

1.3 节点和边的特征

(English Version)

`DGLGraph` 对象的节点和边可具有多个用户定义的、可命名的特征，以储存图的节点和边的属性。通过 `ndata` 和 `edata` 接口可访问这些特征。例如，以下代码创建了2个节点特征（分别第8、15行命名为 `'x'`、`'y'`）和1个边特征（在第9行命名为 `'x'`）。

```
1  >>> import dgl
2  >>> import torch as th
3  >>> g = dgl.graph(([0, 0, 1, 5], [1, 2, 2, 0])) # 6个节点, 4条边
4  >>> g
5  Graph(num_nodes=6, num_edges=4,
6         ndata_schemes={},
7         edata_schemes={})
8  >>> g.ndata['x'] = th.ones(g.num_nodes(), 3)           # 长度为3的节点特征
9  >>> g.edata['x'] = th.ones(g.num_edges(), dtype=th.int32) # 标量整型特征
10 >>> g
11 Graph(num_nodes=6, num_edges=4,
12        ndata_schemes={'x' : Scheme(shape=(3,), dtype=torch.float32)},
13        edata_schemes={'x' : Scheme(shape=(,), dtype=torch.int32)})
14 >>> # 不同名称的特征可以具有不同形状
15 >>> g.ndata['y'] = th.randn(g.num_nodes(), 5)
16 >>> g.ndata['x'][1]           # 获取节点1的特征
17 tensor([1., 1., 1.])
18 >>> g.edata['x'][th.tensor([0, 3])] # 获取边0和3的特征
19 tensor([1, 1], dtype=torch.int32)
```

关于 `ndata` 和 `edata` 接口的重要说明：

- 仅允许使用数值类型（如单精度浮点型、双精度浮点型和整型）的特征。这些特征可以是标量、向量或多维张量。
- 每个节点特征具有唯一名称，每个边特征也具有唯一名称。节点和边的特征可以具有相同的名称（如上述示例代码中的 `'x'`）。
- 通过张量分配创建特征时，DGL会将特征赋给图中的每个节点和每条边。该张量的第一维必须与图中节点或边的数量一致。不能将特征赋给图中节点或边的子集。
- 相同名称的特征必须具有相同的维度和数据类型。
- 特征张量使用“行优先”的原则，即每个行切片储存1个节点或1条边的特征（参考上述示例代码的第16和18行）。

对于加权图，用户可以将权重储存为一个边特征，如下。



```
>>> # 边 0->1, 0->2, 0->3, 1->3
>>> edges = th.tensor([0, 0, 0, 1]), th.tensor([1, 2, 3, 3])
>>> weights = th.tensor([0.1, 0.6, 0.9, 0.7]) # 每条边的权重
>>> g = dgl.graph(edges)
>>> g.edata['w'] = weights # 将其命名为 'w'
>>> g
Graph(num_nodes=4, num_edges=4,
      ndata_schemes={},
      edata_schemes={'w' : Scheme(shape=(), dtype=torch.float32)})
```

相关API: `ndata`、`edata`。