

# 从GCN到Geom-GCN——潜在空间与双级聚合

## GCN回顾

图上的卷积操作有两种类型,

1. 直接对原图数据, 进行卷积操作——>空域图神经网络
2. 先对原图数据进行傅里叶变换(编码), 在进行卷积操作, 之后傅里叶逆变换(解码)

## 图神经网络

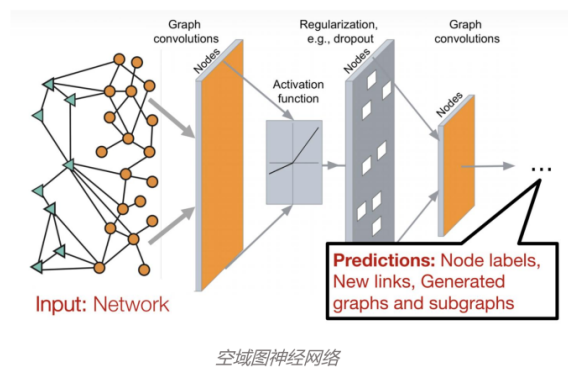
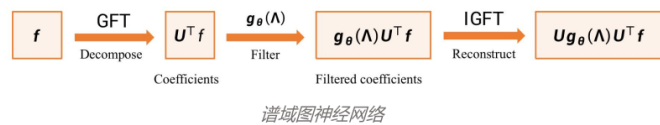
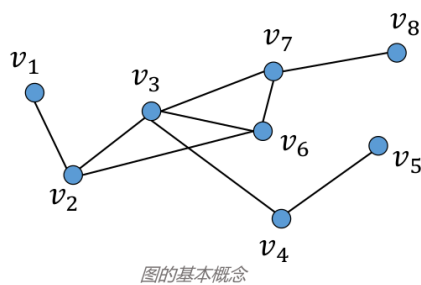


### ➤图深度学习基础

- 图的基本概念
- 图傅立叶变换
- 图卷积

### ➤谱域图神经网络

### ➤空域图神经网络



$$L = D - A \rightarrow \text{归一化: } L^{sys} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} / L^{rw} = D^{-1} L = I - D^{-1} A$$

### 谱域图神经网络过程

图的傅里叶变换, 乘以正交矩阵

➤  $L = U\Lambda U^T$  为其特征值分解， $U$  的列向量类比于傅里叶变换中的基

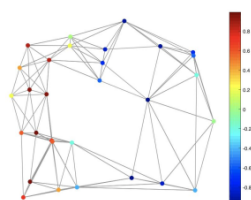
➤ 图傅里叶逆变换：

$$f(i) = \sum_{l=1}^n \hat{f}(\lambda_l) u_l(i) = \mathbf{u}(i)^T \hat{\mathbf{f}}$$

$$\begin{pmatrix} f(1) \\ f(2) \\ \vdots \\ f(N) \end{pmatrix} = \begin{pmatrix} u_1(1) & u_2(1) & \dots & u_N(1) \\ u_1(2) & u_2(2) & \dots & u_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(N) & u_2(N) & \dots & u_N(N) \end{pmatrix} \begin{pmatrix} \hat{f}(\lambda_1) \\ \hat{f}(\lambda_2) \\ \vdots \\ \hat{f}(\lambda_N) \end{pmatrix}$$

任一信号  $f$  是不同频率的特征向量的组合

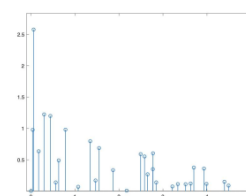
$$\mathbf{f} = \mathbf{U} \hat{\mathbf{f}}$$



Spatial domain:  $\mathbf{f}$

$$\mathbf{f} = \mathbf{U} \hat{\mathbf{f}}$$

Decompose signal  $\mathbf{f}$



Spectral domain:  $\hat{\mathbf{f}}$

*The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and other Irregular Domains. IEEE Signal Processing Magazine*

函数的卷积公式：  $f * g = F^{-1}\{F\{f\} \cdot F\{g\}\}$

对应在上图：

## 谱域上的图卷积

➤ 空域上很难定义卷积 → 转到谱域

➤ 从图信号处理的角度考虑图卷积

➤ 卷积公式  $f * g = F^{-1}\{F\{f\} \cdot F\{g\}\}$

**卷积定理：** 函数卷积的傅里叶变换是函数傅立叶变换的乘积

➤ 给一个图信号  $\mathbf{x}$  和一个卷积核  $\mathbf{g}$   $\mathbf{x} * \mathbf{g} = \mathbf{U}(\mathbf{U}^T \mathbf{x} \odot \mathbf{U}^T \mathbf{g})$

➤  $\mathbf{U}^T \mathbf{g}$  当成整体的卷积核，用参数  $\boldsymbol{\theta}$  表示

$$\mathbf{x} * \mathbf{g} = \mathbf{U}(\mathbf{U}^T \mathbf{x} \odot \mathbf{U}^T \mathbf{g}) = \mathbf{U}(\mathbf{U}^T \mathbf{x} \odot \boldsymbol{\theta}) = \mathbf{U} \mathbf{g}_{\boldsymbol{\theta}} \mathbf{U}^T \mathbf{x}$$

用  $\mathbf{g}_{\boldsymbol{\theta}}$  表示对角线为  $\boldsymbol{\theta}$  的矩阵

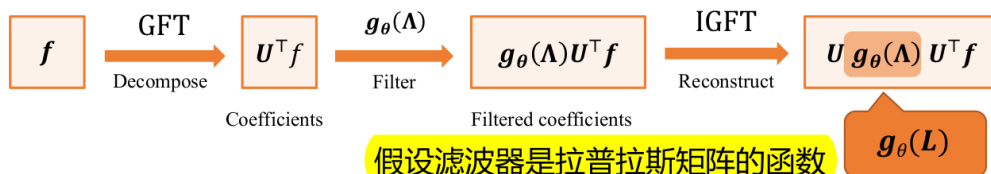
$\mathbf{g}_{\boldsymbol{\theta}}$  = 多项式卷积核,  $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$



# 多项式卷积核 (ChebyNet)



➤局部化:  $L$ 对图信号 $f$ 的操作 $L f$ 相当于在图上传播一步 常用归一化的 $L$ , 比如 $L^{sym}$



➤多项式卷积核:

$$g_\theta(\Lambda) = \sum_{k=0}^K \theta_k \Lambda^k$$

$$U \hat{g}(\Lambda) U^T f = U \sum_{k=0}^K \theta_k \Lambda^k U^T = \sum_{k=0}^K \theta_k L^k f$$

$$L^k = U \Lambda^k U^T$$

$$\hat{g}(\Lambda) = \begin{bmatrix} \sum_{k=0}^K \theta_k \lambda_1^k & & \\ & \sum_{k=0}^K \theta_k \lambda_2^k & \\ & & \dots \\ & & & \sum_{k=0}^K \theta_k \lambda_N^k \end{bmatrix}$$

局部性: 实际上在图上传播了 $K$ 步, 因此一个节点只影响它周围距离为 $K$ 以内的邻居

效率高: 不需要进行特征值分解

参数少: 只有 $K+1$ 个参数

记 $T_k(\tilde{\Lambda}) = 2 \wedge T_{k-1}(\tilde{\Lambda}) - T_{k-2}(\tilde{\Lambda})$ , 其中 $T_k(\tilde{L}) = U T_k(\tilde{\Lambda}) U^T$ , 由归纳法易证

## 多项式卷积核 (ChebyNet)



➤切比雪夫多项式, 通过如下递归定义

$$\begin{aligned} T_0(x) &= 1; T_1(x) = x \\ T_k(x) &= 2xT_{k-1}(x) - T_{k-2}(x) \end{aligned}$$

➤由于其定义域为 $[-1, 1]$ , 因此 $\hat{g}(\Lambda)$ 通过如下方式定义

$$\hat{g}(\Lambda) = \sum_{k=0}^K \theta_k T_k(\tilde{\Lambda}) \quad \text{其中 } \tilde{\Lambda} = \frac{2\Lambda}{\lambda_{\max}} - I$$

➤在归一化之后, 仍然可以不用计算特征值分解

➤有结论  $U \hat{g}(\Lambda) U^T f = \sum_{k=0}^K \theta_k T_k(\tilde{L}) f$ , 其中  $\tilde{L} = \frac{2L}{\lambda_{\max}} - I$

只取1阶chebyshev多项式,  $\lambda = 2$ ,  $\tilde{L} = L - I = -D^{-\frac{1}{2}} A D^{\frac{1}{2}}$



# 从ChebyNet到GCN

➤假设我们只取1阶切比雪夫多项式，且 $\lambda_{max} \approx 2$

$$y = \sum_{k=0}^1 \theta_k T_k(\tilde{L}) x \approx \theta_0 T_0(\tilde{L}) x + \theta_1 T_1(\tilde{L}) x \\ \approx \theta_0 x - \theta_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x$$

$$\text{取 } \theta' = \theta_0 = -\theta_1 \approx \theta' \left( I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) x$$

➤此时节点只能被它周围的1阶邻接点所影响，但只需要叠加K层这样的图卷积层，就可以把节点的影响力扩展到K阶邻居节点

根据类似方法，可证明得 $I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ 的特征值在[0,2]之间，叠加多层图卷积层会多次迭代这个操作，可能造成数值不稳定和梯度爆炸的问题

$$I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \xrightarrow{\text{做归一化}} \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \Rightarrow Y = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta \\ \tilde{A} = A + I \\ \tilde{D} = D + I \\ X \in R^{n \times d}: \text{节点属性矩阵}$$

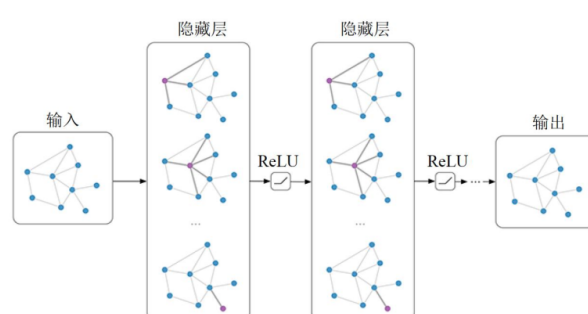
这样得到了图卷积网络迭代公式 $Y = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta$ ；常用多层GCN，下图中的 $W = \Theta$



# 从ChebyNet到GCN

➤叠加多层GCN得到最终的模型(常取两层)

$$\hat{Y} = f(X, A) = \text{Softmax} \left( \hat{A} \text{ReLU}(\hat{A} X W^0) W^1 \right)$$



其中

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \\ \tilde{A} = A + I_n \\ \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$$

## GCN的缺陷

defect1: GCN的简单聚合失去了邻域节点的结构信息。

因为GCN简单地将所有一跳邻居的标准化 "信息" 相加，没有区分来自不同节点的 "消息"。(Kipf & Welling, 2017)。

result1:因此，在这种聚合之后，我们无法知道哪个节点对最终的聚合输出有什么贡献。

inclusion1: 如果不对这种结构信息进行建模，现有的MPNN不能区分某些非同构的图。

improve1:而经典的卷积神经网络 (CNN) 通过使用聚合器 (即卷积滤波器) 来避免这个问题，其结构接收文件定义在网格上，即欧氏空间，

defect2: 聚合器缺乏捕捉异构图中长距离依赖关系的能力。

在MPNN中，邻域被定义为一跳之外的所有邻居的集合（如GCN），或r跳之外的所有邻居（如ChebNet）。

解决这一限制的直接策略是使用多层结构，以便接收来自远处节点的 "信息"。

improvement1：使用多层结构，以便接受来自远处节点的"信息"，如果把每一层ln看作一个时间步， $ln=f(fn-k,...,fn-1)$ ，和RNN很像 $\rightarrow h_t = f(h_{t-k}, \dots, h_{t-1})$

problem1：来自远处节点的相关信息与来自近处节点的大量不相关信息不可区分地混合在一起，这意味着相关信息将被 "洗掉"，无法有效提取。

problem2：在多层MPNN中，不同节点的表征会变得非常相似，每个节点的表征实际上都承载着整个图的信息。

但在LSTM中，可以通过添加激活层和历史门+记忆门的机制+不同的输入，来保证输出的差异性。

## 改进的基本思想

1. 由于连续空间中的静止性、局部性和组成性，经典神经网络有效地解决了类似的限制；
2. 网络几何学旨在通过揭示网络背后的潜伏连续空间来理解网络，它假定节点是从潜伏连续空间中离散采样的，而边缘是根据它们的距离建立的。

## 新的想法

图上的聚合是否可以从连续的潜在空间中获益，比如利用空间中的几何图形来建立结构邻域，并捕捉图中的长程依赖关系？

## 改进的实现方案——Geom-GCN

为了回答上述问题，作者提出了一种新的图神经网络的聚合方案，称为几何聚合方案。在该方案中，我们通过节点嵌入将图映射到一个连续的潜空间，然后利用潜空间中定义的几何关系来建立结构邻域进行聚合。同时，我们设计了一个在结构邻域上操作的双级聚合器，以更新图神经网络中节点的特征表示，这能够保证图结构数据的变异不变性。与现有的MPNN相比，该方案提取了更多的图的结构信息，并且可以通过将远处的节点映射到潜空间中定义的邻域来聚合特征表示。

- 网络的聚合——图卷积操作，输入结点的特征与连接关系，输出结点的标签

我们提出了一个在图卷积网(连续空间)中的几何聚合方案的实现，我们称之为Geom-GCN，用于在图上进行归纳学习、节点分类。

潜在连续空间的获取：我们选择不同的嵌入方法将图映射到合适的潜在空间，用于不同的应用，其中图的合适拓扑结构模式是预先服务的。

结构邻域的获取：我们设计了特殊的几何关系，分别在欧几里得和双曲嵌入空间建立结构邻域。

## 原理分析

### 嵌入f与关系算符τ

几何嵌入=映射 $f: x^v \rightarrow z^d$ (原空间V到潜在空间Z的线性变换或非线性变换)，

example：词嵌入是 $Z^1$ (一维整数)到 $R^d$ (d维实数)的映射

作者在图和潜在空间的基础上，提出了**结构性邻域**的concept，

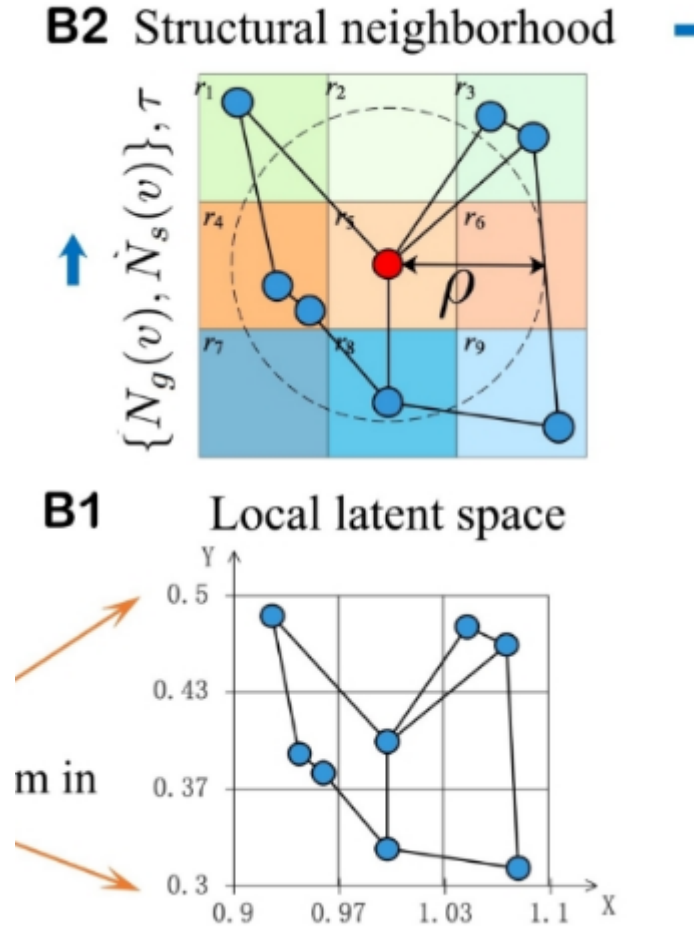
设 $N_g(v)$ 为v的邻点集合， $N_s(v)$ 为与v在潜在空间较近的集合(具有一定的相似性，虽然和v在原空间可能离得很远)，

$N_g(v) = \{u|u \in V, (u, v) \in E\}$ ,  $N_s(v) = \{u|u \in V, d(z_u, z_v) < \rho\}$ , 其中 $d(-, -)$ 取决于空间Z的特定度量,

则 $v$ 的结构性邻域 $N(v) = (\{N_g(v), N_s(v)\}, \tau)$ , 其中 $\tau: (z_v, z_u) \rightarrow r \in R$ , 表示潜在空间中 $u$ 和 $v$ 的几何关系,

$R$ 是几何关系的集合, 定义为任意的感兴趣的几何关系。

example: 在下面的图中, 几何关系 $R$ 用3\*3网格的不同颜色表示, 有9种几何关系。



双级聚合

$$e_{(i,r)}^{v,l+1} = p(\{h_u^l | u \in N_i(v), \tau(z_v, z_u) = r\}), \forall i \in \{g, s\}, \forall r \in R \quad (\text{Low-level aggregation})$$

$$m_v^{l+1} = \frac{q}{i \in \{g, s\}, r \in R} ((e_{(i,r)}^{v,l+1}, (i, r))) \quad (\text{High-level aggregation}) \quad (1)$$

$$h_v^{l+1} = \sigma(W_l \cdot m_v^{l+1}) \quad (\text{Non-linear transform})$$

先在格子里池化, 属于同一个空间的几何关系且为近邻的节点的隐藏层作池化, 得到虚拟节点 $e_{(i,r)}^{v,l+1}$ ,

再把虚拟节点(不同空间下的几何关系)按空间和几何关系作聚合, 得到新的隐藏层。

comments:

1. 分别考虑了节点在潜在空间的相似性以及在原空间的邻接关系, 映射到某个 $e_{(i,r)}^{v,l+1}$ , 表示节点在原空间/潜在空间的某种几何关系的邻域信息,
2. 在函数 $q$ 中, 以邻域信息与邻域位置序列作为输入(而非简单的邻域信息, 使得不同邻域的信息得以有效区分!), 最后作一个常见的线性变换+激活拟合目标变换。

3. 借鉴了LSTM的思想（可以把 $e_{(i,r)}^{v,l+1}$ 视为输入门， $m_v^{l+1}$ 视为内层， $h_v^{l+1}$ 视为外层）——将隐藏层分为内外两层，输入门参考历史信息，并在内层将历史信息与输入门聚合，内层用一个激活函数和线性函数得到外层

输入门： $\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$

内层： $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$

外层： $h_t = o_t \odot \tanh(c_t)$

只不过这里LSTM是将历史外层与当前输入组合得到输入门，将输入门与历史内层组合得到内层隐藏层

而双级聚合是将历史外层聚合得到(空间，几何关系)节点，将空间几何关系节点根据次序聚合得到内层隐藏层，

## 变换不变性

变换不变性是GNN的一个基本要求，

变换不变性是指经过一个映射 $\phi: V \rightarrow V'$ ， $v' = \psi(v)$ ， $G=(V,E)$ 的某种性质 $\phi(G) = \phi(G')$ ， $G'=(V',E')$

## 证明

易见Lemma1：若 $\phi_2(G)$ 具备变换不变性，则 $\phi_1 \circ \phi_2(G)$ 也是变换不变的。

双级聚合公式(1)中，可将 $\sigma(W_l \cdot \sum_{i \in \{g,s\}, r \in R} q_i())$ 视为 $\phi_1$ ，low-level aggregation视为 $\phi_2$ ，以G为输入，

由于low-level具有变换不变性——位置关系与邻接关系由(V,E)决定，结点编号变化，位置关系不变，交换各节点的编号，在 $2R$ 个子空间 ( $i \in \{g,s\}, r \in R$ )的所有结点的邻域信息不变， $\phi_2$ 变换不变，所以由Lemma1， $\phi_1 \circ \phi_2(G)$ 变换不变，

## 有效性的论证

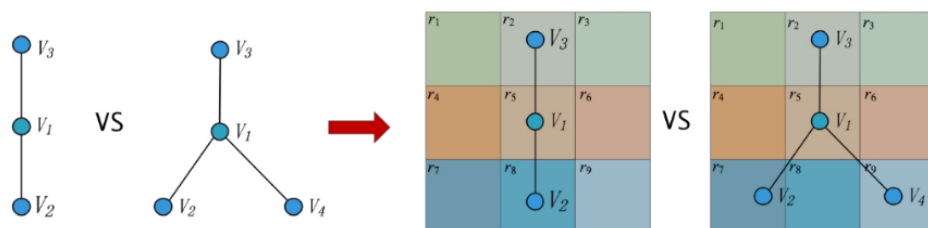
1. 保留结点的邻域信息，而不是非同构的图具有相同的特征表示(变换层是线性的且不区分邻域)，

举个例子，GCN简单地将所有1跳邻居的标准化信息相加，这样的聚合失去了邻域结点的结构信息——v的不同邻域结点在G中的邻接关系不同，空间位置也不同，导致无法知道哪个节点对最终的聚合输出有什么贡献。

在双级聚合中，我们将v的近邻(原空间或潜在空间)按照几何关系R分类，使得邻域结点的结构信息得以区分，

然后再与几何关系的位置信息一起，作为聚合其q的输入。

例子如下：



假设采用V1用平均聚合器抽取邻域结点的信息，V2,V3,V4的特征均为a，

在左1和左2中， $out_1(V1)=out_2(V1)=f(a)$ ，无法区分这两种结构；

而在左3和左4中,  $\text{out3}(V1)=(f_2(a)+f_8(a))/2$ ,  $\text{out4}(V1)=(f_2(a)+f_7(a)+f_9(a))/3$ , 在两种图u结构下,  $V1$  的邻域信息得到了区分。

## 2. 捕捉长距离依赖关系——

在双级聚合中, 使用 $N_g(v)$ 代表原空间的邻居, 用 $N_s(v)$ 代表和 $v$ 在原空间相距较远, 但具有某种相似性的邻点,

$h_u, u \in N_s(v)$ 可以代表 $v$ 相距较远, 但有相互作用的结点的信息,  $h_u, u \in N_g(v)$ 代表 $v$ 的近邻结点的信息,

用函数 $q$ 提取这两种与 $v$ 相关的结点的信息, 传递给外层输出层,

相关工作使用注意力机制和节点和/或边缘属性学习不同邻居的 "信息 "权重。CCN (Kondor等人, 2018) 利用一个协方差架构来学习结构感知表征。

原文则从图的拓扑结构方面利用地理度量关系, 而其他方法则侧重于特征表示, 这两个方面是互补的。

## 方案的实例化

即给出理论中函数 ( $f$ =原空间到潜在空间的映射,  $\tau$ =两个空间坐标到某个实数的映射,  $p$ 和 $q$ 分别为一种聚合函数) 的实例化,

嵌入空间—— $N_s(v) = \{u|u \in V, d(z_u, z_v) < \rho\}$ ,

作者采用了3种嵌入方法(原空间到潜在空间的映射)——Isomap&Poincare&Struc2vec

关于 $\rho$ 的确定值得一提, 作者经 $\rho$ 从0开始增加, 直到在latent space中的结点的平均连通度 $= \frac{\sum_{i \in v} \text{deg}'(v)}{|V|}$ 与原图中的平均连通度 $\frac{\sum_{i \in v} \text{deg}(v)}{|V|}$ 属于同一个量级, 即原空间和潜在空间中, 结点有相近的"邻居"。

几何算子 $\tau: (z_v, z_u) \rightarrow r \in R$ ——二维欧几里得或双曲空间中两个节点之间的相对位置的四个关系,  $R=\{\text{左上}, \text{右上}, \text{左下}, \text{右下}\}$ 。

$\tau(z_v, z_u)$	$z_v[0] > z_u[0]$	$z_v[0] \leq z_u[0]$
$z_v[1] \leq z_u[1]$	左上	右上
$z_v[1] > z_u[1]$	左下	右下

聚合函数 $p$ 可用GCN的归一化隐藏层,

$$e_{(i,r)}^{v,l+1} = \sum_{u \in N_i(v)} \delta(\tau(z_v, z_u), r) (\text{deg}(v) \text{deg}(u))^{\frac{1}{2}} h_u^l, \forall i \in \{g, s\}, \forall r \in R,$$

聚合函数 $q$ 为平均池化层, 输出层为

$$h_v^{l+1} = \sigma(W_l \cdot \parallel_{i \in \{g, s\}} \parallel_{r \in R} e_{(i,r)}^{v,l+1})$$

## comments

精华1: 将原空间映射到潜在空间

精华2: 将各潜在空间的邻点按与 $v$ 的几何关系区分



实验结果与讨论

分类准确率比较

数据集的形式均为节点特征与标签,

在不同数据集下的分类准确率结果如下:

Table 3: Mean Classification Accuracy (Percent)

Dataset	Cora	Cite.	Pubm.	Cham.	Squi.	Actor	Corn.	Texa.	Wisc.
GCN	85.77	73.68	88.13	28.18	23.96	26.86	52.70	52.16	45.88
GAT	<b>86.37</b>	74.32	87.62	42.93	30.03	28.45	54.32	58.38	49.41
Geom-GCN-I	85.19	<b>77.99</b>	<b>90.05</b>	60.31	33.32	29.09	56.76	57.58	58.24
Geom-GCN-P	84.93	75.14	88.09	<b>60.90</b>	<b>38.14</b>	<b>31.63</b>	<b>60.81</b>	<b>67.57</b>	<b>64.12</b>
Geom-GCN-S	85.27	74.71	84.75	59.96	36.24	30.30	55.68	59.73	56.67

Geom-GCN的变体的探讨

作者还探索了对3种Geom-GCN的变体——每个点的邻居仅有Ng(v)或Ns(v)来分别评估邻域Ng(v)和Ns(v)的贡献, 以Geom-GCN-g与Geom-GCN-s来表示, 并用↑表示相比原GCN对准确率的增益, ↓表示负增益

Table 4: Mean Classification Accuracy (Percent)

Dataset	Cora	Cite.	Pumb.	Cham.	Squi.	Actor	Corn.	Texa.	Wisc.
$\beta$	0.83	0.71	0.79	0.25	0.22	0.24	0.11	0.06	0.16
Geom-GCN-I-g	86.26 ↑0.48	<b>80.64</b> ↑6.96	<b>90.72</b> ↑2.59	<b>68.00</b> ↑39.82	<b>46.01</b> ↑22.05	31.96 ↑4.04	65.40 ↑12.70	72.51 ↑21.35	68.23 ↑22.35
Geom-GCN-I-s	77.34 ↓8.34	72.22 ↓1.46	85.02 ↓3.11	61.64 ↑33.46	37.98 ↑14.02	30.59 ↑2.67	62.16 ↑9.46	60.54 ↑8.38	64.90 ↑19.01
Geom-GCN-P-g	86.30 ↑0.52	75.45 ↑1.76	88.40 ↑0.27	63.07 ↑34.89	38.41 ↑14.45	31.55 ↑3.63	64.05 ↑11.35	73.05 ↑21.89	69.41 ↑23.53
Geom-GCN-P-s	73.14 ↓12.63	71.65 ↓2.04	86.95 ↓1.18	43.20 ↑15.02	30.47 ↑6.51	<b>34.59</b> ↑6.67	<b>75.40</b> ↑22.70	<b>73.51</b> ↑21.35	<b>80.39</b> ↑34.51
Geom-GCN-S-g	<b>87.00</b> ↑1.23	75.73 ↑2.04	88.44 ↑0.31	67.04 ↑38.86	44.92 ↑20.96	31.27 ↑3.35	67.02 ↑14.32	71.62 ↑19.46	69.41 ↑23.52
Geom-GCN-S-s	66.92 ↓18.85	66.03 ↓7.65	79.41 ↓8.72	49.21 ↑21.03	31.27 ↑7.31	30.32 ↑2.40	62.43 ↑9.73	63.24 ↑11.08	64.51 ↑18.63

并用 $\beta = \frac{1}{|V|} \sum_{v \in V} \frac{v \text{的邻居中和} v \text{含有相同标签的结点数}}{v \text{的邻居数目}}$  来代表图的同源性(对称性)

结果显示:

1. 邻域和潜空间中的邻域在大多数情况下都有利于聚合;
2. 从每一行来看, 明显可以观察到, 在不对称图中, Geom-GCN-X-g/s的收益比在对称图中要大得多, 这表明, 采用映射到潜在空间后(或在原空间)的双级聚合, 能有效区分结点的邻域信息, 与捕捉长程依赖关系;
3. 一个非常重要的观察是, Geom-GCN的变体在不对称图中的准确率明显高于Geom-GCN。  
合理的推测是有两个邻域的Geom-GCN比只有一个邻域的Geom-GCN聚集了更多不相关的 "信息", 而这些不相关的 "信息" 对性能有不利影响,  
可以采用注意力机制来缓解。
4. Geom-GCN-P在不对称图中的表示是3种嵌入方式种最出色的, Geom-GCN-I在对称图中的表现最出色。

这启示我们根据不同的模型，采用不同的嵌入模式——设计一个能够自动确定Geom-GCN正确嵌入空间的端到端框架。

## 时间复杂度分析

设原空间的结点的特征大小= $n$ ，结点数= $N$ ，Non-linear transform中隐藏单元的数量= $|V'|=m$ ，

$$e_{(i,r)}^{v,l+1} = p(\{h_u^l | u \in N_i(v), \tau(z_v, z_u) = r\}), \forall i \in \{g, s\}, \forall r \in R \quad (\text{Low-level aggregation})$$

$$m_v^{l+1} = \frac{q}{i \in \{g, s\}, r \in R} ((e_{(i,r)}^{v,l+1}, (i, r))) \quad (\text{High-level aggregation}) \quad (1)$$

$$h_v^{l+1} = \sigma(W_l \cdot m_v^{l+1}) \quad (\text{Non-linear transform})$$

"Low-level aggregation"——每个结点 $v$ ，划分 $e_{(i,r)}^{v,l+1}, \forall i \in \{g, s\}, \forall r \in R$ ，耗时 $O(N*n)$ ，聚合耗时 $O(N*n)$ ，共耗时 $O(N*n)$ ，共 $O(N*N*n)$

"High-level aggregation"用时 $O(2|R|*n*N)$ 个， $N$ 个 $v$ ，每个 $v$ 聚合总耗时 $2|R|*n$ ；

"Non-linear transformer"为线性层+激活层(sigmoid函数)，用时 $O(n*m*2|R|)$ ；

总时间复杂度= $O(n*m*2|R|)$ ，而GCN的时间复杂度为 $O(n*m)$ ，需要开发加速技术。

## 收获

通过阅读《GEOM-GCN: GEOMETRIC GRAPH CONVOLUTIONAL NETWORKS》一文，

1. 回顾了GCN的实现机制，并进一步了解传统GCN的缺陷——无法表征邻域的结构信息，无法捕捉长距离依赖关系，并在后者的解决方案中讨论了MPNN与LSTM用于捕捉长程依赖的相似性；
2. 了解通过潜在空间与其中自定义的关系算符建立新的邻域，再通过聚合提取邻域信息的机制，来有效表征邻域关系，并建立远距离依赖关系，观赏到映射空间的强大威力；
3. 在作者不局限于Geom-GCN与GCN和GAT的比较，进一步探讨Geom-GCN变体与数据集对称性的实践中，学习到需要挖掘数据集与模型的特性，合理推测，之后实践与检验新的idea，才能有更深的认识。