

首先简单介绍一下有关图神经网络的基本内容。

图神经网络 (Graph Neural Network, GNN) 是一种深度学习模型, 专门用于处理图结构数据。在许多领域, 如社交网络、生物信息学、推荐系统等, 数据自然地以图的形式存在, 节点代表实体, 边代表实体之间的关系。GNN 旨在捕捉这种结构信息, 以进行有效的特征学习和预测。

而在本次任务中, 不同公司的相关属性, 风险传导路径天然的以图的形式存在, 因此, 为了更好的捕获不同尺度下的风险信息, 我们选择以图神经网络的形式, 而不是传统的手工设计特征, 来对不同公司风险等级进行分类。在图神经网络中, 所有的数据都以图的形式进行组织与输入。

具体而言, 本次任务的可以被建模为:

集合 $C=\{C_1, C_2, C_3, C_4 \dots C_n\}$ 代表着不同公司的集合。 $\forall C_i, i \in \{1, 2, \dots, n\}$, 都有相应的图属性, 分别是股权关系图 C_i^1 , 担保图 C_i^2 , 以及关联交易图 C_i^3 。

对于每张图的定义: $C_i^j = \langle V_j^i, E_j^i \rangle, j \in \{1, 2, 3\}$, 其中 V_j^i 代表着图中的节点集合, E_j^i 代表着图中的边的定义。对于其语义信息, 以担保图为例, 节点就代表着与中心企业 c 有着直接担保关系的企业, 而边也就代表着企业间的担保关系。

每张图都会有相应的点属性与边属性, 即 $H \in R^{N \times d_1}$ 和 $W \in R^{M \times d_2}$, 其中 N 是图中的节点数, d_1 是每个节点的特征维度; M 是图中边的总数, d_2 是边的特征维度。

对于一家特定的企业 C_k 以及给定的其三张图作为输入, 我们需要对该公司的信用评级以及未来趋势做出预测。

基于此, 我们最终选择了以图卷积神经网络 (GCN) 为基本骨架的模型。

图卷积神经网络通过在图的节点之间传播信息来学习节点的表示。GCN 的核心思想是使用图的拓扑结构来指导信息的传播, 从而使每个节点能够聚合来自其邻居的信息。这种方法允许模型捕捉到图中的局部结构特征, 提高了数据分析的准确性和效率。

在图神经网络中, 消息传播和聚合是两个关键的过程, 它们共同决定了如何从图的结构和节点特征中提取有效信息。

消息传播与聚合

消息传播是图神经网络中的一个核心过程, 它涉及到节点之间信息的交换。具体来说, 每个节点会根据其邻居节点的特征和边的信息来更新自己的特征表示。这个过程可以用以下公式表示:

$$m_v^{(l+1)} = \sum_{u \in N(v)} M^{(l)}(h_v^{(l)}, h_u^{(l)}, e_{uv})$$

其中:

- $m_v^{(l+1)}$ 是节点 v 在第 $l+1$ 层接收到的消息。
- $N(v)$ 是节点 v 的邻居节点集合。

- $M^{(l)}$ 是消息函数，用于计算在第 l 层节点 v 和其邻居节点 u 之间传递的消息。
- Σ 是聚合函数，其可以表示任何函数，不一定是求和。
- $h_v^{(l)}$ 和 $h_u^{(l)}$ 分别是节点 v 和 u 在第 l 层的特征表示。
- e_{uv} 是连接节点 v 和 u 的边的特征。

更新

在收到来自邻居节点的消息后，每个节点需要对这些消息进行聚合，以更新自己的特征表示。这个聚合过程可以用以下公式表示：

$$h_v^{(l+1)} = U^{(l)}(h_v^{(l)}, m_v^{(l+1)})$$

其中：

- $h_v^{(l+1)}$ 是节点 v 在第 $l+1$ 层的更新后特征表示。
- $U^{(l)}$ 是更新函数，用于根据节点 v 在第 l 层的特征表示和接收到的消息 $m_v^{(l+1)}$ 来计算更新后的特征表示。

一、模型总体设计

我们选用了 2019 年在图分类竞赛中拿到冠军的 HGP-SL 模型([cszhangzhen/HGP-SL: Hierarchical Graph Pooling with Structure Learning \(github.com\)](https://github.com/cszhangzhen/HGP-SL))作为基础模型，并对其进行一些结构上的修改以更好地适应其在金融领域上的任务。

首先介绍一下 HGP-SL 模型：

以往的 GCN 模型都只是借鉴了 CNN 模型中的卷积的思想，但很少有模型将 CNN 模型中另一个基石—池化层运用在图卷积神经网络中。在 CNN 中池化层可以很好的保留局部信息密度，赋予感受野更灵活的变化，因此带有池化层的模型往往会有更好的泛化性能。

因此 HGP-SL 借鉴 CNN 在池化层上的思想，设计了一套可以很好地利用到图数据上的‘池化层’。其架构如下图所示：

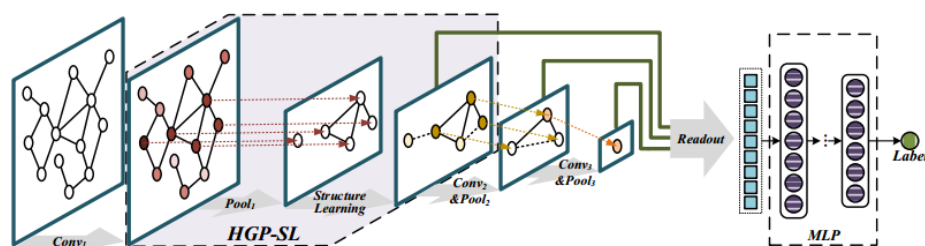


图 1.HGP-SL 架构图

其中图卷积层就是一中介绍的消息传播聚合与更新。接下来我们会具体解释池化层，也就是图 1 中的 POOL 以及 Structure Learning

Pool:

该模型的池化的基本设计思想是在保留原图信息的同时尽可能地去冗余的节点。如果一个节点的表示可以通过它的邻域表示重建,这意味着该节点可以在池图中删除,几乎没有信息损失,也就意味着该节点是冗余的。这里,该模型将节点信息得分正式定义为节点表示本身与由其邻居构造的节点表示之间的曼哈顿距离

$$p = \gamma(G_i) = | (I_i^k - (D_i^k)^{-1} A_i^k) H_i^k |$$

其中 $A_i^k \in R^{n_i^k \times m_i^k}$ 和 $H_i^k \in R^{n_i^k \times d}$ 是相邻矩阵和节点特征矩阵。 D_i^k 表示 A_i^k 的对角矩阵,

I_i^k 是单位矩阵。因此,我们用 $p \in R^{n_i^k}$ 对图中每个节点的信息评分进行编码。节点信息得分相对较大的节点将在池图的构建中被保留,因为它们可以提供更多的信息。

Structure Learning:

由于池化操作可能导致在诱导子图中断开高度相关的节点,这将失去图结构信息的完整性,并进一步阻碍消息传递过程,所以在池化后会进行结构学习,将子图中原本不存在的边但高度相关的节点进行连接。

具体来说,作者利用由权重向量 $\vec{a} \in R^{1 \times 2d}$ 参数化的单层神经网络,然后,通过注意力机制计算的节点 v_p 和 v_q 之间的相似度分数。可以表示为:

$$E_i^k(p, q) = \sigma \left(\vec{a} [H_i^k(p, :) \parallel H_i^k(q, :)]^T \right) + \lambda \cdot A_i^k(p, q)$$

其中 $\sigma(\cdot)$ 是激活函数, \parallel 表示连接操作, $H_i^k(p, :) \in R^{1 \times d}$ 和 $H_i^k(q, :) \in R^{1 \times d}$ 分别表示矩阵 H_i^k 的第 p 行和第 q 行,代表节点 v_p 和 v_q 的表示。特别地, A_i^k 编码了池化子图的结构信息,其中如果节点 v_p 和 v_q 不直接连接,则 $A_i^k(p, q) = 0$ 。我们将 A_i^k 纳入结构学习层,以偏置注意力机制,使其在直接连接的节点之间给出相对较大的相似度分数,同时尝试学习断开连接的节点之间的潜在成对关系。 λ 是它们之间的权衡参数。

为了使不同节点之间的相似度分数易于比较,使用 softmax 函数对它们进行归一化

$$S_i^k(p, q) = \frac{\exp(E_i^k(p, q))}{\sum_{m=1}^{n_i^k} \exp(E_i^k(p, m))}$$

将上面一个 HGP-SL 模块作为一个 Block,接下来是对其的一些更改:

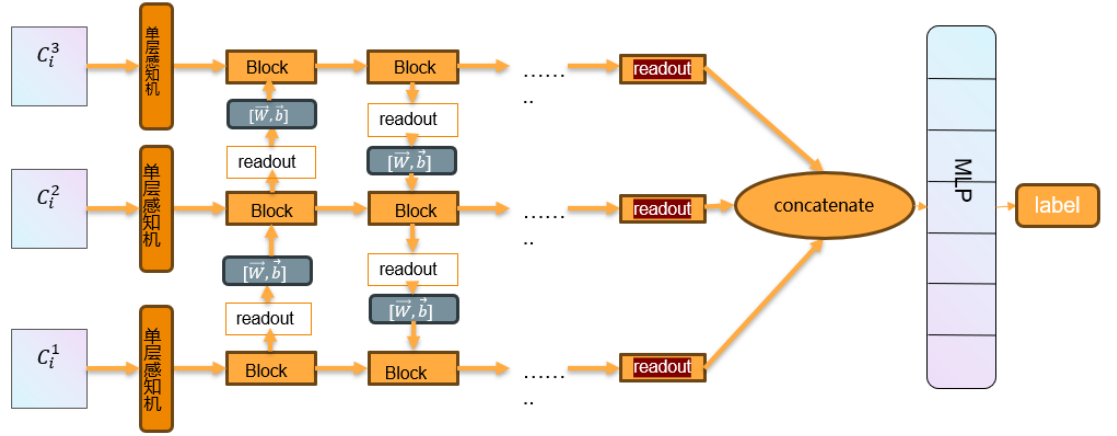


图 2.

首先由于股权结构，资金链等图节点类型并不唯一，有个人，也有企业。因此作为异构图，其节点的特征维度是不同的：

$$G = \langle A, [H_1, H_2, H_3, \dots, H_n] \rangle$$

其中 n 是异构图中节点类型。这会对后续不同节点的信息流通造成阻碍。因此，在对某一企业的三张图进行操作之前，先将他们的点属性集合进行重投影，投影至同一特征空间，即：

$$f([H_1, H_2, \dots, H_n]) \rightarrow H'$$

其中 $H' \in R^{N \times d'}$, d' 为单层感知机的输出特征维度。

其次，由于三张图都属于同一企业，尽管他们代表着不同的信息维度，但他们之间仍应有信息流动。例如，当一家企业全资控股的子公司出现重大事故时，这种风险也会传导至与其担保或被担保的企业。

因此为了实现跨图的信息流动，我们在进行一次完整的卷积，池化操作后，会对每张图进行 readout 操作，即聚合整张图的信息，然后进行一次重投影，将原图特征空间投影至等待信息传递图的特征空间。

然后，在被投影图中添加一个超级节点，与所有节点都相连，其节点特征正是 readout 整合投影后的特征。然后在该扩展后的图中进行一次 GAT 操作，将该超级节点的信息以注意力机制分配到不同节点当中，进行更新后再向前进行信息传递。

具体而言：

$$feature(C_i^j) = readout(C_i^j)$$

$$NewC_i^{(j+1)} = \langle E_i^{(j+1)} \cup E', V_i^{(j+1)} \cup \{supernode\} \rangle$$

其中 E' 为超级节点与 $V_i^{(j+1)}$ 所有节点相连所形成的边集。

然后利用注意力机制：

对于顶点 i , 逐个计算它的邻居们, $j \in \mathcal{N}_i$ 和它自己的相似系数：

$$e_{ij} = a([Wh_i \parallel Wh_j]), j \in \mathcal{N}_i$$

然后利用 softmax 对离散的相似系数进行归一化：

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(e_{ik}))}$$

根据计算好的注意力系数，把特征加权求和，进行聚合

$$h'_i = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W h_j \right)$$

然后将该 supernode 和 E 遮掩，再向前继续 GCN 消息传递与聚合。

值得注意的是，在进行跨图的信息传递时，我们采用了慢融合的方式，因为考虑到不同特征空间差异度较大，如果采用早融合的方式，极有可能使得低纬度的子空间特征被覆盖掉，从而使得该图神经网络无法很好的获得低维空间的特征信息，并且有可能会使网络无法收敛。