ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2022

Prof. R. Wattenhofer
Florian Grötschla
Andrei Constantinescu

Principles of Distributed Computing
Exercise 13

1 GNNs for Algorithmic Problems

Recall from the lecture that we can define a GNN in terms of two functions:

$$a_v{}^{(t)} = \text{AGGREGATE}\left(\{\{h_u{}^{(t-1)} \mid u \in N(v)\}\}\right)$$

and

$$h_v{}^{(t)} = \text{UPDATE}\left(h_v{}^{(t-1)}, a_v{}^{(t)}\right),$$

where Aggregate has to be permutation-invariant. In this question your task is to give concrete implementations of Aggregate and Update to solve certain tasks.

a) Give Aggregate and Update functions to implement a shortest path algorithm (for unweighted undirected graphs) from one source node to all other nodes. We represent the source node s by setting $h_s^{(0)} = 0$ and $h_v^{(0)} = \infty$ for $v \in V \setminus \{s\}$. After $n-1$ steps of the GNN, the state $h_v^{(n-1)}$ of every node $v \in V$ should be its shortest path distance to s.

b) Prove that it is impossible to solve the vertex cover problem with a GNN on a graph where $h_v^{(0)} = 0$ for all $v \in V$ (all states are initialized to the same value)?

c) Give an example of a tree where a GNN is not able to compute a minimal vertex cover on.

d) Implement the functions Aggregate and Update to compute a minimal (not necessarily minimum) vertex cover **for trees with an initial two-coloring** (the initial state of a node is either 0 or 1, corresponding to its color). You can decide how you want to represent nodes that are part of the vertex cover and those that are not in $h_v^{(k)}$ after k iterations. How many iterations k do you need?

2 The Weisfeiler-Lehman Test for Trees

The goal of this question is proving that WL is an exact isomorphism test for trees. For the first parts, we use $G(V, E)$ to denote a general undirected graph.

a) Give an example of a graph G for which WL requires $\Omega(|V|)$ rounds to finish; i.e. for the color classes to not refine anymore.

For the following parts, we will assume that the number of rounds t that WL is run for is provided as input to the algorithm, rather than iteratively increasing t until the color partition converges. For a node $v \in V$, define the k-hop neighborhood of v in G to be the induced subgraph of G consisting of all nodes at distance at most k from v.

b) Given the computed label $s_v^{(t)}$ of some node $v \in V$, show that this information does not suffice to uniquely identify the t-hop neighborhood of v in G up to isomorphism. Formally, show that there exists a value of t such that there are two graphs $G_1(V_1, E_1), G_2(V_2, E_2)$ and two nodes $v_1 \in V_1, v_2 \in V_2$ such that $s_{v_1}^{(t)} = s_{v_2}^{(t)}$, but the t-hop neighborhoods of v_1 in G_1 and v_2 in G_2 are non-isomorphic.

Now, assume graph G is a tree. For a node $v \in V$, define the **rooted** k-hop neighborhood of v in G to be the induced subtree of G consisting of all nodes at distance at most k from v, rooted at v. Formally, a rooted tree is a pair (G, r), where G is a tree and $r \in V$ is the designated root of G.

c) Give an example of two non-isomorphic rooted trees (G_1, r_1) and (G_2, r_2) such that the non-rooted trees G_1 and G_2 are isomorphic.

d) Given the computed label $s_v^{(t)}$ of some node $v \in V$, show that this information uniquely identifies the rooted t-hop neighborhood of v in G up to rooted tree isomorphism. Formally, show that given two graphs G_1, G_2 and two nodes $v_1 \in V_1, v_2 \in V_2$ it holds that $s_{v_1}^{(t)} = s_{v_2}^{(t)}$ if and only if the rooted t-hop neighborhoods of v_1 in G_1 and of v_2 in G_2 are isomorphic as trees rooted at v_1 and v_2.

e) Show that if $t \geq \Delta$, where Δ is the diameter of G, then for any $v \in V$ the value $s_v^{(t)}$ uniquely identifies the tree G up to isomorphism. Moreover, show that if $t \geq \Delta/2$ then there is a node $v \in V$ such that $s_v^{(t)}$ uniquely identifies the tree up to isomorphism.

For the remaining parts, the number t of rounds will be the least number of rounds required for the color classes to stabilize. The precise WL implementation we are targeting is in Algorithm 1, including the stopping condition and returned values. Pay particular attention to the comment next to the returned values.

Algorithm 1 State Refinement/Weisfeiler-Lehman (WL)

$t \leftarrow 0$
for $v \in V$ **do**
  $s_v^{(0)} \leftarrow 0$
end for
do
  $t \leftarrow t + 1$
  for $v \in V$ **do**
    $s_v^{(t)} \leftarrow \text{RELABEL}(s_v^{(t-1)}, \{\{s_u^{(t-1)} \mid u \in N(v)\}\})$
  end for
while $(s_v^{(t)})_{v \in V}$ and $(s_v^{(t-1)})_{v \in V}$ induce different partitions.
return $\{\{s_v^{(t)} \mid v \in V\}\}$ $\quad\triangleright$ $t-1$ is more usual here, but some proofs become more difficult.

f) Show that there are infinitely many trees for which $t < \Delta/2$.

g) Show that given the multiset $\{\{s_v^{(k)} \mid v \in V\}\}$ for some $k \geq t$, then the multiset $\{\{s_v^{(k+1)} \mid v \in V\}\}$ is uniquely determined.

h) Show that two trees are isomorphic if and only if the multisets returned by WL when ran on the two trees are the same. **Hint:** use parts **e)** and **g)**.