Expert Systems With Applications 94 (2018) 70–80

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Revealing the densest communities of social networks efficiently through intelligent data space reduction

Tao Han[a,b,*], Yu-Chu Tian[b,c,*], Yuqing Lan[a], Fenglian Li[b,c], Limin Xiao[a]

[a] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
[b] School of Electrical Engineering and Computer Science, Queensland University of Technology, GPO Box 2434, Brisbane QLD 4001, Australia
[c] College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China

ARTICLE INFO

*Article history:*
Received 30 January 2017
Revised 20 October 2017
Accepted 21 October 2017
Available online 23 October 2017

*Keywords:*
Densest subgraph
Graph
Space reduction
Sampling
Heuristics

ABSTRACT

The inherent structure and connectivity of a group are important features of social networks. Finding the densest subgraphs of a graph directly maps to revealing the densest communities of a social network. Various techniques, e.g., edge density, *k*-core, near-cliques and *k*-cliques, have been developed to characterize graphs and extract the densest subgraphs of the graphs. However, as extraction of subgraphs with constraints is NP-hard, these techniques face a major difficulty of processing big and/or streaming data sets from social networks. This demands new methods from the expert and intelligent systems perspective for computation of the densest subgraph problem (DSP) with big and/or streaming data. The most recent method for this purpose is the “Sampling” method. It samples the big data sets, thus reducing the data space and consequently speeding up the DSP computation. But the sampled data inevitably miss out many useful data items. A new approach is presented in this paper for accelerated DSP computation with big and/or steaming data through data space reduction without loss of useful information. It uses a sliding window of small graphs with a fixed number of edges. Then, it filters out the least connected edges for each small graph. While the small graphs are processed, subgraphs are incrementally put together to reveal the densest subgraphs. Finally, the data space previously filtered out is checked for recovery of globally important edges. The approach is incorporated with existing subgraph extraction techniques for scalable and efficient DSP computation with improved accuracy. It is demonstrated for four subgraph extraction techniques over four Twitter data sets, and is shown to outperform the “sampling” method.

## 1. Introduction

Analysis of social networks is popularly described as graph mining problems (Nettleton & Salas, 2016; Park, Hwang, & Lee, 2016). The Densest Subgraphs Problem (DSP) is a key issue in large-scale graph mining (Tsourakakis, 2015). For example, social graphs (Wang, Wang, Yu, & Zhang, 2015) reveal the friendships among people. They are used to forecast planning stock for stakeholders (Paletto, Hamunen, & Meo, 2015). Social graphs are also applied in examination of the popularity of two undergraduate classrooms (Webster, Gesselman, & Crosier, 2016). Other examples include those in education (Carolan, 2013) and community detection (Atzmueller, Doerfel, & Mitzlaff, 2016).

The DSP has been studied extensively in data mining and theoretical computer science. Investigations in data mining include those reported in Tatti and Gionis (2015), Sozio and Gionis (2010), Balalau, Bonchi, Chan, Gullo, and Sozio (2015), Tsourakakis (2015) and Tsourakakis, Bonchi, Gionis, Gullo, and Tsiarli (2013). Studies in theoretical computer science are found in references (Andersen & Chellapilla, 2009; Bhattacharya, Henzinger, Nanongkai, & Tsourakakis, 2015; Khuller & Saha, 2009a). As various features of densest subgraphs have been studied, there are several ubiquitous definitions of “density”. These definitions mainly concern subgraph nodes and their degrees, such as edge density, *k*-core, $\alpha - quasi - cliques$ and $k - cliques$. They will be discussed later in detail in Section 2.

Corresponding to these density definitions, algorithms for finding the densest subgraphs are also developed, which are complicated in both computational performance and space efficiency. They are generally classified into three categories (Mitzenmacher, Pachocki, Peng, Tsourakakis, & Xu, 2015): 1) Expert and intelligent heuristics without theoretical guarantees (Balalau et al., 2015); 2) Brute-forth or exhaustive search techniques that may be applicable to small graphs; and 3) Poly-time solvable algorithms with theoretical guarantees (Mitzenmacher et al., 2015). Most of these al-

[*] Corresponding authors.
*E-mail addresses:* hantao@buaa.edu.cn, y.tian@qut.edu.au (T. Han), y.tian@qut.edu.au (Y.-C. Tian), lanyuqing@buaa.edu.cn (Y. Lan), ghllfl@163.com (F. Li), xiaolm@buaa.edu.cn (L. Xiao).

https://doi.org/10.1016/j.eswa.2017.10.047
0957-4174/© 2017 Elsevier Ltd. All rights reserved.

gorithms process the data set as a whole. In a typical computing environment, they have the capability of analyzing a graph with 3 million edges at a time. However, recent research has indicated that finding the densest subgraphs without a size limitation is an NP-Complete problem (Khuller & Saha, 2009b) and detecting communities with certain constrains is an NP-Hard problem (Balalau et al., 2015). This implies that the execution time performance of the algorithms deteriorates exponentially with the increase of the size of the data set. Obtaining the densest subgraphs from a vast volume of data sets within an acceptable period of time becomes critical in real-world applications.

It is even more challenging that social network graphs are not static but highly dynamic. Such graphs are formed from streaming data, and thus evolve over time. For example, a friendship graph from Twitter has more than 2 billion vertices and 65 billion edges. It produces approximately 0.1 million creations and deletions of edges and nodes per hour. In this case, edge stream arrives persistently and endlessly over time. Such highly dynamic and large-scale streaming data poses new challenges to the problem of finding the densest subgraphs. This demands effective and efficient techniques to handle the vast volume of steaming data.

As the DSP is an NP problem (Khuller & Saha, 2009b), in order to improve the computational performance and space efficiency, a key idea is to reduce the space over which the search is performed for a solution. It is desirable that as much as possible information of the original graph be retained in the reduced space. Otherwise, the solution obtained from the reduced space may not be a good solution. So far, there is a lack of a rigorous theory to guide the space reduction for social network graphs. Expert and intelligent systems have demonstrated great success in dealing with large-scale and complicated real-world problems. They have the potential to make significant contributions in discovery of inherent structure of social networks, decomposition of large-scale problems into multiple smaller ones, and particularly acceleration of DSP computation of big and streaming data.

Pre-processing of the graph data set will help reduce the data space, thus accelerating the process of finding the densest subgraphs. The "Sampling" method is such a data pre-processing algorithm (Mitzenmacher et al., 2015). Incorporating with subgraph finding algorithms, it gives statistically reasonable and high-quality subgraphs in a polynomial time. However, the sampling method samples the data set in probability (Mitzenmacher et al., 2015) or poses other constraints. It not only accepts less important data items but also discards significant data items. Therefore, it has inherent shortcomings.

To address this problem, an intelligent data space reduction approach is presented in this paper from the perspective of expert and intelligent systems. It uses a sliding window segmentation strategy to segment the vast volume of graph data into a series of small graphs with a fixed number of edges. Then, it eliminates superfluous information of the small graphs by filtering out the least connected edges. When the small graphs are processed, the approach incrementally merges the non-overlapping subgraphs to reveal the densest subgraphs. Finally, the data space previously filtered out due to their local uselessness is checked for recovery of globally useful edges. Incorporating with existing subgraph finding techniques, the approach is expected to behave much better than the sampling method in terms of the quality of the solution.

This paper is organized as follows. The notations used throughout this paper are listed in Table 1. Section 2 discusses related work and motivations. Section 3 defines densest subgraph problem and presents its challenges. This is followed by Section 4 for a data space reduction approach for DSP in social graphs. The presented approach is demonstrated in Section 6 through experiments over four typical data sets. Finally, Section 7 concludes the paper.

**Table 1**
Notations and definitions.

| Notation | Descriptions |
|---|---|
| $c_k$ | the number of $k - cliques$ in a graph |
| $D$ | the diameter of a graph |
| $d_u$ | the degree of node $u$ |
| $den(G)$ | approaches to find densest subgraphs in $G$, including *edge density*, $k - core$, $\alpha - quasi - cliques$, and $k - cliques$ |
| $e$, $E$ | an edge $e$ and edge set $E$, $e \in E$ |
| $dict()$ | dictionary in data structure, "key-value" pairs |
| $\lvert E(S, S)\rvert$ | $\lvert E(S, S)\rvert = \lvert E(S)\rvert$, the number of edges in $S$ |
| $\lvert E(v, S_i)\rvert$ | the number of edges of node $v$ to graph $S_i$ |
| $G(V, E)$ | a graph $G$ with vertices set $V$ and edges set $E$, $v \in V$ and $e \in E$ |
| $G_i$ | induced from graph $w_i$ after pruning |
| $G_i^{'}$ | induced from graph $G_i$ and $G_{i-1}^{'}$ by merging |
| $G_i^{''}$ | the densest subgraphs of $G$ |
| $H()$ | Shannon entry |
| $i$, $k$ | indices |
| $M$, $S$ | graph |
| $m$, $n$ | the numbers of edges and nodes, $m = \lvert E\rvert = E(G)$, $n = \lvert V\rvert$ |
| $p$ | $p \in (0, 1)$, parameter for Incremental Merging Algorithm |
| $Pro()$ | the probability of outcomes |
| $P_i$ | the priority of edge $e_i$ |
| $Q_j$ | $pruned\_edges\_queue[Q_j] = e_j$, $Q_j$ is the "key" in "key-value" pairs of dictionary |
| $sn$ | the number of subgraphs |
| $t[G]$ | the number of triangles in graph G |
| $v$, $V$ | node (vertex) $v$, and set $V$ of nodes, $v \in V$ |
| $w_i$ | a window of graph with a fixed number of edges, e.g., 10,000 |
| $\alpha$ | parameter for $\alpha - quasi - cliques$, $\alpha \in (0, 1)$ |
| $\delta$ | $\delta = E(G)/\binom{\lvert G\rvert}{2}$ |
| $\rho$ | $1 - \rho$ denotes the density of a graph |

## 2. Related work and motivations

The Densest subgraph problem (DSP) is a key primitive in both algorithmic graph theory and graph mining. It is used to discover the most connected communities in Twitter. It is also used to mine co-author relationships in DBLP to find Nucleus hierarchy relationships in bioinformatics (Sariyüce, Seshadhri, Pinar, & Çatalyürek, 2015). Co-authors graph in DBLP helps discover close cooperation in research. Social graphs (Wang et al., 2015) reveal the friendships among people. A road map presents the location of large cities and towns. Bipartite graphs indicate the purchase behaviours of customers, where grouped products can be recommended to the customers in the same subgraph. All these problems are theoretically described as finding the densest subgraphs. There have been comprehensive surveys (Harenberg et al., 2014; Lee, Ruan, Jin, & Aggarwal, 2010; McGregor, 2014) and tutorials (Gionis & Tsourakakis, 2015) on discovery of the densest subgraphs.

It is indicated in the survey (Lee et al., 2010) and tutorial (Gionis & Tsourakakis, 2015) that there are several ubiquitous definitions for density. These definitions mainly concern the nodes and their degrees. According to the definitions of density, there are fowling subgraph finding algorithms: *edge density*, $k - core$ (Bhattacharya et al., 2015), $\alpha - quasi - cliques$, *connectivity*, $near - cliques$ and $k - cliques$. Table 2 tabulates these definitions and descriptions, from which various algorithms have been developed for practical DSP computation.

The above mentioned algorithms have different objectives. The *edge density* algorithm finds the maximum average degree subgraph of a graph. The algorithm for *connectivity* illustrates how a vertex in a subgraph connects to other vertices in the same subgraph. The $k - core$ algorithm finds the nodes that have at least $k$ connections with other vertices in the same subgraph. In