GRAPH INFERENCE LEARNING FOR SEMI-SUPERVISED CLASSIFICATION

一篇通过子图结构和特征信息以及路径可达性进行节点分类的工作，大概看了方法部分，有空再来补全。

Method

1. **问题定义**

   一个图的基本要素。。。跳过了

   阐述当前的一些用于半监督节点分类的模型（GCN，【Zhou et al., 2004】）忽略了从已知的标记节点到未标记节点提取可转移的知识，因为**图结构本身意味着节点的连通性**；而且现有的这些半监督学习的节点分类模型由于标签样本的稀缺性，分类器的性能通常不理想。

   为了解决以上问题，这篇工作提出了一种**元学习**的机制来结合**图结构**、**节点间的可达性**和**节点的特征/属性**来学习在图上推断节点的标签。

2. **结构关系**

   将训练集中带标注的节点看作参考节点，且他们的信息能够传递给未标记节点来提高标签预测的准确性。这种传递性用节点间的相似度来做表示。如给定一个参考节点$v_i \in \mathcal{V}_{Label}$，查询节点$v_j$与$v_i$之间的相似度定义如下：

   $$s_{i \to j} = f_r\left(f_e\left(\mathcal{G}_{v_i}\right), f_e\left(\mathcal{G}_{v_j}\right), f_{\mathcal{P}}\left(v_i, v_j, \mathcal{E}\right)\right) \tag{1}$$

   G_{v_i}和G_{v_j}即以节点v_i与v_j为中心的子图，其中$f_e, f_r, f_{\mathcal{P}}$三个计算函数做了如下定义：

   - Node representation（节点特征向量）——$f_e\left(\mathcal{G}_{v_i}\right) \in \mathbb{R}^{d_v}$

     也就是通过G_{v_i}和G_{v_j}来学习节点v_i与v_j特征表示。这里文章认为不仅要考虑节点的信号（特征？），同时也要考虑子图的拓扑结构，结合起来考虑两个节点的相似性，所以用了谱图卷积来学习更有区分度的节点特征。

   - Path reachability（路径可达性）——$f_{\mathcal{P}}\left(v_i, v_j, \mathcal{E}\right) \in \mathbb{R}^{d_p}$

     通过两个节点之间不同长度的游走可达概率来表示它们的路径可达特征。

   - Structure relation（结构关系）——$f_r\left(\mathbb{R}^{d_v}, \mathbb{R}^{d_v}, \mathbb{R}^{d_p}\right) \in \mathbb{R}$

     相似度计算函数。文章说这个函数中两个节点的先后顺序不能更改，因为可达路径的不对称性，这个应该是针对有向图，无向图的可达路径双向应该是一样的。不是很理解：【If necessary, we may easily revise it as a symmetry function, e.g., summarizing two traversal directions.】

论文上部分讲了需要哪些东西来算参考节点和查询节点之间的参考度，然后将节点分类方案定为以所有训练集节点作为参考节点，验证集节点作为查询节点。并进一步利用式（1）定义了查询节点v_j对于某一个类c的相似度得分计算函数：

$$s_{\mathcal{C}_c \to j} = \phi_r\left(F_{\mathcal{C}_c \to v_j} \sum_{v_i \in \mathcal{C}_c} w_{i \to j} \cdot f_e\left(\mathcal{G}_{v_i}\right), f_e\left(\mathcal{G}_{v_j}\right)\right) \tag{2}$$

$$\text{s.t.} \quad w_{i \to j} = \phi_w\left(f_{\mathcal{P}}\left(v_i, v_j, \mathcal{E}\right)\right)$$

其中ϕ_w是将路径可达的特征向量映射为一个权重值的函数，$F_{\mathcal{C}_c \to v_j}$则为基于这些这些权重值的归一化因子$F_{\mathcal{C}_c \to v_j} = \frac{1}{\sum_{v_i \in \mathcal{C}_c} w_{i \to j}}$。$\phi_r$则是计算查询节点$v_j$和目标类$c$相似度的函数。

从上式中不难理解，对于给定查询节点v_j，它对于某一类c的相似度（其实也就是属于类c的概率）是由属于类c的所有训练集节点的节点特征通过根据路径可达性进行加权平均后，再与自身的节点特征进行相似度计算。

3. **推理学习**

根据上面式(2)的计算，我们能够得到查询节点对于每个类的相似度:

$$\mathbf{s}_{\mathcal{C}\to j}=\left[s_{\mathcal{C}_1\to j},\cdots,s_{\mathcal{C}_C\to j}\right]^\top\in\mathbb{R}^C$$

其中相似度最高的类也就是该查询节点的预测结果，文中选择用交叉熵定义损失函数：

$$\mathcal{L}=-\sum_{v_j}\sum_{c=1}^{C}y_{j,c}\log\hat{y}_{\mathcal{C}_c\to j}$$

然后文中提到了两种参数更新的策略，一种是基于训练集的梯度下降的参数更新：

$$\Theta'=\Theta-\alpha\nabla_\Theta\mathcal{L}_{tr}(\Theta)\tag{3}$$

另一种是再对训练集进行几次迭代更新后针对验证集的元优化：

$$\Theta=\Theta-\beta\nabla_\Theta\mathcal{L}_{val}\left(\Theta'\right)$$

这部分提到了两个细节：1.训练集迭代过程中参考节点和查询节点都出自训练集，也就是说当查询节点在计算类相似度时算到了自己，那么这部分相似度权重设置为0；2.在训练过程中，对训练节点和测试节点进行批次采样（具体不太清楚怎么操作），在测试过程中采用所有训练节点通过式（3）进行更新，并用来计算式（2）。（这边看的挺懵的，没太看懂，先看看下面细节部分会不会有更多信息）

细节

1. 节点表示——$f_e\left(\mathcal{G}_{v_i}\right)$

文中节点的局部特征表示是通过谱卷积实现的，用的是 Defferrard(2016) 通过切比雪夫多项式近似的谱图卷积方法，等看了论文再把这块补上。

2. 路径可达性——$f_{\mathcal{P}}\left(v_i,v_j,\mathcal{E}\right)$

文中使用了概率转移矩阵$\mathbf{P}=\mathcal{D}^{-1}\mathcal{E}$，$\mathcal{D}$ 是度值对角矩阵，前式也就是每一行除以对应行度值，使每一行和为1，符合概率运算。

$$P_{ij}^t=\begin{cases}P_{ij} & \text{if } t=1\\ \sum_h P_{ih}P_{h,j}^{t-1} & \text{if } t>1\end{cases}$$

通过计算不同游走距离所能达到目标节点的概率拼成一个向量：

$$f_{\mathcal{P}}\left(v_i,v_j,\mathcal{E}\right)=\left[P_{ij},P_{ij}^2,\ldots,P_{ij}^{d_p}\right]$$

d_p 也就是两个节点之间的最长路径长度。

3. 类与节点的关系

这部分主要交代了在计算查询节点 v_j 和一个类的相似度时，加权并标准化后的 $F_{\mathcal{C}_c\to v_j}\sum_{v_i\in\mathcal{C}_c}w_{i\to j}$ 最终和 $f_e\left(\mathcal{G}_{v_j}\right)$ 拼接成一个向量并通过全连接层获取得分。

实验部分

实验部分没怎么细看，作者在一些常见的Baseline对比后，关于利用验证集数据和GCN做了对比，确实用和不用，该方法的精度都优于GCN。但是GIL在cora只用训练集精度为83.3，而DGCN能够到达83.5，这应该是同等条件下的，所以为啥后续对比不是和DGCN比较?