Sub-Linear Time Algorithms

If we settle for approximations, we can sometimes get much more efficient algorithms.

Last week - poly-time approx algo for NP-hard problems.
This time - sub-linear time approx algorithms

We'll see two examples:

1) Estimating the number of connected components.

2) Estimating the size of a min spanning tree when weights are in $\mathbb{S}_1, \ldots, \mathbb{S}_n$ constant

Recall in the take-home exam had:
Given a graph represented by adjacency matrix, and a node in the graph, estimate the degree of the node, up to $\pm 1 n$. 
Estimating the Number of Connected Components in a Graph

Will get additive approximation $\pm 3n$, where $\varepsilon$ is a parameter and the running time is $\text{poly} (\max (\text{degree}, \frac{1}{\varepsilon}))$. $s$ bounds prob of error.

We assume the graph is given as adjacency lists.

Lemma If we denote the number of vertices in the connected component of a vertex $v$ by $n_v$, then

$$\# \text{ connected components} = \sum_{v \in V} \frac{1}{n_v}$$
Approximate $\mu_c$

1) Pick $\Theta(\frac{1}{\varepsilon^3 \log \frac{1}{\delta}})$ vertices $v_1 \sim v_t$ uniformly at random.

2) For $i = 1, \ldots, t$, use BFS from $v_i$ to compute $\tilde{\nu}_i = \min \{ \nu_i, \frac{2}{\varepsilon} \}$.

3) Output $\nu = \left( \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\tilde{\nu}_i} \right)$

$$\text{runtime} = \Theta(\frac{1}{\varepsilon^3 \log \frac{1}{\delta}})$$

\[ \forall v \in V \]
\[ \frac{1}{\tilde{\nu}} - \frac{1}{\nu} \leq \frac{\varepsilon}{2} \]
\[ \Rightarrow \sum_{v \in V} \frac{1}{\tilde{\nu}} - \sum_{v \in V} \frac{1}{\nu} \leq \frac{\varepsilon |V|}{2} \]

\[ P \left( \left| \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\tilde{\nu}_i} - \frac{1}{\nu} \sum_{v \in V} \frac{1}{\tilde{\nu}_i} \right| \geq \frac{\varepsilon}{2} \left( \frac{1}{\tilde{\nu}} \sum_{v \in V} \frac{1}{\tilde{\nu}_i} \right) \right) \leq \delta \]

To prove (2) use:

Hoeffding (Chernoff for real r.v.) $X_i \sim X_t$ independent, $a \leq X \leq b$.

\[ P \left( \left| \frac{1}{t} \sum_{i=1}^{t} X_i - \mathbb{E} \left( \frac{1}{t} \sum_{i=1}^{t} X_i \right) \right| \geq \varepsilon \mathbb{E} \left( \frac{1}{t} \sum_{i=1}^{t} X_i \right) \right) \leq \exp \left( -\frac{2 \varepsilon^2 \mathbb{E} \left( \frac{1}{t} \sum_{i=1}^{t} (X_i - a)^2 \right)}{2} \right) \]
Since \( \frac{3}{2} \leq \frac{1}{\hat{\nu}_i} \leq 1 \) \( \forall i \), \( E\left[ \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\hat{\nu}_i} \right] = \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\nu_{\text{ev}} \hat{\nu}_i} \) \{\frac{1}{\hat{\nu}_i}\}_i \) are independent, have:

\[
P \left( \left| \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\hat{\nu}_i} - \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\nu_{\text{ev}}} \right| \geq \frac{\epsilon}{2} \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\nu_{\text{ev}}} \right) \leq 2 \exp \left( -\frac{\epsilon^2}{2} \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\nu_{\text{ev}}} \right)
\]

Note: \( \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\nu_{\text{ev}}} \geq \frac{\epsilon}{2} \) so for \( t = O\left(\frac{\nu_{\text{ev}}}{\epsilon^2}\right) \), can have \( 2 \exp \left( -\frac{\epsilon^2}{2} \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\nu_{\text{ev}}} \right) \) at most 0.8.

\( \square \)

Putting (1) + (2) together: with prob \( \geq 1 - 0.8 \),

\[
\left| \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\hat{\nu}_i} - \frac{1}{\nu_{\text{ev}}} \right| \leq \left| \frac{1}{t} \sum_{i=1}^{t} \frac{1}{\hat{\nu}_i} - \sum_{i=1}^{t} \frac{1}{\nu_{\text{ev}}} \right| + \left| \sum_{i=1}^{t} \frac{1}{\nu_{\text{ev}}} - \sum_{i=1}^{t} \frac{1}{\hat{\nu}_i} \right| \leq \frac{3}{2} + \frac{3}{2} = 3
\]

\( \square \)
Minimum Spanning Tree's Size
When Weights are Small

Input  A graph $G=(V,E)$ given by adjacency lists, $G$ connected, max degree is $d$.
* Weight func. $w: E \rightarrow \{1, \ldots, W\}$. * Parameter $\epsilon > 0$.
Output  A number $t$, such that:

$$(1-\epsilon)\text{MST} \leq t \leq (1+\epsilon)\text{MST}.$$ 

Idea  Can create an MST by:
- Taking all edges of weight 1 that don't close a cycle.
- Taking all edges of weight 2.
- """" 3
- """"

Until took $n-1$ edges / get a tree.
Lemma \[ \text{MST} = |V| - w + \sum_{i=1}^{w-1} C^{(i)} \]

when \( C^{(i)} \) is the number of connected components in the graph that contains only edges of weight \( \leq i \).

\[ \text{Pf} \]

Let: \( A_i = \) number of edges of weight \( i \) in min spanning tree of \( G \).

\( B_i = \) number of edges of weight \( \geq i \).

Note

\[ \text{MST} = \sum_{i=1}^{w} i \cdot A_i = \sum_{i=1}^{w} B_i \]

and \( B_i = C^{(i-1)} - 2 \) \( i \geq 2 \)

\( B_1 = |V| - 1 \) \( i = 1 \)

\[ \Rightarrow \text{MST} = |V| - w + (C^{(r-1)}) + (C^{(r-2)}) + \ldots \]

\[ = |V| - w + \sum_{i=1}^{w-1} C^{(i)} \]
\[ \text{MST-approx} \]

For \( i = 1 \rightarrow w-1 \)

\[ G^{(i)} = \text{graph on edges of weight } i \]

\[ \hat{C}^{(i)} = \text{approx- } \#\text{cc}(G^{(i)}, \frac{\varepsilon}{2w}, \frac{1}{100w}) \]

Return \( |V| - w + \sum_{i=1}^{w-1} \hat{C}^{(i)} \)

Run-time = \( \text{poly}(\frac{1}{\varepsilon}, w, \log d) \).

Note that we don't actually compute \( G^{(i)} \), we just modify the \text{approx- } \#\text{cc} algo to ignore edges of weight \( > i \) (Note: max degree is \( d \)).

\underline{Correctness} \hspace{1cm} \text{By correctness of } \text{approx- } \#\text{cc}, \hspace{1cm} \text{except with prob} \hspace{1cm} \frac{1}{100},

\[ n - w + \sum_{i=1}^{w-1} \hat{C}^{(i)} \geq n - w + \sum_{i=1}^{w-1} C^{(i)} - \sum_{i=1}^{w-1} \frac{\varepsilon}{2w} n \]

\[ = \text{MST} - \frac{\varepsilon}{2} n \]

(MST \( \geq n-1 \)) \( \rightarrow \geq \text{MST} \cdot (1 - \varepsilon) \). Similarly for other direction. \( \square \)