

# Finite Metric Spaces & Their Embeddings: Introduction and Basic Tools

Manor Mendel, CMI, Caltech

## 1 Finite Metric Spaces

**Definition of (semi) metric.**  $(M, \rho)$ :  $M$  a (finite) set of points.  $\rho$  a distance function satisfying  $\forall x \in M \rho(x, x) = 0$ ,  $\forall x, y \in M \rho(x, y) = \rho(y, x)$ ,  $\forall x, y, z \in M \rho(x, z) \leq \rho(x, y) + \rho(y, z)$ .

**CS motivation.** Finite metric spaces arise naturally in combinatorial objects, and algorithmic questions. For example, as the shortest path metrics on graphs. We will also see less obvious connections.

**Properties of finite metrics.** The following properties have been investigated: Dimension, extendability of Lipschitz and Hölder functions, decomposability, Inequalities satisfied by the metric, short representations, additive distortion of embedding, (multiplicative) distortion of embeddings.

We will focus on the last property.

**Embedding.** A mapping  $f : (M, \rho) \rightarrow (H, \nu)$  of a metric space  $M$  into a host metric space  $H$ , that (hopefully) preserves the geometry of  $M$  (usually distances).

**Distortion of embedding.** The distortion of  $f : (M, \rho) \rightarrow (H, \nu)$  is the least  $K \geq 1$  for which exists  $C > 0$  such that  $\forall x, y \in M$ ,

$$C \cdot \rho(x, y) \leq \nu(f(x), f(y)) \leq K \cdot C \cdot \rho(x, y). \quad (1)$$

The distortion of  $f$  is denoted by  $\text{dist}(f)$ .  $C$  is a scaling factor. Another way to define the distortion: The *Lipschitz constant* of a mapping  $f : (M, \rho) \rightarrow (H, \nu)$  is

$$\|f\|_{\text{Lip}} = \max_{\substack{x, y \in M \\ \rho(x, y) > 0}} \frac{\nu(f(x), f(y))}{\rho(x, y)}.$$

Then  $\text{dist}(f) = \|f\|_{\text{Lip}} \cdot \|f^{-1}\|_{\text{Lip}}$ .

**Example of hosts spaces.**  $\ell_p$  is the set of real valued sequences  $(x_i)_{i \in \mathbb{N}}$  for which  $\sum_i |x_i|^p < \infty$ . The  $\ell_p$  norm is defined as  $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$ . Of particular interest are the cases  $p = 1, 2, \infty$ .  $\ell_2$  is called Hilbert space. Euclidean spaces are finite subspaces of Hilbert space.

**Non-embeddability example.** Consider the (SP) metric of the graph  $C_4$ , a cycle on four nodes. Is there an isometric embedding of  $C_4$  in Hilbert space? No: Denote the vertices on the  $C_4$  by  $a_1, \dots, a_4$ . Suppose an isometric embedding exists. Note that  $\rho(a_1, a_3) = \rho(a_1, a_2) + \rho(a_2, a_3)$ , hence the triangle inequality holds with equality, which means (for Euclidean spaces) that  $f(a_2)$  is in the middle of the segment  $[f(a_1), f(a_3)]$ . Analogously,  $f(a_4)$  is in the middle of the segment  $[f(a_1), f(a_3)]$ . Hence  $f(a_2) = f(a_4)$ , a contradiction. [FIGURE]

Interestingly, the natural embedding of  $C_4$  as a square in the plane is the best embedding in Hilbert space, and its distortion is  $\sqrt{2}$ .

**Example of an algorithmic application.** Assume we are given  $n$ -point set in  $X \subset \ell_1^d$ , and we would like to compute  $\text{diam}(X) = \max_{x, y \in X} \|x - y\|_1$ . A straightforward algorithm is to compute the distance between all pairs in  $X$  and taking the maximum. This algorithm takes  $O(dn^2)$  time. We next show an improved algorithm when  $n \gg d$ . We first exhibit an isometric embedding  $f : X \rightarrow \ell_\infty^{d'}$ , where  $d' = 2^d$ . We then compute  $\text{diam}(f(X))$ . The second follows from

$$\begin{aligned} \max_{x, y \in X} \|f(x) - f(y)\|_\infty &= \max_{x, y \in X} \max_j |f(x)_j - f(y)_j| = \\ &= \max_j \max_{x, y \in X} |f(x)_j - f(y)_j| = \max_j (\max_{x \in X} f(x)_j - \min_{x \in X} f(x)_j). \end{aligned}$$

Hence, in  $\ell_\infty^{d'}$  the diameter can be computed in  $O(nd')$  time. The embedding  $f : \ell_1^d \rightarrow \ell_\infty^{\{\pm 1\}^d}$  is defined as follows:  $(f(x_1, \dots, x_d))_\sigma = \sum_j \sigma_j x_j$ , where for  $\sigma \in \{\pm 1\}^d$ . Computing this embedding can be clearly carried in  $O(ndd')$  time (and actually also in  $O(nd')$  time). Note that this embedding has additional nice properties: isometric, linear, and of the whole  $\ell_1^d$  space. In general, it will be rare to encounter embeddings with all these properties.

## 2 Bourgain Embedding

As we have just seen, not all metrics are isometrically embedded in Hilbert space. In fact, it was shown that there exist  $n$ -point metrics for which the Euclidean distortion is  $\Omega(\log n)$  (for any  $n > 1$ ). An important theorem of Bourgain states that this bound is tight.

**Thm. 1.** *Any  $n$ -point metric space can be embedded in  $\ell_2$  with distortion  $O(\log n)$ .*

It has been shown that Bourgain theorem implies an efficient randomized algorithm for the embedding, and that the bound also holds for embedding in  $\ell_p$ ,  $p \geq 1$ .

**The sparsest cut problem.** An algorithmic application of Bourgain's theorem: Let  $G = (V, E)$  be a graph. A cut in  $G$  is a partition of  $V$  into two nonempty subsets  $A$  and  $B = V \setminus A$ . The density of the cut  $(A, B)$  is  $\frac{e(A, B)}{|A| \cdot |B|}$ , where  $e(A, B)$  is the number of edges crossing the cut. Given  $G$ , we look for a cut of the smallest possible density. This problem is NP-hard, and we now present an efficient algorithm that produces a cut of density at most  $O(\log |V|)$  the density of the sparsest cut.

We reformulate the problem using cut metrics. A cut metric  $\tau_A$  on  $V$  corresponds to the cut  $(A, V \setminus A)$  and is defined as  $\tau_A(u, v) = 1$  when  $u$  and  $v$  are not on the same side of the cut  $(A, V \setminus A)$ , and 0 otherwise. The sparsest cut problem asks to find

$$\min_{\tau} \frac{\sum_{(u,v) \in E} \tau(u, v)}{\sum_{u,v \in V} \tau(u, v)},$$

where the minimum is taken over all cut metrics.

We next relax the problem by allowing  $\tau$  to be arbitrary metric. By scaling we obtain the following linear programming problem

$$\begin{aligned} & \min \sum_{(u,v) \in E} \tau((u, v)) \\ & \text{Subject to} \\ & \sum_{u,v \in V} \tau(u, v) \geq 1 \\ & \tau \text{ satisfies the } \Delta \text{ ineq.} \\ & \tau \geq 0 \end{aligned}$$

This LP can be solved in polynomial time. The value of the optimal solution is at most the sparsest cut, but it is not necessarily a cut. We round it by embedding the resulting metric  $\tau_0$  into  $\ell_1^m$  ( $m = O(\log^2 n)$ ) using Bourgain's theorem. We obtain a metric  $\tau_1$  in  $\ell_1^m$ , satisfying

$$\tau_1 \leq \tau_0 \leq O(\log |V|) \tau_1$$

An  $\ell_1$  metric can be written as a positive sum of cut metrics (i.e., it is in the cut-cone). Thus,

$$\tau_1 = \sum_i \alpha_i \theta_i.$$

$\alpha_i > 0$ , and  $\theta_i$  is a cut metric (such a decomposition is easily computed). Note that

$$\begin{aligned} O(\log |V|) \cdot \frac{\sum_{e \in E} \tau_0(e)}{\sum_{u,v \in V} \tau_0(u, v)} &\geq \frac{\sum_{e \in E} \tau_1(e)}{\sum_{u,v \in V} \tau_1(u, v)} = \\ & \frac{\sum_i \alpha_i (\sum_{e \in E} \theta_i(e))}{\sum_i \alpha_i (\sum_{u,v \in V} \theta_i(u, v))} \geq \min_i \frac{\sum_{e \in E} \theta_i(e)}{\sum_{u,v \in V} \theta_i(u, v)}. \end{aligned}$$

The RHS is easily computed and its value is at most  $O(\log |V|)$  the value of the LP (which is at most the density of the sparsest cut).

### 3 Embedding in probabilistic trees

**Tree metrics.** A tree metric is a metric given by the SP on a weighted tree. A finite ultrametric is a metric induced on the leaves of a rooted tree  $T$ , where each node  $v$  of the tree has label  $\Delta(v)$ . The distance between two leaves  $x, y$  is defined by  $\Delta(\text{lca}(x, y))$ . [Figure] Ultrametric is a special type of tree metric.

Since tree metrics, and in particular ultrametrics are relatively simple, it's usually simpler to devise algorithms to these classes than to general metrics. Unfortunately simple metrics (like cycles) do not embed well in tree metrics.

**Embedding into probabilistic trees.** A convex combination of metrics is also a metric. Efficient embedding into a convex combination allows for a randomized algorithm to sample a tree metric from the distribution and apply the tree algorithm. However, for cost approximation algorithms, plain distortion is usually not enough. An embedding  $f : (X, \rho) \rightarrow \sum_i \alpha_i \nu_i$  into convex combination of metrics  $\nu_i$  is called probabilistic embedding if  $\rho \leq \nu_i$ . The distortion of the embedding is the minimum  $\beta \geq 1$  such that  $\sum_i \alpha_i \nu_i \leq \beta \cdot \rho$ .

**Thm. 2.** Any  $n$  point metric can be probabilistically embedded in a convex combination of ultrametrics, with distortion  $O(\log n)$ .

**Algorithmic Example: Metric  $k$ -median.** Algorithmic clustering problem: Given  $n$ -point metric space  $(M, \rho)$ , and an integer  $k$ , find  $k$  "centers"  $C = \{c_1, \dots, c_k\} \subset M$  that minimizes the expression

$$\min_{\substack{C \subset M \\ |C| \leq k}} \sum_{x \in M} \rho(x, C),$$

where  $\rho(x, C) = \min_{y \in C} \rho(x, y)$ . A natural clustering problem.

**An algorithm for ultrametrics.** We may assume that the ultrametric tree is binary (by splitting a vertex with more than two children into a chain with the same labels)

Apply a Dynamic Programming algorithm that computes for each vertex  $v$ ,  $k \in \{0, \dots, n\}$ , compute  $\text{COST}(v, k)$ , which is the optimal cost for serving the vertices in the subtree rooted at  $v$  using  $k$  centers, The algorithm works bottom-up.

Let  $v$  be vertex. Denote by  $l(v)$  the number of leaves in the subtree rooted at  $v$ . If  $v$  is a leaf then  $\text{COST}(v, 0) = \infty$ , and  $\text{COST}(v, k) = 0$  for  $k \geq 1$ . If  $v$  has two children  $v_1, v_2$ , then

$$\text{COST}(v, k) = \min \left\{ \min_{\substack{k_1, k_2 \geq 1 \\ k_1 + k_2 = k}} \text{COST}(v_1, k_1) + \text{COST}(v_2, k_2), \right. \\ \left. \text{COST}(v_1, k) + l(v_2)\Delta(v), \text{COST}(v_2, k) + l(v_1)\Delta(v) \right\}.$$

**A randomized alg' for the  $k$ -median**

$\text{Med}(M, k)$ .

Sample an ultrametric  $T$  from the embedding of  $M$  into probabilistic trees.

Find an  $r$  approximation of the  $k$ -median problem on trees.  
 Output this solution.

We claim that this is  $O(\log n)r$  approximation of the optimal solution. To see this, let  $O_M$  be the optimal solution for  $M$ , and let  $A_T$  the solution of the approximation alg' for  $T$ . Then

$$\forall T \text{ cost}_M(A_T) \leq \text{cost}_T(A_T),$$

since the distance in  $T$  dominates the distances in  $M$ . Also,

$$\text{cost}_T(A_T) \leq r \cdot \text{cost}_T(O_M),$$

since  $A$  is an  $r$ -approximation to the optimal cost in  $T$  (which is at most  $\text{cost}_T(O)$ ). Since  $O$  doesn't depend on the sampling of the trees, we have by linearity of expectation,

$$\mathbb{E}_T[\text{cost}_T(O_M)] \leq O(\log n)\text{cost}_M(O_M).$$

Hence

$$\mathbb{E}_T[\text{cost}_M(A_T)] \leq O(r \log n)\text{cost}_M(O_M).$$

and this is what we have looked for, since the LHS is the expected cost of the randomized algorithm.

**Remarks.**

1. This method works for many cost minimization problems on metrics in which the cost is conically depends on the distances.
2. This algorithm was the first non-trivial approximation algorithm for the  $k$ -median in general finite metrics. By now, a constant approximation algorithm is known.

## 4 The Johnson-Lindenstrauss Lemma.

**Thm. 3.** *For any  $n$ -point in  $\ell_2^d$ , there exists a linear transformation  $F : \ell_2^d \rightarrow \ell_2^{16\epsilon^{-2} \log n}$ .*

*Even more, there exists a probability distribution  $\mathcal{D}$  of linear transformations  $F : \ell_2^d \rightarrow \ell_2^{16\epsilon^{-2} \log n}$ , such that for any  $n$  point set  $X \subset \ell_2^d$ ,*

$$\Pr_{F \in \mathcal{D}} [\text{dist}(F|_X : X \rightarrow \ell_2^d) \leq 1 + \epsilon] \geq 0.9$$

**Algorithmic application.** Consider computing the diameter of  $n$  points in  $\ell_2^d$ , where  $d$  is large (say,  $d = n - 1$ ). The straightforward algorithm takes  $O(dn^2)$  time. By applying dimension reduction first, we obtain an algorithm that computes  $1 + \epsilon$  approximation to the diameter, and the running time is  $O(nd \log n + n^2 \log n)$ .

**Remark.** Better algorithms are known, with running time  $O(n^{2-\epsilon} + dn)$ .

## 5 A proof of a variant of Bourgain embedding

We will prove

**Thm. 4.** *For any  $b \in \mathbb{N}$ , any  $n$ -point metric  $(M, \rho)$  can be embedded into  $\ell_\infty^d$  with distortion  $c = 2b - 1$ , for  $d = O(bn^{1/b} \log n)$*

It implies  $O(\log n)$  distortion embedding in  $\ell_\infty^{O(\log^2 n)}$ , which implies  $O(\log^2 n)$  embedding in  $\ell_2$ . The proof is somewhat simpler, but the technique is the same.

We start with an isometric embedding  $f : M \rightarrow \ell_\infty^n$ . We index the coordinates of  $\ell_\infty^n$  using the points of  $M$ , and define  $f(y)_x = \rho(x, y)$ . Then

- No shrinkage of distances:  $\|f(x) - f(y)\|_\infty = \max_{z \in M} |\rho(x, z) - \rho(y, z)| \geq \rho(x, y) - \rho(y, y) = \rho(x, y)$ .
- No expansion of distances: By the triangle inequality  $|\rho(z, y) - \rho(z, x)| \leq \rho(x, y)$ , so  $\|f(x) - f(y)\|_\infty = \max_{z \in M} |\rho(x, z) - \rho(y, z)| \leq \rho(x, y)$ .

In the general case, the coordinates will be indexed by subsets of  $M$ , and  $f_A(y) = \rho(A, y)$ . The no expansion argument (using the triangle inequality) still holds, but the no shrinkage property will hold only approximately.

The idea: Find (randomly) subsets  $A \in \mathcal{A}$  such that for every pair  $x, y \in M$ , there exists a radius  $r$  and a set  $A \in \mathcal{A}$  for which  $B(x, r) \cap A \neq \emptyset$ , whereas  $B(y, r + \rho(x, y)/c) \cap A = \emptyset$  [FIGURE]. In this case  $\rho(A, x) - \rho(A, y) \geq \rho(x, y)/c$ , as needed.

**Construction of  $\mathcal{A}$ .** We set  $p_j = n^{-j/b}$ . For each  $j \in \{1, \dots, b\}$ , we construct  $10 \log nn^{1/b}$  sets, each by sampling points independently with probability  $p_j$  (this gives  $O(bn^{1/b} \log n)$  coordinates).

**Analysis.** Fix a pair of points  $x, y \in M$ . Denote by  $b_i(x) = B(x, i \cdot d(x, y)/c)$ , and  $b_i(y) = B(y, i \cdot d(x, y)/c)$ . By the pigeon-hole principle, there must be  $i_0, j_0$  for which

$$|b_{i_0}(y)| \geq n^{j_0/b}, \text{ and } |b_{i_0+1}(x)| \leq n^{(j_0+1)/b},$$

or

$$|b_{i_0}(x)| \geq n^{j_0/b}, \text{ and } |b_{i_0+1}(y)| \leq n^{(j_0+1)/b},$$

(say the first case happened).

Consider a set  $A$  chosen with probability  $p_{j_0+1}$ .

$$\Pr[A \cap b_{i_0+1}(x) = \emptyset] \geq (1 - p_{j_0+1})^{n^{(j_0+1)/b}} \approx e^{-1}.$$

$$\Pr[A \cap b_{i_0}(y) \neq \emptyset] \geq 1 - (1 - p_{j_0+1})^{n^{j_0/b}} \approx 1 - e^{-n^{-1/b}} \approx n^{-1/b}.$$

Since  $b_{i_0+1}(x)$  and  $b_{i_0}(y)$  are pairwise disjoint, the two events are independent, and therefore they happen simultaneously with probability  $\Theta(n^{-1/b})$ . Since we repeat this experiment  $O(n^{1/b} \log n)$  times, the probability that one of the sets  $A \in \mathcal{A}$  will satisfies both events is  $1 - 0.1 \cdot n^{-2}$ . Hence with probability 0.9, all pairs has a set  $A \in \mathcal{A}$  which satisfies both events, and this set catches  $1/c$  fraction of the distance between the pair, as needed.

## References

- [1] A. Gupta, and R. Ravi. Algorithmic Applications of Metric Embeddings. A course given in CMU, 2003.
- [2] P. Indyk. Algorithmic Aspects of Geometric Embeddings FOCS 2001
- [3] P. Indyk, J. Matoušek. Low distortion embedding of finite metric spaces. Chapter 8 in the Handbook of Discrete and Computational Geometry, second edition, 2004.
- [4] J. Matoušek. Lectures on Discrete Geometry Chapter 15.