# network representation learning（NRL）

## Label Informed Attributed Network Embedding

这篇论文关注的是带标签的属性网络的特征向量化方法，Attributed Network Embedding（ANE）。

与只观察节点间的相互关系的普通网络（无监督的方法）不同，属性网路中的节点往往与丰富的标签信息相关联。本论文研究如何对标签进行建模和整个，将标签信息嵌入到网络特征向量中改进网络特征向量。

需要解决的问题：1、属性网络和标签信息可能是稀疏的、不完整的和有噪声的；2、考虑到属性网络及其标签的异质性，学习统一表示是具有挑战性的。

贡献：研究了标签对网络特征学习的潜在影响是什么；提出的LANE模型，可以帮助其他学习任务，如节点分类。

## From Properties to Links: Deep Network Embedding on Incomplete Graphs

本文研究不完全网络（具有大量缺失链接的不完全图和有新节点加入的动态图）中的网络特征问题。我们提出了一种基于多视点相关的深层网络特征方法：MVC-DNE,将更有效根据网络结构和节点性能对不完全网络的特征进行学习。

MVC-DNE可以学到特征映射功能，它可以直接新加入节点的特征向量，而不需要重新训练模型。如图1所示，原网络有多个视图数据（社会关系视图和用户信息视图），通过MVC-DNE将其特征化到一个多维视图空间中。

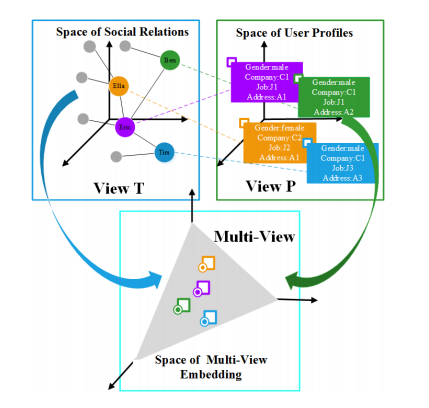


Figure 1

## metapath2vec: Scalable Representation Learning for Heterogeneous Networks

提出了对异质网络的表示学习算法metapath2vec和metapath2vec++。论文的主要亮点在于：通过元路径来指导随机游走，从而构造出节点的邻居节点集合，然后基于异质的skip-gram模型进行节点embedding。

异质网络的表示学习就是学习不同类型的节点的特征表示，这个特征表示能够保留网络的结构特征和语义关系。

虽然异质网络中节点类型不同，但是特征表示空间是相同的。

元路径（Meta Path）是一种通过一组关系连接多个节点类型的路径，可以用来描述异质网络中不同类型对象之间各种连接的不同语义关系。

基于随机游走和Skip-gram的模型进行改进。

## Adversarial Network Embedding

网络特征化是一个具有挑战性的研究问题，因为图形数据具有高度的维数、稀疏性和非线性。

现有的方法除了能捕获网络结构属性，它们中的大多数都缺乏表示鲁棒性的额外约束（实际应用中很常见的噪声网络数据处理时很困难）。这篇文章提出了一个对抗性网络特征学习（Adversarial Network Embedding）ANE框架。

Adversarial Network Embedding Framework如下图2所示。

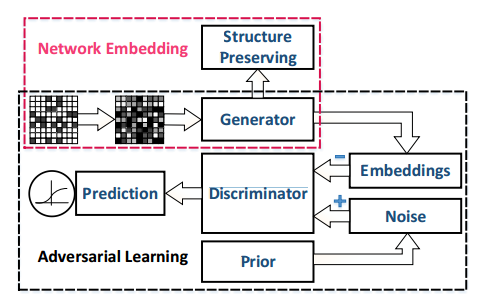


Figure 2

ANE利用对抗性学习原理学习网络的鲁棒性特征表示，它由两部分组成，即结构保持组件和对抗性学习组件。前者的目的是捕捉网络结构的性质，而后者有根据给定的先验知识和后验分布的匹配学习网络的鲁棒特征表示。

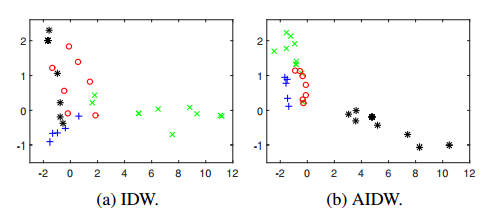


Figure 3

如图2所示，对Zachary’s Karate network网络的二维表示，a部分为Inductive DeepWalk，b部分为Adversarial Inductive DeepWalk。不同颜色表示为不同的社区划分。图中可以看出AIDW能更好的捕捉结构信息。

## On Embedding Uncertain Graphs

不确定图来表示数据存在嘈杂或者不精确，其边缘用概率表示。最近研究不确定图的算法（clustering, classification, and k-NN）。这些算法面临两个问题：（1）高维：不确定图往往高度复杂，影响了挖掘质量；（2）重用性差，现有的挖掘算法需要重新设计来处理不同任务的不确定图。

我们提出了一种称为URGE（UnceRtain Graph Embedding）的解决方法。给定一个不确定图G，给出G的一组低维向量。

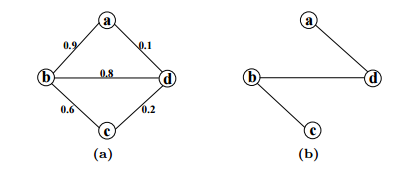


Figure 4

如图3所示，a部分是一个不确定图，b部分为不确定图可能的一中状态。

## Structural Deep Embedding for Hyper-Networks

这篇文论研究的是超网络的特征化方法。

超网络是指超过成对关系，有三个或者更多的对象参与一条边的关系。

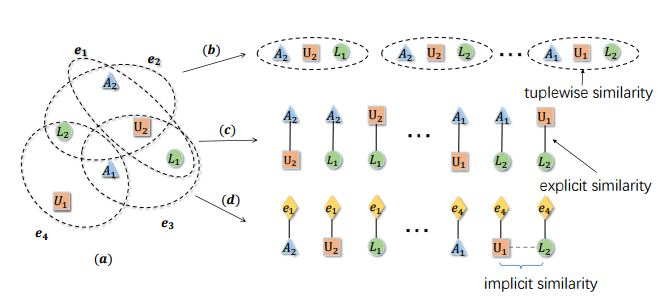


Figure 5

分析超网络的一种典型方法是将其扩展到传统的成对网络。有两种代表性的技术：团体扩张（Clique expansion）每个超边扩展为一个团体（互相连接形成圈）；星型扩张（star expansion）超图转化为一个二分图每个超边通过与一个实例节点相连。

如上图5所示是一个超网络扩展的例子。(a)是一个超网络；（b）是这篇论文中的扩张方法，保留有组相似性；（c）是团体扩张，保留了显式的相似性；（d）是星型扩张，保留了隐式的相似性。

超网络存在的问题：

异构超网络的超边通常是不可分解的：在一个超边的一组节点有很强的相关关系，而在其子集的节点不一定有很强的关系。

提出的新的基于超网络的特征化方法Hyper-Network Embedding (DHNE) ，它在解决超边的不可分解同时可以保留丰富的结构信息。该方法的复杂度与节点数成线性关系，可用于大规模网络。

## Representation Learning for Scale-free Networks

本文研究的是无标度网络的学习表示问题。

人造的网络结构和随机的网络结构中节点与其他节点的连结的数量分布都有规则可循，因此是有尺度的网络。而无尺度网络（或称无标度网络）是带有一类特性的复杂网络，其典型特征是在网络中的大部分节点只和很少节点连接，而有极少的节点与非常多的节点连接。如互联网这样的无标度网络。

之前的关于网络embedding的方法主要集中在保留网络的微观结构上，如节点之间的关系。而无标度特性是网络最基本的宏观属性之一。无标度特性描述网络节点度分布服从幂率分布。

论文通过将问题转化为球面填充问题（？）（Sphere Packing problem），研究在欧氏空间中恢复顶点度幂律分布的可行性。提出了基于保护无标度特性的定点处罚算法：处罚那些接近的度集中的顶点。基于光谱技术和skip-gram实现论文中的方法。

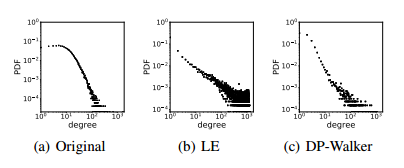


Figure 6

如图6所示一个现实无标度网络的特性。（a）是一个学术网络，（b）和（c）分别是基于LE学习和基于本文提到的方法DPWalker方法学习的特征重建的网络度分布。

## Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering

拉普拉斯特征映射（LE）也是一中常见的降维方法。Laplacian Eigenmaps 看问题的角度和LLE有些相似，也是用graph的角度去构建数据之间的关系。它的直观思想是希望相互间有关系的点（在图中相连的点）在降维后的空间中尽可能的靠近。拉普拉斯特征映射可以反映出数据内在的流形结构。

使用时算法具体步骤为：

步骤1：构建图

使用某一种方法来将所有的点构建成一个图，例如使用KNN算法，将每个点最近的K个点连上边。K是一个预先设定的值。这样构建的图矩阵就是一个稀疏矩阵，只保留了最相似的K个邻居关系。

步骤2：确定权重

确定点与点之间的权重大小，例如选用热核函数来确定（当然这个地方你完全可以选择其他的相似度度量方式来衡量），如果点和点相连，那么它们关系的权重设定为：

另外一种可选的简化设定是如果点，相连，否则。

步骤3：特征映射

计算拉普拉斯矩阵L的特征向量与特征值：

其中D是对角矩阵，满足，

使用最小的m个非零特征值对应的特征向量作为降维后的结果输出。

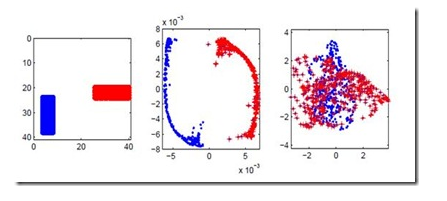


Figure 7

见图7所示，左边的图表示有两类数据点（数据是图片），中间图表示采用Laplacian Eigenmap降维后每个数据点在二维空间中的位置，右边的图表示采用PCA并取前两个主要方向投影后的结果，可以清楚地看到，在此分类问题上，Laplacian Eigenmap的结果明显优于PCA。

## SHINE: Signed Heterogeneous Information Network Embedding for Sentiment Link Prediction

在线社交网络（facebook、Twitter、Weibo）中，人们常常与其他人分享情感和表达态度，这在用户之间形成了大量的情感联系。以往的研究主要集中在文本情感分类上，然而，文本信息只能揭示用户真实意见的“冰山一角”，其中多数是未被观察到的。

为了解决这个问题，本文研究了如何在异构信息存在的情况下预测可能存在的情感链接。

首先，由于主流社交网络缺乏明确的情感链接，通过实体层次情感提取方法，建立了一个由用户情感关系、社会关系和轮廓知识组成的标签异质情感数据集。然后提出了一种新的、灵活的端到端符号异构信息网络特征化框架（SHINE），从异构网络中提取用户潜在的表示，并预测未观察到的情感链接的符号。

使用的数据集是微博，研究基础方法：LINE、node2vec、SDNE、FxG、LIBFM。

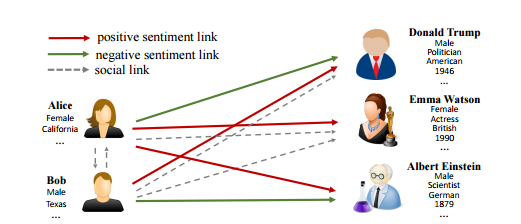


Figure 8

如上图8是一个具有情感、社会关系和用户配置的异构网络片段。

## Structural Deep Network Embedding

一阶估计与二阶估计的概念,最早是在LINE模型中提出的,提出这一概念,主要目的是为了解决前面提出的重构网络信息不全面和网络稀疏的问题。

一阶估计定义为,两个存在链接的节点的相似程度。以下面的图9为例,节点6和节点7因为有公共链接,则它们的一阶估计值会比较大。但是由于网络稀疏的缘故,节点之间的链接并不一定存在,但这不能说明节点之间不相似，所以,为了解决这一问题,这里提出二阶估计的概念。二阶估计衡量的是不同节点相邻结构的相似程度。同样以上图为例,节点5和节点6虽然没有直接相连,但它们的链接结构是类似的,所以它们二阶估计的值,会相对比较大。

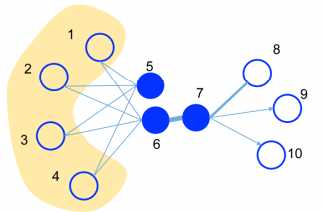


Figure 9

基于上述概念,本文的一大贡献在于提出了一种新的半监督学习模型,结合一阶估计与二阶估计的优点,用于表示网络的全局结构属性和局部结构属性。

## Attributed Network Embedding for Learning in a Dynamic Environment

动态的属性网络特征化

动态网络：节点属性实时更新，动态的增删节点和边的网络

提出模型DANE：第一步offline模型，第二步online 模型

如下图1所示

第一步t，生成网络本身的关系矩阵和属性矩阵分别向量化，最后合并；

第二步t+1，更新网络关系（增删节点和边）和属性，分别向量化合并。

（没有用word2vec和随机游走，好像是矩阵运算）

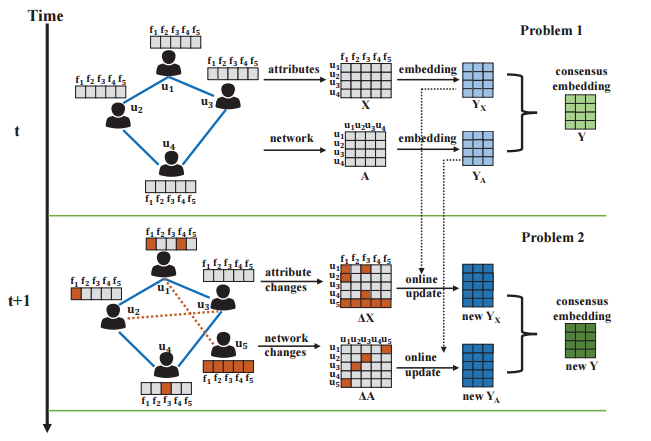


Figure 1

## Enhancing the Network Embedding Quality with Structural Similarity

利用结构相似性提高网络特征质量

基于随机游走的抽样策略不能完全捕获结构等价

提出了一种新的方法，SNS，使用结构信息（graphlets）来执行网络特征化以提高其质量，利用邻居信息和本地子图相似度来学习节点向量。

是第一个将图挖掘和机器学习两个重要领域这两方面结合起来的框架。

使用CBOW

## Asymmetric Transitivity Preserving Graph Embedding

现有的大多数图特征化方法都是针对无向图的。

已有的图特征化方法不能很好地保留非对称传递性，这是有向图的一个重要性质。非对称传递刻画了有向边之间的相关性，即，如果有一条从u到V的有向路径，那么很可能有u到v的有向边。

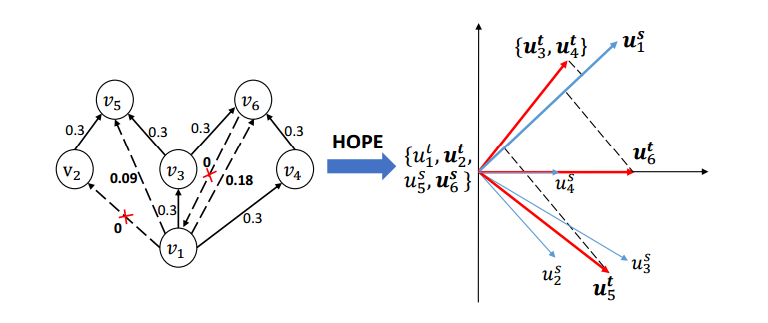


Figure 2

举例如上图2，根据非对称传递性这里有两条路径：和，猜测存在可能性很大为0.18，另一方面的可能性很小为0。

和分别为将i做为源节点和目的节点的向量。

HOPE算法（High Order Proximity preserved Embedding）

实验：链路预测、图重建

## Learning Node Embeddings in Interaction Graphs

符号网络是指边具有正或负符号属性的网络,其中,正边和负边分别表示积极的关系和消极的关系。

研究社会网络与标签属性嵌入的问题。提出新的框架SNEA

利用网络结构和用户属性同时来学习表示网络。

## Max-Margin DeepWalk: Discriminative Learning of Network Representation

DeepWalk作为一种典型的学习社交网络节点向量表示的方法,在一些任务上缺乏足够的区分能力。提出一个半监督模型MMDW。

本文提出Max-Margin(最大间隔)DeepWalk方法,在最大间隔分类器的影响下,原先学习到的节点向量的区分能力有所增强。（对deepwalk的改进）

## Learning Node Embeddings in Interaction Graphs

属性化相互作用网络的向量特征化

特征化方法：IGE (Interaction Graph Embedding)

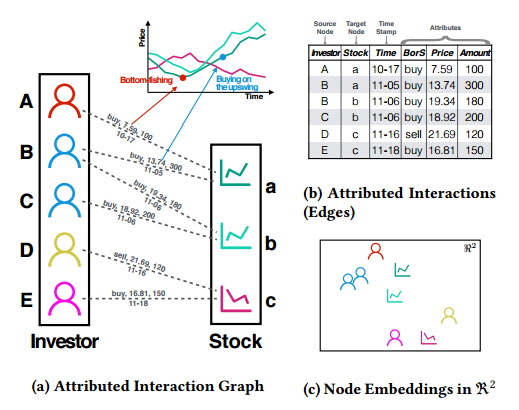
由三种神经网络组成：an attributes encoding network and two prediction networks 

Figure 3：An example of attributed interaction graphs in stock market

## Learning Community Embedding with Community Detection and Node Embedding on Graphs

本文研究的问题是网络社区特征、社区检测和节点特征三者之间的闭环关系。因为节点特征有助于提高社区检测，进行好的社区划分；同时社区特征可以来优化节点特征。

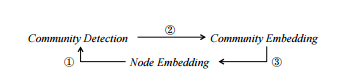


Figure 4：A closed loop for learning community embedding

提出一种新的网络社区特征框架ComE，可以解决社区特征、社区检测和节点特征的一些问题。

# node2vec悲惨世界人物关系网络实验

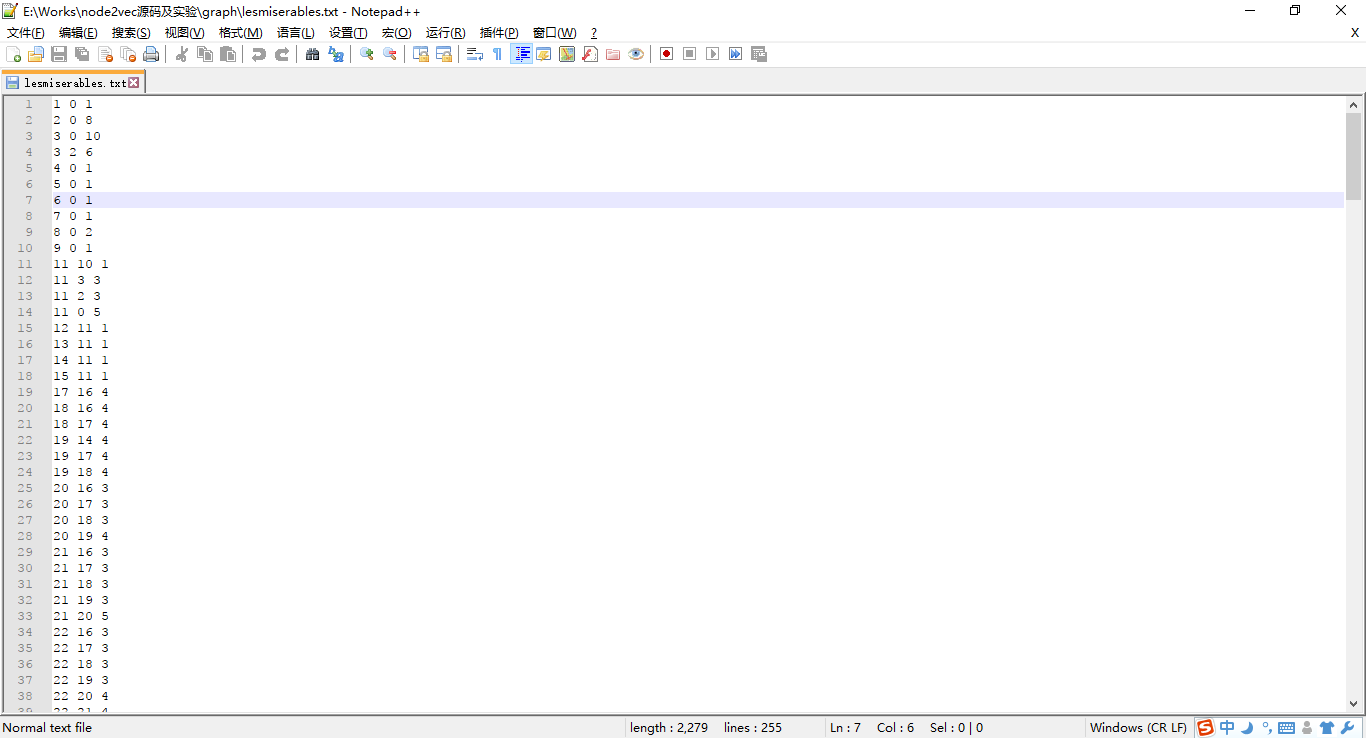


Figure 5

得到的原始的悲惨世界的人物关系数据lesmiserables.txt。

如上图5，每行三个整数，前两个代表有边连接的两个节点编号，第三个整数为边的权值。

运行node2vec，第一次设置p=1（默认为1），q=0.5（默认为1），d向量维数默认为128

python2 src/main.py --input graph/lesmiserables.txt --output emb/top.txt --q 0.5 --weighted

得到向量集 top.txt

第二次设置p=1，q=2，d默认为128

python2 src/main.py --input graph/lesmiserables.txt --output emb/bottom.txt --q 2 --weighted

得到向量集 bottom.txt

实验过程，使用kmeans方法对节点进行社区划分，不同的社区节点使用不同的颜色。分别在同一图上通过得到的两个向量集进行两次画图，进行比较。

实验结果：发现按照原论文实验在q=0.5和q=2时应该分别体现同质性或同构性，如图6。但在我的实验结果中，两次画图结果直观上非常接近，只体现了节点的同质性，相邻的节点被划分为同一社区，如图7和图8。

可能存在的问题：我觉得可能是社区划分的方法还有问题，因为不了解KMeans方法，我是调用了python sklearn库中的KMeans函数，需要固定社区划分数

**kmeans=KMeans(n\_clusters=clusters) # clusters固定的划分数**

第一次我让他们都划分为6个社区如图5； 第二次分别让q=0.5划分为6个社区

q=2划分为3个社区。结果都不太好

下次要尝试手写迭代的自动划分社区的KMeans方法

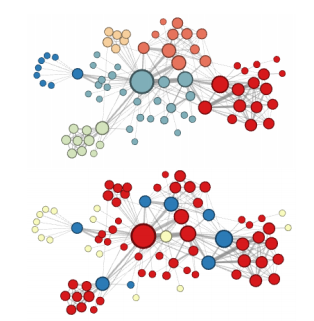


Figure 6：原论文中实验得到的图

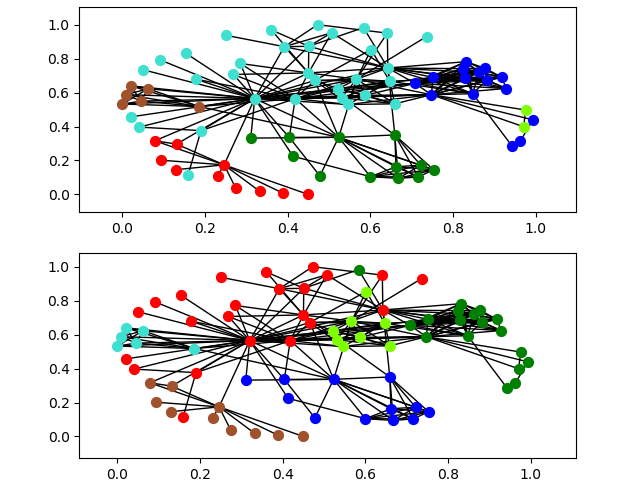


Figure 7：自己实验得到的图，两个划分的社区相同都为6

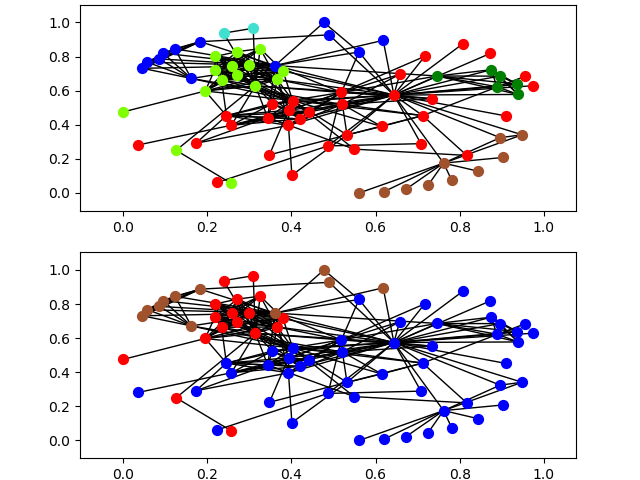
****

Figure 8：第二次试验，两次划分社区数不同，一个为6一个为3

**实验代码**

*#node2vec悲惨世界实验 author：raowei*

*import numpy*

*import networkx as nx*

*import matplotlib.pyplot as plt*

*import math*

*from sklearn.cluster import KMeans*

*row=77 #节点向量的个数*

*col=128+2 #节点向量维数*

*def cos\_dist(a, b): #计算cos距离*

*if len(a) != len(b):*

*return None*

*part\_up = 0.0*

*a\_sq = 0.0*

*b\_sq = 0.0*

*for a1, b1 in zip(a,b):*

*part\_up += a1\*b1*

*a\_sq += a1\*\*2*

*b\_sq += b1\*\*2*

*part\_down = math.sqrt(a\_sq\*b\_sq)*

*if part\_down == 0.0:*

*return None*

*else:*

*return part\_up / part\_down*

*def readvec(vectxt): #读取节点向量保存在vec中*

*f = open(vectxt)*

*line = f.readline()*

*line=f.readline()*

*i=0*

*vec=[[0]\*col]\*row*

*while line:*

*vec[i]=[float(x) for x in line.rstrip("\n").split(" ")]*

*line=f.readline()*

*i+=1*

*f.close()*

*return vec*

*def draw(G,pos,vertxt,clusters): #画图*

*vec=readvec(vertxt)*

*vec0=[int(x[0]) for x in vec] #找到节点编号*

*kmeans=KMeans(n\_clusters=clusters) #使用kmeans方法划分社区*

*vec129=[x[1:129] for x in vec]*

*kmeans.fit(vec129)*

*size = float(len(set(kmeans.labels\_))) #根据不同社区画出不同颜色的节点*

*count=0*

*if clusters==6:*

*nodecolor=['red','blue','green','sienna','turquoise','chartreuse']*

*else:*

*nodecolor=['red','blue','sienna']*

*for k in set(kmeans.labels\_):*

*data\_nodes=set()*

*for index in range(len(kmeans.labels\_)):*

*if kmeans.labels\_[index]==k:*

*data\_nodes.add(int(vec0[index]))*

*list\_nodes=list(data\_nodes)*

*nx.draw\_networkx\_nodes(G, pos, list\_nodes, node\_size = 50, node\_color=nodecolor[count])*

*count+=1*

*nx.draw\_networkx\_edges(G,pos)*

*def main():*

*G1 = nx.Graph()*

*f = open("graph/lesmiserables.txt") #读取节点的边集信息*

*line = f.readline()*

*while line:*

*line=line.split(" ")*

*a=int(line[0])*

*b=int(line[1])*

*weight=int(line[2])*

*G1.add\_edge(a,b) #生成节点和边*

*line = f.readline()*

*f.close()*

*pos = nx.spring\_layout(G1)*

*G2=G1*

*plt.subplot(211)*

*draw(G1,pos,"emb/top.txt",6) #分别用两个不同的向量集做实验*

*plt.subplot(212)*

*draw(G2,pos,"emb/bottom.txt",3)*

*plt.show()*

*if \_\_name\_\_ == '\_\_main\_\_':*

*main()*