Topics: Testing connectivity, estimating number of connected components, estimating weight of minimal-weight-spanning-tree (MST)

0.1 Testing Connectivity

Given a graph $G$, we want to distinguish if $G$ is connected or $\epsilon$--far from being connected. We will assume that we have edge query access, that is, given nodes $u, v$ we can test if $uv \in E$. We want to test if $G$ is connected by using a constant number of edge queries. However if $G$ is dense, then $G$ being $\epsilon$--far from being connected means we need to add at least $\epsilon n^2$ edges in order to connect the graph. However, we only need to add at most $n - 1$ edges to connect any graph on $n$ vertices. When the number of edges is $O(n^2)$, no dense graph is $\epsilon$--far from being connected. Thus in order to avoid this triviality, we will impose an constraint on the number of edges in the graph. In particular, graphs with $O(dn)$ edges where $d$ is a constant. To be clear, we are assuming that $|V| = n$ and that $|E| \in O(dn)$.

**Theorem 1** There exists a one-sided tester algorithm for $P_n = \{G : G \text{ is connected} \}$ with locality $O(\frac{1}{\epsilon d^2})$, where $\epsilon$ is independent of $n$.

Before proving this, we will need a couple of lemmas or observations.

**Lemma 2** If $G$ is $\epsilon$--far from $P_n$, then $G$ has at least $\epsilon dn + 1$ connected components.

**Proof** Suppose not. Then $G$ has at most $\epsilon dn$ connected components. Thus we need to add at most $\epsilon dn - 1$ edges to connect those components. However, $G$ is $\epsilon$--far from $P_n$ which means we need to add at least $\epsilon dn$ edges to connect the graph. This is a contradiction and hence the lemma holds.

**Lemma 3** If $G$ is $\epsilon$--far from $P_n$ then $G$ has at least $\frac{\epsilon}{2} dn$ components of size at most $\frac{2}{\epsilon d}$.

**Proof** Suppose not. That is, suppose $G$ is $\epsilon$--far from $P_n$ and has less than $\frac{\epsilon}{2} dn$ components of size at most $\frac{2}{\epsilon d}$. Then by lemma 1, there must be at least $\left(\epsilon dn + 1 - \frac{\epsilon dn}{2} \right) = \frac{\epsilon dn}{2} + 1$ components of size at least $\frac{2}{\epsilon d}$ (we are looking at the number of components in the complement). This implies that the number of vertices is at least $\left(\frac{\epsilon dn}{2} + 1\right) \cdot \frac{2}{\epsilon d} = n + \frac{2}{\epsilon d} > n$. This is a contradiction.

Intuitively, the above lemma tells us that when a graph is $\epsilon$--far from $P_n$, there are many small components. Thus to construct a tester, we will look for small components as the deciding factor in acceptance or rejection.

**Tester:**

1
Algorithm 1 Connectivity Tester

Pick \( s = \frac{4}{cd} \) vertices, \( v_1, \ldots, v_s \).
For each vertex \( v_i \), run a BFS from \( v_i \).
If we ever encounter a component of size \( < \frac{2}{cd} \), then reject. Otherwise, accept.

1. Pick \( s = \frac{4}{cd} \) vertices, \( v_1, \ldots, v_s \).
2. For each vertex \( v_i \), run a BFS from \( v_i \).
3. If we ever encounter a component of size \( < \frac{2}{cd} \), then reject. Otherwise, accept.

As for the analysis, we must show 3 things: the number of queries matches up, completeness (we accept graphs in \( P_n \)), and soundness (we reject \( \epsilon \)-far graphs with probability at least \( \frac{2}{3} \)).

Analysis

Queries:
We sample \( \frac{4}{cd} \) nodes and for each node, we run a BFS until we have at least \( \frac{2}{cd} \) nodes in the component containing \( v_i \). Now this component could be dense and has at most \( \left( \frac{2}{cd} \right)^2 \) edges. This means we perform at most \( O\left( \frac{4}{cd} \cdot \left( \frac{2}{cd} \right)^2 \right) = O\left( \frac{1}{(cd)^2} \right) \) edge queries where \( \epsilon \) is independent of \( n \).

Completeness:
This is trivial, we never reject such a graph as there is only one component.

Soundness:
We must show that for an \( \epsilon \)-far graph that we reject this graph with probability at least \( \frac{2}{3} \) or accept with probability less than \( \frac{1}{3} \). In this instance, we will look at the acceptance case. Suppose that \( G \) is \( \epsilon \)-far from \( P_n \). Then by lemma 2 there are at least \( \frac{edn}{2} \) connected components that would cause us to reject. Thus we have that

\[
Pr[\text{accept}] = Pr[\text{don’t hit a small component}] = Pr_v[\text{v not contained in small component}]
\]

Now since there at least \( \frac{edn}{2} \) bad components, there must be at least \( \frac{edn}{2} \) nodes that could be sampled from a component that would cause us to reject (each component must contain at least one node). Thus we have that \( Pr_v[v \text{ contained in a bad component}] \geq \frac{edn}{2} \cdot \frac{1}{n} = \frac{ed}{2} \). Denote that quantity by \( p \). Then

\[
Pr[\text{accept}] \leq (1 - p) \frac{4}{cd} = (1 - p)^2 \frac{2^2}{p} \leq e^{-\frac{2^2}{p}} < \frac{1}{3}
\]

where the term \( 1 - p \) represents the probability that \( v_i \) is not in a bad component and the \( \frac{4}{cd} \) represents the number of nodes sampled.
0.2 Estimating the Number of Connected Components

The following is due to a result of Chazelle, Rubinfeld, and Trevisinon [1]. Recall that if \( \text{OPT} \) denotes the solution to some problem and \( x \) some approximation, then \( x \) is an \( \alpha \)-additive approximation if \( |x - \text{OPT}| < \alpha \).

**Theorem 4** There exists an randomized algorithm which on input \( G \) and \( \epsilon \) outputs an \( \epsilon n \)-additive approximation to the number of connected components of a graph with probability \( \frac{2}{3} \).

Let \( v \in V \) and let \( C_v \) denote the component in \( G \) that contains \( v \). Furthermore, let \( n_v = |C_v| \).

Then we have \( \sum_{v \in C_i} \frac{1}{n_v} = 1 \) and also that \( \sum_{v \in V} \frac{1}{n_v} = \sum_i \sum_{v \in C_i} \frac{1}{n_v} = \# \text{ of components of } G \).

**Lemma 5**

\[
0 \leq \frac{1}{\hat{n}_v} - \frac{1}{n_v} < \frac{\epsilon}{2}
\]

**Proof** If \( \hat{n}_v = n_v \) this is obvious. So suppose that \( \frac{2}{\epsilon} = \hat{n}_v < n_v \) then we have that \( 0 < \frac{1}{\hat{n}_v} - \frac{1}{n_v} < \frac{\epsilon}{2} \).

Let \( C = \sum_v \frac{1}{n_v} \) and \( \hat{C} = \sum_v \frac{1}{\hat{n}_v} \).

**Lemma 6**

\[
|C - \hat{C}| \leq \frac{\epsilon n}{2}
\]

**Proof**

\[
|C - \hat{C}| = \sum_v \left( \frac{1}{\hat{n}_v} - \frac{1}{n_v} \right) \leq \sum_v \frac{\epsilon}{2} = \frac{\epsilon n}{2}
\]

**Algorithm 2** Approximate Number of Connected Components of \( G \)

**Require:** \( G, \epsilon \)

Pick \( s = O \left( \frac{1}{\epsilon^2} \right) \) vertices and let \( S = \{v_1, \ldots, v_s\} \).

Run BFS from each \( v_i \) to visit at most \( \frac{2}{\epsilon} \) nodes locally.

Set \( \hat{n}_v = \min \left( n_v, \frac{2}{\epsilon} \right) \)

Output \( C' = \frac{n}{s} \sum_{v \in S} \frac{1}{\hat{n}_v} \)

**Theorem 7**

\[
|C' - C| \leq \epsilon n \text{ with probability } \frac{2}{3} \text{ in time } O \left( \frac{1}{\epsilon^4} \right)
\]
To prove this theorem, we will need to use the Chernoff bounds. Let \( X_1, \ldots, X_s \) be i.i.d. random variables and let \( X = \sum_{i=1}^{s} X_i \). Then we have that

\[
\Pr[|X - E(X)| \geq \delta s] \leq e^{-\Omega(\delta^2 s)}
\]

or equivalently

\[
\Pr\left[ \left| \frac{X}{s} - \frac{E(X)}{s} \right| \geq \delta \right] \leq e^{-\Omega(\delta^2 s)}
\]

Lemma 8 Let \( \hat{C}, C' \) be as defined above. Then

\[
\Pr[|C' - \hat{C}| > \frac{\epsilon n}{2}] < \frac{1}{\epsilon}
\]

Proof Let \( X_i = \frac{1}{\hat{C}_i} \) and \( X = \sum_{i \in [s]} X_i \). Then note that \( E(X_i) = \frac{1}{\hat{C}_i} \sum_{v \in G} \frac{1}{\hat{C}_v} = \frac{\hat{C}}{n} \) by definition of \( \hat{C} \). So \( E(X) = s^2 \frac{\hat{C}}{n} \). By definition of \( C' \) and \( X \) we have that \( \frac{n}{s} X = C' \). Applying the Chernoff bound to the \( X_i \)'s and substituting those expressions for the corresponding \( C' \) and \( \hat{C} \) we have

\[
\Pr \left[ \left| \frac{C' - \hat{C}}{\epsilon} \right| > \frac{en}{2} \right] = \Pr \left[ \left| \frac{n}{s} X - \frac{n}{s} E(X) \right| > \frac{en}{2} \right] = \Pr \left[ \left| \frac{1}{s} X - \frac{1}{s} E(X) \right| > \frac{\epsilon}{2} \right] \leq e^{-\Omega(\epsilon^2 s)}
\]

Now we can set \( s = O \left( \frac{1}{\epsilon^2} \right) \) to get that

\[
\Pr \left[ |C' - \hat{C}| > \frac{en}{2} \right] \leq e^{-\Omega(\epsilon^2 s)} \leq \frac{1}{3}
\]

Now to prove the main theorem, we need to combine the previous 2 lemmas and apply the triangle inequality. We have that

\[
|C - C'| \leq |C - \hat{C}| + |\hat{C} - C'| \leq \frac{en}{2} + \frac{en}{2} \text{ with probability } \frac{2}{3}
\]

where the last inequality is achieved by combing the previous 2 lemmas.

0.3 Estimating Weight of a Minimum-Spanning-Tree (MST)

Suppose we have a graph \( G \) with bounded positive integer edge weights \( \{1, \ldots, w\} \). Our goal is to estimate a Minimum Spanning Tree of \( G \). We will do this by using the previous connected components algorithm as a black box and split the minimum spanning tree into several smaller subtrees with a bounded edge weight.

Let \( G_i \) be the subgraph in \( G = (V, E) \) consisting of vertices \( V \) and the edges of \( E \) whose edge weights are at most 1. Also let \( C_1 \) denote the number of connected components of \( G_1 \). Let \( T \) denote a MST.
Lemma 9  The number of edges in $T$ of weight greater than 1 is $C_1 - 1$.

Proof  Consider the subgraph $G_1$ and its connected components. Furthermore recall the greedy MST algorithm of Kruskal. Kruskal’s algorithm will find a subtree for each component of $G_1$ first when choosing edges of degree 1. So the number of edges with weight greater than 1 must connect the $C_1$ components of $G_1$. This is done with exactly $C_1 - 1$ edges, all of which must have weight greater than 1. ■

In general, we have that

$$(\# \text{ of edges in } T \text{ of weight greater than } j) = \sum_{i=j+1}^{w} N_j,$$

where $N_j$ is the number of edges of weight $j$ in $T$

Lemma 10  

$$w(MST(G)) = \sum_{0 \leq i \leq w-1} C_i - 1$$

References