CS294-1 On-line Computation & Network Algorithms	**Spring 1997**

Lecture 21: April 17

Lecturer: Yair Bartal *Scribe: Tzu-Yi Chen*

This is the third of a series of lectures on probabilistic approximate metric spaces [Bartal96]. This lecture states a theorem relating probabilistic partitions to k-HST trees and then proves it. At the end of the lecture we briefly discuss why probabilistic approximate metric spaces are useful in practice.

21.1 Probabilistic Approximations by HSTs

Theorem 21.1 *(r, ρ, λ)-probabilistic partitions $\Rightarrow$ k-HST α-probabilistic-approximations with $\alpha = O(k\rho\lambda \log(\min\{\Delta, \frac{\rho}{epsilon}\}))$.*

The rest of this section is devoted to proving this theorem. The proof begins with a statement of an algorithm for constructing a k-HST tree and continues by showing that the algorithm constructs a k-HST which satisfies the conditions of the theorem.

For the algorithm we begin with the original graph and find a probabilistic partition at each stage.

At stage i, for $i = 1, 2, \ldots, t$, we have a graph G_i and a $r_i = \lfloor \frac{\text{diam}(G_i)}{\rho k} \rfloor$. At each stage we generate new graphs $C^1_{i+1} C^2_{i+1} \ldots C^s_{i+1}$. We then recurse on these new graphs, which ends up generating a tree. $G_{i+1} = C^i_{i+1} \Rightarrow T^j_{i+1}$ with root q^j_{i+1}.

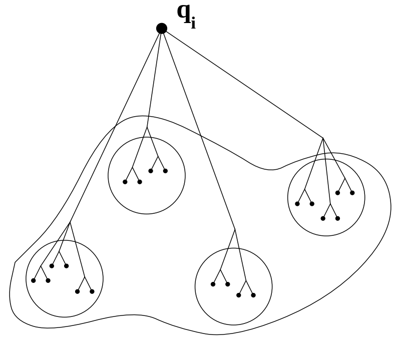

Figure 21.1: Example of a k-HST tree constructed from a partition.

We want to prove that this constructs a k-HST which is an *alpha*-probabilistic approximation. This requires showing a specific relationship between the weights along the edges, which corresponds to showing a specific relationship between the diameters of the clusters. In short, we need to show four things:

1. $\mathrm{diam}(G_{i+1}) \leq \rho r_i$, to prove that the tree constructed is a k-HST.
2. $\mathrm{diam}(T_i) \leq (1 + \frac{1}{k-1})\mathrm{diam}(G_i)$, to show that the tree created does not have a much bigger diameter than that of the original graph.
3. $\forall u, v \in V(G), d_{T_2}(u,v) \geq d_G(u,v)$, to give a lower bound for the ratio between the distances in the tree and in the original graph.
4. $\forall u, v \in V(G), E(d_{T_2}(u,v)) \leq \alpha d_G(u,v)$, to give an upper bound on $\frac{E(d_{T_2}(u,v))}{d_G(u,v)}$.

The rest of this section proves each of these four things separately.

Claim 21.2 *$diam(G_{i+1}) \leq \rho r_i$*

Proof: We want to show $\mathrm{diam}(G_{i+1}) \leq \rho r_i$. We know that $\mathrm{diam}(G_{i+1}) \leq \frac{\mathrm{diam}(G_i)}{k}$ and $\rho r_i \leq \frac{\mathrm{diam}(G_i)}{k}$ by construction. Therefore $\mathrm{diam}(G_{i+1}) \leq \rho r_i$ ∎

Claim 21.3 *$diam(T_i) \leq (1 + \frac{1}{k-1}) diam(G_i)$*

Proof: This is a proof by induction on i. Define $\gamma(T_i)$ to be the maximum weight of a path from the root to a leaf. The proof is as follows:

- **claim:** $\gamma(T_i) \leq \frac{1}{2}(1 + \frac{1}{k-1})\mathrm{diam}(G_i)$
- **base case:** The base case is trivial since everything is a node.
- **induction step:** We assume the claim holds for $i+1$ and prove it holds for i.

$$\begin{aligned}
\gamma(T_i) &\leq \max_{i \leq j \leq s} \gamma(T_{i+1}^j) + \frac{1}{2}\mathrm{diam}(G_i) \\
&\leq \max_{i \leq j \leq s} \left\{ \frac{1}{2}\left(1 - \frac{1}{k-1}\right) \mathrm{diam}(C_{i+1}^j) \right\} + \frac{1}{2}\mathrm{diam}(G_i) \\
&\leq \frac{1}{2}\left(1 + \frac{1}{k-1}\right) \cdot \frac{1}{d}\mathrm{diam}(G_i) + \frac{1}{2}\mathrm{diam}(G_i) \\
&= \frac{1}{2}\left(1 + \frac{1}{k-1}\right) \mathrm{diam}(G_i)
\end{aligned}$$

This implies $\mathrm{diam}(T_i) \leq 2 \cdot \gamma(T_i) \leq (1 + \frac{1}{k-1})\mathrm{diam}(G_i)$. ∎

Claim 21.4 $\forall u, v \in G_i, d_{T_i}(u,v) \geq d_{G_i}(u,v)$

Proof: This is also a proof by induction over i.

- **claim:** $d_{T_k}(u,v) \geq d_{G_k}(u,v)$

- **base case:** Again, the base case is trivial since both T_i and G_i consist of a single vertex.
- **induction step:** For the induction step we assume $d_{T_k}(u,v) \geq d_{G_k}(u,v)$ when $k = i+1$ and show that it holds for $k = i$.
  - If there exists $1 \leq j \leq s$ such that u and v belong to the same cluster C^j_{i+1}, then by induction $d_{T_i}(u,v) = d_{T^j_{i+1}}(u,v) \geq d_{C^j_{i+1}}(u,v) \geq d_{G_i}(u,v)$.
  - Otherwise u and v are in separate clusters. Then $d_{T_i} \geq 2 \cdot \frac{1}{2}\mathrm{diam}(G_i) = \mathrm{diam}(G_i) \geq d_{G_i}(u,v)$.

  In either case the claim holds for $k = i$.

■

Claim 21.5 $\forall u, v \in V(G), E(d_{T_i}(u,v)) \leq \alpha d_{G_i}(u,v)$

Proof: And the last claim is also proved by induction over i.

- **claim:** $E(d_{t_i}(u,v)) \leq \alpha \cdot d_{G_i}(u,v)$, where $\alpha = \rho\lambda k(1+\frac{1}{k-1})(h-i)$. We show that given some $h \leq t$, this claim holds.
- **base case:** In the base case everything is a vertex and the claim clearly holds.
- **induction step:** We assume the claim is true for $i+1$ and show that it holds for i.

  If there exists a j such that $u, v \in C^j_{i+1}$ and the shortest path between u and v is contained in C^j_{i+1}, then:

$$
\begin{aligned}
E(d_{T^j_{i+1}}(u,v)) &\leq \lambda\rho k\left(1+\frac{1}{k-1}\right)(h-i-1)\cdot d_{C^j_{i+1}}(u,v) \\
&= \lambda\rho k\left(1+\frac{1}{k-1}\right)(h-i-1)\cdot d_{G_i}(u,v)
\end{aligned}
$$

  If we let $x_i(u,v)$ be the probability that the shortest path between u and v is not contained in any single cluster, then:

$$
\begin{aligned}
E(d_{T_i}(u,v)) &\leq x_i(u,v)\cdot \mathrm{diam}(T_i) + (1-x_i(u,v)) \times \lambda\rho k\left(1+\frac{1}{k-1}\right)(h-i-1)\cdot d_{G_i}(u,v) \\
&\leq \lambda\frac{d_{G_i}(u,v)}{r_i}\cdot\left(1+\frac{1}{k-1}\right)\mathrm{diam}(G_i) + \lambda\rho k\left(1+\frac{1}{k-1}\right)(h-i-1)\cdot d_{G_i}(u,v) \\
&\leq \lambda\rho k\left(1+\frac{1}{k-1}\right)d_{G_i}(u,v) + \lambda\rho k\left(1+\frac{1}{k-1}\right)(h-i-1)\cdot d_{G_i}(u,v) \\
&\leq \lambda\rho k\left(1+\frac{1}{k-1}\right)(h-i)d_{G_i}(u,v)
\end{aligned}
$$

Since $h \leq t \leq \log_k \Delta$, we establish an upper bound of $O(\lambda\rho k \log_k \Delta)$. To show an upper bound of $O(\lambda\rho k \log_k(\min\{\Delta, \frac{\rho}{\epsilon}\}))$, we consider an $\frac{1}{n}$-forcing term.

Choose l to be the smallest partition such that $\frac{d_{G_l}(u,v)}{>}\epsilon$.

If $i < l$, then we know $x_i(u,v) = 0$, and hence the distance between u and v in T_i is equal to their distance in the $i+1$ level tree. By induction we obtain $E(d_{T_i}(u,v)) \leq \lambda\rho k(1+\frac{1}{k-1})(h-l)\cdot d_{G_i}(u,v)$.

Now we consider the smallest $i > l$ such that both u and v are in G_i. We know $\text{diam}(G_i) \geq d_{G_i}(u,v) \geq d_{G_l}(u,v)$. Therefore:

$$\frac{\text{diam}(G_i)}{\text{diam}(G_l)} \geq \frac{d_{G_l}(u,v)}{\rho k \cdot r_l} > \frac{\epsilon}{\rho k}$$

However, we also know $\text{diam}(G_i) \leq \frac{\text{diam}(G_l)}{k^{i-l}}$, which gives us a bound of $i - l - 1 \leq \log_k(\min\{\Delta, \frac{\rho}{\epsilon}\})$.

Therefore by selecting h appropriately, we obtain the upper bound of $O(\lambda\rho k \log_k(\min\{\Delta, \frac{\rho}{\epsilon}\}))$. ■

Claim 21.2 shows that the constructed tree is in fact a $k-HST$ tree. Claim 21.3 shows that the diameter of the constructed tree is not too much larger than the diameter of the original graph. Claims 21.4 and 21.5 show that this is an α-probabilistic approximation with $\alpha = O(k\rho\lambda \log(\min\{\Delta, \frac{\rho}{\epsilon}\}))$. This concludes the proof of Theorem 21.1.

21.2 Applications

In this section we describe various applications of probabilistic approximations of metric spaces.

Intuitively, these approximations are good because they permit you to approximate algorithms on trees and derive bounds. For example, using probabilistic approximations, you can show the following:

- An $O(k \log^2 n)$ bound for the k-server problem.
- An $O(\log^2 n)$ bound for the mobile user problem.
- An $O(\log m \log^2 n)$ bound for the migration version of the distributed paging problem (so there is only one copy of any page in the distributed system at any time — in other words, no caching of pages).

  The following figure gives some insight regarding how an approximate algorithm on a tree metric system could provide bounds for the distributed paging problem.

  The idea is that we have an algorithm for each subtree. These algorithms are all basically the uniprocessor paging algorithm. So if there's a request for page F in Figure 21.2, the cost of bringing in F should be proportional to the distance.

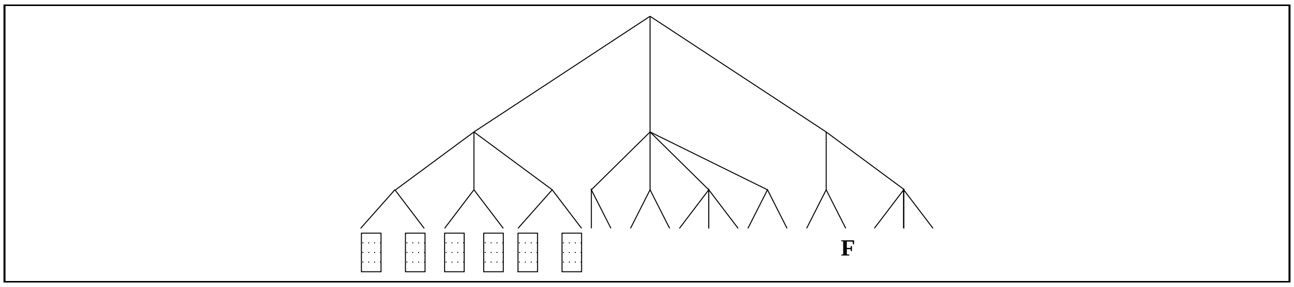

Figure 21.2: Example of a tree metric space for the distributed paging problem.

References

[Bartal96] Y. Bartal, *Probabilistic Approximation of Metric Spaces and its Algorithmic Applications*, in Proc. of the 37rd Ann. IEEE Symp. on Foundations of Computer Science, 1996.