

# Any graph is a bunch of expanders plus a few edges

December 12, 2007

In this note, I want to discuss the rather surprising fact that any graph can be partitioned into connected components that are expanders by removing only a few edges. Precisely:

**Claim 1.** *For any undirected nonbipartite graph  $G = (V, E)$  and for any  $\epsilon \in [0, \frac{1}{2}]$ , there is a polynomial time algorithm that removes  $\epsilon|E|$  edges from  $G$  such that each connected component in the remaining graph has eigenvalue gap of  $\Omega(\epsilon^2/\log^2 |E|)$ .*

We will prove this claim, and then we will look at an example application.

*Proof.* The idea is simple. Either  $G$  already has high expansion in which case we are done; otherwise, there is a small cut and then, we can remove these edges and continue recursively. (This process is called “spectral partitioning.”) We will make this idea more precise by counting the number of edges removed and by bounding the second eigenvalue of the connected components in the remaining graph.

But before we do this, there’s a more basic question we haven’t addressed. In the case when  $G$  is not a good expander, how do we find in polynomial time the small cut that is guaranteed to exist? It turns out that there is a simple algorithm due to Jerrum and Sinclair to do this:

**Lemma 2.** *There is a polynomial-time algorithm that given a nonbipartite graph  $G$  finds a cut  $S$  of conductance<sup>1</sup> at most  $\sqrt{2(1 - \lambda_2(G))}$ .*

The following algorithm produces the desired cut:

On input  $M$ , the transition matrix for  $G$ :

1. Let  $u$  be such that  $Mu = \lambda_2 u$
2. Relabel the vertices of  $G$  such that  $u_i \geq u_{i+1}$ .
3. Output the  $S_i = \{u_1, u_2, \dots, u_i\}$ ,  $1 \leq i \leq n$ , that maximizes  $\frac{e(S_i, \bar{S}_i)}{\sum_{v \in S_i} \deg(v)}$

Jerrum and Sinclair actually proved that the conductance of the graph is at most  $\sqrt{2(1 - \lambda_2(G))}$ , and their proof goes by showing that the cut output by the above algorithm has the requisite conductance. I won’t go through the proof, but it is actually similar to Mihail’s proof of the combinatorial characterization of expansion.

Now, assuming this Lemma as a black-box, we are going to prove the claim. As we said before, we are going to constructively turn the graph into disjoint expanders by continually removing cuts with small conductance. More precisely, at each step of the algorithm, if we can find a cut with conductance at most  $h = O(\frac{\epsilon}{\log |E|})$ , we are going to remove the edges in the cut. So, when the algorithm stops, there will be no

<sup>1</sup>For a graph  $G = (V, E)$ , the conductance of  $G$  is defined to be the quantity:

$$\alpha(G) = \min_{S \subseteq V, \sum_{v \in S} \deg(v) \leq |E|} \frac{e(S, \bar{S})}{\sum_{v \in S} \deg(v)} \quad (1)$$

where  $e(S, T)$  is the number of edges with one endpoint in  $S$  and the other in  $T$ . The conductance of a specific cut  $S$  is defined to be the expression within the min.

cuts of size  $h$ , or from the lemma, the second eigenvalue of each of the remaining connected components will be  $1 - \Omega(\epsilon^2 / \log^2 |E|)$ . Now, let us bound the total number of edges removed in this process.

To do so, we turn to a common proof technique, charging. The idea is that we assign charges to the edges such that running the algorithm redistributes the charges among the edges while never changing the total amount of charge; however, there is some bound on the total charge that can be assigned to each edge, thus leading to an upper bound on the total number of runs the algorithm can go through.

Let us make this idea precise for our proof. Initially, we put the charge 1 on each edge. Whenever the algorithm finds a cut  $(S, C - S)$  in a connected component  $C$  in  $G$  where  $S$  is the side of the cut with the fewer number of edges, we redistribute the total charge in the edges of the cut uniformly among the edges between vertices in  $S$ . Clearly, then the total amount of charge never decreases in the course of running the algorithm. Moreover, the edges in every connected component have the same charge (but edges in different components could have different charges).

Now, we make the observation that the charge of an edge cannot be increased an indefinite number of times. Whenever the charge is increased, the edge must lie on the side of the cut with the smaller number of edges; so, the charge of any edge can be increased at most  $\log_2 |E|$  times.

Finally, we claim that if the charge of an edge has been increased at most  $t$  times, then its current charge is at most  $(1 + 3h)^t$  (where  $h$  is the conductance of the cuts we are removing). The proof of this claim is by induction. Suppose after  $t - 1$  increases, the charge of every edge in a connected component is  $(1 + 3h)^{t-1}$ . Then, if this component is cut again, the charge of an edge in  $S$ , the side of the cut with the fewer number of edges, is going to be:  $(1 + 3h)^{t-1} + \frac{(1+3h)^{t-1}e(S,\bar{S})}{e(S,S)} = (1 + 3h)^{t-1} \left(1 + \frac{e(S,\bar{S})}{\frac{1}{2}(\sum_{v \in S} \deg(v) - e(S,\bar{S}))}\right) \leq (1 + 3h)^{t-1} \left(1 + \frac{2h}{1-h}\right) \leq (1 + 3h)^t$  for  $h \leq \frac{1}{3}$ .

So, the total amount of charge a single edge can accumulate is at most  $(1 + 3h)^{\log_2 |E|} \leq 1 + \epsilon \leq \frac{1}{1-\epsilon}$ . But we know that the total charge of all the edges left must be  $|E|$ , because the total charge hasn't changed. So, the total number of edges left must be at least  $|E|(1 - \epsilon)$ .  $\square$

Before proving this claim, let's look at an example application. In general, the claim is applicable whenever we are trying to approximate some number associated with a graph, and in doing the approximation, it does not hurt to throw away a small fraction of the edges; for the claim to be useful, it should also be the case that it is easier to approximate the number for an expander graph. The application we are going to consider is finding (in polynomial time) an approximate bipartition of a graph when we know that the graph is approximately bipartite.