

A Local Algorithm for Finding Dense Subgraphs

Reid Andersen

February 1, 2008

Abstract

We present a local algorithm for finding dense subgraphs of bipartite graphs, according to the definition of density proposed by Kannan and Vinay. Our algorithm takes as input a bipartite graph with a specified starting vertex, and attempts to find a dense subgraph near that vertex. We prove that for any subgraph S with k vertices and density θ , there are a significant number of starting vertices within S for which our algorithm produces a subgraph S' with density $\Omega(\theta/\log n)$ on at most $O(\Delta k^2)$ vertices, where Δ is the maximum degree. The running time of the algorithm is $O(\Delta k^2)$, independent of the number of vertices in the graph.

1 Introduction

Identifying dense subgraphs has become an important task in the analysis of large networks, and a collection of dense subgraphs may reveal a wealth of information about a graph. In particular, dense subgraphs often form the cores of larger communities or clusters in the graph [9].

Kannan and Vinay [8] introduced a notion of density that is well-suited to bipartite graphs representing incidence matrices. As an example, consider a bipartite graph describing the incidences between a set of groups \mathcal{G} and a set of group members \mathcal{M} . The density of the subgraph induced by a set of groups $S \subseteq \mathcal{G}$ and a set of members $T \subseteq \mathcal{M}$, is defined to be

$$d(S, T) = \frac{e(S, T)}{\sqrt{|S|}\sqrt{|T|}},$$

which is the total number of incidences between the groups and members in the subgraph, divided by the geometric mean of the number of groups and number of members in the subgraph. There are a variety of efficient algorithms for finding a subgraph with nearly optimal density according to this definition. Kannan and Vinay gave a spectral algorithm that produces from the largest eigenvector of A

a subgraph whose density is within an $O(\log n)$ factor of optimal. Charikar [2] showed that a subgraph with optimal density can be identified in polynomial time by solving a linear program, and also gave a greedy algorithm that produces a 2-approximation of the densest subgraph in linear time.

In this paper, we present a local algorithm for finding dense subgraphs. Our algorithm takes as input a graph with a specified starting vertex, and attempts to find a dense subgraph near that vertex. We prove the following local approximation guarantee for our algorithm: for any subgraph H with density θ , there are a significant number of starting vertices within H for which our algorithm produces a subgraph with density $\Omega(\theta/\log n)$. The running time of the algorithm is $O(\Delta k^2)$, where k is the number of vertices in H , and where Δ is the maximum degree in the graph.

There are two principal tasks that our local algorithm can perform which, to our knowledge, can not be accomplished by other known algorithms for the densest subgraph problem. The first is to find a dense subgraph near a vertex of interest, while examining only a portion of the entire graph. The second is to find many small dense subgraphs in parallel, which we can accomplish by applying the local algorithm at many different starting vertices. In addition, our algorithm provides an upper bound on the size of the subgraph it produces, which might make it a useful theoretical tool for producing a dense subgraph of a specified size.

To analyze our algorithm, we build upon the spectral techniques developed by Kannan and Vinay, exploiting the close relationship between the densest subgraph of a graph and the largest eigenvalue of the graph's adjacency matrix. We define a deterministic process called the 'pruned growth process', which produces a sequence of vectors, and show that by computing those vectors we can identify a subgraph with high density. We show that these vectors can be rounded at each step to ensure that the number of nonzero elements is small, which decreases the time required to compute them. A similar type of local approximation algorithm has been developed for the related problem of graph partitioning [10, 1]. The densest subgraph problem is the second problem for which this type of local spectral algorithm has been developed.

In Section 2, we state the definition of density introduced by Kannan and Vinay, compare this definition with others that have appeared in the literature, and survey known algorithms for the densest subgraph problem. In Section 3, we define the 'pruned growth process'. In Section 4, we state our local algorithm and analyze its running time and approximation guarantee. In Section 5, we describe an efficient global approximation algorithm for the densest subgraph problem, which will follow easily from our work in the previous sections.

2 Preliminaries and Related Work

Let $G = (V, E)$ be an undirected bipartite graph with adjacency matrix A , and let L and R be the left and right sides of a fixed bipartition. The edges of the graph may be weighted, in which case the entry $A_{i,j}$ is the weight of edge $\{i, j\}$. For any two sets $S \subseteq L$ and $T \subseteq R$, we let (S, T) denote the induced bipartite subgraph of G on the set of vertices $S \cup T$, and we define $e(S, T)$ to be the sum of the weights of the edges between S and T . We will sometimes use the inner product notation $e(S, T) = \langle 1_S A, 1_T \rangle$, where 1_S is the indicator function for membership in S . We define the *support* of a vector x to be the set of vertices on which x is nonzero.

We will identify induced subgraphs of G which are dense according to the following definition, which was introduced by Kannan and Vinay [8].

Definition 1. *For any induced subgraph (S, T) , we define*

$$d(S, T) = \frac{e(S, T)}{\sqrt{|S|}\sqrt{|T|}}.$$

We define $d(A)$ to be the maximum value of $d(S, T)$ over all induced subgraphs.

Our algorithm may also be applied to an arbitrary directed graph, using the following trick. Given a directed graph with vertex set X , define a bipartite graph where $L = R = X$. For each edge $x \rightarrow y$ in the directed graph, place an undirected edge between the copy of x in L and the copy of y in R .

2.1 Related work

A different definition of density was considered in [7, 5, 2].

Definition 2. *Let $G = (V, E)$ be an undirected graph (not necessarily bipartite). For any set $S \subseteq V$, we define*

$$g(S) = \frac{e(S, S)}{|S|}.$$

We define $g(A)$ to be the maximum value of $g(S)$ over all subsets of V .

Both $d(A)$ and $g(A)$ can be computed exactly in polynomial time. Goldberg showed that a set S achieving $g(S) = g(A)$ can be found using maximum flow computations [7]. Such a set can also be found using the parametric flow algorithm of Gallo, Grigoriadis, and Tarjan [5]. Charikar showed that a subgraph (S, T) achieving $d(S, T) = d(A)$ can be found by solving a linear program [2].

Charikar gave greedy 2-approximation algorithms for both $g(A)$ and $d(A)$ [2]. The running time of these algorithms is $O(m)$ in an unweighted graph, and

$O(m + n \log n)$ in a weighted graph. Kannan and Vinay gave a spectral approximation algorithm for $d(A)$, which produces a subgraph (S, T) with density $d(S, T) = \Omega(d(A)/\log n)$ from the top singular vectors of A .

The closely related densest k -subgraph problem is to identify the subgraph with the largest number of edges among all subgraphs of exactly k vertices. This problem is considerably more difficult, and there is a large gap between the best approximation algorithms and hardness results known for the problem (see [4, 3]).

2.2 Comparison of $d(S, T)$ and $g(S)$

It is easier to compare the two objective functions $d(S, T)$ and $g(S)$ if we restrict $g(S)$ to bipartite graphs. In this case, $g(S)$ takes the following form.

Definition 3. *For any subgraph (S, T) , we define*

$$g(S, T) = \frac{e(S, T)}{|S| + |T|}.$$

We define $g(A)$ to be the maximum value of $g(S, T)$ over all induced subgraphs.

The two objective functions $d(S, T)$ and $g(S, T)$ are far apart when S and T have very different sizes. The quantities $d(A)$ and $g(A)$ can also be far apart. In the complete bipartite graph $K_{a,b}$, we have $d(A) = \sqrt{ab}$, while $g(A) = ab/(a+b)$. In the case where $a = 1$, we have $d(A) = \sqrt{b}$ while $g(A) = b/(b+1) \sim 1$.

The relative merits of $d(S, T)$ and $g(S)$ as objective functions for density were discussed in [2, 8]. In this paper, we consider $d(S, T)$ because it is more amenable to approximation by spectral algorithms than $g(S)$, not because we prefer it as an objective function. The largest eigenvalue of the adjacency matrix A is closely related to $d(A)$. We know of no similar result for $g(A)$, and we do not know how to produce a local algorithm for the objective function $g(S)$.

3 The pruned growth process

We now define the deterministic process that will be the basis for our local algorithm. The process generates a sequence of vectors x_0, \dots, x_T from a starting vector x_0 . The main operation performed at each step is multiplication by the adjacency matrix A , as in the power method. The resulting vector is then rounded by making each entry a power of 2, and then pruned by setting to zero each entry whose value is below a certain threshold. These steps reduce the number of possible values in the vector and reduce the size of the support, minimizing the amount of computation required.

Definition 4. Given a vector z , we define $\text{round}(z)$ to be the vector obtained by rounding each entry of the vector z up to the nearest power of 2,

$$[\text{round}(z)](u) = 2^i, \text{ where } i \text{ is the smallest integer such that } 2^i \geq z(u).$$

Given a vector z and a nonnegative real number ϵ , we define $\text{prune}_\epsilon(z)$ to be the vector obtained by setting to zero any entry of z whose value is at most $\epsilon\|z\|$,

$$[\text{prune}_\epsilon(z)](u) = \begin{cases} z(u) & \text{if } z(u) > \epsilon\|z\|, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 5. Given a starting vector x_0 with entries from $\{0, 1\}$, and a sequence of real numbers $\epsilon_t \in [0, 1]$, we define the pruned growth process to be the sequence of vectors x_0, \dots, x_T defined by the following rule:

$$x_{t+1} = \text{prune}_{\epsilon_{t+1}}(\text{round}(x_t A)).$$

Notice that each entry of x_{t+1} is either zero or a power of two.

Definition 6. Given the vectors x_0, \dots, x_T of the pruned growth process, we define X_i^t to be the set of vertices where $x_t(v) = 2^i$, and define Y_i^t to be the set of vertices where $\text{round}(x_t A)(v) = 2^i$.

We will eventually show that a subgraph with high density can be found whenever the norms $\|x_i\|$ of the pruned growth process vectors grow quickly. The following lemma shows that if none of the subgraphs (X_i^t, Y_j^t) has high density, then $\|x_t\|$ is not much larger than $\|x_{t-1}\|$.

Lemma 1. If $d(X_i^t, Y_j^t) \leq \theta$ for all i, j , then,

$$\|x_{t+1}\| \leq \|\text{round}(x_t A)\| \leq 2\theta\|x_t\| \log \frac{2\Delta}{\epsilon_t}.$$

Proof. We will write $\|\text{round}(x_t A)\|^2$ in terms of the densities $d(X_i^t, Y_j^t)$.

$$\begin{aligned} \|\text{round}(x_t A)\|^2 &= \langle \text{round}(x_t A), \text{round}(x_t A) \rangle \\ &\leq \langle 2x_t A, \text{round}(x_t A) \rangle \\ &= \left\langle 2 \sum_i 2^i 1_{X_i^t} A, \sum_j 2^j 1_{Y_j^t} \right\rangle \\ &= 2 \sum_{i,j} 2^i 2^j \langle 1_{X_i^t} A, 1_{Y_j^t} \rangle \\ &= 2 \sum_{i,j} 2^i 2^j d(X_i^t, Y_j^t) \sqrt{|X_i^t|} \sqrt{|Y_j^t|} \\ &\leq 2\theta \left(\sum_i \sqrt{|X_i^t|} 2^i \right) \left(\sum_j \sqrt{|Y_j^t|} 2^j \right). \end{aligned}$$

In the sum above, we need only sum over those i where X_i^t is nonempty. There are at most $\log \frac{1}{\epsilon_t}$ such values, because every nonzero value in x_t is at most $\|x_t\|$ and at least $\epsilon_t \|x_t\|$. We now apply the Cauchy-Schwarz inequality to show

$$\sum_i \sqrt{|X_i^t|} 2^i \leq \left(\sum_i |X_i^t| 2^{2i} \right)^{1/2} \left(\sum_i 1 \right)^{1/2} \leq \|x_t\| \sqrt{\log \frac{1}{\epsilon_t}}.$$

Similarly, we need only sum over those j where Y_j^t is nonempty. There are at most $\log \frac{2\Delta}{\epsilon_t}$ such values, because every nonzero value of $\text{round}(x_t A)$ is at most $\|\text{round}(x_t A)\| \leq 2\Delta \|x_t\|$, and at least $\epsilon_t \|x_{t-1}\|$. We apply the Cauchy-Schwarz inequality again to show

$$\sum_j \sqrt{|Y_j^t|} 2^j \leq \left(\sum_j |Y_j^t| 2^{2j} \right)^{1/2} \left(\sum_j 1 \right)^{1/2} \leq \|\text{round}(x_t A)\| \sqrt{\log \frac{2\Delta}{\epsilon_t}}.$$

Then,

$$\begin{aligned} \|\text{round}(x_t A)\|^2 &\leq 2\theta \|x_t\| \|\text{round}(x_t A)\| \sqrt{\log \frac{1}{\epsilon_t}} \sqrt{\log \frac{2\Delta}{\epsilon_t}} \\ &\leq 2\theta \|x_t\| \|\text{round}(x_t A)\| \log \frac{2\Delta}{\epsilon_t}. \end{aligned}$$

The lemma follows. \square

4 Local approximation algorithm

In this section, we will state and analyze a local algorithm for finding dense subgraphs. The input to the algorithm is a graph, along with a *starting vertex* v and a *target size* K . We will prove that the running time of the algorithm depends mainly on the target size K , and is independent of the number of vertices in the graph. We will prove that for any subgraph (S, T) , there are a significant number of starting vertices in S for which the algorithm produces a subgraph whose density is within an $O(\log n)$ factor of $d(S, T)$.

LocalDensity(v, K)

Input: A vertex v and a target size K .

Output: A subgraph (X, Y) .

1. Let $x_0 = 1_v$, let $T = \log(\sqrt{2|K|})$, and let $\epsilon_t = \frac{1}{8K} 2^{-t}$.
2. Compute the vectors x_0, \dots, x_T of the pruned growth process.
3. Compute $d(X_i^t, Y_j^t)$ for each pair i, j and each time $t < T$.
4. Output the subgraph (X_i^t, Y_j^t) with the highest density.

Theorem 1. *Let (S, T) be a subgraph such that $d(S, T) \geq 2\theta$. Then there exists a set $S_\theta \subseteq S$, with the following properties.*

1. $e(S_\theta, T) \geq e(S, T)$,
2. *If $v \in S_\theta$ and $K \geq \max(|S|, |T|)$, then $\text{LocalDensity}(v, K)$ outputs a subgraph (X, Y) such that*

$$d(X, Y) \geq \frac{\theta}{8 \log 16 \Delta K} = \Omega\left(\frac{\theta}{\log n}\right).$$

Theorem 2. $\text{LocalDensity}(v, K)$ runs in time $O(\Delta K^2)$.

The proofs of Theorems 1 and 2 are given in section 4.2.

4.1 Lower bounds on growth within a dense subgraph

The main step in analyzing the algorithm LocalDensity is to prove a lower bound on the growth of the norms $\|x_t\|$. We will use the fact that the maximum density $d(A)$ gives a lower bound on the largest eigenvalue of A .

Fact 1. *Let A be the adjacency matrix of an undirected graph, and let λ be the largest eigenvalue of A . Then, $\lambda \geq d(A)$. Furthermore, there is an eigenvector ϕ with eigenvalue λ whose entries are nonnegative.*

Proof. To prove that $\lambda \geq d(A)$, notice that for any sets $S \subseteq L$ and $T \subseteq R$,

$$\lambda \geq \max_{x, y} \frac{\langle xA, y \rangle}{\|x\| \|y\|} \geq \left\langle \frac{1_S}{\sqrt{|S|}} A, \frac{1_T}{\sqrt{|T|}} \right\rangle = \frac{e(S, T)}{\sqrt{|S|} \sqrt{|T|}} = d(S, T).$$

It is not hard to see that if ϕ is an eigenvector with eigenvalue λ , then the vector whose entries are the absolute values of the entries of ϕ is also an eigenvector with eigenvalue λ . \square

The fact above implies the lower bound $\|x_0 A^t\| \geq \langle \phi, x_0 \rangle d(A)^t$, which depends on the maximum density $d(A)$. To analyze the local algorithm, we will give a lower bound that depends on the density of a particular subgraph (S, T) containing the starting vertex. Specifically, we will show that for many vertices in the set S , we can give a bound of the form $\|x_0 A^t\| = \Omega(d(S, T)^t)$ with a not-too-small constant term. We will do so by considering how the pruned growth process would behave if it were restricted to the induced subgraph (S, T) .

Definition 7. *For any induced subgraph (S, T) , we define $A_{(S, T)}$ to be the restriction of the adjacency matrix A to (S, T) ,*

$$A_{(S, T)}(x, y) = \begin{cases} A(x, y) & \text{if } x \in S \text{ and } y \in T, \text{ or if } x \in T \text{ and } y \in S. \\ 0 & \text{otherwise.} \end{cases}$$

The following lemma identifies, for any subgraph (S, T) , a set of starting vertices for which we can give a good lower bound on the norms $\|x_t\|$. This set of good starting vertices touches at least half of the edges in the induced subgraph (S, T) .

Lemma 2. *If (S, T) is a subgraph such that $d(S, T) \geq 2\theta$, then there exists a subset $S_\theta \subseteq S$ with the following properties:*

1. $e(S_\theta, T) \geq e(S, T)/2$
2. For each $v \in S_\theta$, there is a nonnegative unit vector ψ such that
 - (a) $\text{Support}(\psi) \subseteq S \cup T$,
 - (b) $\psi A \geq \theta \psi$,
 - (c) $\psi(v) \geq \frac{1}{\sqrt{2|S|}}$.

Proof. Let S_θ be the largest subset of S for which property (2) holds, and consider the set $S' = S \setminus S_\theta$. If S_θ does not satisfy property (1), then

$$e(S', T) = e(S, T) - e(S_\theta, T) \geq \frac{e(S, T)}{2},$$

and so $d(S', T) \geq d(S, T)/2 \geq \theta$.

Let λ be the largest eigenvalue of $A_{(S', T)}$. We know from fact 1 that there is an eigenvector ψ of $A_{(S', T)}$ whose entries are all nonnegative, and whose corresponding eigenvalue λ satisfies

$$\lambda \geq d(S', T) \geq \theta.$$

It is easy to see that ψ satisfies properties (a) and (b). We will now identify a vertex in S' for which $\psi(v) \geq 1/\sqrt{2|S|}$. This will imply that v is in S_θ , which will show that S_θ must satisfy property (1), and thus complete the proof.

Let $\psi_{S'}$ and ψ_T be the projections of ψ onto S' and T , and observe that $\|\psi_{S'}\| = \|\psi_T\| = \frac{1}{\sqrt{2}}$. This is true because $\psi_{S'} A_{(S', T)} = \lambda \psi_T$, which implies that $\lambda \|\psi_{S'}\| \geq \|\psi_{S'} A_{(S', T)}\| = \lambda \|\psi_T\|$. There must at least one vertex v in S' which satisfies $\psi(v) \geq 1/\sqrt{2|S'|}$, since otherwise we would have $\|\psi_{S'}\|^2 < 1/2$. \square

4.2 Analysis of the local algorithm

Proof of Theorem 1. We will prove that for each vertex v in the set S_θ , which was described in Lemma 2, the algorithm `LocalDensity(v, K)` outputs a subgraph with density at least $\theta/8L$, where $L = \log(2\Delta/\epsilon_0) \leq (\log 16\Delta K)$, provided that $K \geq \max(|S|, |T|)$. The theorem will follow.

Let x_0, \dots, x_T be the pruned growth process vectors computed by the algorithm. We will assume that the algorithm does not find a subgraph with the desired density, and derive a contradiction. That is, we assume that for each i, j , and each time $t < T$, we have $d(X_i^t, Y_j^t) < \frac{\theta}{8L}$. Under this assumption, Lemma 1 shows that for every $t \leq T$,

$$\begin{aligned} \|x_{t+1}\| &\leq \|\text{round}(x_t A)\| \\ &< \left(2 \log \frac{2\Delta}{\epsilon_t}\right) \left(\frac{\theta}{8L}\right) \|x_t\| \\ &\leq \left(\frac{\theta}{4}\right) \|x_t\|. \end{aligned}$$

Since $\|x_0\| = 1$, this implies

$$\|x_t\| \leq \|\text{round}(x_{t-1} A)\| < \left(\frac{\theta}{4}\right)^t \quad \text{for every } t \leq T. \quad (1)$$

Since $v \in S_\theta$, there exists a nonnegative vector ψ such that $\psi A \geq \theta \psi$, such that $\text{Support}(\psi) \subseteq S \cup T$, and such that $\psi(v) \geq \frac{1}{\sqrt{2|S|}}$, as stated in Lemma 2. We will prove the following lower bound on the inner product of x_t with ψ .

$$\langle x_t, \psi \rangle \geq \frac{1}{\sqrt{2|S|}} (\theta/2)^t \quad \text{for every } t \leq T. \quad (2)$$

When we prove equation (2), it will contradict equation (1) when $t = T = \log(\sqrt{2|S|})$, and we will be done.

We will prove that equation (2) holds by induction. We know it holds for $t = 0$. The only difficulty in the induction step is to bound the effect of the pruning step on the projection of x_t onto ψ . We define r_t to be the vector that is removed during the pruning step.

$$\begin{aligned} r_t &= \text{round}(x_{t-1} A) - x_t \\ &= \text{round}(x_{t-1} A) - \text{prune}_{\epsilon_t}(\text{round}(x_{t-1} A)). \end{aligned}$$

The value of r_t at any given vertex is at most $\epsilon_t \|\text{round}(x_{t-1} A)\|$. Since the support of ψ is contained in $S \cup T$, and the support of r_t is contained in either L or R , the intersection of the two supports contains at most $\max(|S|, |T|)$ vertices. The inner product of r_t and ψ can then be bounded as follows.

$$\begin{aligned} \langle r_t, \psi \rangle &\leq \epsilon_t \|\text{round}(x_{t-1} A)\| \sqrt{|\text{Support}(r_t) \cap \text{Support}(\psi)|} \\ &\leq \epsilon_t \|\text{round}(x_{t-1} A)\| \sqrt{K}. \end{aligned}$$

We can now bound $\langle x_t, \psi \rangle$ in terms of $\langle x_{t-1}, \psi \rangle$.

$$\begin{aligned}\langle x_t, \psi \rangle &= \langle \text{round}(x_{t-1}A) - r_t, \psi \rangle \\ &= \langle \text{round}(x_{t-1}A), \psi \rangle - \langle r_t, \psi \rangle \\ &\geq \theta \langle x_{t-1}, \psi \rangle - \epsilon_t \|\text{round}(x_{t-1}A)\| \sqrt{K}.\end{aligned}$$

We now assume that the induction hypothesis holds for $t - 1$, which means $\langle x_{t-1}, \psi \rangle \geq (1/\sqrt{2|S|})(\theta/2)^{t-1}$. Recall that we have assumed for the sake of contradiction that $\|x_t\| \leq \|\text{round}(x_{t-1}A)\| < (\theta/4)^t$. We will now show that the induction hypothesis holds for t .

$$\begin{aligned}\langle x_t, \psi \rangle &\geq \left(\frac{\theta}{\sqrt{2|S|}} \left(\frac{\theta}{2} \right)^{t-1} \right) - \left(\epsilon_t \sqrt{K} \left(\frac{\theta}{4} \right)^t \right) \\ &\geq \left(\frac{\theta}{2} \right)^t \left(\frac{2}{\sqrt{2|S|}} - 4\epsilon_t 2^{-t} \sqrt{K} \right) \\ &\geq \left(\frac{\theta}{2} \right)^t \frac{1}{\sqrt{2|S|}}.\end{aligned}$$

The last step follows because we have set ϵ_t so that

$$\epsilon_t = \frac{2^{-t}}{8K}.$$

This completes the proof. \square

Proof of Theorem 2. We bound the running time of `LocalDensity`(v, K) by bounding the number of vertices in the support of x_t at each step. Since x_t is at least $\epsilon \|x_t\|$ wherever it is nonzero, we have

$$\|x_t\|^2 \geq |\text{Support}(x_t)| \epsilon^2 \|x_t\|^2,$$

and so

$$|\text{Support}(x_t)| \leq \frac{1}{\epsilon^2}.$$

We can compute x_{t+1} from x_t and compute the density of each subgraph (X_i^t, Y_j^t) in time proportional to the sum of the degrees of the vertices in $\text{Support}(x_t)$, which is at most

$$O(\Delta |\text{Support}(x_t)|) = O(\Delta / \epsilon_t^2) = O(\Delta K^2 2^{-2t}).$$

The total running time is therefore

$$\sum_{t=0}^T O(\Delta K^2 2^{-2t}) = O(\Delta K^2).$$

\square

5 An approximation algorithm for $d(A)$

As a simple application of the techniques developed in the previous sections, we give an $O(\log n)$ -approximation algorithm for the globally optimum density $d(A)$ by simulating the pruned growth process for $O(\log n)$ steps. The algorithm produces a subgraph (S, T) with density $\Omega(d(A)/\log n)$ in time $O(m \log \Delta/d)$, where Δ is the maximum degree in the graph, and d is the average degree. The algorithm requires $O(\log n)$ passes through the collection of adjacency lists describing the graph, and requires only $O(n \log \log n)$ bits of additional storage. This provides an efficient way to implement the spectral approximation algorithm of Kannan and Vinay [8], which has the same $O(\log n)$ approximation guarantee and requires computing the largest eigenvalue of A .

Density:

Run the following procedure twice with $x_0 = 1_L$ and $x_0 = 1_R$:

1. Let $T = \log 2\sqrt{n}$ and $\epsilon_t = \frac{2^t}{8\sqrt{n}}$.
2. Compute the pruned growth process vectors x_0, \dots, x_T .
3. Compute $d(X_i^t, Y_j^t)$ for each pair i, j and each time $t < T$.
4. Output the densest subgraph among the sets (X_i^t, Y_j^t) .

Theorem 3. *For at least one of the two starting vectors 1_L and 1_R , there exists a time $t \leq T$ and two indices i and j such that the subgraph (X, Y) output by the algorithm satisfies*

$$d(X, Y) \geq \frac{\lambda}{(8 + 4 \log n)} \geq \frac{d(A)}{(8 + 4 \log n)}.$$

Theorem 4. *Density runs in time $O(m(1 + \log \frac{\Delta}{d}))$, where Δ is the maximum degree in the graph, and d is the average degree. The algorithm requires $O(n \log \log n)$ bits of additional storage.*

Proof of Theorem 3. Let λ be the largest eigenvalue of A , and let ϕ be an eigenvector with eigenvalue λ whose entries are nonnegative. Because ϕ is non-negative, $\langle 1_V, \phi \rangle \geq 1$. We will assume that 1_L has a larger inner product with ϕ than 1_R , so that $\langle 1_L, \phi \rangle \geq (1/2) \langle 1_V, \phi \rangle \geq 1/2$. We let $x_0 = 1_L$, and consider the vectors x_0, \dots, x_T computed by the algorithm.

We assume that $d(X_i^t, Y_j^t) < \lambda/8 \log(2\Delta/\epsilon_0) \leq \lambda/(8 + 4 \log n)$ for every i, j , and $t \leq T$, and derive a contradiction. Under this assumption, Lemma 1 shows

that for every $t \leq T$,

$$\begin{aligned}\|x_{t+1}\| &\leq \|\text{round}(x_t A)\| \\ &< \left(2 \log \frac{2\Delta}{\epsilon_t}\right) \left(\frac{\lambda}{8 \log(2\Delta/\epsilon_0)}\right) \|x_t\| \\ &\leq \left(\frac{\lambda}{4}\right) \|x_t\|.\end{aligned}$$

Since $\|x_0\| \leq \sqrt{n}$, this implies

$$\|x_t\| < \sqrt{n} \left(\frac{\lambda}{4}\right)^t \quad \text{for every } t \leq T. \quad (3)$$

We will soon prove the following lower bound.

$$\langle x_t, \phi \rangle \geq \langle 1_L, \phi \rangle (\lambda/2)^t \quad \text{for every } t \leq T. \quad (4)$$

When $t = T = \log(2\sqrt{n})$, this will imply

$$\|x_T\| \geq \langle x_T, \phi \rangle \geq \frac{1}{2} (\lambda/2)^T \geq \sqrt{n} (\lambda/4)^T,$$

which will contradict equation (3), completing the proof.

We will prove by induction that equation (4) holds for every $t \leq T$. It holds trivially for $t = 0$. We define r_t to be the vector lost in the pruning step,

$$\begin{aligned}r_t &= \text{round}(x_{t-1} A) - x_t \\ &= \text{round}(x_{t-1} A) - \text{prune}_{\epsilon_t}(\text{round}(x_{t-1} A)).\end{aligned}$$

The value of r_t at any given vertex is at most $\epsilon_t \|\text{round}(x_{t-1} A)\| \leq 2\lambda\epsilon_t \|x_{t-1}\|$. Because ϕ is nonnegative, $\langle r_t, \phi \rangle \leq 2\lambda\epsilon_t \|x_{t-1}\| \langle 1_V, \phi \rangle$. In fact, we have the slightly stronger statement $\langle r_t, \phi \rangle \leq 2\lambda\epsilon_t \|x_{t-1}\| \langle 1_L, \phi \rangle$, because the support of r_t is contained in either L or R , and 1_L has a larger inner product with ϕ . We can now bound $\langle x_t, \phi \rangle$ in terms of $\langle x_{t-1}, \phi \rangle$.

$$\begin{aligned}\langle x_t, \phi \rangle &= \langle \text{round}(x_{t-1} A) - r_t, \phi \rangle \\ &= \langle \text{round}(x_{t-1} A), \phi \rangle - \langle r_t, \phi \rangle \\ &\geq \lambda \langle x_{t-1}, \phi \rangle - 2\lambda\epsilon_t \|x_{t-1}\| \langle 1_L, \phi \rangle.\end{aligned}$$

We will assume that the induction hypothesis holds for $t-1$, which means that $\langle x_{t-1}, \phi \rangle \geq \langle 1_L, \phi \rangle (\lambda/2)^{t-1}$, and we have already assumed for the sake of contradiction that $\|x_t\| < \sqrt{n} (\lambda/4)^t$. We now show that the induction hypothesis holds for t .

$$\begin{aligned}\langle x_t, \phi \rangle &\geq \lambda \langle 1_L, \phi \rangle (\lambda/2)^{t-1} - 2\lambda\epsilon_t \langle 1_L, \phi \rangle \sqrt{n} (\lambda/4)^{t-1} \\ &\geq \langle x_0, \phi \rangle (\lambda/2)^t (2 - 8\epsilon_t 2^{-t} \sqrt{n}) \\ &\geq \langle x_0, \phi \rangle (\lambda/2)^t.\end{aligned}$$

The last step follows because we have set ϵ_t so that

$$\epsilon_t = \frac{2^t}{8\sqrt{n}}.$$

This completes the proof. \square

Proof of Theorem 4. We can bound the running time of the algorithm by bounding the number of vertices in the support of x_t . Since x_t is at least $\epsilon_t \|x_t\|$ wherever it is nonzero, we have

$$\|x_t\|^2 \geq |\text{Support}(x_t)| \epsilon_t^2 \|x_t\|^2,$$

and so

$$|\text{Support}(x_t)| \leq \frac{1}{\epsilon_t^2} \leq n 2^{-2(t-3)}.$$

We can compute x_{t+1} from x_t and compute the density of each subgraph (X_i^t, Y_j^t) in time proportional to the number of edges incident with $\text{Support}(x_t)$,

$$\begin{aligned} |e(\text{Support}(x_t), V)| &\leq \min(m, \Delta |\text{Support}(x_t)|) \\ &\leq \min(m, \Delta n 2^{-2(t-3)}). \end{aligned}$$

The total running time over all T steps is at most

$$\begin{aligned} \sum_{t=0}^T \min(m, \Delta n 2^{-2(t-3)}) &\leq \left(\frac{1}{2} \log(n\Delta/m) + 3 \right) m + \sum_{t=\frac{1}{2} \log(n\Delta/m)+3}^T \Delta n 2^{-2(t-3)} \\ &\leq \left(\frac{1}{2} \log(n\Delta/m) + 3 \right) m + 2m \\ &= O(m \log(\Delta/d) + m). \end{aligned}$$

To bound the amount of space used by the algorithm, notice that storing the vector x_t requires $n \log \log \frac{1}{\epsilon} = O(n \log \log n)$ bits, since each vertex takes one of $\log \frac{1}{\epsilon}$ possible values. We need only store two vectors at a given time, x_t and $\text{round}(x_t A)$, so the total amount of storage required is $O(n \log \log n)$ bits. \square

6 Conclusion

We have shown that it is possible to find a dense subgraph near a given vertex without examining the entire graph. The running time of our local algorithm is quadratic in terms of the target size K , where K must be at least as large as $|S| + |T|$ to produce an approximation of the subgraph (S, T) . We conjecture that a better local algorithm exists. In particular, it would be nice to have an algorithm whose running time depends on the size of the subgraph (X, Y) that is produced, rather the subgraph (S, T) whose density is approximated.

References

- [1] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *Proc. 47th Annual Symposium on Foundations of Computer Science*, (2006).
- [2] M. Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proc. Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, (2000).
- [3] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3), 410-421, (2001).
- [4] U. Feige and M. Seltser. On the densest k-subgraph problem. Weizmann Institute Technical Report CS 97-16, (1997).
- [5] G. Gallo, M.D. Grigoriadis, and R. Tarjan. A Fast Parametric Maximum Flow Algorithm and Applications. In *Proc. 39th Annual IEEE Symposium on Foundations of Computer Science*, 370-378 (1998).
- [6] D. Gibson, R. Kumar, and A. Tomkins. Discovering Large Dense Subgraphs in Massive Graphs. In *Proc. 31st VLDB Conference*, (2005).
- [7] A. Goldberg. Finding a maximum density subgraph. Technical report UCB CSD 84/71, University of California, Berkeley, (1984).
- [8] R. Kannan and V. Vinay. Analyzing the Structure of Large Graphs. *Manuscript*, (1999).
- [9] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. In *Proc. 8th WWW Conference*, Computer Networks, 31(11-16):1481-1493, (1999).
- [10] D. Spielman and S.H. Teng. Nearly-Linear Time Algorithms for Graph Partitioning, Graph Sparsification, and Solving Linear Systems. In *Proc. 36th Annual ACM Symposium on Theory of Computing*, (2004).