Reflection on Project

Shengxian Chen, Weiyuan Chen

May 2, 2020

1 OUTLINE

In this project, we are investigating an NP-Hard graph problem that minimizes the average pairwise distance of an undirected, connected, and weighted graph. We decided to use an approximation algorithm to approach this problem.

2 ALGORITHM

Input: $G = (V, E)$ and set d (i.e. A list of edge distance).

Output: Tree T that has the minimized average pairwise distance.

1. Find the MST of G, denoted by T.

2. Let $heap[]$ be a priority queue.

3. Put every leaf node's incident edge e in T into $heap[]$, sorted by $D(e)$, where $e = (n, i)$ and $i \in E$.

4. Removes all leaf node in T and its incident edges.

5. Let $curDist := average\_pairwise\_distance(T)$. Let $minEdge := delMin(heap)$. Create T' where T' is constructed by adding $minEdge$ into T.

6. while $(curDist <= average\_pairwise\_distance(T'))$: do step 5.

7 Remove minEdge and its leaf node in T', return T'.

3 PROOFS

Let OPT be the tree that minimizes the average pairwise distance in G, and let RES be the tree that minimizes the total distance in G. Notation: Let $|A|$ be the number of vertices in graph A.

Claim: The algorithm above produces RES.

Proof: The algorithm finds the MST for the given graph, this guarantees that it minimizes the total pairwise distance of the tree. But since the problem allows if a vertex is adjacent to the tree also counts to be a part of the tree. We are safe to remove all the leafnode, which helps to reduce the pairwise distances. But notice the increasing the nodes number contribute to the decrease of average pairwise distance, therefore, we keep adding the deleted minimum weight edge into the MST and until the new $W(T) >$ old $W(T)$. The algorithm gives us the RES, if not, suppose there exists another T' that has a less weight than the tree produced from the algorithm, denoted T. We can start from T' and add all the missing edges from the MST we just produced, this gives us a new tree MST', where $W(MST') = W(T') + W(newAddedEdge)$ and $W(MST) = W(T) + W(newAddedEdge)$. This implies that $W(MST') < W(MST)$, which is a contradiction. Therefore, the algorithm must produce RES. ∎

Claim: If $|OPT| \leq |RES|$, then $OPT = RES$.

Proof: Notice the definition of average pairwise distance is that

$$\frac{1}{\binom{|T|}{2}} \sum_{(u,v)\in(T\times T)} d(u,v)$$

we can choose to minimize the part $\sum_{(u,v)\in(T\times T)} d(u,v)$ first. Let OPT be the optimal tree that minimize the average pairwise distance. Let RES be the subtree in G such that $\sum_{(u,v)\in(T\times T)} d(u,v)$ being minimized. We will argue that if our claim satisfies, $minG$ must be a subtree of $optG$.

Case 1: $|OPT| = |RES|$

Proceed by contradiction. Suppose $OPT \neq RES$, by definition,

$$cost(OPT) = \frac{1}{\binom{|OPT|}{2}} \sum_{(u,v)\in(OPT\times OPT)} d(u,v)$$

$$cost(RES) = \frac{1}{\binom{|RES|}{2}} \sum_{(u,v)\in(RES\times RES)} d(u,v)$$

since $|OPT| = |RES|$, so $\frac{1}{\binom{|OPT|}{2}} = \frac{1}{\binom{|RES|}{2}}$, but RES has a total weight that is minimized, which means $\sum\limits_{(u,v)\in(RES\times RES)} d(u,v) \leq \sum\limits_{(u,v)\in(OPT\times OPT)} d(u,v)$. This means that $cost(RES) \leq cost(OPT)$, which is a contradicts that $cost(OPT)$ has been minimized.

Case 2: $|OPT| < |RES|$.

Notice that $cost(OPT) \leq cost(RES)$:

$$\frac{1}{\binom{|OPT|}{2}} \sum_{(u,v)\in(OPT\times OPT)} d(u,v) \leq \frac{1}{\binom{|RES|}{2}} \sum_{(u,v)\in(RES\times RES)} d(u,v)$$

since $\frac{1}{\binom{|OPT|}{2}} > \frac{1}{\binom{|RES|}{2}}$, this requires that

$$\sum_{(u,v)\in(OPT\times OPT)} d(u,v) \leq \sum_{(u,v)\in(RES\times RES)} d(u,v)$$

which contradicts that RES has the total distance being minimized. ■

Claim: If $|OPT| > |RES|$, then $cost(RES) \leq (1 + \frac{c}{m})^2 cost(OPT)$, where $m = |RES|$ and $c = |OPT| - |RES|$.

Proof:

$$cost(OPT) = \frac{1}{\binom{m+c}{2}} \sum_{(u,v)\in(OPT\times OPT)} d(u,v)$$

$$cost(RES) = \frac{1}{\binom{m}{2}} \sum_{(u,v)\in(RES\times RES)} d(u,v)$$

Note that it's easy to see $c > 0$.

$$\begin{aligned}
\frac{cost(OPT)}{cost(RES)} &= \frac{\frac{1}{\binom{m+2}{2}} \sum_{(u,v)\in(OPT\times OPT)} d(u,v)}{\frac{1}{\binom{m}{2}} \sum_{(u,v)\in(RES\times RES)} d(u,v)} \\
&= \frac{\frac{2}{(m+c)(m+c-1)}}{\frac{2}{m(m-1)}} \times \frac{\sum_{(u,v)\in(OPT\times OPT)} d(u,v)}{\sum_{(u,v)\in(RES\times RES)} d(u,v)} \\
&\geq \frac{\frac{2}{(m+c)(m+c-1)}}{\frac{2}{m(m-1)}} \\
&= \frac{m(m-1)}{(m+c)(m+c-1)} \\
&\approx \frac{m^2}{(m+c)^2}
\end{aligned} \tag{1}$$

Therefore,

$$\begin{aligned}
\frac{cost(OPT)}{cost(RES)} &\geq \frac{m^2}{(m+c)^2} \\
cost(RES) &\leq \left(\frac{m+c}{m}\right)^2 cost(OPT) \\
cost(RES) &\leq \left(1 + \frac{c}{m}\right)^2 cost(OPT)
\end{aligned} \tag{2}$$

■

4 RUNTIME ANALYSIS

$$
\begin{aligned}
TimeComplexity &= T(MST) + Edges + T(leafNodes) \\
&= O(Elog(V)) + O(E) + O(V) \\
&= O(Elog(V))
\end{aligned}
\tag{3}
$$