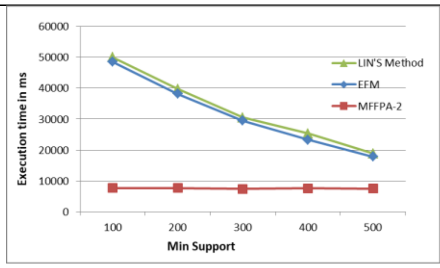

Fig.24. Comparisons of execution times: T10I4D100k dataset

From the results, it is observed that the proposed MFFPA-2 approach works better than the alternative method. A lower minimum support criterion also shows how resilient the MFFPA-2 technique is.

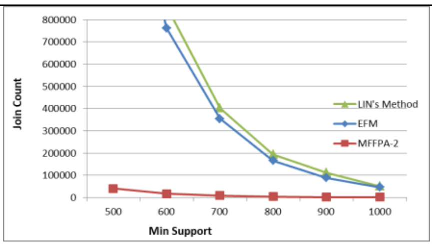

Fig. 25. Comparisons of Join Counts: Chess dataset

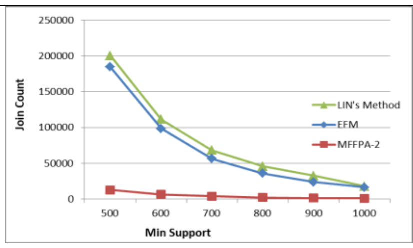

Fig. 26. Comparisons of Join Counts: Mushroom dataset

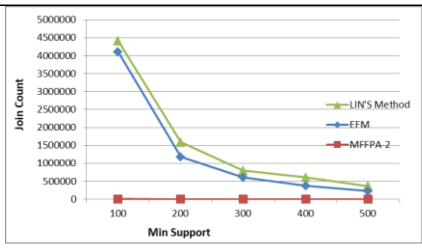

Fig. 27. Comparisons of Join Counts: T10I4D100k dataset

The number of Join Counts Analysis:

The number of joins made during the formation of MFFIs is considered while evaluating performance in this area. Figures 25 through 27 show the findings of the evaluations of the number of join counts on the chess, mushroom, and T10I4D100k datasets.

The results show that the MFFPA-2 method generates less join counts (candidate itemsets). It was noted that the MFFPA-2 method's join count performance is by far the most impressive. The proposed MFFPA-2 method produces fewer candidate itemsets than cutting-edge techniques.

Memory Utilization Analysis:

Here, effectiveness is measured by how extensively memory was used in the studies. Figures 28 through 30 show the results of the memory utilization on the chess dataset, the mushroom dataset, and the T10I4D100k dataset.

The results show that on the chess and mushroom datasets, the MFFPA-2 method utilizes less memory than the comparison strategy. It was noted that the MFFPA-2 method uses more memory than the compared approach on the artificial T10I4D100k dataset. We may deduce from additional trials with other datasets that in a particular case where a dataset has more than 1000 items, the suggested MFFPA-2 will need a more significant memory.

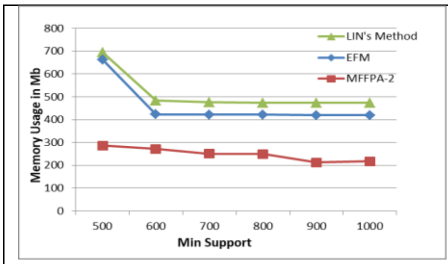

Fig. 28. Comparisons of Memory Usage: Chess dataset

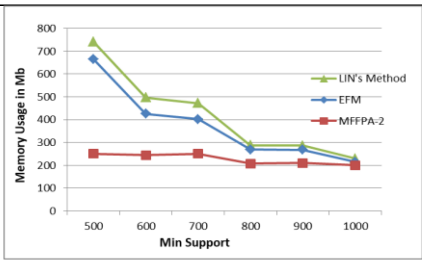

Fig. 29. Comparisons of Memory Usage: Mushroom dataset

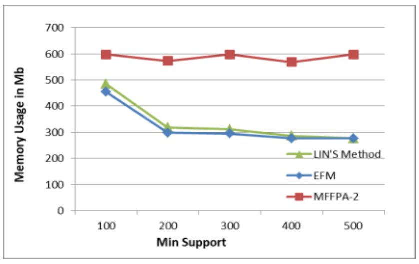

Fig. 30. Comparisons of Memory Usage: T10I4D100k dataset

5. Achievements concerning objectives

We did a detailed literature survey of fuzzy frequent itemsets mining and proposed two algorithms named AMFFI and MFFPA-2. Our proposed algorithm shows on an average 8% to 81% improvement in execution time compared to existing state-of-the-art methods for various dataset. The AMFFI improves execution time by 8% to 81% and node join count by 93% to 99%. The MFFPA-2 improves execution time by 38% to 75% and node join count by 93% to 99%.

6. Conclusion

Frequent itemsets mining is important in this era of growing e-commerce business. The performance of the frequent itemsets mining can be improved using fuzzy theory. Fuzzy Frequent itemsets mining can be improved regarding execution time and memory requirement by reducing the candidate itemsets and the number of database scans. With the growing demand for the identification of frequent itemsets, an efficient and optimal mining method is desirable. The proposed adjacency matrix-based Fuzzy frequent itemsets mining approach significantly reduces the candidate itemsets and scans the database only once. Our proposed approach called AMFFI and MFFPA-2 efficiently work in fuzzy frequent itemtsets mining. Our experimental analysis shows improvement of AMFFI Vs. MFFI-miner for different databases and reduce join counts by on an average 95%, and execution time by around 80% for most of the datasets. The proposed MFFPA-2 improves result against EFM for different databases and reduces join counts by 95% and execution time by 75% for most of the datasets.

7. Research publications

1. Patel, Mahendra Narottamdas, Sanjay M. Shah, and Suresh B. Patel. "An Adjacency matrix-based Multiple Fuzzy Frequent Itemsets mining (AMFFI) technique." International Journal of Intelligent Systems and Applications in Engineering 10, no. 1 (2022): 69–74. **(SCOPUS Approved, ISSN: 2147–6799)**

2. Patel, Mahendra N., S. M. Shah, and Suresh B. Patel. "An Efficient (MFFPA-2) Multiple Fuzzy Frequent Patterns Mining with Adjacency Matrix and Type-2 Member Function." In *International Conference on Advances in Computing and Data Sciences*, pp. 502-515. Cham: Springer Nature Switzerland, 2023.

3. Patel, Suresh B., Sanjay M. Shah, and Mahendra N. Patel. "An Efficient High Utility Itemset Mining Approach using Predicted Utility Co-exist Pruning." International Journal of Intelligent Systems and Applications in Engineering 10, no. 4 (2022): 224–230. **(SCOPUS Approved, ISSN: 2147–6799)**

4. Patel, Suresh B., Sanjay M. Shah, and Mahendra N. Patel. "An Efficient Search Space Exploration Technique for High Utility Itemset Mining." Procedia Computer Science 218 (2023): 937-948. (**SCOPUS Approved, ISSN: 1877-0509**)

8. References

1. R. Agrawal, T. Imielinski, and A. Swami, “Database Mining: A Performance Perspective,” 1993.
2. R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” 1994.
3. P. Berkhin, "A survey of clustering data mining techniques BT - Grouping Multidimensional Data," in Grouping Multidimensional Data, pp. 25–71, 2006.
4. M. Antonelli, P. Ducange, F. Marcelloni, and A. Segatori, "A novel associative classification model based on a fuzzy frequent pattern mining algorithm," Expert Syst. Appl., vol. 42, no. 4, pp. 2086–2097, Mar. 2015.
5. K. Hu, Y. Lu, L. Zhou, and C. Shi, "Integrating classification and association rule mining: A concept lattice framework," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1711, pp. 443–447, 1999.
6. J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," Data Min. Knowl. Discov., vol. 8, no. 1, pp. 53–87, 2004.
7. J. Friedman and L. A. Z. Fuzzy, “Similarity relations and fuzzy orderings,” North-Holland Publishing Company, 1968.
8. T. P. Hong, C. S. Kuo, and S. C. Chi, “Mining association rules from quantitative data,” Intell. Data Anal., vol. 3, no. 5, pp. 363–376, 1999.
9. C. W. Lin and T. P. Hong, “Mining fuzzy frequent itemsets based on UBFFP trees,” J. Intell. Fuzzy Syst., vol. 27, no. 1, pp. 535–548, 2014.
10. T.-P. Hong, C.-W. Lin, and T.-C. Lin, “THE MFFP-TREE FUZZY MINING ALGORITHM TO DISCOVER COMPLETE LINGUISTIC FREQUENT ITEMSETS,” Comput. Intell., vol. 30, no. 1, pp. 145–166, Feb. 2014.
11. J. C.-W. Lin, T.-P. Hong, and T.-C. Lin, “A CMFFP-tree algorithm to mine complete multiple fuzzy frequent itemsets,” Appl. Soft Comput., vol. 28, pp. 431–439, Mar. 2015.
12. J. C. W. Lin, T. P. Hong, T. C. Lin, and S. T. Pan, "An UBMFFP tree for mining multiple fuzzy frequent itemsets," Int. J. Uncertainty, Fuzziness Knowlege-Based Syst., vol. 23, no. 6, pp. 861–879, Dec. 2015.
13. J. C. W. Lin, T. Li, P. Fournier-Viger, and T. P. Hong, “A fast Algorithm for mining fuzzy frequent itemsets,” in Journal of Intelligent and Fuzzy Systems, vol. 29, no. 6, pp. 2373–2379 Nov. 2015.
14. J. C. W. Lin, T. Li, P. Fournier-Viger, T. P. Hong, J. M. T. Wu, and J. Zhan, “Efficient Mining of Multiple Fuzzy Frequent Itemsets,” Int. J. Fuzzy Syst., vol. 19, no. 4, pp. 1032–1040, Aug. 2017.
15. M. N. Patel, S. M. Shah, and S. B. Patel, “An Adjacency matrix-based Multiple Fuzzy Frequent Itemsets mining (AMFFI) technique,” Int. J. Intell. Syst. Appl. Eng., vol. 10, no. 1, pp. 69–74, 2022.
16. J. Mendel and R. John, “Type-2 fuzzy sets made easy,” IEEE Trans. fuzzy Syst., vol. 10, no. 2, pp. 117–127, 2002.
17. C.-H. Chen, T.-P. Hong, and Y. Li, Fuzzy Association Rule Mining with Type-2 Membership

Functions, vol. 9012. Cham: Springer International Publishing, 2015.

18. J. C.-W. Lin, X. Lv, P. Fournier-Viger, T.-Y. Wu, and T.-P. Hong, Efficient Mining of Fuzzy Frequent Itemsets with Type-2 Membership Functions, vol. 9622. Berlin, Hei-delberg: Springer Berlin Heidelberg, 2016.
19. J. C. W. Lin, J. Ming-Tai Wu, Y. Djenouri, G. Srivastava, and T. P. Hong, “Mining multiple fuzzy frequent patterns with compressed list structures,” IEEE Int. Conf. Fuzzy Syst., vol. 2020-July, 2020.
20. M. Delgado, N. Marín, D. Sánchez, and M. A. Vila, “Fuzzy association rules: General model and applications,” IEEE Trans. Fuzzy Syst., vol. 11, no. 2, pp. 214–225, Apr. 2003.
21. T. P. Hong, C. S. Kuo, and S. L. Wang, "A fuzzy AprioriTid mining algorithm with reduced computational time," Appl. Soft Comput. J., vol. 5, no. 1, pp. 1–10, Dec. 2004.
22. C. W. Lin, T. P. Hong, and W. H. Lu, "Linguistic data mining with fuzzy FP-trees," Expert Syst. Appl., vol. 37, no. 6, pp. 4560–4567, Jun. 2010.
23. W. H. L. Lin, Chun Wei, and Tzung Pei Hong, "An efficient tree-based fuzzy data mining approach," Int. J. Fuzzy Syst., vol. 12, no. 2, pp. 150–157, 2010.
24. T. P. Hong, C. W. Lin, and Y. L. Wu, “Incrementally fast updated frequent pattern trees,” Expert Syst. Appl., vol. 34, no. 4, pp. 2424–2435, May 2008.
25. H. Hagras, "Type-2 fuzzy logic controllers: A way forward for fuzzy systems in real-world environments," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. In-tell. Lect. Notes Bioinformatics), vol. 5050 LNCS, pp. 181–200, 2008.
26. O. Castillo and P. Melin, “Introduction to type-2 fuzzy logic,” Stud. Fuzziness Soft Comput., vol. 223, pp. 1–4, 2008.
27. Li, H., Zhang, Y., Hai, M., & Hu, H. Finding Fuzzy Close Frequent Itemsets from Databases. Procedia computer science, 139, 242-247 (2018).
28. S. Kar and M. M. J. Kabir, “Comparative analysis of mining fuzzy association rule using genetic algorithm,” The International Conference on Electrical, Computer and Communication Engineering, pp. 1–5, 2019.
29. D. K. Srivastava, B. Roychoudhury, and H. V. Samalia, “Fuzzy association rule mining for economic development indicators,” International Journal of Intelligent Enterprise, vol. 6(1), pp. 3–18, 2019.
30. L. Wang, Q. Ma, and J. Meng, “Incremental fuzzy association rule mining for classification and regression,” IEEE Access, vol. 7, pp. 121095–121110, 2019.
31. Hong, T.P., Lan, G.C., Lin, Y.H., Pan, S.T.: An effective gradual data-reduction strategy for fuzzy itemset mining. International Journal of Fuzzy Systems 15(2), 170–181 (2013).
32. “Frequent Itemset Mining Dataset Repository” http://fimi.ua.ac.be/ data