Low-stretch spanning trees

The main result

In this lecture we study the topic of approximating arbitrary graph metrics by (efficiently sampleable) distributions over spanning subtrees. More formally, given a graph G, a distribution Ψ over spanning trees of G is said to be a *non-contracting distance approximation by subtrees* with distortion $D \geq 0$ if:

$$\forall\, x, y \in V(G) \qquad d_G(x,y) \leq \mathbf{E}_{T \in \Psi}\big[d_T(x,y)\big] \leq D \cdot d_G(x,y),$$

where $d_T(\cdot,\cdot)$ denotes the distance with respect to the subtree T. The main result that we prove here is:

Theorem 1. *For every unweighted undirected graph G on n vertices, there exists an efficiently computable non-contracting distance approximation by subtrees with distortion $O(\log^3 n)$.*

A modification of this result shown in [2], not presented in this lecture, produces an algorithm with distortion $O(\log^2 n \log\log n)$. The main open problem associated with this result is the existence of $o(\log^2 n \log\log n)$-distortion approximation schemes to match or at least approach the case of approximation by dominating trees and also the lower bound on distortion by dominating tree metrics which is $\Omega(\log n)$ for the n-vertex grid, see [1].

Preliminaries

Basics, disclaimer and notation: Recall that in non-contracting embeddings edges suffer the highest distortion, therefore we will focus on reducing the edge distortion.

For the rest of this lecture, we will assume that G is unweighted. At the end of the lecture, it will be clear in hindsight that all arguments hold unchanged for the weighted case.

From here on after, we use Δ_G to denote the radius of a graph G with respect to a root vertex $r \in G$ which will be clear from the context.

Intuition: For intuition, let us first imagine that we have to approximate a graph G using a deterministic approximation by subtrees. I.e. we have to choose a single subtree of minimal distortion. Consider two extreme choices of subtrees. A breadth-first search (BFS) tree T ensures that every edge $e \in E(G)\backslash E(T)$ has distortion bounded by $2\Delta_T$. Alternatively, a depth-first search (DFS) tree T' may incur an edge distortion of up to $n = |V(G)| \geq 2\Delta_T$. When G is an expander, the BFS tree has distortion $\Omega(\log n)$ while the DFS tree could have distortion $O(n)$. The key idea of [3, 2] will be to approximate G by a randomly perturbed BFS tree, where the perturbation ensures that each edge has a fair chance of being in the chosen spanning subtree tree.

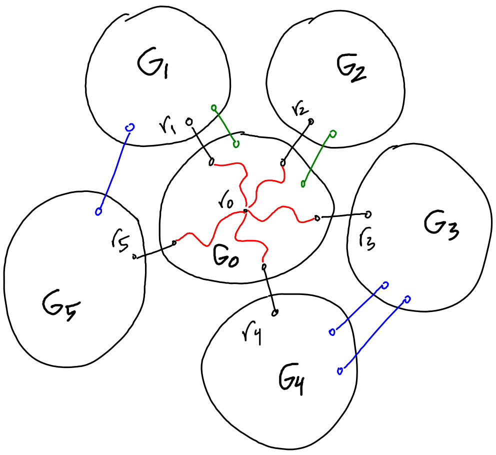
We leave it as an exercise to the reader to show that there exists a family of graphs on n vertices for which choosing a random BFS tree produces $\Omega(\sqrt{n})$ expected edge distortion. Therefore the straightforward approach is not sufficient. The construction of [2] produces subtrees that are close to BFS trees but are not BFS trees. As far as we can tell, it is an open question whether one can find approximations based on appropriately skewed distributions over BFS trees.

Exponential distribution: Let us review a few facts about the exponential distribution $X \sim \mathbf{Exp}(\lambda)$. The density function is given by $f_\lambda(x) = \lambda e^{-\lambda x}$. The cumulative distribution function is $F_\lambda(x) = \mathbf{Pr}[X \le x] = 1 - e^{-\lambda x}$. Furthermore, $\mathbf{E}[X] = 1/\lambda$ and $\mathbf{Var}[X] = 1/\lambda^2$. And most importantly, the *memoryless property* $\mathbf{Pr}[X \ge p + q \mid X \ge p] = \mathbf{Pr}[X \ge q]$.

Star decomposition

The main ingredient in the [2] construction is the *star decomposition* of graphs, which is applied recursively to produce a BFS-like subtree.

Definition 1 (Star decomposition)**.** *A star decomposition of a graph G with a designated root node r_0 is a set of disjoint connected components $G_0 = (V_0, E_0), \ldots, G_k = (V_k, E_k)$ together with a collection of root nodes $r_0, \ldots, r_k$ such that $r_i \in V_i$ for all $0 \le i \le k$ and each r_i has a neighbor in V_0.*



A star decomposition of a graph G is obtained as follows:

i. *Forward cut:* Choose a radius γ' uniformly from the interval $[\Delta_G/4, \Delta_G/2]$ and assign all vertices at distance at most γ' from r_0 to V_0. Set G_0 to be the connected subgraph of G induced by V_0 with root r_0.

ii. Consider the remaining graph $G \backslash V_0$ and replace each edge by two directed edges in opposite directions. Define a the length $\ell(u, v)$ of such a directed edge as:

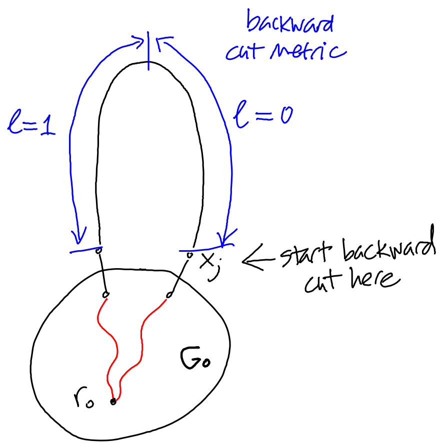
$$\ell(u,v) = \begin{cases} 1, & \text{if } d_G(r_0, v) = d_G(r_0, u) - 1 \\ 1, & \text{if } d_G(r_0, v) = d_G(r_0, u) \\ 0, & \text{if } d_G(r_0, v) = d_G(r_0, u) + 1 \end{cases}$$

The distance induced on $G \backslash V_0$ by $\ell(\cdot, \cdot)$ is called the *backward-edge distance*. Note that $\ell(\cdot, \cdot)$ is crafted so that an edge has non-zero length iff either the corresponding edge of G is not included in any BFS tree of G rooted at r_0, or otherwise it is directed towards r_0 in such a BFS.

iii. Let $x_1, \ldots, x_s$ denote the vertices in $V \backslash V_0$ which have a neighbor in V_0, called *portal nodes*.

iv. In order to select the remaining components $V_1, \ldots, V_k$, we exhaustively cut pieces from $G \backslash V_0$ in the following manner. As long as there exists a portal node x_i not yet assigned to any component, do the following:

v. *Backward cut:* Choose a random radius γ'' with distribution $\Delta_G \cdot \mathbf{Exp}(\lambda)$ for $\lambda = \Theta(\log^2 n)$ and assign all unassigned vertices of $G \backslash V_0$ at backward-edge distance γ'' from x_i to a new component V_j. The portal x_i becomes the root of the new component.



vi. Note that the backward-edge distance guarantees that whenever there is an unassigned vertex in $G \backslash V_0$, there is also a portal unassigned vertex.

Observe that if we fix $\gamma'' = 0$ the edges that are cut during the backward-cut iterations are also cut by at least one BFS tree in G rooted at r_0.

The subtree construction

The subtree of G that approximates $d_G(\cdot, \cdot)$ is constructed using the following randomized algorithm:

i. Construct a star decomposition $G_0, G_1, \ldots, G_k$ of G

ii. Let G^* be the multi-graph obtained by collapsing each G_i to a single node

iii. Select a spanning star of G^* whose edges are the edges connecting $r_1, \ldots, r_k$ to their neighbors in V_0.

iv. Recurse inside $G_1, \ldots, G_k$

v. At the end, all selected edges form a spanning tree of G

In particular, note that if $\gamma'' = 0$ in the star decomposition step, the resulting tree is exactly a BFS tree of G.

Structure Analysis

The following lemmas analyze the probability that an edge is cut by the star-decomposition:

Lemma 2. *The probability that an edge is cut by the forward cut is at most* $4/\Delta_G$*.*

Proof. Consider an edge $e = (u, v)$. If $\max\big(d_G(r_0, u), d_G(r_0, v)\big) \le \Delta_G/4$ or $\min\big(d_G(r_0, u), d_G(r_0, v)\big) \ge \Delta_G/2$ or $d_G(r_0, u) = d_G(r_0, v)$ then e is not cut. Otherwise, without loss of generality $d_G(r_0, u) = d_G(r_0, v) - 1$ and therefore e is cut with probability at most $4/\Delta_G$. ∎

Lemma 3. *The probability that an edge is cut by the backward cut is at most* $O(\log^2 n/\Delta_G)$*.*

Proof. Let $H = G \backslash V_0 \cup \cdots \cup V_{i-1}$ be the directed graph (endowed with the $\ell(\cdot, \cdot)$ edge length) over the unassigned vertices at some stage of the backward cut iteration. Also let p_{H,e,r_i} denote the probability that a fixed edge $e = (u, v) \in H$ is cut during the remaining backward cut rounds, where the portal vertex $x_i \in H$ is the root of the next round.

Assume without loss of generality that $d_\ell(x_i, u) \le d_\ell(x_i, v)$ and denote $d = d_\ell(x_i, u)$. Also note that when $\ell(u, v) = 0$ the edge e will not be cut, therefore also assume for the worst that $\ell(u, v) = 1$. Finally, set $p = \max_{H,e,x_i} p_{H,e,x_i}$, where the maximum is taken over all graphs H of at most n vertices. We can now write the following inequality:

$$p_{H,e,x_i} \le \mathbf{Pr}\left[d \le \gamma'' < d + 1\right] + \mathbf{Pr}\left[\gamma'' < d\right] \cdot \max_{H',e,x_j} p_{H',e,x_j}$$

Keep in mind that d is a function of H, e and x_i both above and in what follows below. We can now take the $\max_{H,e,x_i}$ on both sides of the inequality to get:

$$\begin{aligned} \max_{H,e,x_i} p_{H,e,x_i} &\le \max_{H,e,x_i} \left(\mathbf{Pr}\left[d \le \gamma'' < d + 1\right] + \mathbf{Pr}\left[\gamma'' < d\right] \cdot \max_{H,e,x_i} p_{H,e,x_i}\right) \quad \Leftrightarrow \\ p &\le \max_{H,e,x_i} \left(\mathbf{Pr}\left[d \le \gamma'' < d + 1\right] + \mathbf{Pr}\left[\gamma'' < d\right] \cdot p\right) \end{aligned}$$

Let H, e and x_i be such that the maximum is attained. Then get the recursive relationship:

$$p \le \mathbf{Pr}\left[d \le \gamma'' < d + 1\right] + \mathbf{Pr}\left[\gamma'' < d\right] \cdot p$$

This gives us an upper bound on p:

$$\begin{aligned} p &\le \frac{\mathbf{Pr}\left[d \le \gamma'' < d + 1\right]}{1 - \mathbf{Pr}\left[\gamma'' < d\right]} \\ &= \frac{\mathbf{Pr}\left[\gamma'' \ge d \wedge \gamma'' < d + 1\right]}{\mathbf{Pr}\left[\gamma'' \ge d\right]} \\ &= \mathbf{Pr}\left[\gamma'' < d + 1 \,|\, \gamma \ge d\right] \\ &= \mathbf{Pr}\left[\gamma < 1\right] \\ &\le 1 - e^{-\lambda/\Delta_G} \le \lambda/\Delta_G = O(\log^2 n/\Delta_G) \end{aligned}$$

The third step in the above derivation follows from the memoryless property of the exponential distribution. ∎

Note that as far as the above lemma is concerned, we could have used $\gamma'' \sim \Delta_G \cdot \mathbf{Exp}(1)$, and thus we would have obtained a smaller cut probability. This however will fail us in consequent steps of the analysis. Another interesting exercise to the reader is the following. Assume $\gamma \sim [\Delta_G/64, \Delta_G/32]$; exhibit a family of graphs for which the cut probability is almost 1.

Corollary 4. *The probability that an edge is cut is at most $O(\log^2 n/\Delta_G)$.*

The next lemma ensures that the recursion is not too deep:

Lemma 5. *With high probability the radius of component V_i is at most $\frac{7}{8}\Delta_G$.*

Proof. For a fixed component V_i and $v \in V_i$, let δ be the distance (with respect to d_G) between the root r_i of V_i and v. We will show that:

$$\mathbf{Pr}\left[\delta > \frac{7}{8}\Delta_G\right] < O\left(\frac{1}{n^{\log n}}\right)$$

This implies that with probability $1 - 1/\text{poly}\,(n)$ the above inequality holds for all components, at all levels of the recursion and all pairs r_i and v.

First we observe that:

$$\mathbf{Pr}\left[\gamma'' > \Delta_G/16\right] < \exp(-\lambda\Delta_G/16) = O(1/n^{\log n})$$

This means that all but $\Delta_G/16$ edges on the path from r_i to v increase the distance from r_0. Also recall that $d_G(r_0, r_i) \geq \Delta_G/4$. Therefore $\delta - 2\Delta_G/16 + \Delta_G/4 \leq d_G(r_0, v) \leq \Delta_G$. Which yields $\delta \leq \frac{7}{8}\Delta_G$. ∎

Note that the proof of the above lemma breaks if we had chosen $\gamma'' \sim \Delta_G \cdot \mathbf{Exp}(1)$.

Corollary 6. *The star decomposition algorithm has $O(\log n)$ levels of recursion.*

Stretch Analysis

Theorem 7. *The spanning tree computed by recursively applying the star decomposition algorithm has expected stretch $O(\log^3 n)$.*

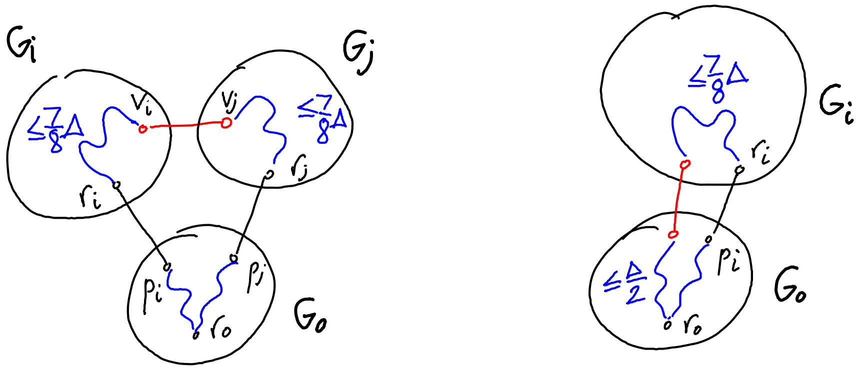
Let $e = (u, v)$ be any edge in the input graph. Recall that it is sufficient to bound the expected stretch of e.

Claim 8. *If e is cut during the star decomposition of some subgraph G with radius Δ_G, then the final stretch of e is $O(\Delta_G)$ with probability $1 - 1/\text{poly}\,(n)$.*

Note that it is sufficient to show that this only holds with high probability: Since the maximum possible stretch is n, bad events that happen with probability $1/\text{poly}\,(n)$ can contribute only a constant to the expectation. This claim immediately implies the main result:

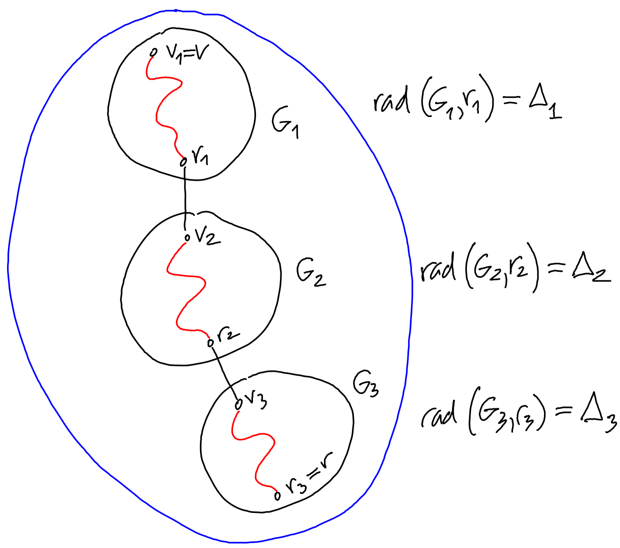
$$\begin{aligned}\mathbf{E}_{T\in\Psi}[d_T(u,v)] &= \sum_{\text{Level } G} \mathbf{Pr}\left[e \text{ cut at this level}\right] \cdot O(\Delta_G) \\ &\leq \sum_{\text{Level } G} O\Big(\frac{\log^2 n}{\Delta_G}\Big) \cdot O(\Delta_G) \\ &= O(\log^3 n)\end{aligned}$$

Proof of Claim: There are two ways in which an edge can be cut: by the forward cut, or by the backward cut. Both are illustrated below, where the edge in red is being cut:



It is easily seen that immediately after the cut the stretch of the edge is $O(\Delta)$ using the paths in blue and the bounds on the radius of the forward and backward components. We will show that every path $r-v$ between the root of a component G and some arbitrary point inside the component G incurs a constant stretch throughout the levels of the recursion. This will complete the proof the claim.

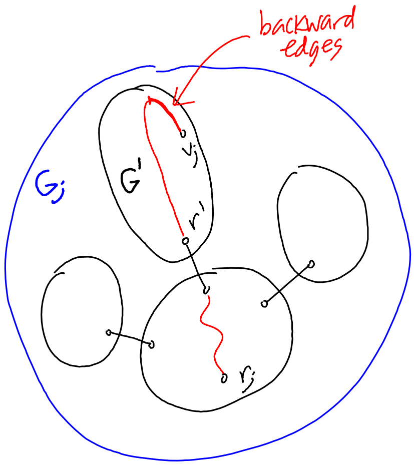
The recursive decomposition algorithm can be viewed as an iterative one with $O(\log n)$ rounds, in the following manner. After each round, the algorithm keeps a collection of connected components $\mathcal{G}$ and a set of spanning tree edges $\mathcal{T}$. The components are labeled as G_j, with respective roots r_j and radius Δ_j. Initially, $\mathcal{G} = \{G\}$ and $\mathcal{T} = \varnothing$. The distance between pairs of points after the i-th round are denoted by $d_i(\cdot, \cdot)$.



Let r be the root of the initial graph and v be some vertex in it. We are trying to show that the distance between r and v in the final tree is $O(\Delta)$ where Δ is the radius of the initial graph. After a round of the algorithm, the path between r and v is represented by the vertices $v = v_1, r_1, v_2, r_2, \ldots, v_m, r_m = r$, where (r_j, v_{j+1}) is an edge in $\mathcal{T}$ and (v_j, r_j) is taken to be the shortest path inside G_j.

First we analyze how the distance $v - r$ changes in a round. And in particular we define $\rho_{i+1}(r_j, v_j) := d_{i+1}(r_j, v_j)/d_i(r_j, v_j)$. Let us assume that the i-th round has just completed. We look at how a $v_j - r_j$ path changes. If $d_i(v_j, r_j) \leq \Delta_j/4$ the path is shorter than the forward cut, and therefore $d_{i+1}(v_j, r_j) = d_i(v_j, r_j)$, equivalently $\rho_{i+1}(r_j, v_j) = 1$.

Now suppose v_j and r_j are separated by the forward cut. Let G' be the new component that contains v_j and let its root be r'.



The distance $v_j - r_j$ increases at most by twice the number of backward edges on the path from r' to v_j. In other words, $d_{i+1}(r_j, v_j) \leq d_i(r_j, v_j) + 2\gamma''$. To this end:

Claim 9. *For every backward component G' produced by the star decomposition of some component G, in every level of the recursion, the backward radius is bounded by $\Delta_G/\log n$ with probability $1 - 1/\mathrm{poly}(n)$.*

This is verified using a union bound in combination with:

$$\mathbf{Pr}\left[\gamma'' \geq \frac{\Delta_G}{\log n}\right] \leq \exp\left(-\frac{\lambda}{\log n}\right) = \frac{1}{\mathrm{poly}(n)}$$

We thus have:

$$\begin{aligned}
\rho_{i+1}(r_j, v_j) &\leq \frac{d_i(r_j, v_j) + 2\frac{\Delta_j}{\log n}}{d_i(r_j, v_j)} \\
&= 1 + \frac{2}{d_i(r_j, v_j)} \frac{\Delta_j}{\log n} \qquad \left(\text{recall that } d_i(r_j, v_j) \geq \frac{\Delta_j}{4}\right) \\
&\leq 1 + \frac{8}{\log n} \\
&= 1 + O\left(\frac{1}{\log n}\right)
\end{aligned}$$

Using this, we can bound the stretch of an entire $r - v$ path as follows:

$$\prod_{i=1}^{O(\log n)} \rho_i = \left(1 + \frac{1}{O(\log n)}\right)^{O(\log n)} = O(1)$$

Note that in the above we implicitly use the fact that the stretch of an $r-v$ path is bounded by the stretch of its $r_j - v_j$ sub-path with maximum stretch, hence we use the shorthand ρ_i notation. ∎

References

[1] Alon, Karp, Peleg, and West. A graph-theoretic game and its application to the k-server problem. *SICOMP: SIAM Journal on Computing*, 24, 1995.

[2] Kedar Dhamdhere, Anupam Gupta, and Harald Räcke. Improved embeddings of graph metrics into random trees. In *SODA*, pages 61–69. ACM Press, 2006.

[3] Elkin, Emek, Spielman, and Teng. Lower-stretch spanning trees. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2005.