COMS W6998-5: Algorithms through Geometric Lens (Fall'17) Nov 20, 2017

Lecture 21 – Distance Oracles

Instructors: *Alex Andoni, Ilya Razenshteyn* Scribe: *Kumar Goutam*

1 Introduction

In today's lecture we continue the discussion of distance oracles and build another distance oracle which has much better approximation as compared to the one we saw in the last lecture. We prove various results and bounds for this distance oracle.

2 Distance Oracle

Theorem 1 (Thorup and Zwick (2005))**.** *For any $k \geq 2, k$ integer, given any unweighted graph G on n nodes and m edges, we can build a distance-oracle data structure with*

- $O(kn^{1+1/k})$ *space*
- $(2k-1)$ *approximation*
- $O(k)$ *time to answer queries of the form: given u, v output an approximation to $\delta(u, v)$*
- $O(kmn^{1/k})$ *pre-processing time*

Observation 2. *Recall that in last lecture we had constructed a distance oracle which had $O(n^{1+1/k})$ space, $O(k)$ approximation and $O(1)$ query time. Although this had a slightly better space and much better query time, the new distance oracle does improve on the approximation ratio, as it gives an exact bound of $2k-1$ whereas earlier we had $O(k)$ and the constant hidden in big-O is definitely much larger than 2.*

3 Proof of Theorem

Now we give a detailed proof of the theorem by proving each part separately. We are given an $n \times n$ matrix with the elements being $\delta(u, v) =$ distance between u and v.

3.1 Preprocessing Algorithm

Start with $A_0 = V$ and build sets $A_1, \ldots, A_k, A_{k+1} = \phi$ as follows:

- for $i = 1, \ldots, k-1$
  - $A_i = A_{i-1}$ sampled with probability $\dfrac{1}{n^{1/k}}$
- for $v \in V$

- for $i = 0, \ldots, k-1$
  - $\delta(A_i, v) = \min_{u \in A_i} \delta(u, v)$
  - $p_i(v) = \arg\min_{u \in A_i} \delta(u, v)$
- $\delta(A_k, v) = \infty$
- $B(v) = \bigcup_{i=0}^{k-1} B_i(v) = \bigcup_{i=0}^{k-1} \{w \in A_i \setminus A_{i+1} \mid \delta(v, w) < \delta(v, A_{i+1})\}$

All these branches $B_i(v)\ \forall\ i, v$ are stored in hash tables along with the the distances $\delta(v, w)$.

Lemma 3. *In expectation, space for the above is bounded by $O(kn + kn^{1+1/k})$.*

Proof. Storing all $p_i(v)$ takes $O(kn)$ space.
Next, fix A_i and sort $w \in A_i$ according to $d(v, w)$ to get $w_1, w_2, \ldots, w_j, w_{j+1}, \ldots$ such that $w_{j+1} \in A_{i+1}$. We have

$$Pr(w_j \notin A_{i+1}) = \big(1 - \frac{1}{n^{1/k}}\big)^j$$

Hence we would get

$$\mathbb{E}_{A_{i+1}}\big[|B_i(v)|\big] = \sum_{j=1}^{|A_i|} \big(1 - \frac{1}{n^{1/k}}\big)^j \leq \big(1 - \frac{1}{n^{1/k}}\big) \frac{1}{1 - 1 + 1/n^{1/k}} \leq n^{1/k}$$

So finally we get $\mathbb{E}\big[|B(v)|\big] = \sum_{i=0}^{k-1} \mathbb{E}\big[|B_i(v)|\big] \leq O(kn^{1/k})$ which gives the required bound for space. ☐

3.2 Query Algorithm

- Input: nodes u and v
- $w = u$
- $i - 0$
- while $w \notin B(v)$
  - $i = i + 1$
  - $(u, v) = (v, u)$
  - $w = p_i(u)$
- return $\delta(u, w) + \delta(w, v)$

Let $\Delta = \delta(u, v)$ and (u_i, v_i, w_i) denote the nodes (u, v, w) after i-th iteration, i.e., $(u_0, v_0, w_0) = (u, v, u)$.

Claim 4. *If iteration i finishes then $\delta(u_i, w_i) \leq i \cdot \Delta$*

Proof. We prove it by induction. For the base case, $i = 0$ and $\delta(u_0, w_0) = \delta(u, u) = 0$, so it holds. We argue that if i-th iteration finishes, then $\delta(u_i, w_i) \leq \delta(u_{i-1}, w_{i-1}) + \Delta$. Since the while loop finishes at the i-th step, we have $w_{i-1} \notin B(v_{i-1}) \implies \delta(w_{i-1}, v_{i-1}) \geq \delta(v_{i-1}, A_i) = \delta(u_i, A_i)$. Hence,

$$\begin{aligned}
\delta(u_i, w_i) &= \delta(u_i, p_i(u_i)) \\
&= \delta(v_{i-1}, p_i(v_{i-1})) \\
&= \delta(v_{i-1}, A_i) \\
&\leq \delta(v_{i-1}, w_{i-1}) \\
&\leq \delta(v_{i-1}, u_{i-1}) + \delta(u_{i-1}, w_{i-1}) && \text{(triangle inequality)} \\
&= \delta(u_{i-1}, w_{i-1}) + \Delta && \text{(finishes our argument)} \\
&\leq (i-1)\Delta + \Delta && \text{(induction hypothesis)} \\
&= i \cdot \Delta
\end{aligned}$$

□

3.3 Approximation Ratio and Query Time

Lemma 5. *Above algorithm gives an output with $2k - 1$ approximation in $O(k)$ query time.*

Proof. Since $A_k = \phi, A_{k-1} \subset B(v)\ \forall v$. Also, when the while loop finishes, we have $i \leq k - 1$. Hence,

$$p_{k-1}(u) \in B(v)\ \forall\ u, v \implies \delta(u, w) \leq (k-1)\Delta$$

Also by triangle inequality, we have

$$\delta(v, w) \leq \delta(v, u) + \delta(u, w) \leq k \cdot \Delta$$

Combining the two, we get

$$\delta(u, w) + \delta(v, w) \leq (2k - 1)\Delta$$

which give us the required approximation.

Also, since the while loop can have at most k iterations with each iteration doing constant work, we get $O(k)$ query time. □

4 An Improvement

Chechik (2014) build an oracle which has the same approximation ratio, uses the same space but runs in $O(1)$ time.

5 Optimality of Space vs Approximation Ratio

In this section, we prove that given the $O(kn^{1+1/k})$ space requirement, the approximation ratio of $2k - 1$ is nearly optimal.

Theorem 6. *Under the combinatorial Erdös conjecture, any data structure achieving an approximation ratio of $t < 2k + 1$ requires $\Omega(n^{1+1/k})$ space.*

Conjecture 7 (Erdös (1965))**.** *There exists graphs with* $\Omega(n^{1+1/k})$ *edges and girth (length of minimum cycle) at least* $2k+2$*.*

Now we prove the theorem using this conjecture.

Proof. Let

$$E = \{\text{conjecture graphs with } 2k+2 \text{ girth and } n^{1+1/k} \text{ edges}\}$$

and

$$S = \{\text{all graphs that are subsets of } E\}$$

If possible let there exist an encoding $Enc(G)$ (binary string of length $m = o(n^{1+1/k})$) from which we can reconstruct all distances in G with an approximation ratio t. Then we have

$$|S| = 2^{\Omega(n^{1+1/k})}$$

If $Enc(G) < m < \log|S|$ then this implies that the encoding cannot be an injective map and hence $\exists\ G_1, G_2 \in S, G_1 \neq G_2$, s.t., $Enc(G_1) = Enc(G_2)$. In particular, all distance queries give the same answer. So,

$$\exists\ (u,v) \in G_1, \notin G_2 \implies \delta_{G_1}(u,v) = 1$$

But on the other hand

$$\begin{aligned} \delta_{G_2}(u,v) &\geq (\text{length of cycle}) - 1 \\ &\geq 2k+2-1 = 2k+1 \end{aligned}$$

Hence the approximation ratio must be $\geq 2k+1$. □

References

S. Chechik. Approximate distance oracles with constant query time. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 654–663. ACM, 2014.

P. Erdös. On some extremal problems in graph theory. *Israel Journal of Mathematics*, 3(2):113–116, Jun 1965. ISSN 1565-8511. doi: 10.1007/BF02760037. URL `https://doi.org/10.1007/BF02760037`.

M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, Jan. 2005. ISSN 0004-5411. doi: 10.1145/1044731.1044732. URL `http://doi.acm.org/10.1145/1044731.1044732`.