CSC2414 - Metric Embeddings*
Lecture 3: Embedding to Random Trees

Notes taken by Nilesh Bansal and Ilya Sutskever
Revised by Hamed Hatami

Summary: It is not always possible to embed a metric space in a tree with low distortion. To overcome this, the metric space can be instead embedded in a distribution of trees. We discuss the applicability of these ideas in online algorithms, and prove that every finite metric space of n points embeds to a distribution of trees with distortion $O(\log n)$.

1 Embedding to Trees

Not every metric can be embedded into a tree isometrically. Embedding C_n into a tree results in a distortion of at least $n-1$ (for example, as in Figure 1, deleting an edge from the cycle results in a tree metric with expansion $n-1$).

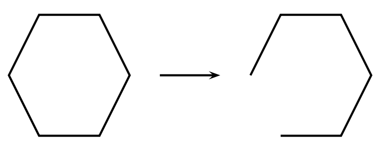

Figure 1: Embedding C_6 to a tree by removing an edge.

Theorem 1.1. *Every embedding of C_n into a tree T incurs distortion $\Omega(n)$.*

Theorem 1.1 appears in [RR98], and it holds true even when the embedding of C_n to tree contains edges and vertices not in C_n (Steiner points).

If C_n is embedded into a tree which is result of removing one of the edges at random from the cycle, one edge will have a large expansion while others will remain intact. Therefore, we are more interested in finding the maximum expected expansion between a pair of vertices in the embedding. We define a new measure

$$\max_{x,y} \frac{E[d(\psi(x), \psi(y))]}{d(x,y)},$$

* Lecture Notes for a course given by Avner Magen, Dept. of Computer Sciecne, University of Toronto.

which will also serve as motivation for Definition 1.2 below. It must be noted that this is very different from the expected maximum expansion for the complete space, as that will be

$$E[\max_{x,y} \frac{d(\psi(x), \psi(y))}{d(x,y)}],$$

which is $\Omega(n)$ according to Theorem 1.1.

The tree obtained by removing an edge from a cycle will "dominate" the original metric, i.e., no distances can get contracted. Also, while computing the expansion in a graph metric, it is sufficient to consider the expansion along the edges because the expansion in distance between any two points will be bounded by the sum of expansion of edges along the shortest path (as proved in the previous tutorial).

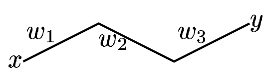

Figure 2: The expansion of distance between x and y is bounded by the sum of expansion of edges w_1, w_2 and w_3.

Definition 1.2. A set of metric spaces $\mathcal{T}$ α-probabilistically approximates metric space (X, d) if

1. Every metric τ in $\mathcal{T}$ dominates (X, d), i.e., $d(i,j) \leq d_\tau(i,j)$ $\forall i,j \in X$ and $\tau \in \mathcal{T}$;

2. and, there exists a probability distribution over $\mathcal{T}$ such that the expected distance is not too much larger than $d(i,j)$, i.e.,

$$\forall i,j \;\; E_{\tau \in \mathcal{T}}[d_\tau(i,j)] \leq \alpha \cdot d(i,j).$$

It must be noted that the probability distribution over $\mathcal{T}$ may not be a uniform distribution. Some metric spaces in $\mathcal{T}$ may be more favorable than others.

When a tree is obtained by removing an edge at random from a cycle, for any edge xy, the distance will remain unaltered with probability $\frac{n-1}{n}$, and with probability $\frac{1}{n}$ the expansion will be $n-1$. The expected expansion therefore is,

$$\frac{n-1}{n} \cdot 1 + \frac{1}{n} \cdot (n-1) = 2\frac{n-1}{n} \leq 2.$$

Hence, C_n can be 2-probabilistically approximated by a distribution of trees.

2 Metrical Task Systems

In [Bar96] Bartal introduced the notion of α-probabilistic embeddings and showed that any metric with n points and with diameter[1] Δ embeds into distributions of dominating

[1]defined formally in Definition 2.1

trees with distortion $O(\log n \log \Delta)$. He also introduced many algorithmic applications of his theorem. His result later was improved to $O(\log n)$ by [FRT03].

Definition 2.1. The diameter of a metric space is the least Δ such that the distance between any two pair of vertices is less than or equal to Δ.

$$\Delta = \sup_{x,y} d(x, y)$$

Embedding into distribution of trees has many algorithmic applications. This stems from the fact that it is usually easy to solve or obtain a good approximation for an optimization problem over the tree metric. For example it is very natural to use the divide and conquer approach with trees, as trees allow the algorithm to reduce the problem by processing one child at a time. In the following we discuss a general framework for using α-probabilistic embeddings into distribution of trees in designing randomized algorithms for certain class of online algorithms.

A task system (X, d) consists of a set X of states and a cost function d where $d(i, j)$ is the cost of changing from state i to state j [BLS92]. In a Metrical Task System (MTS), d is assumed to be a metric, i.e., it satisfies the triangle inequality and $d(x, x) = 0$, $\forall x \in X$. Requests arrive in an online fashion at every time step. Each request σ_i associates a cost with each of the states $x \in X$. An online scheduling algorithm is one that chooses the state s_i of the system at time i only knowing $\sigma_1, \sigma_2, \ldots, \sigma_i$. The aim of the algorithm is to minimize the the cost function

$$\Sigma_{i=1}^{t} \left[d(s_{i-1}, s_i) + \mathrm{cost}(\sigma_i, s_i) \right].$$

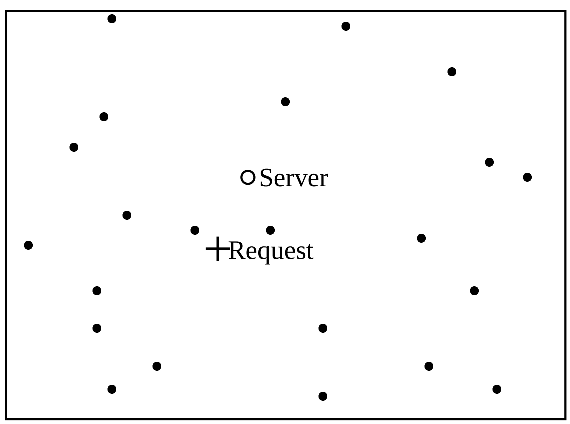

Figure 3: A representation of MTS online scheduling algorithm where each state is represented as a point in space and the cost of changing states is proportional to the distance between them.

The performance ratio is defined as the ratio of the cost of running the algorithm A relative to the optimal performance that can be achieved if the future was known.

$$\text{performance ratio } = \frac{\mathrm{cost}_A(\sigma)}{\mathrm{optimum}(\sigma)}$$

It is intuitive that simple metric space will make design of such an algorithm simple. Hence we seek embeddings that can transform the input space to a simpler space and still guarantee some bounds on the performance.

Theorem 2.2. *If a distribution of trees $\mathcal{T}$ α-probabilistically approximates d and there is an algorithm A_τ for every $\tau \in \mathcal{T}$ with performance ratio less than β, then there is a randomized online scheduling algorithm with performance ratio less than $\alpha \cdot \beta$.*

Theorem 2.2 holds true for any online MTS problem in which the performance ratio is a weighted (positive weights) sum of the distances and cost with respect to requests.

Proof. To prove the theorem, we consider a simple randomized algorithm which selects a tree τ at random from the the distribution $\mathcal{T}$ and then runs the original algorithm (with the objective function changed) on the new space τ. The performance ratio for this new randomized algorithm will be

$$\frac{E_{\tau \in \mathcal{T}}[\mathrm{cost}^M_{A_\tau}(\sigma)]}{\mathrm{optimum}(\sigma)},$$

where $\mathrm{cost}^M_{A_\tau}(\sigma)$ is the cost of running algorithm A_τ (which assumes that the distances are according to τ) while the cost is computed according to M.

From the dominance property of $\mathcal{T}$ over the original metric space $M = (X, d)$,

$$\mathrm{cost}^M_{A_\tau}(\sigma) \leq \mathrm{cost}^\tau_{A_\tau}(\sigma) \;\; \forall \tau \in \mathcal{T},$$

where $\mathrm{cost}^\tau_{A_\tau}(\sigma)$ is the cost of running the algorithm A_τ when the distances are according to τ. Also, by assumption of the theorem

$$\mathrm{cost}^\tau_{A_\tau}(\sigma) \leq \beta \mathrm{optimum}_\tau(\sigma), \forall \tau \in \mathcal{T}.$$

Therefore

$$\begin{aligned} E_{\tau \in \mathcal{T}}[\mathrm{cost}^\tau_{A_\tau}(\sigma)] &\leq \beta \cdot E_{\tau \in \mathcal{T}}[\mathrm{optimum}_\tau(\sigma)] \\ &\leq \beta\alpha \cdot \mathrm{optimum}_M(\sigma). \end{aligned}$$

□

3 Embedding to Random Trees

In this section we discuss Bartal's theorem and its improvement by Fakcharoenphol, Rao, and Talwar [FRT03]:

Theorem 3.1 ([FRT03]). *Every metric space with n points can be $O(\log n)$-probabilistically approximated by a distribution of tree metrics.*

Theorem 3.1 is a very useful result and can be applied to many optimization problems that can be defined in terms of some metric spaces. The MTS problem discussed in the previous section is a good example where this theorem can be advantageous.

Remark 3.2. All tree metrics are ℓ_1 (as proved in the previous tutorial). It is also easy to see that ℓ_1 metrics are additive, d_1, d_2 are ℓ_1 metrics implies that $d_1 + d_2$ is ℓ_1. Also, if d_i are ℓ_1 and $\lambda_i \in \mathbb{R}^+$, then $\Sigma \lambda_i d_i$ is ℓ_1 (left as an exercise).

Remark 3.3. Theorem 3.1 can be related to Bourgain's theorem [Bou85]. Since all tree metrics can be isometrically embedded into ℓ_1 and every metric space with n points can be $O(\log n)$-approximated by a distribution of tree metrics, it follows from Remark 3.2 that any metric space with n points can be $O(\log n)$-approximated by ℓ_1. This is Bourgain's result, except that it does not guarantee any bound on the dimension.

First we state the intuition of the proof of Theorem 3.1. More details are given in Section 4.

Main tool of the algorithm that constructs the distribution of trees for proving Theorem 3.1 is probabilistic decomposition. This is a randomized procedure that given a metric space (X, d) and parameter δ, decompose the metric so that

- Each cluster is of diameter at most δ.
- Probability that an two points x and y land in different clusters is at most $\leq 4\log(n)\frac{d(x,y)}{\delta}$.

Figure 4 provides some intuition for the algorithm. The big metric space is decomposed in three smaller subspaces with diameter at most δ. The decomposition process is then recursively applied on the three smaller subspaces. This process produces a rooted tree with the big metric space as the parent node and subspaces produced as children.

Trees can very well capture the clustering properties of a metric space. The points which are close to each other will have the common ancestor close to the leaf nodes and the distance in the tree metric will also be small, while points belonging to different clusters will have the common ancestor close to the root of the tree.

Example 3.4. To provide motivation for the chosen probability function, consider the ℓ_1 space in $\mathbb{R}^1$ with n points placed along a line at equal distances, as shown in Figure 5. If we remove edges independently with probability $\approx c\frac{\log n}{\delta}$, then the event of not cutting any of the δ consequent edges is $\left(1 - c\frac{\log n}{\delta}\right)^{\delta}$.

$$
\begin{aligned}
\left(1 - c\frac{\log n}{\delta}\right)^{\delta} &= \left[\left(1 - c\frac{\log n}{\delta}\right)^{\frac{-\delta}{c\log n}}\right]^{-c\log n} \\
&\approx e^{-c\log n} \\
&= \frac{e^{-c}}{n}
\end{aligned}
$$

For $c = \log_e 2$, with probability, $1 - (n-\delta)\frac{1}{2n} \approx \frac{1}{2}$, all segment of size δ will be cut. Hence the diameter of resulting subgraphs will be less then δ with probability $\geq \frac{1}{2}$.

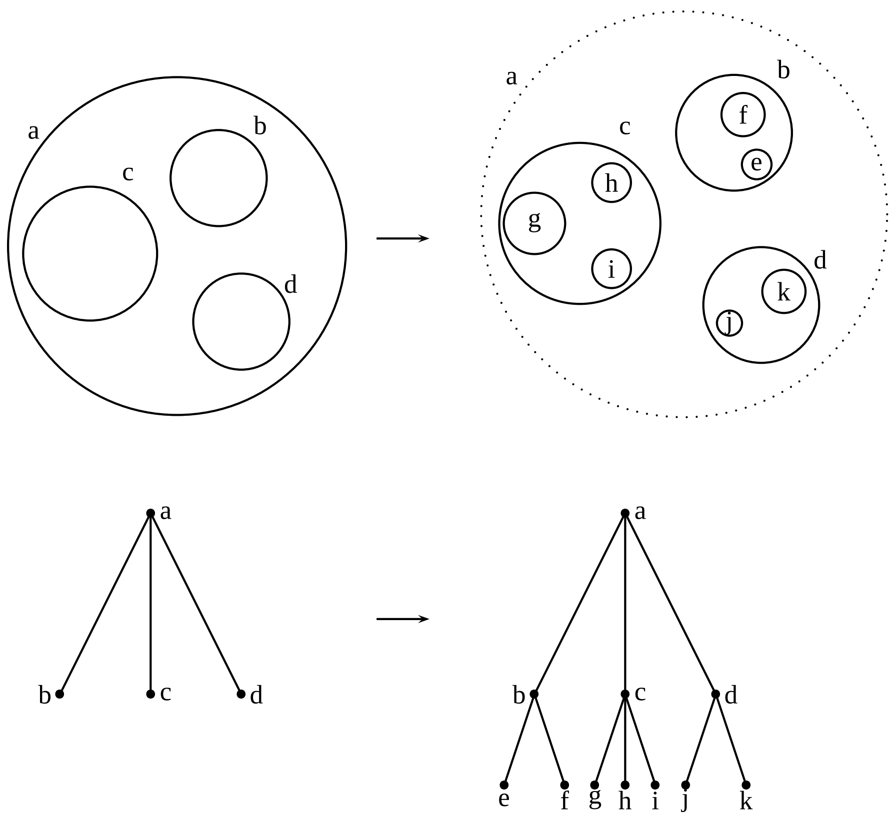

Figure 4: A big metric space decomposed in three smaller subspaces. The three subspaces are further divided and a rooted tree is constructed.

4 Proof of Theorem 3.1

As we saw in Section 3 the main tool in the proof of Theorem 3.1 is the probabilistic decomposition.

Definition 4.1. A distribution Π over the partitions of a finite metric space (X, d) is called *low-diameter solid partition* with parameter $\delta > 0$ if

1. For every partition P of X, such that $\Pi(P) > 0$, $\mathrm{weakdiam}(P) \leq \delta$
2. If P is chosen from distribution Π, then for each $x, y \in X$,

$$\Pr[P \text{ splits } x \text{ and } y] \leq \frac{4 \log n \cdot d(x, y)}{\delta}.$$

It is not obvious that such a distribution over partitions exists. But suppose it does. How can we use it to get a good probabilistic embedding of (X, d) into a mixture of trees?

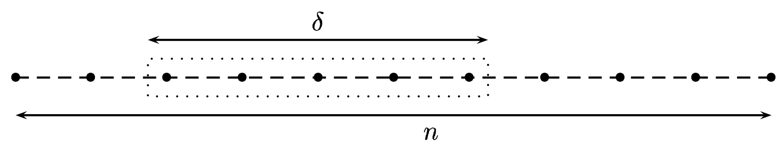

Figure 5: n points in $\mathbb{R}^1$. Edges are removed independently with probability $c\frac{\log n}{\delta}$.

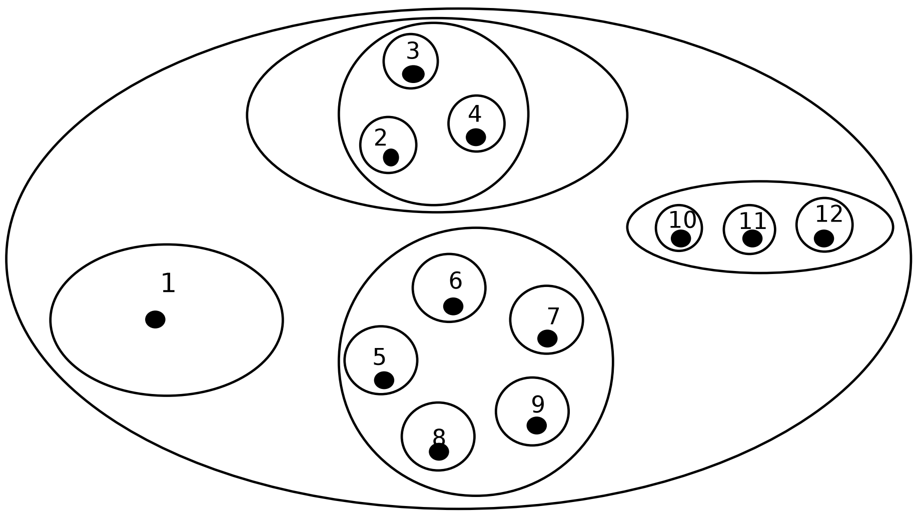

Figure 6: A recursive use of the low-diameter solid partition

Let us suppose that we can find this low-diameter solid partition for every metric space X and any choice of parameter δ.

1. Let $\delta = \mathrm{diam}(X)/2$. In that case, the partition will most likely split points whose distance is comparable to the diameter, or else the probability of the points being split is small.

2. Having got such a partition, we apply this construction recursively on each set in the partition, but using $\delta/2$ as the parameter instead.

   Then we recursively apply the this construction on each sub-partition with parameter $\delta/4$, and so on. Thus, on each recursion we divide the parameter by 2. (Note that $\delta = \mathrm{diam}(X)/2$ is fixed and does not depend on the diameter of sub-partitions.

We will get a family of laminar sets, where every two sets are either disjoint or one is contained in the other. These laminar sets correspond to a rooted tree constructed by the following recursive procedure with initial parameter $\Delta = \mathrm{diam}(X)/2$:

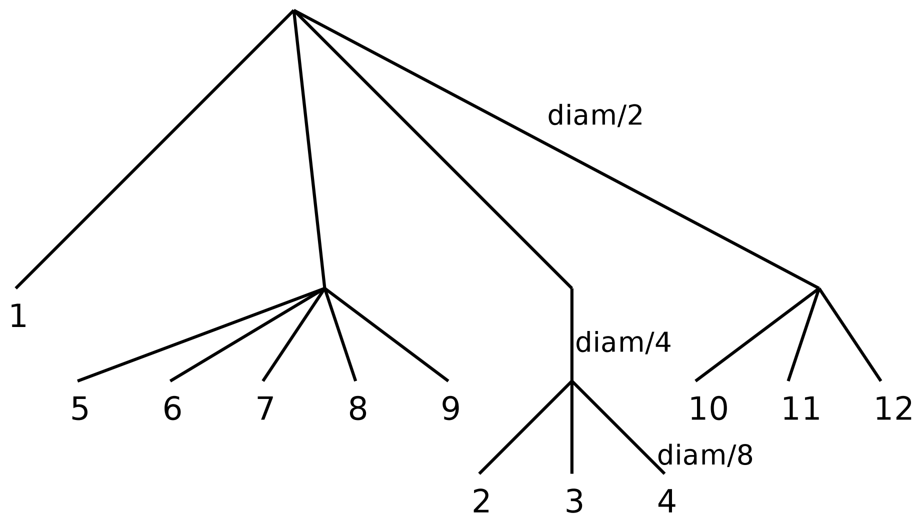

Figure 7: The tree corresponding to the laminar sets in Figure 6

1. For each recursive call: start with a partition with parameter Δ. Recursively create trees for each of the obtained sub-partitions, but with parameter $\Delta/2$.
2. Connect the roots of these trees to a new root with edge-length Δ.
3. If there is only one element in the partition, we stop the recursion.

The tree obtained by the above procedure has certain properties. In particular: The elements of X are mapped to the leaves of the tree. The height of the tree is at most

$$O\left(\log\left(\frac{\operatorname{diam}(X)}{\text{min distance in the metric space } X}\right)\right).$$

Moreover the tree is a 2-*hierarchically well-separated tree*:

Definition 4.2. A k-hierarchically well-separated tree (k-HST) is defined as a rooted weighted tree with following properties

- The edge weight from any node to each of its children is same.
- The edge weight along any path from the root to a leaf are decreasing by a factor of at least k.

Figure 8 demonstrates an example of k-HST. These trees have algorithmic importance, and both [Bar96, FRT03] embed the metrics into a distribution over the trees which are k-HST.

We shall assume without loss of generality that the smallest distance in X is 1[2]. Under this assumption, the property of dominance is always preserved.

[2] See Remark 4.5

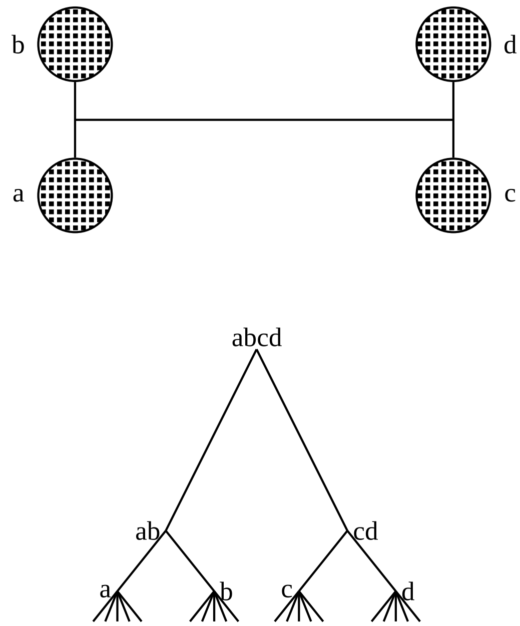

Figure 8: Hierarchical clustering. Original space contains four clusters, a, b, c and d, each containing many points. Points in clusters a and b are relatively closer to each other as compared to c and d. The tree easily captures this, as points from same cluster are siblings in the tree and points in clusters close to each other are close to each other in the tree as well. It is natural to expect the distance between the nodes ab and cd to be much larger than that between a and b. The property of k-HST makes this possible as edge lengths increase by a factor of k as we move toward the root.

Claim 4.3. *Dominance is preserved: Let τ be any tree that is constructed in the above procedure. Then $\tau(x, y) \geq d(x, y)$.*

Proof. If $d(x, y) \in (\Delta/2^j, \Delta/2^{j-1}]$, then Definition 4.1 (1) guarantees that x and y are split in the recursive step with parameter $\Delta/2^j$ or before that. Thus their tree distance is at least twice the edge from their first common ancestor to its children. Therefore their distance in the graph is at least $\Delta/2^{j-1}$. $\square$

4.1 Small Expansion

We saw that the dominance condition is satisfied. Next we need to show that the expansion is small. On the most intuitive level, the expansion is small since points that are close are unlikely to be split with large parameters. More formally, let τ be a random

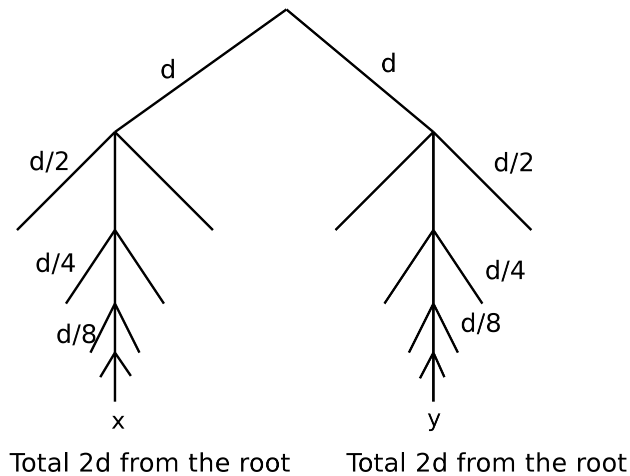

Figure 9: $\tau(x, y)$ is no more than 4Δ

tree generated by the above procedure. Let us bound the expected expansion of $\tau(x, y)$. First let us prove the following claim:

Claim 4.4. *We have*

$$\tau(x, y) \leq 4 \cdot \Delta_{x,y},$$

where $\Delta_{x,y}$ is the parameter value in which x and y are split[3].

Proof. This is due to the fact that the tree is 2-HST. The proof is best seen in Figure 9:

$$2 \sum_{i=0}^{\infty} \frac{d}{2^i} \leq 4d.$$

□

Now let us bound $\mathbb{E}[\tau(x, y)]$.

$$\begin{aligned}
\mathbb{E}[\tau(x,y)] &\leq 4 \cdot \Delta \Pr[x, y \text{ split with parameter } \Delta] \\
&+ 4 \cdot \frac{\Delta}{2} \Pr[x, y \text{ split with parameter } \Delta/2] \\
&+ 4 \cdot \frac{\Delta}{4} \Pr[x, y \text{ split with parameter } \Delta/4] \\
&\vdots \\
&+ 4 \cdot \frac{\Delta}{2^j} \Pr[x, y \text{ split with parameter } \Delta/2^j]
\end{aligned}$$

[3]So $\Delta_{x,y} = \Delta \cdot 2^{-j}$ for some j

Here j is $\lceil \log \mathrm{Diam(X)} \rceil + 2$. Note that Definition 4.1 (1) guarantees that x and y are being split with parameter $\Delta/2^i$ for some $i \leq j$.

Now the second property of the low-diameter solid partition (Definition 4.1 (2)) will be crucial: If the partition has parameter δ,

$$\Pr(x, y \text{ are split}) \leq 4 \log n \cdot d(x,y)/\delta.$$

Therefore, we can upper-bound $\Pr[x, y \text{ split with parameter } \Delta/2^j]$. Applying this we get

$$\begin{aligned}
\mathbb{E}[\tau(x,y)] &\leq 4 \cdot \Delta \cdot 4(\log n) \frac{d(x,y)}{\Delta} \\
&+ 4 \cdot \frac{\Delta}{2} \cdot 4(\log n) \frac{d(x,y)}{\Delta/2} \\
&+ 4 \cdot \frac{\Delta}{4} \cdot 4(\log n) \frac{d(x,y)}{\Delta/4} \\
&\vdots \\
&+ 4 \cdot \frac{\Delta}{2^j} \cdot 4(\log n) \frac{d(x,y)}{\Delta/2^j}
\end{aligned}$$

Therefore

$$\mathbb{E}[\tau(x,y)] = O\big(\log n \cdot \log\left(\mathrm{Diam}(X)\right)\big),$$

where the $\log(\mathrm{Diam}(X))$ is the number of terms in the summation. Note that we have the extra term $\log \mathrm{Diam}(X)$, which will be taken care of later.

Remark 4.5. Remember that we scaled the metric space so that the minimum distance between every two points is at least 1. So here $\mathrm{Diam}(X)$ is not the diameter of the original metric and in fact it is the diameter of the original metric divided by the minimum distance in the original metric.

5 The Low-Diameter Solid Partition

This partition is based on the work in [CKR01]. We would like to have a deterministic constraint that the diameter is always less than δ. We will use a randomized construction that in addition will split close-by points only infrequently. The idea is that we pick a radius, and then start placing balls one on-top of another with this radius.

We place one ball, and this is one set in the partition. Then we place another ball, and let the *new*[4] points covered by this ball to be the second set containing and so on. More formally,

Algorithm 5.1.

[4]If the set we place intersects another set that was placed before, the points remain in the "older" set.

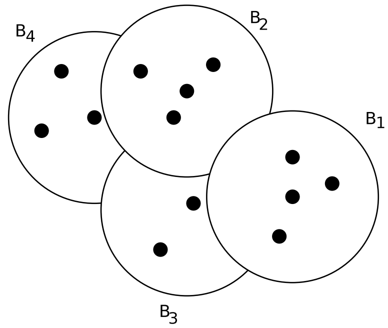

Figure 10: The making of the partition. Each ball is of diameter δ

1. Pick a random permutation σ of $\{1, \ldots, n\}$, and, independently, pick $R \in [\delta/4, \delta/2]$ uniformly at random.
2. Define $B_i := B(j, R) \setminus \bigcup_{k \leq_\sigma j} B(k, R)$.

Notice that it might be the case that $j \notin B_j$ and $B_j = \emptyset$.

5.1 Why does this construction work?

It is evident that the diameter of this partition is less than δ, since R, the radius of the ball is less than $\delta/2$.

What is the probability that a pair x, y is split? For the analysis, number the rest of the points by their relative distance to the set $\{x, y\}$. Thus, the closest point will be called 1 and the second closest will be called 2. If two points are of the same distance to $\{x, y\}$, then it does not matter how we denote them.

When is the first time x and y are split?

1. A ball can be disjoint from $\{x, y\}$, in which the points are not split.
2. A ball can contain $\{x, y\}$, so the points are not split.
3. A ball contains precisely one of $\{x, y\}$, in which case the points are split.

Definition 5.2. For a fixed permutation σ, we say that j *s*ettles x, y if $\{x, y\} \cap B_j \neq \emptyset$.

The probability that a pair x, y is split,

$$\Pr\left[\bigcup_i x, y \text{ is split by } i\right] = \sum_i \Pr\left[x, y \text{ is split by } i\right].$$

Note that we have equality since these events are disjoint; It is not possible for points x and y to be split by both $i \neq j$, for example.

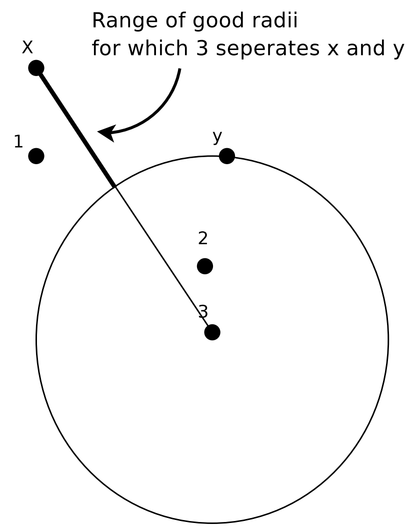

Figure 11: The radius must be in the right range, between $d(j, x)$ and $d(j, y)$. With a good radius, point 2 will settle points x and y, so if point 3 is to split these points, it must come before point 2.

5.2 The main analysis

Let us start by an example. Consider $\Pr[\text{point 3 splits } x \text{ and } y]$, where the points are schematically drawn in Figure 11.

1. The radius should be of the right size.
2. If 2 is prior to 3 in σ, then 3 cannot split x and y. This is because the radius is the same for both points 2 and 3. Therefore, if the radius is such that it is possible for point 3 to split x and y, then the radius is such that point 2 can either split or contain both x and y.

Thus j settles $\{x, y\}$ and separates them iff j is the first among $\{1, \ldots, j\}$ according to σ and the radius is in $[d(j, x), d(j, y)] \cup [d(j, y), d(j, x)]$. Therefore

$$\begin{aligned} & \Pr[j \text{ settles } \{x, y\} \text{ and separates them}] \\ = \; & \frac{1}{j} \cdot \Pr[R \in [d(j, x), d(j, y)] \cup [d(j, y), d(j, x)]] \\ = \; & \frac{1}{j} \cdot \frac{|d(x, j) - d(y, j)|}{\delta/4} \\ \leq \; & \frac{1}{j} \cdot \frac{d(x, y)}{\delta/4}, \end{aligned}$$

where the last equality follows from the triangle inequality. The denominators are $\delta/4$, since we sampled uniformly from $[\delta/4, \delta/2]$.

The probability that points x and y are split is thus less than

$$\sum_{j=1}^{n-2} \frac{1}{j} \cdot \frac{d(x,y)}{\delta/4} \leq 4 \log n \cdot \frac{d(x,y)}{\delta},$$

as required from the low diameter solid partition. But we are not done yet, since we have not handled the $\log \mathrm{Diam}(\mathrm{X})$ term that arises.

5.3 Improving the result

Recall that we used the following identity:

$$\Pr\left[\bigcup_i x, y \text{ is split by } i\right] = \sum_i \Pr\left[x, y \text{ is split by } i\right],$$

and then bounded each term on the right hand side separately. However, for any given value of the parameter δ,

$$\Pr\left[x \text{ and } y \text{ is split by } i\right] = 0,$$

if **(i)** $\delta/2 < d(i, \{x,y\})$ since $\{x,y\}$ will not be settled, or **(ii)** $d(i,\{x,y\}) < \delta/8$, since $\{x,y\}$ will both belong to the same set, and thus will not be separated. So it is sufficient to sum over a smaller number of terms so the bound becomes

$$Pr(x \text{ and } y \text{ are split}) \leq \sum_{j:\delta/8 \leq d(j,\{x,y\}) \leq \delta/2} \frac{1}{j} \cdot \frac{d(x,y)}{\delta/4}$$

Let us see what happens once we apply the above bound to $\tau(x,y)$. Recall the equations

$$\begin{aligned}
\mathbb{E}[\tau(x,y)] \quad &\leq \quad 4 \cdot \Delta \mathrm{Pr}[x, y \text{ split with parameter } \Delta] \\
&+ \quad 4 \cdot \frac{\Delta}{2} \mathrm{Pr}[x, y \text{ split with parameter } \Delta/2] \\
&+ \quad 4 \cdot \frac{\Delta}{4} \mathrm{Pr}[x, y \text{ split with parameter } \Delta/4] \\
&\vdots \\
&+ \quad 4 \cdot \frac{\Delta}{2^j} \mathrm{Pr}[x, y \text{ split with parameter } \Delta/2^j]
\end{aligned}$$

By applying the improved bound, we obtain

$$
\begin{aligned}
\mathbb{E}[\tau(x,y)] \quad \le \quad & 4 \cdot \Delta \cdot 4d(x,y)/\Delta \sum_{j:\Delta/8 \le d(j,\{x,y\}) \le \Delta/2} \frac{1}{j} \\
+ \quad & 4 \cdot \frac{\Delta}{2} \cdot 4d(x,y)/(\Delta/2) \sum_{j:(\Delta/2)/8 \le d(j,\{x,y\}) \le (\Delta/2)/2} \frac{1}{j} \\
+ \quad & 4 \cdot \frac{\Delta}{4} \cdot 4d(x,y)/(\Delta/4) \sum_{j:(\Delta/4)/8 \le d(j,\{x,y\}) \le (\Delta/4)/2} \frac{1}{j} \\
\vdots \quad & \\
+ \quad & 4 \cdot \frac{\Delta}{2^j} \cdot 4d(x,y)/(\Delta/2^j) \sum_{j:(\Delta/2^j)/8 \le d(j,\{x,y\}) \le (\Delta/2^j)/2} \frac{1}{j}
\end{aligned}
$$

In the above bound, we sum each $1/j$ at most twice. We conclude that

$$\mathbb{E}[\tau(x,y)] \le 32 \cdot d(x,y) \sum_{j \le n} \frac{1}{j} = O(d(x,y) \log n)$$

and thus the distortion is at most $O(\log n)$.

References

[Bar96] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *FOCS*, pages 184–193, 1996.

[BLS92] Borodin, Linial, and Saks. An optimal on-line algorithm for metrical task system. *JACM: Journal of the ACM*, 39, 1992.

[Bou85] J. Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 1985.

[CKR01] Gruia Calinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001)*, pages 8–16, Philadelphia, PA, 2001. SIAM.

[FRT03] Fakcharoenphol, Rao, and Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2003.

[RR98] Rabinovich and Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *GEOMETRY: Discrete & Computational Geometry*, 19, 1998.