网络表示学习运行报告

deepwalk：

1 random walk

生成随机游走节点序列

2 wordtovec

对游走序列节点进行词向量学习，为每一个节点生成词向量

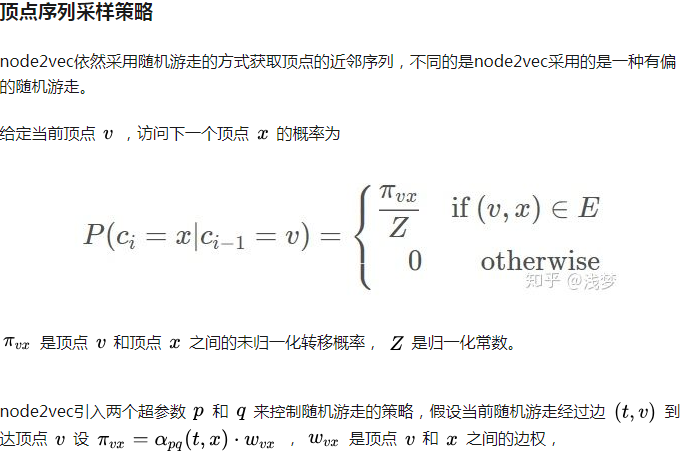
目前针对小规模网络比如karate做测试：

1 feature：网络节点特征的提取： 1 邻接矩阵， 2 邻居vote， 3 randomwalk+word2vec 4 GCN

2 model： 模型的选取就比较弹性了，sklearn系列的很多，目前试了，SVM，K邻近算法，K邻近在deepwalk的特征下面有非常好的表现

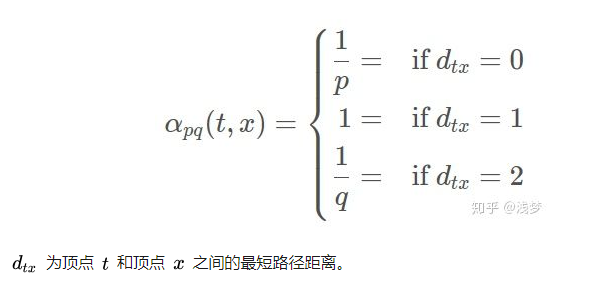
Nodetovec <https://zhuanlan.zhihu.com/p/56542707>

1 basis random walk



转移概率由两个主要因素决定：

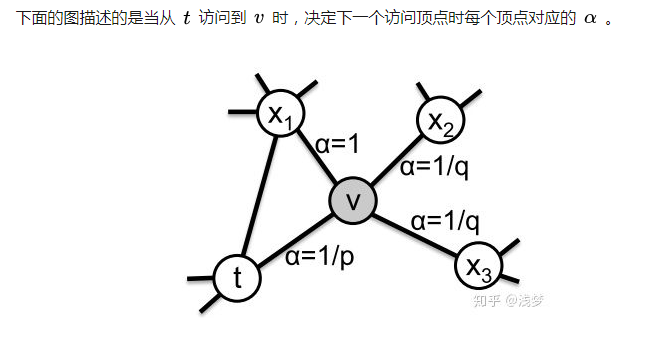
1 图结构上的参考

2边权重的参考

下面讨论超参数 p 和 q 对游走策略的影响

参数p控制重复访问刚刚访问过的顶点的概率。 注意到p仅作用于 d\_{tx}=0 的情况，而 d\_{tx}=0 表示顶点 x 就是访问当前顶点 v 之前刚刚访问过的顶点。 那么若 p 较高，则访问刚刚访问过的顶点的概率会变低，反之变高。

q 控制着游走是向外还是向内，若 q>1 ，随机游走倾向于访问和 t 接近的顶点(偏向BFS)。若 q<1 ，倾向于访问远离 t 的顶点(偏向DFS)。

假设现在节点已经从t-v，下一步的概率分布是：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| V-- |  |  |  |
|  |  |  |  |
|  |  |  |  |

GCN <https://mp.weixin.qq.com/s/sg9O761F0KHAmCPOfMW_kQ>

GCN下的embedding的思想是，每一个节点聚合其领域节点以及自己的特征数据作为当前节点的特征，怎么学习的：

邻接矩阵：A， 特征数据: X 单位矩阵：I 度矩阵：D

从数学上看 A\*X就可以聚合当前节点的一阶领域节点的特征和，

(A+I)\*X可以聚合自身节点以及邻居节点的特征，

D\*\*-1\*A\*X 可以将特征归一化，这一步参考拉普拉斯矩阵分解

一个隐藏层的完整表达就是：relu(D\_hat\*\*-1 \* A\_hat \* X \* W)

D\_hat 具有自环的度矩阵， A\_hat 自环的邻接矩阵 W：权重系数， relu(): 激活函数