International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 02 | May-2015 www.irjet.net p-ISSN: 2395-0072

A Survey of Frequent Subgraph Mining algorithms for Uncertain Graph Data

S. V. S. Santhi[1], Dr.P. Padmaja[2],

[1] *Assistant professor, CSE Department , PRIME College, Vizianagaram, Andhra Pradesh, India*

[2] *Associate professor, Department of Information technology, GITAM University, Andhra Pradesh, India*

---***---

Abstract - *Mining uncertain graph data is different from exact graph data semantically and is more challenging task in recent days of research. Mining frequent sub graphs from uncertain graph data is the latest topic in research. Few algorithms for finding frequent sub graphs from uncertain graph data are Mining Uncertain Subgraph patterns (MUSE), Weighted MUSE (WMUSE), UGRAP Index(UGRAP), Mining Uncertain Subgraph patterns under Probabilistic semantics (MUSE-P). In these algorithms MUSE is the first algorithm proposed to find the frequent sub graphs from uncertain graph data. All the three algorithms MUSE, WMUSE, UGRAP are proposed under expected semantics where as MUSE-P is the algorithm proposed under probabilistic semantics. In this paper we are presenting a survey of various algorithms for mining frequent sub graphs from uncertain graph data. This survey is to help the user in real world applications of various domains.*

Key Words : Uncertain graph data, Frequent subgraphs , Apriori, Isomorphism, Pattern growth.

1. INTRODUCTION

Mining large amount of data using Data Mining is popular in these days. This is due to availability of data in large amounts and using the hidden data into useful information. Graph mining became latest research topic in Data Mining[1]. Graph mining attracted many researchers due to the scope of large graphs that are applicable in numerous applications and domains. Graph mining is structured data mining extracting useful information from semi structured datasets.

In real world, graph data are not precise and complete, they are incomplete, noise and inaccurate due to uncertainties. Graphs which are subjected to uncertainties are called uncertain graphs. Uncertain graphs are generated in the form of exact graphs in which each edge its associated with a probability which indicates that the edge exists. Edge probabilities are used to express the uncertainty and are used to represent the connectivity between nodes. Many recent researches generalizing the exact graph problems to uncertain graphs. Whereas previous studies of graph mining is only on exact graphs which are precise and complete. Uncertainties are associated with both edges and vertices, whereas in expected semantics uncertainties with edges are only considered. An uncertain graph is a special edge weighted graph, where the weight on each edge is known as existence probability which means the probability exists between the vertices u and v. For a K-edge uncertain graph, 2^K sub graph isomorphism tests are required.

Mining frequent sub graph patterns in a graph database is a challenging and important problem. Mining frequent sub graph patterns from an uncertain graph Database is different from exact graph Database. In exact graph Database, support of the sub graph pattern is used to measure the significance. The significance of sub graph S in uncertain graph Database D can be measured by the expected value of the supports of S in all implicated graph Database of D.

2. OVERVIEW OF GRAPH REPRESENTATION

2.1 Exact Graph:

An exact graph is a tuple $G=(V,E,\Sigma, L)$ where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, Σ is a set of labels and $L: V \cup E \rightarrow \Sigma$ is a function assigning labels to vertices and edges. The vertex set of Graph G is V(G) and edge set E(G).The size of the graph is |E(G)| i.e the number of edges it contains.

2.2 Uncertain graph:

A graph which is uncertain is a tuple $G^P=(V,E,\Sigma,L,P)$ where $G=(V,E,\Sigma, L)$ are like an exact graph definition and P is a function for assigning edge probabilities 0 to 1 to each edge known as existing probabilities. An uncertain graph G^P implicates a set of $2^{|E|}$ possible exact graphs. $P(G^P \Rightarrow G)$ means the probability of G being implied by G^P The probability of each edge of G^P

which are included or excluded from graph G is calculated as

$$P(G^P => G) = \prod_{e \in E(G)} P(e) \prod_{e \in E(GP)\setminus E(G)} (1 - P(e))$$

Consequently, an uncertain database D^P implies a set of $\prod_{i=1}^{|DP|} 2^{E(GP)}$ exact graph databases.

Assuming independence among uncertain graphs in the database, the probability of an exact graph database D being implied by D^P is $P(D^P => D)$ $P(G_i^P => G_i)$, where G_i^P are graphs in D^P & D respectively.

2.3 Subgraph:

A subgraph S is a subset of a Graph G (since every set is a subset of itself, every graph is a subgraph of itself.). All the edges and vertices of G might not be present in S but if a vertex is present in S, it has a corresponding vertex in G and any edge in S connected vertices must connect the corresponding vertices in G.

2.4 Frequent subgraph:

It is defined as finding all subgraphs that appears frequently in a database according to a given frequency threshold.

3. OVERVIEW OF FREQUENT SUBGRAPH MINING:

Mining frequent subgraph patterns on an uncertain graph database is different from certain graph database. The significance of subgraph pattern can be measured by considering support of subgraph pattern[2](i.e proportion of input graphs containing subgraphs)
The expected support of S can be calculated by considering the probability distribution over the implicated graphs of graph database. The expected support value should not be less than the user-specified threshold value the subgraph S is frequent.
In traditional process, frequent subgraph pattern mining, a subgraph pattern is subgraph isomorphic to at least one exact graph in the input exact graph database and support of a subgraph pattern S in an exact graph can be formulated as

$$Sup_D(S) = \frac{|\{G | S \subset_{ex} G \text{ and } G \in D\}|}{|D|}$$

Whereas in uncertain graph database embedding of an exact subgraph in an uncertain graph in probabilistic sense hence for uncertain graph database the concept is to redefine.
Def: A connected exact graph S is a subgraph pattern in an uncertain graph database D if S is subgraph isomorphic to the at least one implicated graph database of D. Let S & S' be two subgraph patterns. S is subgraph pattern of S' otherwise S' is a super pattern of S, if $S \subset_{ex} S'$ and $|E(S)| + 1 = |E(S')|$. The support of a subgraph pattern S is an uncertain graph database D be a random variable.
The expected support is defined as
$Esup_D(S) = \sum_{i=1}^{m} S_i \, P(S_i) = \sum_{d \in I(D)} SupD(S) P(D => d)$

A subgraph pattern S is frequent if the expected support of S in D is not less than a threshold min sup Є [0,1] specified by users in an uncertain graph database D.

4.CLASSIFICATION BASED ON ALGORITHM APPROACH:

Algorithms for mining frequent subgraphs from an uncertain graph database follow the apriori property of expected support [7]and pattern growth approach[8]. The first part of finding frequent subgraph pattern is that generation of candidate patterns to be examined. Every subgraph constitutes a candidate pattern. For both exact and uncertain graphs, apriori property is adopted.

Let S and S' be two subgraph patterns. If S C S' then S is called a sub pattern of S' and |e(S')| = |e(S)|+1 then S is direct sub pattern of S' . then sup(S,D) >= sup(S',D). This is referred to apriori property of support and it holds expected the support of subgraphs in uncertain graph data bases.

Using apriori it is possible to avoid unnecessary tests for search of frequent subgraphs because if S is a frequent subgraph then every such pattern of S is frequent. If S is not frequent subgraph then no super pattern of S can be frequent. Consequently, the subgraph patterns in the data base can be organized as DAG(direct acyclic graph) which contains the candidate patterns as nodes with an empty root node. The DAG[8] enumerates all subgraph patterns from single edge to n edges adding additional edges at each level up to nth level. To avoid multiple times the same subgraph, a spanning tree of DAG is selected and is proposed in gspan[8] by imposing lexical order among the patterns. The subgraph patterns are enumerated by applying depth-first search of this tree structure[8]. The advantage of organizing into search tree is that the subgraph patterns which are infrequent then all its descendants can be pruned due to an apriori property of expected support[7].

5. OVERVIEW OF UNCERTAIN GRAPH DATA ALGORITHMS

5.1 MUSE ALGORITHM:

Find all subgraph patterns which are frequent in D, by taking input as an uncertain graph Database 'D' and an expected support threshold.

Muse algorithm[4],[5] adopts 2 crucial techniques.

i) By checking the overlapping relationship between the approximated interval (l,u) and ((1-ε)minsup, minsup) can find subgraph pattern can be output or not.

ii) To examine subgraph patterns and expected support satisfies the Apriori property

i.e. all super graphs of an infrequent subgraph pattern are also infrequent.(To use Apriori property, all subgraph patterns are organized into search tree, and the search tree is traversed in depth-first strategy).

To fulfil the second objective first study the property of the expected support i.e Apriori property of expected support. Apriori property is that all sub-patterns of a frequent subgraph patterns is also frequent and all super patterns of an infrequent subgraph patterns are also infrequent. This is used for reducing complexity in mining.

Finding Apriori property of expected support:

Based on the direct sub pattern relationship all subgraph patterns in uncertain graph database D can be organized as a direct acyclic graph (DAG)[8] with nodes representing subgraph patterns and DAG can be simplified to a tree by keeping only one parent to a subgraph patterns by using some specific schemes for which the subgraph patterns having more than one parent.

For example DFS coding scheme proposed in g-span[8], the DAG can be simplified to a search tree of subgraph patterns. The advantage of organizing subgraph patterns into a search tree is that, if any subgraph pattern is known to be infrequent, the all its descendants can be pruned due to the apriori property of expected support. The proposed approximate mining algorithm implies depth-first strategy[8] to traverse the search tree.

Algorithms for computing Expected Supports:

Expected Support[4],]5] of a subgraph of S in an uncertain DB, D can be computed by averages the probability of S occurring in every uncertain graph G Є D i.e P(S$\subset_U$G). The fundamental technique of a new approach is to transform the problem of computing P(S$\subset_U$G) to the DNF counting problem. Exact and approximate algorithms are developed to compute P(S$\subset_U$G) based on DNF counting.

Exact Algorithm:

i. Exactly compute the P(S$\subset_U$G) for small instances, not more than 30 embeddings.
ii. DNF counting.
iii. Inclusive exclusive principle[9].

Approximate Algorithm:

i. Approximates P(S$\subset_U$G) for large instances.
ii. DNF counting.
iii. Approximate the probability Pr(F)[10], of F being satisfied with an interval (l,u) of width at most ε.minsup such that Pr(F) ε [l,u].

Trade-off between exact Algorithm and Approximation Algorithm:

To choose the algorithms to compute P(S$\subset_U$G) based on time compiles.

Exact-occ-prob:

$\tau_{exact} = O(n^2 2^n |E(S)|^2)$ (1)

where n is number of embeddings of S in G , |E(S)| is the number of edges in S

Approx-occ-prob:

$\tau_{approx} = O[(16 n^2 \ln(2/\delta)|E(S)|)] / [\varepsilon^2 minsup^2]$ (2)

$\tau_{exact} > \tau_{approx}$

i.e $2^{n-4} |E(S)| > \ln(2/\delta) / (\varepsilon.minsup)^2$ (3)

If the above condition (3) is satisfied Approx-occ-prob is selected otherwise select Exact-occ-prob. This means number of embeddings of S in G is not large, the exact algorithm will out performs the approximation algorithm.

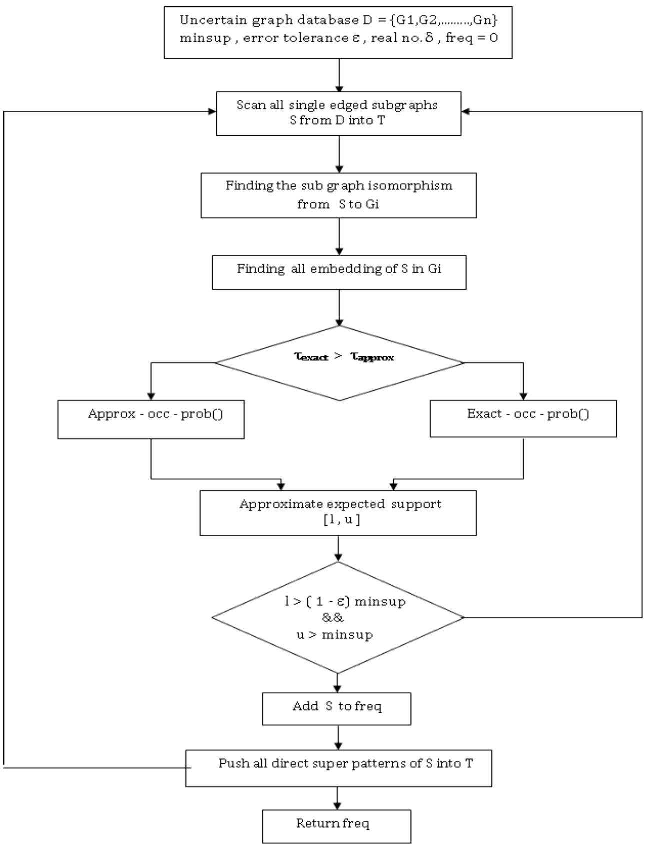

Fig-1: Flow diagram of MUSE algorithm

Experimental Evolution:

MUSE algorithm was implemented, and extensive experiments were performed to evaluate the efficiency, the memory usage , the approximate quality and scalability of MUSE, the impact of optimized pruning on the efficiency of MUSE and the impact of uncertainties on the efficiency of MUSE.

5.2 WMUSE ALGORITHM:

Focus on the finding frequent subgraphs patterns in DBLP uncertain data. Frequent subgraph pattern mining is formalized by using expected support measure. Weighted MUSE has better efficiency compared to MUSE[5] in terms of time complexity. In WMUSE[11] using weight factor, get good results in minimum time compared to MUSE.

MUSE is to discover the approximate set of frequent subgraph patterns from an uncertain graph database. But this is an interval based algorithm in which Monte Carlo algorithm[10] is used to find interval for estimating expected value to find frequent subgraphs. In MUSE they have considered PPI network .where as in WMUSE they have considered DBLP data to represent author and Co-author relationship.

In WMUSE considering DBLP graph database D, and an expected support threshold, we have to find out all frequent subgraph patterns using isomorphism[6] and embedding database D. In MUSE calculation of expected support of subgraph patterns becomes exponential when the embedding having repeated edges with repeated weights more than once. To avoid this repeated evaluation of expected support WMUSE is proposed.

WMUSE is the modified algorithm to MUSE by assigning weight factor w(0,1) to the edges of embeddings includes in the identified frequent subgraph pattern. At every iteration if any embedding traversed earlier we decrease the weight factor to 0.1 this represents low priorities to the embeddings. So that if next time any embedding with same edges, exact algorithm do not waste time to evaluate the expected support. This reduces the computational cost. The subgraph in the stack has low priority than a threshold value w=0.5, on all of its edges prunes then and do not traverse the path and place this pattern in the frequent subgraphs list. This saves time and an exact algorithm can work with more than 100 combinations.

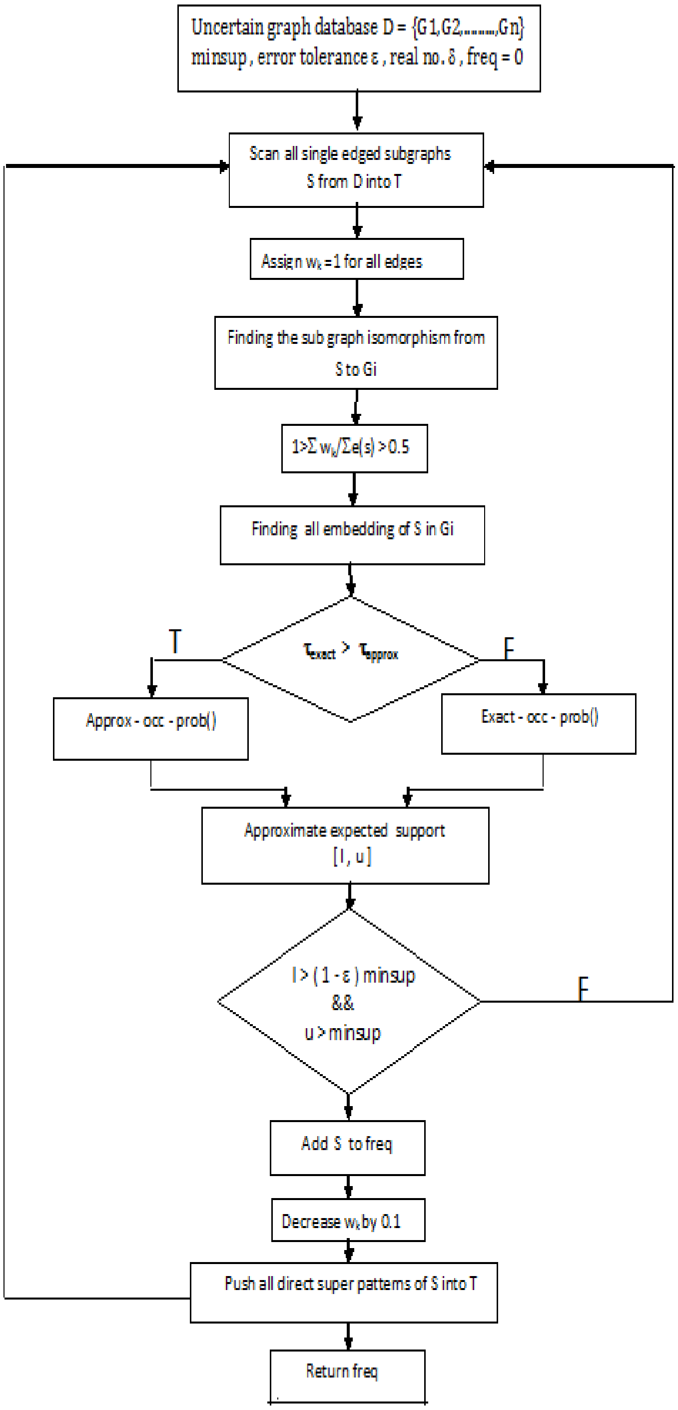

Fig-2: Flow diagram of WMUSE algorithm

Efficiency Analysis Of WMUSE:

WMUSE gives promising results as compared to original MUSE in terms of time. The number of output frequent subgraphs patterns decreases quickly with the increase in minsup. There is no big difference for the generation of number of frequent subgraphs using MUSE & WMUSE.

For small data set it is not significant as for large data set the number of generated frequent graphs and different in time execution are significant. WMUSE works on the large data set and reduces search process by assigning priority through weight factor.

Advantages of WMUSE:

WMUSE is proposed to discover possible frequent subgraph patterns from uncertain graph data. The analysis and the experimental results show that WMUSE has better efficiency as compared to MUSE in terms of time complexity. Due to an assignment of priority by assigning weight factor we get good results in minimum time compared to original MUSE implementation.

5.3 UGRAP ALGORITHM :

The main difficulty in solving the problem of finding frequent subgraphs is to examine the large number of candidate subgraph patterns and the large number of subgraph isomorphism tests to find the graphs that contain a given pattern.

UGRAP[12] proposed a method using the index of uncertain graph DB to reduce the number of comparisons need to find the frequent subgraph patterns. UGRAP relies on the Apriori property[7] for enumerating candidate subgraph patterns efficiently. The index is used to reduce the number of comparisons required for computing the expected support of each candidate pattern. It also enables additional optimization with respect to scheduling and early termination.

Existing approaches solve frequent subgraph patterns mining problem for exact graphs by performing two operations.

i. Generating candidate patterns to be examined.
ii. Testing for subgraph isomorphism to determine which graphs in the DB contain a given candidate pattern.

UGRAP[12] relies on an index of uncertain graph DB[13] to reduce the number of computations to determine the support of candidate patterns.
The index comprises two structures.

i. An inverted index on graph edges enhanced with edge probabilities.
ii. Structure providing summarized information regarding connectivity of graph nodes up to a specified path length.

Main contribution UGRAP :

Introduced an index of an uncertain graph[13] DB comprising information on graph edges along their probabilities and connectivity information between graph nodes. Pruning the search space when computing the expected support of candidate patterns. Efficiency of UGRAP improved by proposing additional optimizations for early termination and effective scheduling of graph comparisons.
For mining frequent subgraph patterns, existing methods are classified into 2 categories.

i. Apriori based.
ii. Pattern growth.

Apriori:

To generate candidate patterns, BFS is applied to generate subgraphs of size (K+1) by joining two subgraphs of the previous level. BFS is used by all these to build the primary building block of candidate generate.
AGM[7] starts the search by considering a single vertex and then proceeds by generating candidates adding one extra vertex at each subsequent step.FSG[19] uses edges instead of vertices as the primary building block for candidate generation. PM[20]uses single vertices or edges as building blocks for pattern generation and it utilizes edge-disjoint paths.

Pattern Growth:

To avoid costly BFS, adopt DFS, where patterns are grown directly from a single graph Instead of joining two previous subgraphs. gspan is the main representative of this pattern growth. It uses tree representation instead of adjacency matrix as coding. FFSM[6] Proposed vertical search scheme.(Join & Extension). ADI[21] Proposed adjacency index structure for large data bases. gSpan algorithm can be adopted to use ADI.

By constructing an index of the uncertain graph Database significantly prune the search space. And enables for additional optimization based on early termination and efficient scheduling to avoid expensive subgraph isomorphism tests.

UGRAP introduced

i. Edge index.
ii. Connectivity index.

The main goal of UGRAP[12] is to prune the search space when computing the expected support of candidate subgraph patterns by limiting the number of uncertain graphs that need to be examined. When dealing with large data bases and large graphs especially in the case of dense graphs, reduces the memory requirements.

Edge Index:

The First component of UGRAP index, denoted with I^E is an inverted index on graph edges extended with information on edge probabilities in order to take uncertainty of edges into account. The structure I^E is a map where each key is a label triple of the form t=(Lo, Lv, Le) representing graph edges and the value of each key is a list containing the identifiers of the graphs in which these edges appear, as well as the corresponding occurrence probability. I^E requires negligible memory and computational resources to be built even for large uncertain graph databases.

Connectivity Index:

The second component of the UGRAP index, denoted by I^C is a structure containing summarized information regarding connectivity of graph nodes. There is no need to construct the index for l=1. since the graphs containing single-edge paths can be efficiently retrieved from the edge index I^E. Therefore only considered
$l \in [2, l_{max}]$ when constructing connectivity index as well as for deciding whether an uncertain graph contains a candidate sub graph pattern.
UGRAP comprises two main parts.

i. Genetation of candidate sub graph patterns for examination [using method in gspan] also used in MUSE.
ii. To evaluate whether the candidate pattern is frequent using UGRAP[12], index structure[13].

The algorithm uses the information stored in the index to compute the upper bounds of the expected support of a pattern. So that infrequent sub graph patterns can be identified and pruned early without performing computationally expensive sub graph isomorphism tests and calculation of the expected supports.
Pruning process of candidate patterns is very efficient because

i. It examines only small subset of the graphs in the DB i.e those graphs contained in the list I_s (I_s is list of relatively small subgraps)

$$I_s = \bigcap_{e \in E(S)} IE\,(T(e))$$

ii. It uses the probabilities stored in the index i.e without involving the isomorphic tests.

I_s does not consider any structured information regarding how these edges are connected. To perform additional filtering before computing upper bound of expsup, I^C, the connectivity index is exploited.If $I^C\,(G^P, Lu, Lv, l) = 0$, then G^P is removed from I_s.This in turn reduces the size of this I_s, allowing for tight upper bound.

Computational cost is calculated using
$cost(G^P,S) = \ln(2/\delta)(\varepsilon.minsup)^2$ for approximation and
$2^{x-5}/X$ for exact.

Experimental Evaluation:

Concluding, the experimental evaluation with three real-world databases and with an extensive set of synthetic DB shows that U GRAP outperforms MUSE, the current state-of-the-art approach for frequent sub graph pattern mining in uncertain graph data bases.

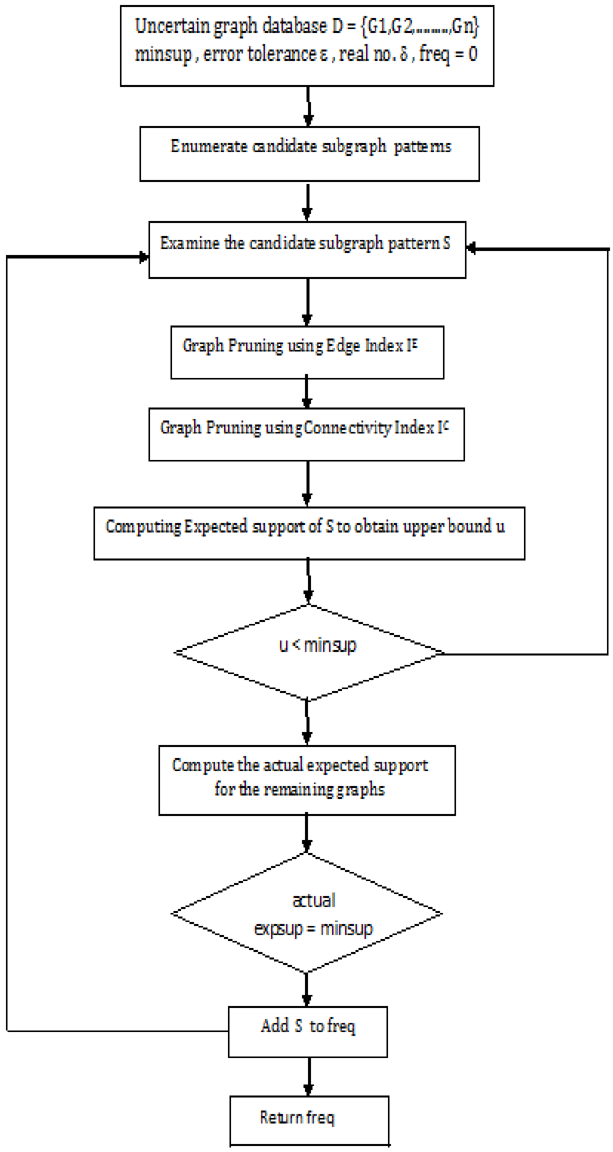

Fig-3: Flow diagram of UGRAP algorithm

5.4 MUSE - P ALGORITHM :

Frequent sub graph mining[2],[19] under the expected semantics is more suitable for exploring Motifs(a repeated one forming a pattern) in a set of uncertain graphs. Frequent sub graph mining under the probabilistic semantics is more suitable for detecting features from a set of uncertain graphs where motif exploration and feature detection are two main scenarios where frequent sub graph mining is applied.
MUSE-P[14],[15] is trying to find a broader set of sub graphs including all frequent sub graphs and a fraction of infrequent sub graphs but with ϕ - frequent probability at least τ-ε, where $0 < \varepsilon < \tau$ is error tolerance.

In other words A sub graph is

i. frequent if its ϕ - frequent probability at least τ.
ii. Infrequent if its ϕ - frequent probability is less than τ-ε
iii. All sub graphs with ϕ - frequent probability in [τ-ε,τ] are approximately frequent.

Frequent sub graph mining has been extensively studied on certain graph data. Very few work has been done on uncertain graph data. Previous studies on uncertain graph data for finding frequent sub graphs MUSE, WMUSE, UGRAP using Expected Semantics[4,5],[11],[12]. MUSE - P is study of uncertain graph data for finding frequent sub graphs on Probabilistic Semantics[14],[15].

To evaluate degree of recurrence of sub graphs, a measure called ϕ - frequent probability is introduced. The goal is to find all sub graphs with ϕ - frequent at least τ taking input as uncertain graphs and two real numbers $0<\phi, \tau<1$. Due to NP-hardness of the problem and to the # P-hardness [18]of computing the ϕ - frequent probability of a sub graph, an Approximate Mining Algorithm is proposed to produce (ε, δ)- approx. set Π of frequent sub graphs where $0<\varepsilon<\tau$ is error tolerance and $0<\delta<1$ is confidence bound.

Approximate alg. guarantees that

i. Any frequent sub graph ' S' contained in Π with probability at least$[(1-\delta)/2]^s$ where 's' is the number of edges in 'S'.
ii. Any infrequent sub graph with ϕ - frequent probability less than τ-ε is contained in Π with probability at most $\delta/2$.

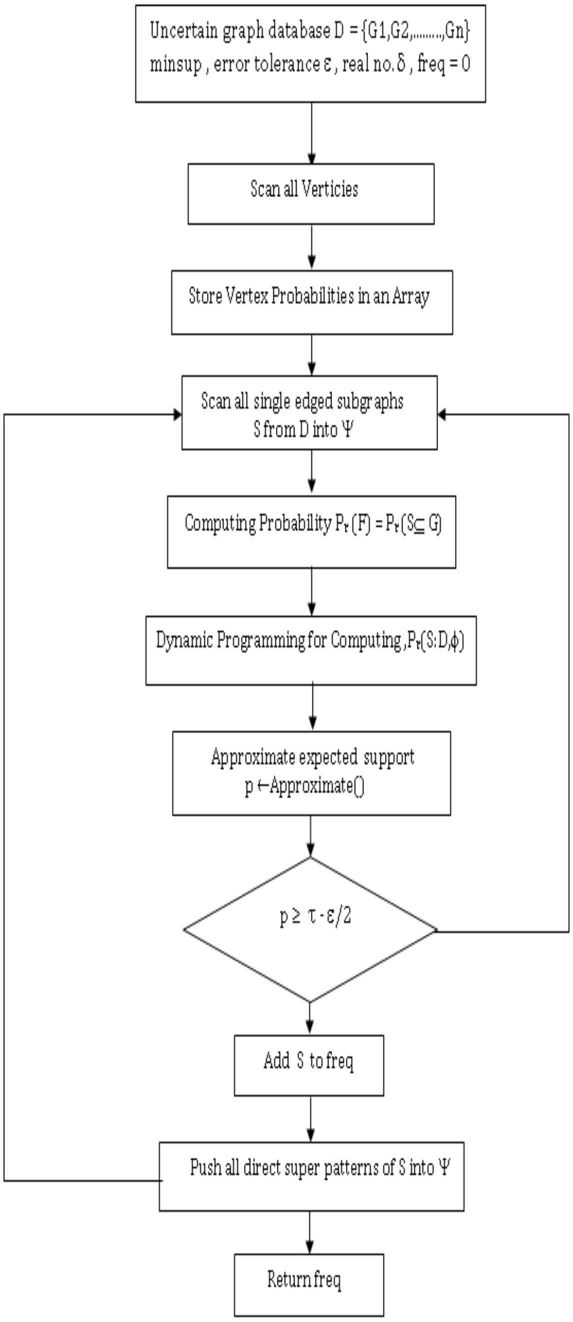

Fig-4: Flow diagram of MUSE-P algorithm

6. CLASSIFICATION OF UNCERTAIN GRAPH ALGORITHM

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 02 | May-2015 www.irjet.net p-ISSN: 2395-0072

Sno	Algorithm	Input	Graph rep-resentation	Candidate genera-tion	Frequency counting	Nature of output	Semantics	Limitations	Advantages
1	MUSE	Un-certain graph data	Adjacency matrix	DNF	DFS coding scheme	Frequent sub graphs	Expected	Freq. sub graphs are not exact and number of isomor-phism tests are needed	First algorithm to find frequent subgraph from uncertain graph data.
2	WMUSE	Uncertain graph data	Adjacency matrix	DNF	DFS coding scheme	Frequent sub graphs	Expected	Similar to MUSE except less number of isomorphi-sm tests due to weight factor. no of com-parisons are more	Reduces the iso-Morphism Tests with the in -troduction of weight factor to edges.
3	UGRAP	Un-certain graph data	Tree structure	DNF	DFS coding	Frequent sub graphs	Expected	The additional time and memory requirement for construction and maintenance of index .	Efficient compared to MUSE algorithm by reducing iso-morphism tests.
4	MUSE-P	Un-certain graph data	Tree structure	DNF	DFS coding	Frequent sub graphs	Probabili-stic	Probable number of frequent subgraphs using approximate mining algorithm	Number of frequent subgraphs are increased due to addition of some infrequent subgraphs as frequent.

Table-1: Classification of uncertain graph algorithms

6. SUMMARY

In this paper, we present a brief overview of mining frequent subgraphs from uncertain graph data under expected semantics and probabilistic semantics. By considering expected support measure, the frequent subgraph mining problem is formalized. To measure average frequentness of a subgraph and the subgraphs that are expected to occur frequently, the expected semantics are more suitable choice, though the ϕ-frequent probability contains more information on the frequentness of a subgraph than the expected support. This paper gives basic idea about various algorithms used for Mining frequent subgraphs from uncertain graph data.

7. REFERENCES

[1] D.J. Cook and L.B. Holder, Mining Graph Data. Wiley, 2006.
[2] M. Kuramochi and G. Karypis, "Frequent Subgraph Discovery," Proc. Int'l Conf. Data Mining, 2001.
[3] C.C. Aggarwal, Y. Li, J. Wang, and J. Wang, "Frequent Pattern Mining with Uncertain Data," Proc. ACM SIGKDD Conf., 2009.
[4] Z. Zou, J. Li, H. Gao, and S. Zhang, "Frequent subgraph pattern mining on uncertain graph data", In Proceeding of the 18th ACMconference on information and knowledge management, CIKM '09, pages 583-592, 2009.
[5]. Zou, Z., Li, J., Gao, H., Zhang, S.: Mining frequent subgraph patterns from uncertain graph data. IEEE Trans. Knowl. Data Eng. 22(9), 1203–1218 (2010).
[6] J. Huan, W. Wang, and J. Prins, "Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism," Proc. Int'l Conf. Data Mining, 2003.
[7] A. Inokuchi, T. Washio, and H. Motoda, "An Apriori-Based Algorithm for Mining Frequent ubstructures from Graph Data," Proc. European Conf. Principles of Data Mining and Knowledge Discovery, 2000.
[8] X. Yan and J. Han, "gSpan: Graph-Based Substructure Pattern Mining," Proc. Int'l Conf. Data Mining, 2002.
[9] M. Mitzenmacher and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge Univ. Press, 2005.
[10] R.M. Karp and M. Luby, "Monte-Carlo Algorithms for Enumeration and Reliability Problems," Proc. Ann. Symp. Foundations of Computer Science, 1983.
[11]. Shawana Jamil, Azam Khan, Zahid Halim and A. Rauf Baig Weighted MUSE for Frequent Sub-graph pattern Finding in Uncertain DBLP Data,International conference in 2011 .
[12] Papapetrou, O., Ioannou, E., Skoutas, D.: Efficient discovery of frequent subgraph patterns in uncertain graph databases. In: Proceedings of EDBT, pp. 355–366 (2011)
[13] X. Yan, P. S. Yu, and J. Han. Graph indexing: A frequent structure-based approach. In *SIGMOD*, pages 335–346, 2004.
[14]. Zou, Z.,Gao,H., Li, J.: Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics. In: Proceedings of KDD, pp. 633–642 (2010)
[15] Jianzhong Li · Zhaonian Zou · Hong Gao "Mining frequent subgraphs over uncertain graph databases under probabilistic semantics. The VLDB Journal (2012) 21:753–777
[16] C. Aggarwal. *Managing and mining uncertain data*. Springer, 2009.
[17] C. Aggarwal and H. Wang. *Managing and mining graph data*. Springer, 2010.
[18] L.G. Valiant, "The Complexity of Computing the Permanent," Theoretical Computer Science, vol. 8, pp. 189-201, 1979.
[19]. Kuramochi, M.,Karypis, G.:An efficient algorithm for discovering frequent subgraphs. IEEE Trans. Knowl. Data Eng. 16(9), 1038–1051 (2004)
[20] E. Gudes, S. E. Shimony, and N. Vanetik. Discovering frequent graph patterns using disjoint paths. *IEEE Trans. Knowl.Data Eng.*, 18(11):1441–1456, 2006.
[21] C. Wang, W. Wang, J. Pei, Y. Zhu, and B. Shi. Scalable mining of large disk-based graph databases. In *KDD*, pages 316–325, 2004.

BIOGRAPHIES

S.V.S. Santhi is an Assistant Professor at PRIME College, Vizianagaram, Andhra Pradesh, India. She received her Master degree in Information Technology from Andhra University in 2008 and is currently Pursuing her Part time PhD degree in Information technology from GITAM University, Visakhapatnam, Andhra Pradesh, India. Her Current research areas include data mining and graph mining.

Padmaja Poosapati is an Associate Professor in Department of Information Technology at GITAM University, Visakhapatnam, Andhra Pradesh, India. She received her Master degree in Computer Science and Engineering from Andhra University in 1999 and PhD degree in Computer Science and Engineering from Andhra University, India in 2010. Her Current Research interests include Clustering and Classification in data mining and graph mining.

