15-859E: Advanced Algorithms **CMU, Spring 2015**
Lecture #6: Distance Preserving Trees January 26, 2015
Lecturer: Anupam Gupta Scribe: Titouan Rigoudy

In this lecture we will be studying the problem of finding low-stretch spanning trees in general graphs. A low-stretch spanning tree T in a graph G is a spanning tree for G with the additional constraint that the distance between any two vertices in T is at most a small constant factor times the distance between the two same vertices in G. In other words, the tree does not "stretch" distances too much.

Throughout this lecture, we will be considering a general graph $G = (V, E)$ with edge lengths $(l_e)_{e\in E}$. We can define $d_G : V \times V \to \mathbb{R}_+$ the distance function in G. That is, for all u, v in V, $d_G(u, v)$ is the length of the shortest path in G from u to v.

1 Shortest Path Trees

At best we could hope to find exact distance preserving trees, i.e.:

Definition 6.1. Let T be a spanning tree of G. T is an (all-pairs) shortest path tree in G if for all u, v in V, $d_T(u, v) = d_G(u, v)$.

For any single source u in V, it is easy to compute a single-source shortest path tree T such that for all v in V, $d_T(u, v) = d_G(u, v)$. This can be obtained by running Dijkstra's algorithm or Bellman-Ford as applicable.

Unfortunately we cannot hope to always find an all pairs shortest path tree in general graphs. Consider the clique of n nodes, K_n, with unit edge lengths. Any spanning tree T for K_n will be missing most of the edges, thus there must be nodes u, v in V such that $d_T(u, v) \geq 2$, even though $d_G(u, v) = 1$.

2 Low-Stretch Spanning Trees

This leads us to consider a relaxed definition. Say we do not require distances in T to be equal to those in G, but to be at most multiplied by a constant factor α.

Definition 6.2. Let T be a spanning tree of G, and let $\alpha \geq 1$. T is an α-stretch spanning tree of G if for all u, v in V, the following inequalities hold: $d_G(u, v) \leq d_T(u, v) \leq \alpha\, d_G(u, v)$.

Naturally we can wonder if there exists a "small" value for α such that for any graph G we can always find an α-stretch spanning tree of G. The answer to that question is also unfortunately no.

For any n, consider C_n the cycle of n nodes. Any spanning tree T in C_n is a path of $n-1$ edges, built by simply removing one edge from C_n. The distance between the two endpoints u, v of that removed edge in T is $d_T(u, v) = n - 1 = (n - 1)d_G(u, v)$. Therefore for all α, there exists graphs containing no α-stretch spanning tree.

3 Motivation

So far it seems as though we cannot give a good solution to this problem in general. But why is this problem even interesting in the first place? Well, some problems are much easier to solve on trees than on general graphs, and using low stretch spanning trees enables us to give approximate solutions in a reasonable amount of time to such problems.

One such example is the k-median problem. Given $k \geq 0$, we want to find a subset C of k vertices which minimizes $\sum_{v \in V} d_G(v, C)$.

If G is a tree, then we can compute an exact solution in polynomial time using dynamic programming. If G is a general graph, then the problem is NP-hard. However, using an α-stretch spanning tree of G we can compute an α-approximate solution for G. The algorithm is very simple:

1. Find an α-stretch spanning tree T of G.
2. Solve the k-median problem on T to get C_T.
3. Return C_T.

Claim 6.3. *C_T is an α-approximate solution to the k-median problem for G.*

Proof. Suppose C_G is an optimal solution for G, then the following inequalites hold:

$$\sum_{v \in V} d_T(v, C_G) \leq \alpha \sum_{v \in V} d_G(v, C_G) \tag{6.1}$$

$$\sum_{v \in V} d_T(v, C_T) \leq \sum_{v \in V} d_T(v, C_G) \tag{6.2}$$

$$\sum_{v \in V} d_G(v, C_T) \leq \sum_{v \in V} d_T(v, C_T) \tag{6.3}$$

Therefore we have:

$$\sum_{v \in V} d_G(v, C_T) \leq \alpha \sum_{v \in V} d_G(v, C_G) \tag{6.4}$$

□

4 Randomization to the rescue

Now that we know that this problem is an interesting one, how can go about solving it without running into the problems outlined above? If we cannot get a small deterministic value for α, let us try to get a small value for α *in expectation*.

First, we will need to amend our definition to accomodate the fact that we are no longer looking for trees in a deterministic fashion.

Definition 6.4. Let $\mathcal{D}$ be a probability distribution over spanning trees of G. $\mathcal{D}$ is said to be an α-stretch probabilistic embedding into trees (α-PET[1]) for G if for all u, v in V:

$$d_G(u, v) \leq d_T(u, v) \quad \forall\, T \text{ in the support of } D \tag{6.5}$$

$$\mathbb{E}_T[d_t(u, v)] \leq \alpha\, d_G(u, v) \tag{6.6}$$

This definition is interesting because an α-PET preserves distances "on average". For example, if one samples $p = c \log n$ trees $T_1, \ldots, T_p$ from such a distribution for some well chosen c, then for all u, v in V, with high probability, there exists $i \in \{1, \ldots, p\}$ such that $d_T(u, v) \leq \alpha\, d_G(u, v)$.

[1]This is not standard notation, do not look for it in the literature.

Remark 6.5. With this definition, we can get a small α for C_n for all n. Let D be the uniform distribution over spanning trees of C_n. Picking a tree from D is equivalent to picking an edge uniformly at random from C_n and deleting it. For all u, v in V, there is only a 1 in n chance of deleting the edge from u to v, thus we now have the following:

$$\begin{aligned} \mathbb{E}_T[d_T(u,v)] &= \frac{n-1}{n} \times 1 + \frac{1}{n}(n-1) \\ &= 2\,\frac{n-1}{n} < 2 \end{aligned}$$

Therefore D is a 2-PET for C_n.

Remark 6.6. The previous algorithm still works for the k-median algorithm, only now it is a randomized algorithm, and the bound only holds in expectation. Inequality (6.1) becomes:

$$\mathbb{E}_T\Big[\sum_{v \in V} d_T(v, C_G)\Big] \leq \alpha \sum_{v \in V} d_G(v, C_G)$$

and thus we obtain:

$$\mathbb{E}_T\Big[\sum_{v \in V} d_G(v, C_T)\Big] \leq \alpha \sum_{v \in V} d_G(v, C_G)$$

We can now prove interesting results using this notion of embeddings into trees.

Theorem 6.7. *[?] For any graph G, there exists $\alpha_{AN} \in \mathbb{R}$ and $\mathcal{D}_{AN}$ an α_{AN}-PET of G such that:*

- *We can sample trees from $\mathcal{D}_{AN}$ in $O(m \log n \log\log n)$ time.*
- *$\alpha_{AN} = O(\log n \log\log n)$.*

Theorem 6.8. *[?] For infinitely many n, there exists graphs G on n vertices such that any α-PET $\mathcal{D}$ on G must verify $\alpha = \Omega(\log n)$. In fact, G can be taken to be the n-vertex square grid, or the n-vertex hypercube.*

Given the Alon et al. lower bound, the Abraham-Neiman result has an α which is within an $O(\log\log n)$ factor of the best possible.

Today we will restrict ourselves to "metric graphs" and prove a looser bound.

Definition 6.9. A metric graph G is a complete graph such that edge lengths satisfy the triangular inequality. That is, for all u, v, w in V, we have $l_{(u,v)} \leq l_{(u,w)} + l_{(w,v)}$.

Theorem 6.10. *[?] Let G be a metric graph, and let $d_{min}(G) = \min\{d_G(u,v) \mid u, v \in V\}$, $d_{max}(G) = \max\{d_G(u,v) \mid u, v \in V\}$. There exists an α_B-PET $\mathcal{D}_B$ which we can sample from, such that:*

$$\alpha_B = O(\log n \log \frac{d_{max}(G)}{d_{min}(G)})$$

We now need to define an additional notion:

Definition 6.11. Given a metric graph $G = (V, E)$ and a bound $D > 0$, a low diameter decomposition scheme (or LDD scheme[2]) is a randomized algorithm which partitions V into $V_1, \ldots,_t$ such that:

[2]This isn't a standard notation either.

- For all i in {1,...,t}, for all u, v in V_i, $d_G(u,v) \leq D$.
- For all u, v in V, $\quad \mathbb{P}[u, v \text{ in different clusters}] \leq \frac{d_G(u,v)}{D}\beta \quad$ for some β.

We will assume the following lemma:

Lemma 6.12. *There exists an LDD scheme with $\beta = O(\log n)$.*

Using this lemma, we can define the following algorithm to sample from Bartal's α_B-PET:

Algorithm: *Bartal*$(G, 2^i)$: // $d_{max}(G) < 2^i$

- $G_1, \ldots, G_t \leftarrow LDD_\beta(G)$
- For every j in $\{1, \ldots, t\}$, recursively apply the algorithm: $T_j \leftarrow Bartal(G_j, 2^{i-1})$
- Add edges from the root of T_1 to the roots of $T_2, \ldots, T_t$
- Return the resulting tree rooted in the root of T_1.

Now we can prove theorem 6.10.

Proof. Without loss of generality, we will assume that $d_{min}(G) = 1$ and $d_{max}(G) = \Delta$. We want to show $\alpha = O(\log n \log \Delta)$.

Let $i = \lfloor \log \Delta \rfloor + 1$, and let $T = Bartal(G, 2^i)$.

Claim: $\mathbb{E}_T[d_T(u,v)] = 8i\beta d_G(u,v)$ for all u, v in V.

Let's prove the claim by induction on i. Let $T_1, \ldots, T_t$ be the result of running the algorithm recursively on $G_1, \ldots, G_t$, a β LDD of G. Let u, v be two vertices in V, let a, b such that u is in T_a and v is in T_b, and let r_a, r_b be the roots of T_a, T_b. Then we have:

$$\mathbb{E}_T[d_T(u,v)] = \mathbb{E}_T[d_T(u,v) \mid a \neq b]\, \mathbb{P}[a \neq b] + \mathbb{E}_T[d_T(u,v) \mid a = b]\, \mathbb{P}[a = b]$$

However we also know:

$$\begin{aligned}
\mathbb{E}_T[d_T(u,v) \mid a \neq b] &= \mathbb{E}_T[d_{T_a}(u, r_a) + d_G(r_a, r_1) + d_G(r_b, r_1) + d_{T_b}(r_b, v)] \\
\mathbb{P}[a \neq b] &\leq \frac{\beta\, d_G(u,v)}{2^{i-1}} \\
\mathbb{E}_T[d_T(u,v) \mid a = b] &= 8(i-1)\beta d_G(u,v) \\
\mathbb{P}[a = b] &\leq 1
\end{aligned}$$

Furthermore, we know that $d_T(r_a, r_1) \leq 2^i$ and $d_T(r_1, r_b) \leq 2^i$. Recursively, since the path from u to r_a is made of edges from roots to roots:

$$d_{T_a}(u, r_a) \leq 1 + 2 + \cdots + 2^{i-1} \leq 2^i$$

Identically, $d_{T_b}(r_b, v) \leq 2^i$. Thus:

$$\mathbb{E}_T[d_T(u,v) \mid a \neq b] \leq 2^i + 2^i + 2^i + 2^i = 2^{i+2}$$

Which finally gives us:

$$\begin{aligned}\mathbb{E}_T[d_T(u,v)] &\leq 2^{i+2}\,\frac{\beta\, d_G(u,v)}{2^{i-1}} + 8(i-1)\beta\, d_G(u,v)\\ &\leq 8(1+i-1)\beta\, d_G(u,v)\\ &\leq 8i\,\beta\, d_G(u,v)\end{aligned}$$

Thus $\alpha_B = O(i\beta) = O(\log\Delta\log n)$. □

And finally, we can give the LDD scheme that proves lemma 6.12.

Algorithm: $LDD(G, D)$:

- Pick any unmarked vertex v
- Sample R_v from the geometric distribution $Geom(p = \min(1, \frac{4\log n}{D}))$
- Mark all unmarked vertices w such that $d_G(v,w) \leq R_v$ as belonging to v's cluster G_v.
- If there exists a $R_v > D/2$, repeat.

Proof. By construction, each cluster's diameter will be $\leq D$.

The probability that one particular $R_v > D/2$ is:

$$\mathbb{P}[R_v > D/2] = (1-p)^{D/2} \leq e^{-pD/2} \leq e^{-2\log n} \leq \frac{1}{n^2}$$

Therefore by union bound the probability that all clusters have diameter less than D, i.e. $R_v \leq D/2$, is:

$$\begin{aligned}\mathbb{P}[\forall v \in V\ d_{max}(G_v) \leq D] &= 1 - \mathbb{P}[\exists v \in V \mid R_v > D/2]\\ &\geq 1 - \frac{n}{n^2}\\ &\geq 1 - \frac{1}{n}\end{aligned}$$

So this algorithm will loop $O(\log n)$ times in expectation.

But why is $\mathbb{P}[u, v \text{ separated}] \leq \frac{\beta\, d_G(u,v)}{D}$ for all u, v in V?

Sampling from the geometric distribution is like repeatedly flipping a coin and counting the number of heads we get before the first tails. This process is memory-less, meaning that if we have already had N heads, the probability that we will get one more head is still p.

If the ball centered in w has already reached u but not v, i.e. if $R_w = d_G(w,u) < d_G(w,v)$, then we will flip a coin at most $d_G(u,v)$ times to know if we should stop growing R_w before the ball reaches u. The probability that R_w will not "make it" to $d_G(w,v)$ is thus, by union bound:

$$\mathbb{P}[u, v \text{ separated}] \leq d_G(u,v)\, p \leq \frac{4\log n\, d_G(u,v)}{D}$$

Therefore this LDD scheme gives us $\beta = O(\log n)$. □