Algorithms for Massive Data Science	Lecture date: April 8, 2019
Instructor: Ben Moseley	Scribe: Thomas Lavastida, Yuyan Wang

1 Densest Subgraph

Let $G = (V, E)$ be a graph and let $S \subseteq V$ be a subset of vertices. Define $E(S) := \{uv \in E \mid u, v \in S\}$ be the edges with both endpoints in S. We define the density of S to be

$$f(S) := \frac{|E(S)|}{|S|}. \tag{1}$$

Note that $2f(S)$ is the average degree of the induced subgraph $G[S]$. The problem of interest is to find the subset S which maximizes $f(S)$. This corresponds to an induced subgraph $G[S]$ with maximum density.

In practice, the graph G may represent a social network. So finding a maximum density subgraph corresponds to finding a community within the social network that is very well connected. This may be of use for people who study social networks. In addition to this, social network graphs can be quite large, motivating the need for distributed algorithms.

Polynomial time algorithms are known for solving this problem exactly [3]. However, they are based on constructing flow networks so they do not easily translate to MPC. Thus we will study a Greedy algorithm that gives a 2-approximation and then show how to adapt it to MPC.

2 LP Formulation

Before defining and analyzing the Greedy algorithm, we will give an exact linear programming formulation that will assist us in the analysis. Consider the following LP in which we have variables x_{ij} for each edge $ij \in E$ and y_i for each vertex $i \in V$.

$$\begin{aligned} \max \quad & \sum_{ij \in E} x_{ij} \\ \text{s.t} \quad & x_{ij} \leq y_j \quad \forall j \in V, \forall ij \in E \\ & \sum_{j \in V} y_j \leq 1 \\ & x_{ij}, y_j \geq 0 \quad \forall ij \in E, \forall j \in V \end{aligned} \tag{2}$$

Let f^* be the optimal value of the LP. We show that the LP exactly captures the densest subgraph problem.

Lemma 1 *For all $\emptyset \neq S \subseteq V$, there is a feasible variable assignment (x, y) such that the objective value of this assignment is at least $f(S)$.*

Proof: We set the variables (x, y) as follows.

$$y_j = \begin{cases} \frac{1}{|S|} & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} \frac{1}{|S|} & \text{if both } i, j \in S \\ 0 & \text{otherwise} \end{cases}$$

(x, y) clearly satisfy non-negativity so we just need to check the other constraints. We have

$$\sum_{j \in V} y_j = \sum_{j \in S} \frac{1}{|S|} = 1.$$

For the other constraint note that if either i or j is not in S, then $x_{ij} = 0$, so $x_{ij} \leq y_j$. Otherwise $x_{ij} = y_j = 1/|S|$. Finally, note that the objective value is

$$\sum_{ij \in E} x_{ij} = \sum_{ij \in E(S)} \frac{1}{|S|} = \frac{|E(S)|}{|S|} = f(S).$$

This proves the lemma. □

The above lemma implies that $f^* \geq \max_{S \subseteq V} f(S)$. We show the other direction in the next lemma.

Lemma 2 *If (x, y) is a feasible LP solution with objective value α, then there exists a subset $S \subseteq V$ such that $f(S) \geq \alpha$.*

Proof: Let (x, y) be a feasible solution to the LP. WLOG we may assume that $x_{ij} = \min\{y_i, y_j\}$ since we can always raise x_{ij} to reduce the slack $\min\{y_i, y_j\} - x_{ij} \geq 0$ to 0, and this only increases the LP's objective value. Now for any $r \geq 0$ define $E(r) := \{ij \in E \mid x_{ij} \geq r\}$ and $S(r) := \{j \in V \mid y_j \geq r\}$

Since $x_{ij} \leq \min\{y_i, y_j\}$, we have that $ij \in E(r) \implies$ both $i, j \in S(r)$. Further if both $i, j \in S(r)$, then $ij \in E(r)$ since $x_{ij} = \min\{y_i, y_j\}$. These two facts imply that $E(r)$ are the edges induced within $S(r)$, i.e. $E(S(r)) = E(r)$. Consider integrating $|S(r)|$ and $|E(r)|$ over $r \geq 0$:

$$\int_{r=0}^{\infty} |S(r)| dr = \sum_j y_j \leq 1.$$

Similarly,

$$\int_{r=0}^{\infty} |E(r)| dr = \sum_{ij} x_{ij} = \alpha$$

If we find an r such that $|E(r)|/|S(r)| \geq \alpha$, then we are done, as taking $S = S(r)$ will correspond to a subset with $f(S) \geq \alpha$. Suppose for contradiction that no such r exists. Then we have that $|E(r)| < \alpha |S(r)|$ for all $r \geq 0$, and so:

$$\alpha = \int_{r=0}^{\infty} |E(r)| dr < \alpha \int_{r=0}^{\infty} |S(r)| dr \leq \alpha$$

producing a contradiction. □

Note that we can make the above proof constructive by considering the distance values $r = y_j$ for each $j \in V$. Lemmas 1 and 2 imply the following.

Theorem 3 $\max_{\emptyset \neq S \subseteq V} f(S) = f^*$, *i.e. the value of the densest subgraph equals the optimal objective value of the LP.*

3 Greedy Algorithm for Densest Subgraph

So we have established that we can solve this problem optimally. We now show that there is a simple 2 approximation. This greedy algorithm due to [2] will be the basis of our distributed algorithm.

Consider the following algorithm:

Algorithm 1 Greedy for Densest Subgraph

Input: A graph $G = (V, E)$

Output: A set $S^* \subseteq V$, such that it maximizes $f(S) = \frac{|E(S)|}{|S|}$.

```
Let S = V, S* = V and f(S*) = f(V).
while S ≠ ∅ do
    Find i_min ∈ S, the vertex of minimum degree in G(S). Delete it from S.
    if f(S) > f(S*) then
        S* = S.
    end if
end while
```

We want to show that this algorithm is a 2 approximation. First we construct a lower bound on the optimal solution by constructing a feasible solution.

Consider the following: For each edge $ij \in E$ assign the edge to one of i and j arbitrarily. Let $d(i)$ be edges assigned to $i \in V$. Let $d_{max} = \max_{i \in V} d(i)$. The following lemma shows d_{max} is an upper bound of OPT.

Lemma 4 $\max_{S \subseteq V}\{f(S)\} \leq d_{max}$ *for any edge assignment.*

Proof: Consider the set S that maximizes $f(S)$. Each edge in $E(S)$ is assigned to exactly one vertex in S. So $|E(S)| \leq |S| d_{max}$. This implies $f(S) = \frac{|E(S)}{|S|} \leq d_{max}$. □

Now consider the following assignment of edges to vertices based on the algorithm. Each edge assigns itself to the first incident vertex deleted by the algorithm, and d_{max} is defined as before. Note that prior lemma still holds. We use this fact to bound the algorithm's objective function value.

Lemma 5 *Let* $\alpha = f(S^*)$ *be the maximum value of* $f(S)$ *over all sets* S *seen by the greedy algorithm. Then* $d_{max} \leq 2\alpha$.

Proof: Fix an iteration of the algorithm and let S be the current set, so algorithm deletes i_{min}. By average argument, let $deg(i_{min})$ be the degree of i_{min}, $deg(i_{min}) \leq \frac{2|E(S)|}{|S|} \leq 2\alpha$ since $\frac{2|E(S)|}{|S|}$ is the average degree. Note that i_{min} is assigned at most this many edges, so $d(i_{min}) \leq deg(i_{min}) \leq 2\alpha$. Thus d_{max} must be at most 2α, since $d_{max} = \max_i\{d(i_{min})\}$ over all iterations of the algorithm. □

We have thus proved the following theorem.

Theorem 6 *Greedy is a* 2 *approximation for the Densest Subgraph problem.*

Now we want to turn this greedy algorithm into a distributed algorithm running in $O(\log n)$ rounds. The greedy algorithm is simulated by the following. Use $deg(i)$ to denote the degree of vertex i.

Algorithm 2 Distributed Greedy for Densest Subgraph

Input: A graph $G = (V, E)$, a parameter ϵ.
Output: A set $S^* \subseteq V$ such that $f(S^*) \geq \frac{1}{2(1+\epsilon)} \max_{S \subseteq V} f(S)$

```
Let S = V, S* = V.
while S ≠ ∅ do
    A(S) = {i ∈ S | deg(i) ≤ 2(1 + ε)f(S)}.
    Remove A(S) from S.
    if f(S) > f(S*) then
        Let S* = S.
    end if
end while
```

First we show that the algorithm gives a $2(1+\epsilon)$ approximation. Then we discuss how to make it distributed.

Lemma 7 *Let $\alpha = f(S^*)$ be the maximum value of $f(S)$ over all sets S seen by the distributed algorithm. Then $d_{max} \leq 2(1+\epsilon)\alpha$.*

Proof: The proof goes in the same way. Fix an iteration of the algorithm and let S be the current set. Each edge assigns itself to first deleted endpoint, breaking ties arbitrarily. Suppose algorithm deletes $A(S)$ in this step. Each $i \in A(S)$ has degree at most $\frac{2(1+\epsilon)|E(S)|}{|S|} \leq 2(1+\epsilon)\alpha$. Notice that each vertex in $A(S)$ is assigned at most so many edges. Thus $d_{max} = \max_i\{i_{max}\} \leq 2(1+\epsilon)\alpha$. □

The next lemma gives an upper bound on the rounds needed by Distribute Greedy.

Lemma 8 *Distributed Greedy can be implemented in $O(\log_{1+\epsilon} n)$ rounds.*

Proof: We have three operations in each iteration of while loop: computing the average degree, finding the minimum degree vertices and deleting low degree vertices. They are all easy to do in $O(1)$ rounds. So we only need to bound iterations of the while loop.

Consider any set S. We have the following:

$$\begin{aligned}
2|E(S)| &= \sum_{i \in A(S)} \deg_S(i) + \sum_{i \in S \setminus A(S)} \deg_S(i) \\
&= 2|S|f(S) \qquad \text{[Definition of density]} \\
&> 2(1+\epsilon)(|S| - |A(S)|)f(S) \quad \text{[Nodes in } S \setminus A(S) \text{ have degree larger than } 2(1+\epsilon)f(S)\text{]} \\
&= 2(1+\epsilon)(|S| - |A(S)|)\frac{|E(S)|}{|S|}
\end{aligned}$$

Rearranging, we have that $\frac{1}{1+\epsilon}|S| > |S| - |A(S)|$ or $|A(S)| > \frac{\epsilon}{1+\epsilon}|S|$.

Thus, the number of nodes decreases by at least a $\frac{1}{1+\epsilon}$ factor each iteration. Thus, the overall algorithm must run in $O(\log_{1+\epsilon} n)$ rounds. □

Putting the results together, we have the following theorem, originally due to [1].

Theorem 9 *There is a $O(\log_{1+\epsilon} n)$ round 2-approximate MPC algorithm for the densest subgraph problem.*

References

[1] Bahman Bahmani, Ravi Kumar, and Sergei Vassilvitskii. Densest subgraph in streaming and mapreduce. *PVLDB*, 5(5):454–465, 2012.

[2] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In Klaus Jansen and Samir Khuller, editors, *Approximation Algorithms for Combinatorial Optimization, Third International Workshop, APPROX 2000, Saarbrücken, Germany, September 5-8, 2000, Proceedings*, volume 1913 of *Lecture Notes in Computer Science*, pages 84–95. Springer, 2000.

[3] A. V. Goldberg. Finding a maximum density subgraph. Technical report, University of California at Berkeley, Berkeley, CA, USA, 1984.