Injective analysis

In this section, we will briefly analyze why we can get different representations from different subgraph structures. According to GIN, GNN currently used for graph-level tasks can be divided into two steps as shown below.

$$h_v^{(k)} = AGGREGATE^{(k)}(h_v^{(k-1)}, COMBINE^{(k)}(N_v^{(k-1)})) \tag{1}$$

$$h_G = READOUT(\{h_v^k | v \in G\}) \tag{2}$$

where $N_v^{(k)}$ denotes the k-layer neighborhood information of node v. If the neighborhood combine, aggregation, and graph-level readout functions are injective, the resulting GNN is as powerful as Weisfeiler-Lehman (WL) isomorphism test [Weisfeiler and Leman(1968)] that works well in general but with a few exceptions such as regular graphs [Cai et al.(1992)]. WL test is sufficient for our isomorphism discrimination.

For node v_j in a directed graph G, the feature combination of all its incoming edges e_{ij} and the starting nodes v_i of these incoming edges form a multiset $\{(h_i, f_{e_{ij}})\}$. Certain popular injective set functions such as mean aggregators are not injective functions over multisets, while sum aggregators can represent injective[Zaheer et al.(2017)]. Thanks to the universal approximation theorem [Hornik et al.(1989), Hornik(1991)], we can use MLPs over summation of representations to model sum aggregators as in GIN [Xu et al.(2019)]. Therefore, recall our modified EGIN convolve and readout equations:

$$h_v^{(k)} = MLP_n^{(k)}((1+\epsilon^{(k)})h_v^{(k-1)} + \textstyle\sum_{e \in IN(v)} MLP_0(h_{S_e}^{(k-1)} || f_e^{(k-1)})) \tag{3}$$

$$f_e^{(k)} = MLP_e^{(k)}(f_e^{(k-1)} || (h_{S_e}^{(k-1)}) + h_{E_e}^{(k-1)}) \tag{4}$$

$$h_G = MLP^{(R)} \textstyle\sum_{v \in G} h_v^{(K)} \tag{5}$$

Equation (4) and the right part of equation (3) guarantee the injective property of the combine function in equation(1), while the left part of equation (4) and equation (5) guarantee the injective property of aggregate function and readout function respectively.

References

[Cai et al.(1992)] Jin-Yi Cai, Martin Fürer, and Neil Immerman. 1992. An optimal lower bound on the number of variables for graph identification. *Combinatorica* 12, 4 (1992), 389–410.

[Hornik(1991)] Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks* 4, 2 (1991), 251–257.

[Hornik et al.(1989)] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.

[Weisfeiler and Leman(1968)] Boris Weisfeiler and Andrei Leman. 1968. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series* 2, 9 (1968), 12–16.

[Xu et al.(2019)] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*.

[Zaheer et al.(2017)] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. *Advances in neural information processing systems* 30 (2017).