CS880: Approximation and Online Algorithms	**Scribe:** Jeremy McMahan
Lecture 12: Tree Embeddings	**Date:** 10/11/2019

12.1 Introduction

In this lecture, we discuss tree embeddings and how to use such embeddings to get good approximation algorithms. In particular, we want to take a given graph with a distance function and find a tree and a distance function so that the distance function on the tree is very similar to the distance function on the graph. This then allows us to use the tree as an easy to use approximation to the graph in algorithms. The benefits to doing this are that many problems are much easier on trees, so we can solve them on a tree approximation to G to get a good solution for G. In this lecture, we will present a simple randomized procedure that gives a good tree embedding for any graph G.

To motivate this further, let's consider a specific problem where the notion of a tree embedding could be advantageous. Recall the Buy at Bulk network design problem, where we are given a graph $G = (V, E)$, costs c_e on the edges, a concave buy at bulk function f, and k pairs of vertices (s_i, t_i). The goal is to find paths P_i from s_i to t_i that minimize $\sum_e c_e f(\ell_e)$, where $\ell_e = |\{i | e \in P_i\}|$. The most crucial aspect of this problem is that the more you use an individual edge, the more you save. Now, notice that if G is a tree, then the problem is trivial. Specifically, if G is a tree there is a unique s_i to t_i path for any i so there is nothing to solve! For graphs very close to trees the situation is similar (which is illustrated by considering K_3 with two unit weights and the other weight being 2).

12.2 Preliminaries

A *Metric* can be represented by a set with a distance function over the elements of the set, (X, d_X). An (α, β)-*distortion metric embedding* from (X, d_X) to (Y, d_Y) is a map $g : X \to Y$ satisfying

$$\forall u, v \in X, \frac{1}{\alpha} d_X(u, v) \leq d_Y(g(u), g(v)) \leq \beta d_X(u, v)$$

The distortion of g is $\alpha\beta$. For tree embeddings, we always take $\alpha = 1$ (an expanding embedding), so we only specify the distortion as β.

Theorem: If there is an embedding from G into a tree with distortion at most α, then we can obtain an α-approximation for Buy at Bulk Network Design Problem for G.

Proof:

Suppose c_T is the tree metric with low-distortion and c_G is the original graph metric. Then, by definition,

$$\forall u, v \in V, c_G(u, v) \leq c_T(u, v) \leq \alpha c_G(u, v)$$

where we just need the rightmost inequality to hold for edges of G. If we have a path $P_i* = (e_{i1}, e_{i2}, ..., e_{ik})$ that is an optimal solution to Buy at Bulk for the tree, then we can construct a nearly optimal path in the original graph $Q_i = (Q_{i1}, ...)$ where each Q_{ij} is a path connecting the endpoints of the edge e_{ij} in the tree T. In other words, we shortcut these paths to get a path in the original graph G, which is possible since $c_T(Q_{i\ell}) \geq c_G(e_{i\ell})$. We then have $c_T(Q_i) \leq \alpha c_G(P_i*)$.

■

12.3 Probabilistic Tree Embeddings

Note, if we only consider deterministic procedures for constructing tree embeddings, we will run into trouble. Specifically, consider the cycle graph, C_n. The only way to embed it into a tree is by removing one of its edges. The distance between the endpoints of the removed edge is then $n-1$, but originally they had distance 1. Thus, the distortion is at least $n-1$, which is very bad. Thus, it's natural to consider using randomness in constructing tree embeddings.

A *probabilistic tree embedding* for metric $(G = (V, E), c_G)$ with distortion α is a distribution over tree metrics, (V_T, c_T), satisfying the following:

1. For any tree T in the support of the distribution, $V_T \supseteq V$.
2. For any $u, v \in V$, for any tree T in the support of the distribution, $c_T(u, v) \leq c_G(u, v)$.
3. For any edge (u, v) in G, $E[c_T(u, v)] \leq \alpha c_G(u, v)$.

Note, saying a property holds for any tree in the support is equivalent to saying it holds with probability 1.

Theorem: For any metric (G, c_G), there is a probabilistic tree embedding for (G, c_G) with distortion $O(\log n)$.

Idea: We present a randomized algorithm that produces such an embedding, which proves its existence. First, make multiple cuts of G, so that each induced component has small diameter. The partition of all the cut sets then become the children of the root, which is the total vertex set, V. Each of the edges to the children of V are given length 2^{i-1}, where i will be the height of the tree and at each level down the tree we reduce the length further. In particular the second level will have all lengths 2^{i-2} and so on. We then recursively subdivide each cut to make the corresponding subtrees rooted at each child. If a cut is ever a single vertex, then we assume we propagate that child down as the same single vertex so that the entire tree has all the leaves, each being a single vertex of G, at the same level.

We want to make cuts having small diameter, so that we guarantee vertices that are close to each other all have a close ancestor in the tree, whereas vertices very far apart will have a high ancestor in the tree closer to the root. In particular, this ensures that the distances between vertices in the tree will not be too much larger than those in the original graph.

Note, this strategy for computing the tree is called a hierarchical decomposition. The tree constructed is a hierarchically well separated tree (HST). It's also an ultra metric, meaning that the distance from the root is the same for all leaves.

12.4 Low-Diameter Partitions

Now, we need a way to compute this sequence of cuts ensuring low-diameter pieces. To this end, we define a *low-diameter partition* with parameter $D > 0$ to be a probabilistic partition of V into sets $\mathcal{S} = \{S_1, ..., S_k\}$ such that:

1. $\forall i$, $diam(S_i) \leq D$
2. $\forall e \in E, Pr[\mathcal{S} \text{ cuts edge } e] \leq c_e/D$

Note, the second property ensures that small edges are unlikely to be cut and so vertices that are close to each other will likely be in the same set of the partition, which is exactly what we want. We can use these partitions to get an $O(\log n)$-approximation for multi-cut as well. Now, we prove the theorem from the previous section by constructing low-diamater partitions.

Proof: We will use the use the random radius idea like we did with the multi-cut problem. The algorithm takes the following steps:

1. Pick a uniformly random permutation π over the vertices, $\{1, ..., n\}$.
2. Pick a radius, r, uniformly from $(\frac{D}{4}, \frac{D}{2})$
3. For each $i \in [n]$, define $S_i = B(\pi(i), r) \setminus \cup_{j<i} S_j$.

The first property of low-diameter partitions follows immediately. For the second property, we have that $Pr[\mathcal{S} \text{ cuts } e] = \sum_{w \in V} Pr[\text{ cut around } w \text{ contains } e]$. This then equals $\sum_{w \in V} Pr[e \in \delta(B(w, r))] Pr[w \text{ is the first node in } \pi \text{ to determine that } e \text{ is cut } | e \in \delta(B(w, r))]$.

This first probability is exactly $\frac{c_e}{|(\frac{D}{4}, \frac{D}{2})|} = \frac{c_e}{\frac{D}{4}}$ since this ball cuts e if exactly one of its endpoints was within distance r from w, which is decided by the choice of r. The second probability is at most $\frac{1}{i}$. This is because for w to be the first vertex to determine e is cut, it must have been the first vertex out of the first i in the permutation that crosses e, otherwise some other vertex decided its fate earlier in our construction. Hence, since the first vertex to determine e's fate is equally likely to be any of the first i vertices in the permutation, the probability it was the last one, w, is $\frac{1}{i}$. Hence, the sum above is bounded above by $\sum_{w \in V} \frac{c_e}{\frac{D}{4}} \frac{1}{i} \leq \frac{4c_e}{D} \log n$. So in total, this procedure gives us a low-diameter partition of V with parameter $\frac{D}{4}$.

Now, we use these partitions as the children of the current root in each level of the recursion for constructing the probabilistic tree embedding as described previously. Thus,

$$\begin{aligned} E[\text{length of e in } T] &\leq \sum_{\text{levels } j} Pr[e \text{ is cut at level } j] 2^{j+1} \\ &= \sum_{\text{levels } j} \frac{c_e}{2^j} O(\log n) 2^{j+1} \\ &= O(\log n) c_e (\# \text{ of levels}) \end{aligned}$$

Note assuming that $c_G(u,v) \geq 1 \forall u \neq v$, the number of levels will be $\log \Delta$ where $\Delta = diam(G)$. Hence, this gives a probabilistic tree embedding with distortion $O(\log n \log \Delta)$

We can do a more refined analysis to show that the distortion is in fact $O(\log n)$.

■

12.5 Next Lecture

We will discuss semi-definite programming techniques and apply them to several combinatorial problems such as Max-Cut and Sparsest-Cut.