Lecture 7 (10.04.2013) Scribe: Grzegorz Prusak

Lecturer: Marcin Mucha

Metric embeddings into trees

1 Introduction

Sometimes general metric problems are easier to solve for specific metrics (for example for euclidean metrics or tree metrics). In this lecture we focus on tree metrics. Specifically we will construct for any metric space (V, d) a mapping $V \to T$, where T is a weighted tree (in fact a distribution on such mappings), such that distances are approximately maintained, i.e. $d_{uv} \leq T_{uv} \leq cd_{uv}$ where T_{uv} is z tree distance between the nodes to which u and v have been mapped. c is called distortion of the mapping.

Theorem 1. *Every embedding of C_n to a tree has distortion $\Omega(n)$.*

Proof. Consider any embedding of C_n into a tree T which does not decrease any distances. Extend this embedding for the whole circle (viewed as a metric space) by linear interpolatoin between each pair of consecutive cycle vertices.

Lemma 2. *There exist 2 points of the circle at distance at least $\frac{2\pi}{3}$, which are mapped to the same point in the tree.*

Proof. Consider how 3 circle points at distance $\frac{2\pi}{3}$ from each other are mapped into the tree. In

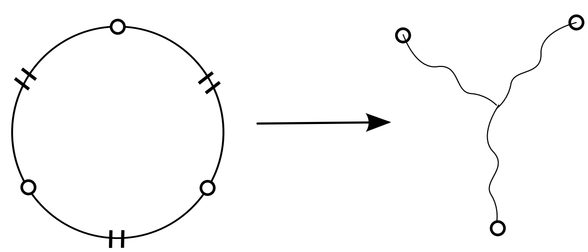

Figure 1: Illustration for the proof of Lemma 2

the tree, paths between them cross in a single point, which is therefore the image of some point in each of 3 segments of the circle (see Figure 1). It is easy to see that 2 of these 3 points have to be at distance at least $\frac{2\pi}{3}$ from each other. $\square$

Now, take 2 circle points from the lemma. Let x be their common image, and also look at the nearest cycle vertices to their left and to their right, denote them by a, b and c, d respectively (see Figure 2). Also, denote the images of all these points by the same symbols.

For n big enough:

$$d_{ac} + d_{bd} \geq \frac{4\pi}{3} - \mathcal{O}(\frac{1}{n})$$

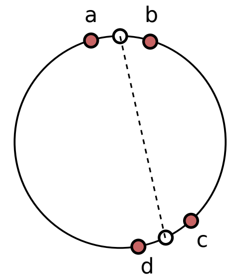

Figure 2: Illustration for the definition of the points a, b, c, d.

and

$$T_{ab} + T_{cd} = T_{ax} + T_{xb} + T_{cx} + T_{xd} \geq T_{ac} + T_{bd} \geq \frac{4\pi}{3} - \mathcal{O}(\frac{1}{n}).$$

which gives the claim since $d_{ab}, d_{cd} = O(1/n)$, so one of these distances is increased $\Omega(n)$ times. $\square$

Although distortion might need to be even linear in the size of the base metric space (as shown in the theorem above), we will present a randomized algorithm, which returns a tree and a mapping, such that expected distortion of every single edge will be $\mathcal{O}(log\ n)$. Namely for all u, v:

- $d_{uv} \geq T_{uv}$ (deterministically) and,
- $\mathbb{E}(T_{uv}) \leq \mathcal{O}(log\ n)d_{uv}$

For example, for a cycle it is enough to remove a random edge and then for u, v at distance d we get:

$$\mathbb{E}(T_{uv}) = \frac{n-d}{n}d + \frac{d}{n}(n-d) = \frac{2d(n-d)}{n} \leq 2d.$$

2 Algorithm of Fakcharoenphol, Rao and Talwar (FRT)

Let us assume that $\forall_{u,v} d_{uv} \geq 1$ (this can be guaranteed by scaling appropriately) and let

$$\Delta = \min\{k : 2^k > 2\max(d_{uv})\}$$

We will use the tree illustrated in Figure 3 in our construction Vertices are mapped to corresponding singletons in the leafs. Each point of the tree represents a cluster of points. Sum of clusters on each tree level gives V. Length of the edges between level i and $i-1$ is 2^i. Every cluster in level i is contained in a ball of radius r (where $2^{i-1} \leq r < 2^i$), centered in some vertex, but not necessary a vertex of the cluster.

Lemma 3. *If lowest common ancestor of x, y is in the level i, then $T_{xy} = 2^{i+2} - 4$*

Proof. Let $2^i \leq d_{xy} < 2^{i+1}$. The the minimum radius of a ball containing both x and y is $\geq 2^{i-1}$. Their LCA is thus in the level $\geq i$, and so $T_{xy} \geq 2^{i+2} - 4$. Also, if $i = 0$, then we still have $T_{xy} \geq 4$. $\square$

This implies

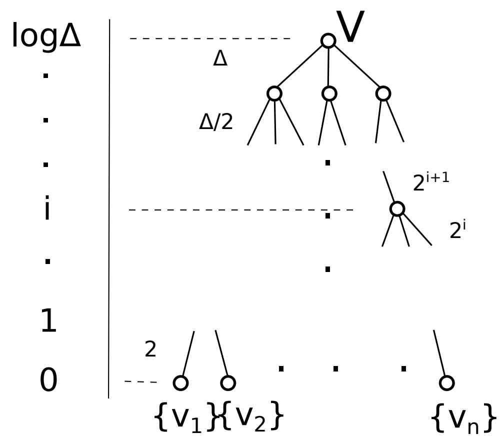

Figure 3: The tree construction used in the FRT algorithm

Lemma 4. $d_{xy} \leq T_{xy}$

Algorithm 1: Fakcharoenphol, Rao and Talwar

FRT(V):
Draw a random permutation π of V;
Draw (uniformly) a random number r_0 from $\left[\frac{1}{2}, 1\right)$;
Start with cluster V on level $\log \Delta$;
while *there is an unsplit cluster S on level $k > 0$* **do**
  Split(S,k);

Split(S,k):
$r := r_0 2^k$;
foreach $v \in V$ *(in π order)* **do**
  Insert ball $B(v, r) \cap S$ as a new son of S;
  $S := S \backslash B(v, r)$;

Theorem 5 (Fakcharoenphol, Rao and Talwar)**.** *The above construction gives a tree embedding with average distortion $\mathcal{O}(\log n)$*

Proof. To analyze the algorithm let us fix a pair of vertices and analyze the expected value of T_{xy}. Let us first introduce the following two classes of events:

- C_{wk} – w separates x from y on level k, i.e. level k ball centered in w contains exactly one of x, y.
- S_{wk} – w decides about x, y on level k, i.e. w is the first vertex at level k which separates x and y, and whose ball is added to the tree.

Also, let $b_w = 1/i$ if w is ranked i-th in the ordering of increasing values of $\min(d_{wx}, d_{wy})$. Notice that

Lemma 6.

$$P(S_{wk}|C_{wk}) \leq b_w.$$

We also have

Lemma 7.

$$\sum_k P(C_{wk})2^{k+3} \leq 16d_{xy}.$$

Proof. Let $R_1 = d_{wx}$ and $R_2 = d_{wy}$. We then have $|R_2 - R_1| \leq d_{xy}$. W.l.o.g. assume that $R_2 > R_1$. Then, since we have

$$P(C_{wk}) = \frac{\left|[R_1, R_2] \cap [2^{i-1}, 2^i]\right|}{|[2^{i-1}, 2^i]|}$$

we get the claim

$$\sum_k P(C_{wk})2^{k+3} \leq \sum_k \frac{\left|[R_1, R_2] \cap [2^{i-1}, 2^i]\right|}{|[2^{i-1}, 2^i]|} 2^{i+3} = \sum_k 16 \left|[R_1, R_2] \cap [2^{i-1}, 2^i]\right| = 16 \left|[R_1, R_2]\right| \leq 16d_{xy}.$$

□

Now we can come back to the proof of the theorem. We have $T_{xy} = 2^{k+3} - 4$, where k is the highest level such that x, y. Therefore

$$T_{xy} = \max_{k=0\ldots\log\Delta-1} \mathbb{1}_{\{\exists w: C_{wk} \wedge S_{wk}\}} 2^{k+3} - 4 \leq \sum_{k=0\ldots\log\Delta-1} \sum_{w\in V} \mathbb{1}_{\{C_{wk} \wedge S_{wk}\}} 2^{k+3}.$$

Going to expectations we get

$$\mathbb{E}T_{xy} \leq \sum_w \sum_k P(C_{wk} \wedge S_{wk})2^{k+3} = \sum_w \sum_k P(C_{wk})P(S_{wk}|C_{wk})2^{k+3}.$$

We now use both the lemmas we proved to get

$$\mathbb{E}T_{xy} \leq \sum_w \sum_k P(C_{wk})b_w 2^{k+3} = \sum_w b_w \sum_k P(C_{wk})2^{k+3} \leq \sum_w b_w 16d_{xy} = \mathcal{O}(\log n)d_{xy}.$$

□

Lemma 8. *We can demand from the tree to have no unmapped nodes (i.e. $V' = V$) and still maintain $\mathcal{O}(\log n)$ expected distortion.*

Proof. To achieve that we use the following algorithm:

Algorithm 2: Getting rid of extra nodes

while *there exist vertices beyond V* **do**
  Take $v \in V : w$(parent of v) $\notin V$;
  Merge v and w;
Multiply all edges by 4;

Let T' be the resulting tree metric. Note that we have

$$T'_{xy} \leq 4T_{xy}$$

for all x, y since there is only one operation increasing the distances (multiplication by 4) and it is performed only once. We also have

$$T'_{xy} \geq T_{xy}.$$

This is because at most one downward edge of the $lca(x, y)$ will be contracted. The other one's length stands for $\frac{1}{4}$ of T_{xy}, therefore after multiplication it will stand for the whole original path. $\square$

Finally let us mention that the trees constructed in this section have special structure which is sometimes used in algorithms. The following definition captures the essense of this structure.

Definition 9. A k-Hierarchically Separated Tree (k-HST) is a rooted tree in which the father edge of any node is at least k times longer than all its children edges.

Note that the tree constructed in this section is a 2-HST.

3 Group Steiner Tree

GROUP STEINER TREE (GST) **Input:** graph $G = (V, E)$; cost function $c : E \to \mathbb{R}^+; \forall_{i=1..k} S_i \subseteq V$ **Question:** What is the min cost tree T in V, such that $\forall_{i=1..k} T \cap S_i \neq \emptyset$?

We now present an $\mathcal{O}(\log n \log \Delta \log k)$-approximation algorithm for GST, due to Garg, Konjevod and Ravi. This algorithm first constructs for a given metric space (V, d) an FRT tree (V, T). We then add, for every non-leaf node, an extra zero-cost edge with a leaf at it's end. In this way, we can assume that all vertices of V are mapped to leaves. We now consider the following LP relaxation of GST:

$$\begin{aligned} \text{minimize } \sum_{e \in E} w_e x_e & \\ \forall_{i,e}\ x_e \geq f_e^i & \\ \forall_{i, v \notin S_i} \sum_{e \in \delta(v)} f_e^i = 0 & \\ \forall_{i,v}\ f_v^i = \sum f_{vv'}^i & \\ x_e, f_e^i, f_v^i \geq 0 & \end{aligned} \tag{1}$$

Here:

- x_e models buying the edge e.
- f_e^i for fixed i describe the flow used to connect group i to the root.
- $f_v^i = f_{p(v)v}^i$ is just an alias used to simplify notation, here $p(v)$ is a parent of v.

We are now ready to describe the algorithm itself.

Algorithm 3: $\mathcal{O}(\log n \log \Delta \log k)$-approximation for GST

Construct an FRT tree (V, T) and augment it as described above;
Solve the LP relaxation for (V, T);

Mark root as active;
while *there exists an active node* v **do**
    foreach *son* v' *of* v **do**
        Buy vv' (i.e. add to the solution) with probability $\frac{x_{vv'}}{x_{p(v)v}}$;
        If vv' is bought, v' becomes active;
    v becomes inactive;

The following is easy to prove by induction.

Lemma 10. *Probability that edge e is purchased is equal to x_e. Therefore the expected cost of the solution is OPT_{LP}*

Let $h = \log \Delta$. Then we also have

Lemma 11. *Fix group S_g and vertex $v \in V$ with depth $i > 0$. Let P_v be the probability that the algorithm does not reach the group S_g from v, given that it does reach v. Then*

$$P_v \leq 1 - \frac{f_v^g}{(h-i+1)X_{p(v)v}}.$$

Proof. We use backwards induction on i. Consider first the case where $i = h$, i.e. v is a leaf. If the algorithm reaches v, then either $f_v^g = 0$ and we are fine, or $f_v^g > 0$, in which case $v \in S_g$ and so the algorithm has already reached S_g.

Consider now the case where $i < h$, i.e. v is not a leaf, and assume the claim holds for larger i. We then have

$$P_v = \prod_{v':p(v')=v} \left(1 - \frac{x_{vv'}}{x_{p(v)v}}(1 - P_{v'})\right) \leq \prod_{v'} \left(1 - \frac{x_{vv'}}{x_{p(v)v}} \frac{f_{v'}^g}{(h-i)x_{vv'}}\right).$$

We now use the fact that $1 - x \leq e^{-x}$ to bound this by

$$\prod_{v'} \exp\left(-\frac{f_{v'}^g}{(h-i)x_{p(v)v}}\right) = \exp\left(-\frac{\sum f_{v'}^g}{(h-i)x_{p(v)v}}\right) = \exp\left(-\frac{f_v^g}{(h-i)x_{p(v)v}}\right).$$

Now, by inverting the inequality $e^x \geq 1 + x$, and applying it to $x = \frac{a}{b}$ we get

$$e^{-\frac{a}{b}} \leq \frac{1}{1 + \frac{a}{b}} = \frac{b}{a+b} = 1 - \frac{a}{a+b}.$$

By applying this inequality to our bound we get

$$P_v \leq 1 - \frac{f_v^g}{(h-i)x_{p(v)v} + f_v^g} \leq 1 - \frac{f_v^g}{(h-i+1)x_{p(v)v}},$$

where the last step follows from $f_v^g \leq x_{p(v)v}$. ∎

Note that the above argument also works when v is the root, even though technically $p(v)$ does not exist (the same argument works, or one can even add a dummy parent vertex). Since the algorithm always reaches the root, we obtain

Corollary 12. *For every group S_g the algorithm reaches S_g with probability at least $\frac{1}{h+1}$.*

If we repeat the algorithm $(h+1)\log k$ times and take the union of all these solutions, the probability that any given group is not connected is at most

$$\left(1 - \frac{1}{h+1}\right)^{(h+1)\log k} \sim \frac{1}{k}.$$

If we do get some groups that are not connected, we simply connect them in the cheapest possible way. Each such connection clearly costs at most OPT. The total cost is therefore bounded by

$$(h+1)\log k \; OPT_{LP} + \sum_{g=1..k} \frac{1}{k} OPT = \mathcal{O}(h \log k) OPT = \mathcal{O}(\log \Delta \log k) OPT.$$

To get the final approximation ratio, we need to multiply this by $\mathcal{O}(\log n)$ since we started with the FRT embeddding.