9 Graph Cutting Procedures

Last class we began looking at how to embed arbitrary metrics into distributions of trees, and proved the following theorem due to Bartal (1996):

Theorem 9.1 (Bartal (1996)) *Given a metric (X, d) with diameter Δ, let $\mathcal{DT}$ be the set of all tree metrics that dominate d. Then,*

$$(X, d) \xhookrightarrow{O(\log n \log \Delta)} \mathsf{distr}(\mathcal{DT}).$$

In the last class, we proved Theorem 9.1 assuming the following theorem on graph decompositions:

Theorem 9.2 *Given a graph $G = (V, E)$ with edge lengths, and a parameter δ, there exists a procedure that deletes edges E' such that:*

1. *Each connected component C in $(V, E - E')$ has (weak) diameter smaller than δ.*
2. *$Pr[\text{edge } e \text{ is cut}] \leq 8 \log n \cdot (d(e)/\delta)$.*

In this class, we will describe two randomized algorithms that accomplish the required graph decomposition, which will implicitly define the distribution of trees that are used in the embedding.

Let us make two assumptions about the edge lengths $d(e)$:

1. The edge lengths are integers; i.e., $d(e) \in \mathbb{Z}^+$. This assumption can be discharged by taking rational approximations and scaling. Hence, the smallest edge length is 1.
2. We can assume for the above theorem that $d(e) \leq \frac{\delta}{8}$. Indeed, if edge e has $d(e) > \frac{\delta}{8}$, then we can just delete the edge. Since $\frac{8d(e)}{\delta \geq 1}$, the quantity $8 \log n \cdot (d(e)/\delta) \geq 1$ for $n \geq 2$, and hence the theorem still holds. This asusmption will be convenient later in the class.

9.1 Bartals Cutting Scheme

The first graph cutting procedure is essentially due to Bartal (1996). The algorithm is simple: it picks an arbitrary vertex, and constructs a component with a sufficiently small diameter by cutting all edges at a certain (random) distance away from the chosen vertex. The distance is essentially chosen from a distribution that falls off exponentially with distance from the vertex. It then removes the component from the graph, and iterates on the remaining graph until the remaining graph has a small diameter. We formalize this as follows:

Algorithm CUT-1(G, δ)

let $G' = G$.
while $G' \neq \emptyset$ **do**
Pick an arbitrary vertex $r \in G'$. Construct a shortest path tree in G' from r.

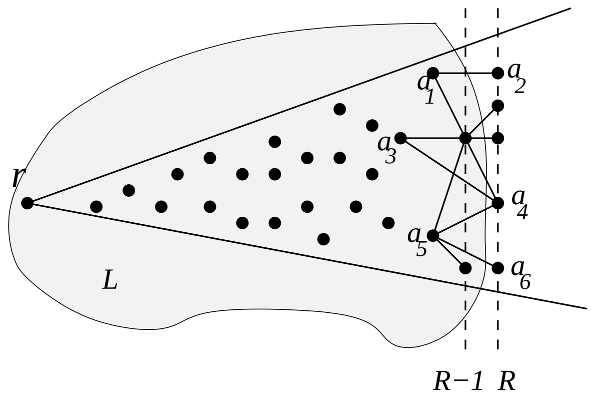

Figure 9.1: Illustrating the procedure CUT-1.

Let R be drawn from the geometric distribution with parameter $p = \frac{4 \log n}{\delta}$.
Cut all edges in G' crossing $[R-1, R]$ in the shortest path tree.
(I.e., edges having one endpoint at distance $< R$ and the other at distance $\geq R$ from r.)
Let L be the resulting component that contains the vertex r. let $G' \leftarrow G' \setminus L$.
end

Recall that sampling from a geometric distribution with parameter p is the same as flipping a coin with "heads" probability p, and returning the number of flips until we see a "heads". Hence, we can imagine the process as setting a counter $R \leftarrow 1$, and deciding the fate of the edges crossing from $R-1$ to R by flipping a coin with "heads" probability p: if the coin turns up tails, then it increments R (and hence "saves" the edges crossing from $R-1$ to R). If the coins turns up heads, it cuts those edges, removes the component thus created, and continues on the remaining graph.

The procedure is illustrated in Figure 9.1. Let r be the vertex chosen as the root from which we construct a shortest-path tree, and R be the (random) distance at which the procedure will cut the graph. The edges (a_1, a_2), (a_3, a_4), (a_5, a_4) and (a_5, a_6) (among others) are cut, and L is the connected component that is removed from the graph.

Now we verify that the procedure CUT-1 satisfies the two conditions required by Theorem 9.2.

Fact 9.3 *In the procedure* CUT-1*, for any edge e with weight $d(e)$*

$$\mathbf{Pr}[\textit{edge } e \textit{ is cut}] \leq 4 \log n \cdot (d(e)/\delta).$$

Proof. For this proof, it is more instructive to consider "cutting or saving" view of the algorithm: the chance that e is cut is precisely the chance that we get "heads" for some coin flip when the edge e is crossing the current interval $[R-1, R]$. Since there are at most $d(e)$

coin flips, and each one can come up heads with probability p, we get that

$$\begin{aligned}\mathbf{Pr}[\text{edge } e \text{ is cut}] &= 1-(1-p)^{d(e)} \\ &\leq 1-(1-pd(e)) = \frac{4\log n}{\delta}d(e) \ .\end{aligned}$$

■

Lemma 9.4 *The probability that some component C created in the algorithm has a diameter* $\mathsf{diam}(C) > \delta$ *is at most* $\frac{1}{n}$.

Proof. Consider any fixed component L: note that if the diameter of the component L exceeds δ, then R must have been $> \delta/2$. Hence,

$$\begin{aligned}\mathbf{Pr}[\mathsf{diam}(L) > \delta] &\leq Pr[R > \delta/2] \\ &= (1-p)^{\frac{\delta}{2}} \\ &\leq \exp\{-p\cdot\frac{\delta}{2}\} = \exp\{-\frac{4\log n}{\delta}\cdot\frac{\delta}{2}\} = \frac{1}{n^2} \ .\end{aligned}$$

Since there are at most n components created by the algorithm, the trivial bound implies the lemma. ■

Therefore, the procedure Cut-1 satisfies the conditions of Theorem 9.2 with probability at least $(1-\frac{1}{n})$. To satisfy Theorem 9.2, we can amplify the probability of success in one of two ways: we can either run the algorithm on any large components L (i.e., those with $\mathsf{diam}(L) > \delta$) remaining until all components are small enough; or, we can truncate the distribution of R at $\frac{\delta}{2}$.

Let us examine the truncation idea more closely. Suppose we truncate the distribution as follows: we follow the regular cutting procedure above; if we pick a value of R so that $R > \frac{\delta}{2}$, we simply set R to be $\frac{\delta}{2}$. Let the cuts at distance $\delta/2$ be said to be *special* cuts. Since a special cut is chosen with probability $1/n^2$ (due to the analysis in Lemma 9.4), and there are at most n such cuts

$$\mathbf{Pr}[\text{edge } e \text{ is cut due to a special cut}] \leq \frac{1}{n} \ .$$

If e.g. $\frac{1}{n} \leq \frac{2}{\delta}d(e)$, then the claims of Theorem 9.2 are satisfied (albeit with $\mathbf{Pr}$[edge e is cut] being $\frac{4\log n+2}{\delta}\cdot d(e)$). If not, though, the probability of cutting an edge is too high. To fix this, we do the following:

Algorithm Preprocessing

1. Contract all edges $e \in E$ with $d(e) < \frac{\delta}{2n}$.
2. Run the procedure Cut-1 on the resulting graph with $\hat{\delta} = \frac{\delta}{2}$.
3. Expand all the edges that were contracted in Step 1.

This procedure will allow us to have components no larger than δ in the original graph, since expanding the contracted edges in Step 3 will increase the diameter by at most $\frac{\delta}{2n}n = \frac{\delta}{2}$, and the cutting procedure creates graphs of diameter $\hat{\delta} = \frac{\delta}{2}$. (This, however, comes at a slightly higher price, since the bound on Pr[edge e is cut] has now increased to $(\frac{8\log n+2}{\delta}\cdot d(e)$, which is slightly higher than promised).

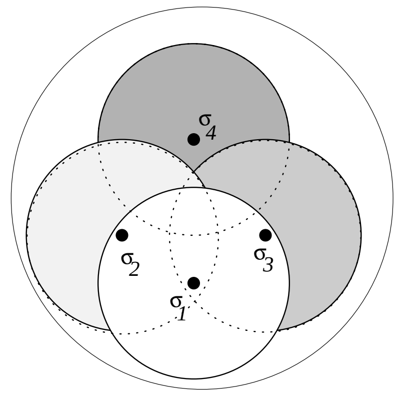

Figure 9.2: Illustrating the construction of $\hat{B}_i$.

9.2 The FRT-Cutting Scheme

While the procedure given above can be tightened further, we will move on to another partitioning procedure. This procedure first appeared in a paper of Calinescu, Karloff and Rabani (2001), and was later used by Fakcharoenphol, Rao and Talwar (2003) to achieve the $O(\log n)$ bound on the distortion for embedding arbitrary graphs into distributions of trees.

Algorithm Cut-2(G, δ)

Pick a radius R uniformly at random from $[\delta/4, \delta/2]$.
Pick a random permutation $\sigma \in S_n$, which defines an order $<_\sigma$ on the vertices.
For every vertex v_i, define a ball $B_i = B(v_i, R)$.
Assign each vertex to the "first" ball it lies in. Formally, define $\hat{B}_i = B_i - \bigcup_{j <_\sigma i} B_j$.
Delete all edges in the cut $(B_i, V \setminus B_i)$ for all i.

An example of this procedure is given in Figure 9.2. Balls of radius R are drawn around the first four vertices in a permutation σ, which are denoted by σ_1, σ_2, σ_3, and σ_4 respectively. The dotted lines indicate the balls B_i; $\hat{B}_1$ is the unshaded region, while $\hat{B}_2, \hat{B}_3$ and $\hat{B}_4$ are the shaded regions from lighest to darkest. Note that $B_1 \cap B_2$ (and thus vertex σ_2) belongs to $\hat{B}_1$, and $B_2 \cap B_4$ belongs to $\hat{B}_2$, and so on.

We now show that this procedure also results in a partition that satisfies the two conditions of the theorem. The first condition holds by construction – the weak diameter of each L_i is no more than the diameter of $\hat{B}_i$. But the weak diameter of $\hat{B}_i$ is no more than that of B_i; since the radius of each B_i is no more than $\frac{\delta}{2}$, the diameter is no more than δ.

We will prove something slightly stronger than the second condition:

Claim 9.5 *Given a vertex v and a radius ρ, the probability that the procedure* Cut-2 *cuts the ball $B(v, \rho)$ is at most* $8 \log n \cdot (\rho/\delta)$.

As can be expected, a set S is cut by the partitioning procedure if there are two components C_1 and C_2 in the partition such that vertices from S lie in both these components.

It is easy to see that Claim 9.5 implies the second condition in Theorem 9.2: given an edge $e = \{u, v\}$, consider the ball of radius $d(e)$ around u. Any partition that cuts the edge e also cuts the ball $B(u, d(e))$, and hence Claim 9.5 gives us the desired condition.

Let us now prove Claim 9.5. Since the names of the vertices do not matter in the algorithm, let us assume that they are numbered in order of their distance from u, i.e., $d(u, v_1) \leq d(u, v_2) \leq \ldots \leq d(u, v_n)$. Now for every vertex v_i in the graph, let us consider the following events. We say a node v_i *intersects* the Ball $B(u, \rho)$ if the random radius R is chosen such that $R \in [d(v_i, u) - \rho, d(v_i, u) + \rho]$ (this means it can possibly happen that nodes from $B(u, \rho)$ lie inside $\hat(B)_i$ and outside of $\hat(B)_i$). We say that a node *protects* the ball if $R > d(v_i, u) + \rho$, because then the ball is contained in the cluster generated by v_i and cannot be cut anymore by nodes that come after v_i in the permutation σ. Finally, we say a node v_i *cuts the ball first* if v_i intersects $B(u, \rho)$ and if no node prior to v_i in the permutation σ intersects or protects the ball. We can make the following observation.

Observation 9.6 *If the ball $B(u, \rho)$ is cut then there must exist a node v_i that* cuts it first *with respect to the definition above.*

This means

$$\mathbf{Pr}[B(u, \rho) \text{ is cut}] \leq \sum_i \mathbf{Pr}[v_i \text{ cuts } B(u, \rho) \text{ first}] \ ,$$

and it suffices to bound the probability that a node cuts the ball first.

Note that for the i-th node to cut a ball first, it must happen that R falls into the right range (i.e., $R \in [d(v_i, u) - \rho, d(v_i, u) + \rho]$) and further that all nodes v_j, $j < i$ come *after* v_i in the permuation σ, as all these nodes would either cut the ball or protect it (recall that we ordered the nodes according to the distance from u). The event thtat v_i precedes v_j, $j < i$ in σ holdes with probability $1/i$ and is independent from the choice of the radius R.

Altogether we get

$$\begin{aligned}\mathbf{Pr}[v_i \text{ cuts } B(u, \rho) \text{ first}] &\leq \mathbf{Pr}[R \in [d(v_i, u) - \rho, d(v_i, u) + \rho]] \cdot \frac{1}{i} \\ &= \frac{2\rho}{\delta/4} \cdot \frac{1}{i} = \frac{1}{i} \cdot \frac{8\rho}{\delta} \ .\end{aligned}$$

By summing over all i we get

$$\mathbf{Pr}[B(u, \rho) \text{ is cut}] \leq \sum_{i=1}^{n} \mathbf{Pr}[v_i \text{ cuts } B(u, \rho) \text{ first}] \leq \sum_{i}^{n} \frac{1}{i} \cdot \frac{8\rho}{\delta} = \frac{8\rho}{\delta} \cdot H_n \ , \qquad (9.1)$$

where $H_n = 1 + \frac{1}{2} + \frac{1}{3} \ldots + \frac{1}{n} = O(\log n)$ denotes the n-th haromnic number. Thus, the random partitions generated by procedure CUT-2 also meet the second condition of Theorem 9.2, and this proves the theorem.

9.2.1 Improved Embedding into Random Trees

We can slightly improve Equation 9.1 by observing that a node that is closer to u than $\delta/4-\rho$ or farther than $\delta/2+\rho$ cannot cut the ball $B(u,\rho)$ at all. Furthermore, we can assume $\rho \le \delta/8$ as otherwise Claim 9.5 trivially holds. This means we can restrict the sum in Equation 9.1 to nodes v_i that are *inside* the $\frac{5}{8}\delta$-ball around u, but *outside* the $\frac{1}{8}\delta$-ball. This gives

$$\mathbf{Pr}[B(u,\rho) \text{ is cut}] \le \sum_{i=|B(u,\delta/8)|}^{|B(u,\delta)|} \mathbf{Pr}[v_i \text{ cuts } B(u,\rho) \text{ first}] \le \frac{8\rho}{\delta} \cdot O\Big(\log\Big(\frac{|B(u,\delta)|}{|B(u,\delta/8)|} \Big)\Big) \ .$$

This improved bound on the probability of cutting an edge can be used to get an improved distortion for embedding into distributions over dominating trees. Recall that in the last lecture we showed that the expected length of an edge $e=(x,y)$ in the random tree when using the recursive partitioning approach is

$$\begin{aligned} & E[d_T(x,y)] \\ & \qquad = 4\Delta \cdot \Pr[x \text{ and } y \text{ are cut in level } 0] \\ & \qquad + 2\Delta \cdot \Pr[x \text{ and } y \text{ are cut in level } 1 \mid \text{they are in the same level 1 graph}] \\ & \qquad + \ldots \cdot \ldots \\ & \qquad + \tfrac{4\Delta}{2^j} \cdot \Pr[x \text{ and } y \text{ are cut in level } j \mid \text{they are in same level } j \text{ graph}\ ] \\ & \qquad + \ldots \cdot \ldots \ , \end{aligned}$$

where the sum has $\log\Delta$ terms. Now, using the partitioning scheme with $\delta = \Delta/2^{j+1}$ in level j gives

$$\Pr[x,\ y \text{ cut in level } j \mid \text{they are in same level } j \text{ graph}] \le \frac{8\cdot 1}{\Delta/2^{j+1}} \cdot O\Big(\log\Big(\frac{|B(u,\Delta/2^{j+1})|}{|B(u,\Delta/2^{j-2})|} \Big)\Big) \ ,$$

and we can bound the sum by

$$E[d_T(x,y)] \le \sum_{j=1}^{\log\Delta} 64 \cdot O\Big(\log\Big(\frac{|B(u,\Delta/2^{j+1})|}{|B(u,\Delta/2^{j-2})|} \Big)\Big) \le O(\log n) \ ,$$

since the sum telescopes. This gives an embedding into a distribution over dominating trees with distortion $O(\log n)$.

References

[Bar96] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.

[CKR01] Gruia Călinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 8–16, 2001.

[FRT03] Jittat Fakcharoenphol, Satish B. Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC)*, pages 448–455, 2003.