International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

IJIEMR Transactions, online available on 17th Sept 2021. Link :http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-09

DOI: 10.48047/IJIEMR/V10/I09/06

Title **Parallel processing Cache Algorithm (PPCA) for efficient Frequent Sub-Graph Mining**

Volume 10, Issue 09, Pages: 27-36

Paper Authors

M. Sailaja, Dr. CVPR Prasad

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Parallel processing Cache Algorithm (PPCA) for efficient Frequent Sub-Graph Mining

[1]M. Sailaja
Research Scholar, Acharya Nagarjuna University, Guntur
sailu.team@gmail.com

[2]Dr. CVPR Prasad
Supervisor, Acharya Nagarjuna University, Guntur
prasadcvpr@gmail.com

Abstract:

Frequent sub-graph mining (FSM) is most widely defined on identifying the sub-graphs in a given input that obtain in more number of times than a given value. For graph data FSM plays the significant role in feature mining. FSM is mainly focused on analyzing the real world graphs, estimates the structure and properties of a given graph may impact some application and enhanced models are to be developed to generate the real graphs that will match the patterns that set up in real time graphs. Many traditional graph mining algorithms are used to process the large and complex datasets. But they didn't get the accurate graph mining results. It is very important to find the accurate graph mining patterns to solve the several complex issues. In this paper, the Parallel processing Cache Algorithm (PPCA) is introduced to find the FSM very effectively by removing noise within the given dataset. To improve the performance of the proposed system an enhanced pre-processing technique is utilized to remove the noise data and cleaning the unwanted data. This can improve the time (s) and retrieving the number huge number of results.

Keywords: graph mining (GM), FSM, pre-processing, PPCA

Introduction

A powerful way to represent relations among various entities of a system network such as social network, biological network, chemical compound, etc is in the form of Graph. The entities are represented by vertices. A subset of a graph is called sub-graph i.e. subpart of the system network. A sub-graph S (E', V') of a graph G (E, V) is defined as a graph where E' ⊆ E and V' ⊆ V. Sub-graph of a given graph is helpful in solving real time problem efficiently.

In a large network, various types of possible sub-graphs are there and their significances are likely to depend on their application. Such search for different/various types of sub-graphs and their significances have attracted researchers to propose different mining algorithm. Hence, a considerable research effort has been carried out towardsmining of graph data for different types of sub-graphs such as 1) Frequent sub-graph 2) Correlated graph pattern 3) Optimal graphpattern 4) Cliques 5) Approximate graph pattern etc. The algorithms for mining aforesaid type of sub-graphs are intended for static graph(s) only, i.e. a type of graph not changing with time. However in recent years there are many developments in virtual social network platforms such as LinkedIn, MySpace, Facebook, YouTube, Twitter etc.. The data from these social networks is also represented as graphs that evolve with respect to time. Such graphs evolving with time are dynamic in nature. Information present due to such dynamic nature of graph cannot be mined by static graph mining techniques. In context to mine such information, researchers have focused to mine sub-graphs or communities that evolve with time. It is observed that these sub-graphs are widely studied for the companion system of popular social networking sites like Facebook and Myspace, the Enron email organized, and phone calling networks, co-origin and reference system, web-page linking networks and so forth. Therefore to mine significant pattern in such networks a new branch of graph mining has emerged i.e. dynamic graph mining. Here, dynamic graph represents a set of varying graph at each successive time steps.

Rest of this paper is organized as follows. Section 2 explains the existing frequent graph mining techniques. Section 3 introduces the new proposed methodology. Section 4 experimental results and implementation. Section 5 describes the conclusion.

Literature Survey

In 2014, the case of limited memory, a memory efficient data structure called DSMatrix is proposed by Cameron et al. [1] for dynamic graph (weighted undirected). The DSMatrix is used to mine frequent subgraphs. Two frequent pattern mining algorithms are proposed on DSMatrix i.e. i) tree-based horizontal mining algorithm and ii) vertical mining algorithm.

In 2014, Vo et al. [2] have proposed a calculation to productively mine every continuous sub-graphs based on gSpan[] and to mine sub-graphs in parallel. It uses multi-core processor architecture for parallelism. In 2015, Neil Shah et al., [3] introduce a dynamic graph summarization algorithm based on the principle of MDL i.e Minimum Description Length which is a practical version of Kolmogorov Complexity. It specifies the process of encoding model andthe errors associated with it so that it can be used to reconstruct the original dynamic graph with a lossless strategy. In 2016 Alissio Conte et al. [9] proposed a technique for finding maximalclique in distributed environment for large networks. The approach is based on decomposition strategy by lowering the size of blocks to achieve efficiency.

Lam et al., [5] proposed a novel graph mining algorithm, MIGDAC (Mining Graph DAta for Classification), that applies graph theory and an interestingness measure to discover interesting sub-graphs which can be both characterized and easily distinguished from other classes. Applying MIGDAC to the discovery of specific patterns of chemical compounds, we first represent each chemical compound as a graph and transform it into a set of hierarchical graphs. This not only represents more information that traditional formats, it also simplifies the complex graph structures. We then apply MIGDAC to extract a set of class-specific patterns defined in terms of an interestingness threshold and measure with residue analysis. The next step is to use weight of evidence to estimate whether the identified class-specific pattern will positively or negatively characterize a class of drug.

Lakshmi et al., [6] proposed the new method for graph representation, that can be used to represent both directed and undirected graphs, with unique node labels. Our algorithm incorporates two new ideas, (i) representation of graphs as a list key, value pairs and (ii) use of an optimization parameter, graph threshold - the number of intermediate frequent sub graphs that are to be in the main memory for further mining of frequent sub graphs. We evaluated the performance of our algorithm using synthetic data set and real time Gnutella network dataset. Qiao et al., [7] proposed a multi-thread frequent subgraph mining algorithm and achieved considerable acceleration in the experiments. A parallel frequent subgraph mining algorithm named PTRGRAM (Parallel Transaction based Graph Mining) which can take full advantage of the multi-core performance of current processors was proposed. In this algorithm, the data synchronization between multiple threads is based on the producer-consumer model. In addition, to speed the support computing, the embedding node list is introduced for optimization. Zhang et al., [8] proposed GraphLib, a parallel graph mining library, based on a BSP (Bulk Synchronous Parallel) service over joint cloud computing which was proposed in our prior work. We firstsummarize the features of commonly-used graph mining algorithms,and present our approaches for parallelizing typical graph mining algorithms. GraphLib includes 17 parallel graph mining algorithms that can be used in 3 scenarios. We evaluate the performance of 4 typical parallelgraph algorithms in GraphLib on three real-world datasets. Our parallelized algorithms can achieve sub-linear scalability. Chaudhary et al., [9] explained about the efficiency of graph mining is the efficient way to analyze such data and get the required information. As the nature of data involved is dynamic in nature i.e changes w.r.t time, therefore to mine this type of dynamic data is a topic of interest these days in the form of dynamic graph mining, where time considerations are explicitly incorporated in a set of graph. Dynamic graph is a set of consecutive timestamp graphs which can be analyzed based on different patterns like frequent, periodic, trend motif, clique etc. The author represents overview of different algorithm used to mine and analyze the evolving patterns in dynamic graph.

Robust Pre-processing Approach

This is very important step in graph mining that helps to process the dataset by removing the missing values, irrelevant data and extract the accurate meaningful data from the dataset. In DBLP dataset which is collected from kaggle, the pre-processing technique cleans the raw data which is creates the better platform for training models. By using this step, the data is converted to understandable and better format that can be readable by the algorithms.

Some transformation algorithms applying to original data can be useful for binning. Standardization method is a widely-used technique for numerous data mining algorithms to resolve the problem of different data distributions. Quantile Transformation (QTF), MinMaxScaler (MMS), and logarithmic computations scalers are considered to convert data before binning. Quantile Transformation is implemented to combine with EQW in these experiments. QTF is considered as a robust pre-processing technique because it can reduce the effect of the outliers in DBLP dataset. Samples in test and validation sets which are smaller or larger than the fitted range then will be assigned to the bounds of the output distribution. Another algorithm illustrated in this study is MinMaxScaler, to make a comparison with QTF and logarithmic computations. MinMaxScaler converts each feature to a given range by (1) and (2) formulas:

$$X_{std} = \frac{X - X.\min}{X.\max - X_{min}} \quad (1)$$

$$X_{scaled} = X_{std} * (\max - \min) + \min \quad (2)$$

Functions which perform the transformation as above are now available in scikit-learn library.

Searching Algorithms in Graph Mining

The sequential search algorithm iterates through each item in our data structure in search for a specific value. If the current item matches, we can return, else we must continue to the next item. In the worse case, this requires that we search through all items because in an unsorted structure, we cannot Say whether an untested value is the value we are searching for in specific values and more execution time is required. The binary search algorithm in data mining begins by comparing the target value to the value of the middle element of the sorted array. If the target value is equal to the middle element's value, then the position is returned and the search is finished. If the target value is less than the middle element's value, then the search continues on the lower half of the array; or if the target value is greater than the middle element's value, then the search continues on the upper half of the array. The techniques and methods and their applications in bioinformatics study, focusing on data integration, text mining and graph-based data analysis [10]. A model that is utilized to represent the master search schema, and an effective interface extraction algorithm based on the hierarchical structure of the web and pattern is developed to capture the rich semantic relationships of the online bioinformatics data sources. The k-means algorithm is a simple iterative method to partition a given dataset into a user specified number of clusters, k. This algorithm has been discovered by several researchers across different disciplines. Apriority is a seminal algorithm for finding frequent item sets using candidate generation [11]. One of the most popular data mining approaches is to find frequent item sets from a transaction dataset and derive association rules [12]. Moreover few others algorithm exist in data mining.

Searching Algorithms

```
procedure keysearch(X)
begin
if(searchmethod(X)) then
score=eval(X)
report searchmethod and score
else
foreach myval X(i) of X
keysearch(X(i))
endfor
endif
end
```

Frequent Graph Pattern

D is the given dataset, find sub-graph g, s.t.

$$freq(g) \geq \theta$$

Where $freq(g)$ is the percentage of graphs in D that contain g.

Optimal Graph Pattern

D is the given dataset and an objective function F (g), graph pattern is identified as

$$g^{*} = \arg max_{g} F(g)$$

Task Graph Model

In the task graph model, parallelism is expressed by a task graph. A task graph can be either trivial or nontrivial. In this model, the correlation among the tasks are utilized to promote locality or to minimize interaction costs. This model is enforced to solve problems in which the quantity of data associated with the tasks is huge compared to the number of computation associated with them. The tasks are assigned to help improve the cost of data movement among the tasks.

Here, problems are divided into atomic tasks and implemented as a graph. Each task is an independent unit of job that has dependencies on one or more antecedent task. After the completion of a task, the output of an antecedent task is passed to the dependent task. A task with antecedent task starts execution only when its entire antecedent task is completed. The final output of the graph is received when the last dependent task is completed.

Parallel processing Cache Algorithm (PPCA)

A parallel algorithm for this problem creates *N* tasks, one for each point in *X*. The *i* th task is given the value $A_i^{(0)}$ and is responsible for computing, in *T* steps, the values$A_i^{(1)}, A_i^{(2)}, A_i^{(3)}, \dots\dots, A_i^{(x)}$. Hence, at step *t*, it must obtain the values $A_{i-1}^{(t)} and A_{i+1}^{(t)}$and from tasks *i-1* and *i+1*. We specify this data transfer by defining channels that link each task with ``left'' and ``right'' neighbors, and requiring that at step *t*, each task *i* other than task 0 and task *N-1*.

1. sends its data $A_i^{(t)}$ on its left and right outputs,
2. receives $A_{i-1}^{(t)} and A_{i+1}^{(t)}$ and from its left and right inputs, and
3. Uses these values to compute $A_i^{(t+1)}$.

Dataset Description

DBLP originally stood for Database systems and Logic Programming. DBLP is a bibliographic database for computer sciences. The main problem in DBLP is the assignment of papers to author entities. This dataset provides bibliographical information about computer science journals and proceedings. It includes 50,000 objects.

- Provides efficient algorithms for locating hidden patterns in information.
- Finds marginal sets of knowledge

- Evaluates significance of knowledge,
- it's simple to know,
- Offers easy interpretation of obtained results,
- Most algorithms supported the rough pure mathematics are significantly fitted to data processing.

Experimental Results

The experimental results are conducted on synthetic dataset and Netbeans 8.0.2 as IDE, Java as programming language. Ram 8 GB and 1TB hard drive is required to process this proposed algorithm.

The overall comparative results are shown in table 1.

Algorithm	**No of Results of One keyword**	**Processing Time (MS)**
Disk based Technique (DBT)	1461	3.155
Partition Based Technique (PBT)	2621	4.248
FSGM-CRD	2621	3.352
CRD-PPA	2621	2.970
Ensemble Distributed Search-FSGM-CRD Compressed Cache	3190	1.580
Parallel processing Cache Algorithm (PPCA)	3312	0.59

Table 1 shows the performance of the proposed system with total no of results from the DBLPL dataset for one keyword (network) and total processing time (ms).

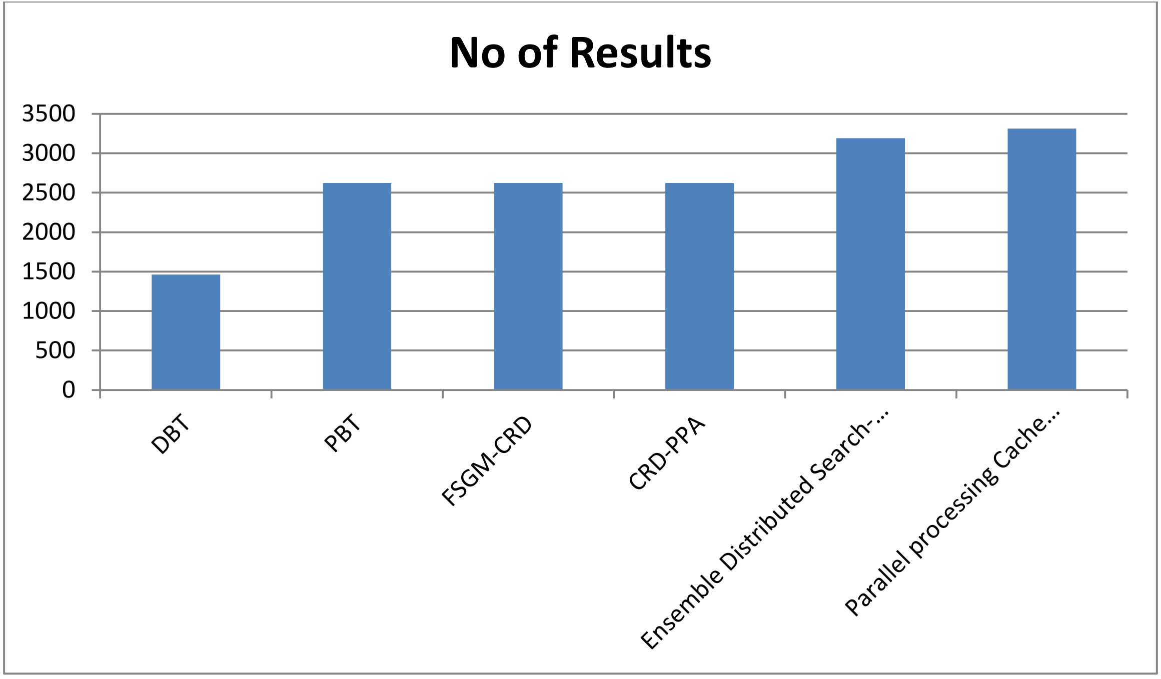

Figure 2: Comparative performance in terms of number of frequent graphs of various FSGM Algorithms

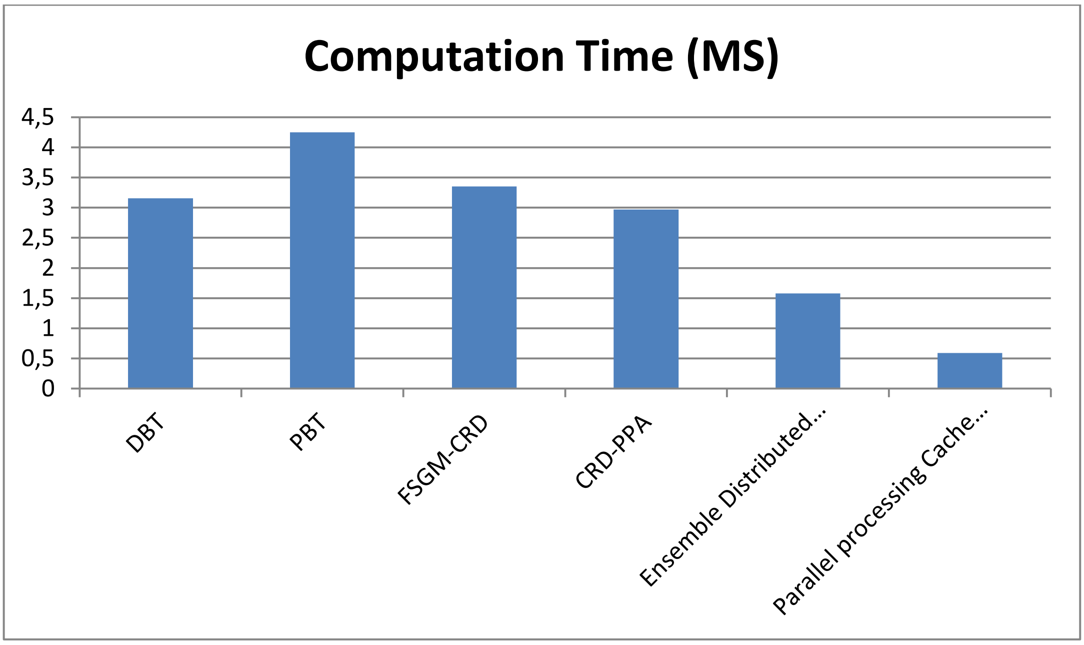

Figure 3: Comparative performance in terms of time taken to complete one keyword frequent graphs of various FSGM Algorithms

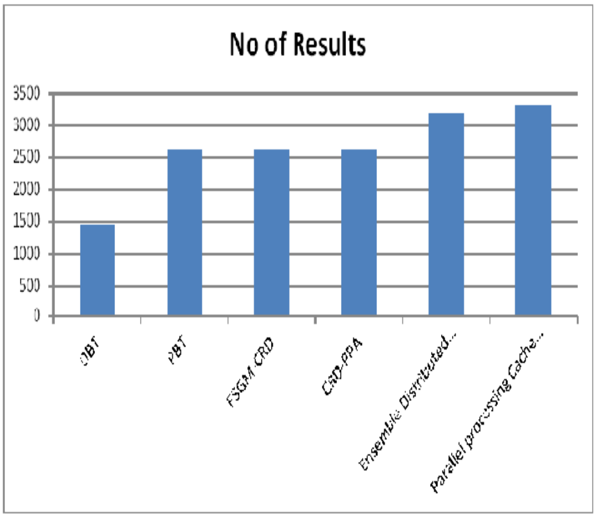

Figure 2: Comparative performance in terms of number of frequent graphs of various FSGM Algorithms

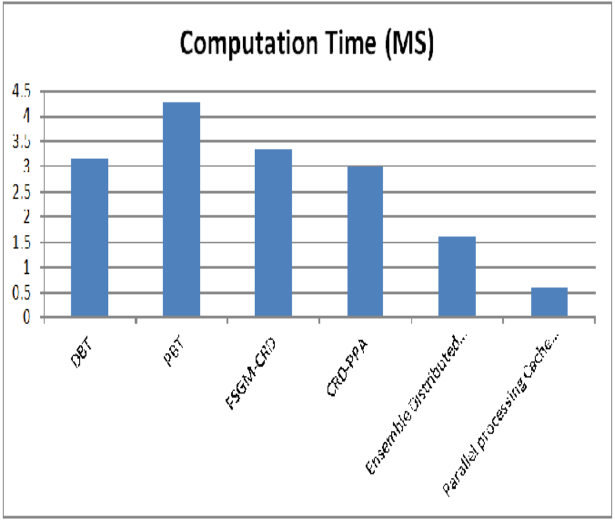

Figure 3: Comparative performance in terms of time taken to complete one keyword frequent graphs of various FSGM Algorithms

Conclusion

In this paper, an improved frequent sub-graph mining with efficient searching algorithm. The algorithm works very efficiently on large datasets. The DBLP is large dataset which consists of 1000's of data. Parallel processing algorithm is with enhanced cache gives the fast performance for searching and showing the results. An efficient searching algorithm is worked on large datasets to search the data accurately and fastly.

References

[1] J.J. Cameron, A. Cuzzocrea, F. Jiang, & C.K. Leung. Frequent pattern mining from dense graph streams. In Proc. EDBT/ICDT Workshops 2014, pp. 240-247.

[2] Vo, B., Nguyen, D., & Nguyen, T. L. (2015). A parallel agorithm for frequent subgraph mining. In Advanced Computational Methods for Knowledge Engineering (pp. 163-173). Springer, Cham.

[3] Shah N, Koutra D, Zou T, Gallagher B, Faloutsos C. Timecrunch: Interpretable dynamic graph summarization. InProceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

2015 Aug 10 (pp. 1055-1064). ACM.

[4] Conte A, Grossi R, Marino A, Versari L. Sublinearspace bounded-delay enumeration for massive network analytics: Maximal cliques. In43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016) 2016 Jul 11 (Vol. 148, pp. 1-148).

[5] W. W. M. Lam and K. C. C. Chan, "A Graph Mining Algorithm for Classifying Chemical Compounds," 2008 IEEE International Conference on

Bioinformatics and Biomedicine, 2008, pp. 321-324.

[6] K. Lakshmi and T. Meyappan, "Efficient mining of frequent sub graphs," 2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), 2017, pp. 1-7.

[7] F. Qiao, Y. Zhang, J. Deng, Z. Ding and A. Li, "A Parallel Algorithm for Graph Transaction Based Frequent Subgraph Mining," 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC), 2020, pp. 351-355.

[8] K. Zhang, Y. Fang, Y. Zheng, H. Zeng, L. Xu and W. Wang, "GraphLib: A Parallel Graph Mining Library for Joint Cloud Computing," 2020 IEEE International Conference on Joint Cloud Computing, 2020, pp. 9-12.

[9] N. Chaudhary and H. K. Thakur, "Survey of Algorithms based on Dynamic Graph Mining," 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2018, pp. 393-399.

[10] Xiaohua Hu," Data Mining and Its Applications in Bioinformatics: Techniques and Methods", 2011 IEEE International Conference on Granular Computing.

[11] John L. Pfaltz, Christopher M. Taylor "Closed Set Mining of Biological Data," BIOKDD02:Workshop on Data Mining in Bioinformatics (with SIGKDD02 Conference).

[12] Jiawei Han, Micheline Kamber,Jian Pei "Data Mining Concepts and Techniques Third EditionMorgan Kaufmann Publishers is an imprint of Elsevier.225 Wyman Street, Waltham, MA 02451, USA.