Lecture #19

Lecturer: Debmalya Panigrahi *Scribe: Xingyu Chen*

1 Overview

Today, we focused on metric embeddings technique more closely. Specifically, we focused on tree metric embedding, which aims to embed an arbitrary graph metric into a tree metric without distorting the original distance by a large factor.

2 Definition

2.1 Tree Embedding Overview

We start by considering the goal of embedding. Ideally, given a graph $G = (V,E)$, we would like have an associated tree T such that,

Theorem 1. *Given a graph $G = (V,E)$, we have a tree T,*

1. $\forall x,y \in v, \quad d_T(x,y) \geq d_G(x,y)$
2. $\forall x,y \in v, \quad d_T(x,y) \leq \alpha \cdot d_G(x,y)$

In other words, we want the distance between vertices in tree T strictly dominates the original distance in G, and at the same time doesn't get stretched by over α factor. However, it is not feasible generally to satisfy both requirements.

The n-length cycle creates such a problematic situation, which distorts the distance by $\Omega(n)$ factor.

So, instead of looking for a single tree, we look for a distribution of trees such that tree drawn from the distribution will satisfy the second requirement in the expected sense. In other words, we are looking for a randomized embedding such that.

Theorem 2. *Given a graph $G = (V,E)$, we have a tree T and a distribution D,*

1. $d_T(x,y) \geq d_G(x,y) \quad \forall x,y \in v, \forall T \in D$
2. $\mathbf{E}_{T\sim D}[d_T(x,y)] \leq O(\log n) \cdot d_G(x,y) \quad \forall x,y \in v$

2.2 Sketch of our approach

Let's first sketch how we should do this. Given a graph G. Imagine a tree T whose length of the first layer edges is 2^i, whose length of the second layer edges is 2^{i-1}. Each next layer's edge length will shrink by 2 as in Figure 1. Now, consider vertices x and y in the original graph, which is $d(x,y)$ away from each other.

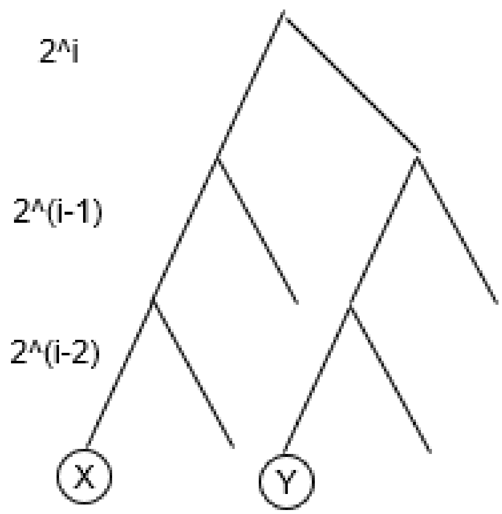

Figure 1: Sketch of Tree T

If I want to preserve the distance exactly in this tree T, I should place x and y such that their least common ancestor is about at $i = \log n$ layer. The reason is because the distance of x and y on this tree is dominated by the direct edges out of their LCA. So $2^i = d(x,y)$ roughly means $i \approx \log n$.

Now let's further consider we are the common ancestor z of x and y, length of each edge branching from z is D. The probability of separating x and y should be less than $\frac{d(x,y)}{D}$. Or else the expected distance of x and y on T will blow up and exceeds the original $d(x,y)$.

So in other words, we want the probability of separating x and y at a layer whose edge length is δ to be less than $d(x,y)/\delta$, while largest distance in each sub tree is no more than δ. These requirements lead to our tentative subroutine.

Tentative Subroutine:
Given any δ, partition the points in the metric space s.t.

- Diameter of each part of the partition $\leq \delta$
- Pr[x,y in different parts] $\leq d(x,y)/\delta$

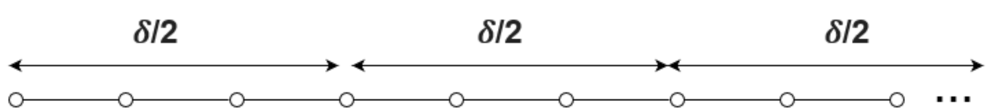

Figure 2: The line we considered

2.3 Test of The Tentative Subroutine

Let's test this subroutine with a line consisting of n points. Each point is distance one from both of its adjacent points, as shown in Figure 2. I want to do a tree metric embedding on this graph to preserve the distance. Consider the simplest situation where I just remove each segment with some probability P_δ, we claim,

Theorem 3. *If we remove any segment in Figure 2 with* $P_\delta = \frac{4logn}{\delta}$*, we have a high probability of separating these vertices into different partitions and satisfy the first requirement of the tentative subroutine, diameter of each part of the partition* $\leq \delta$*.*

Proof. I cut these line into $\frac{2n}{\delta}$ segments. Each segment is of length $\frac{\delta}{2}$. In each segment, I have

$$(1-P_\delta)^{\delta/2} = [(1-P_\delta)^{1/P_\delta}]^{2\log n} \leq \frac{1}{e}^{2\log n} \approx \frac{1}{n^2}$$

probability of not removing an edge in each segment. There are $\frac{2n}{\delta}$ segments. We union bound on these segments. The probability of removing an edge in every segment is larger than $1 - n \cdot \frac{1}{n^2} = 1 - \frac{1}{n}$.

If we have a high probability of removing one edge from each $\delta/2$ segments, the diameter of each partition will not exceed δ. This shows that the approach does guarantee a high probability of satisfying the first requirement. □

Now, let's look at the second requirement, Pr[x,y in different parts] $\leq d(x,y)/\delta$. If I remove every segment with probability $4\log n/\delta$, the probability of one pair of vertices got disconnected is roughly $\log n \frac{4d(x,y)}{\delta}$. This suggests that our tentative subroutine may not be realistic. Instead, our subroutine should be:

Subroutine:
Given any δ, partition the points in the metric space such that

- Diameter of each part of the partition $\leq \delta$
- Pr[x,y in different parts] $\leq \log n \frac{4d(x,y)}{\delta}$

3 Algorithm for Tree Embedding

Assume we do have such a subroutine. Then we can construct an algorithm to find the tree embedding using the following algorithm.

Algorithm

- Set $\delta = \frac{diam}{2}$
- Create a partition P_1, P_2, ... P_k
- Repeat: $\delta = \delta/2$. Recurse on each P_i with δ
- Stop when each partition is a singleton.

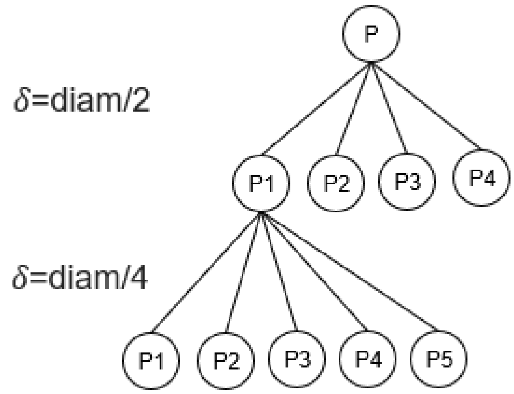

Figure 3: Graphic View of the Algorithm

3.1 Analysis of the algorithm

First, we note that the depth of the recursion is $O(\log \Delta)$, where

$$\Delta = \frac{\max_{x,y\in v} d_G(x,y)}{\min_{x,y\in v} d_G(x,y)}$$

The tree T is formed by setting each layer's edge length to be the δ we used to partition this layer. Now we prove that this tree T gives us the desired properties, which

1. $d_T(x,y) \geq d_G(x,y) \qquad \forall x,y \in v, \forall T \in D$
2. $\mathbf{E}_{T\sim D}[d_T(x,y)] \leq O(\log n)\cdot d_G(x,y) \qquad \forall x,y \in v$

Lemma 4. *This algorithm ensures $d_T(x,y) \geq d_G(x,y)$ $\forall x,y \in v$, $\forall T \in D$.*

Proof. According to the first property of our subroutine, each partition's diameter will be upper bounded by δ. This means when $\delta \leq \frac{d_G(x,y)}{2}$, x and y must have already been separated. Or else, there exists a partition whose diameter is larger than δ.
Then since x and y have already been separated from each other when $\delta = \frac{d_G(x,y)}{2}$, x and y will at least be $2\delta > 2\frac{d_G(x,y)}{2} = d_G(x,y)$ apart. □

Lemma 5. *This algorithm ensures $\mathbf{E}_{T\sim D}[d_T(x,y)] \leq O(\log n)\cdot d_G(x,y)$ $\forall x,y \in v$*

Proof. Let's assume x and y are split at some point z, whose branch edges have length l (i.e. $\delta = l$ in this step). We have $l \geq \frac{d_G(x,y)}{2}$. Now we can sum over all possible points that x and y are split to get the expected value of $d_T(x,y)$

$$\begin{aligned}
\mathbf{E}[d_T(x,y)] &\leq \sum_l 4l \cdot \Pr[\text{x and y are split at } \delta = l] \\
&\leq \sum_l 4l \cdot \frac{d(x,y)}{l/4} \log n \\
&= O(\log n \cdot \log \Delta \cdot d(x,y))
\end{aligned}$$

This is still one step from what we tried to prove. But let's leave this for now and go back latter after we have better understanding of the algorithm. □

4 Subroutine

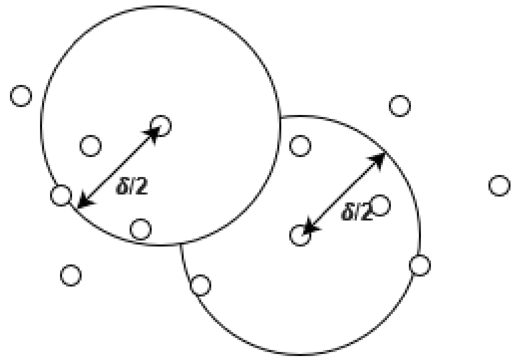

Figure 4: CKR Technique

4.1 CKR Technique

We are going to use this CKR technique devised by [GC01] as our subroutine. This technique can be described as below (See Figure 4): I have all n points in the metric space. Take a point and draw a ball with radius $\frac{\delta}{2}$ and put all vertices inside the ball into one partition. Then pick another point, create a ball and repeat. However, this process is a deterministic process. We need to introduce some randomness here.

1. The first randomness we introduced is that we use a random permutation σ to decide order of points(balls) we draw.

2. The second randomness we introduced is we choose R uniformly from $[\delta/4, \delta/2]$.

If we use this process as subroutine, since $d(x,y) \leq 2 \cdot \delta/2 = \delta$ for every two points inside each ball, the first requirement is immediately satisfied.

We focus on the second requirement in the remaining part of this section. Fix some pair of points x and y. Order all other vertices in increasing distance from $\{x,y\}$ and call them 1,2,3... etc. Here comes our key theorem.

Theorem 6. *If vertex i separates x from y, then i must be the first vertex in σ among all the vertices 1,2...i.*

Proof. Consider j < i. Since j appears before i, j must be closer to $\{x,y\}$ than i. If i separates x from y, then without loss of generality, we can assume

$$d(i,x) \leq R \leq d(i,y)$$

However, either $d(j,x) \leq d(i,x)$ or $d(j,y) \leq d(i,x)$ must hold. Or else, we won't put j before i in our distance increasing order.
Assume $d(j,x) \leq d(i,x) \leq R$. Consider the ball created by j.

1. If y is also inside the ball created by j, then we can't separate them in i's ball since it has already been put together by j. This case is not possible.

2. If y is outside the ball created by j, then we have already separated them in j's ball. i then can't separate x and y. This case is as well not possible.

This means there's no such j before i. i must be the first vertex in σ among all the vertices 1,2,...i. $\square$

With this theorem in our hand, we can calculate the probability of i separates x and y given i is the first vertices among 1,2...i.

$$\begin{aligned}\text{Pr[i separates x,y }|i <_\sigma j \quad \forall j = 1,2,...i-1] &= Pr[d(i,x) \leq R \leq d(i,y)] \\ &\leq \frac{|d(i,x) - d(i,y)|}{\delta/4} \\ &\leq \frac{d(x,y)}{\delta/4}\end{aligned}$$

The first equality is because only when R lies between two vertices, our ball separates x and y. The second inequality comes from the triangular inequality. Following this, the probability of x and y are separated considering all x and y is

$$\begin{aligned}\text{Pr[x,y are separated]} &= \sum_i \text{Pr[i separates x,y]} \cdot (1/i) \\ &\leq O(\log n) \cdot \frac{d(x,y)}{\delta/4}\end{aligned}$$

This inequality shows that the second requirement of the subroutine is also satisfied by CKR technique.

5 Further Analysis of the algorithm

Here, we can still one step from what we promised since in our algorithm analysis section. Our $E[d_T(x,y)]$ is a $\log \Delta$ factor larger than what we promised. But this is due to our analysis. If we analyzed our algorithm from a more detailed perspective, we actually can get exactly what we want.

$$\begin{aligned}\mathbf{E}[d_T \text{ (x,y)}] &\leq \sum_\rho \rho \cdot \text{Pr[x, y are split at } \rho] \\ &= \sum_\rho \rho \cdot \frac{d_T(x,y)}{\rho/4} \cdot \sum_i 1/i \\ &= 4d_T(x,y) \cdot \sum_\rho (\sum_i 1/i) \\ &= 4d_T(x,y) \sum_\rho \sum_{i \text{ s.t. } \rho/4 < d(i,\{x,y\}) < \rho/2} 1/i \\ &\leq 4d_T(x,y) \cdot 2 \cdot \sum_i 1/i \\ &= 8 \log n \cdot d_T(x,y) = O(\log n \cdot d_T(x,y))\end{aligned}$$

The reason why we only need to consider i s.t. $\rho/4 < d(i,\{x,y\}) < \rho/2$ is the key point of this new analysis.

1. We don't need to consider $d(i,\{x,y\}) > \rho/2$ is because if $d(i,\{x,y\}) > \rho/2$, we can never separate x and y even if you choose the largest radius.

2. The reason why we don't need to consider $d(i,\{x,y\}) < \rho/4$ is left as exercise.

Now consider the distance between x and y. If distance between x and y is less than $\rho/4$, they will either have already been separated or both get covered. So i will take care of them. If distance between x and y is large than $\rho/4$, ρ must be larger than $d(x,y)/2$. Then this tells us $d(x,y)$ is no more than 2ρ. So x and y can't be too far from each other. Then the number of ρ we need to consider can be bounded by a constant number. In fact, i only appears for at most 2 different ρs. Then our result follows.

References

[GC01] Y. Rabani G. Calinescu, H. Karloff. Approximation algorithms for the 0-extension problem. *Proceedings of the 12th Annual ACMSIAM Symposium on Discrete Algorithms*, pages 8–16, 2001.