Distance Preserving Trees

Given a graph $G = (V, E)$, we would like to compute a low-stretch tree $T \subseteq G$ such that $\text{dist}(u, v, T) \leq \alpha \cdot \text{dist}(u, v, G)$ 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 to 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 $D$ over trees $T = \{T_1, \ldots, T_k\}$ is an $\alpha$-probabilistic tree embedding if: (1) $V(G) \subseteq V(T_i)$, (2) $\text{dist}_G(u, v) \leq \text{dist}_{T_i}(u, v)$ for every $T_i \in T$ and (3) $\mathbb{E}_{T \sim D} \text{dist}_T(u, v) \leq \alpha \cdot \text{dist}_G(u, v)$.

**Example:** Consider again the $n$-cycle graph $C$ and let $D$ 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 $\mathbb{E}_{T \sim D} \text{dist}_T(u, v) = 1/n \cdot (n - 1) + (1 - 1/n) = 2 - 2/n$. 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 $\alpha$-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 $\mathbb{E} \{\text{dist}_T(u, v)\} = \alpha \cdot \text{dist}_G(u, v)$ for every $u, v \in V$, where the expectation is over the randomness of the randomized algorithm.

As a corollary, we also get that 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 $\text{dist}_{T_{u, v}}(u, v) \leq \alpha \cdot \text{dist}_G(u, v)$.

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 $\max_{u, v \in G'} \text{dist}_{G'}(u, v)$. The strong-diameter of $G' \subseteq G$ is $\max_{u, v \in G'} \text{dist}_{G'}(u, v)$. Let $\text{Ball}_G(u, k)$ 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 $i \in \{1, \ldots, t\}$, $\forall u, v \in V_i$, $\text{dist}_G(u, v) \leq D$, i.e., the weak diameter of $G[V_i]$ is bounded by $D$.
2. For all $u, v \in V$, $\Pr[u \in V_i \text{ and } v \in V_j \neq i] \leq O(\log n/D) \cdot \text{dist}_G(u, v)$.

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 $\text{Geom}(p)$ 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 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 \( R_v \sim \text{Geom}(p) \) for \( p = \min\{1, 4 \log n/D\} \).
   - Add all unmarked vertices in \( \text{Ball}_G(v, R_v) \) to \( V_j \).

![Figure 6.1: Algorithm for computing low diameter decomposition](image)

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 curious reader, see [CKR05]).

**Claim 6.2** W.h.p., \( \text{dist}(u, v, G) \leq D \) 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 \( \text{Geom}(p) \), we have that \( \Pr[R_v \geq D/2] = (1-p)^{D/2} \leq \exp(-pD/2) \leq 1/n^2 \).

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

**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 \( R_z \geq \text{dist}_G(u, z) \). Hence, the only information that we currently have on the coin flips of \( z \), is that the first \( \text{dist}_G(u, z) \) 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 \( d = \text{dist}_G(z, u) \). 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 \( \text{dist}_G(z, v) - \text{dist}_G(z, u) \) future coin flips came up head. Or more formally,

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

**Constructing the Low Distortion Tree Embedding**

We will present a construction with stretch value \( \alpha = O(\log n \cdot \log \text{Diam}(G)) \). 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 \( D = \text{Diam}(G) \). 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
Claim 6.4 \(W.h.p\) applying the low-diameter decomposition on a component the same component \(G\) with respect 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 \(O(\log \Diam(G))\), 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 TE**\((G', D')\)

1. If \(V(G') = \{v\}\), return \(v\).
2. Call \(\text{LDD}(G', D'/2)\) resulting in \(G'_1 = G[\ell_1], \ldots, G'_\ell = G[\ell_\ell]\).
3. For every \(G'_i\), let \(T_i \leftarrow \text{TE}(G'_i, D'/2)\).
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](image)

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** \(W.h.p\) \(\text{dist}_T(u, v) \geq \text{dist}_G(u, v)\) for every \(u, v\).

**Proof:** If \(u\) and \(v\) are separated in level 1, then \(\text{dist}_T(u, v) = 2D > \text{dist}_G(u, v)\), 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, \(\text{dist}_G(u, v) \leq D/2^{i-1}\). 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 \(\text{dist}_T(u, v) \geq D/2^{i}\). \(\blacksquare\)

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

**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:

$$
E(\text{dist}_T(u,v)) = \Pr[A_1] \cdot 4D + \Pr[A_2 | \bar{A}_1] \cdot 4D/2 + \ldots \\
+ \Pr[A_i | \bar{A}_1 \land \bar{A}_2 \land \ldots \land \bar{A}_{i-1}] \cdot 4D/2^{i-1} + \ldots \\
+ \Pr[A_k | \bar{A}_1 \land \bar{A}_2 \land \ldots \land \bar{A}_{k-1}] \cdot 4D/2^{k-1}, \text{ where } k = \lceil \log \text{Diam}(G) \rceil.
$$

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 $O(2^i \log n / D) \cdot \text{dist}_G(u,v)$ for $D = \text{Diam}(G)$. Hence plugging in the above equation that $\Pr[A_i | \bar{A}_1 \land \bar{A}_2 \land \ldots \land \bar{A}_{i-1}] = O(2^i \log n / D) \cdot \text{dist}_G(u,v)$, we get $k = O(\log \text{Diam}(G))$ terms, each of them is bounded by $O(\log n)$, giving a total stretch of $O(k \log n)$. The claim follows.

**References**


