Ap-FSM: A parallel algorithm for approximate frequent subgraph mining using Pregel

Vandana Bhatia*, Rinkle Rani

Department of Computer Science and Engineering, Thapar University, Patiala, India

**A R T I C L E   I N F O**

Article history:
Received 30 August 2017
Revised 6 April 2018
Accepted 7 April 2018
Available online 9 April 2018

Keywords:
Pattern mining
Frequent subgraphs
Graph isomorphism
Large graphs
Sampling

**A B S T R A C T**

Large graphs are scale-free, ubiquitous having irregular relationships and non-trivial topology. Frequent subgraph mining is a popular method for knowledge extraction from graphs. Most of the existing frequent subgraph mining algorithms are centralized algorithms that cannot handle a single large graph efficiently and incur high communication cost. However, to make the task of subgraph mining less expensive computationally, approximate subgraph mining can be applied which will capture similar structure subgraphs as of exact subgraph mining. In this paper, we propose an approximate subgraph mining algorithm named Ap-FSM implemented on distributed graph environment Pregel. The working of Ap-FSM is divided into three phases. The first phase selects the representative graph from the original graph while preserving the original graph properties. The second phase efficiently performs subgraph extension. Phase 3 introduces a novel two-step optimization for performing subgraph pruning. Analyzing such large graph data will be beneficial from the perspective of expert and intelligent systems, as discovered patterns can be used for knowledge discovery and decision making. To evaluate the performance of Ap-FSM, experiments are performed over three real life datasets having up to billion edges. The results show that the proposed Ap-FSM significantly outperforms the state-of-art frequent subgraph mining algorithms and overcome the challenges of performing frequent subgraph mining on a massive large graph. It is also shown that Ap-FSM achieves high scalability and speedup in distributed graph environment and is highly accurate in finding frequent subgraphs from a single large graph.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Big data has a good potential to transfigure most of the aspects of society. But, harvesting the valuable knowledge from the huge amount of data originated from various sources is still a remarkable challenge in the field of knowledge discovery and data mining (Aggarwal & Wang, 2010). Representing data as graphs to understand the relationships among data objects is gaining high interest. But, with the increase in the volume of data, the size of the graph is also increasing. The researchers are showing a lot of interest in applying graph analytics on data available from social media (Bonato & Tian, 2013; Halder & Lee, 2017; Leung, Hui, Li, & Crowcroft, 2009), biological networks (Hill, Srichandan, & Sunderraman, 2012; Karci, 2009), business networks (Wang, Wang, Zhao, & Tan, 2015) etc. Frequent subgraph mining (FSM) is a popular method in the field of graph analytics.

FSM can be either performed on graph transactions or on a single large graph. Most of the existing works focus on graph transactions for performing FSM. But, finding frequent subgraphs from a single massive graph is far more challenging because of high computations cost for performing NP-Complete subgraph isomorphism in a single large graph. In many applications, getting the faster approximate results is significant than getting the exact results. If that large graph is analyzed at small scale, the useful insights can be extracted in lesser time. Analyzing large graph at small scale, while capturing the original graph properties is convenient and relatively easy task for analysis. However, the sample of large graph can be still enough large that could not be efficiently handled by centralized system. Traditional FSM algorithms are centralized and cannot handle such massive large graph. Also, creating a representative but a small sample from a massive large graph is a perplexing task.

Large graphs are handled in distributed environment for efficient processing but applying FSM in distributed environment is challenging due to variation in support values among various computing nodes. In addition, the real-world graphs are dense in nature and traditional FSM algorithms fail in handling dense graphs. Thus, in order to overcome the drawbacks faced

---

* Corresponding author.

E-mail addresses: vbhatia91@gmail.com (V. Bhatia), raggarwal@thapar.edu (R. Rani).

https://doi.org/10.1016/j.eswa.2018.04.010
0957-4174/© 2018 Elsevier Ltd. All rights reserved.
by solely parallelizing the existing traditional frequent subgraph mining approaches, we propose a novel approach called Ap-FSM (Approximate Frequent Subgraph Mining), to iteratively extract frequent subgraphs in a distributed environment. Ap-FSM performs sampling on original single massive graph to select the representative graph for making the task of frequent subgraph mining computationally less expensive in distributed environment. Ap-FSM is designed and implemented on parallel graph processing framework Pregel. Pregel is proficient in dealing with very large graphs and efficiently implementing parallel graph algorithms.

To our knowledge, Ap-FSM is the first algorithm to exploit the aggregation of graph sampling and parallelization for the task of FSM. A number of existing research explored either distributed algorithms (Bhatia & Rani, 2017; Jain, Flick, Pan, Green, & Aluru, 2017) or sampling (Mitzenmacher, Peng, & Xu, 2015; Saha & Al Hasan, 2014; Zou & Holder, 2010) for handling large graph. But, both of the aforementioned approaches have been somewhat orthogonal till now. In Ap-FSM, the shortcoming of either approach is compensated by the advantages of the other. To minimize communication overhead, in-memory computations are performed. The distributed graph data are mined in parallel and their results are collected afterwards.

In summary, following contributions are made by the proposed approach

- An efficient parallel algorithm for approximate frequent subgraph mining on a single large graph.
- An efficient technique named G-Samp for the selection of an approximate subgraph while capturing the original graph properties for convenient and relatively easy analysis.
- A two-step approach for optimizing the subgraph isomorphism task and subgraph pruning.

The rest of the paper is structured as follows. Section 2 presents the literature review of the existing research in the area of frequent subgraph mining. The preliminaries and problem definition are given in Section 3. Section 3 also detail out the Pregel framework. The proposed approach is explained in Section 4. Section 5 discusses the performance evaluations, and the conclusion of the paper is given in Section 6.

### 2. Related work

Frequent subgraph mining has become an imperative area of research because of wide variety of graph mining applications including chemical data analysis (Yan & Han, 2002), drug discovery, communication networks (Hsieh & Li, 2010), computational biology (Chen et al., 2017), Web services (Liu, Zhu, Chu, & Ma, 2018), and many others (Pears, Pisalpanus, & Koh, 2015).

The frequent subgraph mining algorithms can be classified as apriori-based and pattern-growth based approaches. In apriori based approaches, size-(k+1) subgraphs are generated from the set of frequent k subgraphs. Here, breadth first search strategy is preferred because graph is represented in level-wise manner. Many apriori-based algorithms for mining frequent subgraphs have been proposed in literature (Kuramochi & Karypis, 2001; Kuramochi & Karypis, 2004b; Nijssen & Kok, 2001). Kuramochi and Karypis (Kuramochi & Karypis, 2004a) proposed an apriori based algorithm FSG which uses an edge-based method to grow the size of subgraph by single edge during each iteration. In FSG, the number of edges characterizes the size of graph. FSG was further modified to find frequent patterns in geometric graphs in 2007 (Kuramochi & Karypis, 2007). Although apriori-based algorithms find the frequent subgraphs efficiently in small graphs, they have considerable overheads in joining two subgraphs of size-k to generate candidates of size-(k+1). To overcome these overheads, non-apriori based algorithms were introduced, which were basically based on pattern-growth approach.

In Pattern growth approach, a new edge e is added to the subgraph for generating a new candidate subgraph g. By repeating this procedure, extension of frequent subgraphs is performed iteratively in all possible manners. But, during the generation of frequent subgraph it may possible that same subgraph is generated many times (Cook & Holder, 2007). This problem resulted in design of many other algorithms such as FFSM (Huan, Wang, & Prins, 2003), MOFA (Borgelt & Berthold, 2002), gSpan (Yan & Han, 2002), Gaston (Nijssen & Kok, 2004) and SUBDUE (Ketkar, Holder, & Cook, 2005).

CloseGraph (Yan & Han, 2003) introduced by Yan and Han in 2003 is a closed graph mining algorithm that uses pattern growth approach. It eliminates the generation of the unnecessary subgraphs and increases the mining efficiency up to greater extent, specifically in the case of large patterns. Furthermore, certain algorithms were proposed in literature for handling uncertain graphs also (Wu, Ren, & Sheng, 2017; Zou, Li, Gao, & Zhang, 2010).

Some work have also been done to find the maximal frequent patterns in graph (Huan, Wang, Prins, & Yang, 2004; Ozaki & Etoh, 2011; Thomas, Valluri, & Karlapalem, 2010; Wu et al., 2017). Most of them have used the concept of edit distance for allowing structural differences (Flores-Garrido, Carrasco-Ochoa, & Martínez-Trinidad, 2014a, b; Moussaoui & Zaghdoud, 2016). An algorithm named APGM (Jia, Zhang, & Huan, 2011) is proposed to find inexact patterns by using a pre-defined substitution matrix consisting of similarities among labels. Similarity is measured by a product of probabilities. APGM is extended to allow label replacements of both vertices and edges in VEAM (Acosta-Mendoza, Gago-Alonso, & Medina-Pagola, 2012). However, both APGM and VEAM require prior knowledge in the form of substitution matrix which is available in fewer applications only. Some other algorithms such as MgVEAM (Acosta-Mendoza, Gago-Alonso, Medina-Pagola, Jesús Ariel Carrasco-Ochoa, & Martínez-Trinidad, 2017) and AMG-Miner (Acosta-Mendoza, Gago-Alonso, Carrasco-Ochoa, Martínez-Trinidad, & Medina-Pagola, 2016) are designed for dealing with multi-graphs. Both MgVEAM and AMG-Miner are also used for image classification.

We intend to find frequent subgraphs from a single large graph. For a single graph, only few approaches for approximate subgraph mining algorithms were proposed in literature (Chen, Yan, Zhu, & Han, 2007; Flores-Garrido et al., 2014a, b). Most of the algorithms work for only small graph datasets and thus, face scalability issue. For finding the patterns from large graph datasets in distributed computing system, Map-Reduce was used by some researchers (Bhuiyan & Al Hasan, 2013; Lin, Xiao, & Ghinita, 2014; Lu, Chen, Tung, & Zhao, 2013). However, they find frequent subgraphs from graph transactions. Only few research have been conducted for mining frequent subgraphs from a single graph (Bringmann & Nijssen, 2008; Flores-Garrido et al., 2014a, b).

Also, most of the existing research considers only sparse graphs for mining (Kuramochi & Karypis, 2005; Reinhardt & Karypis, 2007). But, the real world graphs are dense and have high average degree. The first algorithm which finds the frequent subgraphs from large dense subgraph is GRAMI (Elseidy, Abdelhamid, Skiadopoulos, & Kalnis, 2014). It uses constraint satisfaction problem for finding the minimum set of instances for satisfying support threshold. AGRAMI is proposed by extending GRAMI to find approximate subgraphs. However, both of them are centralized algorithms that cannot handle very large graph. For handling the large graph in distributed environment, many subgraph mining algorithms using Map-Reduce were proposed (Babu, 2016; Wang et al., 2016). A density based partitioning technique was proposed for large scale subgraph mining using Map-Reduce framework for balancing computational load (Aridhit, D’Orazio, Mad-
دریافت فوری
متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات