2.5 Message Passing on Heterogeneous Graph

(中文版)

Heterogeneous graphs (1.5 Heterogeneous Graphs), or heterographs for short, are graphs that contain different types of nodes and edges. The different types of nodes and edges tend to have different types of attributes that are designed to capture the characteristics of each node and edge type. Within the context of graph neural networks, depending on their complexity, certain node and edge types might need to be modeled with representations that have a different number of dimensions.

The message passing on heterographs can be split into two parts:

1. Message computation and aggregation for each relation r.
2. Reduction that merges the aggregation results from all relations for each node type.

DGL's interface to call message passing on heterographs is `multi_update_all()`. `multi_update_all()` takes a dictionary containing the parameters for `update_all()` within each relation using relation as the key, and a string representing the cross type reducer. The reducer can be one of `sum`, `min`, `max`, `mean`, `stack`. Here's an example:

```
import dgl.function as fn

for c_etype in G.canonical_etypes:
    srctype, etype, dsttype = c_etype
    Wh = self.weight[etype](feat_dict[srctype])
    # Save it in graph for message passing
    G.nodes[srctype].data['Wh_%s' % etype] = Wh
    # Specify per-relation message passing functions: (message_func, reduce_func).
    # Note that the results are saved to the same destination feature 'h', which
    # hints the type wise reducer for aggregation.
    funcs[etype] = (fn.copy_u('Wh_%s' % etype, 'm'), fn.mean('m', 'h'))
# Trigger message passing of multiple types.
G.multi_update_all(funcs, 'sum')
# return the updated node feature dictionary
return {ntype : G.nodes[ntype].data['h'] for ntype in G.ntypes}
```