

超图 & 半监督 论文粗读

Wasserstein Soft Label Propagation on Hypergraphs: Algorithm and Generalization Error Bounds

核心思路：

- 背景和动机：**随着超图在机器学习和数据挖掘算法中的应用不断增加，本文探讨了在超图上通过最优传输进行“软标签”传播的半监督学习算法。软标签（如概率分布、类别成员分数）传播相较于传统的硬标签传播，具有更高的灵活性和信息量，适用于不确定性和分布信息至关重要的场景。
- 算法设计：**借鉴Wasserstein传播算法（Solomon et al., 2014），本文将标签传播过程重新构建为消息传递算法，通过Wasserstein重心实现了其在超图上的一般化。利用2-Wasserstein距离，提出了一种新的消息传递算法，用于在超图上传播一维分布。
- 理论贡献：**在PAC学习框架内，本文提供了在图和超图上传播一维分布的泛化误差界，通过建立所提出算法的算法稳定性，揭示了Wasserstein传播算法在图上的新见解和更深层次的理解。

贡献

- 扩展Wasserstein传播至超图：**提出了基于Wasserstein重心的超图软标签传播算法，并通过多边最优传输问题解决了该算法在超图上的推广。
- 算法稳定性和泛化误差界：**在PAC学习框架下，本文建立了使用2-Wasserstein距离传播一维分布的泛化误差界，提供了关于Wasserstein传播算法在图上泛化能力的首个理论结果。
- 数值实验和实际应用：**通过在随机超图和UCI数据集（包括国会投票记录和蘑菇特征数据）上的实验，展示了所提出算法的有效性。

传统的标签传播算法（如Belkin等人提出的算法）通常用于图（graph）上的标签传播，其标签为数值或类别变量。而**Wasserstein传播算法**通过最优传输理论将标签扩展为概率分布或软标签（soft label），提高了算法的灵活性和信息量。

- Wasserstein传播算法**最初在图（graph）上定义，其目的是在节点之间传播概率分布标签。本文将这一过程重新构建为消息传递算法，并扩展到超图（hypergraph）。

在图上的Wasserstein标签传播

给定一个图 $G = (V, E)$ ，定义一个度量空间上的概率分布映射 $\mu : V \rightarrow P(N)$

假设有一部分节点 $V_0 \subseteq V$ 的标签已知，目标是确定其余节点 $V \setminus V_0$ 的标签。目标函数为：

$$\min_{f:V \rightarrow P(N)} \frac{1}{m} \sum_{i=1}^m W_2^2(\mu_i, f(i)) + \gamma \sum_{(i,j) \in E} \omega_{ij} W_2^2(f(i), f(j))$$

其中 γ 是正则化参数, ω_{ij} 是边 (i, j) 的权重, W_2 是2-Wasserstein距离。

超图上的Wasserstein标签传播

在超图 $H = (V, E)$ 上, 由于超边 (hyperedge) 可以包含任意数量的顶点, 目标函数需要重新定义。超图上的Wasserstein传播算法基于消息传递更新规则:

1. 消息传递更新规则

从节点 i 向超边 E 发送消息 $J_{i \rightarrow E}$:

$$J_{i \rightarrow E}(b_i) = W_2^2(\mu_i, b_i) + \sum_{E' \in E \setminus \{E\}: i \in E'} J_{E' \rightarrow i}(b_i)$$

超边 E 向节点 i 发送消息 $J_{E \rightarrow i}$:

$$J_{E \rightarrow i}(b_i) = \min_{f_i \in P(N)} \left[\text{bar}(E) + \sum_{k \in E \setminus \{i\}} J_{k \rightarrow E}(f_k) \right]$$

$\text{bar}(E)$ 是超边 E 上的 Wasserstein 重心

2. 迭代更新信念

- 节点 i 的信念 b_i 更新为:

$$b_i = \arg \min_{f_i \in P(N)} \left[W_2^2(\mu_i, f_i) + \sum_{E \in E: i \in E} J_{E \rightarrow i}(f_i) \right]$$

3. 最终目标函数

- 对整个超图的目标函数为:

$$\min_{f:V \rightarrow P(N)} \frac{1}{m} \sum_{i=1}^m W_2^2(\mu_i, f(i)) + \gamma \sum_{E \in E} \text{bar}(E)$$

3. Wasserstein重心

Wasserstein重心的计算是核心步骤之一, 具体定义如下:

给定 k 个概率分布 ρ_1, \dots, ρ_k , 它们的Wasserstein重心为:

$$\text{bar}(\{\rho_i\}_{i=1}^k) = \inf_{\nu \in P(N)} \frac{1}{k} \sum_{i=1}^k W_2^2(\rho_i, \nu)$$

对于一维概率分布, 其Wasserstein重心可以通过累积分布函数的逆函数来计算:

$$F_b^{-1}(s) = \frac{1}{k} \sum_{i=1}^k F_{\rho_i}^{-1}(s)$$

Semi-supervised Hypergraph Node Classification on Hypergraph Line Expansion

一句话总结：用 线性扩展（LE）将超图转换为简单图结构

1. 线性扩展（Line Expansion, LE）

线性扩展（LE）方法通过以下步骤实现超图到简单图的转化：

- 顶点-超边对的构建：**对于每个超图 $G_H = (V, E)$ ，LE构建一个顶点-超边对的集合 (v, e) ，其中 $v \in V$ 且 $v \in e$ 。
- 节点和边的定义：**在扩展图 $G_l = (V_l, E_l)$ 中，节点集 V_l 由顶点-超边对构成。如果两个顶点-超边对共享同一个顶点或超边，则在扩展图中定义一条边。
- 邻域聚合：**在扩展图中，基于邻域聚合机制进行节点表示学习。每个节点的表示通过其邻域中的节点表示聚合得到，这些邻域包含共享相同顶点或超边的节点。

2. 邻域特征聚合

在扩展图上应用图卷积网络（GCN）进行特征聚合，具体步骤如下：

- 特征投影：**将超图顶点的初始特征矩阵 $X \in \mathbb{R}^{|V| \times d_i}$ 转换为扩展图的特征矩阵 $H(0) = P_{vertex} X \in \mathbb{R}^{|V_l| \times d_i}$ ，其中 P_{vertex} 是顶点投影矩阵。
- 图卷积层：**在扩展图上应用图卷积层进行邻域特征聚合，通过如下公式进行更新：

$$h_{(k+1)(v,e)} = \sigma \left(\sum_{e'} w_e h_{(k)(v,e')} \Theta_{(k)} + \sum_{v'} w_v h_{(k)(v',e)} \Theta_{(k)} \right)$$

其中， w_v 和 w_e 分别表示顶点相似性和超边相似性的权重， σ 是非线性激活函数（如ReLU）， $\Theta_{(k)}$ 是第 k 层的变换参数。

- 表示回投影：**在 K 层卷积后，得到扩展图的最终节点表示 $H(K)$ 。使用顶点回投影矩阵 P'_{vertex} 将表示回投影到超图顶点上：

$$Y = P'_{vertex} H(K) \in \mathbb{R}^{|V| \times d_o}$$

Self-Supervised Guided Hypergraph Feature Propagation for Semi-Supervised Classification with Missing Node Features

一句话总结： 用来解决只给一部分特征的任务。

SGHFP通过以下步骤进行特征传播和重建：

- **特征超图构建**：根据节点的已知特征构建特征超图。
- **伪标签超图构建**：利用前一迭代产生的重建特征通过两层GNN构建伪标签超图。
- **超图融合**：在每次迭代前，将特征超图和伪标签超图融合，生成更有效的超图。
- **特征传播**：使用融合后的超图进行特征传播，重建缺失的节点特征。
- **迭代优化**：通过多次迭代优化，最终重建的节点特征用于下游的半监督分类任务。

自监督引导的超图特征传播（SGHFP）

2. 超图融合

- **特征超图构建**：根据缺失信息的节点特征生成特征超图 G_f ，每次选择一个节点作为中心节点，将其所有邻居连接为一个超边。
- **伪标签超图构建**：利用前一迭代生成的重建特征，通过两层GNN构建伪标签超图 G_{pl} 。
- **融合超图生成**：特征超图和伪标签超图融合为一个稀疏矩阵，特征超图作为索引，伪标签超图作为对应值，生成融合超图。

3. 超图特征传播

- **特征传播模型**：使用融合后的超图进行特征传播，通过以下步骤迭代更新：

$$\dot{x}(t) = -\nabla \ell(x(t))$$

其中， $\ell(x, G)$ 是迪利克雷能量函数，用于最小化能量并重建未知节点特征。迭代更新公式如下：

$$x^{(k+1)} = \Theta x^{(k)}$$

每次迭代后，重置已知特征为其原始值。

4. 算法步骤

算法1: SGHFP

1. 输入：特征向量 x 、图结构 G 、训练轮数 T
2. 迭代 T 次：
 - 构建 G_f 和 G_{pl}
 - 融合 G_f 和 G_{pl} 生成传播矩阵 Θ
 - 传播特征 Θx
 - 重置已知特征 x_k

...

Nonlinear Feature Diffusion on Hypergraphs

一句话总结：提出一种非线性特征扩散方法，在超图上同时传播特征和标签，以捕捉更复杂的节点间关系。

核心算法：非线性特征扩散 (Nonlinear Feature Diffusion)

1. 问题定义

在一个超图 $H = (V, E)$ 上进行多类半监督分类问题，其中 V 是节点集合， E 是超边集合。每个超边 $e \in E$ 具有一个正权重 $w(e)$ 。节点的特征表示为矩阵 $X \in \mathbb{R}^{n \times d}$ ，其中每行 x_i 是节点 i 的特征向量。已知部分节点的标签，目标是预测其余节点的标签。

2. 非线性扩散过程

非线性扩散过程通过以下步骤进行：

- 初始化**：将节点的标签和特征合并形成初始嵌入矩阵 $U = [Y, X]$ ，其中 Y 是标签矩阵。
- 非线性扩散映射**：定义非线性扩散映射 L 为：

$$L(F) = D^{-1/2} K W \sigma(K^T \rho(D^{-1/2} F))$$

其中 σ 和 ρ 是非线性激活函数， K 是超图的关联矩阵， W 是超边权重矩阵。

- 迭代更新**：

$$F^{(k+1)} = \alpha L(F^{(k)}) + (1 - \alpha) U$$

其中 α 是正则化参数，控制标签传播和特征传播的权重。

- 归一化**：

在每次迭代后，对嵌入进行归一化处理，确保稳定性和收敛性。

3. 正则化损失函数

非线性扩散过程的极限点是一个新的正则化损失函数的全局最小值，该损失函数定义为：

$$\min_F \|F - U / \phi(U)\|^2 + \lambda \Omega_\mu(F)$$

其中， $\Omega_\mu(F)$ 是一种新的正则化项，旨在减少超边上的广义方差。

Sheaf Hypergraph Networks (重了, 你多下了一个)

一句话总结：通过在超图中引入纤维丛 (cellular sheaf) 结构，增强了超图的表示能力，从而更好地建模复杂数据结构。

引入纤维丛超图结构：纤维丛超图通过在超图的节点和超边上关联向量空间，并提供线性投影，使信息在节点和超边之间传递，从而实现更丰富的数据表示。

提出纤维丛超图拉普拉斯算子：包括线性和非线性两种形式，提供了更具表达力的归纳偏差，有效地建模复杂现象。

开发新型神经网络模型：设计了基于纤维丛超图拉普拉斯算子的SheafHyperGNN和SheafHyperGCN模型，实验结果表明，这些模型在多个基准数据集上的表现优于现有方法。

核心算法：纤维丛超图拉普拉斯算子

1. 线性纤维丛超图拉普拉斯算子

定义一个超图 $H = (V, E)$ ，其中 V 是节点集合， E 是超边集合。对于每个超图，定义纤维丛为三元组 $\langle F(v), F(e), Fv \triangleright e \rangle$ ，其中：

- $F(v)$ 是与每个节点关联的向量空间。
- $F(e)$ 是与每个超边关联的向量空间。
- $Fv \triangleright e$ 是每个节点-超边对之间的线性投影。

线性纤维丛超图拉普拉斯算子定义为：

$$(LF)_{vv} = \sum_{e; v \in e} \frac{1}{\delta_e} F_{v \triangleright e}^T F_{v \triangleright e}$$

$$(LF)_{uv} = - \sum_{e; u, v \in e} \frac{1}{\delta_e} F_{u \triangleright e}^T F_{v \triangleright e}$$

其中 δ_e 是超边的度，表示超边中节点的数量。

该算子应用于信号 $x \in \mathbb{R}^{n \times d}$ 时，可以重写为：

$$LF(x)_v = \sum_{e; v \in e} \frac{1}{\delta_e} F_{v \triangleright e} \left(\sum_{u \in e, u \neq v} (F_{v \triangleright e} x_v - F_{u \triangleright e} x_u) \right)$$

2. 非线性纤维丛超图拉普拉斯算子

为了更好地保留超图结构，引入非线性纤维丛超图拉普拉斯算子：

- 对每个超边 e ，计算特征差异最大的节点对 (u_e, v_e) ：

$$(u_e, v_e) = \arg \max_{u, v \in e} \|F_{u \triangleright e} x_u - F_{v \triangleright e} x_v\|$$

- 构建一个无向图 G_H ，其中节点集与超图相同，每个超边连接特征最不相似的节点对。
- 非线性纤维丛超图拉普拉斯算子定义为：

$$\bar{L}_F(x)_v = \sum_{e; u \sim_e v} \frac{1}{\delta_e} F_{v \triangleright e} (F_{v \triangleright e} x_v - F_{u \triangleright e} x_u)$$

3. 纤维丛超图神经网络 (SheafHyperGNN) 和纤维丛超图卷积网络 (SheafHyperGCN)

这些模型通过以下步骤实现：

1. 初始特征投影：将输入特征线性投影到新的特征空间。
2. 使用纤维丛拉普拉斯算子进行消息传递和特征更新。
3. 输出层将更新后的特征用于节点分类任务。

Hypergraph-enhanced Dual Semi-supervised Graph Classification

一句话总结: 提出了一种增强的双半监督图分类框架（HEAL），通过超图和线图两个视角捕捉图语义，从而更好地利用未标记图数据，提高分类性能。

HEAL框架包括三个主要模块：

- **高阶依赖学习模块**：通过自适应学习超图结构，捕捉节点间的复杂依赖关系。
- **线图卷积模块**：利用线图捕捉超边间的交互，从而挖掘更深层次的语义结构。
- **关系一致性学习模块**：促进两个分支间的知识传递，增强模型在未标记图上的表现。

核心算法：超图增强的双半监督图分类（HEAL）

1. 高阶依赖学习

通过学习超图结构，捕捉节点间的复杂依赖关系。具体步骤如下：

- **超图结构学习**：定义超图结构矩阵 Λ ，通过低秩策略高效建模：

$$\Lambda = H \cdot W$$

其中， H 是从GNN编码器得到的隐藏嵌入矩阵， W 是可学习的权重矩阵。

- **超图卷积**：计算超边嵌入 R 并更新节点表示 S ：

$$R = \sigma(U\Lambda^T H) + \Lambda^T H$$

$$S = \Lambda \cdot R$$

2. 线图卷积

利用超图结构构建线图，捕捉超边间的交互。具体步骤如下：

- **线图构建**：基于超图的邻接矩阵 Λ ，构建线图 $L(G)$ 。
- **线图卷积**：对线图进行卷积，获取其图级表示 t_G ：

$$T = \sigma(A_L R W)$$

$$t_G = \sum_i t_i$$

3. 关系一致性学习

促进两个分支间的知识传递，增强未标记图的表示能力。具体步骤如下：

- **锚图选择**：从标记集选择锚图，存储在内存库中。
- **相似度分布计算**：计算未标记图与锚图间的相似度分布：

$$P_u^m = \frac{\exp(\cos(s_u, s_m)/\tau)}{\sum_{m'} \exp(\cos(s_u, s'_m)/\tau)}$$

$$Q_u^m = \frac{\exp(\cos(t_u, t_m)/\tau)}{\sum_{m'} \exp(\cos(t_u, t'_m)/\tau)}$$

- **一致性损失**：通过最小化KL散度，鼓励两个分布间的一致性：

$$L_{con} = \frac{1}{|G_U|} \sum_{u \in G_U} \frac{1}{2} (D_{KL}(P_u \| Q_u) + D_{KL}(Q_u \| P_u))$$

4. 优化框架

结合监督损失 L_{sup} 和一致性损失 L_{con} ，优化总体损失：

$$L = L_{sup} + \beta \cdot L_{con}$$

Hypergraph Transformer for semi-supervised classification

一句话总结:利用Transformer架构有效地捕捉超图中的全局相关性，同时保留局部连接模式。

提出HyperGraph Transformer框架：通过Transformer架构结合超图特定组件，实现同时捕捉全局和局部结构信息，提升了超图表示学习的效果。

设计了新的位置编码机制和结构正则化方法：利用超图关联矩阵进行位置编码和结构正则化，有效地融合了超图的结构信息，提升了模型性能。

实验验证：在四个实际数据集上的实验结果表明，HyperGT在节点分类任务中显著优于现有的最先进方法，并展示了各个设计组件的有效性。

1. 超图关联矩阵位置编码

为了将结构信息融入输入特征中，使用超图关联矩阵 H 进行位置编码。对于节点和超边，定义位置编码如下：

$$P_V = HW_V, \quad P_E = H^T W_E$$

其中， W_V 和 W_E 是可学习的投影矩阵。初始嵌入为：

$$Z(0) = [X_V + P_V, X_E + P_E]$$

其中 X_V 和 X_E 分别为节点和超边的特征。

2. 超图注意力机制

超图注意力机制通过在节点和超边之间建立密集连接，捕捉全局信息。对于第 l 层的特征嵌入 $Z^{(l)}$ ，计算注意力矩阵 $A^{(l)}$ ：

$$A_{ij}^{(l)} = \frac{\exp((z_i^{(l)} W_Q^{(l)}) \cdot (z_j^{(l)} W_K^{(l)})^T)}{\sum_k \exp((z_i^{(l)} W_Q^{(l)}) \cdot (z_k^{(l)} W_K^{(l)})^T)}$$

然后更新每个实例 i 的表示：

$$z_i^{(l+1)} = \sum_j A_{ij}^{(l)} (z_j^{(l)} W_V^{(l)})$$

3. 超图结构正则化

利用超图的星形扩展结构，构建概率转移矩阵 P_s ，定义为：

$$P_s = D_s^{-1} A_s$$

其中 A_s 是星形扩展图的邻接矩阵， D_s 是度矩阵。结构正则化损失定义为：

$$\mathcal{L}_s = -\frac{1}{(n+m)L} \sum_{l=1}^L \sum_{(i,j) \in E_s} \frac{1}{d_i} \log A_{ij}^{(l)}$$

4. 训练策略

模型的最终损失函数结合了监督分类损失和结构正则化损失：

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_s$$

其中 λ 是平衡系数。



Hypergraph Label Propagation Network

一句话总结: 提出了一种超图标签传播网络（HLPN），将超图标签传播与深度神经网络结合，通过端到端架构优化特征嵌入，从而实现更高效的数据标注。

提出了一种新的超图半监督学习框架: 该框架通过稀疏样本捕捉流形结构，并实现端到端学习。

提高超图学习效率: 相比传统超图方法，HLPN不需要构建整个数据集的超图，从而提高了计算效率。

实验验证: 在实际数据集上的实验结果表明，HLPN在性能上显著优于最先进的方法。

3. 标签传播

使用正则化框架在超图上进行标签传播，目标函数定义为：

$$\arg \min_F \{ \lambda R_{\text{emp}}(F) + \Omega(F) \}$$

其中， $R_{\text{emp}}(F)$ 是经验损失，定义为：

$$R_{\text{emp}}(F) = \|F - Y\|_F^2$$

超图结构上的正则化项 $\Omega(F)$ 定义为：

$$\Omega(F) = \frac{1}{2} \sum_{e \in E} \sum_{u, v \in V} \sum_{k=1}^C w(e) H(u, e) H(v, e) \frac{(F(u, k) / \sqrt{d(u)} - F(v, k) / \sqrt{d(v)})^2}{\delta(e)}$$

目标函数可以重写为：

$$\arg \min_F \{ \text{tr}(F^T \Delta F) + \lambda \|F - Y\|_F^2 \}$$

其解为：

$$F = \left(I + \frac{1}{\lambda} (I - D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}) \right)^{-1} Y$$

4. 端到端架构

HLPN通过深度神经网络进行特征嵌入，将输入数据映射到特征空间，然后构建超图并进行标签传播。通过联合优化特征嵌入网络和超图标签传播模块，实现高效的半监督学习。

Hypergraph Convolutional Network based Weakly Supervised Point Cloud Semantic Segmentation with Scene-Level Annotations

一句话总结: 提出了一种基于加权超图卷积网络（WHCN）的弱监督点云语义分割方法，通过场景级别标注学习点级别伪标签，从而提高分割性能。

算法设计：

- **超点生成与超图构建：**利用几何同质划分生成超点，以平衡不同类别点数的不均衡性，并降低模型复杂度。基于高置信度超点种子构建超图。
- **加权超图卷积网络：**设计了加权超图卷积网络（WHCN），通过标签传播生成高精度的点级别伪标签。WHCN包括谱超图卷积块和超边注意力模块，调整超边权重以优化标签传播。
- **伪标签生成与分割网络训练：**利用生成的伪标签训练分割网络，实现高精度的点云语义分割。

核心算法：加权超图卷积网络（WHCN）

1. 超点生成与超图构建

1. 超点生成：

- 利用全局能量模型生成超点（superpoints），将点云划分为几何同质区域。
- 通过最小化以下能量函数生成超点：

$$g' = \arg \min_g \sum_{i \in V} \|g_i - f_i\|^2 + \rho \sum_{(i,j) \in E} \delta(g_i - g_j \neq 0)$$

- 采用贪心图割算法（l0-cut pursuit）进行优化，生成超点集合 $S = \{s_1, s_2, \dots, s_N\}$ 。

2. 超图构建：

- 将生成的超点作为超图顶点集 V ，根据高置信度超点种子构建超边。
- 定义超图 $G = (V, \epsilon)$ ，其中 ϵ 表示超边集，每个超边 e 分配一个正权重 $w(e)$ 。

2. 伪标签生成

1. 超点种子标签生成：

- 利用类激活图（CAM）生成初始种子标签，并选择高置信度的超点级种子标签作为最终种子标签。

2. 标签传播：

- 设计加权超图卷积网络（WHCN），通过标签传播生成点级别伪标签。
- WHCN包括谱超图卷积层和超边注意力模块，利用超图结构进行高阶特征聚合。

3. 加权超图卷积网络

1. 超边注意力模块：

- 超边权重描述了每个超边的亲和属性，权重越大，表示超边中顶点之间的关系越强。
- 定义超边权重：

$$w(e) = \frac{1}{R(R-1)} \sum_{\{v_i, v_j\} \in e} \exp \left(-\frac{\sigma(\text{sim}(x_i \Theta, x_j \Theta))}{\mu} \right)$$

- 其中， $\sigma(\cdot)$ 是LeakyReLU非线性函数， $\text{sim}(\cdot)$ 是计算顶点对之间相似度的函数。

2. 谱超图卷积：

- 定义归一化超图拉普拉斯矩阵：

$$\Delta = I - D^{-1/2} H W B^{-1} H^T D^{-1/2}$$

- 利用切比雪夫多项式参数化卷积，简化计算：

$$g * x \approx \left(\sum_{q=0}^Q \theta_q T_q(\Delta) \right) x$$

- 每层传播规则定义为：

$$X^{(l+1)} = \sigma(D^{-1/2} H W B^{-1} H^T D^{-1/2} X^{(l)} \Theta^{(l)})$$

4. 损失函数

通过最小化交叉熵损失函数训练模型：

$$L = - \sum_{i=1}^T \sum_{j=1}^C Y_{ij} \log(p_{ij})$$

From Hypergraph Energy Functions to Hypergraph Neural Networks

这个你看了, 就不总结了

CHGNN: A Semi-Supervised Contrastive Hypergraph Learning Network

一句话总结: 提出了一种对比超图神经网络（CHGNN），结合自监督对比学习技术，从标记和未标记数据中学习。CHGNN包括一个自适应超图视图生成器、改进的超图编码器和一个结合相似度损失、节点分类损失和超边同质性损失的联合损失函数。

算法设计:

- **自适应超图视图生成器**: 采用自动增强策略，学习最小充分视图的扰动概率分布。
- **改进的超图编码器**: 考虑超边的同质性，有效融合信息。
- **联合损失函数**: 结合视图生成器的相似度损失、节点分类损失和超边同质性损失，增强监督信号。此外，还包括基础对比损失和交叉验证对比损失，提高对比学习效果。

核心算法：对比超图神经网络（CHGNN）

1. 自适应超图视图生成器

利用三种操作（保留超边、移除超边和掩码超边中的部分节点）扰动超图 G ，生成两个超图视图 $G1$ 和 $G2$ 。通过Gumbel-softmax从这些操作的概率中采样，并根据节点重要性进行掩码。

2. 超边同质性感知编码

基于超边的同质性进行聚合，赋予超边更高的权重:

$$h_v^{(l)} = \sigma \left(W^{(l)} \left(h_v^{(l-1)} + \sum_{e \in \mathcal{E}(v)} \text{homo}(e) \cdot h_e^{(l)} \right) \right)$$

其中, $h_v^{(l)}$ 是节点 v 在第 l 层的嵌入, $h_e^{(l)}$ 是超边 e 在第 l 层的嵌入, $\text{homo}(e)$ 是超边同质性。

3. 联合损失函数

结合以下损失函数:

- 分类损失 L_{cls} :

$$L_{cls} = - \sum_{v \in \mathcal{V}_L} y_v \log C(h_v)$$

其中, y_v 是节点 v 的真实标签, C 是分类器。

- 超边同质性损失 L_h :

$$L_h = \lambda_h \cdot \frac{1}{2} \sum_{i=1}^2 \text{MSE}(R(H_e^{(i)}), v_{\text{homo}})$$

- 基础对比损失 L_{cl} 和 交叉验证对比损失 L_{crocl} :

$$L_{cl} = \lambda_c L_c + \lambda_e L_e$$

$$L_{crocl} = \lambda_{nc} L_{nc} + \lambda_{ne} L_{ne} + \lambda_{ec} L_{ec}$$

- 相似度损失 L_{sim} :

$$L_{sim} = \text{sim}(v_{aug}^1, v_{aug}^2)$$

总损失函数:

$$L = L_{sim} + L_{csc} + L_h + L_{cl} + L_{crocl}$$

Hypergraph Dynamic System

一句话总结: 出了一种超图动态系统 (HDS)，将超图和动态系统联系起来，描述表示的连续动态演化过程。提出的控制扩散超图动态系统通过常微分方程 (ODE) 实现，设计了一个多层HDSode作为神经实现，包含控制步骤和扩散步骤。

看起来解决了深层的 HGNN 模型的问题?

算法设计:

- **超图动态系统**: 基于ODE，引入控制和扩散函数，形成超图动态系统模型。
- **HDSode框架**: 设计了一个多层HDSode框架，实现可控且稳定的超图表示学习。
- **稳定性分析**: 证明了HDSode的稳定性，并展示了其在捕捉长距离顶点关系方面的能力。

核心算法：HDSode (Hypergraph Dynamic System ODE)

1. 超图动态系统定义

定义超图动态系统为：

$$\begin{bmatrix} \dot{X}_v \\ \dot{X}_e \end{bmatrix} = f \left(\begin{bmatrix} X_v(t) \\ X_e(t) \end{bmatrix} \right)$$

其中， $X_v(t)$ 和 $X_e(t)$ 分别表示时间 t 的顶点表示和超边表示。

2. 基于ODE的超图动态系统

引入控制和扩散函数，形成ODE模型：

$$\begin{bmatrix} \dot{X}_v \\ \dot{X}_e \end{bmatrix} = \begin{bmatrix} g_v(X_v(t)) \\ g_e(X_e(t)) \end{bmatrix} + A \begin{bmatrix} X_v(t) \\ X_e(t) \end{bmatrix}$$

其中， g_v 和 g_e 是控制函数， A 是扩散矩阵，描述顶点和超边间的扩散速度。

3. HDSode的神经实现

利用Lie-Trotter分裂法对ODE进行离散化，分为控制步骤和扩散步骤：

$$\begin{bmatrix} X_v(t + \frac{1}{2}) \\ X_e(t + \frac{1}{2}) \end{bmatrix} = \begin{bmatrix} X_v(t) \\ X_e(t) \end{bmatrix} + \begin{bmatrix} g_v(X_v(t)) \\ g_e(X_e(t)) \end{bmatrix}$$

$$\begin{bmatrix} X_v(t + 1) \\ X_e(t + 1) \end{bmatrix} = \begin{bmatrix} X_v(t + \frac{1}{2}) \\ X_e(t + \frac{1}{2}) \end{bmatrix} + A \begin{bmatrix} X_v(t + \frac{1}{2}) \\ X_e(t + \frac{1}{2}) \end{bmatrix}$$

扩散矩阵 A 的设计确保表示的稳定性：

$$A = \begin{bmatrix} -\alpha_v I & \alpha_v D_v^{-1} H \\ \alpha_e D_e^{-1} H^T & -\alpha_e I \end{bmatrix}$$



VilLain: Self-Supervised Learning on Hypergraphs without Features via Virtual Label Propagation

一句话总结: 提出了一种新的自监督超图表示学习方法VilLain，基于虚拟标签（v-labels）的传播

算法设计:

- **虚拟标签生成**: 假设存在 d 个虚拟标签，并为每个节点分配一个稀疏的v-label概率分布。
 - **超图上的v-label传播**: 通过交替在节点和超边之间传播v-label，来获得节点和超边的v-label分配矩阵。
 - **多v-label传播**: 为了捕捉复杂的结构-标签模式，将嵌入空间划分为多个子空间，分别进行v-label传播，并最终连接各子空间的输出作为节点嵌入。
1. **自监督损失函数设计**: 结合局部损失和全局损失，通过最大化节点和超边v-label分配向量的熵，确保标签的同质性和全局分布的均衡性和区分性。

核心算法: VilLain（基于虚拟标签传播的自监督超图表示学习）

1. 虚拟标签生成

给定超图 $G = (V, E)$ ，假设存在 d 个虚拟标签。每个节点 v_i 的初始v-label概率分布 $X_i(0)$ 通过以下方式计算:

$$X_i(0) = \text{softmax}(X_i + g)$$

其中, X_i 是可学习的特征向量, g 是随机噪声, 鼓励向量偏向单个标签。

2. v-label传播

通过交替在节点和超边之间传播v-label, 更新v-label分配矩阵。具体步骤如下:

$$Y^{(l)} = D_E^{-1} H^T X^{(l-1)}$$

$$X^{(l)} = D_V^{-1} H Y^{(l)}$$

其中, D_V 和 D_E 分别是节点和超边的度矩阵, H 是关联矩阵。

3. 多v-label传播

将嵌入空间划分为多个子空间, 每个子空间内进行独立的v-label传播, 最终连接各子空间的输出:

$$Z_i^* = [Z_i^{(1)} \| Z_i^{(2)} \| \dots \| Z_i^{(D)}]$$

其中， $\|$ 表示连接操作， $Z_i^{(j)}$ 是第 j 个子空间的节点 v_i 嵌入。

4. 自监督损失函数

结合局部损失和全局损失，优化v-label分配向量：

- **局部损失**：最小化节点和超边v-label分配向量的熵，确保同质性：

$$L_{local} = \sum_{l=1}^k \left(\frac{1}{|V|} \sum_{i=1}^{|V|} H(X_i^{(l)}) + \frac{1}{|E|} \sum_{i=1}^{|E|} H(Y_i^{(l)}) \right)$$

其中， $H(p)$ 是熵函数。

- **全局损失**：最大化v-label的全局分布熵，防止所有节点和超边被分配到同一标签，并增强标签的区分性：

$$L_{global} = J_{cls} + J_{dst}$$

其中， J_{cls} 和 J_{dst} 分别为全局分类损失和区分性损失。

5. 目标函数

最终的目标函数为局部损失和全局损失的加权和：

$$L = L_{local} + L_{global}$$

接下来的scope 不是特别符合

Multi-Task Hypergraphs for Semi-supervised Learning using Earth Observations

背景和动机：在地球观测（Earth Observation）任务中，通常需要多任务处理不同的观测层数据。然而，由于传感器故障等原因，观测数据常常不完整。这一问题迫切需要一种能够在数据缺失情况下仍能有效学习的半监督学习方法。本文提出了一种强大的多任务超图（Multi-Task Hypergraph），通过利用不同任务之间的高阶依赖关系，在超图上实现多任务半监督学习。每个节点表示一个任务，通过不同路径形成的超边作为无监督教师，生成任务的可靠伪标签。

多任务超图结构：每个任务作为超图的节点，不同路径通过超图中的超边形成无监督教师。这些超边既作为教师也作为学生，通过自监督的方式进行学习。

超边集成模型：通过集成多个路径生成伪标签，提升伪标签的可靠性和鲁棒性。

自监督学习过程：利用NASA NEO数据集，进行大量实验验证多任务半监督学习方法的有效性。实验结果表明，该方法在数据分布逐渐变化的情况下，能够可靠地恢复多达七年的缺失数据。

主要贡献：

- 提出多任务超图框架：**通过高阶任务依赖关系，实现了一个集成的、鲁棒的半监督学习方法。
- 设计了超边集成模型：**利用多个路径生成伪标签，提升了伪标签的可靠性和模型的鲁棒性。
- 实验验证：**在NASA NEO数据集上的实验表明，本文方法在多个基准任务上均取得了显著的性能提升，并能够适应数据分布的逐渐变化。

Efficient and Effective Attributed Hypergraph Clustering via \boxtimes -Nearest Neighbor Augmentation

Efficient and Effective Attributed Hypergraph Clustering via \boxtimes -Nearest Neighbor Augmentation

属性超图聚类（Attributed Hypergraph Clustering, AHC）旨在将属性超图中的节点划分为k个不相交的簇，使得同一簇内的节点在结构上紧密连接，并且属性相似，而不同簇的节点则属性差异显著。现有的AHC方法在处理大规模属性超图时面临计算成本高、聚类质量不佳等问题。

算法设计：本文提出了一种高效的属性超图聚类方法AHCKA（Attributed Hypergraph Clustering via K-nearest neighbor Augmentation），通过几个关键算法设计实现了高效的AHC：

- K-最近邻增强策略：**利用属性信息优化超图结构，通过引入KNN图来构建额外的连接。
- 联合随机游走模型：**设计了一个联合随机游走模型，以优化AHC目标。
- 高效求解器：**利用矩阵运算和贪心迭代框架，大幅提升计算效率。

主要贡献：

- 提出了KNN增强策略：**通过KNN图构建额外连接，有效利用属性信息提升超图结构的聚类质量。
- 设计了联合随机游走模型：**结合超图和KNN图的高阶关系，以优化AHC的目标函数。
- 实现了高效求解器：**通过矩阵运算和贪心迭代框架，显著提升了AHC的计算效率。

核心算法：AHCKA (Attributed Hypergraph Clustering via K-nearest neighbor Augmentation)

1. K-最近邻增强策略

- 属性相似度计算：定义属性相似度函数 $f(v_i, v_j) = \text{cosine}(X[i], X[j])$ 。
- KNN图构建：对于每个节点 v_i ，选择K个最相似的节点作为其邻居，构建KNN图 G_K 。
- 邻接矩阵定义：KNN图的邻接矩阵 A_K 定义如下：

$$A_K[i, j] = \begin{cases} 2 \cdot f(X[i], X[j]), & \text{if } v_i \in N_K(v_j) \text{ and } v_j \in N_K(v_i) \\ f(X[i], X[j]), & \text{otherwise} \end{cases}$$

- 增强超图构建：将原始超图 H_O 和KNN图 G_K 结合，构建增强超图 H_A 。

2. 联合随机游走模型

- 随机游走定义：定义 (α, β, γ) -随机游走模型，以联合超图和KNN图中的多跳关系：

$$s(v_i, v_j) = \alpha \sum_{\ell=0}^{\gamma} (1 - \alpha)^{\ell} P^{\ell}[i, j]$$

其中，过渡矩阵 P 定义为：

$$P = (I - B) \cdot D_V^{-1} A^T D_E^{-1} A + B D_K^{-1} A_K$$

- 目标函数：最小化多跳导电率 $\phi(C)$ ：

$$\phi(C) = \frac{1}{|C|} \sum_{v_i \in C} \sum_{v_j \notin C} s(v_i, v_j)$$

3. 高效求解器

- 矩阵迹优化问题：将AHC问题转化为矩阵迹优化问题，通过求解前k+1个特征向量进行近似求解：

$$\Psi(Y) = \frac{1}{k} \text{trace}(Y^T S Y)$$

- 贪心迭代框架：通过正交迭代法近似求解特征向量，并使用离散化算法更新簇分配矩阵 Y 。

总结

本文提出的AHCKA方法通过KNN增强策略、联合随机游走模型和高效求解器，实现了高效的属性超图聚类。在多个实际数据集上的实验验证了其在聚类质量和计算效率上的优势，为大规模属性超图的聚

类问题提供了一种有效的解决方案。未来工作将进一步优化KNN增强策略和随机游走模型，提高模型的适用性和鲁棒性。

MEGA: Multi-View Semi-Supervised Clustering of Hypergraphs

提出了一种多视图半监督超图聚类方法MEGA（Multi-view sEmi-supervised hyperGrAph clustering），结合实体的多种特征和部分已知标签信息，提升聚类性能。

背景和动机：在现实世界中，超图模型可以有效地表示复杂的实体间关系。传统的超图聚类方法主要依赖超图的连接结构，忽略了实体的其他属性和多视图关系。本文提出了一种多视图半监督超图聚类方法MEGA（Multi-view sEmi-supervised hyperGrAph clustering），结合实体的多种特征和部分已知标签信息，提升聚类性能。

算法设计：MEGA通过以下步骤实现多视图半监督聚类：

- 超图标准化切割与加权核K均值：**证明超图标准化切割目标与加权核K均值目标的数学等价性，并基于此开发了一种高效的多级超图聚类算法hGraclus，为MEGA提供良好的初始化。
- 多视图聚类目标函数：**将超图结构、实体的多种特征和关系以及部分已知标签信息整合到统一的非负矩阵分解（NMF）框架中，定义多视图聚类目标函数。
- 半监督聚类扩展：**通过部分已知标签和成对约束，将多视图聚类目标函数扩展为半监督聚类框架。

核心算法：MEGA（多视图半监督超图聚类）

1. 超图标准化切割与加权核K均值

超图标准化切割目标定义为：

$$hNCut(G) := \min_{V_1, \dots, V_k} \sum_{i=1}^k \frac{hCut(V_i)}{vol(V_i)}$$

其中， $hCut(V_i)$ 是超边的切割值， $vol(V_i)$ 是节点的度量和。

将其转化为加权核K均值目标，通过定义合适的核矩阵和权重矩阵，使得两者在数学上等价。

2. 多视图聚类目标函数

定义多个视图特征矩阵 X_i 和关系矩阵 S_j 的非负矩阵分解（NMF）目标函数：

$$\min_{W_i, H, \hat{H}_j} \sum_{i=1}^p \alpha_i \|X_i - W_i H\|_F^2 + \sum_{j=1}^q \beta_j \|S_j - \hat{H}_j^T H\|_F^2 + \sum_{j=1}^q \gamma_j \|\hat{H}_j - H\|_F^2$$

其中， H 是共享的聚类分配矩阵， W_i 和 \hat{H}_j 是视图特定的矩阵。

3. 半监督扩展

利用部分已知标签矩阵 P 和成对约束矩阵 S ，扩展目标函数：

$$\min_{W, H, \hat{H}_j} \sum_{i=1}^p \alpha_i \|X_i - W_i H\|_F^2 + \sum_{j=1}^q \beta_j \|S_j - \hat{H}_j^T H\|_F^2 + \sum_{j=1}^q \gamma_j \|\hat{H}_j - H\|_F^2 + \|M \circ (P - V$$

其中， M 是掩码矩阵，表示标签是否已知。

目标函数：

$$\min_{W, H, \hat{H}_j} \sum_{i=1}^p \alpha_i \|X_i - W_i H\|_F^2 + \sum_{j=1}^q \beta_j \|S_j - \hat{H}_j^T H\|_F^2 + \sum_{j=1}^q \gamma_j \|\hat{H}_j - H\|_F^2 + \|M \circ (P - VWH)\|_F^2$$