CS-67720 Metric Embedding Theory and Its Algorithmic Applications

Lecture 13

Lecturers: Yair Bartal, Nova Fandina *Fall 2018/19*
For comments contact fandina@gmail.com

13.1 Algorithmic Applications of Metric Embeddings

Thus far our focus was on understanding the main results of low distortion metric embedding theory. Let us now complete the review of the field with a few algorithmic applications.

13.1.1 Approximation Algorithms by Embedding with Distortion

The obvious application of high quality (i.e., low distortion) embedding is in design of approximation algorithms. The idea is simply to translate a problem defined over a "complicated"' metric space into a problem defined over a simpler space. Then, (approximately) solve the problem over the simpler space (in a simple way, hopefully) and translate the solution back to get an approximate solution to the original problem. The distortion of the embedding exhibits the quality of the approximation. This intuition is formalized in the following claim.

Claim 13.1. *Let $f : X \to Y$ be a non-contractive embedding with distortion α. Let P be a minimization problem defined on X, and let $w : 2^{\binom{X}{2}} \to \mathbb{R}^+$ be its linear objective function: for a given $S \subseteq \binom{X}{2}$, $w(S) = \sum_{(x_i,x_j)\in S} c_{ij} \cdot d(x_i, x_j)$, $c_{ij} \geq 0$. If P allows approximation on Y, with approximation ratio β, then P allows approximation on X, with approximation ratio $\alpha \cdot \beta$.*

Proof. Let A_Y be an approximation algorithm for P on Y, and let β be its approximation ratio. Given an input to P, $\sigma_X \subseteq X$, $\sigma_X = \{\sigma_{x_i}\}$ (we assume that the input is a subset of X) denote by $\sigma_Y = \{f(\sigma_{x_i})\}$. An approximation algorithm A_X for P on X is as follows: run $A_Y(\sigma_Y)$; return $f^{-1}(A_Y(\sigma_Y))$. To prove that the above algorithm is an $\alpha \cdot \beta$ approximation to P on X we have to show that $w(A_X(\sigma_X)) \leq \alpha\beta \cdot w(OPT_X(\sigma_X))$, where $OPT_X(\sigma_X)$ is the optimal solution on the input $\sigma_X \subseteq X$.

Since f is a non-contractive and the objective function w is linear on the distances with non-negative coefficients, we have: $w(A_X(\sigma_X)) \leq w(A_Y(\sigma_Y)) \leq \beta \cdot w(OPT_Y(\sigma_Y))$. In addition,

$$w(OPT_Y(\sigma_Y)) \leq w(f(OPT_X(\sigma_X))) \leq^{distortion\ \alpha} \leq \alpha \cdot w(OPT_X(\sigma_X)).$$

Therefore, $w(A_X(\sigma_X)) \leq \beta w(OPT_Y(\sigma_Y)) \leq \beta\alpha \cdot w(OPT_X(\sigma_X))$. □

Almost the same proof holds in the probabilistic setting:

Claim 13.2. *Let $f : X \to S$ be a probabilistic embedding with distortion α and P be a minimization problem, with linear objective function on the distances, with non negative coefficients. If P admits a β-approximation algorithms on each metric space of collection S, then P admits an approximation algorithm on X, such that $E[weight\ of\ solution\ on\ X] \leq \alpha\beta \cdot\ weight(OPT_X)$.*

13.1.2 The Multi-Cut Problem

The multi-cut probem is defined as follows. For a *complete* graph $G(V, E)$, a weight function $c(u, v) : E \to \mathbb{R}^+$, and k pairs $(s_i, t_i) \subseteq \binom{V}{2}$, find a minimal weight subset of edges $E' \subseteq E$, such that removing it from the graph will disconnect every pair (s_i, t_i). Namely, after removing the edges E', the new graph does not have a path connecting s_i and t_i, for each i. The general version of the problem does not assume the graph is complete.

The problem is given to approximation within factor $O(\log k)$ [4]. We show an approximation factor $O(\min\{\log n, \log(\Phi(c))\})$, where $\Phi(c) = \max c_{uv} / \min c_{uv}$. Note that the approximation factor can be a constant, for weights close the uniform. We will use the following lemma, which is basically a simple corollary of the probabilistic embedding theorem into k-HST's.

Lemma 13.3. *Let (X, d_X) be any n-point metric space. There is a probabilistic embedding f of X into a distribution of $k-HST$'s, such that for all weight functions $c : \binom{X}{2} \to \mathbb{R}^+$: $E_f\left[\sum_{(u,v)\in\binom{X}{2}} c(u, v) d_Y(f(u), f(v))\right] = O(\log n) \sum_{(u,v)\in\binom{X}{2}} c(u, v) d_X(u, v)$.*

The proof is immediate, by the linearity of expectation.

The above lemma implies an existence of one k-HST tree, with $O(\log n)$ approximation to the weighted sum of the distances. Namely, given any weight function c, here exists an ultrametric Y and embedding $f : X \to Y$, such that

$$\sum_{(u,v)\in\binom{X}{2}} c(u, v) d_Y(f(u), f(v)) = O(\log n) \sum_{(u,v)\in\binom{X}{2}} c(u, v) d_X(u, v).$$

In [1], authors proved a more precise bound on the approximation factor $O(\min\{\log n, \log(\Phi(c))\})$.

The Approximation Algorithm for the Multi-Cut Problem. We formulate the following integer linear program. For each $(u, v) \in \binom{V}{2}$, define a variable $\tau_{uv} = 1$ iff edge (u, v) is chosen to be in the cut E'. For each path p_i^j connecting a pair (s_i, t_i), the variables should satisfy $\sum_{(u,v)\in p_i^j} \tau_{u,v} \geq 1$. In addition, we will require for all $u, v, w \in V$ to satisfy $\tau_{uv} \leq \tau_{uw} + \tau_{wv}$. Note, that this constraint doesn't affect the solution of the ILP, i.e. any feasible solution is such that the triangle inequality holds (in any feasible solution there are no triangles with edge lengths $0, 0, 1$). We add this constraint in order to obtain a metric space. Thus, the integer optimization program is given by:

$$\begin{aligned}
\min_{\tau} \quad & \sum_{(u,v)\in\binom{V}{2}} c(u, v)\tau_{u,v} \\
s.t. \quad & \sum_{(u,v)\in p_i^j} \tau_{u,v} \geq 1, \quad \forall p_i^j, \\
& \tau_{u,v} \leq \tau_{u,w} + \tau_{w,v} \quad \forall u, v, w \in V, \\
& \tau_{u,v} \in \{0, 1\}, \quad \forall (u, v) \in \binom{V}{2}.
\end{aligned}$$

Solve the relaxation of the above IP (i.e. $\tau_{u,v} \geq 0$), and let $\hat{\tau}_{u,v}$ denote the fractional optimal solution of the relaxed linear program. Consider the metric space $(V, \hat{\tau})$ (remove the edges with $\hat{\tau}_{u,v} = 0$). By Lemma 13.3, there is an ultra-metric tree (T, d_T), such that

$$\sum_{(u,v)} c(u,v) d_T(u,v) \leq O(\min\{\log n, \log(\Phi(c))\}) \sum_{(u,v)} c(u,v)\hat{\tau}_{u,v} \leq O(\min\{\log n, \log(\Phi(c))\}) \cdot OPT.$$

The "rounding" algorithm is as follows. For every pair (s_i, t_i), let $lca_T(s_i, t_i) = r_i$. Denote by T_{s_i} the child subtree of r_i that contains the vertex s_i, and by T_{t_i} the child subtree of r_i that contains the vertex t_i. The leaves of these subtrees define two disjoint sebsets of verticies V: let $V_{s_i} \subseteq V$, and $V_{t_i} \subseteq V$ denote the subsets that contain s_i and t_i respectively. Add to E' the set of edges that are in the cuts $(V_{s_i}, V \setminus V_{s_i})$, and $(V_{t_i}, V \setminus V_{t_i})$. Note that the output set E' is a feasible solution to the Multi-Cut problem, as by the construction it clearly separates between every pair (s_i, t_i). In addition, it holds that $\sum_{(u,v) \in p_i^j} \hat{\tau}_{u,v} \geq 1$, for each path connecting (s_i, t_i). Since f is not contractive, $\Delta(r_i) = d_T(s_i, t_i) \geq 1$. Therefore, for each $e \in E'$, it holds that $d_T(e) \geq 1$, implying

$$\sum_{(u,v) \in E'} c(u,v) \leq \sum_{(u,v) \in \binom{V}{2}} c(u,v) d_T(u,v) \leq O(\min\{\log n, \log(\Phi(c))\}) OPT.$$

13.1.3 Min-Sum k-Clustering in Metric Spaces

Clustering is a fundamental problem that arises in many applications in different fields. It has various versions, depending on the application. The basic version has the following formulation: given a weighted graph $G = (V, E, w)$, and an integer $k \geq 1$, partition the vertices of the graph into k disjoint clusters $C_1, C_2, \ldots, C_k$ as to minimize

$$\sum_{i=1}^{k} \sum_{e \in E, e \in C_i} w(e).$$

For general graphs this problem is known to be NP-complete, moreover, this problem is even hard to approximate within a factor of $n^{2-\epsilon}$, for any $k \geq 3$. However, for a clustering of a metric space points the picture is different.

We present an $O\left(\frac{1}{\epsilon} \log^{1+\epsilon} n\right)$ approximation algorithm with running time $n^{O(1/\epsilon)}$, for clustering problem in general metric spaces. This algorithm was developed in [2], and improved to have an approximation factor $O(\log n)$ in [3].

The result is obtained by devising a constant factor approximation algorithm for the min-sum clustering problem in $\Omega(\log^{\epsilon} n)$-HST, with running time $n^{O(1/\epsilon)}$. Applying the probabilistic embedding into such HST's we obtain the stated approximation factor $O(\alpha\mu \log_{\mu} n)$, where α is a distortion of the embedding, and μ is a parameter of μ-HST.

Let us present the *Balanced k-median* problem: given a weighted graph G partition its vertices into k clusters with centers $c_1, c_2, \ldots c_k$ as to minimize

$$\sum_{i} |C_i| \cdot \sum_{v \in C_i} d(v, c_i).$$

The solution to this problem is within a factor of 2 of the original k-clustering problem.

Thus, we focus on designing a constant factor approximation algorithm for balanced k-median problem in μ-HST's. This is done in two conceptual steps.

First, the exact dynamic programing algorithm is designed. It's running time is $n^{O(\log n)}$. Intuitively speaking this algorithm computes solutions for the possible partitions of points into sets, which connect to particular centers from the same distance. Second, in order to reduce the running time, we restrict the set of possible solutions. This is done by approximating the sizes of the sets we choose to within a small factor $1 + 1/polylog(n)$. The running time of this algorithm is $(\log n)^{O(L)}$, where L is the number of levels in the μ-HST. Note that without loss of generality we can assume that the μ-HST has $O(\log_\mu n)$ levels. In order to get a polynomial running time, we use a μ-HST with $\mu = \Omega(\log^\epsilon n)$ that results in a running time of $n^{O(1/\epsilon)}$. Let us start.

A simple (exact) dynamic program algorithm.

The algorithm builds up a table, from the bottom up on the tree, starting with leaves. For each node v located on the level l of the tree (level of a node it is its depth, while root is of depth 1) algorithm computes the value $D^*(v, k, s, n_1, n_2, \ldots, n_l)$. This is defined to be the value of the solution for the subtree T_v rooted at v, such that k leaves are chosen as centers among the leaves in T_v, at most s of the leaves in T_v draw their service from outside the tree (i.e., are not accounted for in the cost), and for $i = 1, \ldots l$, n_i vertices connect to the centers within the subtree T_v from level i. This parameter fully describes the cost of the appropriate solution. Also, going over all possible values for $s, n_1, \ldots, n_l$, for some node v, is equivalent to going over all possible k-clustering solutions on the subspace T_v. For each node it takes $n^{O(\log n)}$ to compute its table value, since we have to go over all possible values for n_i (which can be anything from 0 to n). Thus, the total running time is $n \cdot n^{(O(\log n))} = n^{O(\log n)}$.

An improved (approximation) dynamic program algorithm.

The idea is to reduce the number of values computed by the dynamic programming. Let $\alpha = 1 + \frac{1}{c \log^2 n}$. We define the set of *valid integers* to be the set $\{\lceil \alpha^i \rceil : \ 0 \leq i \leq B\}$, where $B = O(\log^3 n)$ is such that $\alpha^B \geq n$. In the new dynamic program we compute only values of the form $P(v, k, s, n_1, \ldots, n_l)$, where each n_i is a valid integer. It is not hard to see that such solution does not lose much in its cost with respect to the optimal solution (it actually loses just a constant factor). We omit the details of the proof.

13.1.4 The Metric Labeling Problem

The problem was suggested in [5] as a general framework for various classification problems. The authors formulated the problem itself and presented polynomial time approximation algorithm based on the probabilistic embedding into trees.

The Metric Labeling Problem Definition.
Given a set P of n objects (for example, factories), and a set L of k labels. The graph $G = (P, E)$

is defined over the set of objects, where the weight function $w_e(e)$, for all $e \in E$ defines the strength of relation between two objects (for example, the cost of communication per unit of distance between two factories) - strongly related objects will seek similar labels. The label set L is equipped with a distance function, defining a metric space (L, d) (for example, L is a set of locations with distances between them). In addition, for all pair (p, l), $p \in P$, $l \in L$ we have the function $c(p, l) \geq 0$ that estimates the cost of assigning the label l to the object p (for example, the price of building a factory p at a location l).

A labeling of P over L is simply a function (not necessarily one-to-one) $f : P \to L$, with a quality measure based on contribution of two following terms:

$$Q(f) = \sum_{p_i \in P} c(p_i, f(p_i)) + \sum_{(p_i, p_j) \in E} w_{i,j} \cdot d(f(p_i), f(p_j)).$$

The labeling problem asks for a discreet labeling of minimum total cost. (In our example, we want to design a plan for setting up factories such that the cost of the construction and the overall communication cost are minimized.)

The problem is known to be NP-hard, thus we seek an approximation algorithm. In [5] the authors show the following results:

1. **The Uniform Labeling Problem**. If d is a uniform metric over L (equilateral metric space), then there is a randomized polytime algorithm that solves the metric labeling problem within a factor 2 (the costs $c(p, l)$, and the weights on the edges are arbitrary).
2. **The General Labeling Problem**. For a general metric space over L (recall that $|L| = k$) the authors suggest a randomized, polynomial time algorithm, with approximation factor $O(\log k)$. This is achieved by probabilistic embedding of L into a distribution of μ-HST's, with distortion $O(\log k)$, and the by solving the problem on a tree metric with $O(1)$ approximation.

13.1.4.1 2-Approximation for The Uniform Labeling Problem

The idea is to write an ILP that encodes the problem, and then to design a smart (randomized) rounding scheme.

For each object $p \in P$ and each label $a \in L$, define the variable $x_{pa} = 1$ if $f(p) = a$, and 0 otherwise. We require that $\sum_{a \in L} x_{pa} = 1$. The label cost of an object p is then expressed as $\sum_{a \in L} c(p, a) \cdot x_{pa}$. Consider two objects p and q. The distance between the assigned labels can be expressed as $d(f(p), f(q)) = \frac{1}{2} \sum_{a \in L} |x_{pa} - x_{qa}|$. Hence, the separation cost of an edge $e = (p, q)$ is expressed as $\frac{1}{2} w_e \sum_{a \in L} |x_{pa} - x_{qa}|$. For all $e = (p, q) \in E$, we introduce variables z_{ea} to express the absolute value $|x_{pa} - x_{qa}|$, and the variables z_e to express the distance between labels assigned to it's end points. Therefore, we can write the following IP:

$$min \quad \sum_{p \in P, a \in L} c(p, a) x_{pa} + \sum_{e \in E} w_e z_e$$

$$
\begin{aligned}
\textit{subject to} \quad & \sum_{a \in L} x_{pa} = 1, && p \in P \\
& z_e = \frac{1}{2} \sum_{a \in L} z_{ea}, && e \in E \\
& z_{ea} \geq x_{pa} - x_{qa}, && e = (p,q), a \in L \\
& z_{ea} \geq x_{qa} - x_{pa}, && e = (p,q), a \in L \\
& x_{pa} \in \{0,1\}
\end{aligned}
$$

We solve the relaxation of this integer program to to get for each p a fractional solution $\{x_{pa}\}$ - the set of rational values such that $\sum_{a \in L} x_{pa} = 1$. Next we describe a randomized rounding technique. We assign labels to vertices in phases; initially each vertex p has no label assignment. In a single phase, we pick a label $a \in L$ uniformly at random, and a real number α uniformly in $[0,1]$. For each object p that currently has no label assignment, we give it the label a if $\alpha \leq x_{pa}$. We iterate these phases until each vertex has a label assignment.

The following two lemmas state simple properties of this randomized process:

Lemma 13.4. *Consider a particular phase, and an object p that has not yet been assigned a label at the start of this phase. The probability that p is assigned label a in this phase is precisely x_{pa}/k, and the probability that object p is assigned any label in this phase is precisely $1/k$. Further, over all phases, the probability that an object p is assigned label a is precisely x_{pa}.*

The proof is almost immediate.

We say that the two ends of an edge are separated, or "split," by this process if they receive different labels. We say that an edge $e = (p,q)$ is separated by a single phase, if before the phase both p and q are unassigned, and exactly one of p and q is assigned in this phase. Note that an edge that is separated is separated by some phase, but the reverse is not true, as later the other end of the edge may be assigned the same label.

Lemma 13.5. *For an edge $e = (p,q)$ the probability that edge e is separated by a given phase, assuming neither end is assigned before the phase, is exactly $2z_e/k = \sum_{a \in L} |x_{pa} - x_{qa}|/k$. The probability that the two objects p and q are separated by the process is at most $2z_e$.*

Proof. If p and q are both assigned a label in this phase (recall that they are unassigned at the beginning of the phase), then they clearly are not separated by the process—since all objects assigned a label in this phase receive the same label. The conditional probability that p and q are separated in this phase, given that at least one of p and q is assigned by this phase, is

$$
\frac{Pr[\text{p, q are separated by this phase}]}{Pr[\text{at least one of p, q is assigned a label by this phase}]}.
$$

The numerator is equal to $2z_e/k$. The denominator is at least the probability that object p is assigned, which is $1/k$ by Lemma 13.4, so the conditional probability is at most $2z_e$. This is true in any phase that starts with both p and q unassigned, and hence the probability that edge $e = (p,q)$ is split by the process is at most $2z_e$. □

Now we are ready to show that the above rounding strategy results in a 2-approximation algorithm.

Theorem 13.6. *Let x be a solution of the linear program with assignment cost c_{LP} and separation cost w_{LP}. The randomized rounding procedure described above finds a labeling whose expected assignment cost is c_{LP}, and whose expected separation cost is at most $2w_{LP}$.*

Proof. By Lemma 13.4, the probability that object p is assigned label a is x_{pa}, and hence the expected contribution of p to the objective function is $\sum_{a \in L} c(p,a)x_{pa}$, precisely its contribution in the linear relaxation. Hence, the total expected assignment cost over all objects is c_{LP}. What is the contribution of edge $e = (p,q)$ to the expected separation cost? This is precisely w_e times the probability that its ends are split, since we are using the uniform metric. For an edge $e = (p,q)$ the probability that the two objects p and q are separated by the process is at most $2z_e$ by Lemma 13.5. Hence, the expected contribution of edge e is at most $2w_e z_e$, which is twice its contribution in the linear relaxation. The total expected assignment cost over all objects is at most $2w_{LP}$. □

13.1.4.2 $O(1)$-Approximation Algorithm for k-HST Metric

Here again we build an ILP, for which the rounding scheme repeatedly uses the rounding scheme of the uniform case. In total the algorithm achieves constant factor approximation.

References

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. STOC '06, pages 271–286, 2006.

[2] Yair Bartal, Moses Charikar, and Danny Raz. Approximating min-sum k-clustering in metric spaces. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 11–20. ACM, 2001.

[3] Babak Behsaz, Zachary Friggstad, Mohammad R. Salavatipour, and Rohit Sivakumar. Approximation algorithms for min-sum k-clustering and balanced k-median. In *Automata, Languages, and Programming*, volume 9134 of *Lecture Notes in Computer Science*, pages 116–128. 2015.

[4] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 698–707, New York, NY, USA, 1993. ACM.

[5] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 14–23, 1999.