6.842 Randomness and Computation	March 5, 2014

Lecture 9

Lecturer: Ronitt Rubinfeld | *Scribe: Pritish Kamath*

1 Lecture overview

The problem of "undirected s-t connectivity" is given a graph G and vertices $s, t \in V(G)$, return 'yes' if s and t are in the same connected component of G, return 'no' otherwise. We show how to solve this problem *deterministically* in *logspace*.

This result was first shown by Omer Reingold [1]. Most of the content in this scribe has been adapted from Ronitt's scanned notes and Chapter 21 in the text by Arora-Barak [2].

2 Preliminaries

We define some notions from spectral graph theory which will be relevant for the algorithm.

Definition 1 (Normalized Adjacency Matrix, Eigenvalues)
For any d-regular graph G, let A_G denote the normalized adjacency matrix of G, namely,

$$(A_G)_{ij} = \begin{cases} 1/d & \textit{if } (i,j) \in E(G) \\ 0 & \textit{if } (i,j) \notin E(G) \end{cases}$$

Also, let $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$ be the eigenvalues of A_G. We would refer to these eigenvalues as $\lambda_i(G)$. Also, when the context is clear, we would often refer to them as simply λ_i.

Definition 2 ((N, d, λ)-expander)
An N-vertex, d-regular graph G is said to be an (N, d, λ)-expander if $\lambda_2(G) \leq \lambda$

The following theorem follows from Cheeger's inequality, and was first shown independently by Tanner and by Alon and Milman.

Theorem 3 (Tanner, Alon-Milman) *For any $\lambda < 1$, there exists $\varepsilon > 0$ such that for any (N, d, λ)-expander $G = (V, E)$, it holds that, for any $S \subseteq V$ such that $|S| \leq N/2$,*

$$|\mathcal{N}(S)| \geq (1 + \varepsilon)|S|$$

where $\mathcal{N}(S) \subseteq V$ is a subset of vertices consisting of S and the neighbors of S.

The above theorem implies that for an (N, d, λ)-expander, the number of neighbors of S is at least constant fraction of $|S|$ (for $|S| \leq N/2$), where the constant ε depends only on λ and is independent of the size of the graph. This implies that the diameter of G is at most $O(\log N)$ because of the following argument: Consider any two vertices s and t in G. The number of vertices within distance r of s (or t) grows *exponentially* in r ($(1 + \varepsilon)^r$ to be precise), and hence there are more than $N/2$ vertices within a $O(\log N)$ distance of s (or t). Thus, the distance between any s and t is also $O(\log N)$.

3 Overview of Algorithm for Undirected s-t connectivity

3.1 An easy case

Suppose the given graph is such that every component of the graph is an (N, d, λ)-expander (for potentially different values of N), where d and $\lambda < 1$ are universal constants.

We have seen that if G is an (N, d, λ)-expander, then the diameter of G is $\Delta = O(\log N)$. Since d is a constant independent of N, we can trace out all d^{Δ} paths of length Δ starting from s in logarithmic space[1]. If t lies in the same component as s then we would find t in the depth-first search and return 'yes', else we would return 'no'.

3.2 Reducing to the easy case

To solve the problem of undirected s-t connectivity, we would convert our problem instance (G, s, t) to a problem instance (G^*, s^*, t^*) where,

- $|G^*| = O(\text{poly}(|G|))$
- The vertices/edges of G^* can be enumerated on the fly in logspace, and thus we wouldn't have to construct G^* *explicitly* in memory
- s^* and t^* would be connected in G^* *if and only if* s and t are connected in G
- G^* would be d-regular and every component of G^* would have $\lambda_2 \leq \lambda < 1$, where d and λ would be some universal constants

If we had such a reduction we could check s-t connectivity in G by simply checking s^*-t^* connectivity in G^*, which can be done in $O(\log |G^*|) = O(\log |G|)$ space (the easy case!).

3.3 Algorithm overview

We want to reduce G to G^* where $\lambda_2(G^*) \leq \lambda < 1$. This is done via an iterative process of alternately applying "powering" and "replacement product". The "powering" step would decrease λ, but would blow up the degree. The "replacement product" would bring down the degree to a constant, while ensuring that λ does not increase. The iterations would go as $G_0 \to G_1 \to G_2 \to \cdots \to G_\ell$, where G_ℓ will have $\lambda \leq 7/10$ and constant degree.

The steps of the complete algorithm are as follows,

- **Pre-processing step:** Convert G to G_0 where G_0 is non-bipartite and 3-regular.
- For $i = 0, 1, \cdots, \ell$ (for a suitable choice of ℓ, which will be chosen later)
  - ◇ **Replacement product:** Obtain G' by applying *replacement product* on G_i
  - ◇ **Powering stage:** Construct G_{i+1} by powering G' some t times, i.e. $G_{i+1} \leftarrow (G')^t$
- Run depth-first search on G_ℓ (uses only logspace - easy case!)

[1] this is equivalent to doing a depth-first search, where the stack size would never exceed Δ, and hence the total space requirement is only $O(\Delta)$

4 Powering of Graphs

To start with, we have the following lemma (which we state without proof),

Lemma 4 *For any non-bipartite, d-regular graph G on N vertices :* $\lambda_2(G) \leq 1 - \frac{1}{dN^2}$

That is, for any non-bipartite, regular graph, we have that λ_2 is bounded away from 1 by an inverse-polynomial factor. A natural idea to reduce λ_2 is to *power* the graph.

Definition 5 (Graph Powering) *For any graph G, define G^t as: $V(G^t) = V(G)$ and $(u, v) \in E(G^t)$ if there exists a walk of length exactly t from u to v (that is, there exist $v_1, v_2, \cdots, v_{t-1}$ such that $(u, v_1), (v_1, v_2), \cdots, (v_{t-1}, v) \in E(G)$), with multiple edges if there are multiple such walks.*

It is easy to see that $A_{G^t} = A_G^t$, and hence $\lambda_2(G^t) = \lambda_2(G)^t$

Thus, by powering the graph, we are able to decrease the value of λ_2. However, the side-effect of powering is that that if we start with a d-regular graph G, then G^t has degree d^t. Since we want $\lambda \leq 1/2$ we would need $t = \theta(\log n)$, which would make $d^t = \theta(\text{poly}(n))$, and hence the *easy case* analysis as described above would not work as is for G^t.

5 Replacement Product

To fix this issue of larger degree we use the *replacement product* technique to reduce the degree of the graph, without increasing the value of λ_2. First we define the notion of rotation maps for regular graphs.

Definition 6 (Rotation Maps) *For a d-regular graph G, consider an ordering on the neighbors of every vertex, numbering them from 1 to d. The* rotation map *$\hat{G} : V(G) \times [d] \to V(G) \times [d]$, is defined as $\hat{G}(\langle u, i \rangle) = \langle v, j \rangle$ if v is the i-th neighbor of u and u is the j-th neighbor of v.*

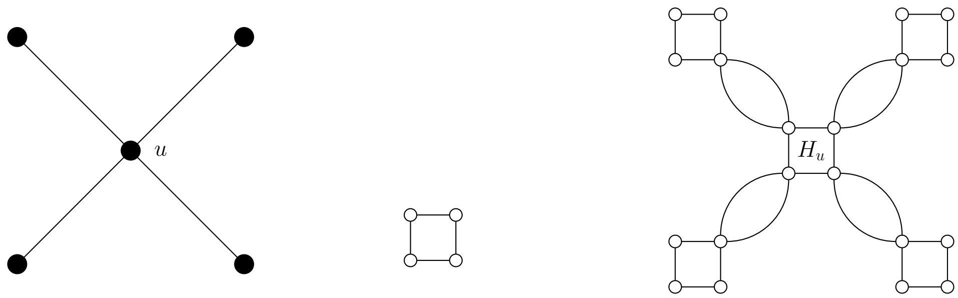
Definition 7 (Replacement Product) *Let G be a N-vertex, D-regular graph, and let H be a D-vertex, d-regular graph. The* replacement product *of G and H, denoted by $G Ⓡ H$ is defined as follows,*

- *For every vertex u of G, the graph $G Ⓡ H$ has a copy of H (including both edges and vertices), and we denote this copy of H as H_u.*
- *If u, v are two neighboring vertices in G then we place d parallel edges between the i-th vertex in H_u and the j-th vertex in H_v, where $\hat{G}(u, i) = (v, j)$.*

Note that $G Ⓡ H$ is $(2d)$-regular and has ND vertices.

The replacement product has been summarized in Figure 1. [*Note:* This example is only for illustrative purposes; G shown below is non-regular, although the definition of replacement product is for regular graphs only.]

For the main algorithm, we would like to have a constant sized expander H, that is $\lambda_2(H)$ should be a constant less than 1. To this end, we have the following lemma,



G **H** **G®H**

Figure 1: Replacement product $G \circledR H$

Lemma 8 (Existence of Expanders) *(from class notes : citation needed)*
For a sufficiently large constant d_0, there exists a $(d_0^{16}, d_0, 1/2)$ expander.

We also require that $\lambda_2(G \circledR H) \leq \lambda_2(G)$ if $\lambda_2(H) \leq 1/2$. This is true due to the following lemma,

Lemma 9 (Expansion of Replacement Product) *(from class notes : citation needed)*
If G is an (N, D, λ)-expander and H is an (D, d, α) expander, then $G \circledR H$ is an $(ND, 2d, \lambda_{G \circledR H})$, where,

$$\frac{1}{2}(1-\lambda)(1-\alpha^2) \leq 1 - \lambda_{G \circledR H}$$

Using the above lemma, and the fact that H is a $(d_0^{16}, d_0, 1/2)$ expander, and assuming that $\lambda > 2/3$, we get,

$$\begin{aligned}
\lambda_{G \circledR H} &\leq 1 - \frac{1}{2} \cdot (1-\alpha^2) \cdot (1-\lambda) \\
&\leq 1 - \frac{1}{2} \cdot \frac{3}{4} \cdot (1-\lambda) \quad \text{[since } \alpha \leq 1/2\text{]} \\
&= 1 - \frac{3}{8} \cdot (1-\lambda) \\
&\leq 1 - \frac{1}{3} \cdot (1-\lambda) \\
&= \frac{2}{3} + \frac{\lambda}{3} \\
&\leq \lambda \qquad \text{[since } \lambda > 2/3\text{]}
\end{aligned}$$

Thus we obtain that λ_2 of every connected component of $G \circledR H$ is less than λ, where λ is an upper bound on the second eigenvalue of any connected component of G.

6 Putting it all together

We are finally ready to present the final algorithm.

- **Pre-processing step:** Convert G to G_0 where $G_0 = G Ⓡ C_n$, where C_n is the n-cycle. This would make G_0 a 4-regular graph. Add sufficiently many self loops on all vertices, to make G_0 a d_0^{16}-regular graph. [For ease of notation, let $D_0 = d_0^{16}$]

- For $i = 0, 1, \cdots, \ell$ (where ℓ is smallest integer such that $\left(1 - \frac{1}{D_0 N^2}\right)^{2^\ell} \leq 7/10$)

  ◇ $G_{i+1} = (G_i Ⓡ H)^8$ (replacement product, followed by powering)

- G_ℓ is now a D_0-regular graph with λ_2 of every component being less than 7/10. Thus, we can run depth-first search on G_ℓ to check connectivity between s^* and t^* where s^* is some vertex in the *cloud* generated by s and similarly for t^* (Note: this uses only logspace!)

Note:
(1) H in the algorithm denotes the $(d_0^{16}, d_0, 1/2)$-expander that we had in Lemma 8
(2) Since the edges in a replacement product or powered graph can be computed on the fly in logspace, we can run this entire algorithm in logspace!

References

[1] Omer Reingold Undirected s-t connectivity in logspace *STOC*, 2005 *J. ACM*, 2008

[2] Sanjeev Arora, Boaz Barak Computational Complexity: A Modern Approach *Cambridge University Press*, 2009