Georgia State University
ScholarWorks @ Georgia State University

Computer Science Technical Reports | Department of Computer Science

Spring 4-3-2019

Finding densest subgraph in a bi-partite graph

Dhara Shah
dshah8@student.gsu.edu

Sushil Prasad
sprasad@gsu.edu

Danial Aghajarian
daghajarian@cs.gsu.edu

Finding densest subgraph in a bi-partite graph

Dhara Shah[1], Danial Aghajarian[1], and Sushil Prasad[1]

Department of Computer Science, Georgia State University, Atlanta 30303, USA
{dshah8@, daghajarian@cs, sprasad@}gsu.edu

Abstract. Finding the densest subgraph in a bi-partite graph is a polynomial time problem. Also, each bi-partite graph has a densest connected subgraph. In this paper, we first prove that each bi-partite graph has a densest connected subgraph. This proof is different than that of an undirected graph, since our definition of the density is different. We then provide a max-flow min-cut algorithm for finding a densest subgraph of a bi-partite graph and prove te correctness of this binary search algorithm.

Keywords: densest subgraph · bi-partite · max-flow · densest connected

1 Densest subgraph of a bi-partite graph

We observe that there can be multiple densest bi-partite subgraphs of a bi-partite graph. We produce the following proof for this.

Theorem 1. *Let* $G(S_{a_1}, S_{b_1}, E(S_{a_1}, S_{b_1})), G(S_{a_2}, S_{b_2}, E(S_{a_2}, S_{b_2}))$ *be bipartite subgraphs, with* $S_{a_1} \cap S_{a_2} = \phi, S_{b_1} \cap S_{b_2} = \phi, E(S_{a_1}, S_{b_2}) = \phi, E(S_{a_2}, S_{b_1}) = \phi, E(S_{a_1}, S_{b_1}) \cap E(S_{a_2}, S_{b_2}) = \phi$.
Let $|S_{a_1}| = a_1, |S_{a_2}| = a_2, |S_{b_1}| = b_1, |S_{b_2}| = b_2, |E(S_{a_1}, S_{b_1})| = e_1, |E(S_{a_2}, S_{b_2})| = e_2$.
Let the density of this graphs defined by
$\rho(G(S_{a_1}, S_{b_1}, E(S_{a_1}, S_{b_1})) = \frac{e_1}{\sqrt{a_1 b_1}}$,
$\rho(G(S_{a_2}, S_{b_2}, E(S_{a_2}, S_{b_2})) = \frac{e_2}{\sqrt{a_2 b_2}}$,
$\rho(G(S_{a_1} \cup S_{a_2}, S_{b_1} \cup S_{b_2}, E(S_{a_1}, S_{b_1}) \cup E(S_{a_2}, S_{b_2})) = \frac{e_1+e_2}{\sqrt{(a_1+a_2)(b_1+b_2)}}$
Prove that $\frac{e_1+e_2}{\sqrt{(a_1+a_2)(b_1+b_2)}} \leq max\{\frac{e_1}{\sqrt{a_1 b_1}}, \frac{e_2}{\sqrt{a_2 b_2}}\}$

Proof. Without loss of generality, let $max\{\frac{e_1}{\sqrt{a_1 b_1}}, \frac{e_2}{\sqrt{a_2 b_2}}\} = \frac{e_1}{\sqrt{a_1 b_1}}$.
This implies,

$$\frac{e_1}{\sqrt{a_1 b_1}} \geq \frac{e_2}{\sqrt{a_2 b_2}} \Leftrightarrow e_2 \leq e_1 \frac{\sqrt{a_2 b_2}}{\sqrt{a_1 b_1}} \tag{1}$$

Now, under this assumption,

$$\begin{aligned} &\frac{e_1 + e_2}{\sqrt{(a_1 + a_2)(b_1 + b_2)}} \leq max\{\frac{e_1}{\sqrt{a_1 b_1}}, \frac{e_2}{\sqrt{a_2 b_2}}\} \\ \Leftrightarrow &\frac{e_1 + e_2}{\sqrt{(a_1 + a_2)(b_1 + b_2)}} \leq \frac{e_1}{\sqrt{a_1 b_1}} \end{aligned} \tag{2}$$

Also, LHS of equation (2)=

$$\frac{e_1 + e_2}{\sqrt{(a_1 + a_2)(b_1 + b_2)}} \le \frac{e_1 + e_1\frac{\sqrt{a_2 b_2}}{\sqrt{a_1 b_1}}}{\sqrt{(a_1 + a_2)(b_1 + b_2)}} \text{ Because (1)}$$
$$= \frac{e_1(\sqrt{a_1 b_1} + \sqrt{a_2 b_2})}{\sqrt{a_1 b_1}\sqrt{(a_1 + a_2)(b_1 + b_2)}}$$

Hence, if we prove

$$\frac{e_1(\sqrt{a_1 b_1} + \sqrt{a_2 b_2})}{\sqrt{a_1 b_1}\sqrt{(a_1 + a_2)(b_1 + b_2)}} \le \frac{e_1}{\sqrt{a_1 b_1}} = \text{ RHS of equation (2)}$$

we prove (2).
Here,

$$\frac{e_1(\sqrt{a_1 b_1} + \sqrt{a_2 b_2})}{\sqrt{a_1 b_1}\sqrt{(a_1 + a_2)(b_1 + b_2)}} \le \frac{e_1}{\sqrt{a_1 b_1}}$$
$$\Leftrightarrow (\sqrt{a_1 b_1} + \sqrt{a_2 b_2}) \le \sqrt{(a_1 + a_2)(b_1 + b_2)}$$
$$\Leftrightarrow (\sqrt{a_1 b_1} + \sqrt{a_2 b_2})^2 \le (a_1 + a_2)(b_1 + b_2)$$
$$\Leftrightarrow 2\sqrt{a_1 b_1 a_2 b_2} \le a_1 b_2 + a_2 b_1$$
$$\Leftrightarrow \sqrt{(a_1 b_2)(a_2 b_1)} \le \frac{a_1 b_2 + a_2 b_1}{2}$$

This is true since arithmetic mean of two non-negative real numbers is always greater than or equal to their geometric mean.
Hence

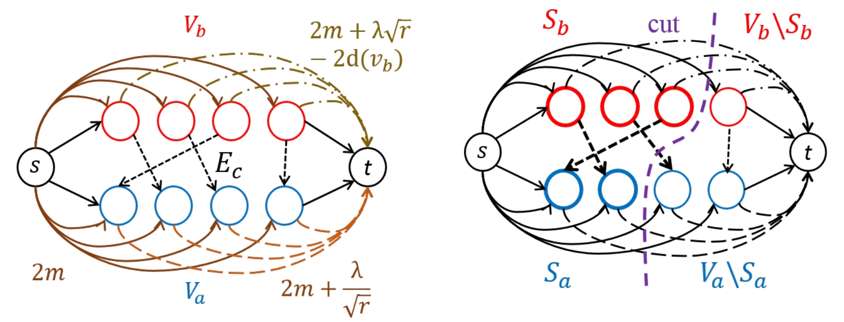
$$\frac{e_1 + e_2}{\sqrt{(a_1 + a_2)(b_1 + b_2)}} \le \frac{e_1(\sqrt{a_1 b_1} + \sqrt{a_2 b_2})}{\sqrt{a_1 b_1}\sqrt{(a_1 + a_2)(b_1 + b_2)}}$$
$$\le \frac{e_1}{\sqrt{a_1 b_1}} = max\{\frac{e_1}{\sqrt{a_1 b_1}}, \frac{e_2}{\sqrt{a_2 b_2}}\}$$

2 Maxflow Densest Subgraph (MDS)

MDS algorithm finds a densest bi-partite subgraph of a Triple Network in polynomial time. Inspired by [2] and [1], we use the max-flow min-cut strategy to obtain the densest bi-partite subgraph.

Definition 1. *(Maximum density of a Triple Network) In a Triple Network* $G(V_a, V_b, E_a, E_b, E_c)$*, maximum density is* $\rho^* = \max_{S_a \subseteq V_a, S_b \subseteq V_b} \frac{|E_c(S_a, S_b)|}{\sqrt{|S_a||S_b|}}$.

Let $G_c[S_a, S_b]$ be a bi-partite subgraph of the Triple Network G. Consider the number λ for which $|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|} = 0$. λ, thus the density of this graph, depends on ratio $r = \frac{|S_a|}{|S_b|}$ and $|E_c(S_a, S_b)|$. Ratio r can take at most



(a) Construction of the flow graph for finding a densest subgraph of the Triple Network $G(V_A, V_b, E_c)$

(b) Finding the minimum cut for given ratio guess r and iteratively adjusting the bounds of maximum density renders a densest subgraph $G(S_a, S_b)$

Fig. 1. MDS algorithm: Flow construction and iterations

$|V_a||V_b|$ different values, and λ, the density guess, ranges in $(0, \sqrt{|V_a||V_b|}]$. It is evident from definition 1 that finding a densest subgraph of the Triple Network is equivalent to finding $\underset{S_a \subset V_a, S_b \subset V_b}{max} \{\lambda | \, |E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|} = 0\}$ over all subgraphs $G_a[S_a], G_b[S_b]$. hence, to find ρ^*, instead of enumerating all possible subgraphs $S_a \subset V_a$ and $S_b \subset V_b$, we can guess λ and r. For these guessed values of λ and r, if we can find a subgraph $G_c[S_a, S_b]$ such that $\rho(S_a, S_b) \geq \lambda$, then $\rho^* \geq \rho(S_a, S_b) \geq \lambda$. In that case, for the same r, the next guess for λ would be higher than the current guess. If no such subgraph exists, then $\rho^* < \lambda$. In this case, for the same r, the next guess for λ would be lower than the current guess. The verification that such a subgraph exists or not could be done using flow networks. Finding a densest bi-partite subgraph for a given r thus could be viewed as a binary search for λ. By enumerating all such r, we guarantee to obtain the densest subgraph.

Given a Triple Network G and values of λ and r, we construct the following flow network. This flow network yields a subgraph $G_c[S_a, S_b]$ with $\rho(S_a, S_b) \geq \lambda$ if such a subgraph exists in G. Else it yields an empty set.

- (f_1) Initialize weighted directed graph $G'(V', E')$ with $V' = V_a \cup V_b, E' = \phi$, and a constant $m = |E_c|$
- (f_2) For all edges $\{v_a, v_b\} \in E_c$, add (v_b, v_a) with weight 2 to E'
- (f_3) Add source node s and sink node t to V'
- (f_4) For all vertices $v \in V_a \cup V_b$, add edge (s, v) with weight $2m$ to E'
- (f_5) For all vertices $v_a \in V_a$, add edge (v_a, t) with weight $2m + \frac{\lambda}{\sqrt{r}}$ to E'
- (f_6) For all vertices $v_b \in V_b$, add edge (v_b, t) with weight $2m + \sqrt{r}\lambda - 2d(v_b)$ to E', where $d(v_b)$ is the degree of v_b in G

Now, we apply the MDS algorithm 1 to this graph.

Theorem 2. *MDS algorithm yields a densest subgraph of the Triple Network.*

Proof. Let $G(V_a, V_b, E_a, E_b, E_c)$ be a Triple Network with $V_a \neq \phi, V_b \neq \phi$. Let $G'(V', E')$ be the weighted directed flow network constructed from this network as mentioned above. Let S, T be the minimum s-t cut of this flow network. From figure 1(a), if $S = \{s\}$ and $T = V_a \cup V_b \cup \{t\}$, then the value this trivial cut is $2m(|V_a| + |V_b|)$. However, if $S = \{s\} \cup S_a \cup S_b$ and $T = \{V_a \setminus S_a\} \cup \{V_b \setminus S_b\} \cup \{t\}$ then the value of a cut in this flow network is

$$2m|V_a| + 2m|V_b| - \sum_{v_a \in V_a \setminus S_a} 2m - \sum_{v_b \in V_b \setminus S_b} 2m + \sum_{v_a \in S_a} (2m + \frac{\lambda}{\sqrt{r}})$$

$$+ \sum_{v_b \in S_b} (2m + \sqrt{r}\lambda - 2d(v_b)) + \sum_{\substack{\{v_b, v_a\} \in E \\ ,v_b \in S_b, \\ v_a \in V_a \setminus S_a}} 2$$

$$= 2m(|V_a| + |V_b|) + \lambda\sqrt{r}|S_b| + \frac{\lambda}{\sqrt{r}}|S_a| - 2|E_c(S_a, S_b)|$$

$$= 2m(|V_a| + |V_b|) - 2(|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|})(\text{substitute } r = \frac{|S_a|}{|S_b|})$$

This non-trivial s-t cut, if exists, is minimal. Hence the value of this cut is less than the value of trivial cut. In other words,

$$2m(|V_a| + |V_b|) \geq 2m(|V_a| + |V_b|) - 2(|E_c(S_a, S_b)|$$
$$- \lambda\sqrt{|S_a||S_b|}). \text{ Hence, for a non-trivial s-t cut, } |E_c(S_a, S_b)|$$
$$- \lambda\sqrt{|S_a||S_b|} < 0.$$

So if, for given values of λ and r, the flow network renders a non-trivial s-t cut S, T; then the subgraph $S \setminus \{s\} = G_c[S_a, S_b]$ has density λ such that $|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|} \geq 0$. Which implies that $\rho(S_a, S_b) \geq \lambda$. Hence, maximum density of G has to be higher than the current guess of λ. However, if the flow network renders a trivial s-t cut, no subgraph of G has density λ with given r. Hence, maximum density of G has to be lower than current guess of λ. By repeating this process as a binary search, eventually we will find the smallest λ with $|E_c(S_a, S_b)| - \lambda\sqrt{|S_a||S_b|} = 0$ for the given r. By iterating on possible values of r, the maximum value of such λ is found. This value is maximum density and the corresponding subgraph is a densest subgraph of G.

Theorem 3. *MDS algorithm is a polynomial time algorithm.*

Proof. The density difference of any two subgraphs of a bi-partite graph $G_c[V_a, V_b]$ is $\left|\frac{m}{\sqrt{v_1 v_2}} - \frac{m'}{\sqrt{v_1' v_2'}}\right| \geq \frac{1}{|V_a|^2|V_b|^2}$ with $0 \leq m, m' \leq |E_c|, 1 \leq v_1, v_1' \leq |V_a|, 1 \leq v_2, v_2' \leq |V_b|$. This guarantees that the search for maximum density in the range $(0, \sqrt{|V_a||V_b|}]$ can be performed with step size $\frac{1}{|V_a|^2|V_b|^2}$, halting in $O(|V_a|^{3/2}|V_b|^{3/2})$ iterations.

Within each iteration of this binary search, the minimum cut of the flow graph is calculated in $O(|V_a| + |V_b|)^2(2(|V_a| + |V_b|) + |E_c|))$. Hence, the complexity of algorithm 1 is
$O(|V_a|^{4.5}|V_b|^{4.5})$. Adding the cost of BFS for finding connected components in G_a and G_b, the upper-bound still remains unchanged.

Algorithm 1 Maxflow Densest Subgraph (MDS)

Input: Triple Network $G(V_a, V_b, E_a, E_b, E_c)$,with $V_a \neq \phi, V_b \neq \phi$
Output: A densest bi-partite subgraph $G_c[S_a, S_b]$ of G

$possible\_ratios = \{\frac{i}{j} | i \in [1, \cdots |V_a|], j \in [1, \cdots |V_b|]\}$
$densest\_subgraph = \phi, maximum\_density = \rho(V_a, V_b)$
for $ratio\ guess\ r \in possible\_ratios$ **do**
    $low \leftarrow \rho(V_a, V_b), high \leftarrow \sqrt{|V_a||V_b|}, g = G_c[V_a, V_b]$
    while $high - low \geq \frac{1}{|V_a|^2|V_b|^2}$ **do**
        $mid = \frac{high+low}{2}$
        construct a flow graph G' as described in (f_1) - (f_6) and find the minimum s-t cut S, T
        $g' = S \setminus \{\text{source node } s\}$
        if $g' \neq \phi$ **then**
            $g \leftarrow g'$
            $low = max\{mid, \rho(g)\}$
        else $high = mid$
        if $maximum\_density < low$ **then**
            $maximum\_density = low$
            $densest\_subgraph = g$

References

1. Goldberg, A.V.: Finding a maximum density subgraph. University of California Berkeley, CA (1984)
2. Khuller, S., Saha, B.: On finding dense subgraphs. In: International Colloquium on Automata, Languages, and Programming. pp. 597–608. Springer (2009)