Approximate Distance Preservers

Parv Mor, 160479
CS648 — Randomized Algorithms

November 4, 2018

1 Aim

Given an undirected graph $G = (V_G, E_G)$ where each edge has positive weight find a sparse graph H of G that preserves distance by a given stretch $t = 2k - 1$ for all pair of nodes. Number of edges in the sparse graph H should be $\mathcal{O}(n^{1+\frac{1}{k}})$ and expected running time to compute H must be $\mathcal{O}(km)$.

2 Certain Insights

- Let us denote distance between two vertex, u and v in graph G by, $\delta_G(u, v)$. Also let weight of an edge $(u, v) \in E_G$ be defined as $w_G(u, v)$. If H is a sparse graph of G with stretch t then the following property must hold:

$$\forall\ u, v \in V_H, \delta_G(u, v) \leq \delta_H(u, v) \leq t \times \delta_G(u, v) \tag{1}$$

- We always ensure that any edge in H is also present in G. Adding new edges does not make sense. Hence, $E_H \subseteq E_G$ and $V_H = V_G = V$.

- Since $E_H \subseteq E_G$, we have already ensured the lower bound of (1).

- For any edge, say $(a, b) \in E_G \setminus E_H$, we should always try to ensure that there is a path between a and b in H such that its weight is $\leq t \times w_G(a, b)$. Since we want to avoid computing distance between nodes in H due to required linear time, we can ensure a simple and much more stronger condition on the required path. We should try to ensure that there is a path in H between a and b such that its length is $\leq t$ and weight of any edge in this path is atmost $w_G(a, b)$.

- See that any path between a and b in G can be reduced to a walk in H if above condition is ensured. Also the weight of this walk is $\leq t \times \delta_G(a, b)$. Moreover, this walk can be reduced to a path in H and hence, the upper bound of (1) will also be satisfied.

3 A Stretch of 3

We require a sparse graph H, such that $|E_H| = \mathcal{O}(n^{1.5})$ and cost of its computation given G is $\mathcal{O}(|E_G|)$ on expectation. We will be doing some kind of random sampling. If we try to randomly sample edges of G, we have an issue of maintaining connectivity. This seems to be hard. So instead let us randomly sample vertices with some probability p. Let the random sample be denoted by S. $\mathbb{E}[|S|] = np$.
For each node v in S, let $S_v = \{v\}$. For each node in $V \setminus S$ we would try to add edges incident on it into E_H using the following method:

- Let $X = (V \setminus S) \cap (V \setminus N(S))$. For each node in X add all edges incident on it into E_H. Consider any such node v in X with degree d. Expected number of edges that are added into E_H by such a node is $d(1-p)^d$.

- Let $Y = (V \setminus S) \cap N(S)$. Consider any such node u in Y with degree d. Find a $v \in S$ such that $w_G(u, v)$ is least. Update S_v to $S_v \cup \{u\}$. Add all edges incident on u in E_H such that their weight is less than $w_G(u, v)$. Also add (u, v) to E_H. Expected number of edges added by such a node is $\sum_{j=0}^{d-1}(j+1)(1-p)^j p$.

Hence,

$$\mathbb{E}[|E_H|] = \mathcal{O}(\frac{n}{p}) \quad \text{Using the infinite A.G.P summation}$$

But note that we still haven't guaranteed connectivity in H. Each node either belongs to a set S_v or it is connected via a path in H to a set S_v. All nodes in a set S_v are connected. The edges not present in E_H are always connected by two members from two sets. Hence, we only need to connect this sets somehow without blowing up the size and maintain a stretch of 3.
Also see that any edge between nodes in the same set that is not present in E_H can be replaced by a path of H. Consider any two such nodes a and b. Let them belong to the set S_v. Consider the path $a \to v \to b$. We already have $w_G(a, v) \leq w_G(a, b)$ and $w_G(v, b) \leq w_G(a, b)$. Hence, $a \to v \to b$ is a 2 stretch path in H of G.
So consider all the edges that $\notin E_H$ and have their endpoints in different sets. For any such endpoint, say a, find all sets that a is connected to. For each such set (say S_u) add the edge with minimum weight (say (a, b)) into E_H. Reason for this is as follows:
Consider a edge $(a, c), c \in S_u$ not added into E_H. The path $a \to b \to u \to c$ is of stretch 3 in H of G as $w_G(a, b) \leq w_G(a, c)$ and $w_G(u, c) \leq w_G(a, c)$ and $w_G(b, u) \leq w_G(b, a) \leq w_G(a, c)$.
By this step,

$$\mathbb{E}[|E_H|] = \mathcal{O}(\frac{n}{p}) + \mathcal{O}(n\mathbb{E}[|S|]) = \mathcal{O}(\frac{n}{p} + n^2 p)$$

This expression will be minimized when $p = n^{-0.5}$. Hence,

$$\mathbb{E}[|E_H|] = \mathcal{O}(n^{1.5})$$

Note that the size is expected not worst case. But the time complexity is deterministic. Any given step above requires only constant number of traversal through the adjacency list of G.

Hence, time complexity is $\mathcal{O}(|E_G|)$. To achieve the aim we need to invert this. This can be done by the following observation:

$$P(|E_H| > 2\mathbb{E}[|E_H|]) \leq \frac{1}{2} \quad \text{By Markov's Inequality}$$

Hence, repeat until size is what we desire. The size for H is now $\mathcal{O}(n^{1.5})$ in worst case. But the time complexity on expectation is $\mathcal{O}(|E_G|)$.

4 A stretch in general

Now we want to find out a sparse graph H of G with a stretch $t = 2k - 1$.
If we want to adopt our 3 stretch approach we would require that $n^2 p \approx n^{1+\frac{1}{k}}$. This is due to connectivity of sets step. Therefore, $p = \frac{1}{n^{1-\frac{1}{k}}}$. This means that $\mathbb{E}[|S|] = n^{\frac{1}{k}}$. But building of sets step would give us $\mathcal{O}(n^{2-\frac{1}{k}})$ edges with this p. To counter this we could try running the same for say t iterations. But since we are sampling nodes instead of edges we do not have a dependence of edges in sparse graph on edges in original graph.

Instead suppose we use p from building of sets step. This would give us $\frac{n}{p} = n^{1+\frac{1}{k}}$. $\implies p = n^{-\frac{1}{k}}$. But then connecting of sets would contribute $\mathcal{O}(n^{2-\frac{1}{k}})$ edges in E_H. However this we can counter by running the same for $k-2$ more times decreasing the contribution to $\mathcal{O}(n^{1+\frac{1}{k}})$. But note that this increases the contribution of building of sets step by a factor of k because in each iteration we will be adding $\mathcal{O}(n^{1+\frac{1}{k}})$ edges.
Let the randomly sampled set at end of $(i-1)^{th}$ iteration be S_{i-1}. We are going to create S_i by randomly sampling sets from S_{i-1} each with probability $n^{-\frac{1}{k}}$. All nodes present in S_{i-1} and not present in a sampled set are processed in a similar manner as in stretch of 3.
This gives us a sparse graph of size $\mathcal{O}(kn^{1+\frac{1}{k}})$ with computation time $\mathcal{O}(km)$ on expectation.

But somehow we want to get rid of the k factor in size. One way to do this would be to find out probability of deviation of mean of $|E_H|$ w.r.t desired value. But this would drastically increase the runtime of algorithm.