Succinct Graph Structures and Their Applications **Spring 2018**

Lecture 2: April 11

Lecturer: Merav Parter

Hitting Sets

The following tool is very useful in many randomized constructions of spanners and related structures. In the most general setting, we are given universe $\mathcal{U} = \{u_1, \ldots, u_n\}$ (e.g., the set of vertices in the graph) and a collection $\Sigma = \{S_1, \ldots, S_m\}$ of $m = poly(n)$ subsets $S_i \subseteq \mathcal{U}$ consisting of elements of $\mathcal{U}$ (e.g., the neighborhood sets of the vertices). We are in particular interested in the case where each $S_i \in \Sigma$ is large, say, has at least Δ elements (e.g., vertices of degree at least Δ). A subset $S \subseteq \mathcal{U}$ is a *hitting set* for Σ if $S \cap S_i \neq \emptyset$ for every $S_i \in \Sigma$. Our goal is to compute a *small* hitting set for Σ.

Lemma 2.1 *[Randomized Hitting-Set] Let $\Sigma = \{S_1, \ldots, S_n\}$ where each $S_i \subseteq \mathcal{U}$ has size $|S_i| \geq \Delta$ and $m = n^c$ for some constant c. There is a randomized algorithm that finds a subset $S \subseteq \mathcal{U}$ with $|S| = O(n \log n/\Delta)$ such that S is a hitting set for Σ with probability at least $1 - n^{-c'}$ for some $c' \geq 0$.*

Proof: We add each element $u_i \in \mathcal{U}$ to S with probability $p = (c + 3c') \cdot \log n/\Delta$. Using Chernoff, we get that $|S| \leq 2p \cdot n$ with probability at least $1 - n^{-2c'}$. The probability for S to miss a subset $S_i \in \Sigma$ is $\prod_{u \in S_i} Pr[u \notin S] = (1-p)^{\Delta} \leq n^{-(c+3c')}$. The proof follows by taking the union bound over all $m = n^c$ subsets in Σ. ■

Approximate Distance Oracles [TZ05]

Distance oracles are space-efficient data structures that answer fast distance queries between pairs of vertices. For an n-vertex undirected graph G, the distance oracle scheme consists of two algorithms: (1) *preprocessing* algorithm that given G computes the oracle $\mathcal{O}$ and (2) *query* algorithm that given $\mathcal{O}$ and a pair of vertices $u, v \in V(G)$ computes a distance estimate $\widehat{\delta}(u, v)$. We say that a distance oracle $\mathcal{O}$ is t-approximate if

$$\mathtt{dist}(u, v, G) \leq \widehat{\delta}(u, v) \leq t \cdot \mathtt{dist}(u, v, G).$$

The standard complexity measures are then the preprocessing time, the query time and the tradeoff between the size of the oracle and the approximation (or stretch) t. In this class, we mainly care about the query time and the space − stretch tradeoff. We will present an efficient construction of $(2k-1)$ approximate distance oracles ([TZ05]) of size $O(k \log n \cdot n^{1+1/k})$ that answers distance queries in $O(k)$ time. As we will see, this space–stretch tradeoff is nearly the best possible assuming the girth conjecture by Erdős. We start by illustrating the construction for $k = 2$.

3-approximate distance oracles. Our goal is to construct an oracle of size $O(n^{3/2} \log n)$ (i.e., same size as that of 3-spanners) that answers distance queries in $O(1)$ time and distort the distances in G up to factor 3. The idea is to compute for each vertex v, a collection of $O(\log n\sqrt{n})$ *important* vertices, $B(v)$, and to keep in the oracle the distances between v and each $w \in B(v)$. This is indeed within our budget. The main challenge is in picking this collection of important vertices.

The construction is based on a random sample of $O(\sqrt{n})$ vertices $S \subseteq V$, which is computed by adding each $v \in V$ into S independently with probability $q = 1/\sqrt{n}$. This process is denoted by $S \leftarrow \mathsf{Sample}(V, q)$. Using S, we define a subset $B_S(v)$ for every v by

$$B_S(v) = \{w \in V \setminus S \mid \mathtt{dist}(v, w, G) < \mathtt{dist}(v, p_S(v))\},$$

where $p_S(v)$ is the closest vertex to v in S. The final set of important vertices for v is $B(v) = B_S(v) \cup S$. The algorithm keeps (by 2-level hash) all the (w, v) distances in G for every $v \in V$ and $w \in B(v)$. For simplicity, we also store (explicitly) in the oracle the $p_S(u)$ and $\texttt{dist}(u, p_S(u))$ for every $u \in V$. This completes the description of the construction of the oracle $\mathcal{O}$. We next claim that the resulting oracle has size $O(n^{3/2}\log n)$ with high probability. To do that, it is sufficient to bound the cardinality of the $B_S(v)$ sets.

Claim 2.2 (Size) *W.h.p., $|B_S(v)| = O(\sqrt{n}\cdot\log n)$ for every $v \in V$.*

Proof: We define an auxiliary subset $N_S(v)$ that consists of the $\sqrt{n}\cdot\log n$ closest vertices to v in V. By Lemma 2.1, w.h.p., S *hits* (i.e., intersects) the subset $N_S(v)$, and hence $B_S(v) \subseteq N_S(v)$. The claim follows. ■

We now turn to describe the query algorithm, that given the oracle $\mathcal{O}$ and a vertex pair (u, v) computes $\widehat{\delta}(u, v)$. First, if $v \in B(u)$, the oracle returns $\widehat{\delta}(u, v) = \texttt{dist}(u, v, G)$. Otherwise, it returns $\widehat{\delta}(u, v) = \texttt{dist}(u, p_S(u), G) + \texttt{dist}(p_S(u), v, G)$. Note that since $p_S(u) \in S$, the oracle $\mathcal{O}$ indeed stores the distances $\texttt{dist}(u, p_S(u), G)$ and $\texttt{dist}(v, p_S(u), G)$.

Claim 2.3 (Stretch) *For every u, v, $\widehat{\delta}(u, v) \leq 3\cdot\texttt{dist}(u, v, G)$.*

Proof: If $v \in B(u)$, the claim trivially holds. Otherwise, if $v \notin B(u)$, it implies that $\texttt{dist}(u, p_S(u), G) \leq \texttt{dist}(u, v, G)$. Hence, $\texttt{dist}(p_S(u), v) \leq \texttt{dist}(p_S(u), u, G) + \texttt{dist}(u, v, G) \leq 2\cdot\texttt{dist}(u, v, G)$. We therefore have that: $\widehat{\delta}(u, v) = \texttt{dist}(u, p_S(u), G) + \texttt{dist}(p_S(u), v, G) \leq \texttt{dist}(u, v, G) + 2\cdot\texttt{dist}(u, v, G) \leq 3\cdot\texttt{dist}(u, v, G)$. ■

Note that just like in the randomized 3-spanner construction, we saw last time, also here the high probability guarantee is only for the size while the stretch guarantee holds deterministically (with probability 1).

$(2k-1)$-approximate distance oracles. To handle the general case of $k \geq 1$, instead of computing one sample set S, we compute an hierarchy of k subsets:

$$V = A_0 \supseteq A_1 \supseteq \ldots \supseteq A_{k-1}, \quad \text{where} \quad A_i = \mathsf{Sample}(A_{i-1}, n^{-1/k}), i \in \{1, \ldots, k-1\}.$$

We will compute for every v and for every $i \in \{0, \ldots, k-1\}$, a subset $B_i(v)$ of $O(n^{1/k}\log n)$ important vertices to v in A_i. Since the desired size of the oracle is $O(k\log n\cdot n^{1+1/k})$, we have the capacity to store all the distances between v to each of its important vertices, $B_i(v)$, in each level of the hierarchy. For each $i \in \{0, \ldots, k-1\}$, define:

$$B_i(v) = \{w \in A_i \setminus A_{i+1} \mid \texttt{dist}(v, w, G) < \texttt{dist}(v, p_{i+1}(v), G)\},$$

where $p_i(v)$ is the closest vertex to v in A_i (in particular, $p_0(v) = v$). Also, let $B_{k-1}(v) = A_{k-1}$ and $B(v) = \bigcup_{i=0}^{k-1} B_i(v)$. The algorithm stores (by 2-level hash) the distances between each $v \in V$ and $w \in B(v)$. To show that the output oracle has size $O(k\log n\cdot n^{1+1/k})$, it is sufficient to show:

Claim 2.4 *W.h.p, $|B_i(v)| = O(n^{1/k}\cdot\log n)$ for every $v \in V$ and every $i \in \{0, \ldots, k-1\}$.*

Proof: Since $B_{k-1}(v) = A_{k-1}$, by Chernoff $|A_{k-1}| = O(n^{1/k}\cdot\log n)$ and the claim follows for $i = k-1$. We now consider the case where $i \leq k-2$ and define the auxiliary subset $N_i(v)$ to be the subset of $n^{1/k}\log n$ closest vertices to v in A_i. By Lemma 2.1, w.h.p., A_{i+1} hits $N_i(v)$ and hence $B_i(v) \subseteq N_i(v)$. ■

We proceed by describing the query algorithm. See Fig. 2.1.

Note that Algorithm Query always returns an answer since $p_{k-1}(v)$ is in A_{k-1} and hence in $B(u)$. The intuition for the query algorithm is that for a given pair $\langle u, v\rangle$, the goal is to find the minimum i such that $p_i(v) \in B(u) \cap B(v)$. When $p_i(v)$ is not in $B_i(u)$, it implies that $p_{i+1}(u)$ is sufficiently close to u (as a function of the distance between u and v in G). We now analyze this algorithm formally.

Claim 2.5 *Let u_i, v_i, w_i be the nodes that play the u, v, w roles in iteration i. Then, $\texttt{dist}(v_i, w_i) \leq (i-1)\cdot\delta$ for every $i \in \{1, \ldots, k-1\}$.*

Algorithm $\mathsf{Query}(u, v, \mathcal{O})$

1. $w \leftarrow p_0(v) = v$.
2. For $i = 1$ to k do:
   (a) If $w \in B(u)$ then: (here $w = p_{i-1}(v)$)
      i. Return $\widehat{\delta}(u, v) = \mathtt{dist}(v, w, G) + \mathtt{dist}(w, u, G)$.
   (b) Else:
      i. $w \leftarrow p_i(u)$,
      ii. swap u and v.

Figure 2.1: An $O(k)$-time query algorithm

Proof: In iteration i, $w_i = p_{i-1}(v_i)$. The proof is shown by induction on i. For $i = 1$, the claim holds trivially as $w_i = v$. Assume that the claim holds up to $i - 1$ and consider iteration $i \geq 2$. In iteration $i - 1$, $w_{i-1} = p_{i-2}(v_{i-1})$, hence $w_{i-1} \in A_{i-2}$. Since the algorithm did not halt at iteration $i - 1$, we have that $w_{i-1} \notin B(u_{i-1})$ and thus

$$\begin{aligned} \mathtt{dist}(u_{i-1}, p_{i-1}(u_{i-1}), G) &\leq \mathtt{dist}(u_{i-1}, w_{i-1}, G) \leq \mathtt{dist}(u_{i-1}, v_{i-1}, G) + \mathtt{dist}(v_{i-1}, w_{i-1}, G) \\ &\leq \delta + (i-2)\delta \leq (i-1) \cdot \delta \ , \end{aligned}$$

where the penultimate inequality follows by induction assumption. Since $u_{i-1} = v_i$ and $p_{i-1}(u_{i-1}) = w_i$, the claim follows. ■

Claim 2.6 *For every u, v, $\widehat{\delta}(u, v) \leq (2k - 1) \cdot \mathtt{dist}(u, v, G)$.*

Proof: Let $i \in \{1, \ldots, k\}$ be the iteration in which the query algorithm halts given the query u, v. By Claim 2.5, $\mathtt{dist}(v_i, w_i) \leq (i - 1) \cdot \delta$. The returned estimate is then: $\widehat{\delta}(u, v) = \mathtt{dist}(v_i, w_i) + \mathtt{dist}(w_i, u_i) \leq \mathtt{dist}(v_i, w_i) + \mathtt{dist}(w_i, v_i) + \mathtt{dist}(v_i, u_i) \leq 2(i - 1) \cdot \delta + \delta = (2i - 1)\delta$. Since $i \leq k$, the claim follows. ■

Size Lower Bound. We saw in the previous class, that the greedy construction of $(2k - 1)$ spanners with $O(n^{1+1/k})$ edges is optimal assuming Erdős' girth conjecture. It is tempting to believe that removing the *subgraph* requirement, and allowing to compress the graph in any arbitrary way yields sparser structures. The next lemma shows that this is not the case, demonstrating a space lower bound of $\Omega(n^{1+1/k})$ bits for any $(2k - 1)$ approximate distance structure.

Lemma 2.7 *Assuming Erdős' girth conjecture, for every $k \geq 1$ and sufficiently large n, there exists an n-vertex graph for which any $(2k - 1)$ approximate distance oracle has size $\Omega(n^{1+1/k})$.*

Proof Sketch: Let G be an n-vertex graph with girth at least $2k + 2$ and $\Omega(n^{1+1/k})$ edges (such a graph exists by the girth conjecture). We claim that every two subgraphs G_1, G_2 of G must have different oracles. Since G has $2^{|E(G)|=n^{1+1/k}}$ subgraphs, this would imply the claim. Without loss of generality, let (u, v) be an edge in $G_1 \setminus G_2$. We will show that a $(2k-1)$ approximate oracle $\mathcal{O}$ for G_1 cannot be a $(2k-1)$ approximate oracle for G_2. Since $(u, v) \in G_1$, we have: $1 \leq \widehat{\delta}(u, v, \mathcal{O}) \leq (2k - 1)$. On the other hand, recalling that G has girth at least $2k + 2$, we have: $\mathtt{dist}(u, v, G_2) \geq \mathtt{dist}(u, v, G \setminus \{(u, v)\}) \geq 2k + 1 > \widehat{\delta}(u, v, \mathcal{O})$.

References

[TZ05] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM (JACM)*, 52(1):1–24, 2005.