# FP-GROWTH BASED NEW NORMALIZATION TECHNIQUE FOR SUBGRAPH RANKING

E.R.Naganathan[1] S.Narayanan[2] K.Ramesh kumar[3]

[1]Department of Computer Applications, Velammal Engineering College
Ambattur-Redhills Road, Chennai – 600066, Tamil Nadu, South India.
ern_india@yahoo.co.in
[2]Computer Centre., Alagappa University, Karaikudi-630003. South India.
narayanan_alu@yahoo.com
[3]Department of Computer Applications., Karunya University
Coimbatore-641114. South India.
rameshkumar_phd@yahoo.co.in

***ABSTRACT***

*The various problems in large volume of data area have been solved using frequent itemset discovery algorithms. As data mining techniques are being introduced and widely applied to non-traditional itemsets, existing approaches for finding frequent itemsets were out of date as they cannot satisfy the requirement of these domains. Hence, an alternate method of modeling the objects in the said data set, is graph. Modeling objects using graphs allows us to represent an arbitrary relation among entities. The graph is used to model the database objects. Within that model, the problem of finding frequent patterns becomes that of finding subgraphs that occur frequently over the entire set of graphs. In this paper, we present an efficient algorithm for ranking of such frequent subgraphs. This proposed ranking method is applied to the FP-growth method for discovering frequent subgraphs. In order to find out the ranking of subgraphs we present a new normalization technique which is the modified normalization technique applied at each position for a chosen value of Discounted Cumulative Gain (DCG) of a subgraph. Instead of DCG another modified approach called Modified Discounted Cumulative Gain (MDCG) is introduced. The MDCG alone cannot be used to achieve the performance from one query to the next in the search engine's algorithm. To obtain the new normalization technique an ideal ordering of MDCG (IMDCG) at each position is to be found out. A Modified Discounted Cumulative Gain (MDCG) is calculated using "lift" as a new approach. IMDCG is also evaluated. Then the new approach for finding the normalized values are to be computed. Finally, the values for all rules can be averaged to get an average performance of a ranking algorithm. And also the ordering of obtained values as a result at each position will provide the order of evaluation of rules which in turn gives an efficient ranking of mined subgraphs.*

## *Key words*

*New Normalization Technique, FP-growth, Lift, Modified Discounted Cumulative Gain(MDCG),*

## 1. INTRODUCTION

Structured data mining is a major research topic in recent study of Data Mining. One of the most common type of representation of structured data is graph. Graph-based data mining exhibits a number of methods to mine the relational aspects of data. Two major approaches to graph-based data mining are frequent subgraph mining and graph-based relational learning. Graph is an alternate way of modeling the objects [16]. In such model, the technique for finding frequent patterns leads to that of discovering subgraphs that occur frequently over the entire set of graph. The sparse graph will represent the subgraph. This representation will store input transactions, intermediate candidates and frequent subgraphs [14]. Graph-based data mining (GDM) is the task of finding novel, useful, and understandable graph-theoretic patterns

DOI: 10.5121/ijdms.2011.3106 

in a graph representation of data. So many approaches to GDM exist based on the task of identifying

frequently occurring subgraphs in graph transactions, that is, those subgraphs meeting a minimum level of support [22].

Currently, there are two major trends in frequent subgraph mining: the Apriori-based approach and the Pattern-growth approach [23] [4] [9]. The key difference between these two approaches is how they generate candidate subgraphs. The Apriori heuristic achieves good performance gain significantly by reducing the size of candidate sets. However, in situations with prolific frequent patterns, long patterns, or quite low minimum support thresholds, an Apriori-like algorithm may still suffer from the nontrivial costs. If one can avoid generating a huge set of candidates, the mining performance can be substantially improved. To overcome this problem, another technique called FP-Growth algorithm was introduced which satisfies the same achievements without candidate generation [9]. The only outcome of this FP-Growth method is to discover the frequent subgraphs from the graph data set.

In this paper, we present a new method to find out the Normalization Technique for the subgraphs obtained from the FP-Growth model. By, applying the proposed method to this algorithm, the same can be extended to rank the frequent subgraphs also. This ranking algorithm FPGBG (FP-Growth Based Graphgain) will provide the substantial and essential techniques in improving the performance of the frequent subgraph mining. This new approach will also provide the average performance of the search algorithms and also the ranking of the frequent subgraphs obtained. Based on this subgraph ranking rules, the performance of the FP-Growth graph pattern will be improved. By arranging the new normalized values in descending order we will get the best priority of ranking of subgraphs. Here the sample data of subgraph is based on the FP-growth algorithm. The FP-growth method mines the complete set of frequent itemsets without candidate generation. FP-growth works in a divide-and-conquer way. The first scan of the database derives a list of frequent items in which items are ordered by frequency descending order. According to the frequency descending list, the database is compressed into a frequent-pattern tree, or FP-tree, which retains the itemset association information. FP-tree creation is required by the FP-growth approach. Compared to large document graphs, mining of FP-tree is easier. This is due to the fact that, itemsets in a transaction database is smaller compared to the edge list of document-graphs. In original FP-tree mining procedure, there is no direct connection between the transactions. In contrast, they become related to each other in the context of connectivity of the subgraph. The FP-growth algorithm transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix.

This paper is organized as follows : Section 1 gives the introduction and illustrate the basic concept of Graph Data Mining(GDM), FP-growth method and the introduction for subgraphs and new normalization technique. The evaluation of Normalization technique is obtained using “lift” in MDCG. Section 2 discusses the existing work regarding the subgraph mining and some ranking methods. Next section (Section 3) illustrates the proposed method for modified discounted cumulative gain (MDCG) by constructing the rules for “lift” on every point of subgraph. In Section 4 the proposed New Normalization Technique is explained. Section 5 will explain the proposed algorithm for ranking the subgraphs. Section 6 gives the implementation details on sample example. Section 7 discusses the result based on the performance of the algorithm on a given sample. Finally, the last section concludes.

## 2. RELATED WORK

The problem of ranking, in which the goal is to learn a realvalued ranking function that induces a ranking or ordering over an instance space, has recently gained much attention in machine learning (Cohen et al., 1999; Herbrich et al., 2000; Crammer & Singer, 2002; Freund et al., 2003). In developing algorithms for ranking, the main form of data that has been considered so far is vector-valued data, i.e., the algorithm receives as input a finite number of objects in some Euclidean space $\mathbb{R}^n$, together with examples of order relationships or preferences among them, and the goal is to learn from these examples a ranking function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that orders future objects accurately.

The advent of this existing research area around the turn of the millennium, several clever algorithms for frequent subgraph mining have been developed. All the algorithms developed originally for frequent item set mining. Examples include MolFea [15], FSG [13], MoSS/MoFa [3], gSpan [24], CloseGraph [25], FFSM [10], and Gaston [17, 18]. A related, but slightly different approach is used in Subdue [5], which is geared towards graph compression with common subgraphs rather than frequent subgraph mining. An overview of several methods and related problems can be found in [6].

All the above methods are to discover the frequent pattern only in a graph dataset and some of the ranking schemes exists are for the web search engine purpose. There is no high amount of research in the area of applying ranking over the subgraphs. The frequent subgraphs with similar patterns have to be compulsorily classified under the ranking. That is, if similar pattern exists, then ranking of them will lead to a solution of placing them in an appropriate order for further applications. Because ranking plays an important role in many graph applications as discussed earlier. The lack of research in this area motivated this research.

## 3. MODIFIED DISCOUNTED CUMULATIVE GAIN(MDCG)

Modified Discounted Cumulative Gain (MDCG)[16] is a modified measure of Discounted Cumulative Gain (DCG). Discounted Cumulative Gain is a measure of effectiveness of a Web search engine algorithm or related applications, often used in information retrieval[8]. The concept of DCG is that highly relevant documents appearing lower in a search result list should be changed as the graded lift value and reduced logarithmically proportional to the position of the result. Using a graded lift scale of documents in a search engine result set, MDCG measures the usefulness, or gain, of a document. From top of the result to the bottom with the gain of each result discounted at lower ranks [12], the gain is accumulated cumulatively The DCG is given by

$$DCG_p = \sum_{i=1}^{p} \frac{2^{reli}-1}{\log_2(1+i)} \qquad (1)$$

Hence the Modified Discounted Cumulative Gain (MDCG) is obtained using a new measure called 'lift" [11, 16]. And is defined as :

$$MDCG_p = lift(F_1) + \sum_{i=2}^{p} \frac{lift(F_i)}{\log_2 i} \quad (2)$$

There has not been any theoretically bold justification for using a logarithmic reduction factor[7]. An alternative formulation of MDCG recorded much stronger emphasis on relevant documents of higher ranking using a power distribution and is formulated as :

$$MDCG_n = \sum_{i=1}^{n} \frac{2lift(F_i)-1}{\log_2(1+i)} \qquad (3)$$

where lift is the statistical definition of dependence of two sets X and Y which is given by

$$Lift = \frac{P[A \cap B]}{P[A]\,P[B]} \qquad (4)$$

with the obvious extensions to more than two sets [20].

Lift originally called Interest, was first introduced by Motwani, et al., (1997), it measures the number of times *X* and *Y* occur together compared to the expected number of times if they were statistically independent.[21]

The function of confidence can also define the Lift[2]

$$Lift\ (A \rightarrow C) = \frac{|\,D\,|conf\ (A \rightarrow C)}{\sup\ (C)} \qquad (5)$$

Where

Support of a graph is given by [1]

In a given graph $F_G$, the support $F_S^G$ is defined as

$$Sup(F_G) = F_S^G = \frac{number\ of\ graph\ transactions\ F}{total\ number\ of\ graph\ transactions} \qquad (6)$$

And confidence is given by [1]

Given two induced subgraph $F_b$ and $F_h$, the confidence of the association rule

$F_b \Rightarrow F_h$ is defined as

$$Conf = \frac{no.of\ graphs\ F\ where\ F_b \cup F_h \subset F \in FD}{no.of\ graph\ F\ where\ F_b \subset F \in FD} \qquad (7)$$

In the case of subgraph architecture, lift can be defined as

$$Lift = \left(F_{Lift}^G\right) = \frac{no.of\ graph\ of\ F_b\ and\ F_h}{no.of\ F_b\ x\ number\ of\ F_h} = \frac{P(AUB)}{P(A)\ P(B)} \qquad (8)$$

The relationship of *A* and *B* are defined by the lift as

i) lift value > 1 then *A* and *B* depend on each other

ii) lift value < 1 then *A* depends on the absence of *B* or vice-versa

iii) lift value close to 1 then *A* and *B* are independent.

## 4. NEW NORMALIZATION TECHNIQUE

Depending on the query, the search result lists vary in length. The query performance is incomparable with another query in this form, since the other query may have more results resulting in a large overall MDCG which may not be better. For the comparison purpose the MDCG values are to be normalized. With MDCG alone, for example, search engine's performance comparison from one query to the next cannot be consistently achieved, so the cumulative gain at each position for a chosen value of p should be normalized across queries. This is done by sorting documents of a result list by lift, producing an ideal MDCG(IMDCG) at position p. For a query, the new normalization value is computed as :

$$New\,Normalized\ Value = \frac{MDCG_n}{IMDCG_n} \qquad (9)$$

An ideal ordering is needed for the query to normalize MDCG values. All the normalized values are averaged to obtain a average performance measure of ranking algorithm. In a perfect ranking algorithm, the value of MDCGp will be the same as that of IMDCGp and the normalized value produced is 1.0. All new normalized calculations are then relative values on the interval 0,.0 to 1.0 and so are cross-query comparable.

## 5. PROPOSED ALGORITHM FOR RANKING THE SUBGRAPH

The 'lift' values for every rule of a subgraphs are calculated first. Then the corresponding MDCG and IMDCG values are to be find out. Finally the algorithm computes the new normalized values. This new normalized values obtained are then sorted to obtain the ranking rule of the mined subgraphs. This involves some time delay which depends on the number of lift values/rules taken into account. If more than one rules or lift values are having the same measure, then this algorithm will provides the ordering for such similarities also. This is the main advantage of this algorithm.

### Algorithm

Input : FP-growth subgraphs $F_1, F_2, \ldots F_{k-2}$ , support $\sigma$ , confidence

Output : Lift values, MDCG, IMDCG, Normalization values and Order of Normalized values.

Step 1 : Compute the Lift value for each subgraph such that

$$Lift = \left(F^{G}_{Lift}\right) = \frac{no.of\ \ graph\ \ of\ F_b\ and\ F_h}{no.of\ F_b\ x\ number\ \ of\ F_h} = \frac{P(AUB)}{P(A)\ P(B)}$$

The Lift also defines the relationship of A and B by the condition

i) if the lift value is greater than 1 then A and B depend on each other

ii) if the lift value is less than 1 then A depends on the absence of B or B depends on the absence of A.

iii) if the lift value is close to 1 then A and B are independent.

Step 2 : Calculate the MDCG value such as

$$MDCG_p = lift(F_1) + \sum_{i=2}^{p} \frac{lift(F_i)}{\log_2 i}$$

Step 3 : Then calculate the value of IMDCG by considering the descending ordering of the lift and follow the calculation of MDCG as in the above step.

Step 4 : Evaluate the new normalized values at each point as follows :

$$New\ \ Normalized\ \ \ Value\ \ = NV_n = \frac{MDCG_n}{IMDCGP_n}$$

Step 5: Using sort by exchange method, sort the newly obtained normalized values in descending order to find the ranking position of subgraphs.

### Rule Construction

In any given graph $F_G$, the support $F_S^G$ is given as

$$Sup(F_G) = F_S^G = \frac{no.of\ graph\ transactions\ F}{total\ no.of\ graph\ transactions} \qquad \text{By (6)}$$

The confidence of the association rule for a given two induced subgraph $F_b$ and $F_h$

$F_b$ => $F_h$ is defined as

$$Conf(F_b \Rightarrow F_h) = \frac{no of graphs\ where F_b \cup F_h \subset F \in FD}{no.\ of\ graph\ where F_b \subset F \in FD} \quad By\ (7)$$

The given graph $F_G$ is called as frequent induced subgraph only if the value of sup($F_G$) is more than a threshold value

## 6. EXAMPLE

Finding of subgraphs follows the FP-growth method to achieve the most effective pruning. FP-growth works in a divide-and-conquer way. The first scan of the database derives a list of frequent items in which items are ordered by frequency descending order. According to this, the database is compressed into a frequent-pattern tree, or FP-tree which retains the itemset association information. The FP-tree is mined by starting from each frequent length- pattern, constructing its conditional pattern base, then constructing its conditional FP-tree and performing mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree. This algorithm transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix.. The transaction example referred by the authors of [9] is taken as our example to explain the concept as follows :

Table 1. Resultant Frequent Pattern

| Subgraphs | Frequent Patterns |
|---|---|
| F1 | 2 16 35 14 7 |
| F2 | 2 4 37 |
| F3 | 2 16 10 37 |
| F4 | 2 16 35 |
| F5 | 2 4 10 7 |
| F6 | 2 10 |
| F7 | 4 16 35 14 18 43 18 7 |
| F8 | 4 14 18 37 43 |
| F9 | 4 10 35 43 |

Hence the subgraphs listed are $F_1$, $F_2$, . $F_9$. And the rules are

| Rules | P(A) | | P(B) |
|---|---|---|---|
| R1 | 2 | ➔ | 10 |
| R2 | 2 → 4 | ➔ | 10 |
| R3 | 2 → 4 | ➔ | 10 → 7 |
| R4 | 2 → 16 | ➔ | 35 |
| R5 | 2 → 16 | ➔ | 10 → 37 |
| R6 | 2 → 16→ 35 | ➔ | 14 → 7 |
| R7 | 4 → 10 → 35 | ➔ | 43 |
| R8 | 4→14→18→37 | ➔ | 43 |
| R9 | 4→16→35→14→18 | ➔ | 43 → 18 → 7 |

The P(A) and P(B) are taken for the nine rules and is labeled as R1, R2 … R9. The "lift" values are calculated according to the rules from the above example. The corresponding new normalized values for all the rules are obtained. The ranking can be considered on the basis of the ordered(descending) newly calculated normalized values.

## 7. RESULTS AND DISCUSSION

The proposed algorithm is applied into the data of Transaction Example with minimum support of 3 handled in the FP-growth method. The following is the complete summary of result :

Based on the frequent subgraphs generated from the sample data set, construction of rule for identified frequent subgraphs are then made to find out the 'lift' measure. The value of lift at each position of the rule is taken into the account of calculating MDCG and IMDCG vaues. The new normalized value is then computed for each frequent subgraph. By sorting these values in descending order, the rules for ranking of subgraph is obtained. This ranking method may play an important role in the subgraph ranking algorithms.

The following Tables provides the summary of results.

Table 2 Lift and MDCG values for the Rules

| i | Log i | Rule | lift | Lift/Log I | MDCG |
|---|---|---|---|---|---|
| 1 | 0.0000 | R1 | 0.03333 | 0.00000 | 0.03333 |
| 2 | 1.0000 | R2 | 0.20000 | 0.03333 | 0.06667 |
| 3 | 1.5850 | R3 | 0.50000 | 0.12619 | 0.19285 |
| 4 | 2.0000 | R4 | 0.08333 | 0.25000 | 0.44285 |
| 5 | 2.3219 | R5 | 0.33333 | 0.03589 | 0.47874 |
| 6 | 2.5850 | R6 | 0.50000 | 0.12895 | 0.60769 |
| 7 | 2.8074 | R7 | 0.33333 | 0.17810 | 0.78580 |
| 8 | 3.0000 | R8 | 0.33333 | 0.11111 | 0.89691 |
| 9 | 3.1699 | R9 | 1.00000 | 0.10516 | 1.00206 |

Table 3. IMDCG Values

| i | Log I | Rule | Lift | Lift/Log I | IMDCG |
|---|---|---|---|---|---|
| 1 | 0.0000 | R1 | 1.00000 | 0.00000 | 1.00000 |
| 2 | 1.0000 | R2 | 0.50000 | 1.00000 | 2.00000 |
| 3 | 1.5850 | R3 | 0.50000 | 0.31547 | 2.31547 |
| 4 | 2.0000 | R4 | 0.33333 | 0.25000 | 2.56547 |
| 5 | 2.3219 | R5 | 0.33333 | 0.14356 | 2.70902 |
| 6 | 2.5850 | R6 | 0.33333 | 0.12895 | 2.83798 |
| 7 | 2.8074 | R7 | 0.20000 | 0.11874 | 2.95671 |
| 8 | 3.0000 | R8 | 0.08333 | 0.06667 | 3.02338 |
| 9 | 3.1699 | R9 | 0.03333 | 0.02629 | 3.04967 |

Table 4. New Normalized Values

| Rules | New Normalized Value($NV_n$) (MDCG/IMDCG) |
|---|---|
| R1 | 0.033333 |
| R2 | 0.033333 |
| R3 | 0.083289 |
| R4 | 0.172621 |
| R5 | 0.176721 |
| R6 | 0.214129 |
| R7 | 0.265767 |
| R8 | 0.296658 |
| R9 | 0.328581 |

Table 5. Ordered $NV_n$ Values

| Rule | Ordered MDCG/IMDCG |
|---|---|
| R9 | 0.328581 |
| R8 | 0.296658 |
| R7 | 0.265767 |
| R6 | 0.214129 |
| R5 | 0.176721 |
| R4 | 0.172621 |
| R3 | 0.083289 |
| R1 | 0.033333 |
| R2 | 0.033333 |

Table 6. Ordered Lift Values

| Rule | Ordered Lift |
|---|---|
| R9 | 1.00000 |
| R3 | 0.50000 |
| R6 | 0.50000 |
| R5 | 0.33333 |
| R7 | 0.33333 |
| R8 | 0.33333 |
| R2 | 0.20000 |
| R4 | 0.08333 |
| R1 | 0.03333 |

The main advantage of this technique is that, if similar cases are present, then there is a possibility of finding the priority of the subgraph. For example, the lift value for the rules R5, R7 and R8 are 0.33333 (Table 2). Suppose, if the lift is used as a base for ranking, then there should be a decision to take the correct order among the above three values. In such case, the new normalization technique will play an efficient role to carryout the placement of rules according to the order. Based on the result, the ranking position of the rules R5, R7 and R8 are 5, 3, and 2 respectively(Table 5). Another advantage of this technique is that no rule will have the same rank eventhough their lift values are same. But more than one rules have the same normalized values which is very rare. In that case, the ordering of lift values can be taken into consideration for fixing the priority. For example, the rules R1 and R2 will have the same normalized values as 0.033333(Table 5). In this case, the lift values corresponding to R1 and R2 such as 0.03333 and 0.20000 can be ordered and thus the rule R2 is considered as prior to R1(Table 6).

## 8. CONCLUSION

This paper illustrates two techniques. The first thing is the Normalized values obtained by a new method using lift measure at each position of large number of frequent subgraphs generated by the FP-Growth method. Second one is the ordering of the normalized values for ranking of subgraphs. Experimental results proves the performance and practical usefulness of the presented algorithm. The presented technique can also be extended for ranking methods in web mining, medical data mining and for other similar problems. Once the subgraph data set is represented by the FP-growth method, then this algorithm will present an efficient way of constructing the ranking of mined subgraphs with the help of newly founded normalization technique. The proposed ranking technique can also be applied for the subgraphs mined from any other methods. This method may be the one of the perfect ranking scheme among the subgraphs mined and this ranking scheme will play an efficient role in the subgraph applications.

## REFERENCES

[1] Akihiro Inokuchi, Takashi Washio and Hiroshi Motoda, (2000), "An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data", PKDD2000, Sept. 13-16, Lyon, France.

[2] Bayardo Jr R.J. and Rakesh Agrawal,, (1999) "Mining the most Interesting Rules", Procedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 145-154.

[3] Borgelt C. and Berthold M.R. (2002), Mining Molecular Fragments: Finding Relevant Substructures of Molecules. Proc. IEEE Int. Conf. on Data Mining (ICDM 2002, Maebashi, Japan), 51–58. IEEE Press Piscataway, NJ, USA 2002.

[4] Cohen M. and Gudes E (2004) *Diagonally subgraphs pattern mining*. In: Workshop on Research Issues on Data Mining and Knowledge Discovery proceedings, 2004, Pages: 51–58.

[5] Cook D.J. and Holder L.B.. (2000) "Graph_Based Data Mining". IEEE Trans. On Intelligent Systems 15(2):32-41. IEEE Press, Piscataway, NJ, USA.

[6] Cook D.J. and Holder L.B. (2007), Mining Graph Data. J. Wiley & Sons, Chichester, United Kingdom 2007.

[7] Croft. B. Metzler. D. and Strohman T. (2009), "Search Engines. Information retrieval in practice". Adison Wesley.

[8] "Discounted Cumulative Gain" – Wikipedia, the free encyclopedia.

[9] Han J. Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In:Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD'00), Dallas, TX, pp 1-12.

[10] Huan J, Wang W, and Prins J. (2003), Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism. Proc. 3rd IEEE Int. Conf. on Data Mining (ICDM 2003), 549–552. IEEE Press, Piscataway, NJ, USA 2003.

[11] Information Management Online. (2002) Why Lift? – Data Modeling and Mining, June 21.

[12] Kalerro Jarrelin, Jaana Kekalainen : (2002) "Cumulated gain based evaluation of IR techniques". ACM Transactions on Information system 20(4), 422-446.

[13] Kuramochi M and Karypis G. (2001), Frequent Subgraph Discovery. Proc. 1st IEEE Int. Conf. on Data Mining (ICDM 2001, San Jose, CA), 313–320. IEEE Press, Piscataway, NJ, USA 2001.

[14] Michihiro Kuramochi and George Karypis. (2001) "Frequent Subgraph Discovery", IEEE International Conference on Data Mining (ICDM).

[15] MolFea - S. Kramer, L. de Raedt, and C. (2001), Helma. Molecular Feature Mining in HIV Data. Proc. 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2001, San Francisco, CA), 136–143. ACM Press, New York, NY, USA 2001

[16] Naganathan, E.R, Narayanan S and Ramesh kumar K, (2010) "Modified Discounted Cumulative Gain – a New Measure for Subgraphs", International journal of Computer Science and Communications, Vol.1, No.2, July-December 2010, pp 137-139.

[17] Nijssen S and Kok J.N (2004), A Quickstart in Frequent Structure Mining Can Make a Difference. Proc. 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD2004,

Seattle, WA), 647–652. ACM Press, New York, NY, USA 2004.

[18] Nijssen S and Kok J.N, (2005), The Gaston Tool for Frequent Subgraph Mining. Electronic Notes in Theoretical Computer Science 127(1):77-87. Elsevier Science, Amsterdam, Netherlands 2005.

[19] Ping Guo, Xin-Ru Wang and Yan-Rong Kang, (2006) *Frequent mining of subgraph structures*. J. Exp. Theor. Artif. Intell., 2006, vol. 18, no. 4, Pages: 513-521.

[20] Sergey Brin, Motwani. R. and Silverstein. C. (1997) "Beyond Market Baskets : Generalizing Association Rules to Correlations", SIGMOD'97 AI, USA.

[21] Sergey Brin, Rajeev Motwani, Jeffrey Ullman. D. and Shalom Tsur., (1997), "Dynamic itemset counting and implication rules for market basket data". In SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, pages 255-264, Tucson, Arizona, USA, May.

[22] Washio, T., & Motoda, H. (2003). "State of the art of graph-based data mining". SIGKDD Explorations, 5(1), 59-68.

[23] Yan X and Han. J, (2002) *gSpan, Graph–Based Substructure Pattern Mining*. In Proc. IEEE Int'l Conf. on Data Mining ICDM, Maebashi City, Japan, November 2002. Pages: 721–723.

[24] Yan X. and Han J. (2003), gSpan: Graph-Based Substructure Pattern Mining. Proc. 2nd IEEE Int. Conf. on Data Mining (ICDM 2003, Maebashi, Japan), 721–724. IEEE Press, Piscataway, NJ, USA.

[25] Yan X and Han J. (2003), Closegraph: Mining Closed Frequent Graph Patterns. Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2003, Washington, DC), 286–295. ACM Press, New York, NY, USA 2003.