LECTURE 12

Last time

- Graph streaming
- Linear sketching for graph connectivity
- L_0 sampling

Today

- Graph property testing (for dense graphs)
- Testing bipartiteness
- Approximate Max-Cut
  [Goldreich Goldwasser Ron 98]

Testing Properties of Dense Graphs

Adjacency matrix model [Goldreich Goldwasser Ron 98]

- Input: a graph G represented by $n \times n$ adjacency matrix A

$$dist(G, G') = \frac{\text{number of entries on which } A \text{ and } A' \text{differ}}{n(n-1)}$$

Equivalently, for undirected graphs

$$dist(G, G') = \frac{\text{number of edges present in exactly one of } G \text{ and } G'}{n(n-1)/2}$$

- Goal: accept (w.h.p.) if G has property $\boldsymbol{\mathcal{P}}$;

  reject (w.h.p.) if G is ε-far from $\boldsymbol{\mathcal{P}}$

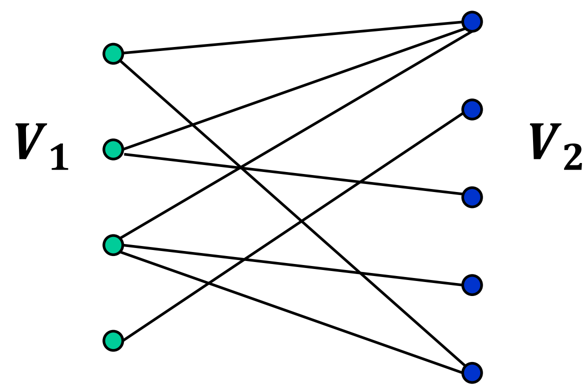
  (that is, at least ε fraction of entries in A

  must be changed to get a graph satisfying $\boldsymbol{\mathcal{P}}$)

Bipartite Graphs and Partitions

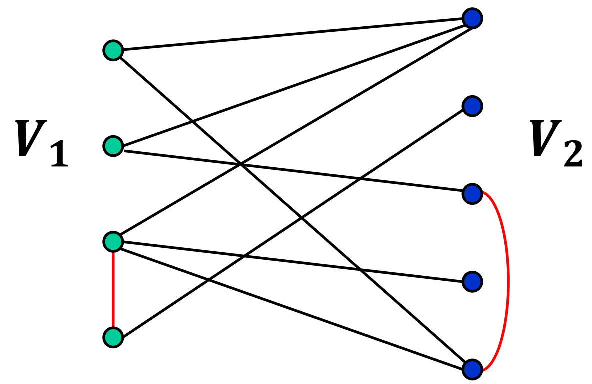
- A pair (V_1, V_2) of sets is a partition of V if
  - V_1 and V_2 are disjoint subsets of V and
  - $V_1 \cup V_2 = V$

- A graph $G = (V, E)$ is bipartite
  if there exists a partition (V_1, V_2) of V such that
  every edge in E has one endpoint in V_1 and the other in V_2



Bipartite Graphs and Partitions

- An edge $\{u, v\}$ is violating w.r.t. a partition (V_1, V_2) if either $u, v \in V_1$ or $u, v \in V_2$



> ***Observation***
>
> If an n-node graph $G = (V, E)$ is ε-far from bipartite then,
> for every partition (V_1, V_2),
> there exist at least $\varepsilon n(n-1)/2$ violating edges w.r.t. (V_1, V_2).

Testing Bipartiteness

- We can check if a graph is bipartite (exactly) in linear time (in the size of the graph) by a BFS
- Today: a bipartiteness tester from [GGR98] that runs in time $\tilde{O}\left(\frac{1}{\varepsilon^4}\right)$
- The best tester for bipartiteness in [GGR98] runs in time $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$
- There is a nonadaptive $\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$-time tester [Alon Krivelelvich 02]
- $\Omega\left(\frac{1}{\varepsilon^2}\right)$ queries for nonadaptive testers
  $\Omega\left(\frac{1}{\varepsilon^{1.5}}\right)$ queries for adaptive testers [Bogdanov Trevisan 04]

First Attempt

- Consider an algorithm of the following form

Bipartiteness Tester

1. Sample t pairs of nodes uniformly and independently.
2. **Reject** iff they rule out all possible partitions of V.

- How large should t be?

If G is bipartite, it is always accepted

- Suppose G is ε-far from bipartite.
- We would like to rule out all 2^n possible partitions of V
- Fix a partition (V_1, V_2) of V,

$$\Pr_{u,v \in [n], u \neq v}[\{u, v\} \text{ is violating w.r.t. } (V_1, V_2)] \geq \varepsilon$$

By Observation

- $BAD(V_1)$ = event that all t pairs are non-violating w.r.t. (V_1, V_2)

$1+x \leq e^x$

$$\Pr[BAD(V_1)] \leq (1 - \varepsilon)^t \leq e^{-\varepsilon t} \leq 1/3 \cdot 2^{-n}$$

if $t \geq \frac{n \ln 2 + \ln 3}{\varepsilon}$

- BAD = event that $\exists (V_1, V_2)$ s.t. all t pairs are non-violating w.r.t. (V_1, V_2)

$$\Pr[BAD] \leq \sum_{V_1 \subseteq V} \Pr[BAD(V_1)] \leq 2^n \cdot \frac{1}{3} \cdot 2^{-n} = \frac{1}{3}$$

By a union bound

If we wanted to rule out all partitions for a graph on ℓ nodes, would need $t = \Theta(\ell/\varepsilon)$

The $\widetilde{O}(1/\varepsilon^4)$-Time Bipartiteness Tester [GGR]

Bipartiteness Tester (Input: ε, n and query access to adjacency matrix of G)

1. Pick a set of S nodes uniformly and independently, $|S| = \Theta\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\right)$
2. Query all pairs (u, v), where $u, v \in S$
3. If the queried subgraph G' is bipartite, **accept**; otherwise, **reject**.

Query complexity and running time:

If G is bipartite, it is always accepted

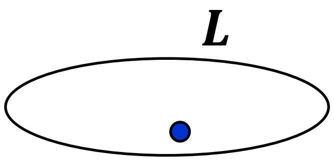
- We can check whether G' is bipartite with a BFS.
- Query and time complexity: $O\left(\binom{|S|}{2}\right) = O\left(\frac{1}{\varepsilon^4}\log^2\frac{1}{\varepsilon}\right)$

Correctness: Main Idea

- Assume G is ε-far from bipartite

Main idea behind the analysis:

- Break the samples S into two sets:

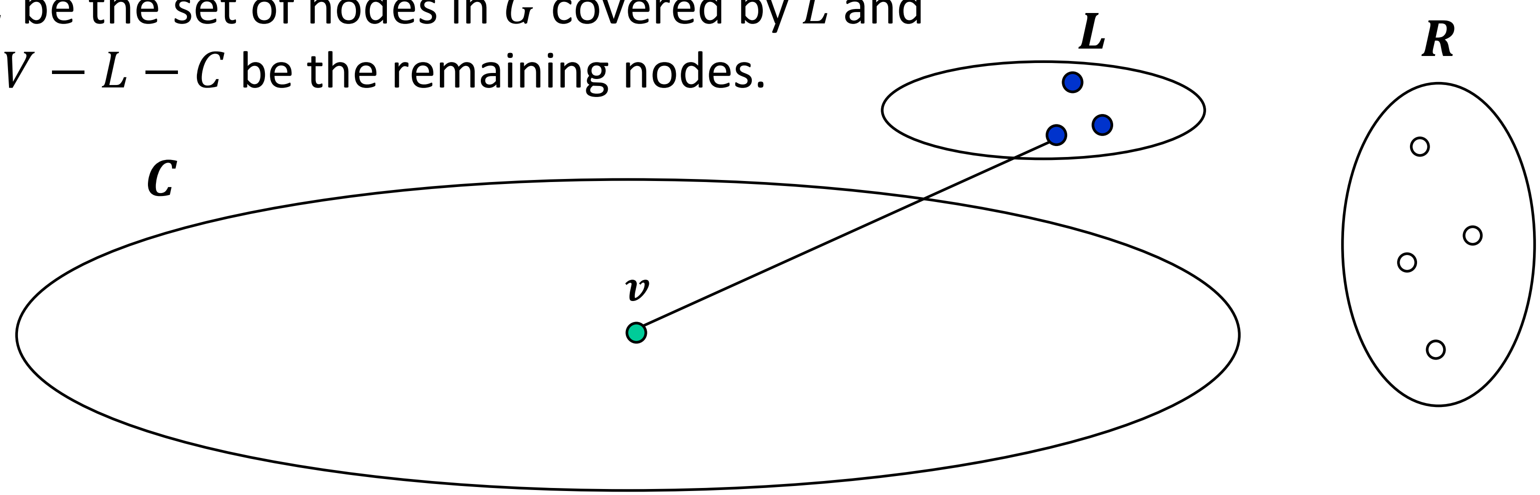


1. Learning set L of size $\ell = \Theta\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)$
2. Testing set T of size $t = \Theta\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\right)$

- Every partition of the learning set L induces a partition of (most of) V
- We use T to check for violating pairs w.r.t. such partitions

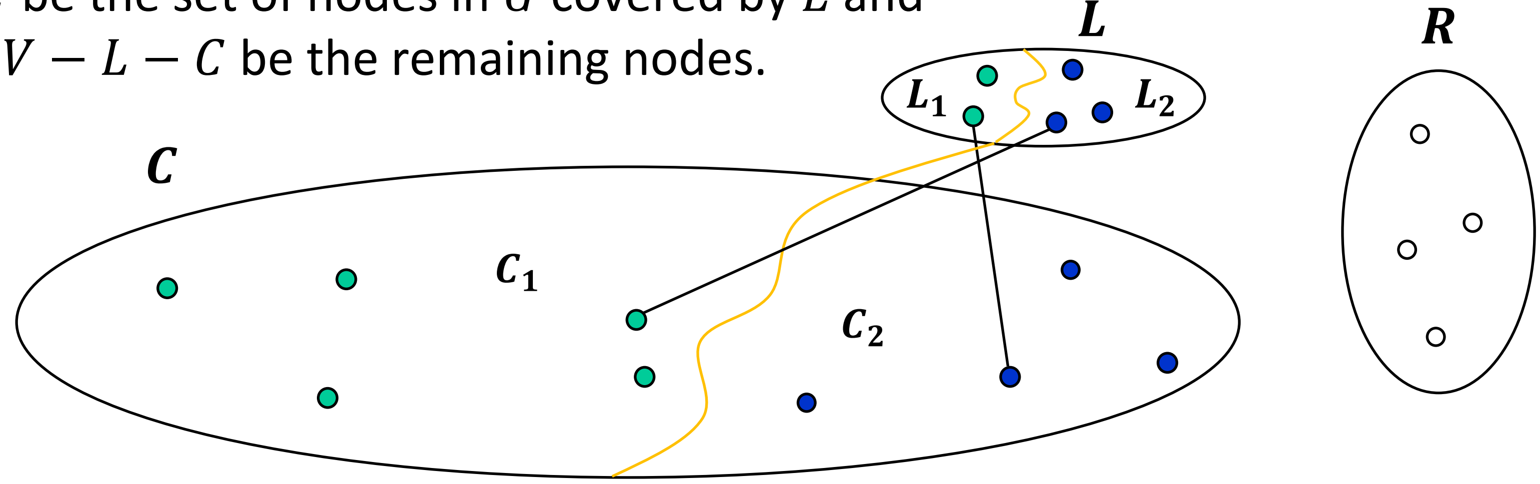
Correctness: Partitions of L and V

- A node v is covered by a set L if v has a neighbor in L.
- Let C be the set of nodes in G covered by L and $R = V - L - C$ be the remaining nodes.



Correctness: Partitions of L and V

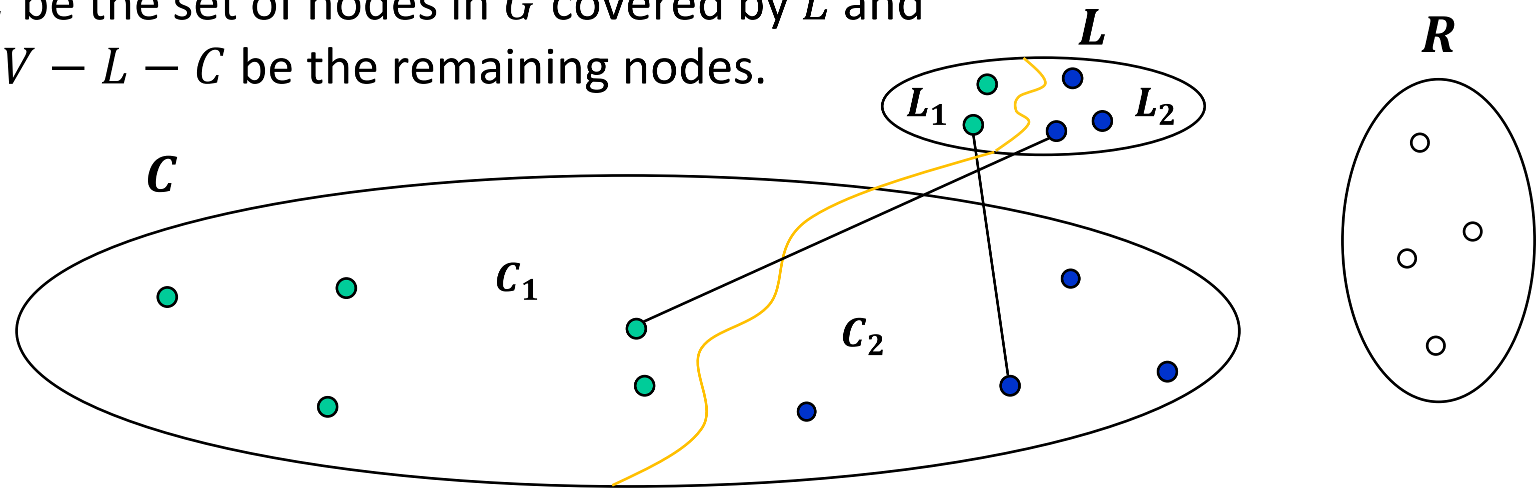
- A node v is covered by a set L if v has a neighbor in L.
- Let C be the set of nodes in G covered by L and $R = V - L - C$ be the remaining nodes.



A partition of L induces a partition of C

Correctness: Influential Nodes

- A node v is covered by a set L if v has a neighbor in L.
- Let C be the set of nodes in G covered by L and $R = V - L - C$ be the remaining nodes.



A partition of L induces a partition of C

- A node is influential if its degree is at least $\frac{\varepsilon n}{8}$.

Most of the edges in the graph are between influential nodes.
We don't want to miss them.

Correctness: Analysis of the Learning Set L

Lemma 1

Let BAD_L be the event that $\geq \frac{\varepsilon n}{8}$ influential nodes are in R (i.e., not covered by L).

$$\Pr[BAD_L] \leq 1/6$$

Proof: For each influential node v, define the indicator random variable

$$X_v = \begin{cases} 1 & \text{if } v \text{ is not covered by } L \\ 0 & \text{otherwise} \end{cases}$$

v has degree $\geq \frac{\varepsilon n}{8}$

$|L| = \Theta\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)$

$$\Pr[X_v = 1] \leq \left(1 - \frac{\varepsilon}{8}\right)^{|L|} \leq e^{-\frac{\varepsilon |L|}{8}} \leq \frac{\varepsilon}{48}$$

- Let $X = \sum_v X_v$. Then $\Pr[BAD_L] = \Pr\left[X \geq \frac{\varepsilon n}{8}\right]$

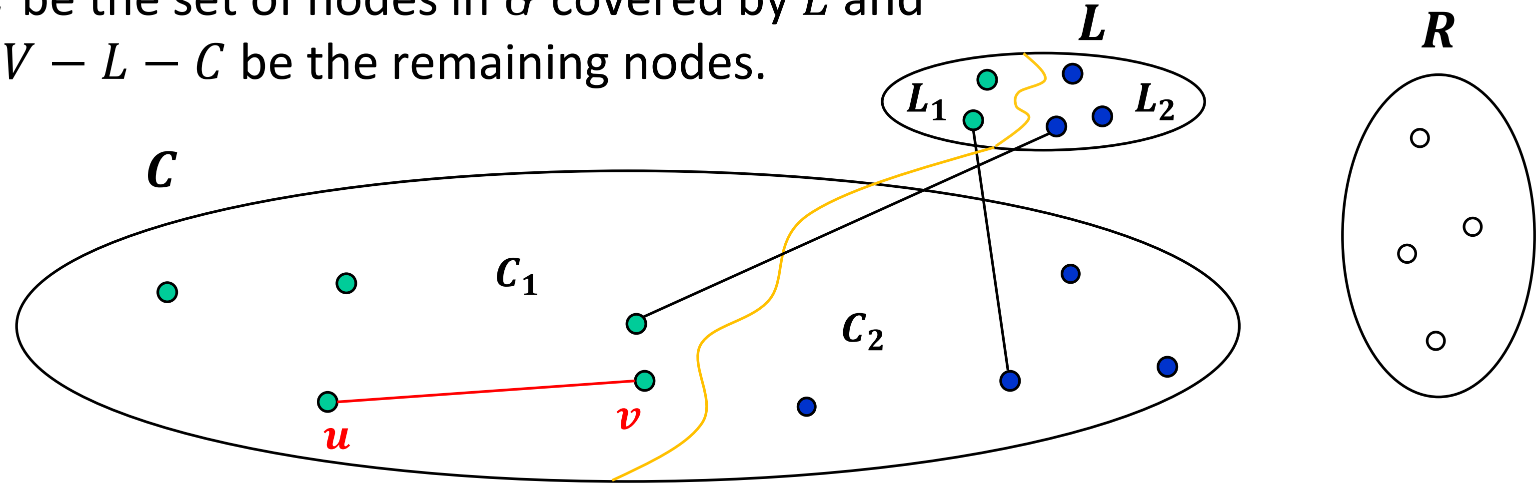
$$\mathbb{E}[X] = \sum_v \mathbb{E}[X_v] \leq \frac{\varepsilon n}{48}$$

$$\Pr\left[X \geq \frac{\varepsilon n}{8}\right] \leq \frac{\mathbb{E}[X]}{\varepsilon n/8} \leq \frac{1}{6}$$

By Markov's inequality

Correctness: Witness w.r.t. (L_1, L_2)

- A node v is covered by a set L if v has a neighbor in L.
- Let C be the set of nodes in G covered by L and $R = V - L - C$ be the remaining nodes.



A partition of L induces a partition of C

- An edge (u, v) is a witness w.r.t. a partition (L_1, L_2) if $u, v \in C_1$ or $u, v \in C_2$

Correctness: Analysis of the Learning Set L

Lemma 2

If BAD_L does not occur then for every partition (L_1, L_2) of L, then at least $\frac{\varepsilon}{4}$ fraction of node pairs are witnesses w.r.t. (L_1, L_2).

Proof: Consider any partition (V_1, V_2) of V s.t. $V_1 \cap L = L_1$ and $V_2 \cap L = L_2$

- By Observation, $\geq \frac{\varepsilon n(n-1)}{2}$ violating edges w.r.t. (V_1, V_2)

Violated edges incident to	Number of nodes	Degree	Number of violating edges
Influential nodes in ***R***			
Non-influential nodes in ***R***			
Nodes in ***L***			

- Then: $\geq \frac{\varepsilon n(n-1)}{2} - \frac{\varepsilon n(n-1)}{8} - \frac{\varepsilon n(n-1)}{8} - \frac{\varepsilon n(n-1)}{8} \geq \frac{\varepsilon n(n-1)}{8}$ violating edges between nodes in C
- Each such edge is a witness w.r.t. (L_1, L_2)

Correctness: Analysis of the Training Set T

View samples from T as pairs $(v_1, v_2), (v_3, v_4), \dots, (v_{|T|-1}, v_{|T|})$

Lemma 3

Let BAD_T = event that there is a partion of L *such that no pair* (v_{2i-1}, v_{2i}) *is a witness w.r.t. that partition.*

$$\Pr[BAD_T | \overline{BAD_L}] \leq 1/6$$

Proof: Fix a partition (L_1, L_2) of L, which defines a partition of C.

- The probability that no pair (v_{2i-1}, v_{2i}) *is a witness* w.r.t. (L_1, L_2) is

$\geq \frac{\varepsilon n(n-1)}{8}$ pairs out of $\frac{n(n-1)}{2}$ are witnesses (by Lemma 2)

$$\leq \left(1 - \frac{\varepsilon}{4}\right)^{|T|/2} \leq e^{-\frac{\varepsilon |T|}{8}} \leq \frac{2^{-|L|}}{6}$$

Since $|T| = \Theta\left(\frac{1}{\varepsilon}|L|\right)$

- Since there are $2^{|L|}$ partitions of L,

$$\Pr[BAD_T | \overline{BAD_L}] \leq 2^{|L|} \cdot \frac{2^{-|L|}}{6} = \frac{1}{6}$$

By a union bound

Correctness: Putting It All Together

- Recall that G is ε-far

$$\Pr[G \text{ is accepted}]$$

$$\leq \Pr[BAD_L] + \Pr\left[\, BAD_T \mid \overline{BAD_L}\, \right] \cdot \Pr\left[\overline{BAD_L}\right]$$

By product rule

$$\leq \frac{1}{6} + \frac{1}{6} \cdot 1$$

By Lemmas 1 and 3

$$\leq \frac{1}{3}$$

- We got: run time $\tilde{O}\left(\frac{1}{\varepsilon^4}\right)$
- Exercise: improve to $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$

Bipartiteness in the Streaming Model

A bipartite double-cover of $G = (V, E)$ is a graph $G' = (V', E')$, where for each node $v \in V$, we add two nodes, v_1 and v_2, to V';

For each edge $(u, v) \in E$, we add two edges, (v_1, u_2) and (v_2, u_1), to E'.

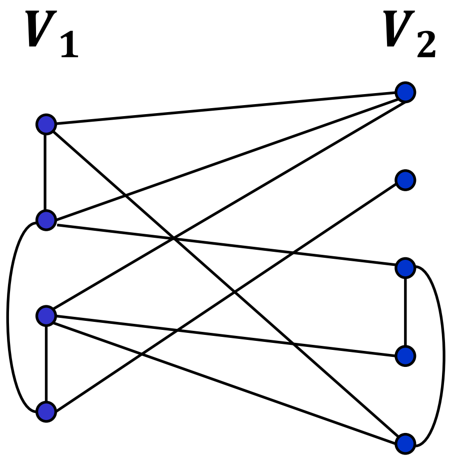
Lemma

G is bipartiate iff the number of connected components in G' is twice the number of connected components in G

We can solve bipartiteness exactly (w.h.p.) in the semi-streaming model.

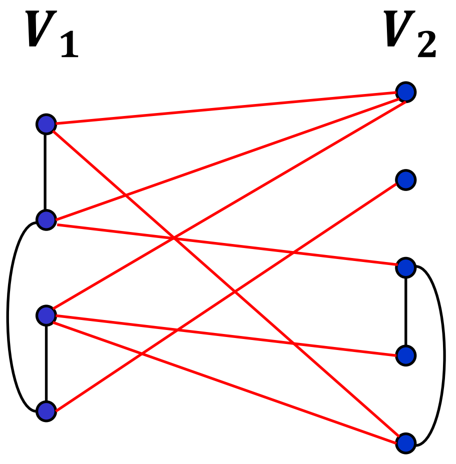
Max Cut in Dense Graphs

- Let $G = (V, E)$ be an undirected n-node graph.
- Let (V_1, V_2) be a partition of V.
  $e(V_1, V_2)$ = set of edges crossing the cut



Max Cut in Dense Graphs

- Let $G = (V, E)$ be an undirected n-node graph.



- Let (V_1, V_2) be a partition of V.
  $e(V_1, V_2)$ = set of edges crossing the cut

- The edge density of the cut,
  denoted $\mu(V_1, V_2)$, is $\frac{|e(V_1,V_2)|}{n^2}$.

- The edge density of the largest cut in G is

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

Approximate Max-Cut Problem

[Goldreich Goldwasser Ron 98]

Input: parameter ε, access to an undirected graph $G = (V, E)$ represented by $n \times n$ adjacency matrix.

Goal 1: Output an estimate $\hat{\mu}$ such that:

$$\Pr[|\hat{\mu} - \mu(G)| \leq \varepsilon] \geq 2/3$$

- [GGR98]: $\text{poly}\left(\frac{1}{\varepsilon}\right)$ queries and $O(2^{poly\left(\frac{1}{\varepsilon}\right)})$ time

Goal 2: Output a partition (V_1, V_2) with edge density

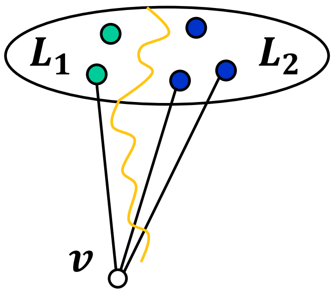
$$\mu(V_1, V_2) \geq \mu(G) - \varepsilon$$

with probability at least 2/3.

- [GGR98]: $O\left(2^{poly\left(\frac{1}{\varepsilon}\right)} + n \cdot poly\left(\frac{1}{\varepsilon}\right)\right)$ time

Greedy Partitioning

- Suppose we have a partition (L_1, L_2) of $L \subset V$.
- In which part should we place a new node v to maximize edge density?



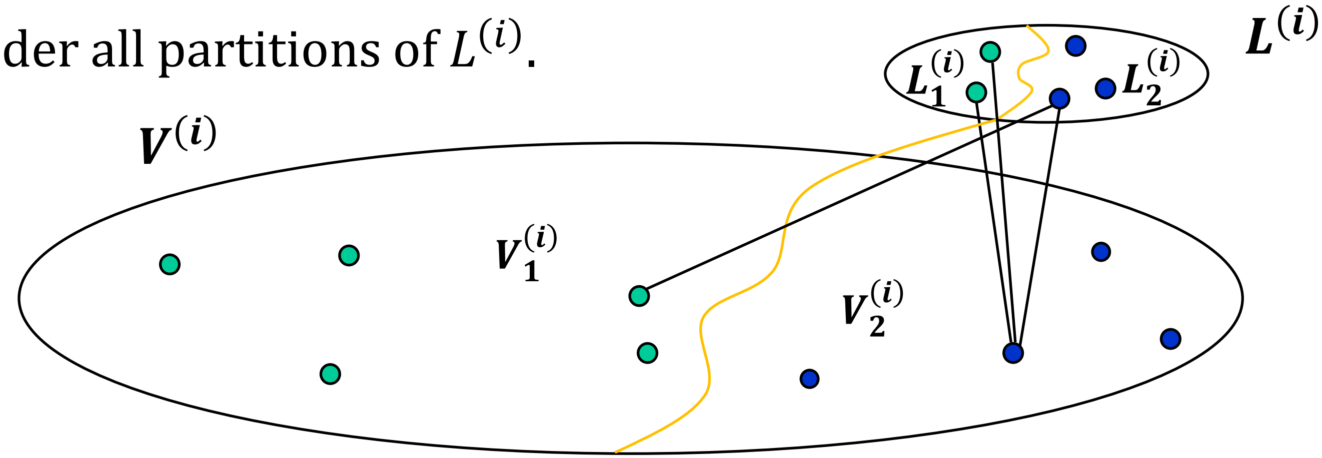
- Let $\Gamma(v, U)$ be the number of neighbors of v in U.
- Greedy: If $\Gamma(v, L_1) \leq \Gamma(v, L_2)$, place v in L_1; otherwise, place it in L_2.

Main Idea

- Partition V into sets $V^{(i)}$ of (almost) equal size.

Assume they are of equal size.

- For each set $V^{(i)}$, sample a learning set $L^{(i)}$ from the vertices not in $V^{(i)}$.
- Consider all partitions of $L^{(i)}$.



A partition of $L^{(i)}$ induces a partition of $V^{(i)}$ via the greedy rule

A partition sequence $\pi(L) = \left(\left(L_1^{(1)}, L_2^{(1)}\right), \ldots, \left(L_1^{(t)}, L_2^{(t)}\right)\right)$ induces a partition of V

- Consider all such partitions of V and pick the best.

Preliminary Max-Cut Approximation Algorithm

Algorithm (**Input:** ε, n; query access to adjacency matrix of $G=(V,E)$)

1. Partition V into $t = 4/\varepsilon$ sets $V^{(1)}, V^{(2)}, \dots, V^{(t)}$ of (almost) equal size.
2. For each $i \in [t]$, select a set $L^{(i)}$ of size $\ell = \frac{1}{\varepsilon^2} \cdot \log \frac{1}{\varepsilon}$ u.i.r. from $V \backslash V^{(i)}$. Let $L = \left(L^{(1)}, L^{(2)}, \dots, L^{(t)}\right)$.
3. For each partition sequence $\pi(L) = \left(\left(L_1^{(1)}, L_2^{(1)}\right), \dots, \left(L_1^{(t)}, L_2^{(t)}\right)\right)$
4. For each $i \in [t]$
5. Partition $V^{(i)}$ into $\left(V_1^{(i)}, V_2^{(i)}\right)$ using the greedy rule: place v in $V_1^{(i)}$ iff $\Gamma\left(v, L_1^{(i)}\right) \leq \Gamma\left(v, L_2^{(i)}\right)$.
6. Let $V_1^{\pi} = \bigcup_i V_1^{(i)}$ and $V_2^{\pi} = \bigcup_i V_2^{(i)}$; calculate $\mu(V_1^{\pi}, V_2^{\pi})$.
7. Output the cut (V_1^{π}, V_2^{π}) with the largest density.

- Number of partition sequences: $\left(2^{\ell}\right)^t = 2^{poly\left(\frac{1}{\varepsilon}\right)}$
- Running time: $n^2 \cdot 2^{poly\left(\frac{1}{\varepsilon}\right)}$

$O(n^2)$ time for calculating each density