International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online) 2249 –071X, ISSN (Print): 2278 – 4209

New Normalized Technique using FP Growth for Subgraph Ranking

Rishma Chawla
Asst. Professor, M.Tech. (CSE)
RIET, Phagwara
chawlan02@yahoo.com

Bhanu Arora
M.Tech (CSE)
RIET, Phagwara
bhanuarora874@gmail.com

***Abstract* – Data mining techniques are being introduced and widely applied to non-traditional itemsets; existing approaches for finding frequent item sets were out of date as they cannot satisfy the requirement of these domains.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. It presents an efficient algorithm for ranking of such frequent subgraphs. This proposed ranking method is applied to the FP-growth method for discovering frequent subgraphs.**

***Keywords* – New Normalization-Technique, FP Growth, Ranking Subgraph.**

I. INTRODUCTION

Structured data mining is a major research topic in recent study of Data Mining. One of the most common types 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. To apply the concept of "Lift" into Discounted Cumulative Gain of Subgraph mining algorithms. The application of lift in subgraph mining can be treated as Modified Discounted Cumulative Gain (MDCG). 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. 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.

Graph-based data mining (GDM) is the task of finding novel, useful, and understandable graph-theoretic patterns 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. Currently, there are two major trends in frequent subgraph mining: the Apriori-based approach and the Pattern-growth approach. 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. The only outcome of this FP-Growth method is to discover the frequent subgraphs from the graph data set. 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 Graph gain) 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.

II. RELATED WORK

For the subgraph ranking problem, we present a framework based on an exact and an approximate solution to compute Page Rank on a subgraph. The Ideal Rank algorithm is an exact solution. It assumes that the Page Rank scores of external pages are known. We prove that

the Ideal Rank scores for pages in the subgraph converge to the true Page Rank scores. Since the Page Rank scores of external pages may not be available, we present the ApproxRank algorithm to estimate Page Rank scores for the subgraph. Both Ideal Rank and ApproxRank represent the set of external pages with an external node Λ and extend the subgraph with links to Λ. They also modify the Page Rank transition matrix with respect to (the links to) Λ.

The Ideal Rank and ApproxRank framework formalizes the problem of ranking a subgraph. It allows us to model multiple scenarios where ranking a subgraph is important. Ideal Rank can be used to model scenarios where Page Rank scores of the global graph are known a priori and can potentially be re-used. This includes the case where the subgraph contains the pages that have been updated, or the subgraph represents the pages that represent all the semantic types of interest to a domain expert in Object Rank. ApproxRank can be applied in general to all these problems, when we do not know the Page Rank scores of external pages.

For the subgraph ranking problem, our contributions are as follows:

- We define an efficient algorithm, Ideal Rank, to compute Page Rank scores for a subgraph when Page Rank scores of the external pages are known. Ideal Rank performs a random walk on a modified local graph called the extended local graph, where an external node Λ is added to the local graph. Λ represents the set of pages that are not local. The random walk defined by Ideal Rank utilizes the Page Rank scores of the external pages. The Ideal Rank algorithm can be applied when the Web graph is updated.
- We prove that the IdealRank scores converge to the true PageRank scores for all local pages in the subgraph, and the IdealRank score for the external node Λ converges to the sum of true PageRank scores for all external pages. Since IdealRank converges to the true PageRank, it provides a golden standard for the approximate solution, ApproxRank.
- When PageRank scores of external pages are not known, we define an efficient algorithm ApproxRank to estimate the PageRank scores for a subgraph. The ApproxRank random walk is defined on the extended local graph as well. Since there is no knowledge about the external pages, ApproxRank assumes the authority flow from external pages is equally important. We conduct error analysis for ApproxRank scores. We show that the error of the ApproxRank scores of the subgraph depends on the accuracy of estimation of external page ranking scores.
- We show through empirical results that the ApproxRank ranking accuracy is similar (sometimes superior) to the best competitor SC and it overwhelmingly outperforms the runtime efficiency of SC. We use two real datasets, on which we conduct experiments on three types of subgraph: topic specific subgraph, domain specific subgraph, and BFS subgraph (gathered by a Breadth First Search crawler). We compare ApproxRank against three algorithms, local PageRank, LPR2, and SC. We use two ranking distance metrics to evaluate the accuracy of the algorithms, L_1 distance and the Spearman's Footrule distance. The experiments show that, even without assuming any knowledge about the external pages, ApproxRank behaves well.

III. Modified Discounted Cumulative Gain (MDCG)

Modified Discounted Cumulative Gain (MDCG) is a modified measure of Discounted Cumulative Gain (DCG). Discounted Cumulated Gain is measure of effectiveness of a web search engine algorithm or related applications, often used in information. 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, the gain is accumulated cumulatively The DCG is given by

$$DC = \sum_{i=1}^{p} \frac{2^{r \cdot ali.}}{\log_2 (1}$$

Hence the Modified Discounted Cumulative Gain (MDCG) is obtained using a new measure called "lift", and is defined as

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

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

$$MDC = \sum_{i=1}^{n} \frac{2lift(P_i)}{\log_2 (1+}$$

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

$$Lift = \frac{P[A \cap}{P[A]P}$$

where the obvious extensions to more than two sets.

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.

The function of confidence can also define the Lift.

International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 4, ISSN (Online) 2249 –071X, ISSN (Print): 2278 – 4209

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

Where
Support of a graph is given by
In a given graph F_G, the support F_S^G is defined as

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

And confidence is given by
Given two included subgraph F_b and F_h the confidence of the association rule

$F_b \Rightarrow F_h$ is defined as

$$Conf = \frac{no.of\ graphs\ F\ where\ F_b \cup F_b \in F \in FD}{no.of\ graphs\ F\ where\ F_b \in F \in FD}$$

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

$$Lift = (F_{Lift}^G) = \frac{no.of\ graphs\ F\ where\ F_b and\ F_h}{no.of\ F_b x number\ of\ F_h} = \frac{P[A \cup B]}{P[A]P[B]}$$

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

1) lift value > 1 then A and B depend on each other.
2) lift value < 1 then A depends on the absence of B or vice-versa.
3) lift value close to1 then A and B are independent.

IV. Proposed Algorithm for Ranking the Subgraph

The 'lift' values for every rule of 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 rule 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 $F_1, F_2, \ldots \ldots F_k$, support σ, confidence

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

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

$$Lift = (F_{Lift}^G) = \frac{no.of\ graphs\ F\ where\ F_b and\ F_h}{no.of\ F_b x number\ of\ F_h} = \frac{P[A \cup B]}{P[A]P[B]}$$

The lift also defines the relationship A and B by the condition

1) lift value > 1 then A and B depend on each other.
2) lift value < 1 then A depends on the absence of B or vice-versa.
3) lift value close to1 then A and B are independent.

Step 2: Calculate the MDCG value such as

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

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

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

Given two included subgraph F_b and F_h the confidence of the association rule

$F_b \Rightarrow F_h$ is defined as

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

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.

V. Example

Finding of subgraphs follows the FP-growth method to achieve the most effective pruning. FPgrowth works in a divide-and-conquer way.

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

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 FPtree. This algorithm transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix.

VI. Result 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 is then made to find out the 'lift' measure. Figure1 shows the Lift and MDCG values for the rules.

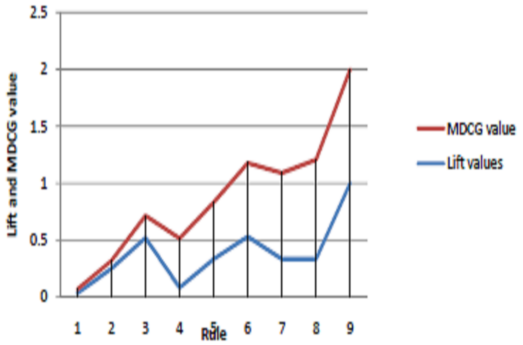

Fig.1. Lift and MDCG values for the Rules

Figure 2 shows the new normalized values for the rules.

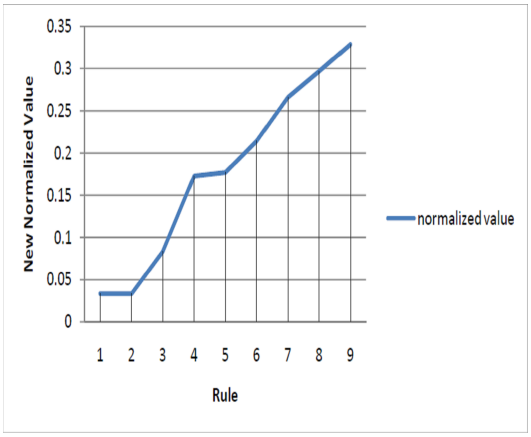

Fig.2. New Normalized values for the rules

Figure 3 shows the ordered lift values for the rules.

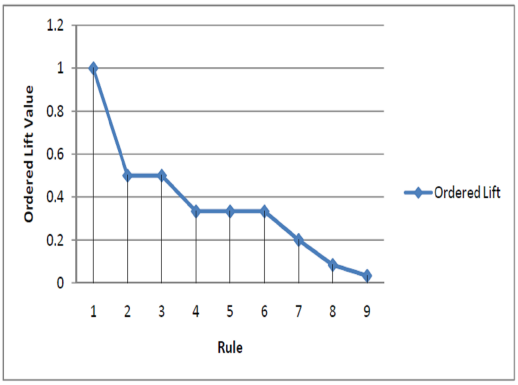

Fig.3. Ordered lift values for the rules.

The main advantage of this technique is that, if similar cases are present, and then there is a possibility of finding the priority of the subgraph. Another advantage of this technique is that no rule will have the same rank even though their lift values are same. But more than one rule have the same normalized values which are very rare. In that case, the ordering of lift values can be taken into consideration for fixing the priority.

VII. Conclusion

The first technique that is used, the normalized values which are obtained from the method using lift measure at each position of large number of frequent subgraphs generated by the FP-Growth method and the second technique is the ordering of the normalized values for ranking of subgraphs. The subgraph data set is explained 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. In web mining, medical data mining and for other similar problems the ranking methods that extended by these proposed techniques. The proposed ranking technique can also be applied for the subgraphs mined from any other methods. It may be 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] S. Narayanan, K. Ramesh Kumar and E. R. Naganathan, "FP-Growth based New Normalization Technique for Subgraph Ranking". Proceedings of International Journal of Database Management Systems (IJDMS), vol. 3, no. 1, February 2011.

[2] 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.

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

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

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

Author's Profile

Rishma Chawla
Asst. Professor, M.Tech. (CSE)
RIET, Phagwara, chawlan02@yahoo.com

Bhanu Arora
M.Tech (CSE)
RIET, Phagwara, bhanuarora874@gmail.com