

来deepwalk一下我的社交网络

原创 penny本妮 penny菜鸟学cs 2020-04-06

我们知道图中含有丰富的信息如社交网络，web网络，铁路网络。由于图可以轻松捕获节点之间的关系，因此图可以解决一些常规数据结构所无法解决的问题，例如社交网络之间节点的关系可以衡量两个朋友之间的亲近程度。我们试图从图中提取特征。

1. 节点属性：图中的节点代表实体，并且这些实体具有自己的特征属性。我们可以将这些属性用作每个节点的特征。例如：Wikipedia网络中，每个页面将会有一些关键词作为特征。
2. 局部结构特点：节点的度，中心性，图中的环。
3. 节点嵌入：节点属性仅包含与该节点自身有关的信息，它不捕获有关节点上下文的信息。在上下文中（即周围节点），节点嵌入通过用固定长度向量表示每个节点，并试图捕获上下文信息（有关周围节点的信息）。

DeepWalk受到Word2Vec的启发，试图用Word2Vec这样的方式获得节点的向量。那么想要使用Word2Vec，首先我们需要获得序列信息，此时DeepWalk想到用随机游走的方式产生一个节点序列。

随机游走即从一个节点出发，随机的走向与该节点连接的下一节点直到产生一个设置长度的序列停止。（显然此时已经捕获了一定的周围节点的信息）

得到序列后,如[node1,node3,node8,node4,node5]，用Word2Vec直接训练其向量表示。

现在有一个社交网络：

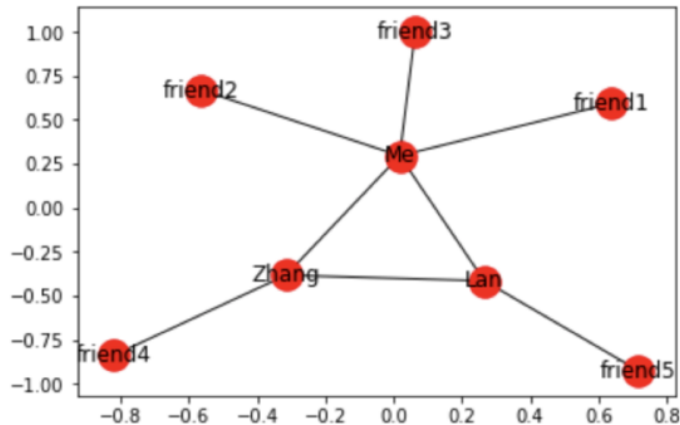
我认识小张和小蓝，然后又认识了我的研究生同学，实习的同事，踢足球的小伙伴
小张去UCSD认识了新的朋友，小蓝去HKUST也认识了新的朋友。

首先我们需要构建图，python中有一个networkx包可以构建，可以直接由邻接矩阵，也可以由边-边-权重的文档构建。

```

In [3]: import networkx as nx
import gensim
import numpy as np
import matplotlib.pyplot as plt
node_list=['friend1','friend2','friend3','Me','Zhang','Lan','friend4','friend5']
edge_list=[('friend1','Me',1),('friend2','Me',1),('friend3','Me',1),('Zhang','Me',1),
('Lan','Me',1),('friend4','Zhang',1),('friend5','Lan',1),('Zhang','Lan',1)]
#创建空图
G=nx.Graph()
#从一个列表中添加节点
G.add_nodes_from(node_list)
#根据 (边, 边, 权重) 加载
G.add_weighted_edges_from(edge_list)
#plot
nx.draw_networkx(G,node_list=G.nodes(),edges=edge_list)

```



开始随机游走:

```

In [58]: import random
random.seed(666)
def random_walk(G,start=None,path_length=20,alpha=0,rand=random.Random()):
    """return a random walk path"""
    if start:
        path=[start]
    else:
        path=[rand.choice(list(G.nodes()))]
    while len(path)<path_length:
        cur=path[-1]
        #find it's neighbors
        if len(G[cur])>0:
            if rand.random()>=alpha:
                path.append(rand.choice(list(nx.all_neighbors(G,cur))))
            else:
                path.append(path[0])
        else:
            break
    return path
print(random_walk(G,start='Me'))

```

['Me', 'friend1', 'Me', 'friend1', 'Me', 'friend1', 'Me', 'friend2', 'Me', 'friend1', 'Me', 'friend3', 'Me', 'friend3', 'Me', 'Lan', 'friend5', 'Lan', 'Me', 'friend2']

建立随机游走的“语料库”:

['Me', 'friend2', 'Me', 'Zhang', 'Me', 'Lan', 'Me', 'friend3', 'Me', 'friend3', 'Me', 'friend3', 'Me', 'friend3', 'Me', 'friend2', 'Me', 'friend2', 'Me', 'friend1'],
 ['friend1', 'Me', 'friend2', 'Me', 'Zhang', 'Lan', 'Me', 'friend1', 'Me', 'Zhang', 'Lan', 'Me', 'Zhang', 'Me', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang',
 'friend4'], ['Lan', 'Me', 'friend1', 'Me', 'friend2', 'Me', 'Zhang', 'Lan', 'friend5', 'Lan', 'Me', 'friend3', 'Me', 'friend1', 'Me', 'friend1', 'Me', 'friend1', 'Me', 'friend3',
 'Me', 'friend5', 'Lan', 'Me', 'Lan', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'Me', 'friend1', 'Me', 'friend1', 'Me', 'friend3',
 'Me'], ['friend2', 'Me', 'friend1', 'Me', 'friend1', 'Me', 'Lan', 'Me', 'Lan', 'friend5', 'Lan', 'Me', 'friend2', 'Me', 'friend3', 'Me', 'Zhang', 'Lan', 'Me', 'friend3',
 'Zhang', 'friend4', 'Zhang', 'Me', 'friend1', 'Me', 'friend3', 'Me', 'friend1', 'Me', 'Lan', 'Zhang', 'Lan', 'Me', 'friend2', 'Me', 'Zhang', 'Lan', 'Zhang', 'Me'],
 ['friend4', 'Zhang', 'Lan', 'friend5', 'Lan', 'Zhang', 'Lan', 'Me', 'Lan', 'friend5', 'Lan', 'Zhang', 'Me', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'friend4',
 'Zhang'], ['friend3', 'Me', 'Zhang', 'Me', 'Zhang', 'Lan', 'Zhang', 'friend4', 'Zhang', 'Lan', 'friend5', 'Lan', 'Zhang', 'friend4', 'Zhang', 'Me', 'friend2', 'Me', 'Lan', 'friend5', 'Lan', 'Zhang', 'friend4', 'Zhang',
 'Me', 'friend3'], ['Lan', 'friend5', 'Lan', 'Zhang', 'friend4', 'Zhang', 'Lan', 'friend5', 'Lan', 'Zhang', 'Lan', 'Me', 'friend1', 'Me', 'Lan', 'Me', 'friend2', 'Me', 'friend3',
 'Me'], ['friend5', 'Lan', 'friend5', 'Lan', 'Zhang', 'Lan', 'Zhang', 'Lan', 'friend5', 'Lan', 'Me', 'friend3', 'Me', 'Lan', 'Zhang', 'Lan', 'friend5', 'Lan', 'Zhang',
 'Lan'], ['friend1', 'Me', 'Lan', 'friend5', 'Lan', 'friend5', 'Lan', 'friend5', 'Lan', 'friend5', 'Lan', 'Zhang', 'friend4', 'Zhang', 'Lan', 'Me', 'friend1', 'Me', 'friend2', 'Me', 'friend3',
 'Me', 'friend3', 'Me', 'friend3', 'Me', 'friend3', 'Me', 'Zhang', 'friend4', 'Zhang', 'Lan', 'Me', 'Zhang', 'friend4', 'Zhang', 'Lan', 'Me', 'Lan', 'friend5', 'Lan', 'friend5',
 'Lan', 'friend5', 'Zhang', 'Me', 'friend1', 'Me', 'friend3', 'Me', 'Zhang', 'Lan', 'Me', 'Lan', 'Zhang', 'Lan', 'Zhang', 'Lan', 'friend5', 'Lan', 'Me', 'friend2',
 'Me'], ['Zhang', 'Lan', 'friend5', 'Lan', 'friend5', 'Lan', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'Me', 'Zhang', 'Lan', 'friend5', 'Lan', 'friend5', 'Lan', 'friend5',
 'Lan'], ['Me', 'Lan', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'Lan', 'Me', 'friend1', 'Me', 'friend1', 'Me', 'friend2', 'Me', 'friend2'],
 ['friend3', 'Me', 'Lan', 'Me', 'Lan', 'Zhang', 'Lan', 'Zhang', 'friend4', 'Zhang', 'Lan', 'Zhang', 'friend4', 'Zhang', 'friend4', 'Zhang', 'Me', 'friend2', 'Me', 'friend1']]

训练每个节点的向量:

```
Out[81]: [('Zhang', 0.9844411611557007),
('Lan', 0.9722907543182373),
('friend5', 0.9581975340843201),
('friend1', 0.865242063999176),
('Me', 0.8347228765487671),
('friend2', 0.8336395621299744),
('friend3', 0.8204163312911987)]
```

让我们来看看小张和谁玩得最好

```
Out[82]: [('Lan', 0.991287350654602)]
```

竟然不是我，好气袄！TAT说明deepwalk还不够好啊(´_ゝ｀)

喜欢此内容的人还喜欢

1875门首批国家级一流本科线上课程 入选高校、开发方式、学科归类和依托平台分析