CS 270 Algorithms **Spring 2021**

Lecture 12: Low-diameter decompositions and HSTs

Lecturer: Prasad Raghavendra Scribes: Saachi Mutreja and Sylvia Jin

12.1 Metric Embeddings

Definition 12.1. *A **metric space** $(X, d : X \times X \to \mathbb{R}^+)$ is a set X with a distance function d, where for all $u, v \in X$:*

1. $d(u, u) = 0$
2. $d(u, v) = d(v, u)$
3. $d(u, v) + d(v, w) \geq d(u, w)$ *(a.k.a. **Triangle Inequality**)*

Often when we want a distance metric on n points (e.g. shortest-path metric on a graph), we want some way to approximate them via embeddings (e.g. shortest-path on a tree, which may be easier to compute). Approximations via metric embeddings are most useful in approximation algorithms and online algorithms.

However, there are cases where a metric embedding is impossible, e.g. if we have a metric embedding from a graph G to a tree, $(X, d) \to (T, d_T)$, then if G has a cycle anywhere, then it's impossible to satisfy $d_T(x, y) = d(x, y)$ everywhere. In fact, cycles of length n will incur $\Theta(n)$ distortion!

12.2 Randomized/Probabilistic Tree Embeddings

To remedy such drastic distortion, we can "cut edges" to get rid of cycles, motivating randomized algorithms that have low distortion on average.

Definition 12.2. *Given a metric (X, d) over graph $G = (V, E)$, a **(randomized) α-low-stretch spanning tree** of stretch α is a probability distribution D over spanning trees of G, where for all $u, v \in X$:*

1. $d_G(u, v) \leq d_T(u, v)$ *for all trees* $T \sim D$
2. $\mathbb{E}_{T \sim D}[d_T(u, v)] \leq \alpha d_G(u, v)$

How good is this spanning tree approximation?

Theorem 12.3. *For any metric space (X, d), there exists an α-low-stretch spanning tree distribution D, where $\alpha \in \mathcal{O}(\log |X|) = \mathcal{O}(\log N)$.*

The proof involves introducing low-diameter decompositions (LDDs), which partition a metric space into clusters of low diameter, and then using an LDD reursively to construct a low-stretch hierarchical tree decomposition.

12.2.1 Low-diameter Decompositions

Definition 12.4. *The* ***diameter*** *of a set X under some metric d is the maximum distance between any 2 points in X, or $\max_{x \in X} |x_i - x_j|_d$.*

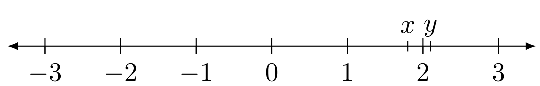
Definition 12.5. *A* β***-low-diameter decomposition scheme*** *is a randomized algorithm with inputs:*

- *stretch parameter β*
- *a metric space (X, d)*
- *bound $\Delta > 0$ on set diameter*

that outputs a partition P of X as $P = \bigcup_{i=1}^{t} X_i$ such that

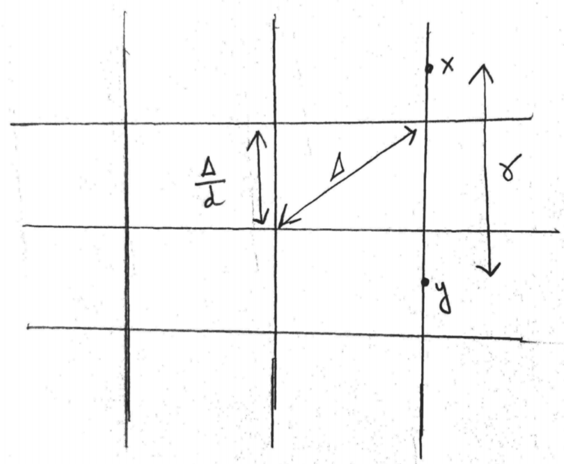
- *for all $i \in [t]$, $\text{diameter}(X_i) \leq \Delta$*
- *for all $x, y \in X, x \neq y$, $\Pr[P(x) \neq P(y)] \leq \beta \cdot \frac{d(x,y)}{\Delta}$*

Example 12.6. *Consider $(X, d) = (\mathbb{R}, |\cdot|)$, $\Delta = 1$, β unspecified. If we mark the number line at integer values, we can see we satisfy the first definition, but run into an issue when x and y are close together but are in different clusters:*

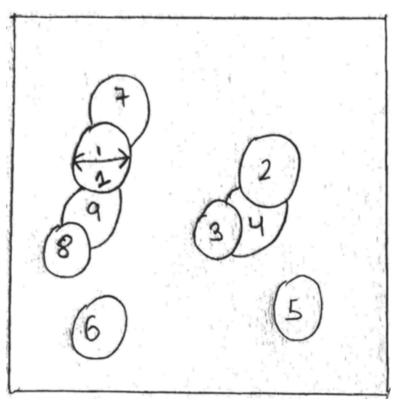


Specifically, to remedy situations where $|x - y| < \Delta$ but $\Pr[P(x) \neq P(y)]$, we can apply a random shift to the marks: choose a shift $R \in [0, \Delta]$ uniformly at random, and mark the line at intervals of $k\Delta + R$ for $k \in \mathbb{Z}$. Then we can see that $\Pr[P(x) \neq P(y)] = \frac{|x-y|}{\Delta} = |x - y|$, proportional to the distance between them.

Example 12.7. *Consider $(X, d) = (\mathbb{R}^d, \|\cdot\|_1)$, Δ the length of the largest diagonal of a d-length hypercube, and β unspecified. Then if we cut our space into hypercubes of side length Δ/d, we see that $\Pr[P(x) \neq P(y)] = \frac{r}{\Delta/d} = \frac{dr}{\Delta}$, so ultimately the separation probability of 2 points doesn't exceed the distance between them.*



Example 12.8. *Consider the embedding $\ell_2 = (\mathbb{R}^2, \|\cdot\|_2)$, $\Delta = 1$, where we tile randomly. This means we randomly pick some x_1, assign all the points within Δ of it to its partition, pick some other unpartitioned point x_2, and continue this way until all points have been tiled.*



Analysis: for any $x, y \in \mathbb{R}^2$, $\|x-y\|_2 = \varepsilon$, we have that $\Pr[P(x) \neq P(y)] = \dfrac{[B(x,1) \triangle B(y,1)]}{[B(x,1) \cup B(y,1)]}$, *where $A \triangle B$ is the **symmetric difference** between sets A, B, and $[A]$ represents the area/volume of A.*

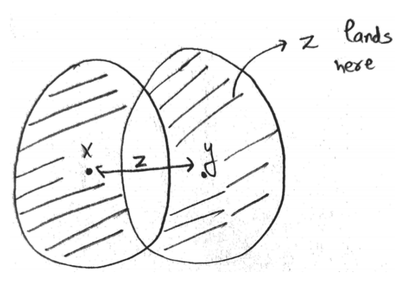


Figure 12.5: geometric explanation of symmetric difference formula

To see this, without loss of generality, let x be assigned a partition first. From the diagram above, note that for a point z to be in x's partition but not y's, $\|x - z\| \leq 1$ by definition of z, but $P(x) \neq P(y)$ means $\|y - z\| > 1$. Repeating the same logic for when y is partitioned first, this means that z falls in the shaded region above. Geometrically, the denominator $[B(x, 1) \cup B(y, 1)]$ is proportional to the surface area of B.

But we can do even better! If we randomize both the radius and our tiling order, we have an algorithm that satisfies both the low radius and low cutting property:

Inputs: metric (X, d), stretch β, bound $\Delta > 0$ as above

Output: partition $P = \bigcup_{i=1}^{t} X_i$ such that

- for all $i \in [t]$, diameter$(X_i) \leq \Delta$
- for all $x, y \in X, x \neq y$, $\Pr[P(x) \neq P(y)] \leq \beta \cdot \frac{d(x,y)}{\Delta}$

Algorithm:

1. Pick a random radius $R \in [\Delta/4, \Delta/2]$
2. Pick a random permutation π of $X = \{x_1, \ldots, x_n\}$
3. In the order given by $\pi(x_1), \pi(x_2), \ldots, \pi(x_n)$, iteratively tile X with balls such that $\pi(x_1)$ is partitioned into $B(\pi(x_1), R)$, and for all points $\pi(x_i)$ afterward, we have that x_i is part of

$$P := \left\{ B(\pi(x_i), R) \setminus \bigcup_{j<i} B(\pi(x_j), R) \middle| i \in [n] \right\}$$

Theorem 12.9. *Suppose $d(x, y) = r \leq \Delta/8$. Then* $\Pr[P(x) \neq P(y)] \geq \exp(-\frac{8r}{\Delta} \log \frac{|B(x,\Delta)|}{|B(x,\Delta/8)|})$.

Proof. Recall that $R \in [\Delta/4, \Delta/2]$, and $r \leq \Delta/8$. First, the diagram below inspires the geometric observation:

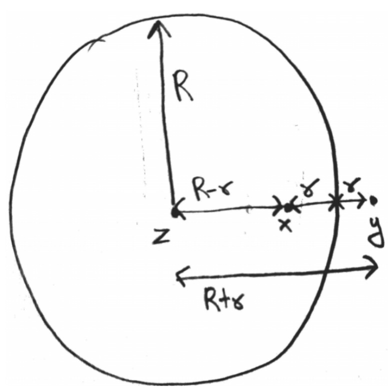


Figure 12.5: conditioning on R, we need the center to be within $R - r$ of x to be in the right partition.

$$\begin{aligned}\Pr[P(x) \neq P(y)] &\geq P(d(x,z) < R - r)\\ &\geq \frac{|B(x, R-r)|}{|B(x, R+r)|}\\ &= \frac{\text{volume(intersection)}}{\text{volume(union)}}\end{aligned}$$

By convexity of $\exp(x)$, we have that $\mathbb{E}[\exp(z)] \geq \exp \mathbb{E}[z]$.

We also know from the definitions of R and r that $\begin{cases} R + r < \Delta/2 + \Delta/8 < \Delta \\ R - r \geq \Delta/4 - \Delta/8 \geq \Delta/8 \end{cases}$, so

$$\begin{aligned}\Pr[P(x) \neq P(y)] = \Pr\left[B(x,r) \subseteq P(x)\right] &\geq \mathbb{E}\left[\frac{|B(x,R-r)|}{|B(x,R+r)|}\right]\\ &= \mathbb{E}\left[\exp\left(-\log\frac{|B(x,R+r)|}{|B(x,R-r)|}\right)\right]\\ &\geq \exp\left(\mathbb{E}\left[-\log\frac{|B(x,R+r)|}{|B(x,R-r)|}\right]\right)\\ &\geq \exp\left(\frac{-8r}{\Delta}\log\frac{|B(x,\Delta)|}{|B(x,\Delta/8)|}\right),\end{aligned}$$

where the last inequality comes from integrating

$$\begin{aligned}\mathbb{E}\left[\log\frac{|B(x,R+r)|}{|B(x,R-r)|}\right] &= \frac{4}{\Delta}\int_{\Delta/4}^{\Delta/2}\log\frac{|B(x,R+r)|}{|B(x,R-r)|}\,dR\\ &\leq \frac{8r}{\Delta}\log\frac{|B(x,\Delta/2+r)|}{|B(x,\Delta/4-r)|}\\ &\leq \frac{8r}{\Delta}\log\frac{|B(x,\Delta)|}{|B(x,\Delta/8)|}\end{aligned}$$

□

We can now use this algorithm recursively to construct a hierarchical tree decomposition: given a metric (X, d), where X has diameter d, we use LDD to decompose it into clusters with diameter $\Delta \leq d/2$. Then we recursively build a tree for each cluster, and combine them into 1 tree.

12.2.2 Hierarchical Spanning Tree Decomposition

For this LST (low-stretch tree) algorithm we have as input: metric space (X, d), where diameter$(X) \leq 8^{\delta}$, where the output is a low-stretch *hierarchical* spanning tree with low distortion.

Algorithm:

1. Use LDD decomposition on X with $\Delta = 8^{\delta-1}$ to get a partition $P = \bigcup_{i=1}^{t} X_i$ of X.

2. For $j \in [t]$, define X_j as a metric restricted to the points in X_j, and recursively run this algorithm on the smaller $(X_j, \delta - 1)$.
3. Add edges of length 8^{δ} from the root of this tree T_1 to the roots of $T_2, \ldots, T_t$.

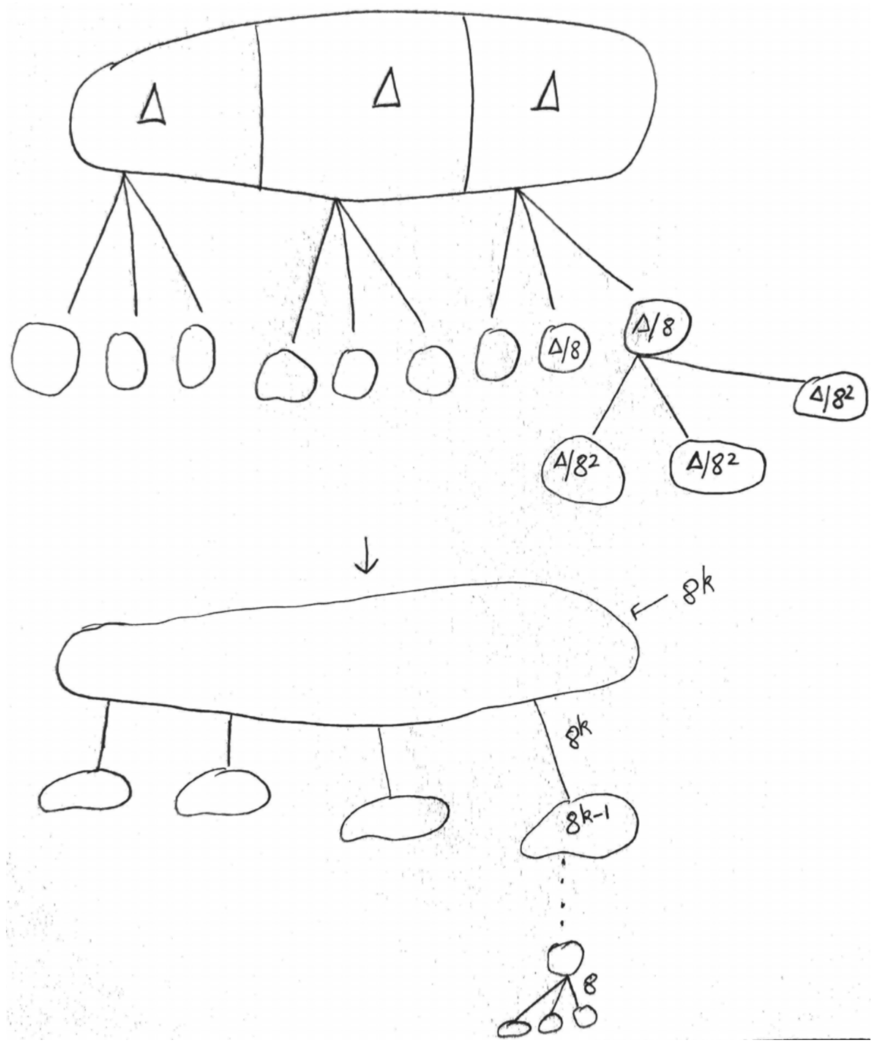


Figure 12.5: construction of our tree using the LDD decomposition, where at the highest level, we partition X into sets that each have a diameter of Δ. The root has weight 8^k, and at each level below it, the edge has weight 8^k and connects to a node with weight 8^{k-1}, and the diameter also decreases by a factor of 8.

How good is this spanning tree approximation? We can use the corollaries proved so far to prove our original theorem.

Theorem 12.10. *For any metric space (X, d), there exists an α-low-stretch spanning tree distribution D, where $\alpha \in \mathcal{O}(\log |X|) = \mathcal{O}(\log N)$.*

Proof. First, consider 2 vertices $x, y \in X$, where $8^{j-1} \leq d(x,y) \leq 8^j$. They must have been separated in the tree sometime during the algorithm, so let's call the *least common ancestor* of x and y $lca(x,y)$. Because $d(x,y) \leq 8^j$, $lca(x,y)$ must have height at least j. Call this height h, and note that $h \geq j$.

Then traversing the path in the low-stretch spanning tree from x to y, we have that $d_T(x,y) \geq 8^h + 8^{h-1} + \ldots + 8^2 + 8^1 \geq 8^j \geq d(x,y)$.

We now use our previous corollary (with $\Delta = 8^t$) to compute

$$\begin{aligned}
\mathbb{E}[d_T(x,y)] &\leq 8^j + \sum_{k=j+1}^{\infty} 8^k \Pr[P_k(x) \neq P_k(y)] \\
&\leq 8^j + \sum_{k=j+1}^{\infty} 8^k \frac{8d(x,y)}{8^t} \log \frac{|B(x,8^t)|}{|B(x,8^{t-1})|} \\
&\leq 8^j + d(x,y) \sum_{k=j+1}^{\infty} 8 \log \frac{|B(x,8^t)|}{|B(x,8^{t-1})|} \\
&\leq 8^j + d(x,y) \log n \\
&\in \mathcal{O}(8^j \log n)
\end{aligned}$$

where the last inequality comes from the summation being a telescoping sum. Defining $\delta = \mathcal{O}(\log \text{diameter}(X))$ and $\beta = \mathcal{O}(\log(n))$ completes the proof. A paper by Fakcharoenphol, Rao, and Talwar actually shows that the bound $\mathcal{O}(\log \text{diameter}(X) \log n)$ can be improved to just $\mathcal{O}(\log n)$.

One application is for buy-at-bulk network design, where the goal is to design a graph that can support many flows $x_i \to t_i$, each edge e has a capacity c_e, and we want to minimize $\sum_{e \in E} \text{cost}(c_e)\text{dist}(e)$.

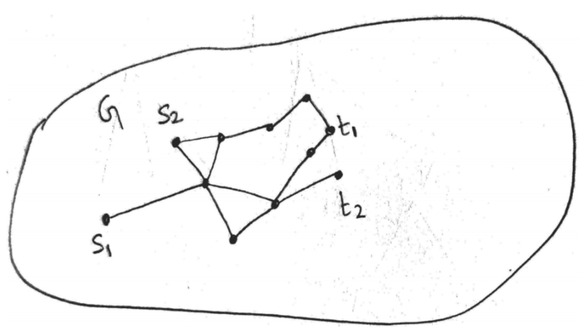


Figure 12.5: sample buy-at-bulk network graph

The solution, after noticing that the problem is easy and deterministic on trees, is to use a hierarchical tree decomposition to approximate the solution. □