Sublinear Algorithms January 31, 2012

Lecture 9

Lecturer: Sofya Raskhodnikova *Scribe(s): Sandeep Narayanaswamy, Kashyap Dixit*

1 Max-cut additive approximation scheme

In the previous lectures, we have seen the sublinear algorithm to test if a graph is bipartite or ϵ-far from it. Here we study the algorithm to approximate the size of maximum-cut in dense graphs with additive error. The algorithm here is based on Goldreich, Goldwasser and Ron's paper [GGR98]

Let $G = (V, E)$ be an undirected unweighted graph and (V_1, V_2) be a partition of the set of vertices V.

Definition 1. *The* edge density *of a cut (V_1, V_2), denoted by $\mu(V_1, V_2)$, is the ratio between the number of edges crossing the cut, and the number of entries in the adjacency matrix.*

$$\mu(V_1, V_2) = \frac{|e(V_1, V_2)|}{n^2},$$

where $e(V_1, V_2)$ is the set of edges with one endpoint in V_1 and the other in V_2.

The edge density μ is normalized to lie in $[0, 1]$, but in fact never exceeds $\frac{1}{4}$: the cut with the largest edge density results from a complete bipartite graph with $n/2$ vertices in each partition. It may also result from a bipartition of clique K^n (with n-vertices), each partition containing $n/2$ vertices respectively.

Definition 2. *The edge density of the largest cut in G is denoted by $\mu(G)$:*

$$\mu(G) = \max_{(V_1, V_2)} \mu(V_1, V_2).$$

Given a graph G, the Max-Cut problem is to determine the partition (V_1, V_2) whose edge density equals $\mu(G)$. Here we look at an algorithm that approximates Max-Cut in sublinear time.

1.1 The Approximation Algorithm

We will see the approximation algorithm that has oracle access to the graph and approximates $\mu(G)$ with an additive error ϵ, i.e., outputs an approximate value $\tilde{\mu}$ such that $\tilde{\mu} \geq \mu - \epsilon$ w.p. $\geq \frac{5}{6}$, and runs in time $O(n \cdot (2^{poly(\frac{1}{\epsilon})} + poly(\frac{1}{\epsilon})))$.

Theorem 3. *For every ϵ there is an algorithm that returns a partition V_1, V_2 of V such that $\mu(V_1, V_2) \geq \mu(G) - \epsilon$ with probability at least $5/6$*

1.1.1 Basic Idea

Consider two subsets L_1 and L_2 of V, and a vertex v that is not in either of L_1 or L_2. If we wish to decide whether to place v in L_1 or L_2 with a view to achieving a large edge density, the greedy way to do this would be to place v in the set where it has fewer neighbors. To make this formal, we introduce some notation: we denote by $N(v, U)$ the set of neighbors a vertex v has in set U and let $\Gamma(v, U) = |N(v, U)|$. Now the greedy idea is: if $\Gamma(v, L_1) > \Gamma(v, L_2)$, we move v to L_2.

1.1.2 A preliminary algorithm that approximates Max-Cut

1. Partition V into $\ell = \frac{4}{\epsilon}$ sets of (almost) equal size, $V^1, V^2, \ldots, V^\ell$. (To make things simpler, we assume these sets are of equal size.) We will consider each V^i separately.

2. For each i in $\{1, 2, \ldots, \ell\}$, select a set L^i of size $\frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon}$ uniformly at random from $V \backslash V_i$, the set of vertices not in V_i. We denote the ℓ-tuple of these sets by L: $L = (L^1, L^2, ..., L^\ell)$.

3. Note that there are $2^{|L^j|} = 2^{\frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon}}$ possible partitions of the set L^j into (L_1^j, L_2^j). Also, there are $O(2^{\ell |L^i|}) = O(2^{\frac{4}{\epsilon^3} \cdot \log \frac{1}{\epsilon}})$ partition sequences $\pi(L) = ((L_1^1, L_2^1), \ldots (L_1^\ell, L_2^\ell))$. For each partition sequence, partition the whole vertex set V greedily, as follows:

   - For i in $\{1, 2, \ldots, \ell\}$, partition V^i into (V_1^i, V_2^i) using the greedy rule: For every vertex $v \in V^i$, if $\Gamma(v, L_1^i) > \Gamma(v, L_2^i)$, put v in V_2^i, and otherwise in V_1^i. See figure 1.
   - Define $V_1 = \bigcup_i V_1^i$ and $V_2 = \bigcup_i V_2^i$. (Thus every partition of L induces a partition of V.) Calculate the edge density for this cut, $\mu(V_1, V_2)$.

4. Output the cut with the best density.

Note that this algorithm samples the sets $L^1, L^2, \ldots$ only once, but considers all possible partitions of L. Also, for each partition sequence, note that each vertex $v \in V_i$ makes at most $|L_i|$ querries to make a greedy choice. Thus, total number of querries is bounded by $\sum_i |L_i||V_i| = |L_1| \sum_i^k |V_i| = O((\frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon})n)$ (since V_i are disjoint). Since L^is are fixed for every iteration, we may store these querries in our local memory and to use it in further iterations. Thus the total number of querries is bounded by $O((\frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon})n)$ and time complexity is bounded by $O(2^{\ell |L_1|} \frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon} \cdot n) = O((2^{\frac{4}{\epsilon^3} \cdot \log \frac{1}{\epsilon}} \frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon})n)$. We prove the correctness below.

1.2 Correctness

To study the correctness, we prove the following lemma.

Lemma 4. *Let (H_1, H_2) be any cut of V. Then with probability $\geq 5/6$ over the choice of L, some partition sequence $\pi(L)$ is **good**, i.e., it induces a partition (V_1^π, V_2^π) of V such that $\mu(V_1^\pi, V_2^\pi) \geq \mu(H_1, H_2) - \frac{3\epsilon}{4}$.*

Note that the above lemma holds for every cut (H_1, H_2), so in particular, it holds for the max-cut as well.

Proof. The proof goes via a hybrid argument. We define the i^{th} hybrid partition (H_1^i, H_2^i) as follows. Let $(H_1^0, H_2^0) = (H_1, H_2)$ and (W_1^0, W_2^0) be the partition of $W^0 = V \backslash V^1$ induced by (H_1, H_2). Similarly, we define W_1^{i-1} be the partition of $V \backslash V^i$ such that the sets $V^0, V^1, .., V^{i-1}$ are partitioned according to our algorithm and $V^{i+1}, V^{i+2}, ..., V^\ell$ are partitioned according to (H_1, H_2). The partition (H_1^i, H_2^i) is depicted in figure 2. We define $H_1^i = W_1^{i-1} \cup V_1^i$ and $H_2^i = W_2^{i-1} \cup V_2^i$.

Simply stated, in partition (H_1^i, H_2^i), the nodes in $V^1, ..., V^i$ are partitioned according to our algorithm and nodes in $V^{i+1}, ..., V^\ell$ are partitioned according to (H_1, H_2). We will argue that density drop between successive hybrid partitions is small. More precisely, we want to argue that with probability at least $\frac{5}{6}$, $\mu(H_1^i, H_2^i) \geq \mu(H_1^{i-1}, H_2^{i-1}) - \frac{3\epsilon}{4\ell}$ *for every* i. Note that the proof of above statement will imply the proof of lemma, which may be obtained by summing up all the inequalities.

Let A_i be the event that $\mu(H_1^i, H_2^i) \geq \mu(H_1^{i-1}, H_2^{i-1}) - \frac{3\epsilon}{4\ell}$ holds for the stage i. Since we want to prove that A_i should happen for every i w.p. $\geq 5/6$, we want that $\Pr[\bigcap_i A_i] \geq 5/6$, or $\Pr[\overline{\bigcap_i A_i}] = \Pr[\bigcup_i \bar{A}_i] \leq 1/6$. We prove this in the following claim.

Claim 5. $\Pr[\overline{\bigcap_i A_i}] = \Pr[\bigcup_i \bar{A}_i] \leq 1/6$

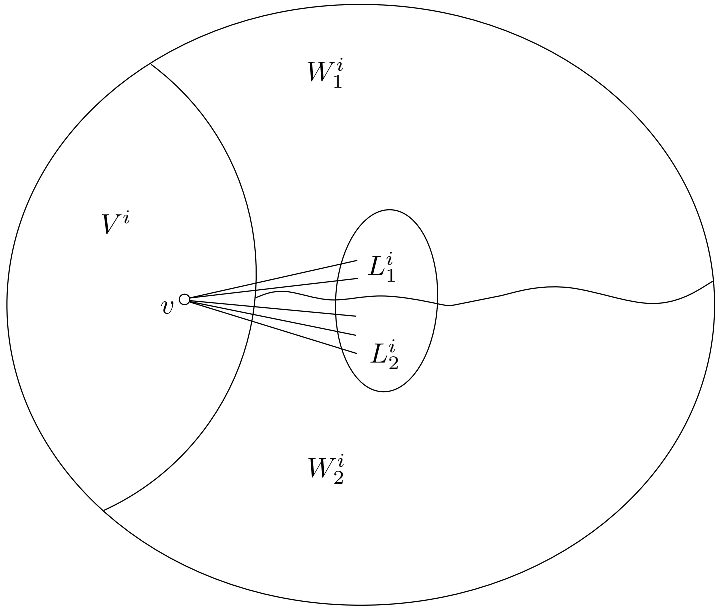

Figure 1: Greedy strategy will place v in L_1^i. Note that $W^i = V \backslash V^i$

.

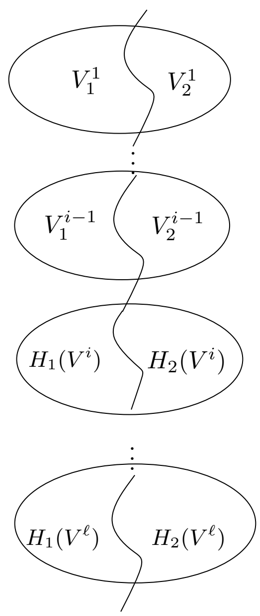

Figure 2: Partition (H_1^i, H_2^i): The vertex set V^i is being re-partitioned according to the algorithm

Proof. Note that the claim holds (by union bound) if we could prove that $\Pr[A_i] \geq 1 - \frac{1}{6\ell}$ for each i. We prove this by the following accounting argument. We say that a vertex is *bad* w.r.t. (L_1^i, L_2^i) if $\left|\frac{\Gamma(v,L_j^i)}{|L|} - \frac{\Gamma(v,W_j^{i-1})}{n}\right| > \frac{\epsilon}{32}$. Otherwise, the vertex is a *good* vertex. A sample L^i is *good* with respect to (W_1^{i-1}, W_2^{i-1}) if at most $\frac{\epsilon}{8}$ fraction of vertices in V^i are bad. Assume in fact that for each i, the set L^i is good for (W_1^{i-1}, W_2^{i-1}). We will later see in claim 7 that this happens with high probablity. We call a vertex *unbalanced* with respect to (W_1^{i-1}, W_2^{i-1}) if $\frac{\Gamma(v,W_j^{i-1})}{n} - \frac{\Gamma(v,W_{j'}^{i-1})}{n} \geq \frac{\epsilon}{8}$ for $j \neq j'$ and $j, j' \in \{1,2\}$. Note that if a vertex $v \in V^i$ is unbalanced with respect to (W_1^{i-1}, W_2^{i-1}) such that $\left|\frac{\Gamma(v,L_1^i)}{|L|} - \frac{\Gamma(v,W_2^{i-1})}{n}\right| \leq \frac{\epsilon}{32}$ then

$$\begin{aligned}
\frac{\epsilon}{8} &\leq \frac{\Gamma(v, W_j^{i-1})}{n} - \frac{\Gamma(v, W_{j'}^{i-1})}{n} \\
&\leq \frac{\Gamma(v, L_j^i)}{|L|} + \frac{\epsilon}{32} - \left(\frac{\Gamma(v, L_{j'}^i)}{|L|} - \frac{\epsilon}{32}\right) \\
&= \frac{\Gamma(v, L_j^i)}{|L|} - \frac{\Gamma(v, L_{j'}^i)}{|L|} + \frac{\epsilon}{16}
\end{aligned}$$

Thus we have that $\Gamma(v, L_j^i)) \geq \Gamma(v, L_{j'}^i)) + \frac{1}{16}\epsilon|L|$, which means that an unbalanced vertex always has one partition L_j^i with surplus neighbors as against the other partition $L_{j'}^i$. From our assumption, we also have that $\Gamma(v, W_j^{i-1})) \geq \Gamma(v, W_{j'}^{i-1})) + \frac{\epsilon n}{16}$ (noting that the order of j and j' in the inequalities is same). Thus, we have the following useful observation.

Observation 6. *We are guaranteed that when the partition (L_1^i, L_2^i) is used, v is put opposite to the majority of neighbors in W^{i-1}*

Note that in each step i, only the partitions of V^i are changed. Thus, the potential reduction of cut-edges may be caused only between the partitions V^i and W^{i-1}.

Now we will upper bound the number of cut-edges that we loose by re-partitioning the vertices of V^i, i.e. going from hybrid partition (H_1^{i-1}, H_2^{i-1}) to (H_1^i, H_2^i).

1. The number of cut-edges due to *unbalanced* vertices that are *good* remains unchanged. This follows from observation 6.
2. The number of cut-edges due to *unbalanced* vertices that are *bad* decreases by at most $\frac{\epsilon}{8}|V_i|n \leq \frac{\epsilon}{t\ell}n^2$
3. The number of cut-edges due to *balanced* edges decreases by at most $\frac{\epsilon n}{8}|V_i| \leq \frac{\epsilon}{4\ell}n^2$
4. Edges lost inside V^i are at most $|V^i|^2 \leq \frac{n^2}{\ell^2} \leq \frac{\epsilon}{4\ell}n^2$

Hence, the totla number of cut-edges lost following the re-partitioning of V^i is $\frac{3\epsilon}{4}n^2$ ☐

Hence, the loss in density at each step is at most $\frac{3\epsilon}{4}$

☐

It remains to show that each set L^i is good with high probability.

Claim 7. *Probability that some set L^i is bad is at most $\frac{1}{6}$*

Proof. Let $L^i = \{u_1, u_2, ..., u_{|L|}\}$. First fix a vertex $v \in V^i$. Recall that L^i is chosen uniformly at random from $W^{i-1} = V \backslash V^i$. Define the indicator random variable

$$X_j^k = \begin{cases} 1 & \text{if } u_k \in N(v, W_j^{i-1}) \\ 0 & \text{otherwise} \end{cases}$$

.

Note that $X_j = \sum_k X_j^k$ is the number of neighbors of v in L_j^i and the probability that $X_j^k = 1$ is $\frac{1}{n}\Gamma(v, W_j^{i-1})$ $(j \in \{1,2\})$. Thus, by additive Chernoff bound, we have that

$\Pr\left[\left|\frac{\Gamma(v,L_j^i)}{|L|} - \frac{\Gamma(v,W_j^{i-1})}{n}\right| > \frac{\epsilon}{32}\right] = exp(-\Omega(\epsilon^2|L|)) \leq \epsilon/96\ell$. By Markov's inequality, the probability of finding more than $\frac{\epsilon}{8}$ bad vertices is upper bounded by $\frac{1}{12\ell}$ for a particular j. Thus L_1^i is bad with probability at most $\frac{1}{6\ell}$ (by union bound over all $j \in \{1,2\}$). Thus, using an argument similar to claim 5, we prove this claim. □

1.3 Improved Algorithm for approximating Max-cut-density

1. Partition V into $\ell = \frac{4}{\epsilon}$ sets of (almost) equal size, $V^1, V^2, \ldots, V^\ell$. (To make things simpler, we assume these sets are of equal size.) We will consider each V^i separately.

2. For each i in $\{1, 2, \ldots, \ell\}$, select a set L^i of size $t = \frac{1}{\epsilon^2} \cdot \log \frac{1}{\epsilon}$ uniformly at random from $V \backslash V_i$, the set of vertices not in V_i. We denote the ℓ-tuple of these sets by L: $L = (L^1, L^2, ..., L^\ell)$.

3. Select uniformly $S = \{s_1, ..., s_m\}$ of size $m = \Theta(\frac{\ell t}{\epsilon^2})$

   - For each partition sequence $\pi(L) = ((L_1^1, L_2^1), \ldots (L_1^\ell, L_2^\ell))$ partition S^i into two disjoint sets S_1^i and S_2^i and let $S_j^{\pi(L)} = \bigcup_{i=1}^{\ell} S_j^i$ (for $j = 1, 2$)

   - For each partition $(S_1^{(\pi(L))}, S_2^{(\pi(L))})$ compute the fraction of the cut-edges between pairs of vertices (s_{2k-1}, s_{2k}). More precisely, define $\hat{\mu}(s_1^\pi, \mu(s_2^\pi)) = \frac{(s_{2k-1}, s_{2k}) \in e(S_1^\pi, S_2^\pi)}{m/2}$

4. Output $\hat{\mu}_{max} = max_\pi \hat{\mu}(S_1^\pi, S_2^\pi)$.

Claim 8. *For any fixed U, with probability at least $5/6$ over the choice of S, $\hat{\mu}_{\max} = \mu(V_1^{\tilde{\pi}}, V_2^{\tilde{\pi}}) \pm \frac{\epsilon}{4}$, where $\hat{\mu}_{\max}$ is as defined in step 4 of the algorithm, and $(V_1^{\tilde{\pi}}, V_2^{\tilde{\pi}})$ is as in Corollary.*

Proof. For every $s \in S$ and $j \in \{1, 2\}$, $s \in S_j^\pi$ if and only if $s \in V_j^\pi$ (since we apply the same decision rule as in the partitioning algorithm). Thus, for each sequence of partitions π of U, we are effectively sampling from (V_1^π, V_2^π), and approximating the density of this cut. For $1 \leq k \leq \frac{m}{2}$, let X_k be the indicator variable for the event that $(s_{2k-1}, s_{2k}) \in e(S_1^\pi, S_2^\pi)$. Then $\Pr[X_k = 1] = \mu(V_1^\pi, V_2^\pi)$, and by the definition in step 3 of the algorithm $\hat{\mu}(S_1^\pi, S_2^\pi) = \frac{2}{m} \sum_{k=1}^{m/2} X_k$. Using an additive Chernoff bound and the choice of m,

$$\Pr\left[|\hat{\mu}(S_1^\pi, S_2^\pi) - \mu(V_1^\pi, V_2^\pi)| \geq \frac{\epsilon}{4}\right] = e^{-\Omega(m\epsilon^2)} = O(2^{-lt}) \ .$$

By the Union bound, since there are 2^{lt} partition sequences π of U, the claim holds. □

References

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998. Preliminary version in 37th FOCS, 1996.