

从空域角度看图神经网络

伍元凯

对于图卷积神经网络的卷积层，表达式如下：

$$Z = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} X \Theta), \quad (1)$$

这是一个 1-局部化的图神经网络，即每个节点只接收和它距离为 1(邻居) 的信息，从空间角度来看，每个节点 v 是如何处理信息的呢？

$$h_v^{(k)} = \sigma \left(\Theta \text{MEAN} \{ h_u^{(k)} : \forall u \in N(v) \cup v \} \right) \quad (2)$$

其中 $N(v)$ 是 v 的 1-hop 邻居的集合，假如图的所有边都为 1，这里使用普通的均值处理信息，而如果是带权重的边，则是考虑边的权重的均值。正是这样一种构建的方法，文献 [1] 发现了图神经网络和用于测试图同构的威斯费勒-莱曼测试法 [2] 的关系。

1.1 威斯费勒-莱曼测试法

威斯费勒-莱曼测试的目的在于测试是否两张图 G_1 和 G_2 是同构的，同构指的是存在一个将 G_1 的节点 $1, \dots, n$ 映射到 G_2 的节点 $1, \dots, n$ 的一一对应 σ ，使得 G_1 中任意两个节点 i 和 j 相连接，当且仅当 G_2 中对应的两个节点 $\sigma(i)$ 和 $\sigma(j)$ 相连接。同构可表示为 $G_1 \simeq G_2$ 。下面举两个单词的例子，非常便于理解。

$$puppy \simeq daddy, d = \sigma(p), a = \sigma(u).$$

判断两个图是否同构被认为是一个 NP 难问题(虽然还没有 NP 难的证明),并不是很好解决,即使人类自身对于某些图也要发很长时间去解决,有兴趣可以参考这个网站<http://github.ericmickelsen.com/graf/isomorphism.html>自己做做测试。

威斯费勒-莱曼测试法提供了一种解决判断两个图是否同构的方法,一维的威斯费勒-莱曼测试法只需要两步,第一步:聚合 (aggregate) 节点 v 及其邻居 $N(v)$ 的特征 $h^{(k)}$ 。第二步:将聚合的特征映射成节点 v 的新特征 $h_v^{(k+1)}$ (采用单射的 hash 映射)。重复以上步骤 K 次直到收敛,如果 G_1 和 G_2 的特征集合相同,则判断两者同构。在威斯费勒-莱曼测试刚提出时,其被认为是一种可以在多项式时间内解决图同构判定的方法。但后来被发现该方法会将某些异构图判定为同构图。下面给出一个例子。

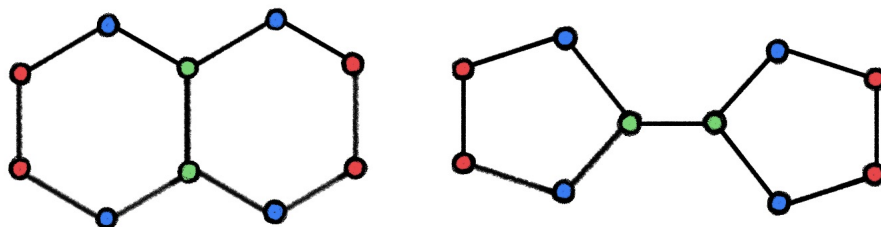


Fig.1 威斯费勒-莱曼测试将异构的图判断为了同构的图.

采用图卷积的方式,我们同样无法区分 G_1 和 G_2 ,代码如下,最终得到的图像的特征 h_1 和 h_2 的值一模一样,给 G_1 和 G_2 上色后的效果如图 1 所示。

```
h1 = np.ones((10, 1))
h1 = h1.copy()
h2 = h1.copy()
```

```
EDGES = [
    ('A', 'B'),
    ('B', 'C'),
```

```

        ('C', 'D'),
        ('E', 'D'),
        ('E', 'F'),
        ('F', 'A'),
        ('E', 'G'),
        ('D', 'H'),
        ('J', 'H'),
        ('G', 'I'),
        ('J', 'I'),
    ]
    G = nx.Graph()
    G.add_edges_from(EDGES)
    P1 = nx.linalg.graphmatrix.adjacency_matrix(G).toarray()
    G1 = nx.from_numpy_matrix(P1)
    nx.draw_networkx(G1, node_color=h1[:, 0], cmap = 'cool')
    plt.savefig('G1.pdf')
    plt.show()

    EDGES = [
        ('A', 'B'),
        ('B', 'C'),
        ('E', 'D'),
        ('E', 'F'),
        ('E', 'C'),
        ('F', 'A'),
        ('D', 'G'),
        ('D', 'H'),
        ('J', 'H'),
        ('G', 'I'),
        ('J', 'I'),
    ]
    G = nx.Graph()
    G.add_edges_from(EDGES)
    P2 = nx.linalg.graphmatrix.adjacency_matrix(G).toarray()
    G2 = nx.from_numpy_matrix(P2)
    nx.draw_networkx(G2, node_color=h1[:, 0], cmap = 'cool')
    plt.savefig('G2.pdf')
    plt.show()

    A_hat1 = P1 + np.eye(10,)
    D_hat1 = np.diag(np.power(np.sum(A_hat1, axis = 1), -0.5))
    MP1 = np.dot(np.dot(D_hat1, A_hat1), D_hat1)
    for i in range(100):
        h1 = np.matmul(MP1, h1)

    A_hat2 = P2 + np.eye(10,)
    D_hat2 = np.diag(np.power(np.sum(A_hat2, axis = 1), -0.5))
    MP2 = np.dot(np.dot(D_hat2, A_hat2), D_hat2)
    for i in range(100):
        h2 = np.matmul(MP2, h2)

    nx.draw_networkx(G1, node_color=h1[:, 0], cmap = 'cool')
    plt.savefig('G1_100.pdf')

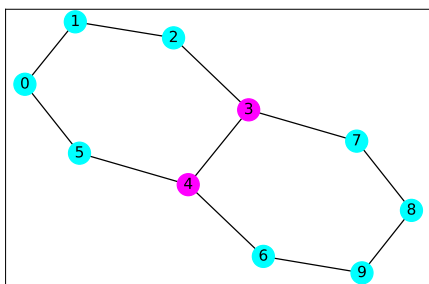
```

```
plt.show()

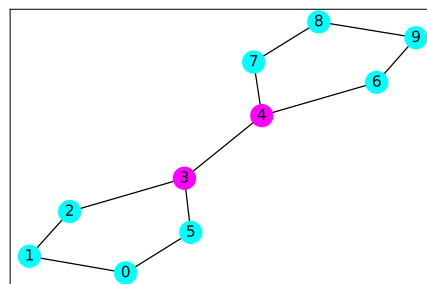
nx.draw_networkx(G2, node_color=h2[:, 0], cmap = 'cool')
plt.savefig('G2_100.pdf')
plt.show()

h1
array([[0.96650635],
       [0.96650635],
       [0.96650635],
       [1.1160254 ],
       [1.1160254 ],
       [0.96650635],
       [0.96650635],
       [0.96650635],
       [0.96650635],
       [0.96650635]])

h2
array([[0.96650635],
       [0.96650635],
       [0.96650635],
       [1.1160254 ],
       [1.1160254 ],
       [0.96650635],
       [0.96650635],
       [0.96650635],
       [0.96650635],
       [0.96650635]])
```



(a) 图卷积操作 $G1$ 最终收敛效果.



(b) 图卷积操作 $G2$ 最终收敛效果.

Fig.2 利用图卷积判断 $G1$ 和 $G2$ 是否同构.

从上面这个例子，我们发现威斯费勒-莱曼测试法做出错误判断的情况，图卷积一样无法做到，论文 [1] 的作者不仅建立了图神

神经网络和威斯费勒-莱曼测试法的关系，还证明了图神经网络最多能够实现威斯费勒-莱曼测试法的效果。

1.2 图神经网络是一种特殊的威斯费勒-莱曼测试

文献 [1] 的作者将图神经网络总结为以下的形式：

$$a_v^k = \text{AGGREGATE}^{(k)} \left(\{h_u^{k-1} : u \in N(v)\} \right), \quad h_v^k = \text{COMBINE}^{(k)}(h_v^{k-1}, a_v^k). \quad (3)$$

前面一部分表示图神经网络中的每个节点如何聚合信息，后面一部分表示聚合的信息如何与上一层图神经网络在该节点获得的特征进行结合。

目前比较经典的图神经网络框架都可以描述为以上的框架，公式中的 (2) 的图卷积显然符合。另外几种框架如 GraphSage[3]:

$$\begin{aligned} a_v^k &= \text{POOL} \left(\left\{ \text{ReLU}(\Phi h_u^{k-1}), \forall u \in N(v) \right\} \right), \\ h_v^k &= W [h_v^{k-1}, a_v^k] \end{aligned} \quad (4)$$

其中 POOL 可以是图卷积中使用的均值，也可以是最大值，另外也可以使用 LSTM 来聚合信息。

针对空间域的图神经网络经常只是把不同邻居之间的权重都一视同仁设为 1 的问题，图注意力网络 [4] 利用注意力机制根据每

个节点的特征来确定各个边的权重，具体方式如下：

$$\begin{aligned}
e(h_v^{k-1}, h_u^{k-1}) &= \text{LeakyReLU} \left(a^T [W h_v^{k-1} \| W h_u^{k-1}] \right), \\
\alpha_{vu} &= \text{softmax}_u (e(h_v, h_u)), \\
h_v^k &= \sigma \left(\sum_{\forall u \in N(v)} \alpha_{vu} \cdot W h_u^{k-1} \right)
\end{aligned} \tag{5}$$

实际上在 GAT 在聚合信息是根据相邻节点特征的加权和而已，仍然属于聚合-综合这一框架之内。

更抽象的总结性的消息传导网络 [5]：

$$\begin{aligned}
a_v^k &= \sum_{\forall u \in N(v)} M^k(h_v^{k-1}, h_u^{k-1}, e_{uv}), \\
h_v^k &= U^k(a_v^k, h_v^{k-1}).
\end{aligned} \tag{6}$$

其中 M^k 和 U^k 是第 k 层的某种神经网络。

文献 [1] 证明了图神经网络架构下的方法在解决图同构问题时，最多只能达到威斯费勒-莱曼测试的水平。要证明这一点，作者首先做了一个假设，图上的节点的特征只能来自于一个可数集，这里注意我们并没有必要区分连续和离散的特征，因为在计算机中即使是连续的实数也有一个限度，所以无论如何我们的操作都在可数集上完成。作者首先证明了，如果图神经网络可以判断两个图 G_1 和 G_2 异构，那么威斯费勒-莱曼测试 (WL 测试) 同样可以判断 G_1 和 G_2 异构。

这是一个显而易见的结论，在 WL 测试中，只要节点的特征集合不同，WL 测试就会给这些结合不同的特征。那么在一个图中，

如果 WL 测试的特征 $l_u = l_v$ ，等价于 u 和 v 其邻居聚合的特征在上一步也完全相等:

$$l_v^{k+1} = l_u^{k+1}, \left(l_v^k, \{l_w^k : w \in N(v)\} \right) = \left(l_u^k, \{l_w^k : w \in N(u)\} \right), \quad (7)$$

由于 WL 测试和图神经网络在第 0 次聚合时 G_1 和 G_2 上处理的信息是一模一样的，而在任何一次聚合过程中，只要聚合的节点特征有所不同，WL 测试就会给节点 v 和 u 不一样的特征 l_u 和 l_v ，那么如果 $l_u = l_v$ ，必然有图神经网络的输出

$$\left(h_v^k, \{h_w^k : w \in N(v)\} \right) = \left(h_u^k, \{h_w^k : w \in N(u)\} \right), \quad (8)$$

因为只要 $l_u = l_v$ 就有 $h_u = h_v$ ，对于 $w \in G_1 \text{ or } G_2$ ，我们能找到一个固定的映射为 $h_w = \phi(l_w)$ 。那么对于 WL 测试判断同构的 G_1 和 G_2 中所有节点，我们总有

$$\left(h_v^k, \{h_w^k : w \in N(v)\} \right) = \left(\phi(l)_u^k, \{\phi(l)_w^k : w \in N(u)\} \right), \quad (9)$$

那么显然如果图神经网络的判断为异构的两个图，WL 测试也会判断其为异构。

文献 [1] 同样给出了当且仅当图神经网络的聚合方法在处理邻居的特征集时是单射的 (必须将不同特征集合映射到不同的特征) 才能百分百将 WL 测试判断异构的图也同样判断为异构 (达到 WL 测试的水平)。这一点也是显而易见的，因为 WL 测试的聚合 hash 函数是一个事先定义好的单射函数。那么如果 u 节点和 v 节点邻居聚合的特征在上一步是不一致的话， l_u^k 和 l_v^k 也会不一致，但如果图神经网络的聚合方法是非单射的，其有可能将 $\phi(l)_u^{k-1}$ 和 $\phi(l)_v^{k-1}$ 变为相同的特征 h_u^k 和 h_v^k ，假如 WL 测试正是依靠 G_1 中的 u 和 G_2 中的 v 判断出 G_1 和 G_2 异构，此时 WL 测试判断异构的图图神经网络

可能判断为同构。

这一研究给分析图神经网络的聚合函数打开了一扇门，我们可以通过研究聚合函数 AGGREGATION 判断图神经网络的表达能力（能否达到 WL 测试的水平）。下图给出了不同非单射聚合函数失效的情况。

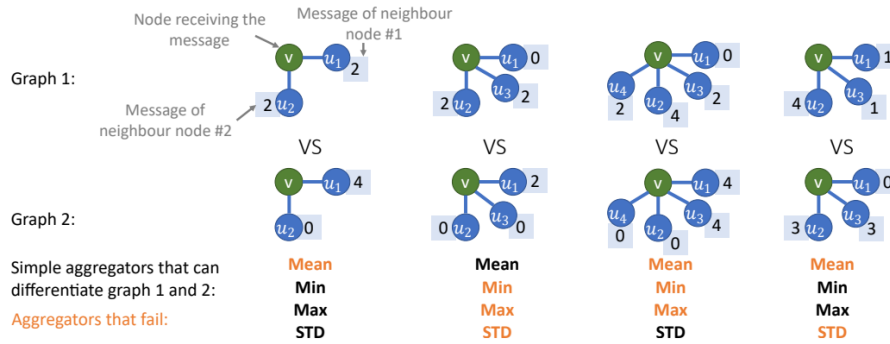


Fig.3 非单射函数（均值，最大值，最小值以及标准差）失效的情况 [6].

1.3 如何提升图神经网络？

论文 [1] 的工作可以说理论上给分析图神经网络的表达能力提供了一种方法，也从理论上证明提升图神经网络性能的方法。那就是聚合信息时的单射函数。譬如说论文 [1] 并没有对图注意力网络的表达能力进行分析，只是阐明了最大值和均值聚合失效的情况，论文 [7] 找到了图注意力网络失效的一个情况，即当 $G1$ 和 $G2$ 中都有一个子图，这个图都有一个中心点，但中心点连接的节点数量不同的时候，图注意力网络的聚合函数不是一个单射函数。假设 WL 测试开始第 0 步时所有节点的特征都为相同的数，那么此时图注意力网络会把 $G1$ 和 $G2$ 中的连接节点数量不同的中心节点都映射为

相同的特征:

```
h1 = np.ones((4, 1)) * np.random.random(1)
h2 = np.zeros((5, 1))
h2[0:4, :] = h1
h2[4, :] = h2[3, :]

EDGES1 = [
    ('A', 'B'), ('A', 'C'), ('A', 'D')
]

EDGES2 = [
    ('A', 'B'), ('A', 'C'), ('A', 'D'), ('A', 'E')
]

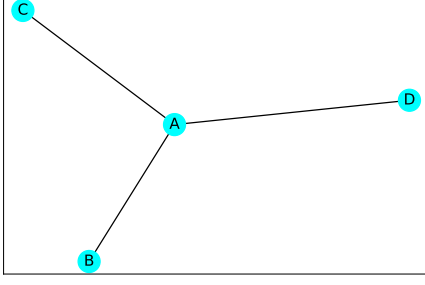
G1 = nx.Graph()
G1.add_edges_from(EDGES1)
nx.draw_networkx(G1, node_color=h1[:, 0], cmap = 'cool')
plt.savefig('G1_before_after.pdf')
plt.show()

G2 = nx.Graph()
G2.add_edges_from(EDGES2)
nx.draw_networkx(G2, node_color=h2[:, 0], cmap = 'cool')
plt.savefig('G2_before_after.pdf')
plt.show()

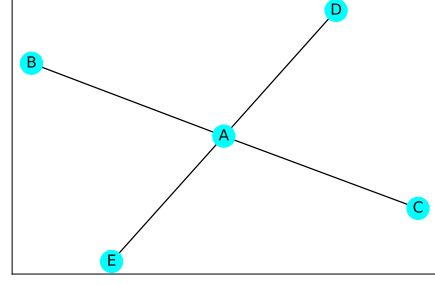
att1 = np.matmul(h1, h1.T)
att1 = sp.special.softmax(att1, axis = 1)
p1 = 0
for i in range(4):
    p1 += h1[i, 0] * att1[0, i]
0.819318244075470

att2 = np.matmul(h2, h2.T)
att2 = sp.special.softmax(att2, axis = 1)
p2 = np.dot(h2.T, np.expand_dims(att2[0, :], -1))
p2 = 0
for i in range(5):
    p2 += h2[i, 0] * att2[0, i]
0.819318244075470
```

那么基于这一判断,就可以优化图注意力网络的聚合函数,规



(a) 图注意力网络失效的情况，有 3 个邻居的中心节点。



(b) 图注意力网络失效，有 4 个邻居的中心节点。

Fig.4 图注意力网络失效的情况。

避这一失效的情况，作者提出了两种简单的解决方案：

$$\begin{aligned}
 e(h_v^{k-1}, h_u^{k-1}) &= \text{LeakyReLU} \left(a^T [W h_v^{k-1} \| W h_u^{k-1}] \right), \\
 \alpha_{vu} &= \text{softmax}_u (e(h_v, h_u)), \\
 h_v^k &= \sigma \left(\sum_{\forall u \in N(v)} (\alpha_{vu} + 1) \cdot W h_u^{k-1} \right)
 \end{aligned} \tag{10}$$

或者

$$h_v^k = \sigma \left(|N(v)| \cdot \sum_{\forall u \in N(v)} \alpha_{vu} \cdot W h_u^{k-1} \right) \tag{11}$$

两者都考虑了每个节点连接的节点的数量，自然规避了图 4 中出现的失效情况。

1.4 参考文献

[1] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018, September). How Powerful are Graph Neural Networks?. In International Conference on Learning

Representations.

- [2] Weisfeiler, B., & Leman, A. (1968). The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9), 12-16.
- [3] Hamilton, W. L., Ying, R., & Leskovec, J. (2017, December). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 1025-1035).
- [4] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- [5] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017, July). Neural message passing for quantum chemistry. In *International conference on machine learning* (pp. 1263-1272). PMLR.
- [6] Corso, G., Cavalleri, L., Beaini, D., Liò, P., & Veličković, P. (2020). Principal Neighbourhood Aggregation for Graph Nets. *Advances in Neural Information Processing Systems*, 33.
- [7] Zhang, S., & Xie, L. (2020, July). Improving attention mechanism in graph neural networks via cardinality preservation. In *IJCAI: proceedings of the conference* (Vol. 2020, p. 1395). NIH Public Access.