COL751 - Lecture 4

1 Computing approximate distance matrix

In this section we will see how to use the ideas from +2 distance preserver from lecture 3 to compute approximate distance matrix M with additive error at most $+2$.

In order to compute M, we will compute three matrices M_{low}, M_{mid}, M_{high}.

1.1 M_{low}

Let G_{low} be graph in which we add all edges incident to low degree vertices (i.e. vertices of degree at most $n^{1/3}$). We simply run BFS from each possible vertex v in graph G_{low} to compute distance matrix M_{low}.

Lemma 1 *Let $x, y \in V$ satisfy that all vertices on an (x, y) shortest path have degree at most $n^{1/3}$. Then $M_{low}[x, y] = dist(x, y, G)$.*

The time to compute M_{low} is $O(n \times |E(G_{low})|)$ which is $O(n^{7/3})$.

1.2 M_{high}

We sample a set R of size $8n^{1/3} \log n$ and compute BFS tree rooted at each vertex $r \in R$ in graph G. This provides us distance between all vertex pairs in $R \times V$.

Next, for each $x, y \in V$ set $M_{high}[x, y] \;=\; \min_{r \in R}\ dist(x, r, G) + dist(r, y, G)$.

Lemma 2 *Let $x, y \in V$ satisfy that at least one vertex on an (x, y) shortest path has degree larger than $n^{2/3}$. Then $M_{high}[x, y] \leqslant dist(x, y, G) + 2$.*

Proof: Let w be a vertex on (x, y) shortest path having degree larger than $n^{2/3}$. Then in this case with high probability a neighbor of w, say r, will lie in the set R.

We have:

$$\begin{aligned} M_{high}[x, y] &\leqslant dist(x, r, G) + dist(r, y, G) \\ &\leqslant dist(x, w, G) + 1 + dist(w, y, G) + 1 \\ &= dist(x, y, G) + 2. \end{aligned}$$

This proves the claim. □

The time to compute M_{high} is $O(n^2|R| + m|R|)$ which is $O(n^{7/3} \log n)$.

1.3 M_{mid}

We are left to handle the following class of vertex pairs.

$\mathcal{P}$: The set of all pairs $(x, y) \in V \times V$ for which the maximum degree on an (x, y) shortest path lie in the range $[n^{1/3}, n^{2/3}]$.

Let G_{mid} be graph in which we include all edges incident to vertices of degree at most $n^{2/3}$. Observe that the vertex pairs in $\mathcal{P}$ have the corresponding shortest path in graph G_{mid}. Also the number of edges in G_{mid} is at most $n^{5/3}$.

Let R_0 be a uniformly random subset of size $O(n^{2/3} \log n)$. For each vertex v having degree at least $n^{1/3}$, we compute a representative vertex r_v in set $R_0 \cap N(v)$ and add edge (v, r_v) to G_{low}. As at most n edges are added to G_{low}, the number of edges in G_{low} remains $O(n^{4/3})$.

Lemma 3 *For each* $(x, y) \in \mathcal{P}$, $\min_{r \in R_0} dist(x, r, G) + dist(r, y, G_{low}) \leqslant dist(x, y, G) + 2$.

Proof: Consider a pair $(x, y) \in \mathcal{P}$. Let Q be an (x, y) shortest path in G, and $w \in Q$ be the vertex farthest from x having degree in the range $[n^{1/3}, n^{2/3}]$. So all the successors of w in Q have degree at most $n^{1/3}$.

Let r_w be a representative neighbor of w lying in set R_0. Observe that the edge (r_w, w) as well as the subpath $Q[w, y]$ lie in graph G_{low}. We have:

$$\begin{aligned} \min_{r \in R_0} dist(x, r, G) + dist(r, y, G_{low}) &\leqslant dist(x, r_w, G) + dist(r_w, y, G_{low}) \\ &\leqslant dist(x, w, G) + 1 + dist(w, y, G_{low}) + 1 \\ &= dist(x, y, G) + 2. \end{aligned}$$

This proves the claim. □

The steps to compute matrix M_{mid} are as follows:

1. We first compute distances between all the vertex pairs in $R_0 \times V$ in graph G_{mid} in $O(|R_0| \times n^{5/3}) = O(n^{7/3} \log n)$ time.

2. For each $x \in V$, we compute an auxiliary graph H_x from G_{low} by adding for each $r \in R_0$ an edge (x, r) of weight $dist(x, r, G)$ to graph G_{low}. Clearly the number of edges in H_x is bounded by $O(n^{4/3})$.

3. Finally for each vertex $x \in V$, we compute a shortest path T_x rooted at x in graph H_x in $O(n^{4/3} \log n)$ time and set

$$M_{mid}[x, y] = dist(x, y, H_x),$$

for $y \in V$.

Clearly M_{mid} is in worst case an overestimation to distances in graph G. Further, by Lemma 3 it follows that for pair $(x, y) \in \mathcal{P}$, $M_{mid}[x, y] \leqslant dist(x, y, G) + 2$.

On setting M as $\min(M_{low}, M_{mid}, M_{high})$ we get the following theorem.

Theorem 4 (Dor, Halperin, Zwick (FOCS 1996)) *For any n vertex unweighted undirected graph $G = (V, E)$ we can construct in $O(n^{7/3} \log n)$ time an approximate distance matrix M satisfying*

$$dist(x, y, G) \ \leqslant\ M[x, y] \ \leqslant\ dist(x, y, G) + 2,$$

for every $x, y \in V$.

2 Hardness of distance approximation

We next prove that computing a $+1$ approximation to all-pairs distances in n vertex undirected graphs takes at least $\Omega(n^\omega)$ time, where ω is constant of matrix multiplication.

Theorem 5 *The problem of computing a distance matrix with a +1 additive error in an n vertex undirected graph is as hard as Boolean matrix multiplication and thus takes $\Omega(n^\omega)$ time.*

Proof: Let A, B be two $n \times n$ Boolean matrices. Consider the following graph G:

1. The vertex set of G consists of three disjoint sets $X = (x_1, \ldots, x_n)$, $Y = (y_1, \ldots, y_n)$, and $Z = (z_1, \ldots, n_n)$.
2. For each $i, j, k \leqslant n$, (x_i, y_j) is an edge in G if and only if $A_{ij} = 1$, and (y_j, z_k) is an edge if and only if $B_{jk} = 1$.

Observe that for any i, k in the range $[1, n]$,

1. $dist(x_i, z_k)$ is even as $(Y, X \cup Z)$ is a bi-partition of G.
2. $dist(x_i, z_k, G) = 2$ if and only if $(AB)_{ik} > 0$.

Thus, if $\hat{M}$ is a $+1$ approximate distance matrix then for any i, k in the range $[1, n]$, $\hat{M}(x_i, z_k, G) \leqslant 3$ if and only if $(AB)_{ik} > 0$. This proves that computing $\hat{M}$ is as hard as Boolean matrix multiplication. □

The above result can be generalized to directed graphs for arbitrary stretch.

Theorem 6 *The problem of computing a distance matrix with any finite stretch in an n vertex directed graph is as hard as Boolean matrix multiplication and thus takes $\Omega(n^\omega)$ time.*

Proof: Let A, B be two $n \times n$ Boolean matrices. We construct a graph G similar to Lemma 5 construction with the variation that edges are directed from X to Y, and from Y to Z. Observe that for any i, k in the range $[1, n]$, $dist(x_i, z_k, G)$ is finite if and only if $(AB)_{ik} > 0$. This proves that obtaining any finite approximation to distances in directed graphs is as hard as Boolean matrix multiplication. □

3 Multiplicative spanners

In last lecture we considered following construction of 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 $unweighted - distance(x, y, G) \gnsim 3$ then add e_i to H.
end

Algorithm 1: 3-multiplicative spanner of weighted graph G

Lemma 7 *The graph H computed by algorithm 1 satisfies following claims:*

1. *For any $x, y \in V$, $dist(x, y, H) \leqslant 3 \cdot dist(x, y, G)$.*
2. *Each cycle in H has length strictly larger than 4.*

This combined with the lemma below proved the construction of 3-multiplicative spanners with $O(n^{1.5})$ edges.

Lemma 8 (Lecture 2) *Any graph H with girth 5 or more contains $O(n^{1.5})$ edges.*

3.1 Construction of $(2k-1)$ Multiplicative spanners

In order to generalize our construction, we first establish the following lemma providing a bound on the number of edges in a graph with high girth.

Lemma 9 *Every n-vertex graph G with girth $2k+1$ or more contains $O(n^{1+1/k})$ edges.*

Proof: Let G be any graph with girth at least $2k+1$. We first repeatedly remove all the edges incident to vertices of degree at most $n^{1/k}$. After this procedure if G becomes an empty graph then we are done. Suppose not. Consider any vertex s in the resultant graph G_0.

Since the girth of G_0 is at least $2k+1$, two vertices x, y at distance at most $k-1$ from s will have no common neighbor. Hence, the number of vertices at distance i from s is at least $(n^{1/k})^i$ for every $i \leqslant k$. So, the number of vertices at distance k from s is at least $(n^{1/k})^k$ which is at least n, leading to contradiction. □

Now, we generalize the algorithm and property of constructed H as follows.

Lemma 10 *The graph H computed by algorithm 2 satisfies following claims:*

1. *For any $x, y \in V$, $dist(x, y, H) \leqslant (2k-1) \cdot dist(x, y, G)$.*
2. *Each cycle in H has length strictly larger than $2k$.*

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 $unweighted - distance(x, y, G) \gneq 2k - 1$ then add e_i to H.
end

Algorithm 2: $(2k-1)$-multiplicative spanner of weighted graph G

Proof: To prove part 1, 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 $(2k-1) \cdot wt(x, y)$. 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 $2k-1$ in H_i, say P. Since all edges in H_i have weight at most $wt(e_i)$, we have that $wt(P) \leqslant (2k-1) \cdot wt(e_i)$. This proves $dist(x_i, y_i, H_i) \leqslant dist(x_i, y_i, H) \leqslant (2k-1) \cdot wt(x_i, y_i)$, for each $i \in [1, m]$.

To prove part 2, assume on contrary there is a cycle C of length at most $2k$ 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 $2k-1$ in H. This contradicts the assumption that e lies in H. □

On combining above two lemmas we get the following theorem.

Theorem 11 (Althöfer et al. (Discrete Comput. Geom. 1993)) *For any n vertex m edges weighted undirected graph G we can construct in $O(mn^{1+1/k})$ time a subgraph H with $O(n^{1+1/k})$ edges satisfying*

$$dist(x, y, H) \ \leqslant \ (2k-1)\ dist(x, y, G),$$

for every $x, y \in V$.

Homework Prove that the running time of Algorithm 2 is indeed $O(m \cdot n^{1+1/k})$.