CMPUT 675: Approximation Algorithms	Fall 2014

Lecture 23 (Oct 29): Tree Metrics — Part 1

Lecturer: Zachary Friggstad *Scribe: Leah Hackman*

23.1 Tree Metrics

Let (V, d) be a metric (with V locations, and d is our metric function $d : V \times V \to \mathbb{R}$ which satisfies the standard metric properties).

Definition 1 *A* tree metric *for (V, d) as a tree T with vertices $V' \subseteq V$ with edge lengths such that for each $u, v \in E$ we have $d(u, v) \leq T(u, v)$. Here, $T(u, v)$ is the distance between between u and v in the tree.*

Trees have a lot of nice properties which make some problems easier to solve or approximate, and so it can be helpful to reframe a problem to use a tree metric. That being said, not all tree metrics are good. We have defined tree metrics so that they provide an upper bound of our original distance metric, but have made no stipulations on how tight this bound is.

We say that the tree metric T for (V, d) has *stretch* α if:

$$T(u, v) \leq \alpha \cdot d(u, v) \quad \text{for each } u, v \in V$$

Knowing the stretch of a tree metric allows us to put an approximation bound on our solution. For an appropriate problem (such as one where the objective is to minimize the sum of edge lengths used), if OPT_T is the optimal solution on a tree metric T for the original metric (V, d) and if T has stretch α, then $OPT_T \leq \alpha \cdot OPT$. If we have a β-approximation of OPT_T, then this leads to an $\alpha \cdot \beta$-appxorimation of OPT.

Unfortunately, some metrics cannot be well approximated by a tree metric at all. For example, one can show that the cycle C_n with unit weight edges requires stretch $\alpha \geq \frac{n-1}{8}$. However, we can get around this barrier with a randomized approach: a probability distribution over tree metrics.

Theorem 1 *For any metric (V, d), we can randomly construct a tree metric T in polynomial time such that:*

$$\mathrm{E}[T(u, v)] \leq \mathcal{O}(\log(|V|)) \cdot d(u, v) \quad \textit{for each } u, v \in V$$

23.1.1 The Construction

We will describe some general properties of all of the trees the randomized construction produces. First, for simplicity later, let us assume that $\min_{\substack{u,v \in V \\ u \neq v}} d(u, v) = 1$. We can scale all our distances to make this true if it isn't.

In the tree we construct, we can associate every node $\ell \in T$ as being associated to a subset of V: $S_\ell \subseteq V$. The leaves of T will correspond to the vertices in V and we sometimes index the leaves of T with $v \in V$.

For any subset $S \subseteq V$, let $\operatorname{diam}(S) := \max_{u,v \in S} d(u,v)$. Let Δ be the smallest power of 2 that is at least $\operatorname{diam}(V)$. That is, $\Delta = 2^h$ where $h = \lceil \log_2(\operatorname{diam}(V)) \rceil$.

The tree we randomly construct will also satisfy the following properties:

- Any edge from i to level $i+1$ has cost 2^{i+1}

- For a tree vertex u with children $w_1, \ldots, w_a$

$$S_u = \bigcup_{1 \le i \le a} S_{w_i} \text{ and } S_{w_i} \cap S_{w_j} = \emptyset, i \neq j$$

  That is, the sets S_{w_i} partition S_u.

- For a tree vertex u at level i, $\operatorname{diam}(S_u) \le 2^i$

- The root node r of the tree has $S_r = V$ and is at level h.

See Figure 23.1 for an example tree and a visual explanation of the term “level”.

Before we show how to build such a tree, let us first prove that such a tree forms a valid tree metric.

Lemma 1 *For any such tree T (which satisfies the above properties), and any two $u, v \in V$ then $d(u,v) \le T(u,v)$. Also, if the least common ancestor of u and v in T is on level i, then $T(u,v) \le 2^{i+2}$*

Proof. The second bound is simple:

$$T(u,v) = \sum_{j=1}^{i} 2 \cdot 2^j = 2 \cdot (2^{i+1} - 2) \le 2^{i+2}.$$

Note also that $T(u,v) = 2 \cdot (2^{i+1} - 2) \ge 2^{i+1}$.

For the first bound, let $k = \lfloor \log_2(d(u,v)) \rfloor$. We know that since u, v lie in some set S_w where w is at level i, then $\operatorname{diam}(S_w) \le 2^i$ means $d(u,v) \le 2^i$. So, $i \ge k$, otherwise $d(u,v) \le 2^{k-1} < 2^{\log_2(d(u,v))} = d(u,v)$ which is impossible.

From here, we see $d(u,v) \le 2^{k+1} \le 2^{i+1} \le T(u,v)$. ■

To sample this tree, we use Algorithm 1. In this algorithm, $\mathcal{C}(i)$ defines the partition of the nodes of V in level i. The sets in the partition $\mathcal{C}(i)$ are nodes of the tree T.

For $v \in V$ and $r \ge 0$, we let $B(v,r)$ denote the ball of radius r around v, namely $\{u \in V : d(u,v) \le r\}$.

The algorithm runs in time that is polynomial in $\log \Delta$ and $|V|$, which is polynomial in description length of the metric.

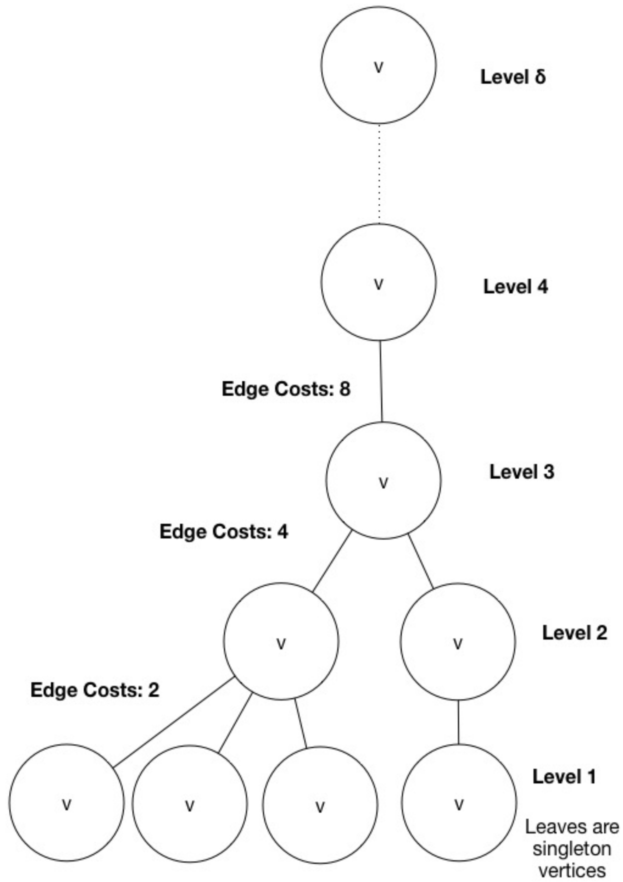

Figure 23.1: A labelled sample tree to explain "level"

Algorithm 1 An algorithmic for randomly building a tree metric T for (V, d) which has the property $\mathrm{E}[T(u, v)] \leq \mathcal{O}(\log(|V|)) \cdot d(u, v) \quad$ for each $u, v \in V$

Sample r_0 uniformly from $[1/2, 1)$. Set $r_i = 2^i \cdot r_0$ for $1 \leq i \leq \Delta$. (Recall that $\log(\Delta) = \lceil \log(\text{diam}(V)) \rceil$)
Let $\pi : V \to V$ be a random permutation of V
$\mathcal{C}(\log(\Delta)) = \{V\}$
for i from $\log(\Delta)$ down to 1 **do**
  $\mathcal{C}(i-1) \leftarrow \emptyset$
  for each $\mathcal{S} \in \mathcal{C}(i)$ **do**
    $S' \leftarrow S$
    for each $v \in V$ in order of π **do**
      $S^* \leftarrow B(v, r_{i-1}) \cap S'$
      if $S^* \neq \emptyset$ **then**
        $\mathcal{C}(i-1) \leftarrow \mathcal{C}(i-1) \cup \{S^*\}$
        $S' \leftarrow S' - S^*$
        Add an edge with distance 2^i connecting the vertices corresponding to S^* and S