COL751 - Lecture 3

1 Construction of +2 distance preserver in $\widetilde{O}(n^2)$ time

Recall the algorithm for +2 distance preserver from lecture 2. It involved:

- Adding to H all edges of low ($\leqslant \sqrt{n}$) degree vertices.
- Computing a hitting set R (of size $\sqrt{n}\log n$) that intersects neighborhood of high degree vertices.
- For each $v \in R$, adding to H edges of BFS tree T_v rooted at vertex v in G.

The time complexity of the algorithm is clearly $O(m|R|)$ which is $O(n^{2.5}\log n)$ in the worst case. As running APSP algorithm on graph H will give us +2 approximate distances we have the following corollary.

Corollary For any unweighted undirected graph $G = (V, E)$ we can compute all-pairs approximate distance with a +2 additive error in $O(n^{2.5}\log n)$ time.

In this section we will see how to improve the running time to $\widetilde{O}(n^2)$ at expense of an extra log factor in the size of H[1].

Assumption We assume n and $\sqrt{n}$ are powers of 2.

Improving running time to $O(n^2\log^2 n)$ We partition the vertex set into $O(\log n)$ sets: For each $d \in [1, 2, 4, 8, \ldots, n/2]$, let V_d be subset of all those vertices whose degree lie in the range $[d, 2d-1]$.

The alternate construction of H is as follows:

1. Initialize $H = (V, E_H)$, where E_H contains all edges incident to vertices of low degree (i.e. degree at most $\sqrt{n}$).
2. For each $d \in [\sqrt{n}, n] \cap [1, 2, 4, 8, \ldots, n/2]$:
   - Let G_{2d} be a graph in which keep all edges of vertices with degree at most $2d$.
   - Let R_d be a hitting set of size $O(\frac{n\log n}{d})$ that hits all neighbourhoods of size at least d.
   - For each $r \in R_d$, compute a BFS tree T_v rooted at v in graph G_{2d}, and add all edges of T_v to H.

[1] $\widetilde{O}(\cdot)$ hides the polylogarithmic factors

Lemma 1 *The number of edges in H is at most $O(n\sqrt{n}\log^2 n)$.*

Lemma 2 *For any $x, y \in V$, we have $dist(x, y, H) \leqslant dist(x, y, G) + 2$.*

Proof: Consider a vertex pair $(x, y) \in V \times V$. Let w be a vertex of maximum degree on an (x, y) shortest path.

Case 1: Degree of w is at most $\sqrt{n}$.
In this case the entire shortest path will lie in H as all vertices on the shortest path have degree at most $\sqrt{n}$.

Case 2: Degree of w lie in the range $[d, 2d]$, where $d \geqslant \sqrt{n}$ is a power of 2.
In this case the entire shortest path will lie in G_{2d} as all vertices on the shortest path have degree at most $2d$. Further, as degree of w is at least d, a neighbor of w, say r, will lie in the set R_d.

We have:

$$\begin{aligned}
dist(x, y, H) &\leqslant dist(x, r, H) + dist(r, y, H) \\
&\leqslant dist(x, r, G_{2d}) + dist(r, y, G_{2d}) \\
&\leqslant dist(x, w, G_{2d}) + 1 + dist(w, y, G_{2d}) + 1 \\
&= dist(x, y, G_{2d}) + 2 \\
&= dist(x, y, G) + 2.
\end{aligned}$$

This proves that distances are stretched by an additive factor of at most two. □

Theorem 3 (Dor, Halperin, Zwick (FOCS 1996)) *For any unweighted undirected graph $G = (V, E)$ we can construct in $O(n^2 \log^2 n)$ time a subgraph graph H of $O(n^{1.5} \log^2 n)$ size satisfying*

$$dist(x, y, H) \leqslant dist(x, y, G) + 2,$$

for every $x, y \in V$.

Homework What is maximum possible log factors that can be eliminated in running time and/or size of H?

Remark A drawback of +2 additive distance preserver is that it fails for weighted graphs with edge weights in range $[1, W]$, as the additive stretch can be extremely high ($2W$ in the worst case). So, for weighted graphs we next consider distance preservers with multiplicative stretch.

2 Multiplicative spanners

In an unweighted graph, a +2 additive spanner is also a 3-multiplicative spanner, so the first natural question is if we can have a construction of 3-multiplicative spanner for weighted graphs.

Algorithm 1 presents an $O(mn\sqrt{n})$ time algorithm for 3-multiplicative spanners.

Let $H = (V, \emptyset)$.
Let $(e_1, e_2, \ldots, e_m)$ be the sequence of $m = |E|$ edges in G sorted in increasing order of weight.
for $i = 1$ *to* m **do**
  Let x, y be endpoints of e.
  If $unweighted - distance(x, y, G) \gneq 3$ then add e_i to H.
end

Algorithm 1: 3-multiplicative spanner of weighted graph G

Lemma 4 *The graph H computed by algorithm is indeed a 3-multiplicative spanner. In particular, for any $x, y \in V$, $dist(x, y, H) \leqslant 3 \cdot dist(x, y, G)$.*

Proof: It suffices to show that for any $e = (x, y)$ not in H, there exists an (x, y) path P in H of weight at most $3 \cdot wt(x, y)$.

Consider an edge $e_i = (x_i, y_i)$ not lying in H, where $i \in [1, m]$. Further, let H_i be the graph H after i^{th} iteration of for loop in Algorithm 1. As e_i is not in H_i, there would exists an unweighted path of length at most 3 in H_i, say P. Since all edges in H_i have weight at most $wt(e_i)$, we have that $wt(P) \leqslant 3 \cdot wt(e_i)$. This proves $dist(x_i, y_i, H_i) \leqslant dist(x_i, y_i, H) \leqslant 3 \cdot wt(x_i, y_i)$, for each $i \in [1, m]$. □

Lemma 5 *Each cycle in H has length strictly larger than 4.*

Proof: Assume on contrary there is a cycle C of length at most 4 in H. Let $e = (x, y)$ be an edge of maximum weight in C. Then before adding e we would have added all edges of $C \setminus \{e\}$ to H. In other words, before adding e, there would exists an unweighted path of length at most 3 in H. This contradicts the assumption that e lies in H. □

Lemma 6 (Lecture 2) *Any graph H with girth 5 or more contains $O(n^{1.5})$ edges.*

On combining above three lemmas we get the following theorem.

Theorem 7 (Althöfer et al. (Discrete Comput. Geom. 1993)) *For any n vertex m edges weighted undirected graph G we can construct in $O(mn\sqrt{n})$ time a subgraph H with $O(n^{1.5})$ edges satisfying*

$$dist(x, y, H) \ \leqslant \ 3 \ dist(x, y, G),$$

for every $x, y \in V$.

Homework Prove that the running time of Algorithm 1 is indeed $O(mn\sqrt{n})$.