

图拉普拉斯的特征向量和图神经网络

伍元凯

近日读到 ICML2021 会议的一篇文章 [1] 非常有意思。对于图神经网络的现有研究的不足, 文章认为“there are currently no methods that allow asymmetric graph kernels that are dependent on the full graph structure or directional flows. They either depend on local structures or local features. This is in opposition to images, which exhibit canonical directions: the horizontal and vertical axes.”。这一句话的大体意思是, 对于二维的图像, 我们有一个全局性的方向性, 即水平和竖直两个方向。卷积神经网络正是沿着水平和竖直两个方向进行卷积。而图神经网络的信息聚合模式则只是基于局部的相邻关系, 没有考虑全局的方向性。那对于图我们有没有可能找到其全局的方向性呢? 作者在论文中给出了解答。

给定一个 n 个节点的图 G , 其图拉普拉斯矩阵 $L_{n \times n}$ 为

$$L = D - A \quad (1)$$

其中 D 是度数矩阵, A 是邻接矩阵。作者认为对应于 L 小的特征值的特征向量 (即低频特征向量) 可有效地体现图的整体方向, 在 IGNNK[2] 工作基础上, 我对 USHCN (美国气候监测站数据) 进行了分析, 发现最小的奇异值对应的特征向量确实可以做到这一点。代码如下:

Fig.1 Insert code directly in your document.

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import pylab
import joblib

G = nx.Graph()
latlon = np.loadtxt("data/ushcn/latlon.csv", delimiter=",")

for i in range(latlon.shape[0]):
    G.add_node(str(i), pos = (latlon[i][1], latlon[i][0]))

dist_mx = joblib.load('data/ushcn/sim.joblib')

std = dist_mx[dist_mx<np.inf].std()
A = np.exp(-np.square(dist_mx / std))
A[A<0.1] = 0

D = np.diag(np.sum(A, axis=-1))
L = D - A

w, v = np.linalg.eig(L)

color = v[:, 1]
cmap=plt.cm.cool
vmin = min(color)
vmax = max(color)
direction = np.argsort(color)
for i in range(latlon.shape[0]-1):
    G.add_edges_from([(str(direction[i]), str(direction[i+1])))])

nx.draw(G, nx.get_node_attributes(G, 'pos'), node_color = color, with_labels=False, node_size=10, cmap=cmap)
sm = plt.cm.ScalarMappable(cmap=cmap, norm=plt.Normalize(vmin = vmin, vmax=vmax))
plt.colorbar(sm)
plt.show()
```

画出的图的结果如下所示：在图 1 中我们可以明显发现根据每个节点对应的特征向量值从小到大连接的图，其方向整体从美国的西北方向流向美国的东南方向。

基于这一有趣的现象，在这篇文章中本人想对图拉普拉斯的特征向量的一些性质及其和图神经网络的发展的关系做一些论述。

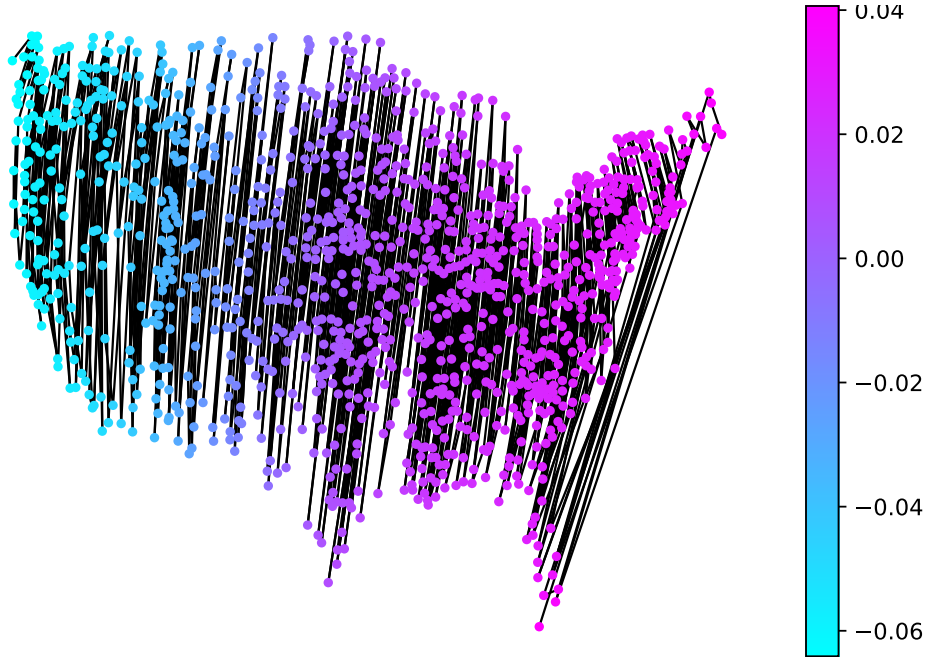


Fig.1 USHCN 数据集倒数第二小的特征向量对应的方向.

1.1 最小非零特征值对应的特征向量的物理性质的一种解释

我们可以用 $W = D^{-1}A$ 作为转移矩阵，在一个时间步长内从 x 到 y 移动的概率为 $W(x, y)$ ，而在 k 步骤内从 x 移动到 y 的概率为 $W^k(x, y)$ 。假设时间 t 服从均值为 1 的 Poisson 分布，那么在时间 t 从 x 转移到 y 的概率 $q_t(x, y)$ 等于

$$q_t(x, y) = \sum_{k=0}^{\infty} \frac{e^{-t} t^k}{k!} p_k(x, y), \quad (2)$$

上面公式的右侧实际是均值为 1 的泊松分布和时刻为 k 时的状态转移概率的乘积，我们进而有 $q_t = e^{-t(W-I)} = e^{-tL_{norm}}$ ， q_t 被称为 continuous time heat kernel，因为其服从热量在图上的转移偏微分方程 $\frac{\partial q}{\partial t} + \Delta q = 0$ 。

利用 q_t 可定义出节点 x 和 y 之间的扩散距离

$$d_t(x, y) = \left(\sum_{z \in V} (q_t(x, z) - q_t(y, z))^2 \right)^{\frac{1}{2}} \quad (3)$$

x 和 y 的扩散距离越小，说明两者之间的相互作用越强。

沿着特征向量 ϕ ，在该特征向量上有 $\phi(x) < \phi(y)$ ，沿着最小的特征向量 ϕ_1 从小到大移动一步从 x 移动到 x' ，我们有 $d_t(x', y) < d_t(x, y)$ ，且 $d_t(x, y) - d_t(x', y) < C \leq t$ 。 C 和 $e^{\frac{1}{\lambda_1}}$ 成正比， λ_1 是最小的特征值。该性质的证明可见于论文 [1] 中的附录 C.3。证明中用到了特征值都正交的性质，有

$$d_t(x, y) = \left(\sum_{i=1}^{n-1} e^{-2t\lambda_i} (\phi_i(x) - \phi_i(y))^2 \right). \quad (4)$$

这一性质说明，沿着最小的特征值移动，节点 x 可以一步步地移动到很远的节点 y (随机的移动会一步一步移动到离 y 扩散距离更近的节点)。这一定理似乎可以让我们直接运用普通的卷积神经网络处理图数据（对节点信号根据最小特征向量的大小重新排列），虽然论文 [1] 的方法还是基于图神经网络的信息传导的方式来进行操作，但似乎这一研究的目的就是找到图的“水平”和“垂直”方向，这样就可以利用普通卷积神经网络来处理图数据了。此外，论文的作者似乎只对基于空间关系聚合信息的图神经网络比较熟悉，但对于谱神经网络并未进行论述，其实很多基于图拉普拉斯的 Graph Convolutional Networks 早已利用了这一方法来处理图数据。

1.2 图拉普拉斯的特征向量和图卷积的起源

实际上，图拉普拉斯矩阵的特征值的性质早已被进行了深入的研究，在图卷积神经网络的起源过程中起了很大的作用。实际上最早的图卷积就是受到了图拉普拉斯特征值和特征向量的启发，只是目前的图神经网络研究偏向于空间上的图神经网络，工作 [1] 就是研究空间的图神经网络的学者对图卷积神经网络的再发现罢了。

在早年的流形学习中很关注一个问题，就是把信号映射到一个新的空间之上，能否保障其局部的关系。假设某个空间的邻接矩阵为 A ， A 实际保存的是一种空间的关系，即 $A(i, j)$ 越大，证明节点 i 和 j 距离越近。给定一个向量 $z \in \mathbb{R}^m$ ，可以给出如下的距离来衡量其在 A 定义的空间之上的平滑程度 (和该空间的相符水平)

$$\|\nabla z\|_A^2 = \sum_i \sum_j A_{ij} |z(i) - z(j)|^2, \quad (5)$$

已知图拉普拉斯为 $L = D - A$ ，有 $\frac{1}{2} \|\nabla z\|_A^2 = z^T L z$ 。那么对于使得 $\|\nabla z\|_A^2$ 最小化的 z 满足

$$v_0 = \arg \min_{z \in \mathbb{R}, \|z\|=1} z^T L z = (1/\sqrt{m}) \mathbb{1}_m, \quad (6)$$

$\|z\| = 1$ 是加入的限制，如果我们继续加入 (v_1, v_2, \dots, v_n) 必须都正交，

$$v_i = \arg \min_{z \in \mathbb{R}, \|z\|=1, z \perp (v_0, v_1, \dots, v_{i-1})} z^T L z, \quad (7)$$

对于 L 的特征值和特征向量有 $L v_i = \lambda_i v_i$ ，显然 L 的从小到大的特征值 $(\lambda_0, \lambda_1, \dots, \lambda_i)$ 对应的特征向量是 A 定义的图上平滑程度从高到低的一组基。

那么图拉普拉斯的特征值和特征向量的这些性质是如何被应用到图卷积中呢？图像中的卷积操作，实质上是在输入信号上滑动不同的卷积核并执行乘积。然而对于非欧式空间的图形结构，节点和多个节点可能都是连通的，我们很难在空间上定义出一种普适的卷积核 (没有对应于图像的水平 and 竖直方向，也是论文 [1] 希望解决的问题)，而卷积定律告诉我们，实域上的卷积对应的是频域上的乘积，即

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{g\} \quad (8)$$

\mathcal{F} 指的是信号的傅里叶变换。

由于图拉普拉斯特征向量构成的矩阵 $V = (v_o, v_1, \dots, v_m)$ 的向量互相正交，且有 $V^T V z = z$ ，我们可以利用 V 定义图信号的傅里叶变换 [4] $\mathcal{F}(x)$ 为 Vx ，傅里叶逆变换 \mathcal{F}^{-1} 为 $V^T x$ ，图卷积可被定义为 $x *_G g = V^T (Vx \odot Vg)$ 。对于 $(Vx \odot Vg)$ 有

$$Vx = \begin{pmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \vdots \\ \bar{x}_{m-1} \end{pmatrix}, Vg = \begin{pmatrix} \bar{g}_0 \\ \bar{g}_1 \\ \vdots \\ \bar{g}_{m-1} \end{pmatrix}. \quad (9)$$

进而有

$$Vx \odot Vg = \begin{pmatrix} \bar{x}_0 \bar{g}_0 \\ \bar{x}_1 \bar{g}_1 \\ \vdots \\ \bar{x}_{m-1} \bar{g}_{m-1} \end{pmatrix} = \begin{pmatrix} \bar{g}_0 & & & \\ & \bar{g}_1 & & \\ & & \ddots & \\ & & & \bar{g}_{m-1} \end{pmatrix} \begin{pmatrix} \bar{x}_0 \\ \bar{x}_1 \\ \vdots \\ \bar{x}_{m-1} \end{pmatrix} \quad (10)$$

如前所述，我们并不关心原始信号 g 在实域上的特性，转而直接在频域上操作，即神经网络的学习参数为 $w_g = (\bar{g}_0, \bar{g}_1, \dots, \bar{g}_{m-1})$ 。因

此图卷积也可定义为

$$x *_G g = V^T (\text{diag}(w_g) V x) \quad (11)$$

w_g 是图卷积网络的学习参数。

1.3 图卷积网络的进一步简化

公式 (11) 中定义的图卷积操作通过图拉普拉斯的特征值把图信号变到了频域，卷积完全在频域上进行。但是，这一操作需要计算图拉普拉斯的特征向量，对于有上万个节点的图，计算 V 的复杂度会增加。论文 [5] 参考小波变换工作 [6] 中的切比雪夫多项式逼近方法，进一步对图卷积做了简化。对于图拉普拉斯 L 及其特征向量组成的矩阵，我们有

$$L = V^T \Lambda V = V^T \begin{pmatrix} \lambda_0 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_{m-1} \end{pmatrix} V, \quad (12)$$

其中 $(\lambda_0, \lambda_1, \dots, \lambda_{m-1})$ 是 L 的特征值，图拉信号的卷积操作等价于

$$x *_G g = V^T g(\Lambda) V x = V^T \begin{pmatrix} g(\lambda_0) & & & \\ & g(\lambda_1) & & \\ & & \ddots & \\ & & & g(\lambda_{m-1}) \end{pmatrix} V, \quad (13)$$

这里我们把 g 想象成了一个函数，有 $g(\lambda_i) = \bar{g}_i$ for $i = 1, 2, \dots, m$ 。任何的函数都可以使用多项式进行逼近（如泰勒展开），我们运用

一个多项式来逼近 $g(\lambda_i)$, 有

$$g(\lambda) \approx \sum_{k=1}^K \theta_k \lambda^k \quad (14)$$

θ_k 是第 k 个多项式的参数, 对于图拉普拉斯矩阵, 又有

$$L^k = V^T \Lambda^k V = V^T \begin{pmatrix} \lambda_0^k & & & \\ & \lambda_1^k & & \\ & & \ddots & \\ & & & \lambda_{m-1}^k \end{pmatrix} V, \quad (15)$$

进一步有 $V^T g(\lambda) V = \sum_{k=1}^K \theta_k L^k$ 。通过这样一种变换, 我们在进行图卷积的过程之中不再需要计算图拉普拉斯矩阵的特征向量, 可以直接在图拉普拉斯矩阵 L 上操作, 图卷积变成了图拉普拉斯和向量 θ 的乘积操作。[5] 和 [6] 的作者没有直接计算 L, L^2, \dots, L^k (有些工作是直接使用它们的), 而是采用切比雪夫多项式以递归的方式计算多项式, 对于切比雪夫序列有

$$T_0(x) = 1, T_1 = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), \quad (16)$$

在切比雪夫多项式序列中, 如果 $x \in [-1, 1]$, $T(x) \in [-1, 1]$, 这是使用切比雪夫多项式的原因之一。继而有

$$g(\lambda) \approx \sum_{k=1}^K \theta_k T_k(\lambda^k) \quad (17)$$

图卷积操作变为

$$x *_G g \approx \sum_{k=1}^K \theta_k T_k(\lambda^k) x. \quad (18)$$

这便是 Chebynet 的图卷积操作。理论上 K 需要趋向于无穷大才可能逼近 $g(\lambda)$ ，我们已知在图拉普拉斯 L 中只有当节点 i 和 j 连通时, $L(i, j)$ 才不为 0, 而图信号往往又都是局部相连的。如果我们选择 $K = 3$ ，那么 x 中的节点 i 最多受到空间上和它距离为 3 的节点的影响 (邻居的邻居的邻居的邻居 3-hop)。因此 Chebynet 也被视为 **K -局部化的网络**， K -局部化是一个空间的概念，但却可以通过对频域上的卷积操作简化得到。

在一个每一层的卷积核长度为 K 欧式空间卷积层中，每一个位置接收的是和它距离为 K 的邻居的信号，因此其也可被视为一个 K -局部化的网络，而卷积神经网络则是多层 K -局部化网络组合的产物，一个深层的卷积神经网络可以通过逐层的局部操作获得全局的信息。同样的，对于卷积神经网络，如果将每一层的接受域设为 1，并将其叠加为一个 K 层的深度网络，那么也会和一个单层的 K -局部化网络相似的效果。这也是文献 [7] 提出的图卷积网络的工作, 使用 $K = 1$ 的切比雪夫多项式逼近图卷积的形式，得到 $K = 1$ 的图卷积的形式

$$x *_G g \approx \theta_0 x + \theta_1 (L - I_m) x = \theta_0 x - \theta_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x, \quad (19)$$

进一步减少参数的数量，设定 $\theta = \theta_0 = -\theta_1$ ，得到

$$x *_G g \approx \theta (I_m + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x, \quad (20)$$

我们的目标是进行多层的图卷积操作，并采用梯度下降对其求解，假设 $P = I_m + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ ，并考虑一个深度的神经网络 $y = \sigma(P(\sigma(P \cdots \sigma(Px))))$ 。对这个网络求梯度有，假设 σ 是线性函数

$$\frac{\partial y}{\partial P} \propto P^{K-1} x, \quad (21)$$

对于 P^{K-1} 有 $P^{K-1} = V_P^T \Lambda_P^{K-1} V_P$, Λ_P 和 V_P 是 P 的特征值和特征向量, 如果 λ_P 中存在大于 1 的数, 显然梯度会逐层指数性增长, 形成梯度爆炸, 那么有必要把 P 的特征值限定在 $[0, 1]$, 且 P 还需要保持 $I_m + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ 的性质。文献 [7] 的操作如下:

$$\begin{aligned} I_m + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} &\rightarrow \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} \\ \hat{A} &= A + I_m, \\ \hat{D}(i, i) &= \sum_j \hat{A}(i, j) \end{aligned} \tag{22}$$

显然 $\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}$ 的特征值全部在 $[0, 1]$ 之间。这样图卷积层的操作被定义为

$$Z = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} X \Theta), \tag{23}$$

X 是输入信号, Θ 是卷积核的参数。

1.4 从空域的消息传递理解图卷积网络

[7] 提出的图卷积层已经转化为一个 1-局部化的网络, 在这个网络之中, 图只接收相邻节点的消息, 因此图卷积网络即可以视为一种空间域的卷积网络, 也可以视为一种频域的网络。使用以下的代码可以对图卷积网络的信息传导可视化分析:

Fig.2 Insert code directly in your document.

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

hl = np.random.randint(10, size = (8, 1))
A = np.array(random_adjacency_matrix(8))
array([[0, 0, 0, 0, 1, 1, 0, 1],
       [0, 0, 0, 0, 1, 1, 1, 0],
       [0, 0, 0, 0, 1, 1, 0, 0],
```

```

[0, 0, 0, 0, 0, 0, 1, 1],
[1, 1, 1, 0, 0, 1, 1, 1],
[1, 1, 1, 0, 1, 0, 1, 0],
[0, 1, 0, 1, 1, 1, 0, 0],
[1, 0, 0, 1, 1, 0, 0, 0]])

A_hat = A + np.eye(8,)
D_hat = np.diag(np.power(np.sum(A_hat, axis = 1), -0.5))
MP = np.dot(np.dot(D_hat, A_hat), D_hat)

G = nx.from_numpy_matrix(A)
labels = {}
for i in range(hl.shape[0]):
    labels[i] = hl[i, 0]
pos = nx.spring_layout(G)
nx.draw_networkx_nodes(G, pos, node_color=hl[:, 0], cmap = 'cool')
nx.draw_networkx_edges(G, pos, arrows=False)
nx.draw_networkx_labels(G, pos, labels)
hl1 = np.dot(MP, hl)
labels = {}
for i in range(hl.shape[0]):
    labels[i] = np.around(hl1[i, 0], decimals=1)
edge_labels = dict([(n1, n2), d['weight']]
                    for n1, n2, d in G.edges(data=True))
for key in edge_labels.keys():
    edge_labels[key] = np.around(MP[key[0], key[1]], decimals=2)
nx.draw_networkx_edge_labels(G, pos, edge_labels)
plt.savefig('before.pdf')
plt.show()

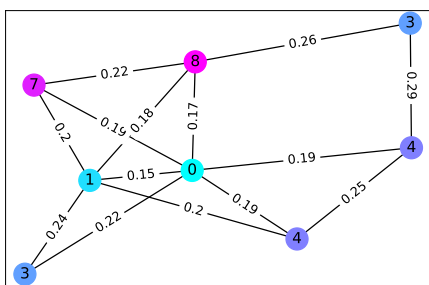
nx.draw_networkx_nodes(G, pos, node_color=hl1[:, 0], cmap = 'cool')
nx.draw_networkx_edges(G, pos, arrows=False)
nx.draw_networkx_labels(G, pos, labels)
plt.savefig('after.pdf')
plt.show()

D_hat_ = (np.power(np.sum(A_hat, axis = 1), 0.5))
array([2.          , 2.          , 1.73205081, 1.73205081, 2.64575131,
        2.44948974, 2.23606798, 2.          ])

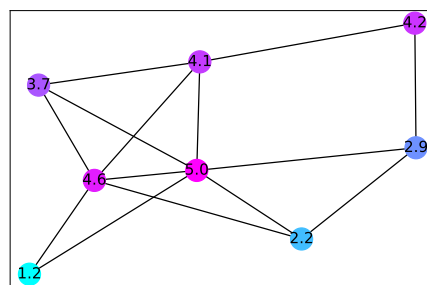
```

可视化的结果如下所示，我们可以发现在经过一次聚合之后，相连节点上的信息变得更相近了，我们发现在聚合 10 次之后，再进行信息传送，节点的数值已经不再发生变化了。这也引出了一个问题，现有的图卷积网络都面临着过平滑的问题，这也被认为是限制构建深层图网络的问题之一。另外在信息传导 100 次（100 层

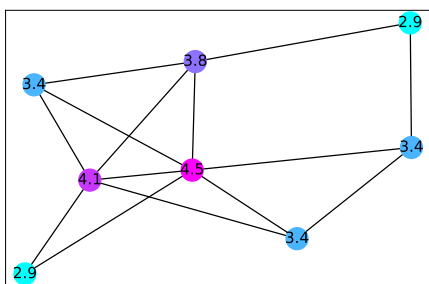
GCN) 以后, 我们发现有两个邻居的节点的数值都变为 2.9, 有三个的节点的都变为 3.4, 而其中左上和右下的节点离得很远, 但他们由于邻居相同, 最终数值也相同, 这显然不是我们想要的效果 (距离上比较远的节点不应该有相同的特征), 下一节的一篇论文对这种缺陷进行了数学上的分析。



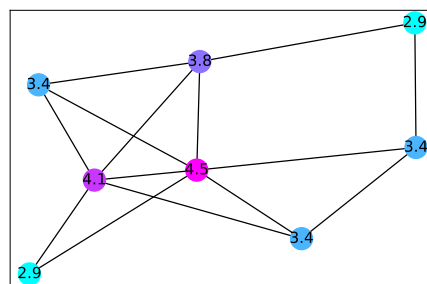
(a) 图卷积消息传送前的效果.



(b) 图卷积消息传送后的效果.



(c) 图卷积消息传送 10 层后的效果.



(d) 图卷积消息传送 100 层后的效果.

Fig.2 图卷积中的信息传送, 节点中的信息表示此时节点的数值, 边上的信息表示 $\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}$ 的权重.

1.5 图卷积网络的缺陷

文献 [8] 利用拉普拉斯平滑器的特点对图卷积网络进行了分析, 拉普拉斯平滑利用图拉普拉斯对图信号 x 进行如下操作:

$$\begin{aligned} (I - \gamma L_{rw})x, \quad L_{rw} &= \hat{D}^{-1} \hat{L}, \\ (I - \gamma L_{sym})x, \quad L_{sym} &= \hat{D}^{-\frac{1}{2}} \hat{L} \hat{D}^{-\frac{1}{2}}, \end{aligned} \quad (24)$$

其中 $\gamma \in (0, 1]$, \hat{L} 是加了自连接的 A 的图拉普拉斯。显然, 去掉激活函数 σ 和线性变换 Θ 的图卷积 $\sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} X \Theta)$ 等价于 $\gamma = 1$ 的对称拉普拉斯 (L_{sym}) 平滑。

同理将一个无限层的图神经网络作用在信号 x 上可简化为 $\lim_{m \rightarrow \infty} (I - \gamma L_{sym})^m x$, 已知 L_{sym} 的特征值属于 $[0, 2)$, 显然 $(I - \gamma L_{sym})$ 的特征值属于 $(-1, 1]$, 在进行无限次矩阵相乘后, 特征值绝对值小于 1 对应的特征向量的作用都会趋于 0, 那么我们只需要考虑特征值为 1 的特征向量的作用 (即 L_{sym} 的特征值为 0 的特征向量的作用。)

关于 L_{sym} 的特征值, 可以利用瑞利商来进行证明, 瑞利商由下式给出

$$R(L_{sym}, x) = \frac{x^T M x}{x^T x}. \quad (25)$$

将瑞利商变形为

$$R(L_{sym}, x) = \frac{x^T M x}{x^T x} = \frac{x^T v^T \text{diag}(\lambda) v x}{x^T v^T v x^T}. \quad (26)$$

假设 $y = v x$, 继而有

$$R(L_{sym}, x) = \frac{\sum_{i=1}^{m-1} \lambda_i y_i^2}{\sum_{i=1}^{m-1} y_i^2}. \quad (27)$$

显然 $R[M, x] \in [\lambda_{min}, \lambda_{max}]$ 。考察 L_{sym} ，其元素满足如下条件

$$L_{sym}(i, j) = \begin{cases} 1, & \text{if } i = j \\ -\frac{1}{\sqrt{d_i d_j}}, & \text{if } i \text{ and } j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases} \quad (28)$$

利用这一性质，可以得到 L_{sym} 的瑞利商为

$$R(L_{sym}, x) = \frac{\sum_{i,j} (y(i) - y(j))^2}{\sum_j y(j)^2 d_j}, \quad (29)$$

只要确定 $R(L_{sym}, x)$ 的最大值和最小值就可以确认 L_{sym} 的最大和最小值了。最小值显然为 0，即当 y 的所有元素相等的时候，最大值可以通过 $(y(i) - y(j))^2 \leq 2(y(i)^2 + y(j)^2)$ 得到，显然 $\lambda_{max} = 2$ 。此时，所有连通的 i 节点和 j 节点上，向量的元素 $y(i) = -y(j)$ 。得到 L_{sym} 的最大和最小特征值后，我们接下来考察图拉普拉斯 $L = D - A$ 特征值 0 对应的特征向量的性质，有

$$0 = v_0^T L v_0 = \frac{1}{2} \sum_{i,j} A(i, j) (v_0(i) - v_0(j))^2, \quad (30)$$

显然 $v_0(i) = v_0(j)$ 。因此， L_{rw} 的 0 特征值对应的特征向量为全为 1 的向量 $\mathbb{1}$ ($L v_0 = \lambda_0 v_0$ 两边乘以 D^{-1})。而对于 L_{sym} ，有

$$D^{-\frac{1}{2}} L D^{-\frac{1}{2}} D^{\frac{1}{2}} v_0 = D^{-\frac{1}{2}} \lambda_0 D^{-\frac{1}{2}} D^{\frac{1}{2}} v_0, \quad (31)$$

显然 L_{sym} 的 0 特征值对应的特征向量为 $D^{\frac{1}{2}} \mathbb{1}$ 。对于 $\lim_{k \rightarrow \infty} (I - \gamma L_{sym})^k x$ ， L_{sym} 为 0 的特征值对应的特征向量也为 $D^{\frac{1}{2}} \mathbb{1}$ ，由于其它特征向量对应的特征值都为 0，最后的形式如下：

$$\lim_{k \rightarrow \infty} (I - \gamma L_{sym})^k x \propto D^{\frac{1}{2}}, \quad (32)$$

注意论文 [8] 中有笔误，定理 1 将 $D^{\frac{1}{2}}$ 误写为 $D^{-\frac{1}{2}}$ 。参考图 2 (c) 和图 2(d)，每个节点的数值最后确实收敛到了 $D^{\frac{1}{2}}$ 。

要证明 $\lim_{m \rightarrow \infty} (I - \gamma L_{sym})^m x \propto D^{\frac{1}{2}}$ ，可采用如下的方式，先将 x 写成 v_0, v_1, \dots, v_{m-1} 这些基的加权和的形式：

$$I - \gamma L_{sym} = \hat{A}_\gamma = V \begin{pmatrix} \lambda_0 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_{m-1} \end{pmatrix} V^T, \quad x = Vc. \quad (33)$$

其中向量 c 储存了 c 表示 v_0, v_1, \dots, v_{m-1} 上的分量大小

$$\hat{A}_\gamma^k x = V \begin{pmatrix} \lambda_0^k & & & \\ & \lambda_1^k & & \\ & & \ddots & \\ & & & \lambda_{m-1}^k \end{pmatrix} c. \quad (34)$$

k 趋向于无穷大时，除了 $\lambda_i = 1$ 的值还为 1，其余 λ^k 全部趋向于 0，因此 $\lim_{k \rightarrow \infty} (I - \gamma L_{sym})^k x = c_i \cdot D^{\frac{1}{2}}$ 。

这说明无限层的拉普拉斯平滑器的输出中各节点之间的关系已经和原始输入 x 无关，对于节点的分类和回归等问题（如本人做的 kriging 问题），这显然不符合要求。这一个工作简单的把图卷积网络简化为了拉普拉斯平滑器。图卷积神经网络还有激活函数和线性变换的元素，更为复杂的数学分析目前也有很多研究在做。

1.6 未来计划

从空域看图神经网络。

交通中的图神经网络

1.7 参考文献

- [1] Beaini, D., Passaro, S., Létourneau, V., Hamilton, W. L., Corso, G., & Liò, P. (2020). Directional graph networks. arXiv preprint arXiv:2010.02863.
- [2] Wu, Y., Zhuang, D., Labbe, A., & Sun, L. (2021, May). Inductive Graph Neural Networks for Spatiotemporal Kriging. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 5, pp. 4478-4485).
- [3] Bruna, Joan, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. "Spectral networks and deep locally connected networks on graphs." In 2nd International Conference on Learning Representations, ICLR 2014. 2014.
- [4] Henaff, M., Bruna, J., & LeCun, Y. (2015). Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163.
- [5] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems, 29, 3844-3852.
- [6] Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis,

30(2), 129-150.

[7] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[8] Li, Q., Han, Z., & Wu, X. M. (2018, April). Deeper insights into graph convolutional networks for semi-supervised learning. In Thirty-Second AAAI conference on artificial intelligence.