Randomized Algorithms 2015A
Lecture 6 – Distance Oracles and Distance Labeling*

Robert Krauthgamer

1 Distance Oracles

Goal: Preprocess a graph $G = (V, E)$ with edge lengths $l : E \to \mathbb{R}_+$ into a (small) data structure that can answer in time $O(1)$ queries about the distance $d = d_G$ (between any two vertices $u, v \in V$).

We denote $n = |V|$ and $m = |E|$.

Naive solution: Store all $\binom{n}{2}$ distances in a matrix/array, with direct access in time $O(1)$.

Can one "compress" the information, perhaps at the expense of accuracy, i.e., the distances are only approximated?

Theorem 1 [Thorup-Zwick, 2001]: There is an algorithm that preprocesses an integer $k \geq 2$ and a graph G in expected time $O(kmn^{1/k})$ and produces a data structure of expected size $O(kn^{1+1/k})$ words that can be used to answer in time $O(k)$ distance queries with approximation factor $2k - 1$.

Remark: We will ignore the preprocessing time, and focus on storage (space). In particular, we assume the shortest path between every two vertices is computed, and essentially use only the fact that distances satisfy the triangle inequality (i.e., it holds for every n-point metric space).

Algorithm Prep(G,k):

1. $A_0 = V$; $A_k = \emptyset$.
2. for $i = 1, \ldots, k - 1$
3. Construct A_i by including each $u \in A_{i-1}$ independently with probability $1/n^{1/k}$.
4. for every $v \in V$
5. for $i = 0, \ldots, k - 1$
6. store $d(v, A_i) = \min\{d(v, w) : w \in A_i\}$ and the minimizer w as $p_i(v)$
7. set $d(v, A_k) = \infty$.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

8. store $B(v) = \cup_{i=0}^{k-1}\{w \in A_i \setminus A_{i+1} : \ d(v,w) < d(v,A_{i+1})\}$ in a hash table that answers whether $w \overset{?}{\in} B(v)$ and if so, what is its distance to v, in $O(1)$ worst-case time.

Remark: We can use cuckoo hashing with storage $O(|B(v)|)$.

Intuition of preprocessing:

The sets A_i are subsamples of V at different "levels", and provide "landmarks".

Each "pivot" $p_i(v)$ is just the level i landmark closest to v.

What is the "bunch" set $B(v)$? sort V by distance from v, and partition it into k levels (rings) at positions $n^{1/k}, n^{2/k}, \ldots$; store $n^{1/k}$ random vertices from each ring.

Analysis of preprocessing storage:

The main concern is $\sum_v |B(v)|$, as the pivots can be stored in $O(kn)$ space. The total space bound $O(kn^{1+1/k})$ follows from the next lemma.

Lemma 2: For every $v \in V$, we have $\mathbb{E}[|B(v)|] \leq O(kn^{1/k})$.

The proof was seen in class.

Algorithm Query(u,v):

1. $i = 0$; $w = u$ // throughout $w = p_i(u)$
2. while $w \notin B(v)$
3. $\quad i = i + 1$
4. $\quad (u, v) = (v, u)$ // swap
5. $\quad w = p_i(u)$
6. return $d(u, w) + d(w, v)$

The runtime is obviously $O(k)$.

Analysis of query algorithm: The entire $A_{k-1} \subseteq B(v)$, hence some answer is always returned, and the number of $u - v$ swaps is at most $k - 1$.

Lemma 3: Each swap of u, v increases $d(w, u)$ by at most $\Delta = d(u, v)$.

The proof was seen in class.

The lemma implies the approximation factor (strech bound), since we start with $d(w, u) = 0$, and at the final i we have $d(w, u) \leq i \cdot \Delta \leq (k-1)\Delta$, and thus by triangle inequality $d(w, v) \leq d(w, u) + d(u, v) \leq k\Delta$.

Exer: Analyze the following construction for stretch 3 ($k = 2$), and show that it is almost as good in terms of storage (without good query time). Explain whether the storage is worst-case, in expectation, or with high probability, and similarly for the accuracy of the querying algorithm.

Preprocess(G): Choose $L \subset V$ as a random set of $l = \sqrt{n}\,\mathrm{polylog}\, n$ "landmark" vertices (with or without repetitions). For every vertex $v \in V$, store its distances (i) to the $\sqrt{n}$ vertices closest to it (denoted $B_v \subset V$); and (ii) to all the landmark vertices.

Query(u,v): If $u \in B_v$, i.e., u is among the $\sqrt{n}$ closest to v, report the distance. Otherwise, report $\min_{w \in L}[d(u,w) + d(w,v)]$.

Hint: in the "otherwise" case, show that $L \cap B_v \neq \emptyset$.

2 Distance labeling via embedding into ℓ_∞

Goal: Preprocess a graph $G = (V, E)$ with edge lengths $l : E \to \mathbb{R}_+$ to create a (small) label for each vertex, so that the distance $d = d_G$ (between any two vertices $u, v \in V$) can be computed from their labels.

Remark: We actually require that the evaluation algorithm does not depend on G, i.e., a single algorithm for the entire family of graphs of size n.

Frechet embedding: An embedding (map) $f : V \to R^s$ where each coordinate f_i is defined as $f_i : x \to d(x, A_i)$ for some subset $A_i \subseteq V$, where by definition $d(x, A) = \min\{d(x, a) : \ a \in A\}$.

Fact 4: Each coordinate f_i is 1-Lipschitz (nonexpansive), i.e.,

$$|f_i(x) - f_i(y)| \leq d(x, y) \qquad \forall x, y \in V.$$

Proposition 5: Every n-point metric space embeds isometrically into ℓ_∞^n, and thus G admits an exact distance labeling with label-size $O(n)$ words.

Proof: Consider a Frechet embedding with n singleton sets $A_x = \{x\}$. By the above fact, $\|f(x) - f(y)\|_\infty \leq d(x, y)$. For the opposite direction, for every pair $x, y \in X$, we can look at coordinate f_x and get $\|f(x) - f(y)\|_\infty \geq |f_x(x) - f_x(y)| = d(x, y)$.

Question: Can we reduce the dimension? If we allow distortion?

Theorem 6 [Matousek 1996, based on Bourgain 1985]: For every integer $k \geq 2$, every n-points metric space (X, d) embeds with distortion $2k - 1$ into ℓ_∞^s where $s = O(kn^{1/k} \log n)$. This implies a distance labeling with approximation $2k - 1$ and label size $s = O(kn^{1/k} \log n)$.

Proof of Theorem 6: We employ a Frechet embedding whose sets are constructed at random, as follows. Let $q = 1/n^{1/k}$. For each $i = 1, \ldots, k$, construct at random a "group" of $m = \frac{24}{p} \ln n$ sets $A_{i,1}, \ldots, A_{i,m}$ that include every point in V independently with probability $q_j = \min\{1/2, q^i\} = \min\{1/2, 1/n^{i/k}\}$.

Lemma 7 (sketch): For every $x, y \in V$ there exists i such that with probability $\geq p/12$,

$$|d(x, A_{i,1}) - d(y, A_{i,1})| \geq \tfrac{1}{2q-1} d(x, y).$$

Using the lemma to finish the proof: For every $x, y \in X$, the probability that all the m random sets fail this event (from group j as above, using the lemma) is at most

$$(1 - p/12)^m < e^{-(p/12)\cdot(24/p)\ln n} = 1/n^2.$$

Finally, apply union bound over the $\binom{n}{2}$ pairs of points.

Proof of Lemma 7: Set $\Delta = \frac{1}{2k-1}d(x,y)$. Define the following sequence of balls: Let $B_0 = \{x\}$, let B_1 be a (closed) Δ-ball around y, let B_2 be a 2Δ-ball around x and continue this way (with alternating centers) until B_k. Observe that the last two radii add up to $(k-1)\Delta + k\Delta = d(x,y)$.

We claim it is possible to find indices $i \geq 1$ and t such that

$$|B_t| \geq n^{(i-1)/q} \quad \text{and} \quad |B_{t+1}| \leq n^{i/q}.$$

Assume for now the claim is true. Let E_1 be the event that $A_{i,1}$ contains a point from B_t, and E_2 the event that $A_{i,1}$ contains NO point from B_{t+1}. Clearly,

$$\Pr[|d(x,A_{i,1}) - d(y,A_{i,1})| \geq \Delta] \geq \Pr[E_1 \cap E_2] = \Pr[E_1] \cdot \Pr[E_2],$$

because the two events are independent. It is not difficult to verify, using the claim, that $\Pr[E_2] \geq 1/4$ and $\Pr[E_1] \geq q/3$.

To prove the claim, partition the interval $[1,n]$ to k intervals $I_1, \ldots, I_k$ where $I_i = [n^{(i-1)/k}, n^{i/k}]$. If the sequence $|B_1|, \ldots, |B_k|$ is not monotone increasing, i.e., there is t such that $|B_t| > |B_{t+1}|$, and we can choose i such that $|B_t| \in I_i$. Otherwise, we have $k+1$ balls and only k intervals, hence by the pigeon hole principle there must be some $|B_t|, |B_{t+1}|$ in the same interval. QED