论文笔记 | node2vec: Scalable Feature Learning for networks

论文【node2vec: Scalable Feature Learning for networks】为可度量的网络特征学习，是由斯坦福大学的`Aditya Grover`和`Jure Leskovec`在2016年提出的，它实际上是对DeepWalk的改进，是基于DeepWalk的升华。所以在了解node2vec之前，可以先看对**DeepWalk**的解读。

简介

DeepWalk是图特征学习的开山之作，它的主要是思想是借助了在处理文本任务中词向量的处理方法来解决图特征学习的问题，主要方法是随机漫步+语言模型，先通过随机漫步找出N个经过路径节点的有序序列，再对序列进行skip gram训练，得到对应的向量。但是该方法也有一些缺陷或者不严谨的地方，第一是随机漫步可以理解为简单的深度优先搜索，深度往往增加了算法和模型的复杂度，但并没有考虑到广度优先带来的周围邻居结构的影响；第二是，文中并没有给出一个明确的优化目标函数。

node2vec在DeepWalk的基础上提出了更加合理的图特征学习方法，提出了用于网络中可伸缩特征学习的半监督算法，使用SGD优化一个自定义的基于图的目标函数，该方法可以最大化的在D维特征空间保留节点的网络领域信息；在随机游走的基础上设计了一种二阶随机游走的过程，相当于对DeepWalk算法的一种扩展，它保留了邻居节点的图特征。

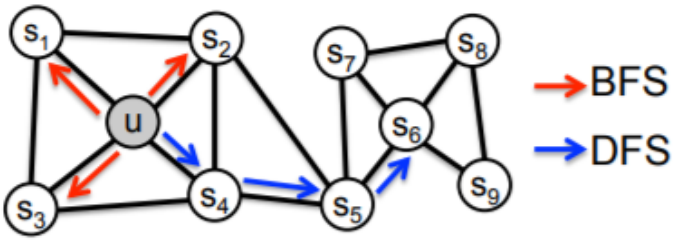
节点相似性度量

网络节点相似性度量的依据：

- 内容相似性：相邻节点之间的相似性，人以群分，我们往往和我们直接关联的人存在着相似；
- 结构相似性：网络局部拓扑结构上的相似性，中介和中介存在结构上的相似性；

重要假设：

- 同质性假设：连通的节点并且属于相似网络簇或者社群的节点应该相似；
- 结构等价假设：具有相似结构角色的节点应该相似；



如上图所示，结点u和结点`s6`在局部网络结构中具有相似的结构角色，所以应该具有相似性，虽然两个结点之间并没有直接相连。又如，结点`s1`和u属于同一个网络簇（直连），所以更应该相似。

学习框架

将网络中的特征学习问题描述为一个极大似然优化问题，设$G=(V,E)$为给定网络，设$f:V\Rightarrow\mathbb{R}^d$为节点到特征表征的映射函数，我们的目标是学习一个后续节点的预测任务，这里d是一个参数，指定特征表示的维数，f是一个$|V|\times d$参数的矩阵，为每个源节点$u\in V$，定义$N_S(u)\subset V$的网络邻居节点的社区抽样策略。

目标优化函数如下，该目标函数根据节点u的特征表示，最大化节点u的网络观测邻域$N_S(u)$f$:

$$\max_f \sum_{u\in V}\log \Pr(N_S(u)|f(u)) \tag{1}$$

为了让优化任务易于处理，做了两点假设：

- 条件独立性假设：即假设结点间相互独立，简单来说就是，对于某一个源结点，其采用到的邻居结点是独立的，采用其中一个邻居结点不会对其他邻居结点造成影响。

$$\Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} \Pr(n_i|f(u))$$

- 特征空间对称假设：源结点和邻居结点的特征空间有一个对称性影响。简单来说就是，一个源结点和其某一个邻居结点有关系，那么对于这个邻居结点来说，这个源结点也是其邻居结点，影响是相互的。

$$\Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}$$

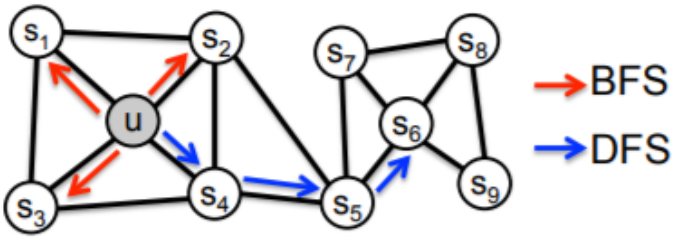
根据以上两个假设，最终目标函数f可以优化为以下形式：

$$\max_f \sum_{u \in V} \left[-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \right] \tag{2}$$

每个节点分割函数$Z_u = \sum_{v \in V} \exp(f(u) \cdot f(v))$，对于大型网络来说，计算成本很高，所以采用负采样来近似它，在定义特征函数f模型参数上，采用随机梯度上升法对公式(2)进行优化。

经典搜索策略

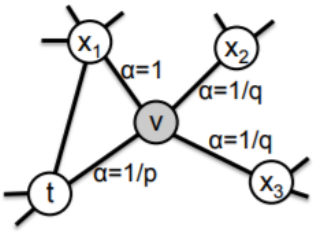
可以将采样源节点邻域问题视为一种网络局部搜索问题，所熟知的无非就是广度优先遍历（BFS）和深度优先遍历（DFS），广度优先更容易采样邻居节点，从而获得每个节点邻居的微观视图，这更容易表示结构的相似性，比如限制一个$k = 3$邻居域，节点u的BFS就会对`s1,s2,s3`采样，广度优先遍历采样邻居节点往往重复对此采样，这有利于减少偏差。对深度优先遍历来说，它尽可能深的遍历网络，采样节点更准确的反映了邻居节点的宏观情况，这更容易表示内容相似性，即验证同质性假设，而u使用DFS就会对`s4,s5,s6`进行采样。



Node2Vec

基于上面的问题，Node2Vec提出了参数化的游走方式。具体来说，N2V提出了两个重要参数p和q用于控制游走。

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$



论文定义了一个二阶的随机游走策略，$\alpha_{pq}(t,x)$是转移概率。

- 返回概率参数（Return parameter）p，对应BFS，p控制回到原来节点的概率，如图中从t跳到v以后，有1/p的概率在节点v处再跳回到t。
- 离开概率参数（In out parameter）q，对应DFS，q控制跳到其他节点的概率。

p用来控制重新访问walk的的结点的可能性，p越大，可能性越小。同理，q越大，加入q>1，那么随机游走就用参数控制游走的结点更结点结点t(降低了往外游走的概率)。如果q<1，则游走方式更大概率探索远离结点t的结点。

总的来说，整个策略是在DFS和BFS之间采取某种平衡，也提供了参数化的控制方式，可以根据不同的需求进行调参，增大了普适性。

总结

总的来说，论文做了以下事情：

1、提出了一种高效的可伸缩网络特征学习算法，该算法使用SGD实现网络感知领域保持的优化目标；

2、寻找一种在广度优先遍历和深度优先遍历之间的平衡点；

3、基于邻域保留目标的特征学习方法扩展到节点和边的预测任务；

4、对多个真实的数据集进行了实证评估；

备注：具体其他的内容可以参考原始论文，更具论文题目可以很轻松的找到原论文，且原论文还提供的Node2vec的源码，PYthon版：【https://github.com/aditya-grover/node2vec】（源码中在保存模型的地方存在部分的问题，需要自行修改一下），当然速度上只能计算百万级节点网络，对于超大型网络来说需要在spark上进行实现。

参考

【1】Aditya Grover and Jure Leskovec.node2vec: Scalable Feature Learning for Networks

【2】Bryan Perozzi,Rami Al-Rfou and Steven Skiena.DeepWalk: Online Learning of Social Representations

【3】https://blog.csdn.net/wen_fei/article/details/82416141