PROCEEDINGS OF SCIENCE

An Algorithm for Densest Subgraphs of Vertex-weighted Graphs

Zhongling Liu
School of Computer Science and Educational Software, Guangzhou University
Guangzhou, 510006,China
E-mail: 240922247@qq.com

Wenbin Chen[1]
Department of Computer Sicence and Guangdong Provincial Engineering Technology Research Center for Mathematical Educational Software, Guangzhou University
Guangzhou, 510006, China
E-mail: cwb2011@gzhu.edu.cn

Fufang Li[a]**; Ke Qi**[b]**; Jianxiong Wang**[c]
School of Computer Science and Educational Software, Guangzhou University
Guangzhou, 510006, China
E-mail: liff@gzhu.edu.cn; 526903128@qq.com; xiong.ong@gmail.com

It's a significant problem to find the densest subgraph in many research areas. Now, there are so many groups in the WeChat, QQ and other online chat softwares. In order to find the closely connected subgraphs with the maximum average density for the vertex-weighted graph, we introduce the concept of the densest subgraph, and then an exactly algorithm is presented as an extension of the Goldberg's algorithm. By theoretical analysis, we prove that our proposed algorithm is correct and it runs in polynomial time. Thus, our proposed algorithm can be used to find the closely connected subgroups with the maximum average density in fact.

ISCC2017
16-17, December, 2017
Guangzhou, China

[1]Corresponding Author
The work of Wenbin Chen is supported by the NSFC (Grant No.11271097) and the Program for IRT in Education Department of Guangdong (Grant No.2015KCXTD014, 2016KCXTD017). The work of Fufang Li is supported by NSFC (Grant No.61472092) and Guangdong Provincial STPP (Grant No. 2013B010401037) and Guangzhou MHSSRF (Grant No.1201421317). Ke Qi's research is supported by Guangzhou STPP (Grant No. 201605061403261) and Guangdong Provincial NSF (Grant No. 2017A030313374). The work of Jianxiong Wang is supported by FDYT in Higher Education of Guangdong (Grant No. 2012WYM0105 and 2012LYM0105) and FPRDIHL (Grant No. 2012A143).

PoS(ISCC 2017)013

1. Introduction

In 1980s, the problem of finding densest subgraph (DS) was presented. Then scientists have proposed some algorithms tfor solution because it was highly significant in research of network analysis, web communities inferred, hypertext analysis and the function of complex biological networks discovery[9,10,12,15,17]. In 1984, Goldberg firstly proposed the definition of the densest subgraph(DS): Ḡ (V, E) is an undirected graph, $|V|=n, |E|=m$. The density of its subgraph Ḡ (V',E') is m/n. DS in G is a subgraph that its density is the maximum value [11]. So the problem of looking for the DS is to find Ḡ (V',E') with the largest density. In this paper, Goldberg gave a algorithm with polynomial time for this problem. It worked as follows. At first, they constructed a network, used the min-cut algorithms to divide the network and finally found the densest subgraph by binary method [11]. In 1989, Gallo et al. gave an algorithm by using max-flow techniques [8]. Then some approximation algorithms were given For instance, in 1999, Kannan and Vinay proposed an approximation algorithm in directed graphs. In this paper, Kannan and Vinay introduced the density concept for the web graphs [13]. Soon, Charikar gave two algorithms to optimize the finding densest subgraph according to Kannan and Vinay's definition. Charikar firstly gave a linear programming (LP)-based exactly algorithm, then they developed a greedy algorithm with 2-approximation, the algorithm removed the minimum degree vertex by iterated [3].

From the early 1990s, an extensive research of densest subgraph (DS) problem has achievedwide attentions. The problem was to find the densest k-subgraph(DkS), implying an induced subgraph with the maximum density which has k vertices. Some approximation algorithms have been given by researchers. For the directed graph, Kortsarz and Peleg designed an algorithm for this problem in 1993. The algorithm hasdan $O(n^{0.3885})$-ratio approximation[16]. In 1997, an semi-definite programming (SDP) based approximation algorithm was proposed by Feige and Seltser, it had n/k-radio[6]. Soon after Srivastav and Wolf presented an approximation algorithm based on SDP with the same radio[18]. Four years later, Feige developed another approximation algorithm with O (n^{δ}) ratio, when $\delta<1/3$. In 2010, an $O(n^{1/4})$-approximation algorithm was presented by Bhaskara. This algorithm worked in the streaming model, and it was suitable for distributed models [2].

In 2009, in order to find the densest subgraph of size constraint undirected graphs, Andersen and Chellapilla introduced the notions of two problems: the densest at-least-k-subgraph (DalkS) and the densest at-most-k-subgraph(DamkS)[1]. For DalkS, they also gave two approximation algorithms, the 3-approximation which extended the algorithm, as proposed by Charikar and 2-approximation on the basis of Gallo's maximum flow algorithm [8]. In the same year, Khuller and Saha also gave a max-flow algorithm based algorithm with 2-approximation and a LP-based algorithm which had the same radio. Tey also proved that DalkS was a NP-hard problem [14]. Chen et al. designed two approximation algorithms for DamkS. Both were $(n-1)/(k-1)$ ratio and $O(n^{\delta})$ time for some $\delta<1/3$ respectively. Chen et al. still gave a polynomial algorithm when *k* was constrainted by the constant[4]. An improved algorithm was proposed with better time complexity in Chen et al.'s another paper. A greedy approximation algorithm was also proposed to get the densest subgraph of the specified subset [5].

However, for the vertex-weighted graph, how can we get the densest subgraph(DS)? To our knowledge, no one studied this problem. We began to study it in the paper. We firstly introduced a notion of DS of vertex-weighted graphs, and then we proposed an algorithm. And we have proved that this algorithm is polynomial. Our algorithm is an extension of Goldberg's algorithm[11]. Firstly, we assume a vertex-weighted graph Ḡ, G' is its subgraph. Let D be the density of G'. We make a estimated value g of D. Secondly, we try to construct a new graph whose edges' weights relate to g and divide the graph by using the min-cut algorithms. At last, we find the dsg of G by a binary search.

2. Definitions

Some definitions are given as follows.

Definition 1: the densest subgraph of undirected graph.

For an undirected graph (UDG) Ḡ(V',E'), $V_x \in V'$. Ḡ(V,E) is an induced subgraph from the vertex V_x. D(V) is the density of Ḡ (V,E) and defined as $|E|/|V|$, i.e.

$$D(V)=\frac{|E|}{|V|} \tag{2.1}$$

For an undirected graph, its DS problem is a computational problem to find the maximum density as an induced subgraph from G .

Definition 2: the densest subgraph of vertex-weighted UDG.

Give Ḡ (V', E', ω') is a vertex-weighted UDG, in which, ω' is a weighted function, ω': V'→N, N is a natural number set, $V_\chi \in V'$. For the induced subgraph Ḡ(V,E,ω) from V_χ, the density D(V'') is $|e|/\sum \omega(\chi)$, $\chi \in V$, i.e.

$$D(V'')=\frac{|E|}{|\sum_{\chi\in V} \omega(\chi)|} \tag{2.2}$$

If all vertices' weight is 1, D(V'') is the result as the undirected Definition 1 .

For a vertex-weighted UDG, the DS problem is equal to find the maximum density induced subgraph.

Definition 3; the minimum cut.

For an UDG G(V, E), $\{V_\lambda | \lambda \in [1,n]\}$, and $\{E_\kappa | \kappa \in 1,2\ldots m\}$, in which, E_κ has a capacity $c(E_\kappa)$.

1) A cut refers to splitting V into two sets V_1 and V_2. O is an edge set, $O=\{(\kappa_1,\kappa_2)\in E, \kappa_1 \in V_1, \kappa_2 \in V_2\}$. $c(V_1,V_2)$ is the capacity of (V_1,V_2) equal to adding the edge's capacity in this cut, i.e.

$$c(V_1,V_2)=\{\sum c(\kappa_1,\kappa_2), \kappa_1\in V_1, \kappa_2 \in V_2\} \tag{2.3}$$

2) Given the two vertices $s,t\in V$,divide V into two sets. s and t belong to two different sets M and N, so the cut is a s-t cut. i.e. $\{(M,N)|s\in M, t\in N\}$. Hence,

$$c(M,N)=\{\sum c(\kappa_1,\kappa_2), \kappa_1 \in M, \kappa_2 \in N\} \tag{2.4}$$

So the min-cut is a s-t cut with the minimum capacity.

3. Algorithm for G

In this section, by extending the Goldberg's algorithm[11], we propose an algorithm for the DS in a vertex-weighted UDG as follows. At first, we give graph G, then let D be the maximum density of all subgraphs. We make an estimated value g of D. Secondly we try to construct a new graph whose edges' weights relate to g and divide the graph by using the min-cut algorithms. At last, we find the densest subgraph of G by a binary search.

3.1 Undirected Graph Construction by A Guess Value

Assume that G (V, E, ω) be a vertex-weighted UDG, $V_\iota | \iota \in n, E_\kappa | \kappa \in m$, in which, ω is a weighted function, ω: V→N, N is a natural number set. d(κ) denotes the vertex κ 's degree and the weight of κ is ω(κ). Let D be the density of the induced subgraphs. We make a guess value g of D, then we construct the graph $G_g=(V_g, E_g, \omega_g)$ related to g. This is an extension of the Goldberg's algorithm [11].

We add two vertex r, h of E. For any edge (κ,φ), the capacity c(κ,φ) is 1. The capacity of edge (r, κ) is m. The capacity of edge (κ, h) is $m+2g*\omega(\kappa)-d(\kappa)$.

$G_g=(V_g, E_g, \omega_g)$ is as follows.

$$\boxed{\begin{array}{l} V_g=V\cup\{r,h\} \\ E_g=\{(\kappa,\varphi),\kappa,\varphi\in E\}\cup\{(r,\kappa),\kappa\in V\}\cup\{(\kappa,h),\kappa\in V\} \\ c(\kappa,\varphi)=1,(\kappa,\varphi)\in E \\ c(r,\kappa)=m,\kappa\in V \\ c(\kappa,h)=m+2g\omega(\kappa)-d(\kappa),\kappa\in V \end{array}} \tag{3.1}$$

3.2 Relationship Establishment with g and the Minimum Cut

We split the Vg into two sets M and N. If $r\in M, h\in N$, cut (M,N) as a s-t cut. Let V_1=M-{r},V_2=N-{h}. If $|V_1|=0$, c(M,N)=m|V|. If $|V_1|\neq 0$, c(M,N) is $\sum c(\kappa,\varphi),\kappa\in M,\varphi\in N$. Let D_1 be the density of the induced vertex-weighted subgraph from V_1, so we have the following conclusion:

$$c(M,N)=m|V|+2|V_1|(g-D_1) \tag{3.2}$$

The reason is as follows:

$$\begin{aligned} c(M,N)&=\sum_{\kappa\in S,\varphi\in T} c(\kappa,\varphi) \\ &=\sum_{\varphi\in V_2} c(r,\varphi)+\sum_{\kappa\in V_1} c(\kappa,h)+\sum_{\kappa\in V_1,\varphi\in V_2} c(\kappa,\varphi) \\ &=m|V_2|+\left(m|V_1|+2g\sum_{\kappa\in V_1}\omega(\kappa)-\sum_{\kappa\in V_1} d(\kappa)\right)+\sum_{\kappa\in V_1,\varphi\in V_2} 1 \\ &=m|V|+|V_1|2\left(g-\frac{\sum_{\kappa\in V_1} d(\kappa)-\sum_{\kappa\in V_1,\varphi\in V_2} 1}{2\sum_{\kappa\in V_1}\omega(\kappa)}\right) \end{aligned}$$

Because $\frac{1}{2}\left(\sum_{\kappa\in V_1} d(\kappa)-\sum_{\kappa\in V_1,\varphi\in V_2} 1\right)$ is the edge number of $G(V_1)$, so:

$$D_1=\frac{\sum_{\kappa\in V_1} d(\kappa)-\sum_{\kappa\in V_1,\varphi\in V_2} 1}{2\sum_{\kappa\in V_1}\omega(\kappa)} \tag{3.3}$$

Hence: $c(M,N)=m|V|+2|V_1|(g-D_1)$. According to similar argument of Goldberg's paper, we get the result as follows[11].

Lemma 1: let cut (M,N) be the min-cut. If $|V_1|\neq 0$, then $g\leqslant D$; if V_1=0, (i.e. M={κ}), then $g\geqslant D$.

3.3 Algorithm Description

Before describing the algorithm, let's prove the results below.

Lemma 2: give the vertex-weighted UDG G (V, E, ω) , if G (V', E', ω') is one of the subgraphs, the density of G (V', E', ω') is D' and the other's is D'', if $D^{''} \leqslant D^{'} + \frac{1}{N(N+1)}$, N is $\sum_{i=1}^{n} \omega(i)$. Then G (V', E', ω') is the DS of the vertex-weighted UDG.

Proof: let the maximum density of G (V', E', ω') be H. m' is edge number, n' is ω(μ). H is m' divided by n',

$$H \in \{ \frac{m^{'}}{n^{'}} | 0 \leqslant m^{'} \leqslant m, min(\omega(\kappa)) \leqslant n^{'} \leqslant N, N = \sum_{\kappa=1}^{n} \omega(\kappa) \} \quad (3.4)$$

D is between 0 and $\frac{m}{min(\omega(\kappa))}$. If the difference between the two values of D is Δ,

$$\Delta = \frac{m_1}{n_1} - \frac{m_2}{n_2} = \frac{m_1 n_2 - m_2 n_1}{n_1 n_2} \quad (3.5)$$

If $n_1 = n_2$, then $\Delta \geqslant \frac{1}{N}$. Otherwise, $n_1 n_2 \leqslant N(N-1)$, then $|\Delta| \geqslant \frac{1}{N(N-1)}$. So the minimum difference between the two values of D is $\frac{1}{N(N-1)}$.

Our algorithm is given below.

$$
\begin{array}{l}
x \leftarrow 0 ; y \leftarrow \frac{m}{min(\omega(\mu))} ; V_1 \leftarrow null ; \\
while \left(y - x \geqslant \frac{1}{N(N-1)} \right) \\
\{ \quad g \leftarrow \frac{x+y}{2} ; \\
\quad construct\, G_g = (V_g, E_g, \omega_g) ; \\
\quad find\ \ mincut(S, T) ; \\
\quad if (S = \{f\}) \quad y \leftarrow g ; \\
\quad else \{ \\
\qquad x \leftarrow g ; \\
\qquad V_1 \leftarrow S - \{f\} ; \} \\
\} \\
return(denest\ subgraph\ of\ G\ \ induced\ \ by\ V_1)
\end{array}
$$

Let $V_1 \subseteq V'$, x is an estimated lowest value for D', and y is the highest one , so $D^{'} \in [x, y]$, g is an estiamated maximum value of D'. Lemma 2 tells us that if $y \geqslant x + \frac{1}{N(N-1)}$, then G' is the densest subgraph. So it is the loop condition for the while loop. In the while loop, we firstly assign the value $\frac{x+y}{2}$ to g, then construct an undirected graph N, divide N and output the minimum cut for (M,N), then we get the value of g by using Lemma 1. At last, the return is the DS of G.

Theorem 3. The time complexity of algorithm is O(n(n+m)log^2 n).

Proof: for the algorithm, while loop is O(log n) time. And K(ς,η) denotes the time to find the min-cut. The number of V_g is n+2 and E_g is 2(m+n). In the time complexity, we can ignore constants to find the min-cut in O(K(n, n+m)log n) time. Hence, our algorithm time is O (K(n, n+m)log n). However, the time of the mininum-cut problem is O(|V||E|log |V|) as introduced by Sleator and Tarjan's algorithm [19]. So our algorithm is polynomial.

4. Conclusion

In this paper, we propose the concept of DS, and then an exactly algorithm is presented as an extension of the Goldberg's algorithm. By theoretical analysis, we've proved that our proposed algorithm is correct and it runs in polynomial time. Thus, our proposed algorithm can be used to find the closely connected subgroups with the maximum average density in fact.

References

[1] Reid Andersen, Kumar Chellapilla. *Finding dense subgraphs with size bounds.* in: WAW 2009, 2009, pp. 25–37.

[2] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, Aravindan Vijayaraghavan. *Detecting high log-densities: an O(n1/4) approximation for densest k-subgraph.* in: STOC 2010, 2010, pp. 201–210.

[3] M. Charikar. *Greedy approximation algorithms for finding dense components in a graph.*in: Proceedings Third International Workshop on Approximation Algorithms for Combinatorial Optimization. in: Lecture Notes in Computer Science, vol. 1913, Springer, Berlin, 2000, pp. 84–95.

[4] Wenbin Chen, Nagiza F Samatova, Matthias F Stallmann, William Hendrix , Weiqin Ying. *On size-constrained minimum s-t cut problems and size-constrained dense subgraph problems.* Theoretical Computer Science, Jan. 2016, Vol.609, pp. 434-442.

[5] Wenbin Chen,Lingxi Peng,Jianxiong Wang,Fufang Li,Maobin Tang. *Algorithms for the Densest Subgraph with at Least k Vertices and with a Specified Subset.*The 9th Annual International Conference on Combinatorial Optimization and Applications (COCOA 2015), Dec. 18-20,2015, Houston. LNCS 9486,pp.566-573.

[6] U. Feige, M. Seltser. *On the densest k-subgraph problems.*Technical report: CS 97-16, Department of Applied Mathematics and Computer Science, 1997.

[7] U. Feige, G. Kortsarz, D. Peleg. *The dense k-subgraph problem.* Algorithmica 29 (2001) 410–421.

[8] G. Gallo, M. Grigoriadis, R. Tarjan. *A fast parametric maximum flow algorithm and applications.* SIAM J. Comput. 18 (1) (1989) 30–55.

[9] D. Gibson, J. Kleinberg, P. Raghavan. *Inferring web communities from Web topology.* in: Proc. HYPERTEXT, 1998, pp. 225–234.

[10] D. Gibson, R. Kumar, A. Tomkins. *Discovering large dense subgraphs in massive graphs.* in: Proc. 31st VLDB Conference, 2005.

[11] A. Goldberg. *Finding a maximum density subgraph.*Technical report UCB/CSB 84/171, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1984.

[12] H. Hu, X. Yan, Y. Huang, et al. *Mining coherent dense subgraphs across massive biological networks for functional discovery.* Bioinformatics 21 (2005) 213–221.

[13] R. Kannan, V. Vinay. *Analyzing the structure of large graphs.* Manuscript, 1999.

[14] Samir Khuller, Barna Saha. *On finding dense subgraphs.* in: ICALP 1, 2009, pp. 597–608.

[15] J. Kleinberg. *Authoritative sources in hypertext linked environments.* in: Proc. 9th Annual ACM–SIAM Symposium on Discrete Algorithms, 1998, pp. 668–677.

[16] G. Kortsarz, D. Peleg. *On choosing a dense subgraph.* in: Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, 1993, pp. 692–701.

[17] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins. *Trawling the Web for emerging cyber-communities.* in: Proc. 8th WWW Conference, WWW, 1999.

[18] A. Srivastav, K. Wolf. *Finding dense subgraph with semi-definite programming.* in: K. Jansen, J. Rolim (Eds.), Approximation Algorithms for Combinatorial Optimization, 1998, pp. 181–191.

[19] D.D. Sleator. PH.D. *Dissertation.* Stanford University,1980.