International Journal of Engineering & Technology, 7 (4.5) (2018) 16-19

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Enhancing Sub Graph Matching With Set Correlation Technique in Large Graph Database

Bharti Nana Durgade, Prof. K. S. Kadam

[1]*Computer Science and Engineering, DKTE's society Textile and Engineering Institute, Ichalkaranji, India.*
[2]*Computer Science and Engineering, DKTE's society Textile and Engineering Institute, Ichalkaranji, India*
**Corresponding author E-mail:* bhartidurgade@gmail.com

Abstract

As the increase of users in social networking sites gives rise to complex relation between the users, which often leads to understand the group of users with similar taste. Now days, this is a one of the growing research area to find the similar users in the relational graph database. Many systems are been introduced to identify the matching sub graphs using similarity between the users. This often yields not much appropriate results due to strict similarity measures. So proposed system uses a technique of identifying correlation between the users for the fired query using pattern identification by incorporating frequent pattern analysis and Pearson correlation which is catalyzed by strong pruning techniques.

Keywords: *Sub Graph, Similarity, Correlation, Pattern Identification*

1. Introduction

Graph similarity has numerous applications in various fields such as social networks, image processing, biological networks, chemical compounds, and computer vision [1][2][3], graph databases have been widely used to model and query complex graph data. The sub graph matching [4] is a fundamental graph query type. Given a query graph Q and a large graph G, a typical sub graph matching query retrieves those sub graphs in G that exactly match with Q in terms of both graph structure and vertex labels[5].
In this paper, the pre-processing is a well-known technique of reducing data size for faster computation. In data mining and machine learning techniques a lots of unused data is presents along with the useful data , so that reduces the efficiency of the task processing. Often these unwanted data misleads the results, so in such applications pre-processing is at core part. Generalized pre-processing has four steps as below
Data Cleaning:- It is a technique of the missing data, smoothing the noisy data and removing the outliers. Data cleaning helps a lot in finding the attributes of interest.
Data Integration: - Data Integration is a technique of gathering the data from multiple stores at a same place so it will get easy to retrieve the data from the same source.
Data transformation: - Data Transformation is a technique putting the data in more appropriate form so it can be used effectively in mining process. Data transformation uses different sub techniques such as normalization, smoothing, aggregation, generalization etc.
Data reduction: - In this technique complex datasets are minimized to its simpler forms without comprising the originality of the data. Stemming is one of the best techniques especially used for the purpose of the data reduction. In this technique root word of derived word is find out in such way that meaning of the word will not change much. Like Stemming there is one more method known as lemmatization which goes in parallel with the stemming but with the slight difference. In case of stemming a set of rules are applied on derived words but here part of the speech is not considered at all.
In contrast in lemmatization part of speech and the meaning of the word is first understood and then root word is obtained.

Often several quantitative variables are measured on each member of a sample. If we consider a pair of such variables, it is frequently of interest to establish if there is a relationship between the two; i.e. to see if they are correlated. We can categories the type of correlation by considering as one variable increases what happens to the other variable:

1. Positive correlation – the other variable has a tendency to also increase;
2. Negative correlation – the other variable has a tendency to decrease;
3. No correlation – the other variable does not tend to either increase or decrease.

Pearson's correlation coefficient [6] to measure the correlation between a query graph and an answer graph. The starting point of any such analysis should thus be the construction and subsequent examination. Pearson's correlation coefficient is a statistical measure of the strength of a linear relationship between paired data. In a sample it is denoted by r and is by design constrained as follows furthermore:

1. Positive values denote positive linear correlation;
2. Negative values denote negative linear correlation;
3. A value of 0 denotes no linear correlation;
4. The closer the value is to 1 or –1, the stronger the linear correlation.

2. Literature Survey

J. R. Ullmann, "An algorithm for sub graph isomorphism,"[7] Sub graph isomorphism can be determined by means of a brute-force

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

tree-search enumeration procedure. In proposed system a new algorithm is introduced that attains efficiency by deductively eliminating successor nodes in the tree search. To assess the time actually taken by the new algorithm, sub graph isomorphism, clique detection, graph isomorphism, and directed graph isomorphism have been carried out with random and with various non-random graphs. However, this algorithm is prohibitively expensive for querying against a large database graph.

L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs,"[8] an algorithm for graph isomorphism and sub graph isomorphism suited for dealing with large graphs. The algorithm is improved here to reduce its spatial complexity and to achieve a better performance on large graphs; its features are analyzed in detail with special reference to time and memory requirements. on a publicly available database of synthetically generated graphs and on graphs relative to a real application dealing with technical drawings are presented, confirming the effectiveness of the approach, especially when working with large graphs. These algorithms do not utilize any indexing structures by pre-processing the database graph.

S. Zhang, S. Li, and J. Yang, "Gaddi: Distance index based sub graph matching in biological networks,"[9] A huge amount of biological data can be naturally represented by graphs, e.g., protein interaction networks, gene regulatory networks, etc. Each graph may contain thousands (or more) vertices. In Proposed system, finding all the matches of a query graph in a given large graph of thousands of vertices, which is a very important task in many biological applications. This increases the complexity significantly. The novel structure distance based approach (GADDI) to efficiently find matches of the query graph. It was also used all paths up to length as index features.

P. Zhao and J. Han, "On graph query optimization in large networks,"[4] In proposed system a high performance graph indexing mechanism, SPath , to address the graph query problem on large networks. SPath leverages decomposed shortest paths around vertex neighborhood as basic indexing units, which prove to be both effective in graph search space pruning and highly scalable in index construction and deployment. The drawback of this proposed system, to accommodate noise and failure in the networks, we need to extend the method to support approximate graph queries.

Y. Tian, R. C. McEachin, C. Santos, D. J. States, and J. M. Patel, "Saga: A sub graph matching tool for biological graphs,"[10] is an approximate sub graph matching technique that finds sub graphs in the database that are similar to the query, allowing for node mismatches, node gaps, and graph structural differences. SAGA provides a powerful tool for biomedical text comparison. The disadvantages are that one has to maintain a database of small structures and that it is query based. In applications such as graph mining in biological networks, it‟s possible that we want to extract subgraph without having identified queries.

M. Hadjieleftheriou and D. Srivastava, "Weighted set-based string similarity,"[11] Weighted string similarity
queries are useful in applications like data cleaning and integration for finding approximate matches in the presence of typographical mistakes, multiple formatting conventions, data transformation errors, etc. In this problem has semantic properties that can be exploited to design index structures that support very efficient algorithms for query answering.

L. Zou, L. Chen, and Y. Lu, "Top-k sub graph matching query in a large graph,"[12] proposed a top-k sub graph matching problem, the similarity between objects associated with two matching vertices. All vertex similarities are given, and does not exploit set similarity pruning techniques to optimize sub graph matching performance.

3. System Design

Create a graph stored in graph database using neo4j from excel sheet. When the users enter the requirement of the sub graph data in graph database and the query will be passed to pre-processing module. Each pre-processed word is gathered in a vector container called as a Bucket. Patterns of the query words can be identified using power set algorithm. A sub graph is identified based on the semantics of the relationship between the entities. Pearson correlation technique is to identify the correlation between the query words and the vertex comments to extract the sub graph using pruning technique.

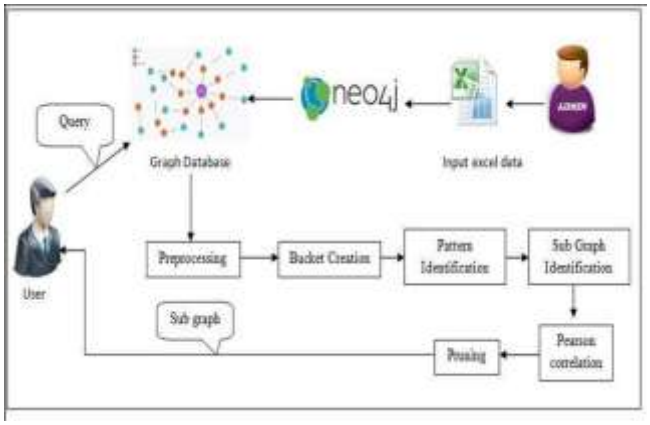


Fig:-1: System Architecture

3.1. Graph Database Creation

The input of usernames and their comments in excel sheet, and thereby creates the graph where every vertices indicates each username and comments are their weight. And then relative users are get connected using the common elements with respect to their weight to create a proper graph in neo 4j graph data base.

3.2. Query Pre-Processing

Users enter their requirement for the sub graph data in graph database as a query and the query will be passed to pre-processing module. The string is processed to its basic meaning words by the following four main activities: Sentence Segmentation, Tokenization, Removing Stop Word and Word Stemming.

3.3. Bucket Creation

The matrix space translation is applied to create combination of words of the keywords which eventually enhances the process of similarity search. Here each pre-processed word is splitting and combining from the third character till the length of the word. Then all this words are gathered in a vector container called as the Bucket.
For Example: Nagpur word is having bucket like
{ Nag, Nagp, Nagpu, Nagpur}

3.4. Pattern Identification

Patterns of the query words can be identified with below mentioned steps
1. The power set is the set of all subsets of a set.
Example,
The power set of the set {a, b, c} consists of the sets: {}
{a}
{b}

{c}
{a,b}
{a,c}
{b,c}
{a, b, c}

This can be more clearly explained as follows. Consider user fires a query like "Studying in Mumbai for Btech" , So after pre-processing the query becomes " Study Mumbai Btech" and now these three qwords form many patterns as mentioned below

Pattern 1 ---{ Study}
Pattern 2 ---{ Mumbai}
Pattern 3 ---{Btech}
Pattern 4 ---{ Study, Mumbai}
Pattern 5 ---{ Study, Btech}
Pattern 6 ---{ Mumbai, Btech}
Pattern 7 ---{ Study Mumbai Btech }

Now these patterns are been used to identify the relationships of the nodes to find the proper sub graph.

2. Power set algorithm

A subset can be represented as an array of boolean values of the same size as the set, called a characteristic vector. Each boolean value indicates whether the corresponding element in the set is present or absent in the subset.
This gives following correspondences for the set {a, b, c}:
[0, 0, 0] = {}
[1, 0, 0] = {a}
[0, 1, 0] = {b}
[0, 0, 1] = {c}
[1, 1, 0] = {a, b}
[1, 0, 1] = {a, c}
[0, 1, 1] = {b, c}
[1, 1, 1] = {a, b, c}
The algorithm then simply needs to produce the arrays shown above. The simplest way to do this is to just count in binary.

3.5. Sub Graph identification

All the frequent words are been checked with the existing vertices of the graph data base and returns matched vector.

3.6. Correlation Matching

The resultant vector obtained from the last step is been feeding to the Pearson correlation to identify the correlation between the query words and the vertex comments to extract the sub graph.
Pearson correlation can be represented with an equation as follows

$$r = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{10}}{\sqrt{(\sum x_i^2 - \frac{(\sum x_i)^2}{10})}\sqrt{(\sum y_i^2 - \frac{(\sum y_i)^2}{10})}}$$

Where x and y represent the vectors of the semantic words in our case it is

X={„s‟,‟t‟,‟u‟,‟d‟,‟y‟}
Y={„s‟,‟t‟,‟u‟,‟d‟,‟y,}

So this equation yields the correlation result as 1.
Correlation value nearer to 1 represents more similar, Nearer to 0 represents dissimilar.

3.7. Pruning

First unwanted vertices from the sub graph is been removed using horizontal and vertical pruning techniques, based on the support of the graph vertices.

4. Snapshot

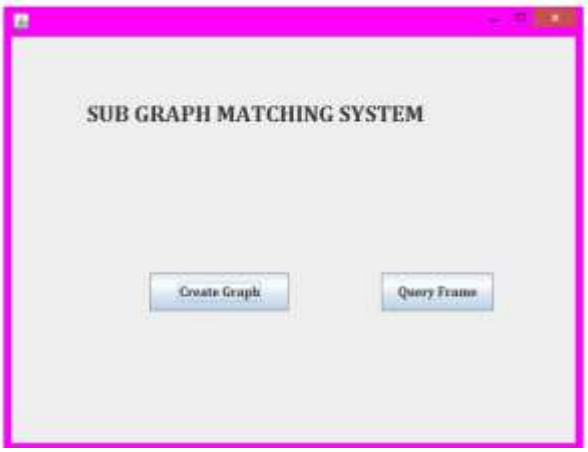


Fig.2: Main Frame

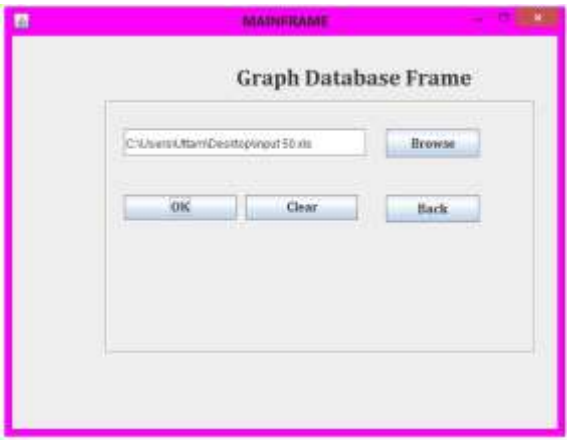


Fig.3: Graph Database Frame

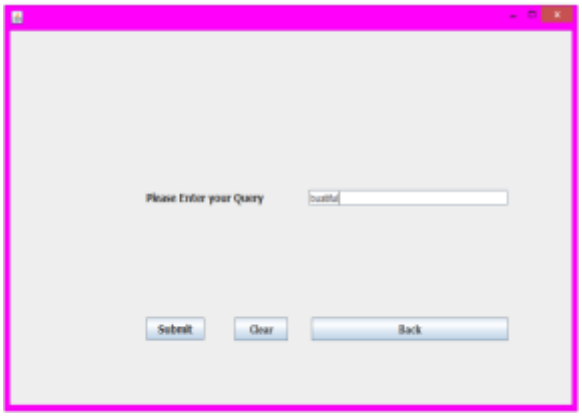


Fig.4: Query Frame

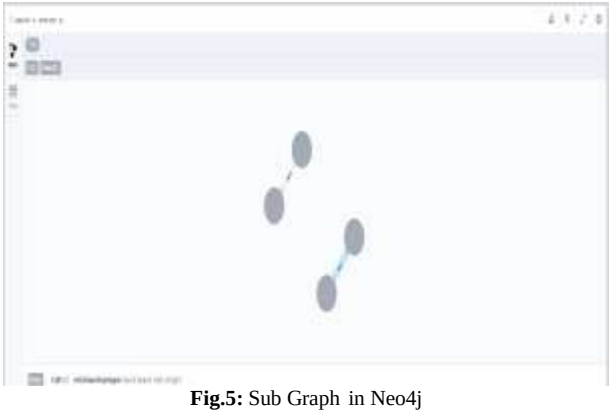


Fig.5: Sub Graph in Neo4j

5. Conclusion

In this paper, we study the problem of enhance the process of sub graph matching and extraction for the given query in large graph database by using pattern identification and correlation matching process with pruning technique. We propose increase the efficiency of sub graph search and improve the real time response.

Acknowledgement

The authors would like to thank the researchers as well as publishers for making their resources available. We are very much thankful to our principal Dr. P.V. Kadole & HOD Prof. Dr. D. V. Kodavade. We are also thankful to my project guide prof. K. S. Kadam. We are thankful to the authorities of shivaji university Kolhapur, for their constant guidelines and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

References

[1] B. Cui, H. Mei and B. C. Ooi, "Big data: The driver for innovation in databases," Nat. Sci. Rev., vol. 1, no. 1, pp. 27–30, 2014.
[2] J. Cheng, J. X. Yu, B. Ding, P. S. Yu, and H. Wang, "Fast graph Pattern matching," in Proc. Int. Conf. Data Eng., 2008, pp.913–922.
[3] X. Zhu, S. Song, X. Lian, J. Wang, and L. Zou, "Matching heterogeneousevent data," in Proc. ACM SIGMOD Int. Conf. Manage. Data,2014, pp. 1211–1222.
[4] P. Zhao and J. Han, "On graph query Endowment, vol. 3, nos. 1/2, optimization in large networks," Proc. VLDB pp. 340–351, 2010.
[5] Y. Tian and J. M. Patel, "Tale: A tool for approximate large Graph matching," in Proc. 24th Int. Conf. Data Eng., 2008, pp. 963–972.
[6] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right Interestingness measure for association patterns. In KDD,pages 32–41, 2002.
[7] J. R. Ullmann, "An algorithm for subgraph isomorphism," J. ACM, vol. 23, no. 1, pp. 31–42, 1976.
[8] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," IEEE Trans.Pattern Anal. Mach. Intell., vol. 6, no. 10, pp. 1367– 1372, Oct. 2004.
[9] S. Zhang, S. Li, and J. Yang, "Gaddi: Distance index based sub-graph matching in biological networks," in Proc. 12th Int.Conf. Extending Database Technol.:Adv. Database Technol., 2009, pp. 192–203.
[10] Y. Tian, R. C. McEachin, C. Santos, D. J. States, and J. M. Patel ,"Saga: A subgraph matching tool for biological graphs," Bioinformatics, vol.23, no. 2, pp. 232–239, 2007.
[11] M. Hadjieleftheriou, A. Chandel, N. Koudas, and D. Srivastava, "Fast indexes and algorithms for set similarity selection queries," in Proc. 24th Int. Conf. Data Eng.,pp.2008, 267–276.
[12] L. Zou, L. Chen, and Y. Lu, "Top-k subgraph matching query in a large graph," in Proc. ACM 1st PhD Workshop CIKM, 2007, pp. 139– 146.
[13] Liang Hong, Lei Zou, Xiang Lian and Philip S. Yu, "Subgraph Matching with Set Similarity in a Large Graph Database" IEEE Transactions on Knowledge and Data Engineering, Vol.27, No.9, September 2015