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

Lecture 6: May 17

Lecturer: Merav Parter

Distance Preserving Trees

Given a graph $G = (V, E)$, we would like to compute a low-stretch tree $T \subseteq G$ such that $\texttt{dist}(u, v, T) \leq \alpha \cdot \texttt{dist}(u, v, G)$ for every $u, v \in V$. By considering the cycle graph C, it is easy to see that any spanning tree of C increases the distance of a single edge endpoint from 1 to $n-1$, hence, in general, $\alpha = \Omega(n)$. Raz and Rabinovich [RR98] showed that this lower bound still holds even if we relax the condition of T being a subgraph of G, and allow it to be an arbitrary weighted tree containing the vertices of G.

To bypass this impasse, we will use randomization! instead of finding one small stretch tree, we will compute a *distribution* over trees. This class is devoted for trees that are not subgraphs of G. The next class considers the more restricted case where the trees are required to be subgraphs of G.

Definition 6.1 (Probabilistic Tree Embedding) *For a given graph G, a probability distribution $\mathcal{D}$ over trees $\mathcal{T} = \{T_1, \ldots, T_k\}$ is an α-probabilistic tree embedding if: (1) $V(G) \subseteq V(T_i)$, (2) $\texttt{dist}_G(u, v) \leq \texttt{dist}_{T_i}(u, v)$ for every $T_i \in \mathcal{T}$ and (3) $\mathbb{E}_{T \sim \mathcal{D}} \texttt{dist}_T(u, v) \leq \alpha \cdot \texttt{dist}_G(u, v)$.*

Example: Consider again the n-cycle graph C and let $\mathcal{D}$ be the uniform distribution over all spanning trees of C. Sampling from such a distribution is equivalent for picking one edge e in C uniformly at random, and returning the tree $C \setminus \{e\}$. We then have for an edge $(u, v) \in C$ that $\mathbb{E}_{T \sim \mathcal{D}} \texttt{dist}_T(u, v) = 1/n \cdot (n-1) + (1 - 1/n) = 2 - 2/n$. This is because with probability of $1/n$ the edge $e = (u, v)$ is removed, and in such a case the stretch is $n-1$ and with probability $(1 - 1/n)$, the edge e is in the tree and hence has a stretch of 1. This yields a 2-probabilistic tree embedding for cycles, however, for general n-vertex graphs, there is a lower bound of $\alpha = \Omega(\log n)$.

The Main Theorem: For every graph G, there is an α-probabilistic tree embedding with $\alpha = O(\log n)$. To put it differently, there exists a randomized algorithm that constructs a weighted tree $T = (V', E')$ with $V(G) \subseteq V'$ such that $\mathbb{E}(\texttt{dist}_T(u, v)) = \alpha \cdot \texttt{dist}_G(u, v)$ for every $u, v \in V$, where the expectation is over the randomness of the randomized algorithm.

As a corollary, we also get that by sampling $O(\log n)$ trees from such a distribution, for every u, v w.h.p., there is a tree $T_{u,v}$ in that sample, such that $\texttt{dist}_T(u, v) \leq \alpha \cdot \texttt{dist}_G(u, v)$.

The construction that we present is based on [Bar98] and [FRT04] and it is based on an extremely useful tool that decomposes an input graph into low-diameter components with few inter-cluster edges. We first provide some definitions. For a subgraph $G' \subseteq G$, the *weak-diameter* of G' is given by $\max_{u,v \in G'} \texttt{dist}_G(u, v)$. The *strong-diameter* of $G' \subseteq G$ is $\max_{u,v \in G'} \texttt{dist}_{G'}(u, v)$. Let $\mathsf{Ball}_G(u, k)$ be all vertices at distance at most k from v in G.

Low Diameter Decomposition (Ball Carving)

Given a graph $G = (V, E)$ and a parameter D, a *low-diameter decomposition* is a randomized partitioning of the vertices into $V = V_1 \cup V_2 \cup \ldots \cup V_t$ such that:

(1) For every $i \in \{1, \ldots t\}$, $\forall u, v \in V_i$, $\texttt{dist}_G(u, v) \leq D$, i.e., the weak diameter of $G[V_i]$ is bounded by D.

(2) For all $u, v \in V$, $\Pr[u \in V_i \text{ and } v \in V_{j \neq i}] \leq O(\log n / D) \cdot \texttt{dist}_G(u, v)$.

Fig. 6.1 describes the decomposition algorithm. The idea is to grow carved balls around vertices where the radius of the balls is chosen from the geometric distribution with parameter $p = 4 \log n / D$. Recall that sampling from a geometric distribution $\mathsf{Geom}(p)$ is equivalent for flipping a coin that comes head with probability p and counting the number of flips until the first head is observed. Another pictorial way to simulate the selection of R_v is as follows. Imagine we set a counter of $R_v = 1$ and flip a coin with probability p, if it comes head, we stop and cut all edges at distance R_v from v. Otherwise, we increase R_v by one, and repeat.

Algorithm $\mathsf{LDD}(G, D)$

1. Set $j \leftarrow 1$ and initialize all vertices to be unmarked.
2. While there are unmarked vertices do:
   - Pick an unmarked vertex v.
   - Sample $R_v \sim \mathsf{Geom}(p)$ for $p = \min\{1, 4\log n/D\}$.
   - Add all unmarked vertices in $\mathsf{Ball}_G(v, R_v)$ to V_j.

Figure 6.1: Algorithm for computing low diameter decomposition

We start by showing that the diameter of each component is at most D w.h.p. We note that there is an alternative partitioning in which the diameter is at most D with probability 1 (for the curios reader, see [CKR05]).

Claim 6.2 *W.h.p.,* $\mathtt{dist}(u, v, G) \leq D$ *for every* $u, v \in V_i$.

Proof: We will show that w.h.p. $R_v \leq D/2$ and by applying the union bound over all n vertices, the claim will hold. Since R_v is sampled from $\mathsf{Geom}(p)$, we have that $\Pr[R_v \geq D/2] = (1-p)^{D/2} \leq \exp(-pD/2) \leq 1/n^2$. ■

Claim 6.3 $\Pr[u \in V_i$ *and* $v \in V_{j \neq i}] = O(\log n/D) \cdot \mathtt{dist}_G(u, v)$.

Proof: For simplicity, consider first the case where u and v are neighbors. Without loss of generality let u be the vertex that becomes clustered not after v (i.e., u belongs to V_i and v to V_j for $j \geq i$). Let z be the center of the cluster of V_i. By the fact that $u \in V_i$, we know that $R_z \geq \mathtt{dist}_G(u, z)$. Hence, the only information that we currently have on the coin flips of z, is that the first $\mathtt{dist}_G(u, z)$ coin flips came up tail. The vertex v does not belong to V_i only if the next coin flip comes up head, and this happens with probability p. More formally, we have that

$$\Pr[u \in V_i \text{ and } v \in V_{j \neq i}] = \Pr[R_v < d + 1 \mid R_v \geq d] = p,$$

where $d = \mathtt{dist}_G(z, u)$. This is naturally extended for any (non-neighboring) pair u, v as follows. By applying the union bound, v is not in V_i if one of the $\mathtt{dist}_G(z, v) - \mathtt{dist}_G(z, u)$ future coin flips came up head. Or more formally,

$$\begin{aligned} \Pr[u \in V_i \text{ and } v \in V_{j \neq i}] &= \Pr[R_v \in [\mathtt{dist}_G(z, u), \mathtt{dist}_G(z, v) - 1] \mid R_v \geq \mathtt{dist}_G(z, u)] \\ &\leq p \cdot (\mathtt{dist}_G(z, v) - \mathtt{dist}_G(z, u)) \leq p \cdot \mathtt{dist}_G(u, v). \end{aligned}$$

■

Constructing the Low Distortion Tree Embedding

We will present a construction with stretch value $\alpha = O(\log n \cdot \log \mathtt{Diam}(G))$. The tight result of $\alpha = O(\log n)$ is described in [FRT04]. The tree embedding algorithm is recursive. First, it applies the low-diameter decomposition on the graph G with parameter $D/2$ where $D = \mathtt{Diam}(G)$. This results in components $G_i \subseteq G$ with weak diameter at most $D/2$. A rooted tree T_i is then constructed recursively on each G_i. The final tree T is given by connecting all the roots of all the T_i trees to a new root node r_0 and setting the weight of these edges to D.

Alternatively, the algorithm can also be described as an iterative algorithm[1] with $O(\log D)$ iterations. In iteration $i \geq 1$, we are given a collection of vertex disjoint components, each of weak diameter at most $D/2^{i-1}$. Layer i of T has a vertex for each such component. For $i = 1$, we have a single node, namely the root vertex, which corresponds to G. Then in iteration i, the low-diameter decomposition is applied for each of these layer-i components. We include in layer $i+1$, a vertex for each of the output components. Each layer i component (i.e., the vertex corresponding to that component) is connected in the tree to its child components (the output of applying the LDD algorithm on it) with weight $D/2^{i-1}$.

Note that the vertices of G are the leaf nodes of the output tree T. Also T has depth $O(\log \mathtt{Diam}(G))$, since in each recursive level, the weak diameter is cut by half. See Fig. 6.2 for a complete description. Note that since the tree is defined based on the randomized low-diameter decomposition, its structure is randomized as well.

Algorithm $\mathsf{TE}(G', D')$

1. If $V(G') = \{v\}$, return v.
2. Call $\mathsf{LDD}(G', D'/2)$ resulting in $G'_1 = G[V_1], \ldots, G'_\ell = G'[V_\ell]$.
3. For every G'_i, let $T_i \leftarrow TE(G'_i, D'/2)$.
4. Introduce a root r_0 and connect it in T to each $r_1, \ldots, r_\ell$ with weight D' where r_i is the root of T_i.
5. Return T.

Figure 6.2: Algorithm for computing a randomized low-distortion tree $T = (V', E)$. The input of the algorithm is a subgraph $G' \subseteq G$ and D' where D' is an upper bound on the weak diameter of G' (with respect to G). The final tree $T = \mathsf{TE}(G, \mathtt{Diam}(G))$.

We next turn to analyze the algorithm. We say that u and v are *separated* in level i, if u and v belong to the same component G' in level i but to different components in level $i+1$, i.e., u and v are separated when applying the low-diameter decomposition on a component G' in level i.

Claim 6.4 *W.h.p* $\mathtt{dist}_T(u, v) \geq \mathtt{dist}_G(u, v)$ *for every* u, v.

Proof: If u and v are separated in level 1, then $\mathtt{dist}_T(u, v) = 2D > \mathtt{dist}_G(u, v)$, this holds since the root is connected to its children in T with edge weight of D and u and v belong to different subtrees.

Suppose now that u and v are separated in level $i \geq 1$. Each level-i component G' has weak diameter at most $D/2^{i-1}$. Hence, $\mathtt{dist}_G(u, v) \leq D/2^{i-1}$. Since the vertex of level-i component (in the tree T) is connected to its children in T with weight $D/2^{i-1}$, we have that $\mathtt{dist}_T(u, v) \geq D/2^i$. ■

Claim 6.5 *For every* u, v, $\mathbb{E}(\mathtt{dist}_T(u, v)) = O(\log n \cdot \log \mathtt{Diam}(G)) \cdot \mathtt{dist}_G(u, v)$.

Proof: Fix a pair u, v. We use the fact that if u and v are separated in level $i \geq 1$, their distance in T is at most $4D/2^{i-1}$. To see this, consider the case where u and v are separated in level 1, then the path from root to leaf has a total weight of $D + D/2 + D/4 + \ldots + D/2^{k-1} \leq 2D$. Similarly, the path from node at level i to the leaf has total weight of $D/2^{i-1} + D/2^i + \ldots + D/2^{k-1} \leq 2D/2^{i-1}$. Let A_i denote the event

[1] This is for the recursion haters :-)

where u and v are separated level i, we then have that:

$$\begin{aligned}\mathbb{E}(\texttt{dist}_T(u,v)) &= \Pr[A_1]\cdot 4D + \Pr\big[A_2 \mid \bar{A}_1\big]\cdot 4D/2 + \ldots \\ &+ \Pr\big[A_i \mid \bar{A}_1 \wedge \bar{A}_2 \wedge \ldots \wedge \bar{A}_{i-1}\big]\cdot 4D/2^{i-1} + \ldots \\ &+ \Pr\big[A_k \mid \bar{A}_1 \wedge \bar{A}_2 \wedge \ldots \wedge \bar{A}_{k-1}\big]\cdot 4D/2^{k-1}, \quad \text{where} \quad k = \lceil \log \texttt{Diam}(G) \rceil.\end{aligned}$$

By the properties of the low-diameter decomposition, the probability that u and v are separated in level i (conditioned on the fact that they belong to the same level i component) is bounded by $O(2^i \log n/D)\cdot \texttt{dist}_G(u,v)$ for $D = \texttt{Diam}(G)$. Hence plugging in the above equation that $\Pr\big[A_i \mid \bar{A}_1 \wedge \bar{A}_2 \wedge \ldots \wedge \bar{A}_{i-1}\big] = O(2^i \log n/D)\cdot \texttt{dist}_G(u,v)$, we get $k = O(\log \texttt{Diam}(G))$ terms, each of them is bounded by $O(\log n)$, giving a total stretch of $O(k \log n)$. The claim follows. ■

References

[Bar98] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 161–168. ACM, 1998.

[CKR05] Gruia Calinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. *SIAM Journal on Computing*, 34(2):358–372, 2005.

[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[RR98] Yuri Rabinovich and Ran Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *Discrete & Computational Geometry*, 19(1):79–94, 1998.