0.1 Approximating Arbitrary Metrics by Tree Metrics

0.1.1 Problem Definition

A *metric* (V, d) is a set of points V and a distance function d on pairs of points such that the triangle inequality is satisfied. Given two distance functions d, d', if $d'(u, v) \geq d(u, v)$ for all $u, v \in V$, then we say that the metric (V, d') dominates (V, d). A *tree metric* is an embedding of V into a tree T such that the distance between any two vertices is the length of the (unique) path in T. Here is the problem:

Problem 1. *In the* approximating arbitrary metrics by tree metrics *problem, we are given an arbitrary metric* (V, d)*, and the goal is to find a tree metric* (V, d') *such that* (V, d') *dominates* (V, d) *and the distortion* $\max\{d'(u, v)/d(u, v) : u, v \in V\}$ *is minimized.*

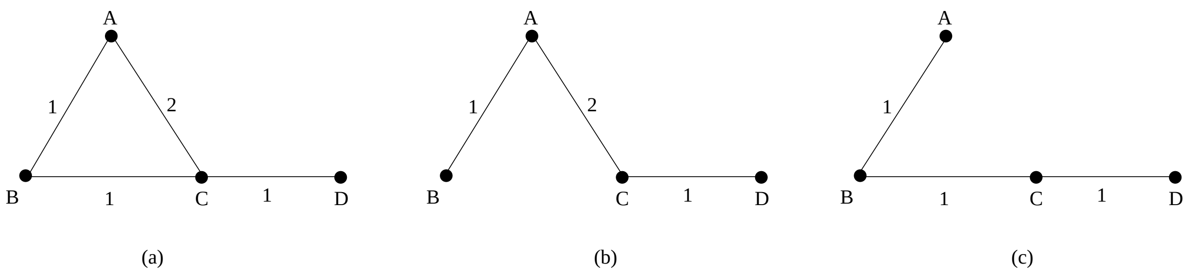

Figure 1: Metrics illustration.

As an example, suppose a metric shown in part (a) of Figure 1 is given where the distance function between any pair of vertices is just the shortest path between them. Part (b) of the figure shows an embedding of the four vertices into a tree such that the distances here are at least the distances in part (a). In particular, we see that it has a distortion of 3 as BC has length 1 in (a) but length 3 in (b). We can improve this by using the embedding in part (c), which is a solution with no distortion.

The bad news is there are graphs (such as a cycle of ones) where the distortion must be at least $\Omega(n)$ [3]. However, we can "improve" this using probabilistic arguments. Bartal [1] has shown that any probabilistic embedding into a tree must have distortion at least $\Omega(\log n)$. Fakcharoenphol, Rao and Talwar [2] give a probabilistic algorithm which yields a distortion of $O(\log n)$, which is a tight bound on this problem. This is the algorithm that we are going to discuss here.

0.1.2 Algorithm Preliminaries

Suppose we are given the metric (V, d) on n points. Assume, wlog, that $d(u, v) > 1$ for all $u, v \in V$, and the diameter $\Delta = \max\{d(u, v) : u, v \in V\} = 2^\delta$ for some δ. For a parameter r, an *r-cut* is a partition of V into subsets (or parts/clusters) where the radius[1] of each subset is no more than r (each cluster has a "centre" where all other vertices are within r from this centre). Note that for each cluster, the distance between any pair of vertices in the cluster is at most $2r$. A *hierarchical cut decomposition* of (V, d) is a sequence of r-cuts $D_\delta, D_{\delta-1}, \ldots, D_1, D_0$ where $D_\delta = \{V\}$, and for $i \leq \delta - 1$, D_i is a 2^i-cut and a refinement of D_{i+1} (i.e. each cluster of D_i is a subset of a cluster in D_{i+1}). Immediately from this definition, we see that D_0 is the partition where each cluster is a single point of V (this is due to the assumption that $d(u, v) > 1$ for all $u, v \in V$).

A hierarchical cut decomposition can be viewed as a rooted tree. The root is $D_\delta = \{V\}$. Each subsequent level of the tree corresponds to a partition D_i where each node of the tree represents points

[1] Recall that the radius of a graph is $\min_{v\in V}\{\max_{u\in V\setminus\{v\}} d(u, v)\}$, and the centre is the vertex satisfying the min in the formula.

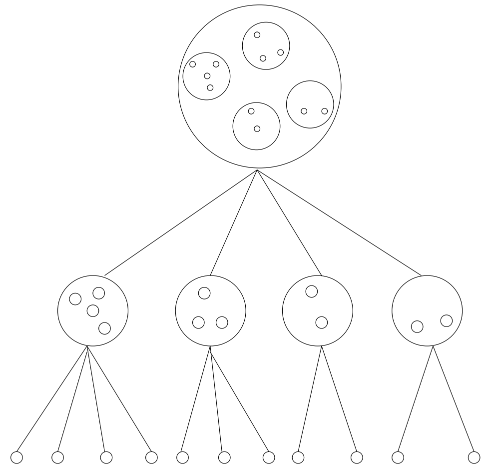

Figure 2: A rough hierarchical cut decomposition.

in one cluster of D_i, and the parent of such a node is the one representing the cluster in D_{i+1} that contains it (see Figure 2 for a rough visualization). Notice that each point in V is in a singleton cluster in some partition. This cluster will be represented as a leaf somewhere in the tree, and this is the node where we embed the point to. For each edge connecting a cluster of D_i to a cluster of D_{i-1}, we give this edge a cost of 2^i. This induces a distance function d_T on V.

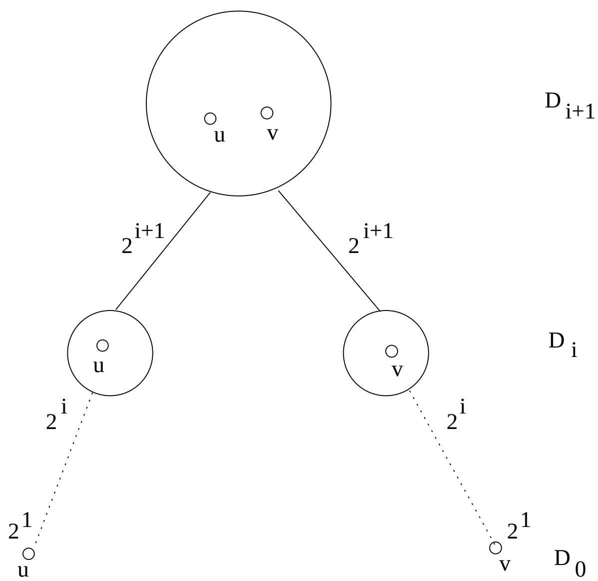

Figure 3: What happens when (u, v) is separated at D_i.

For any pair of points (u, v), we say that the pair is *separated at* D_i if u and v are in the same cluster in D_{i+1}, but they are in different clusters in D_i (see Figure 3). (We distinguish this from the terminology that D_i *separates* (u, v), which simply means u and v are in different clusters in D_i.) The fact that u and v are in the same cluster in D_{i+1} implies that $d(u, v) \leq 2 \cdot 2^{i+1} = 2^{i+2}$. At the same time, the distance between u and v in the tree has to be at least $2 \cdot 2^{i+1}$. So we see that $d_T(u, v) \geq d(u, v)$ for all $u, v \in V$, and d_T dominates d. Also notice that in the worst case, $d_T(u, v) \leq 2 \sum_{j=1}^{i+1} 2^j \leq 2^{i+3}$. This gives an upper bound on the tree metric.

0.1.3 The Algorithm

We will first give a description of the algorithm, and then give a more formal version of it.

We will construct a hierarchical cut decomposition of the given metric in this randomized algorithm. We first fix a random permutation π of V (we see this as an arbitrary ordering of the vertices). We also fix a random number β in the interval $[1, 2]$. We build the tree from top to bottom. The root of the tree is D_δ, which consists of one cluster containing all of V. Suppose we have built the tree up to D_{i+1} (so each cluster has radius at most 2^{i+1}). To build D_i, we first set $\beta_i = 2^{i-1}\beta$, so then $\beta \in [2^{i-1}, 2^i]$. For each vertex v in V, we assign it to the first vertex in π whose distance to v is less than β_i. Then for each cluster S in D_{i+1}, we create subclusters in D_i by grouping those vertices in S that are assigned to the same vertex. The centre of each subcluster S' is the common vertex u that all vertices in S' are assigned to. Since each vertex in S' is no more than β_i away from u, we see that this cluster has radius at most $\beta_i \leq 2^i$. This shows that the thing we created is indeed a hierarchical cut decomposition. Note that it is not necessary that the centre u of a cluster S' is in S'. Also note that it is possible that u is the centre of multiple clusters in D_i.

Algorithm 1 A randomized algorithm for generating a hierarchical cut decomposition.

Pick a random permutation π on V.
Pick a random $\beta \in [1, 2]$.
Set $D_\delta = \{V\}$, $i = \delta - 1$.
while D_{i+1} has non-singleton clusters **do**
  Set $\beta_i = 2^{i-1}\beta$.
  For each $v \in V$, find the smallest j such that $d(v, \pi(j)) \leq \beta_i$. Assign v to $w = \pi(j)$.
  for j=1,2,...,n **do**
    for each cluster S in D_{i+1} **do**
      Create a new cluster in D_i consisting of all vertices $v \in S$ that is assigned to $\pi(j)$.
    end for
  end for
  Set $i = i - 1$.
end while

0.1.4 Analysis of the Algorithm

We want to show that the expected distortion from the algorithm is at most $O(\log n)$. Fix an arbitrary pair (u, v), and we will show that $d_T(u, v) \leq O(\log n) \cdot d(u, v)$.

We have already mentioned that if (u, v) is separated at D_i, then $d_T(u, v) \leq 2^{i+3}$. Therefore,

$$E[d_T(u,v)] \leq \sum_{i=0}^{\delta} \mathtt{Pr}[(u,v) \text{ is separated at } D_i] \cdot 2^{i+3}. \tag{1}$$

We can simplify the analysis by noting that

$$\begin{aligned}\mathtt{Pr}[(u,v) \text{ is separated at } D_i] &= \mathtt{Pr}[D_i \text{ separates } (u,v)] \cdot \mathtt{Pr}[D_j \text{ does not separate } (u,v) \text{ for all } j > i] \\ &\leq \mathtt{Pr}[D_i \text{ separates } (u,v)].\end{aligned}$$

Define $K_i^u = \{w : d(u, w) \leq 2^i\}$. Consider the clustering step at D_i, and suppose that D_i separates (u, v). We say that a vertex w captures u if u is assigned to w. Clearly, if w captures u at level D_i, then $d(u,w) \le \beta_i \le 2^i$, hence $w \in K_i^u$. Also, we say w cuts (u,v) if $w = \pi(l)$ is the smallest l such that w captures exactly one of u and v. We say w cuts u out of (u,v) if w cuts (u,v) and captures u (but not v). Then we have

$$\begin{aligned}\mathtt{Pr}[D_i \text{ separates } (u,v)] &= \sum_{w \in V} \mathtt{Pr}[w \text{ cuts } (u,v)] \\ &= \sum_{w \in W_i^u} \mathtt{Pr}[w \text{ cuts } u \text{ out of } (u,v)] + \sum_{w \in W_i^v} \mathtt{Pr}[w \text{ cuts } v \text{ out of } (u,v)]. \qquad (2)\end{aligned}$$

We wish to bound the probabilities on the RHS of this inequality. Suppose wlog that in D_i, $w = \pi(l)$ cuts u out of (u,v), and suppose $w' = \pi(l')$ captures v. Then $l < l'$. Now order the vertices in W_i^u by their distances from u, say $w_1, w_2, \ldots, w_{|K_i^u|}$, and suppose $w = w_s$ for some s. We already know that $d(u,w_s) \le \beta_i$. Also, $d(v,w_s) > \beta_i$, for otherwise $w_s = \pi(l)$ is a candidate for capturing v, and $\pi(l')$ must have the property that $l' \le l$, which is a contradiction. So we have that $\beta_i \in [d(u,w_s), d(v,w_s)]$. By the triangle inequality, $d(v,w_s) - d(u,w_s) \le d(u,v)$. Since β_i was randomly chosen among $[2^{i-1}, 2^i]$, we see that the probability that β_i lies in $[d(u,w_s), d(v,w_s)]$ is at most $d(u,v)/2^{i-1}$. In addition, for this particular value of β_i, any one of $w_1, w_2, \ldots, w_s$ (plus perhaps other vertices) is within β_i from u, and can capture u. The fact that w_s captures u implies that w_s must be ordered ahead of the other vertices in π. Since π is a random permutation of the vertices, the probability that w_s is the one that captures u is at most $1/s$. Therefore, we have

$$\mathtt{Pr}[w_s \text{ captures } u \text{ but not } v] \le \frac{d(u,v)}{2^{i-1}} \cdot \frac{1}{s}.$$

Substituting this to (2), we get that

$$\begin{aligned}\mathtt{Pr}[D_i \text{ separates } (u,v)] &\le \sum_{s=1}^{|K_i^u|} \frac{d(u,v)}{2^{i-1}} \cdot \frac{1}{s} + \sum_{s=1}^{|K_i^v|} \frac{d(u,v)}{2^{i-1}} \cdot \frac{1}{s} \\ &\le \frac{d(u,v)}{2^{i-1}} (\ln |K_i^u| + \ln |K_i^v|),\end{aligned}$$

where the last inequality is due to the fact that the harmonic series $H_n = \sum_{s=1}^{n} 1/s \le \ln n$. So now back to our original equation (1), we find that

$$\begin{aligned}E[d_T(u,v)] &\le \sum_{i=0}^{\delta} \mathtt{Pr}[D_i \text{ separates } (u,v)] \cdot 2^{i+3} \\ &\le \sum_{i=0}^{\delta} \frac{d(u,v)}{2^{i-1}} (\ln |K_i^u| + \ln |K_i^v|) \cdot 2^{i+3} \\ &\le \sum_{i=0}^{\delta} d(u,v)(2\ln n) \cdot 16 \\ &= 32 \log \Delta \ln n \cdot d(u,v),\end{aligned}$$

where the last line is due to the fact that $\delta = \log \Delta$. So we find that the expected distortion is at most $O(\log \Delta \log n)$. Of course, this is not good enough for our purposes, so we must analyze it more carefully.

If $d(u,v) > 2^{i+1}$ for some i, then (u,v) must be separated in D_i, since any cluster in D_i has diameter at most 2^{i+1}. In particular, D_j separates (u,v) for any $j \le i$. This implies that for $j < i$, (u,v) can never be separated *at* D_j, since D_{j+1} already separated it. So we let j^* be the integer satisfying $2^{j^*+1} < d(u,v) \le 2^{j^*+2}$. Then $\mathtt{Pr}[(u,v) \text{ is separated at } D_j] = 0$ for any $j < j^*$, and (1) can be simplified to

$$E[d_T(u,v)] \leq \sum_{i=j^*}^{\delta} \texttt{Pr}[(u,v) \text{ is separated at } D_i] \cdot 2^{i+3}$$
$$\leq \sum_{i=j^*}^{\delta} \texttt{Pr}[D_i \text{ separates } (u,v)] \cdot 2^{i+3}. \tag{3}$$

Again, we want to bound the probability that a vertex w cuts u out of (u,v). Suppose we have an $i \geq j^* + 4$. The idea here is that we can reduce the number of candidates in K_i^u that can possibly cut u out of (u,v), in particular those vertices in K_{i-2}^u (which is a subset of K_i^u) can never cut (u,v). To see this, suppose that u is assigned to w and $w \in K_{i-2}^u$. Then by definition, $d(u,w) \leq 2^{i-2}$. Also, since $i \geq j^*+4$, $d(u,v) \leq 2^{j^*+2} \leq 2^{i-2}$. Therefore, by triangle inequality, $d(v,w) \leq d(v,u) + d(u,w) \leq 2^{i-2} + 2^{i-2} = 2^{i-1} \leq \beta_i$. Hence, w is one candidate for capturing v, and w can never cut u out of (u,v). So we have

$$\begin{aligned}\sum_{w \in W_i^u} \texttt{Pr}[w \text{ cuts } u \text{ out of } (u,v)] &= \sum_{w \in W_i^u \setminus W_{i-2}^u} \texttt{Pr}[w \text{ cuts } u \text{ out of } (u,v)] \\ &\leq \sum_{s=|K_{i-2}^u|+1}^{|K_i^u|} \frac{d(u,v)}{2^{i-1}} \cdot \frac{1}{s} \\ &= \frac{d(u,v)}{2^{i-1}}(H_{|K_i^u|} - H_{|K_{i-2}^u|}).\end{aligned}$$

Applying the same argument to v, we have

$$\texttt{Pr}[D_i \text{ separates } (u,v)] \leq \frac{d(u,v)}{2^{i-1}}(H_{|K_i^u|} - H_{|K_{i-2}^u|} + H_{|K_i^v|} - H_{|K_{i-2}^v|})$$

This holds for $i \geq j^* + 4$. For $j^* \leq i \leq j^* + 3$, we use the previous bound of $(d(u,v)/2^{i-1}) \cdot 2H_n$. For the grand finale, we now compute the expected value of $d_T(u,v)$ from (3) as follows:

$$\begin{aligned}E[d_T(u,v)] &\leq \sum_{i=j^*}^{\delta} \texttt{Pr}[D_i \text{ separates } (u,v)] \cdot 2^{i+3} \\ &\leq \sum_{i=j^*}^{j^*+3} 2H_n \cdot \frac{d(u,v)}{2^{i-1}} \cdot 2^{i+3} + \sum_{i=j^*+4}^{\delta} (H_{|K_i^u|} - H_{|K_{i-2}^u|} + H_{|K_i^v|} - H_{|K_{i-2}^v|}) \cdot \frac{d(u,v)}{2^{i-1}} \cdot 2^{i+3} \\ &\leq 16 \cdot 2 \cdot 4 \cdot H_n \cdot d(u,v) + 16(H_{|K_\delta^u|} + H_{|K_\delta^v|} + H_{|K_{\delta-1}^u|} + H_{|K_{\delta-1}^v|}) \cdot d(u,v) \\ &\leq 128 H_n \cdot d(u,v) + 64 H_n \cdot d(u,v) \\ &\leq 192 \ln n \cdot d(u,v),\end{aligned}$$

where the third last inequality is due to the telescoping sum in the previous line. So we have concluded that $E[d_T(u,v)] \leq O(\log n) d(u,v)$, and the expected distortion is $O(\log n)$. This concludes the analysis of the algorithm.

Bibliography

[1] Bartal, Y., On approximating arbitrary metrics by tree metrics, *IEEE Symposium on Foundations of Computer Science* 1996, 184-193.

[2] Fakcharoenphol, J., Rao, S., and Talwar, K., A tight bound on approximating arbitrary metrics by tree metrics, *STOC* 2003, 448-455.

[3] Rabinovich, Y. and Raz, R., Lower bounds on the distortion of embedding finite metric spaces in graphs, *GEOMETRY: Discrete & Computational Geometry* 19(1998).