Advanced Algorithms 2012A
Lecture 14 – Graph Compression for Distances*

Robert Krauthgamer

1 Graph Sparsification for Distances I – Spanners

Let $G = (V, E)$ be an unweigted graph with $n = |V|$, and let $d_G(\cdot, \cdot)$ denote shortest-path distances in G. Suppose we want a subgraph G' that approximates all pairwise distances. We will see a tradeoff between the number of edges in G' and its (multiplicative) approximation.

Defn of k-spanner. A stretch k spanner (in short k-spanner) of $G = (V, E)$ is a subgraph $G' = (V, E')$ such that

$$\forall u, v \in V, \qquad d_G(u, v) \leq d_{G'}(u, v) \leq k \cdot d_G(u, v).$$

Observe this is equivalent to requiring

$$\forall (u, v) \in E, \qquad d_{G'}(u, v) \leq k \cdot d_G(u, v).$$

Exer: Show a graph G for which every 2-spanner has $\Omega(n^2)$ edges.

Theorem 1 [Awerbuch-Peleg'92, Althofer-Das-Dobkin-Joseph-Soares'93]. For every integer $t \geq 1$, every n-vertex graph G has a $2t - 1$ spanner with at most $n^{1+1/t}$ edges. Moreover, such a spanner can be computed greedily in polynomial time.

Notice the (small) difference from the bound stated in class. For instance, for stretch 3 (i.e. $t = 2$) we indeed get the bound $n^{3/2}$.

Greedy Algorithm. 1. Initialize G' to be an empty graph.

2. Scan the edges $(u, v) \in E$ in arbitrary order:

3. If currently $d_{G'}(u, v) > 2t - 1$, add the edge (u, v) to G'.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

4. Output G'.

The proof was shown in class using the following theorem.

Theorem 2. Let G be an n-vertex graph with girth (defined to be the length of the shortest cycle) $> 2t$, for an integer $t \geq 1$. Then G has less than $n^{1+1/t}$ edges.

The proof of for $t = 2$ was shown in class.

Exer: Extend Theorem 2 to general t.

Remark: For $t = 2$ this bound is tight, i.e. there are graphs with $\Omega(n^{3/2})$ edges, based on the projective plane. For general t, it is a conjecture of Erdös that above bound is tight. The bounds known are off by a constant factor in the girth, namely there are graphs with $n^{1+\Omega(1/t)}$ edges.

Exer: Extend Theorem 1 to the case of positive edge-lengths. (Note: edge-lengths are used to define distances and stretch; the size of the spanner is still the number of edges.)

Query time. A spanner $G' = (V, E')$ "encodes" (approximate) pairwise distances using $O(|E'|)$ memory words (each of $O(\log n)$ bits). While it is efficent in terms of space (memory), computing the distances in G' might not be very fast. Other data structure, called Distance Oracles [Thorup-Zwick'05], use roughly similar space but can answer distance queries in $O(1)$ time.

We will now see a different construction that achieves $O(\log n)$ stretch with $O(n \log^2 n)$ size, and fast queries.

Theorem 3. For every n-vertex graph G with edge-lengths ℓ_e that are integers between 1 and $W \geq n$, there is a graph G' with edge-lengths ℓ'_e that approximates all distances within an $O(\log n)$ factor and has $O(n \log n \log W)$ edges. Moreover, this spanner can be used to answer (approximate) distance queries in time $O(\log n \log W)$.

Remark: (1) G' need not be a subgraph of G', but rather of the of the complete graph defined by the metric d_G. (2) The technique actually extends to smaller stretch, but we will not discuss it here.

The proof uses the following theorem, which we actually proved several weeks ago as CKR/FRT clustering (here we have terminal set $T = V$).

Theorem 4 [Bartal, CKR/FRT]. For every metric d on a set of points V and parameter $R > 0$, there is a randomized construction of a partition P of V, such that

- With probability 1, every cluster $S \in P$ has $\mathrm{diam}(S) < 4R$.
- For every $u, v \in V$,

$$\Pr[u, v \text{ are separated in } P] \leq O(\log n) \cdot d(u, v)/R.$$

Moreover, such a randomized partition can be computed in polynomial time.

The proof was sketched in class. The main idea was to use Theorem 4 $O(\log n)$ times (independently) for each value $R = 2^i$ between 1 and the diameter of G (at most nW).

Exer: Think why we building a shortest-paths tree (BFS or DFS) from the center t of each cluster S_t might not work.

2 Graph Sparsification for Distances II – Terminal Distances

Let the input graph with edge lengths $G = (V, E, \ell)$ come with a set of *terminals* $T \subset V$, and we care only about distances between terminals: $d_G(t', t'')$ for $t', t'' \in T$. If we want to represent these distances *exactly*, we can certainly use a complete graph on T (with new edge lengths). But this G' is not similar to G topologically. So instead, we ask for G' that is a minor of G. Intuitively, we would contract irrelevant or redundant parts of the graph, finishing with a vertex set $T \subset V' \subset V$, i.e. V' includes some non-terminals, but is relatively small. If we could do this, then e.g. a planar G would yield a planar G', which would be useful if the intended computation on T is say TSP, where planar graphs admit better (approximation) algorithms than general graphs.

For illustration, suppose G is a tree. Then we can "contract" all non-terminals of degree ≤ 2 (and adjust edge lengths). The resulting tree G' would have at most $|T|$ leaves, and thus size $\leq 2|T| - 1$.

Exer: Show that a tree with no vertices of degree 2 and at most $|T|$ leaves (degree 1), has size $\leq 2|T| - 1$.

Theorem 5. For every graph with edge length $G = (V, E, \ell)$ and terminals $T \subset V$, there is a minor with new edge length $G' = (V', E', \ell')$, such that $V' \supset T$ and

$$\forall t', t'' \in T, \qquad d_G(t', t'') = d_{G'}(t', t''),$$

and $|V'| \leq |T|^4$.

The proof was sketched in class.

Open question. Is it possible to do better than $O(|T|^4)$? Even for planar graphs? The currently known lower bound [Krauthgamer-Zondiner] is $|V'| \geq \Omega(|T|^2)$, and holds even for planar graph (in the worst-case).

3 Course recap via the story of Vertex Cover

The goal here is to get a high level view of the topics seen during the course, and there will be less emphasis on proofs.

The Vertex Cover Problem: Given a graph $G = (V, E)$, find a minimum-size subset $S \subset V$ that covers all the edges, i.e. for every edge $e \in E$, at least one of its endpoints is in S.

The problem is NP-hard, even to approximate within 1.36.

Exer: Show that in every graph, the minimum size of a vertex cover is equal to $|V|$ minus the maximum size of an independent set.

Theorem. Vertex Cover can be approximated within factor 2 in polynomial time.

LP relaxation. We shall have a variable x_i for every vertex $i \in V$.

$$\boxed{\begin{array}{lll} \text{minimize} & \sum_{i\in V} x_i & \\ \text{subject to} & x_i + x_j \geq 1 & \forall (i,j) \in E \\ & x_i \geq 0 & \forall i \in V. \end{array}} \tag{1}$$

A proof was sketched in class, essentially by rounding an optimal LP solution $\hat{x}$ as follows:

$$x_i = \begin{cases} 1 & \text{if } \hat{x}_i \geq 1/2; \\ 0 & \text{otherwise,} \end{cases}$$

Theorem (half-integrality). The vertex-cover LP always has an optimal solution $\hat{x}$ that is half-integral, i.e., all $\hat{x}_i \in \{0, \frac{1}{2}, 1\}$.

Remark: Related to, but more subtle than, total unimodoluarity.

Duality to matching. So it is not surprising that the dual of vertex-cover LP asks to pack edges (without "violating" the vertices):

$$\boxed{\begin{array}{lll} \text{maximize} & \sum_{e\in E} y_e & \\ \text{subject to} & \sum_{j:ij\in E} y_{ij} \leq 1 & \forall i \in V \\ & y_e \geq 0 & \forall e \in E. \end{array}} \tag{2}$$

This is a relaxation of Maximum Matching, the problem of finding a maximum size subset of edges $M \subset E$ such that two edges are adjacent (touch the same vertex).

A matching is called *maximum* if it has maximum size, and is called *maximal* if it is maximal with respect to containment.

Approximating vertex-cover via maximal matching. Find a maximal matching $M \subset E$ (say greedily, i.e. scan all edges and add $e \in E$ to M if possible), and then let the vertex-cover be all the endpoints of M.

Combinatorial analysis: a vertex cover must cover the edges of M, each by a distinct vertex. Thus $OPT_{VC} \geq |M|$. The algorithm outputs a set of size $ALG = 2|M| \leq 2OPT_{VC}$.

Analysis via LP: M is a feasible solution to the dual, hence by weak duality $|M| \leq LP_{VC}$, and again $ALG = 2|M| \leq 2LP_{VC} \leq 2OPT_{VC}$.

Open question. Can Vertex Cover be approximated within factor better than 2?

Bipartite graphs. In bipartite graphs, vertex-cover can be solved in polynomial time. It can be shown directly, but also via in our LP above; the matrix A is really the vertex-edge incidence matrix of G, and recall we saw (in homework) that this matrix is totally unimodular. The same applies to the dual LP with respect to maximum matchings, and thus maximum matching is equal to minimum vertex cover (in bipartite graphs).

Planar graphs. In planar graphs, vertex cover is still NP-hard, but it can be approximated within factor $1+o(1)$. The first step is to show that we may assume $OPT \geq n/2$, e.g., using some argument that uses the half integrality. Then we can use the planar separator theorem repeatedly until we break the graph to pieces of at most $\log n$ vertices. We solve each piece optimally by exhaustive search, and add to the solution all the removed vertices (from all separators), which are most $O(n/\sqrt{\log n}) \leq O(OPT/\sqrt{\log n})$. It can be seen that this gives $1 + O(1/\sqrt{\log n})$ approximation.