

23.1 Problem Statement and Results

REcall that in the previous lecture, we discussed the use of probabilistic tree embeddings for approximations algorithms. We now present an algorithm for obtaining low distortion tree embeddings.

Let G be a graph and let \mathcal{T} be a collection of trees on the same vertices as G . We say the \mathcal{T} is a γ distortion if, for any vertices x and y and any $T \in \mathcal{T}$:

$$d_G(x, y) \leq d_T(x, y)$$

and

$$E[d_T(x, y)] \leq \gamma d_G(x, y)$$

The expected value is taken over a distribution given to the elements in \mathcal{T} . It has been shown in [2] that for an arbitrary graph we can have $\gamma = O(\log n \log \log n)$. This result has been improved to a $O(\log n)$ distortion in [1], which is essentially optimal.

23.2 Construction

Let G be a graph with n nodes and diameter $\Delta = 2^\delta$. The constructions for good tree embeddings have the set \mathcal{T} made of trees of the following form. They are called Hierarchically well separated trees, which means the distance from each parent to its children is the same, and at each level the distance decrease by a constant factor. A tree will have a root that corresponds to the entire graph. Then, considering each node in the tree as a subset of the graph, the children nodes of each node, v , will be a partition of the vertices of the graph in the subset corresponding to v . We shall make it so that a node at a depth of i shall correspond to a subset of the graph with diameter less than $2^{\delta-i}$. Notice that at depth δ we are left with the vertices of the original graph.

We set the cost of traveling on an edge of depth i to be $2^{\delta-i}$. Thus, the cost of getting from a vertex u to a vertex v in one of our trees is at least the cost of getting from u to v in the graph since in a tree you have to travel up to a subset that contains both u and v and then back down. This cost will be large enough from the cost put on edges of depth i . Thus, we have that a set of trees with this property satisfies the first requirement to be a γ distortion embedding of G . The key to the argument, is that if the graph is partitioned well then the expected distance between two points won't change much. As seen by Bartal, if a metric graph can be probabilistically partitioned into pieces where the diameter has decreased by a constant factor and the chance an edge is cut is about its length times λ over the diameter, then, by recursively using these partitions for our

tree, we get a final probabilistic embedding into trees with distortion $O(\lambda \log(\Delta))$. The existence of a partition with $\lambda = \log(n)$ was given by Calinescu et al, see [3].

In [1] a method for partitioning the graph G randomly into a tree of the above form is given. By a better analysis, they were able to achieve $O(\log(n))$ as the final distortion, removing the dependence on Δ . The basic idea is as follows. The vertices are ordered in an arbitrary manner. Then, a random $\beta \in [1, 2]$ is chosen. The set G is partitioned by moving through the vertices in the order given and including all points at a radius of $2^{\delta-1}\beta$ from the current vertex. The points in the ball are made into a cluster that form a node at the next level. We do this for each vertex, only including a new vertex in a cluster if it hasn't been assigned yet. We then treat each cluster as a new graph and repeat, adjusting the radius by a factor of 2.

Now, we look at the expected distance between vertices u and v . As shown in [1], the expected distance is raised by at most a factor of $O(\log(n))$. Let e be the edge between u and v . The expected distance between u and v is less than:

$$\sum_j Pr[e \text{ is cut at time } t]2^t$$

Let $B(e, r)$ be the balls of radius r around u and v . By proving that the probability that e is cut at time t is less than $4d(u, v) \log\left(\frac{|B(e, 2^{\delta-t})|}{|B(e, 2^{\delta-t+1})|}\right)2^{-t}$ and noticing that this is a telescoping sum with first term $4d(u, v)\log(n)$, we get the $O(\log(n))$ factor.

This result is tight because tree metrics are contained in L^1 metrics, which we know have a distortion of $O(\log(n))$. For more details on the proof, please see the references below.

References

- [1] Fakcharoenphal et. al., *A Tight Bound on Approximating Arbitrary Metrics by Tree Metrics*. STOC 2003, San Diego, CA.
- [2] Bartal, *On Approximating Arbitrary Metrics by Tree Metrics*. <http://www.cs.huji.ac.il/~yair/pubs/B-prob-approx2.ps>
- [3] Calinescu, et. al., *Approximation Algorithms for the 0-Extension Problem (2000)*. SODA 2001, Washington D.C., USA