

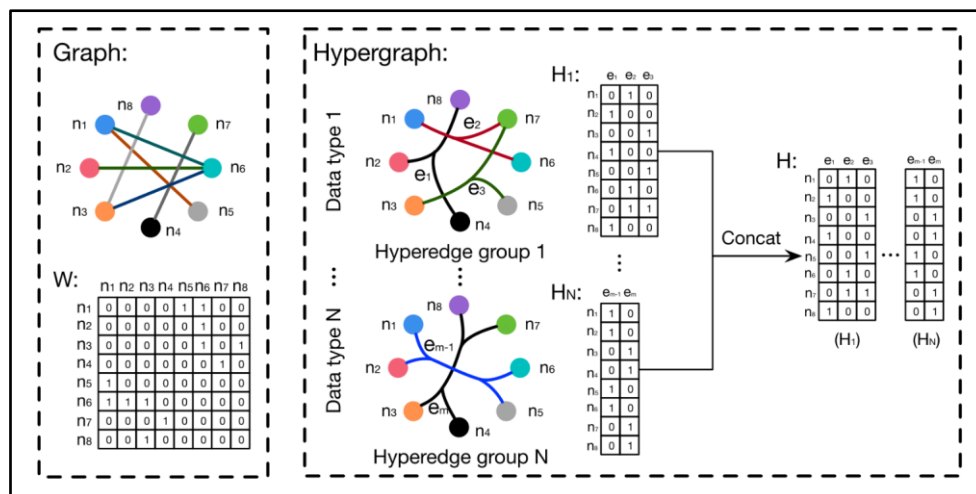
《Hypergraph Neural Networks》

《超图神经网络》

摘要:

1. **本文的背景:** 在现实世界中, 复杂的网络关系如果单纯地依靠普通的图结构来表示有时并不能很好地解决问题。而研究者们发现在处理复杂数据时, 超图(一种比图更通用的数据结构)在数据建模方面往往更加灵活。因此, 本文旨在提出一种基于超图结构的图神经网络来应对复杂数据带来的挑战。
2. **本文的贡献:** 提出了一种超图神经网络(*HGNN*)框架, 该框架使用超图结构进行数据建模, 从而能够处理多模态数据和复杂的数据相关性。
3. **主要创新点:** 类比*GCN*, 设计了一种超边卷积操作, 从而能够更好地利用高阶数据相关性来进行节点的嵌入表示学习。
4. **实验结果:** 在 4 个数据集上的实验结果以及与*GCN*等传统方法的比较表明, *HGNN*具有更好的性能。并且, 这些结果还表明*HGNN*方法在学习使用高阶复杂关联表示的数据时更加有效。

超图的定义:



如上图所示, 与所有边的度都必须为 2 (一条边仅连接两个节点) 的简单图相比, 超图可以使用其无限制的超边对高阶数据相关性 (超越成对连接) 进行编码。即, 超图上的一个“边”连接的不再仅是两个节点, 而可以是任意多个节点。此外, 在上图中我们还可以看到, 简单图的邻接矩阵是用“节点—节点”的方法表示的, 其中 1 代表有连接, 0 代表无连接, 这种表示方式的作用极其有限。

而超图的邻接矩阵是用“节点—超边”的方法来表示的，因此它可以通过其灵活的超边扩展以实现多模态、异构的数据表示。例如，上图中对两种不同类型的超边集合的合并可以直接通过对其邻接矩阵的拼接（*Concat*）来实现。【一个超图可以通过合并邻接矩阵，联合使用多模态数据来生成一个新的超图。】

因此，超图 G 的定义为 $G = (V, E, W)$ 。其中， V 代表节点集， E 代表超边集，而 W 是超边的对角线权重矩阵（每条超边都会按照 W 来分配一个权重）。此外，超图 G 还可以用 $\mathbb{R}^{|V| \times |E|}$ 的邻接矩阵 H 来表示，而其中各元素的定义为：

$$h(v, e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e, \end{cases}$$

最后，对于节点 $v \in V$ ，其度定义为 $d(v) = \sum_{e \in E} w(e)h(v, e)$ ；对于边 $e \in E$ ，其度数定义为 $\delta(e) = \sum_{v \in V} h(v, e)$ 。而 D_e 和 D_v 分别表示“边度”和“节点度”的对角矩阵。

超图上的谱卷积：

与 GCN 上的谱卷积类似，通过利用切比雪夫不等式的一阶近似以及假设 $\lambda_{max} = 2$ ，使得 $g_\theta \star x \approx \sum_{k=0}^K \theta_k T_k(\tilde{\Delta})x$ 化简为：

$$g_\theta \star x \approx \theta_0 x + \theta_1 (\Delta - I_N)x = \theta_0 x - \theta_1 D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} x$$

其中， θ_0 和 θ_1 为可调节参数。

接着，对参数 θ_0 和 θ_1 做出进一步限制，使得：

$$\begin{cases} \theta_1 = -\frac{1}{2}\theta \\ \theta_0 = \frac{1}{2}\theta D_v^{-1/2} H D_e^{-1} H^T D_v^{-1/2}, \end{cases}$$

从而得到卷积操作的简化形式：

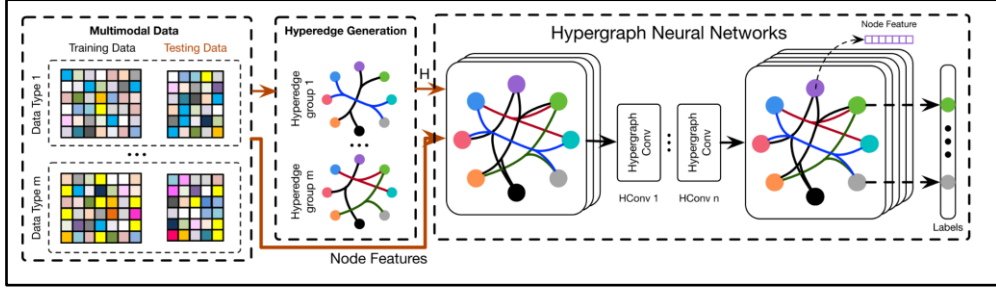
$$\begin{aligned} g \star x &\approx \frac{1}{2} \theta D_v^{-1/2} H (W + I) D_e^{-1} H^T D_v^{-1/2} x \\ &\approx \theta D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} x, \end{aligned}$$

其中， $(W + I)$ 可以被认为是超边的权重矩阵，而 W 初始化为单位矩阵，这意味着初始条件下每一条超边的权重相等。因此，当我们有一个具有 n 个节点和 C_1 维特征的超图信号 $X \in \mathbb{R}^{n \times C_1}$ 时，我们的超图上的谱卷积就可以由下式表示：

$$Y = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} X \Theta$$

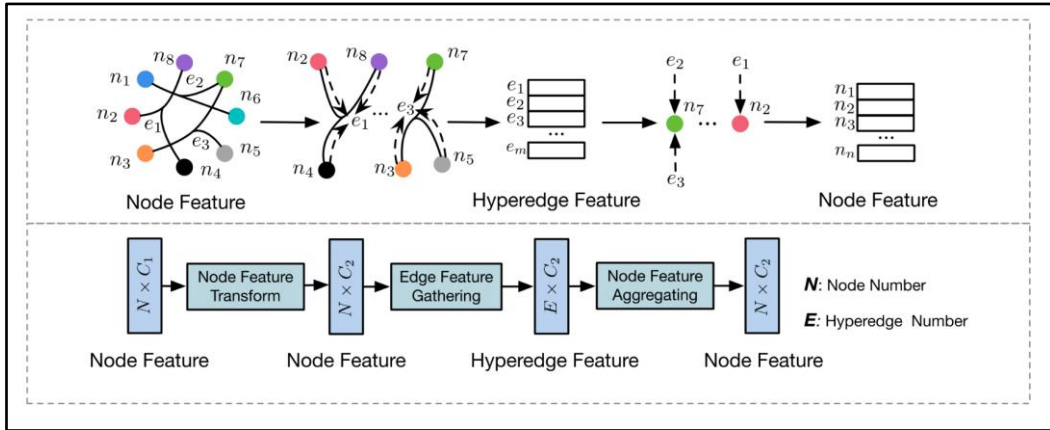
其中, $W = \text{diag}(w_1, \dots, w_n)$, $\theta \in \mathbb{R}^{C_1 \times C_2}$ 是训练过程中要学习的参数。(我们将滤波器 θ 应用于超图中的节点以提取特征。)而在经过卷积后,我们就可以得到 $Y \in \mathbb{R}^{n \times C_2}$, 可应用于分类的下游任务。

HGNN模型的整体框架:



HGNN的整体框架如上图所示。首先, 将多模态的数据关系作为输入 (包含训练数据和测试数据)。然后, 将它们同时考虑来进行超边的生成, 得到 H 和 X 。接着, 通过利用 H 和 X 进行超图卷积得到最终的节点嵌入结果 $X^{(L)}$ 。最后, 根据下游分类任务的要求进行输出, 得到分类结果 Y 。

而超图卷积层 (**Hypergraph Conv**) 的具体细节过程如下图所示:



简单来说, HGNN层执行的是利用超图结构“节点—>边缘—>节点”的转换。首先, 对初始节点特征 $X^{(l)} \in \mathbb{R}^{N \times C_1}$ 进行特征提取, 利用可学习的过滤处理矩阵 $\theta^{(l)}$ 提取 C_2 维的特征 $\mathbb{R}^{N \times C_2}$; 然后, 再根据超边 H 收集节点特征, 形成超边特征 $\mathbb{R}^{E \times C_2}$, 该过程由 $H^T \in \mathbb{R}^{E \times N}$ 实现; 最后, 再通过矩阵 H 的相乘, 聚合其相关超边特征得到输出的节点特征 $X^{(l+1)} \in \mathbb{R}^{N \times C_2}$ 。于是, HGNN层就通过“节点—>边缘—>节点”的变换有效地提取到了超图上的高阶相关性。

因此，超边卷积层 $f(X, W, \theta)$ 的卷积公式可表示为：

$$X^{(l+1)} = \sigma \left(D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} X^{(l)} \theta^{(l)} \right)$$

其中， $X^{(l)} \in \mathbb{R}^{N \times C_l}$ 是第 l 层超图的信号， $X^{(0)} = X$ ， σ 表示非线性激活函数。

HGNN与GCN的关系：

当超边只连接两个节点时，超图就会被简化为简单图，它的拉普拉斯算子 Δ 也会与简单图的拉普拉斯算子重合，其系数为 $\frac{1}{2}$ 。而与GCN相比，HGNN可以更加自然地模拟数据之间的高阶关系，并在形成特征提取时更加有效地利用和编码这种关系。此外，与传统的超图方法相比，HGNN模型具有很高的计算效率，因为它不需要进行拉普拉斯算子 Δ 的逆运算。还需要注意的是，HGNN对多模态特征也具有很大的可扩展性，因为它具有超边生成的灵活性。

实验及结果：

1) 超边构造

在视觉对象分类任务中， N 个视觉对象数据的特征可以表示为 $X = [x_1, x_2, \dots, x_n]^T$ 。而我们根据两个特征之间的距离建立超图。具体来说，使用欧几里得距离计算任意一对节点的 $d(x_i, x_j)$ 。在构造中，每个节点代表一个可视对象，而每个超边由一个节点与其 K 个最近邻连接而成，从而产生 N 个超边，连接 $K + 1$ 个顶点。因此，我们得到关联矩阵 $H \in \mathbb{R}^{N \times N}$ ，其中 $N \times (K + 1)$ 项等于 1，而其他项等于 0。

而在引文网络分类中，数据以图的结构组织，每个超边都是根据图上的邻接关系将一个顶点与其相邻的顶点连接起来建立的。所以，也得到了 N 条超边和 $H \in \mathbb{R}^{N \times N}$ 。

2) 节点分类任务

对于节点分类问题，实验将数据集分为训练数据和测试数据。然后，构造超图，生成关联矩阵 $H \in \mathbb{R}^{N \times N}$ 和对应的 D_e 、 D_v 对角矩阵。其次，利用超边卷积层强大的能力构建两层HGNN模型，并利用 $softmax$ 函数生成预测标签 \hat{Y} 。此外，在模型的训练过程中，训练数据通过熵损失反向传播来更新参数 θ 。然后，利用测

试集标签预测的准确率来评估模型性能。而当存在多模态信息时，通过构造超边组将其融入其中，然后将各种超边融合在一起，从而对数据上的复杂关系进行建模。

3) 数据集

引文网络数据集 Cora 和 Pubmed:

| Dataset | Cora | Pubmed |
|-----------------|------|--------|
| Nodes | 2708 | 19717 |
| Edges | 5429 | 44338 |
| Feature | 1433 | 500 |
| Training node | 140 | 60 |
| Validation node | 500 | 500 |
| Testing node | 1000 | 1000 |
| Classes | 7 | 3 |

Table 1: Summary of the citation classification datasets.

视觉对象识别数据集 ModelNet40 和 NTU:

| Dataset | ModelNet40 | NTU |
|---------------|------------|------|
| Objects | 12311 | 2012 |
| MVCNN Feature | 4096 | 4096 |
| GVCNN Feature | 2048 | 2048 |
| Training node | 9843 | 1639 |
| Testing node | 2468 | 373 |
| Classes | 40 | 67 |

Table 3: The detailed information of the ModelNet40 and the NTU datasets.

4) 引文网络分类实验结果

| Method | Cora | Pubmed |
|--|--------------|--------------|
| DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) | 67.2% | 65.3% |
| ICA (Lu and Getoor 2003) | 75.1% | 73.9% |
| Planetoid (Yang, Cohen, and Salakhutdinov 2016) | 75.7% | 77.2% |
| Chebyshev (Defferrard, Breuss, and Vandergheynst 2016) | 81.2% | 74.4% |
| GCN (Kipf and Welling 2017) | 81.5% | 79.0% |
| HGNN | 81.6% | 80.1% |

Table 2: Classification results on the Cora and Pubmed datasets.

如图所示，可以看到与现有方法相比，*HGNN*模型能够达到最佳或可比的性能。与*GCN*相比，*HGNN*方法在 Cora 数据集上略有改进，而在 Pubmed 数据集上提高了1.1%。由于生成的超图结构与简单图结构非常相似，因为这些数据中既没有额外信息，也没有更复杂的信息。因此，*HGNN*获得的收益不是很大。

5) 视觉对象分类实验结果

| Feature | Features for Structure | | | | | |
|--------------------------|------------------------|--------------|-------|--------------|-------------|--------------|
| | GVCNN | | MVCNN | | GVCNN+MVCNN | |
| | GCN | HGNN | GCN | HGNN | GCN | HGNN |
| GVCNN (Feng et al. 2018) | 91.8% | 92.6% | 91.5% | 91.8% | 92.8% | 96.6% |
| MVCNN (Su et al. 2015) | 92.5% | 92.9% | 86.7% | 91.0% | 92.3% | 96.6% |
| GVCNN+MVCNN | - | - | - | - | 94.4% | 96.7% |

Table 4: Comparison between GCN and HGNN on the ModelNet40 dataset.

| Feature | Features for Structure | | | | | |
|----------------------------|------------------------|--------------|-------|--------------|-------------|--------------|
| | GVCNN | | MVCNN | | GVCNN+MVCNN | |
| | GCN | HGNN | GCN | HGNN | GCN | HGNN |
| GVCNN ((Feng et al. 2018)) | 78.8% | 82.5% | 78.8% | 79.1% | 75.9% | 84.2% |
| MVCNN ((Su et al. 2015)) | 74.0% | 77.2% | 71.3% | 75.6% | 73.2% | 83.6% |
| GVCNN+MVCNN | - | - | - | - | 76.1% | 84.2% |

Table 5: Comparison between GCN and HGNN on the NTU dataset.

如上图所示，与 GCN 相比， $HGNN$ 在所有实验中均取得了更好的性能。当只使用一个特征生成图/超图结构时， $HGNN$ 可以得到轻微的改进。例如，当仅使用 $GVCNN$ 作为对象特征，使用 $MVCNN$ 生成图/超图结构时， $HGNN$ 在 $ModelNet40$ 和 NTU 数据集上分别比 GCN 提高了0.3%和2.0%。而当有更多的特征时，即 $GVCNN$ 和 $MVCNN$ 都用于图/超图结构的生成时，与 GCN 相比， $HGNN$ 的性能要好得多。

| Method | Classification Accuracy |
|---------------------------------|-------------------------|
| PointNet (Qi et al. 2017a) | 89.2% |
| PointNet++ (Qi et al. 2017b) | 90.7% |
| PointCNN (Li et al. 2018) | 91.8% |
| SO-Net (Li, Chen, and Lee 2018) | 93.4% |
| HGNN | 96.7% |

Table 6: Experimental comparison among recent classification methods on ModelNet40 dataset.

如上图所示，所提出的 $HGNN$ 方法优于 $ModelNet40$ 数据集中最先进的目标识别方法。更具体地说，与 $PointCNN$ 和 $SO-Net$ 相比， $HGNN$ 方法可以分别实现4.8%和3.2%的提升。而上述这些结果表明，本文所提出的 $HGNN$ 方法在视觉对象识别方面具有优越的性能。

总结：

本文提出了一个超图神经网络（*HGNN*）框架，将图卷积运算推广到了超图学习过程中。而与传统图方法相比，*HGNN*是一个更通用的框架，能够通过超图结构处理复杂和高阶的相关性进行表示学习。此外，本文还通过引文网络分类和视觉对象识别任务进行了实验，而实验结果和与现有方法的比较均表明，*HGNN*模型具有更好的性能。

对本文的感悟：

本文所提出的*HGNN*模型其实主要也是对图结构做出进一步的改进，而与在异构图上的创新方法相比，利用超图结构是另一种思路（方向）的创新，值得我们进一步学习参考。