# 案例三 图与网络分析

## 【1】案例背景

本题由 **顾翔 921127940122**负责

空手道俱乐部网络是一个经典的社交网络，在20世纪70年代初期，由美国社会学家经过近三年的时间研究获得，该网络包括两个真实的社区，很多研究者是用该网络来检验社区发现算法的效果，本案例提供了空手道俱乐部网络数据集，请尝试利用本章介绍的社区发现算法，对该网络进行社区划分，社区数量选取为2，并与真实的社区划分结果进行对比，读者可以使用igraph进行网络可视化和社区发现

## 【2】方法陈述

使用python-igraph库处理图，使用GN算法进行社区发现，并对结果进行分析

## 【3】实验代码

### 步骤一、数据导入

使用graph库导入GML文件

karate=Graph.Read\_GML('karate.gml')

*# 使用字典 visual\_style 对图形的一些布局进行设置*

layout = karate.layout('kk') *# 使用常见的Kamada-Kawai布局*

visual\_style = {}

visual\_style["edge\_color"] = "gray" *# 设置边的颜色*

visual\_style["vertex\_size"] = 20 *# 节点大小设置*

visual\_style["vertex\_color"] = 'rgb(218.89, 232.93, 245.96)'*# 节点颜色设置*

visual\_style["layout"] = layout *# 设置布局模板*

visual\_style["bbox"] = (300, 300) *# 设置大小*

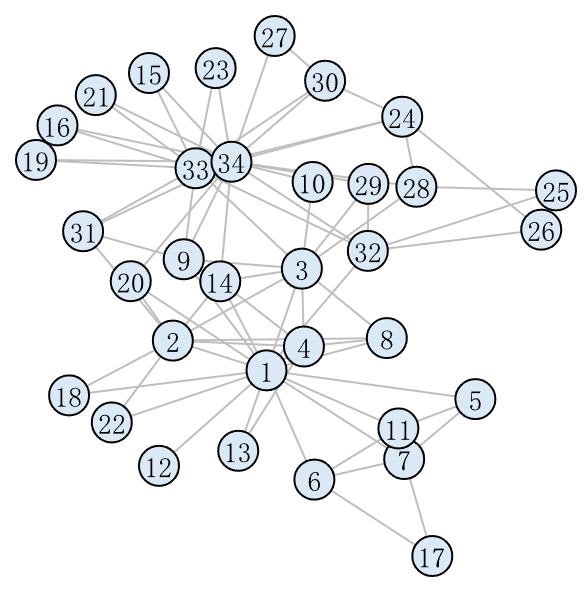
visual\_style["margin"] = 20 *# 设置图形离边缘的距离*

visual\_style["edge\_curved"] = False *# 指定边为弯曲或者直边*

karate.vs['label'] = [int(item) for item in karate.vs['id']]

plot(karate, \*\*visual\_style)

plot(karate, \*\*visual\_style)



**（二）步骤二、数据处理**

使用igraph库可以快速计算各种中心度,社区发现算法采用GN算法,GN算法使用的是介数中心度

vsnumber = karate.vcount() *# 网络节点数*

color\_num = 20

*#sns.palplot(sns.color\_palette("Blues",color\_num))*

colors = [x for x in sns.color\_palette("Blues",color\_num)]

def colors\_by\_centrality(centrality):

    """

    输入：centrality，中心度值的列表

    输出：karate.vs["colors"],每个节点对应的颜色值列表

    """

    centrality\_df = pd.DataFrame(centrality, index=range(1,vsnumber+1), columns=['centrality'])

    bound\_low = min(centrality\_df['centrality'])

    bound\_high = max(centrality\_df['centrality'])

*# 获得color\_num个区间*

    bins = list(np.linspace(bound\_low, bound\_high, color\_num+1))

    bins[-1] = bound\_high+0.001 *# 保证最大值在最后一个区间内*

    bins[0] = bound\_low-0.001 *# 保证0在第一个区间内*

*# 每个节点对应的区间用(0,...,9)标记*

    vs\_belongs = pd.cut(centrality\_df['centrality'],bins, labels= range(0,color\_num))

*# 设置每个节点的颜色属性colors*

    for vertex in karate.vs():

        vs\_belong = vs\_belongs.iloc[vertex.index]

        rgb\_code = tuple(map(lambda x:x\*255, colors[vs\_belong])) *#将颜色百分比转换为0-255的RGB值*

        vertex["colors"] = 'rgb'+str(rgb\_code)

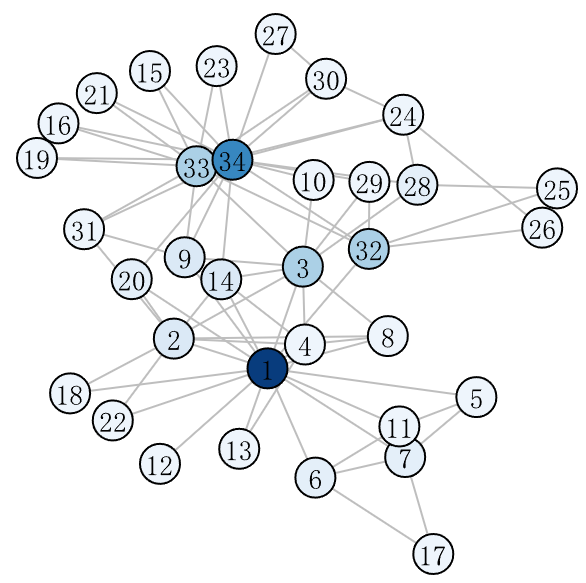
    return karate.vs["colors"]

*# 介数中心度*

between\_cent = karate.betweenness() *# 介数中心度*

visual\_style["vertex\_color"] = colors\_by\_centrality(between\_cent)

plot(karate,\*\*visual\_style)



## 结果分析

与书中原图对照仅有三节点划分错误，准确率较高。

gn = karate.community\_edge\_betweenness(clusters=2)

*#karate.com*

membership\_gn = gn.as\_clustering().membership

colors\_cd = ["powder blue",'eggshell','light sage'] *# cd表示community\_dectection*

*#sns.palplot(sns.xkcd\_palette(colors\_cd),size=2)*

colors\_cd = [x for x in sns.xkcd\_palette(colors\_cd)] *# 转成一个rgb百分值列表*

for vertex in karate.vs():

    color\_belong = membership\_gn[vertex.index]

    rgb\_code = tuple(map(lambda x:x\*255, colors\_cd[color\_belong])) *#将颜色百分比转换为0-255的RGB值*

    vertex["colors"] = 'rgb'+str(rgb\_code)

visual\_style["vertex\_color"] = karate.vs["colors"]

plot(karate,\*\*visual\_style)

