CPSC 536N: Randomized Algorithms	2011-12 Term 2
Lecture 24	
Prof. Nick Harvey	*University of British Columbia*

1 Probabilistic Approximation of Metrics

For many optimization problems, the input data involves some notion of distance, which we formalize as a metric. But unfortunately many optimization problems can be quite difficult to solve in an arbitrary metric. In this lecture we present a very approach to dealing with such problems, which is a method to approximate any metric by much simpler metrics. The simpler metrics we will use are trees, i.e., the shortest path metric on a graph that is a tree. Many optimization problems are easy to solve on trees, so in one fell swoop we get algorithms to approximate a huge number of optimization problems.

Roughly speaking, our main result is: *any metric on n points can be represented by a distribution on trees, while preserving distances up to a $O(\log n)$ factor*. Consequently: *for many optimization problems involving distances in a metric, if you are content with an $O(\log n)$-approximate solution, you can assume that your metric is a tree.*

In order to state our results more formally, we will need to deal with a important issue. To illustrate the issue, and how to deal with it, we first present an example.

1.1 Example: Approximating a cycle

Let $G = (V, E)$ be a cycle on n nodes. The (spanning) subtrees of G are simply the paths obtained by deleting a single edge. So let uv be an edge and let $T = G \setminus uv$ be the corresponding tree. Is the shortest path metric of T a good approximation of the shortest path of G? The answer is no: the distance between u and v in G is only 1, whereas the distance between u and v in T is $n - 1$. So, no matter which subtree T of G we pick, there will be some pair of nodes whose distance is poorly approximated.

Is there some way around this problem? Perhaps we don't need T to be a *sub*tree of G. We could consider a tree $T = (U, F)$ (possibly with lengths on the edges) where $U \supseteq V$ and F is completely unrelated to E. Can such a tree do a better job of approximating distances in G? It turns out that the answer is still no: there will always be a pair of nodes whose distance is only preserved up to a factor $\Theta(n)$.

But here is a small observation: any subtree of T approximately preserves the *average* distances. One can easily check that the total distance between all pairs of nodes is $\Theta(n^3)$, for both G and for any subtree of G. Thus, subtrees approximate the distances in G "on average".

So for the n-cycle, a subtree cannot approximate all distances, but it can approximate the average distance. This motivates us to apply a trick that is both simple and counterintuitive. It turns out that we *can* approximate all distances if we allow ourself to pick the subtree randomly. (The trick is Von Neumann's minimax theorem, and it implies that approximating the average distance is equivalent to finding a *distribution* on trees for which *every* distance is approximated in expectation.)

To illustrate this, choose any pair of vertices $u, v \in V$. Let $d = d_G(u, v)$ be the distance between u and v in G. Pick a subtree T by deleting an edge e at random and let $d_T(u, v)$ be the u-v distance in T.

Obviously $d_T(u,v) \geq d_G(u,v)$ since we constructed T by removing e from G. We now give an upper bound on $\mathrm{E}[d_T(u,v)]$. If e is on the shortest u-v path then $d_T(u,v) = n - d$; the probability of that happening is d/n. Otherwise, $d_T(u,v) = d$. Thus,

$$\mathrm{E}[d_T(u,v)] \;=\; (d/n)\cdot(n-d) + (1-d/n)\cdot d \;\leq\; 2d.$$

So, *every* edge of G is approximated to within a factor of 2, in expectation.

1.2 Main Theorem

We now show that, for *every* metric (X,d) with $|X| = n$, there is an algorithm that generates a random tree for which *all* distances are approximated to within a factor of $O(\log n)$, in expectation.

Theorem 1 *Let (X,d) be a finite metric with $|X| = n$. There is a randomized algorithm that generates a set of vertices Y, a map $f : X \to Y$, a tree $T = (Y,F)$, and weights $w : F \to \mathbb{R}_{>0}$ such that*

$$d_X(x,y) \;\leq\; d_T\big(f(x), f(y)\big) \tag{1}$$

and

$$\mathrm{E}[d_T\big(f(x), f(y)\big)] \;\leq\; O(\log n)\cdot d_X(x,y) \quad \forall x,y \in X. \tag{2}$$

The main tool in the proof is the random partitioning algorithm that we developed in the last two lectures. For notational simplicity, let us scale our distances and pick a value m such that such that $1 < d(x,y) \leq 2^m$ for all distinct $x,y \in X$. Note that m does not appear in the statement of the theorem, so we do not care how big it is.

The main idea is to generate a 2^i-bounded random partition $\mathcal{P}_i$ of X for every $i = 0,\ldots,m$ then assemble those partitions into the desired tree. Assembling them is not too difficult, but there is one annoyance: the parts of $\mathcal{P}_i$ have absolutely no relation to the parts of $\mathcal{P}_{i'}$ for any $i \neq i'$. If the parts of $\mathcal{P}_i$ were nicely nested inside the parts of $\mathcal{P}_{i+1}$ then this would induce a natural hierarchy on the parts, and therefore give us a nice tree structure.

The solution to this annoyance is to forcibly construct a nice partition $\mathcal{Q}_i$, for $i = 0,\ldots,m$, that is nested inside all of $\mathcal{P}_i, \mathcal{P}_{i+1}, \ldots, \mathcal{P}_m$. In lattice theory terminology, we define the partition

$$\mathcal{Q}_i \;=\; \mathcal{P}_m \wedge \cdots \wedge \mathcal{P}_{i+1} \wedge \mathcal{P}_i,$$

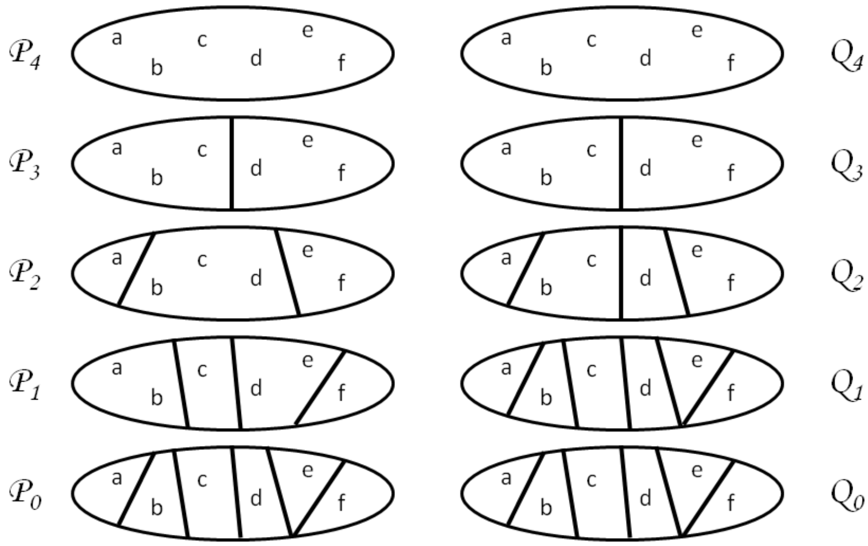
where $\wedge$ is the meet operation in the partition lattice. If you're not familiar with this notation, don't worry; it is easy to explain. Simply define $\mathcal{Q}_m = \mathcal{P}_m$, then let

$$\mathcal{Q}_i \;=\; \{\, A \cap B \;:\; A \in \mathcal{Q}_{i+1},\; B \in \mathcal{P}_i \,\}.$$

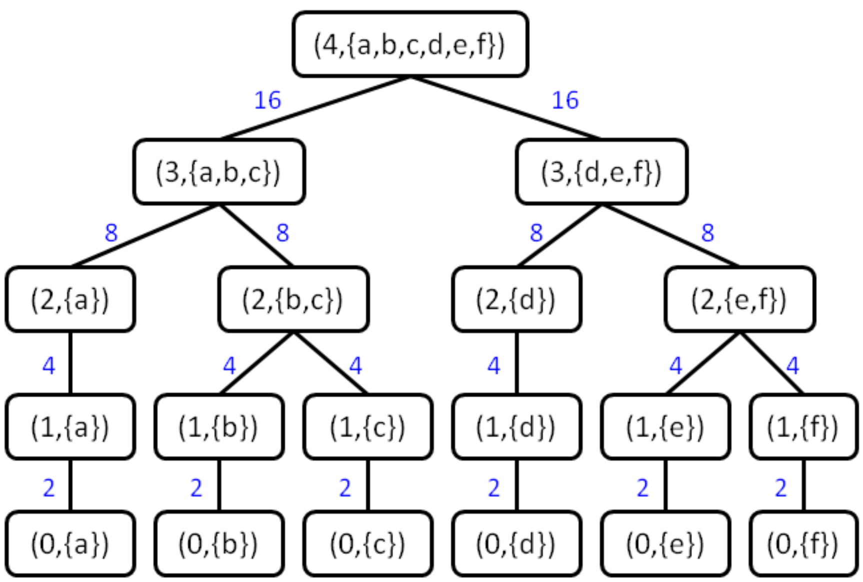
Note that $\mathcal{Q}_i$ is also a partition of X. Furthermore, the parts of $\mathcal{Q}_i$ are nicely nested inside the parts of $\mathcal{Q}_{i+1}$, so we have obtained the desired hierarchical structure.

1.3 Example

Consider the following example which shows some possible partitions $\mathcal{P}_0,\ldots,\mathcal{P}_4$ for the points $X = \{a,b,c,d,e,f\}$, and the corresponding partitions $\mathcal{Q}_0,\ldots,\mathcal{Q}_4$.



The tree corresponding to these partitions is as follows.



1.4 Algorithm

More formally, here is our algorithm for generating the random tree.

- For $i = 0, \ldots, m$, let $\mathcal{P}_i$ be a 2^i-bounded random partition generated by our algorithm from the last lecture.
- $\triangleright$: *The vertices in Y will be pairs of the form (i, S) where $i \in \{0, \ldots, m\}$ and $S \subseteq X$. The vertices and edges of the tree T are generated by the following steps.*
- Define $\mathcal{Q}_m = \mathcal{P}_m$. Add the vertex (m, X) as the root of the tree.
- For $i = m - 1$ downto 0

- Define $\mathcal{Q}_i = \{ A \cap B \;:\; A \in \mathcal{Q}_{i+1},\; B \in \mathcal{P}_i \}$.
- For every such set $A \cap B \in \mathcal{Q}_i$, add the vertex $(i, A \cap B)$ to T as a child of $(i+1, A)$, connected by an edge of length 2^{i+1}.

- Since $1 < d(x,y)$ for all distinct x,y and since $\mathcal{P}_0$ is 1-bounded, the partition $\mathcal{P}_0$ must partition X into singletons. Therefore we may define the map $f : X \to Y$ by $f(x) = (0, \{x\})$.

1.5 Analysis

Claim 2 *Fix any distinct points $x, y \in X$. Let $\ell \in \{0, \ldots, m-1\}$ be the largest index with $\mathcal{P}_\ell(x) \neq \mathcal{P}_\ell(y)$. Then $d_T\big(f(x), f(y)\big) = 2^{\ell+3} - 4$.*

PROOF: The level ℓ is the highest level of the $\mathcal{P}_i$ partitions in which x and y are separated. A simple inductive argument shows that ℓ is also the highest level of the $\mathcal{Q}_i$ partitions in which x and y are separated. So the least common ancestor in T of $f(x)$ and $f(y)$ is at level $\ell + 1$. Let us call the least common ancestor v. Then

$$d_T\big(f(x), v\big) \;=\; d_T\big(f(y), v\big) \;=\; \sum_{i=0}^{\ell} 2^{i+1} \;=\; 2^{\ell+2} - 2.$$

Since $d_T\big(f(x), f(y)\big) = d_T\big(f(x), v\big) + d_T\big(v, f(y)\big)$, the proof is complete. □

Claim 3 (1) *holds.*

PROOF: Let i be such that $2^i < d_X(x,y) \leq 2^{i+1}$. Since $\mathcal{P}_i$ is 2^i-bounded, x and y must lie in different parts of $\mathcal{P}_i$, i.e., $\mathcal{P}(x) \neq \mathcal{P}(y)$. By Claim 2,

$$d_T\big(f(x), f(y)\big) \;\geq\; 2^{i+3} - 4 \;>\; 2^{i+1} \;\geq\; d_X(x,y),$$

as required. □

Claim 4 (2) *holds.*

PROOF: Fix any $x, y \in X$ and let $r = d_X(x,y)$. We have

$$\begin{aligned}
\mathrm{E}[\, d_T\big(f(x), f(y)\big) \,] \;&=\; \sum_{i=0}^{m} \Pr[\, i \text{ is the largest index with } \mathcal{P}_i(x) \neq \mathcal{P}_i(y) \,] \cdot (2^{i+3} - 4) \\
&<\; \sum_{i=0}^{m} \Pr[\mathcal{P}_i(x) \neq \mathcal{P}_i(y)] \cdot 2^{i+3} \\
&\leq\; \sum_{i=0}^{m} \Pr[B(x,r) \not\subseteq \mathcal{P}_i(x)] \cdot 2^{i+3} \\
&\leq\; O(\log n) \cdot r,
\end{aligned}$$

where the last inequality, proven in the following claim, applies Theorem 2 of Lecture 22 and peforms a short calculation. □

Claim 5 *For any $x \in X$ and $r > 1$,*

$$\sum_{i=0}^{m} \Pr[B(x,r) \not\subseteq \mathcal{P}_i(x)] \cdot 2^{i+3} \;\leq\; O(\log n) \cdot r.$$

Proof: Let k be the integer with $2^k < r \leq 2^{k+1}$. Then

$$\begin{aligned} & \sum_{i=0}^{m} \Pr[B(x,r) \not\subseteq \mathcal{P}_i(x)] \cdot 2^{i+3} \\ \leq\; & \sum_{i=0}^{k+3} 2^{i+3} \;+\; \sum_{i=k+4}^{m} \frac{8r}{2^i} \cdot H\Big(|B(x, 2^{i-2} - r)|, |B(x, 2^{i-1} + r)|\Big) \cdot 2^{i+3} \\ \leq\; & 128r \;+\; 64r \cdot \sum_{i=k+4}^{m} H\Big(|B(x, 2^{i-3})|, |B(x, 2^{i})|\Big), \end{aligned}$$

since $r \leq 2^{i-3}$ when $i \geq k+4$. The final sum is upper bounded as follows.

$$\begin{aligned} & \sum_{i=k+4}^{m} H\big(|B(x,2^{i-3})|, |B(x,2^i)|\big) \\ =\; & \sum_{i=k+4}^{m} \Big(H\big(|B(x,2^{i-3})|, |B(x,2^{i-2})|\big) + H\big(|B(x,2^{i-2})|, |B(x,2^{i-1})|\big) + H\big(|B(x,2^{i-1})|, |B(x,2^{i})|\big)\Big) \\ <\; & 3 \sum_{i=k+2}^{m} H\big(|B(x,2^{i-1})|, |B(x,2^i)|\big) \\ =\; & 3 \cdot H\big(|B(x,2^{k+1})|, |B(x,2^m)|\big) \\ =\; & O(\log n). \end{aligned}$$

This proves the claimed inequality. □