

日期： /

## Traditional ML Methods: Nodes

在传统的ML模型中，通常有2个步骤：

1. 提取节点/边/子图/图的特征表示 (A)

2. 训练模型并进行预测

使用有效的图特征是保证模型质量的关键

### 节点级特征

• Degree: 对所有节点一视同仁

• Centrality: 测量节点在图中的重要性

如果节点 $v$ 被重要的邻居节点 $u \in N(v)$ 包围，

就认为 $v$ 也是重要的

$$C_\lambda = \frac{1}{\lambda} \sum_{u \in N(v)} C_{uv} \Leftrightarrow \lambda C = \underbrace{AC}_{(\lambda > 0)} \xrightarrow{\text{特征方程}} \begin{array}{l} \text{邻接矩阵} \\ \text{中心向量} \end{array}$$

通常将最大特征值( $\lambda_{max}$ )对应的特征向量作为节点中心性的度量

• Betweenness centrality (介数中心性): 如果一个节点位于许多其他对之间的最短路径上，则节点重要。

$$C_V = \sum_{s \neq v \neq t} \frac{\# \text{(shortest paths between } s \text{ and } t \text{ that contains } v)}{\# \text{(shortest paths between } s \text{ and } t)}$$

日期： /

- Closeness centrality (接近中心性)：如果节点到图中其他节点的最短路径短，则节点重要。

$$C_V = \frac{1}{\sum_{u \neq v} \text{shortest path length between } u \text{ and } v}$$

- Clustering Coefficient (聚类系数)：衡量一个节点的邻居之间的连接紧密性。

$$C_V = \frac{\#(\text{edges among neighbouring nodes})}{\binom{k_v}{2}} \in [0, 1]$$

node pairs among  $k_v$  neighbouring nodes

或写作  $C_v^2$

### [ Graphlets(图元) ] (A)

- 聚类系数计算了节点自我网络(ego-network)中三角形的数量。
- 图元是一个根植的(rooted), 连通的(connected), 非同构的(non isomorphic)子图。

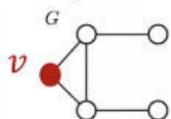
### • 图元度向量(GDV)：

- Graphlet-based node feature

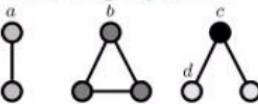
- 给定图元在给定节点处根植的次数

日期: /

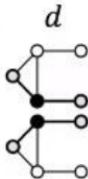
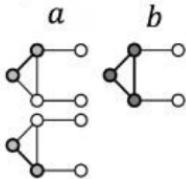
■ Example:



List of graphlets



Graphlet instances:



GDV of node  $v$ :  
 $a, b, c, d$   
[2,1,0,2]