International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 16 (2018) pp. 12774-12782

CP Fuzzy Graph Based Approach for Mining Frequent Itemsets

S.Venkatesh[1], S.Sujatha[2]

[1]Department of Mathematics, JJ College of Engineering and Technology, Tiruchirappalli, Tamil Nadu, India.

[2]Department of Computer Applications, BIT campus, Anna University, Tiruchirappalli, Tiruchirappalli, Tamil Nadu, India.

Abstract

The vast majority of concentrates for mining frequent patterns depend on developing tree for arranging the items to mine frequent patterns. Numerous algorithms proposed as of late have been propelled by FP-growth patterns and utilizations a FP-tree to mine frequent patterns. This paper presents an algorithm called CP Fuzzy Graph which utilizes graph ratherthan tree to organize the items for mining frequent itemsets. The advantage of utilizing graph structure comes as space intricacy since graph utilizes an item a node precisely once as opposed to at least two times as was done in tree. The database outputs would thus be able to be significantly lessened with the assistance of the extra data. Exploratory outcomes additionally analyze the execution of the proposed approach both in the execution time and the quantity of tree nodes at two distinct quantities of regions,respectively.

Keywords: Fuzzy data mining, fuzzy set, Frequent pattern, CP-tree, CP-fuzzy graph

INTRODUCTION:

Contingent upon the different necessities of the mined information, association rule mining is the central method to locate the potential connections among the items from the binary databases. Agrawal et al initially introduced the Apriori algorithm to mine association rules in a level shrewd manner. Han et al at that point outlined the frequent pattern tree structure with a FP-development mining algorithm to discover frequent itemsets without candidate generation. All things considered, circumstances, it is hard to deal with the quantitative databases.In the past numerous algorithms have been created to discover frequent pattern itemsets. Hang et al expressed the fuzzy data mining way to deal with find fuzzy frequent itemsets in a level astute manner.

TheFP-tree development takes precisely two scans: The main output gathers the arrangement of frequent items and the second scan builds the FP-tree. Despite the fact that FP-tree can mine frequentitems by examining the database twice there is another algorithm called CP-tree,which mine the regular examples with one output. The fundamental thought in CP-tree is to periodicallyreorganize the prefix-tree in visit subordinate thing request after insertingsome exchanges into the prefix-tree utilizing past thing request. Through repeated reorganization, CP-tree gives better mining exhibitions subsequently and it alsosupports both incremental and intuitive mining

The Fuzzy graph model has fascinated plenty of responsiveness who works on data mining from the societies of data science and fuzzy logic. In this Paper, we design a novel technique on CP-Fuzzy Graph to mine we design a novel methodology on CP- Fuzzy Graph to mine frequent patterns from quantitative databases.

RELATED WORK:

In 1993, the principal known frequent pattern mining algorithm is Apriori, which was proposed by Agrawal [1]. From that point forward different algorithms have been proposed forimproving the execution of Apriori-based algorithm. This sort of methodologies requires multi database scan and countless sets are produced to find frequent patterns. In Apriori, execution issues, for example, memory space and executiontime is observed to be high. This limits utilization of these techniques for incremental, interactiveand information stream mining. This issue can be dealt with by utilizing frequent pattern tree algorithm proposed by Han et al. [5], which mine frequent patterns with two scans and eliminate the generation of candidate sets. Among tree based methodologies, the very minimal prefix tree structure called FP-tree is another approach for mining frequent patterns.

While mining incessant items inCATS-tree, one have to movement both upward and descending heading. Kai-Sang Leung[7] proposed the CAN-tree algorithm, it build prefix tree in authoritative request withone database check. Since, it is orchestrated in sanctioned request, frequent diving order of items isn't kept up and this outcome in high mining time contrasted with FPtree.To settle the disadvantage of CAN-tree, the Compact Pattern (CP-tree) [8] was proposed by Syed Tanbeer, which mines the continuous examples with one output.

Anurag Choubey [2] proposed a graph structure that catches just those itemsets that requirements to characterize an adequately huge dataset into a sub lattice speaking to imperative weights and does not allow to exceptions. They have concocted a procedure that spreads noteworthy realities of information by penetrating down the extensive information into a compact type of an Adjacency Matrix at various phases of mining process. The graph structure is designed to the point that it can be effortlessly kept up and the exchange off in compacting the substantial information esteems is diminished. They have demonstrated that graph based approach is quicker than the partition algorithm.

PRELIMINARIES AND PROBLEM STATEMENT

“Let QD be a quantitative data base with n be the number of transactions. Let $I = \{i_1, i_2, i_3, \dots, i_m\}$be a finite set of m distinct items (attributes) in a quantitative database.$QD = \{T_1, T_2, \dots. T_n\}$, where each transaction $T_q \in QD$is a subset of I, contains several items with its purchase quantities and has an unique identifier, called *TID*. An itemset X is a set of k distinct items $\{i_1, i_2, \dots, i_k\}$, where k is the length of an itemset called k-itemset. An itemset X is said to be contained in a transaction $T_q$if $X \subseteq T_q$. A minimum support threshold is defined as δ. The user-specified membership functions is set as μ.”

“The linguistic variable $R_i$is an attribute of a quantitative database whose value is the set of fuzzy terms represented in natural language as $(R_{i1}, R_{i2}, \dots, R_{ih})$ and can be defined in the membership functions μ.The quantitative value of i denoted as v_{iq}, is the quantitative of the item i in transaction T_q.The fuzzy set, denoted as f_{iq}, is the set of fuzzy terms with their membership degrees (fuzzy values) transformed from the quantitative
value $v_{iq}$of the linguistic variable i by the membership functions μ as:

$$f_{iq} = \mu_i(v_{iq}) \left(= \frac{fv_{iq1}}{R_{i1}} + \frac{fv_{iq2}}{R_{i2}} + \cdots \dots + \frac{fv_{iqh}}{R_{ih}} \right)$$

where h is the number of fuzzy terms of i transformed by μ, $R_{il}$is the l-th fuzzy terms of i, $fv_{iql}$is the membership degree (fuzzy value) of $v_{iq}$of i in the l-th fuzzy terms $R_{il}$and fv_{iql}⊆[0, 1].”

“The support of the transformed fuzzy terms, denoted *sup* (R_{il}), is the summation of scalar cardinality of the fuzzy values of fuzzy term R_{il}, which can be defined as

$$sup(R_{il}) = \sum_{R_{il} \subseteq T_q \wedge T_q \in QD'} fv_{iql}$$

where$QD'$is the quantitative database QD transformed by membership functions (= μ).”

Problem Statement

The problem of mining frequent itemset by using Compact pattern Fuzzy Graph (CPFG) approach to discover frequent patterns with the help of fuzzy graph edge density values.

THE PROPOSED CP-FUZZY GRAPH APPROACH

This proposed approach can be divided into four parts: In first part we make graph for the given database QD. In second partquantitative data set is transformed into fuzzy set. In third part graphis pruned to remove all non-frequent nodes. Finally in the fourth part frequent itemsets are mined from the pruned graph using Edge density.

Table 1.1: Quantitative Data Base

TID	Items with quantity
1	A:5, C:10, D:2, E:9
2	A:8, B:2, C:3
3	B:3, C:9
4	A:7, C:9, D:3
5	A:5, B:2, C:5
6	A:3, C:10, D:2, E:2

Construction Phase of proposed CP-Graph

The CP- Graph consist of nodes and edges. Number of node in the graph is equal to number of distinct items in the database QD. Each node is associated with a value count which stores the number of occurrence of item in QD.

This construction phase of proposed CP-Graph clarifies the arrangement of graph for exchange datasets. Each edge in a graph contains three qualities set apart on it. We begin with first transaction of database and we take first time of exchange and influence a node with this thing to name and set its tally to 1.Then we make a node for second thing and draw an edge between these two nodes. Each time we make a chain of nodes in a single transaction, the estimation of node tally is expanded by one for the progressive edges of graph. While influencing graph for first exchange in database we to make node A and C at that point draw an edge amongst them and check it with 1A2. Presently make a node D and connection it with C by an edge set apart by 1A3.This process proceeds till all transaction of data base. The process of making graph from the table 1.1 of quantitative data set is shown in Figure 1.2

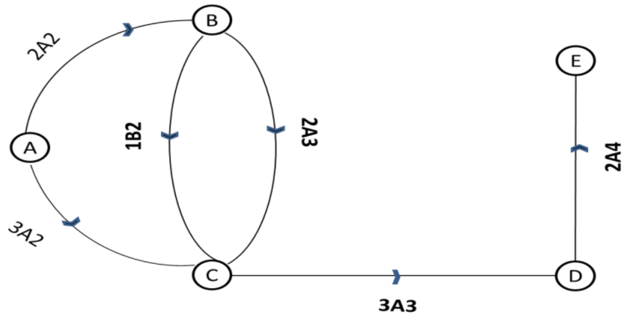


Figure 1.2: Graph for the Transaction

Algorithm for Graph Construction

Input: All transaction T in database QD.

Output: CP-Graph.

Method:

1. Read DBQD, N and set j=1.
2. Create the node for each itemset
3. while (j<=N){
4. Scan the transaction t_j once.
5. Let the transaction be [A/Q], where A is the first element and Q is the remaining element in the transaction, then call construct_Graph ([A/Q], G).
6. Fun construct_Graph([A/Q], G){
7. If $(\text{previous} \neq \text{null})$ Create edge $(\text{previous } \mathrm{I_j}, 1, \text{first node count})$then $\text{Create}(\mathrm{I_j}) = \text{count}(\mathrm{I_j}) + 1$then $\text{Create}(\mathrm{I_j}) = \text{count}(\mathrm{I_j}) + 1$.
8. If $(\text{previous} = \text{null})$

   If $\left(\begin{matrix}\text{an edge with same source , destinatin start node, no fo node does not}\\ \text{exists in all edges generated so for create edge(previous, } \mathrm{I_j}1, \text{first node count}\end{matrix}\right)$

   $\text{thenEdge frequency} = \text{edge frequency} = 1;$
9. else
10. $\text{previous} = \mathrm{I_j}$,first=null, previous=null
11. Increment j variable }.

Once, the graph is constructed by using all the transactions, it is necessary to prune the graph with its frequency of occurrences. For pruning the graph, the transformation phase of quantitative database is needed. The transformation of quantitative database into fuzzy set is explained in the next section.

Transformation Phase:

Table 1.1 consists of 6 transactions and 5 items, which are respectively denoted as (A) to (E). The minimum support threshold is initially set as δ (= 30%).

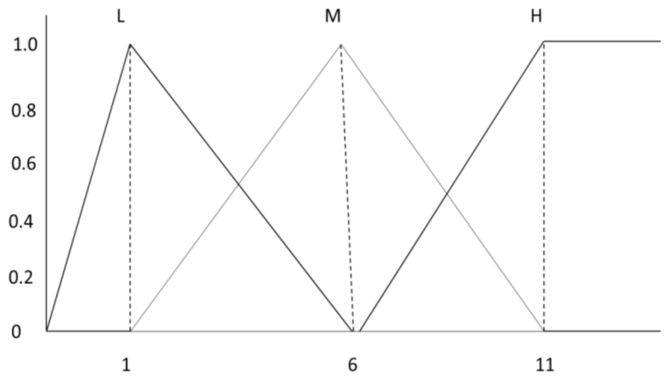


Figure 1.3: The Membership function of Fuzzy Linguistic 3-terms

All the items in the table 1.1 is used in the membership functions of Fuzzy Linguistic 3-terms

STEP 1: convertingthe quantitative values of all the items in the transactions is represented as fuzzy sets using the given membership functions

Table 1.4 Quantitative Database and Transformed Fuzzy Set

TID	Items
1	$\left(\frac{0.2}{A.L} + \frac{0.8}{A.M}\right), \left(\frac{0.2}{C.M} + \frac{0.8}{C.H}\right), \left(\frac{0.8}{D.L} + \frac{0.2}{D.M}\right), \left(\frac{0.4}{E.M} + \frac{0.6}{E.H}\right)$
2	$\left(\frac{0.6}{A.M} + \frac{0.4}{A.H}\right), \left(\frac{0.8}{B.L} + \frac{0.2}{B.M}\right), \left(\frac{0.6}{C.L} + \frac{0.4}{C.M}\right)$
3	$\left(\frac{0.6}{B.L} + \frac{0.4}{B.M}\right), \left(\frac{0.4}{C.M} + \frac{0.6}{C.H}\right)$
4	$\left(\frac{0.8}{A.M} + \frac{0.2}{A.H}\right), \left(\frac{0.4}{C.M} + \frac{0.6}{C.H}\right), \left(\frac{0.6}{D.L} + \frac{0.4}{D.M}\right)$
5	$\left(\frac{0.2}{A.L} + \frac{0.8}{A.M}\right), \left(\frac{0.8}{B.L} + \frac{0.2}{B.M}\right), \left(\frac{0.2}{C.L} + \frac{0.8}{C.M}\right)$
6	$\left(\frac{0.6}{A.L} + \frac{0.4}{A.M}\right), \left(\frac{0.2}{C.M} + \frac{0.8}{C.H}\right), \left(\frac{0.8}{D.L} + \frac{0.2}{D.M}\right), \left(\frac{0.8}{E.L} + \frac{0.2}{E.H}\right)$

STEP 2:

The scalar cardinality of each fuzzy region in all the transactions is calculated as the *count* value of the region.

Table 1.5: Scalar Cardinality of itemset

Item	count
A.low	1.0
A.middle	3.4
A.high	0.6
B.low	2.2
B.middle	0.8
B.high	0.0
C.low	0.8
C.middle	2.4
C.high	2.8
D..low	2.2
D.middle	0.8
D.high	0.0
E.low	0.8
E.middle	0.6
E.high	0.1

STEP 3: The fuzzy region with the maximum count among the three possible regions of each item is found.

Table 1.6: Fuzzy Region with Maximum Count

Item	count
A.middle	3.4
B.low	2.2
C.high	2.8
D..low	2.2
E.low	0.8

STEP 4: The counts of the fuzzy regions selected in STEP 3 are checked against the predefined minimum count, which is 6*30% (= 1.8).

Table 1.7: Fuzzy Region with Minimum Support Count

Item	count
A.middle	3.4
B.low	2.2
C.high	2.8
D.low	2.2

STEP 5: The occurrence number of each fuzzy region in $L1$ is also calculated. For example, the fuzzy region

Table 1.8: Fuzzy Region with L1 set

Item	count	Occurrence
A.middle	3.4	5
B.low	2.2	3
C.high	2.8	4
D..low	2.2	3

Now, the pruned value of the fuzzy itemset are considered as node of the fuzzy graph..

Consider a fuzzy graph $G = (V, E)$ such that

$$V = \{A.middle = v_1,\ B.low = v_2, C.high = v_3, D.low = v_4\}$$

and$E = \{(v_1, v_2), (v_1, v_3), (v_2, v_3)(v_3, v_4)\}$

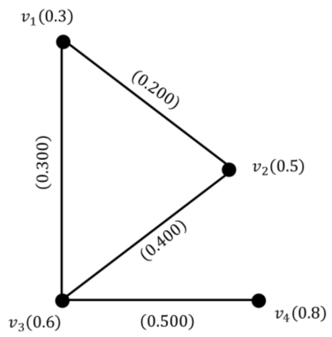


Figure 1.9: Fuzzy Graph with pruned Itemsets

Finding Frequent Item Set Using Edge Density

Let $G = (V, E)$ be a fuzzy graph. Then the edge density of an edge e of G is $ed(e) = \frac{2\mu(u,v)}{\sum(\sigma(u)\wedge\sigma(v))}$ for every $u, v \in V$ and $0 \leq ed(e) \leq 2$. with edge density on its each edge is called edge density fuzzy graph of G and is denoted by $ED(G)$.

Weight of the fuzzy cut: Let $ED(G)$ be an edge density fuzzy graph and $\{v_i, \overline{v_i}\}$ be a partition of its vertex set V. The of edges joining vertices of V_i and vertices of $\overline{V_i}$ is called a cut-set of $ED(G)$ and is denoted by E_i. The weight of the cut set $(V_i, \overline{V_i})$is defined to be

$$\sum_{e_{ij} \in E_i} \mu(e_{ij})$$

Edge density connectivity: Let $ED(G)$ be an edge density fuzzy graph G. The edge density connectivity of $ED(G)$, denoted by $\lambda(ED(G))$ is defined to be the minimum weight of cut sets of $ED(G)$. The edge density fuzzy graphs $ED(G)$ is called $\tau -$edge connected if $ED(G)$ is connected and $\lambda(ED(G)) \geq \tau$. A τ-edge component of $ED(G)$ is a maximal τ -edge connected subgraph of ED(G).

Minimal slicing: “A slicing of an edge density fuzzy graph $ED(G)$ is minimal slicing, if there is no subpartition which is slicing of $ED(G)$. Each member of the slicing is called minimum cut set slicing and also a slicing of an edge density fuzzy graph $ED(G)$ is a narrow slicing of $ED(G)$,if each cut C_i is a minimal cut of some component of $ED(G) \backslash \bigcup_{j=1}^{i-1} C_j$ “

Step 1 : The edge density of an edge e of G is $ed(e) = \frac{2\mu(u,v)}{\sum(\sigma(u)\wedge\sigma(v))}$ for every $u, v \in V$ and $0 \leq ed(e) \leq 2$.

Step 2: The edge density fuzzy matrix of an edge density fuzzy graph $ED(G)$is denoted by

$$ED(G) = [d_{eij}]$$

$$d_{eij} = \begin{cases} \frac{2\mu(v_i, v_j)}{\sum(\sigma(v_i) \wedge \sigma(v_i))} & for\ v_i \neq v_j \\ 0, & othere\ wise \end{cases}$$

An edge density fuzzy matrix $ED_M(G) = \begin{pmatrix} 0.000 & 0.235 & 0.353 & 0.000 \\ 0.235 & 0.000 & 0.471 & 0.000 \\ 0.353 & 0.471 & 0.000 & 0.588 \\ 0.000 & 0.000 & 0.588 & 0.000 \end{pmatrix}$

Step 3 : Sum along each row,

v_1	v_2	v_3	v_4
0.588	0.716	1.412	0.588

The minimum value occurs at row v_1. So weget $C_1 = (\{v_1, v_4\}, \{v_2, v_3, \})$.

Let $ED(G_1)$ be an edge density fuzzy subgraph of $ED(G)$induced by the remaining vertices.

Then the edge density fuzzy matrix $ED_M(G_1) = \begin{pmatrix} 0.000 & 0.471 \\ 0.471 & 0.000 \end{pmatrix}$

Computing the sum along each row

v_2	v_3
0.471	0.471

The minimum value occurs at v_2 *and* v_3. So we get

$$C_2 = (\{v_2, v_3\})$$

Therefore, frequent 2-itemset is $\{v_2, v_3\}$ie,$\{B, C, \}$ and $\{v_1, v_4\}$ that is $\{A, D\}$

For the given Quantitative Database, there are two frequent items occurs, 1, $\{B, C\}$and $\{A, D\}$.

EXPERIMENTAL RESULTS

In this section, we present the performance of the proposed CP Fuzzy Graph on frequent patterns mining from the datasets. All the experiments have been performed in computer with Intel(R) Core(TM) 2DUO 3GHz CPU and 1.96 GB RAM. We implement our proposed algorithm and the CP-fuzzy Graph algorithm in java. For evaluation, we have used one dense benchmark datasets such as MUSHROOM [11] and one sparse dataset such as T10I4D100K[11].

In CP fuzzy graph the user supplies a value for "n", which represents the number of transactions after restructuring phase to be done. In our work we choose the following values for the execution purpose.

Data	**No. of Transaction (n)**
Mushroom	1000
T10I4D100K	25,000

In each graph x-axis shows vary in minimum support value (%) in ascending order and y-axis shows the execution time represent in seconds.

Total Processing time for Mushroom Data		
Support (%)	**CP- Tree**	**CP-Fuzzy Graph**
5	31.82	28.79
15	1.03	0.92
30	0.187	0.173
50	0.109	0.102
70	0.097	0.092

<u>Trend Analysis for CP-Tree and CP=Fuzzy Graph using Mushroom Data</u>

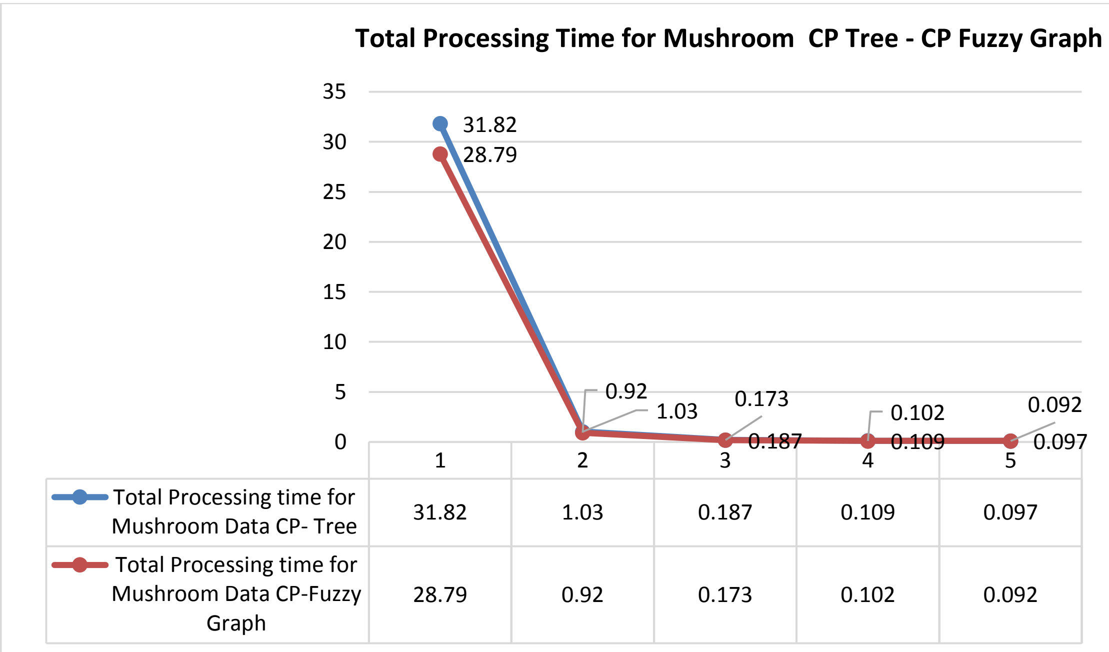


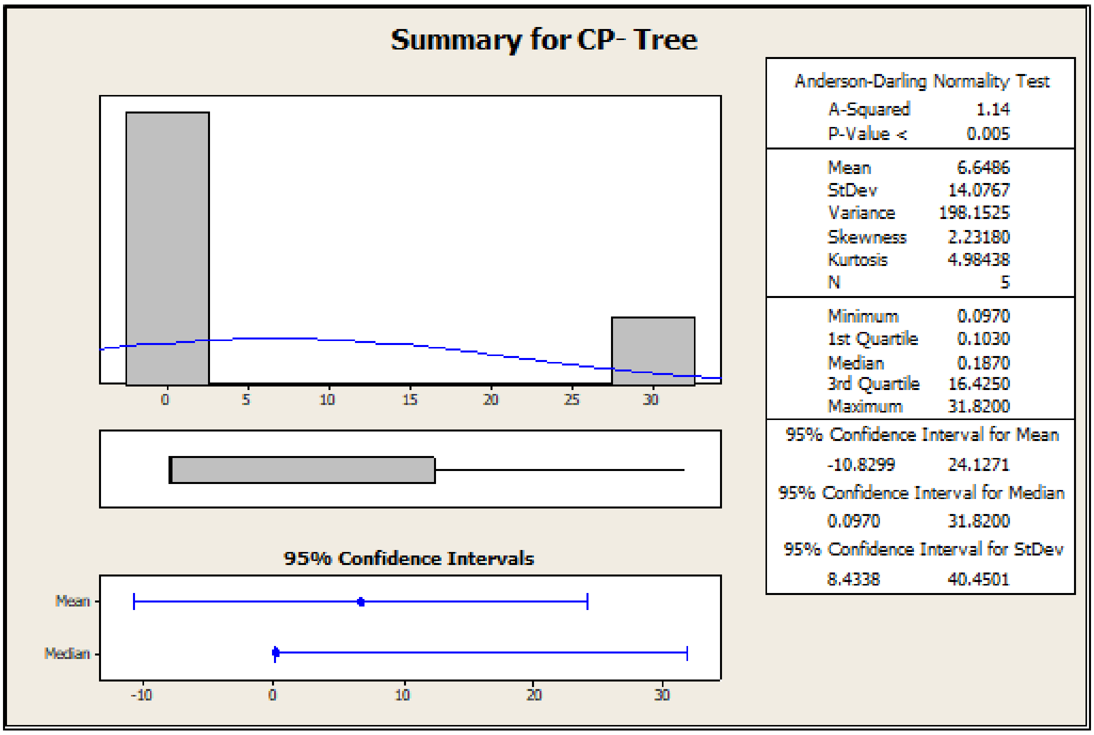
Table – Comparing the CP Tree and CP Fuzzy Graph

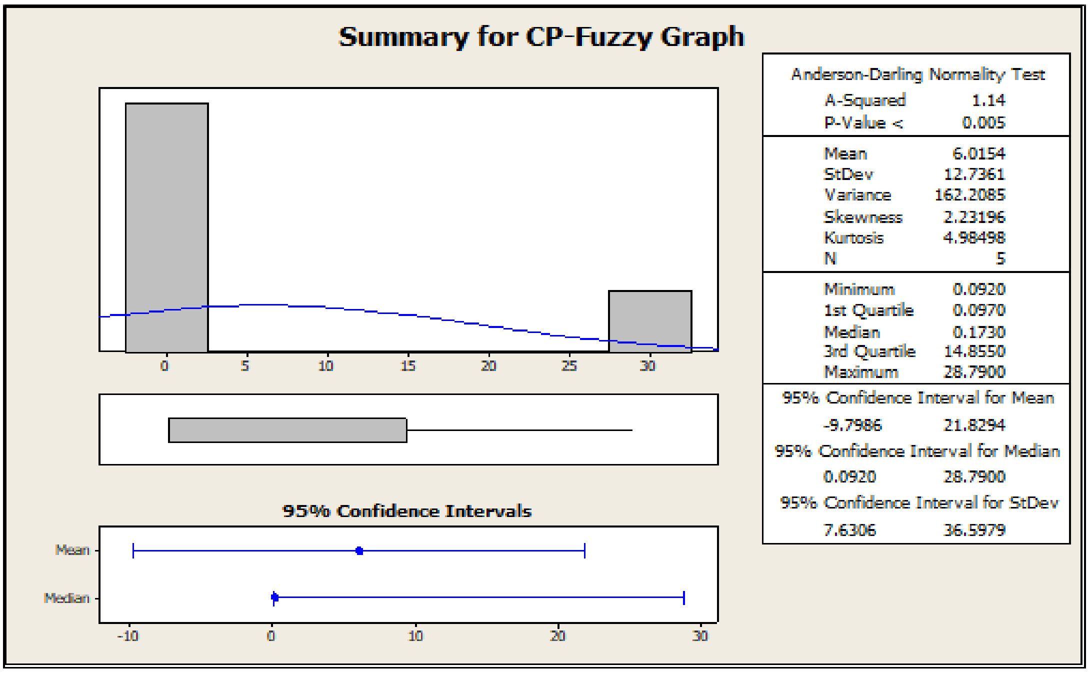
International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 16 (2018) pp. 12774-12782

The comparison is made between the CP-Tree and the CP-Fuzzy Graph analysis for the Mushroom data set. We found the shift in the mean for 2 data set namely the support percentage of 5 and 15 when compared to others.

Descriptive Statistical Measure for CP-Tree and CP-Fuzzy Graph for Mushroom Data

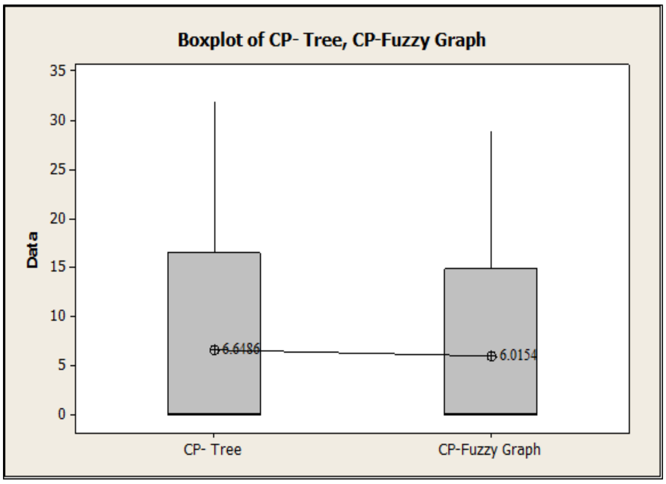
The descriptive statistical measure was used the study impact of the CP-Tree and CP-Fuzzy Graph. It has understood the location and spread for the Mushroom data set.





Box Plot Analysisfor Mushroom Data

The Box plot analysis used to study the variation among the CP-Tree and the CP-Fuzzy Graph. It is found that CP-Tree has a mean value of 6.65 when compared to the CP-Fuzzy Graph with the mean value of 6.02.



Statistical Hypothesis for the CP – Tree and CP-Fuzzy Graph for Mushroom Data

The student t-test is used to study the significant difference between the CP-Tree and CP-Fuzzy Graph.

Null Hypothesis (H_0) - There is no significant difference between the CP-Tree and CP-Fuzzy Graph for Mushroom data,

Alternative Hypothesis (H1) - There is no significant difference between the CP-Tree and CP-Fuzzy Graph for Mushroom data,

Level of Significant – 5%

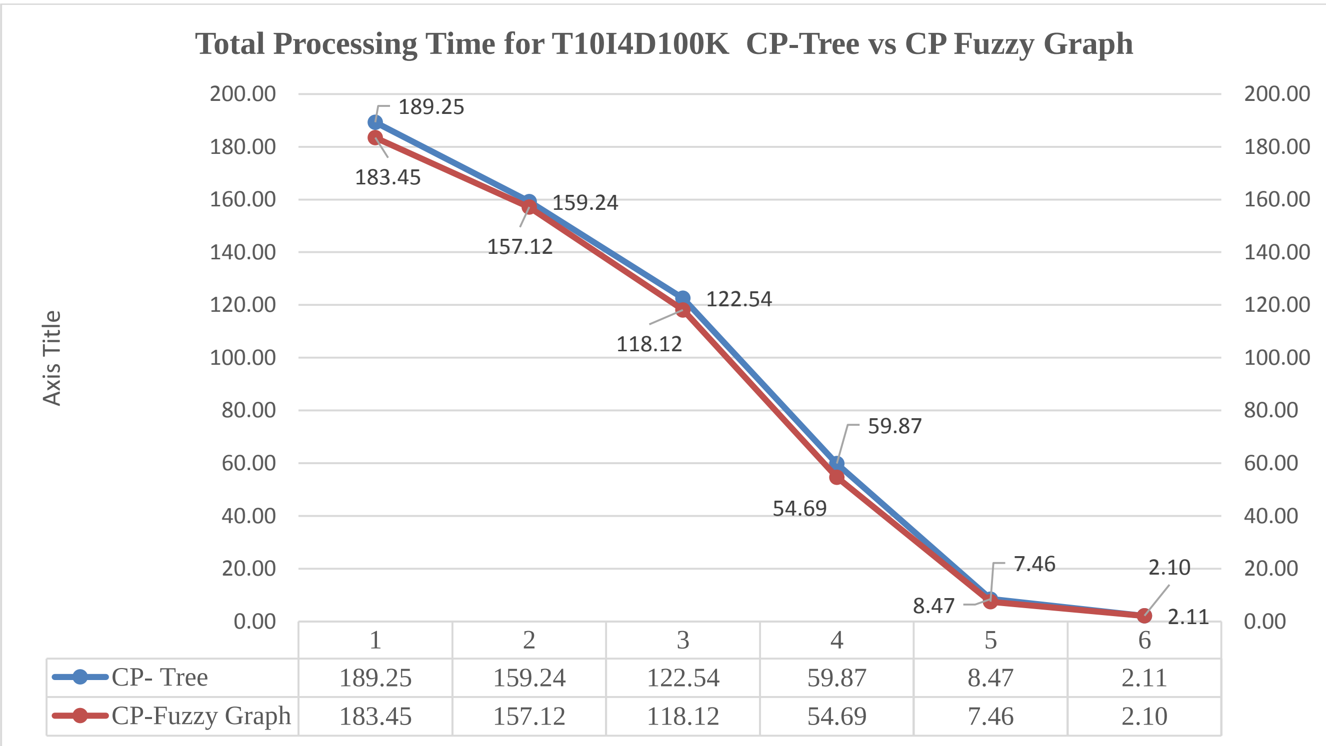
Two-Sample T-Test and CI: CP- Tree, CP-Fuzzy Graph

```
Two-sample T for CP- Tree vs CP-Fuzzy Graph

               N  Mean  StDev  SE Mean
CP- Tree       5   6.6   14.1      6.3
CP-Fuzzy Graph 5   6.0   12.7      5.7


Difference = mu (CP- Tree) - mu (CP-Fuzzy Graph)
Estimate for difference:  0.633200
95% CI for difference:  (-19.441362, 20.707762)
T-Test of difference = 0 (vs not =): T-Value = 0.07  P-Value = 0.943  DF = 7
```

Inference – Since the p-value is not low i.e. p-value 0.07, we accept the alternative hypothesis, and conclude that there is significant difference between the CP-Tree and CP-Fuzzy Graph for the Mushroom data.

T10I4D100K		
Support (%)	CP- Tree	CP-Fuzzy Graph
30	189.253	183.452
50	159.235	157.124
60	122.541	118.123
70	59.873	54.694
80	8.468	7.458
90	2.109	2.101

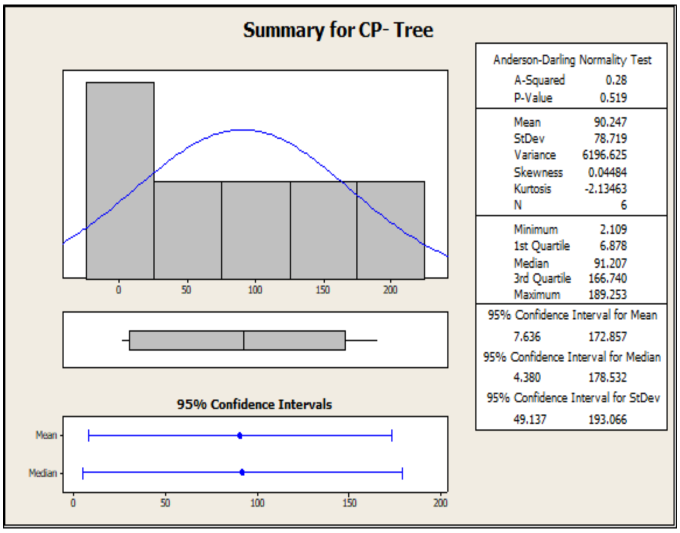
Trend Analysis for CP-Tree and CP=Fuzzy Graph using T10I4D100K Data

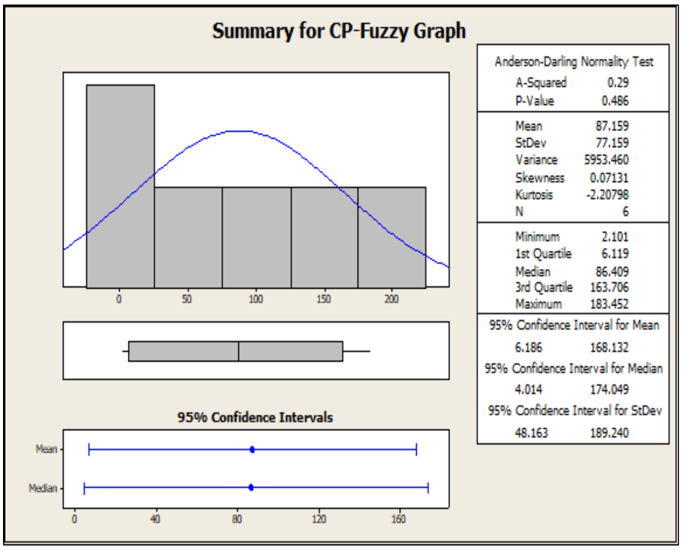


The comparison is made between the CP-Tree and the CP-Fuzzy Graph analysis for the T10I4D100K data set. We found the shift in the mean for all the data sets. For the support percentage of 30 and 60 shows the maximum difference.

Descriptive Statistical Measure for CP-Tree and CP-Fuzzy Graph for T10I4D100K Data

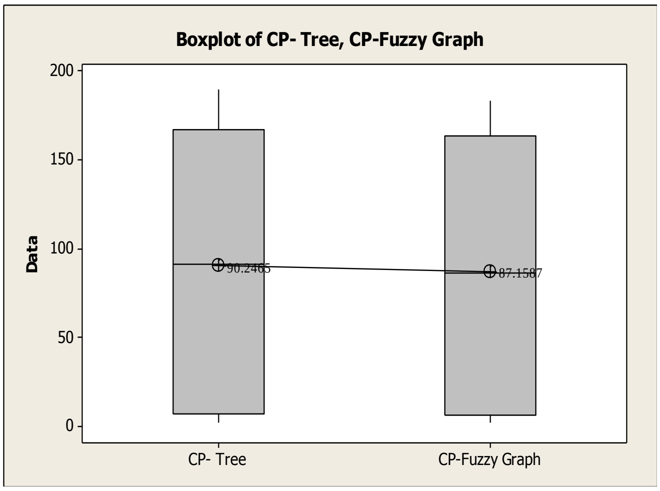
The descriptive statistical measure was used the study impact of the CP-Tree and CP-Fuzzy Graph. It has understood the location and spread for the T10I4D100K data set.





Box Plot Analysis for the T10I4D100K Data

The Box plot analysis used to study the variation among the CP-Tree and the CP-Fuzzy Graph. It is found that CP-Tree has a mean value of 90.2 when compared to the CP-Fuzzy Graph with the mean value of 87.1.



Statistical Hypothesis for the CP – Tree and CP-Fuzzy Graph for theT10I4D100K Data

The student t-test is used to study the significant difference between the CP-Tree and CP-Fuzzy Graph.

Null Hypothesis (H_0) - There is no significant difference between the CP-Tree and CP-Fuzzy Graph for the T10I4D100K data.

Alternative Hypothesis (H1) - There is no significant difference between the CP-Tree and CP-Fuzzy Graph for the T10I4D100K data.

Level of Significant – 5%

Two-Sample T-Test and CI: CP- Tree, CP-Fuzzy Graph

```
Two-sample T for CP- Tree vs CP-Fuzzy Graph

                                  SE
               N  Mean  StDev   Mean
CP- Tree       6  90.2   78.7     32
CP-Fuzzy Graph 6  87.2   77.2     31


Difference = mu (CP- Tree) - mu (CP-Fuzzy Graph)
Estimate for difference:  3.08783
95% CI for difference:  (-98.70960, 104.88526)
T-Test of difference = 0 (vs not =): T-Value = 0.07  P-Value = 0.947  DF = 9
```

Inference – Since the p-value is not low i.e. p-value 0.94, we accept the alternative hypothesis, and conclude that there is significant difference between the CP-Tree and CP-Fuzzy Graph for the theT10I4D100K Data.

We can see that the execution time of our proposed CP- Fuzzy Graph is low comparing to the execution time of the CP-tree for mining frequent patterns. While the minimum support values increases gradually the execution time for mining patterns is decreases.

This is due to the fact that, for lower support the frequent patterns are more, hence it requires high mining times that increase the rate of change in overall runtime. In contrary, for high minimum support the frequent patterns to mine are less, hence it requires low mining time.

CONCLUSION

In this paper CP Fuzzy Graph is proposed to find mining frequent itemsets. The benefit of using graph structure is greatly reduces the space complexity. . We have evaluated the performance of the proposed CP Fuzzy Graph on benchmark databases such as MUSHROOM and T10I4D100K. It is noticed that the time taken by the proposed fuzzy graph method for extracting frequent item is encouraging compared to the other relevantly proposed methods.

REFERENCES

[1] Agrawal, R., Imielinski, T., Swami, A.1993, Mining association rules between sets of items inlarge databases. In: Proceedings. ACM-SIGMOD International Conference on Managementof Data (SIGMOD), Washington, DC, pp. 207–216 .

[2] Anurag Choubey, RavindraPatil&Rana, JL 2012, ‘Graph Based New Approach for Frequent Pattern Mining’, International Journal of Computer Science & Information Technology, vol. 4, pp. 221-235.

[3] Cheung, W., Za, O.R, 2003, Incremental mining of frequent patterns without candidate generation or support constraint. In: Proceedings of the Seventh International Database Engineering and Applications Symposium, IDEAS 2003.

[4] Han, J., Pei, J., Yin, Y.,2000 Mining frequent patterns without candidate generation. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1– 12 .

[5] Han, J, Pei, J, Yin, Y & Mao, R 2004, ‘Mining frequent patterns without candidate generation: a frequent-pattern tree approach’, Data mining and Knowledge Discovery, vol. 8, no. 1, pp. 53-87.

[6] Hong, T.-P., Lin, C.-W., Wu, Y.-L. 2008, Incrementally fast updated frequent pattern trees.Expert Systems with Applications 34(4), 2424–2435.

[7] Karunambigai.M.G., Muhammed Akram., Sivasankar Shanmugam., Palanivel Kasilingam., 2016, “Clustering Algorithm for Intuitionistic Fuzzy Graphs”, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems,Vol.25, No.3,367-383.

[8] Leung, C.K., Khan, Q.I., Li, Z., Hoque, T., 2007, CanTree: a canonical-order tree for incremental frequent-pattern mining. Knowledge and Information Systems 11(3), 287–311.

[9] Tanbeer, S.K., Ahmed, C.F., Jeong, B.-S., Lee, Y.-K., 2008, Efficient single-pass frequent pattern mining using a prefix-tree.Information Sciences 179, 559–583.

[10] R .Vishnu Priya, A.Vadivel, and R.S. Thakur, 2010., “Frequent Pattern Mining Using Modified CP-Tree for Knowledge Discovery”**,**ADMA , Part I, LNCS 6440, pp. 254–261.

[11] VivekTiwari, VipinTiwari, Shailendra Gupta &RinuTiwari 2010, ‘Association Rule Mining : A Graph Based Approach for Mining Frequent Item sets’, International Conference on Networking and Information Technology, IEEE, pp. 309-312.

[12] http://fimi.cs.helsinki.fi/data