Exercises and answers

《计算社会科学导论》习题和答案

项目统筹：　　吕鹏（中国社会科学院大学）
范晓光（浙江大学）、陈忱（浙江工业大学）
李轩涯、计湘婷（百度校园合作部）
周旅军（中华女子学院）

Baidu 百度

清華大學出版社

目录

第二章 习题

1. 挖掘用户评价的信息

消费者评价是互联网交易中的重要信息。对消费者而言，好评会吸引他们购买某件商品；对商家而言，从好评中能得知商品的优点，从差评中能提取出需要改进的地方。请分析 https://www.kaggle.com/sid321axn/amazon-alexa-reviews 网站的消费者评价文本，这些消费者都购买了亚马逊公司推出的一款智能家居机器人 Alexa。

- 选用合适的方法，将文件读取到程序中。
- 对于文字评价，请完成分词工作。在英语中，词和词之间用空格分隔，在去掉标点符号后，能以空格为标识完成分词工作。你可能发现部分词语对了解人们的想法帮助不大，可以将这些词语去掉。
- 根据评分，对文字评论进行归类，然后在每一类中进行词频统计工作。
- 最后尝试将你得出的结果进行可视化，在图中显示不同评分下评论词频的情况。

2. 谁是《西游记》的主角？

《西游记》是中国古代第一部浪漫主义章回体长篇神魔小说，也是我国古典四大名著之一。请对《西游记》的主题进行探究，分析唐僧、孙悟空、猪八戒、沙僧四人中，谁被提及的次数最多。

- 要做到这点，我们首先需要让程序知道我们要统计的词语。《西游记》原著中用多种方式称呼唐僧、孙悟空、猪八戒、沙僧四人，如孙悟空有时会被叫做孙行者、齐天大圣、斗战胜佛等，这些称呼都需要与同一个人联系起来。请创建一些列表，完成这个任务。
- 请用恰当的方式，将《西游记》的全文本读取到程序中，并进行一定处理，去掉换行符等无关内容。
- 利用 jieba 分词划分文本。请检查结果，是否有部分分词结果不尽如人意。如果是，那么请查阅相关帮助文档，将新词添加到 jieba 分词的词典中，重新进行分词。
- 使用最开始创建的列表，进行词频统计，找出唐僧、孙悟空、猪八戒、沙僧四人中被提及的次数最多的人。

3. 今日热点在哪里？

新闻头条、社会热点是了解社会舆情的有效手段。你想知道今天的热搜是什么吗？是科技、经济、体育、娱乐还是民生？是新闻事件、热点话题、人物动态还是产品资讯？请选择一个新闻聚合网站，使用浏览器中的“开发者控制”功能，查看网站首页新闻标题所在的标签。然后编写相关 Python 代码，爬取新闻聚合网站首页的数据，筛出新闻标题，并用适当的方式进行可视化。

- 选择一个新闻网页，查看相关标签。
- 编写代码，爬取新闻标题数据。
- 进行一次到多次爬取，形成一个文字档案，并完成分词工作，然后进行词频统计。
- 将词频统计结果进行可视化，如直方图、词云图等。

答案

1. 挖掘用户评价的信息

```
1.      # 读取数据分析所需要的库
2.      import numpy as np
3.      import pandas as pd
4.      import re
5.
6.      # 使用 pandas 读取 csv 文件
```

```
reviews = pd.read_csv("work/Amazon alexa reviews.csv",encoding = "utf-8")

# 查看读取文件结果
reviews

# 以一个评价为例，尝试完成分词工作
sample = reviews.iloc[2]["verified_reviews"]
sample_word = re.split("[,.?:; ]", sample.strip())
sample_word

# 构建 5 个字典，完成不同评分的词频统计
rate1 ={}
rate2 ={}
rate3 ={}
rate4 ={}
rate5 ={}

# 分评分做分词
for i in range(0,len(reviews)):
    comment = reviews.iloc[i]["verified_reviews"]
    word = re.split("[,.?:; ]", comment.strip())
    if reviews.iloc[i]["rating"] == 1:
        for j in word:
            if j not in rate1:
                rate1[j] = 1
            else:
                rate1[j] += 1
    if reviews.iloc[i]["rating"] == 2:
        for j in word:
            if j not in rate2:
                rate2[j] = 1
            else:
                rate2[j] += 1
    if reviews.iloc[i]["rating"] == 3:
        for j in word:
    if j not in rate3:
                rate3[j] = 1
            else:
                rate3[j] += 1
    if reviews.iloc[i]["rating"] == 4:
        for j in word:
            if j not in rate4:
                rate4[j] = 1
            else:
                rate4[j] += 1
    if reviews.iloc[i]["rating"] == 5:
        for j in word:
            if j not in rate5:
                rate5[j] = 1
            else:
                rate5[j] += 1

# 删掉空白项目
rate1.pop("")
rate2.pop("")
rate3.pop("")
rate4.pop("")
```

```
63.     rate5.pop("")
64.
65.     # 根据词频排序
66.     rating1 = sorted(rate1.items(), key = lambda kv:(kv[1], kv[0]),reverse=True)
67.     rating2 = sorted(rate2.items(), key = lambda kv:(kv[1], kv[0]),reverse=True)
68.     rating3 = sorted(rate3.items(), key = lambda kv:(kv[1], kv[0]),reverse=True)
69.     rating4 = sorted(rate4.items(), key = lambda kv:(kv[1], kv[0]),reverse=True)
70.     rating5 = sorted(rate5.items(), key = lambda kv:(kv[1], kv[0]),reverse=True)
71.
72.     # 导入库，绘制柱状图
73.     %matplotlib inline
74.     import matplotlib.pyplot as plt
75.     import seaborn as sns
76.
77.     # 1 分评价可视化
78.     x =[]
79.     y = []
80.     for i in range(0,10):
81.         x.append(rating1[i][0])
82.         y.append(rating1[i][1])
83.
84.     sns.barplot(x=x, y=y)
85.     plt.show()
86.
87.     # 2 分评价可视化
88.     x =[]
89.     y = []
90.     for i in range(0,10):
91.         x.append(rating2[i][0])
92.         y.append(rating2[i][1])
93.
94.     sns.barplot(x=x, y=y)
95.     plt.show()
96.
97.     # 3 分评价可视化
98.     x =[]
99.     y = []
100.    for i in range(0,10):
101.        x.append(rating3[i][0])
102.        y.append(rating3[i][1])
103.
104.    sns.barplot(x=x, y=y)
105.    plt.show()
106.
107.     # 4 分评价可视化
108.    x =[]
109.    y = []
110.    for i in range(0,10):
111.        x.append(rating4[i][0])
112.        y.append(rating4[i][1])
113.
114.    sns.barplot(x=x, y=y)
115.    plt.show()
116.
117.    # 5 分评价可视化
118.    x =[]
119.    y = []
```

```
    for i in range(0,10):
        x.append(rating5[i][0])
        y.append(rating5[i][1])

    sns.barplot(x=x, y=y)
    plt.show()
```

注：本题由方琦贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式至以下二维码。

2. 谁是《西游记》的主角？

```
    # 导入包
    import jieba
    import string

    # 放入自定义词典（该文件为自己筛出的词）
    jieba.load_userdict(r"work/西游记分词.txt")

    # 读取文件并分词
    with open("work/西游记.txt", encoding='gb18030') as fin :
        l = []
        d = {}
        txt = fin.read()
        txt = txt.replace('\n', '')
        txt = txt.translate(str.maketrans('', '', string.punctuation+"！ ？ 。。 " # $ % & ' （） * + , － / ： ； ＜ ＝ ＞ @ ［ \ ］ ^ _ ｀ ｛ | ｝ ～ 《》「」、、〃《》「」『』【】〔〕〖〗〔〕〖〗～〝〞〃 〜〒 ☒ ——‘’“”„‟… · ﹏."))

        words = jieba.lcut(txt)
        for w in words:
            d[w] = d.get(w,0) +1
        for k,v in d.items():
            l.append((k,v))
        l.sort(key = lambda e:e[1],reverse = True)
        print(l)

    monkey = ["孙悟空", "美猴王", "齐天大圣", "孙行者", "大师兄", "泼猴", "斗战胜佛", "孙爷爷", "孙大圣", "猴王","悟空","老孙"]
    pig = ["猪八戒", "八戒", "悟能", "天蓬元帅", "净坛使者"]
    sha = ["悟净", "卷帘大将", "沙和尚", "沙悟净", "沙僧"]
    master = ["唐三藏","唐僧","江流儿","玄奘","三藏","旃檀功德佛","金蝉子","陈祎","玄奘法师","三藏法师"]

    # 计算师徒 4 人出现次数
    monkey_count =0
    for i in l:
        if i[0] in monkey:
            monkey_count = monkey_count+i[1]
    monkey_count

    pig_count =0
    for i in l:
        if i[0] in pig:
            pig_count = pig_count+i[1]
```

```
    pig_count

    sha_count =0
    for i in l:
        if i[0] in sha:
            sha_count = sha_count+i[1]
    sha_count

    master_count =0
    for i in l:
        if i[0] in master:
            master_count = master_count+i[1]
    master_count
```

注：本题由徐圆贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

4. 今日热点在哪里?

```
#request 模块是用于获取网络资源的模块，可以使用 REST 操作，即 post、put、get、delete 等操作对网络资源进行存取。
import requests  #导入请求库
#BeautifulSoup4 模块可以按照 Document Object Model Tree 对获取的网络资源进行分层，然后通过 select 方法对各个层次中的细节资源进行获取。
from bs4 import BeautifulSoup  #导入解析库

#在请求网页爬取的时候，输出的 text 信息中会出现抱歉，无法访问等字眼
#headers 是解决 requests 请求反爬的方法之一，相当于我们进去这个网页的服务器本身，假装自己本身在爬取数据。
#在谷歌浏览器搜索:chrome://version/  复制粘贴其中的用户代理部分
headers = {'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.182 Safari/537.36'}

url = 'https://news.baidu.com/'  #引入网址
res = requests.get(url)  #请求访问网址信息
res.encoding = 'utf-8'  #将编码格式转变成中文格式
#print(res.text)  #显示所获取的资源的内容
soup = BeautifulSoup(res.text)  #将文档传入 BeautifulSoup，得到文档的对象

#进入百度新闻页面，随意查看一条新闻的元素（右键——检查），可以发现这些新闻都是位于一个 class 为 mod-tab-content 的<div>内，返回该标签
info = soup.find("div",class_="mod-tab-content").find_all("ul")
#然后查看每一条新闻的具体内容，发现所需内容在标签</a>内
for i in info:
    l = i.find_all("a")
    for j in l:
        site = j.get("href")  #查找网址链接
        title = j.get_text()  #查找新闻标题
        print(title,site)

#需要在电脑或者终端安装相关的库，如 jieba、wordcloud 等
import jieba
from wordcloud import WordCloud
from matplotlib import pyplot as plt
from imageio import imread
```

```
32.
33.  #读取文本数据
34.  text = open('work/新闻中文词云图.txt','r',encoding='utf-8').read()
35.  #读取停用词，创建停用词表
36.  stopwords = [line.strip() for line in open('work/新闻停用词.txt', encoding='UTF-8').readlines()]
37.  #对文章进行分词
38.  words = jieba.cut(text,cut_all= False,HMM= True)
39.
40.  #对文本清洗，去掉单个词
41.  mytext_list=[]
42.  for seg in words:
43.      if seg not in stopwords and seg!=" " and len(seg)!=1:
44.          mytext_list.append(seg.replace(" ",""))
45.  cloud_text=",".join(mytext_list)
46.
47.  #读取背景图片
48.  jpg = imread('work/1.jpeg')
49.  #创建词云对象
50.  wordcloud = WordCloud(
51.      mask = jpg, #背景图片
52.      background_color="white", #图片底色
53.      font_path='work/MSYH.TTC', #指定字体
54.      width = 1500, #宽度
55.      height = 960, #高度
56.      margin = 10
57.  ).generate(cloud_text)
58.
59.  #绘制图片
60.  plt.imshow(wordcloud)
61.  #去除坐标轴
62.  plt.axis("off")
63.  #显示图像
64.  plt.show()
65.
66.  import jieba
67.  from pyecharts import options as opts
68.  from pyecharts.charts import WordCloud
69.
70.  #读入原始数据
71.  text_road = 'work/新闻中文词云图.txt'
72.  #对文章进行分词
73.  text = open(text_road,'r',encoding='utf-8').read()
74.  #选择屏蔽词，不显示在词云里面
75.  excludes = {"今天",'一度','不同','不应','最新','一刻','什么','没有','一枚','高于'}
76.  # 使用精确模式对文本进行分词
77.  words = jieba.lcut(text)
78.  # 通过键值对的形式存储词语及其出现的次数
79.  counts = {}
80.
81.  for word in words:
82.      if  len(word) == 1:   # 单个词语不计算在内
83.          continue
84.      else:
```

```
            counts[word] = counts.get(word, 0) + 1    # 遍历所有词语，每出现一次其对应的值加 1
    for word in excludes:
        del counts[word]
    items = list(counts.items())#将键值对转换成列表
    items.sort(key=lambda x: x[1], reverse=True)    # 根据词语出现的次数进行从大到小排序
    #print(items)    #输出列表

    (
        WordCloud()
        #调整字大小范围 word_size_range=[6, 66]
        .add(series_name="百度热点新闻", data_pair=items, word_size_range=[6, 66])
        #设置词云图标题
        .set_global_opts(
            title_opts=opts.TitleOpts(
                title="百度热点新闻", title_textstyle_opts=opts.TextStyleOpts(font_size=23)
            ),
            tooltip_opts=opts.TooltipOpts(is_show=True),
        )
        #输出为词云图
        .render_notebook()
    )
```

注：本题由方琦贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

第三章 习题

1. **在机器学习中，理解一个概念的最好方法就是用代码计算它。给定波士顿房价数据的训练集和测试集，你能对基于 OLS 的线性回归算法的平均方差和平均偏差进行估计吗？习题内容**

2. **当岭回归和拉索回归的泛化误差特别接近时，从模型易解释性考虑，你应该选择哪个模型？为什么？**

3. **在 sklearn 模块中，程序给出的均方误差往往以负值的形式出现（即负均方误差，neg_mean_squared_error）。猜测一下，这是为什么呢？**

4. **在 3.2.2 中，我们提到了如何对包含调参过程的算法所产生模型的泛化能力进行评估。该流程被称为嵌套交叉验证（Nested cross-validation）。你能基于爱荷华州小麦数据集，对包含调参过程的岭回归、拉索回归和弹力网的泛化误差进行估计吗？（提示：考虑 cross_validate( )函数）**

答案

1. 书中原处已给出答案。
2. 应该选择拉索回归。原因在于拉索回归不仅能够进行参数估计和缩减，还能够进行变量选择。拉索回归通过在损失函数中加入一个 L1 正则化项（绝对值的惩罚项），能够将一些系数精确压缩至 0，从而实现特征的自动选择。这意味着，通过拉索回归，我们可以得到一个更稀疏的模型，即模型中只包含少数几个非零的系数。这种稀疏性使得模型更加简洁。
3. 这是因为在 sklearn 的交叉验证（cross-validation）框架中，评分规则（scoring rule）是按照“越高越好”的原则设计的。也就是说，评分函数返回的数值越高，模型的性能被认为越好。为了适应 sklearn 的“越高越好”的评分框架，将均方误差取负值（-MSE）作为评分指标。这样，模型性能的提高（即均方误差的减小）会导致负均方误差的增大，从而使得算法能够在优化过程中正确地最小化原始的均方误差。简而言之，通过使用负均方误差，将一个最小化问题转化为了一个最大化问题，以便与 sklearn 的评分机制保持一致。
4. 这里给出岭回归的代码，其他同理。

```
1. import pandas as pd
2. import numpy as np
3. from sklearn.model_selection import train_test_split, KFold, cross_validate
4. from sklearn.linear_model import RidgeCV
5.
6. # 读取数据
7. data = pd.read_csv("data/data101272/Iowa.csv")
8. X = data.drop(['Yield'], axis=1)
9. y = data['Yield']
10.
11. # 使用 sklearn 中的 train_test_split 函数拆分训练集和测试集
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=728)

# 定义待搜索的惩罚 lambda
alphas = np.logspace(-5, 2, 100)

# 定义 K 折方法（这里采用整个数据集进行交叉验证）
fold5 = KFold(n_splits=5, random_state=728, shuffle=True)

# 自带调参过程的岭回归
ridgeReg = RidgeCV(alphas=alphas, cv=fold5, scoring="neg_mean_squared_error")

# 使用 cross_validate 评估模型的泛化能力
scores = cross_validate(ridgeReg, X, y, cv=fold5, scoring='neg_mean_squared_error')

# 输出交叉验证的结果
print(f"MSE: {-scores['test_score']}")
```

注：本题由陈心想和董书昊贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

第四章 习题

1. 请以下面的案例画分类器决策树。

某银行给客户是否推荐办理信用卡：卡内月平均资产低于 1 万，不推荐；卡内月平均资产高于 1 万，且每月有大于 5000 的固定收入，推荐；卡内月平均资产高于 1 万，每月没有大于 5000 的固定收入，学历在大专及以上，推荐；卡内月平均资产高于 1 万，每月没有大于 5000 的固定收入，学历在大专以下，不推荐。

2. 某知名企业招聘职员时，考察毕业学校，笔试成绩和发展潜力。

这三项内容中，毕业学校分为 985 大学 1、211 大学 2 和普通大学 3 三个级别，笔试成绩和发展潜力氛围高 1、中 2、低 3 三类。分类为合格 1 和不合格 0。已知 10 个申请人的数据。试用回归法、决策树、随机森林、提升法分别学习一个分类器，比较不同算法的异同。

表 4.2 申请人情况

	1	2	3	4	5	6	7	8	9	10
大学	1	1	3	2	2	3	2	3	3	3
笔试成绩	2	1	3	2	3	1	1	2	2	2
发展潜力	1	2	2	1	1	1	2	3	1	2
分类	1	1	0	0	0	1	0	0	0	0

3. 用 Python 调用 student-por 数据集 ，进行决策树估计。其中，响应变量是教育期望 higher，特征值可根据自己的理论假设在数据库中选择 15 个。

1）随机选择 300 个学生为测试集，估计分类树模型；

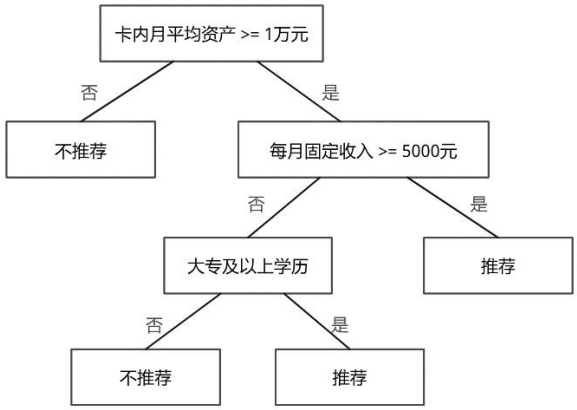
2）通过交叉验证，画图并确定最优复杂性参数；

3）在 2）基础上画分类树；

4）在测试集中预测，评估模型准确率

答案

1. 习题答案



2. 代码

```
1.  import numpy as np
2.  from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import accuracy_score

# 申请人情况
X = np.array([
    [1, 2, 1],  # 大学, 笔试成绩, 发展潜力
    [1, 1, 2],
    [3, 3, 2],
    [2, 2, 1],
    [2, 3, 1],
    [3, 1, 1],
    [2, 1, 2],
    [3, 2, 3],
    [3, 2, 1],
    [3, 2, 2]
])

# 申请人分类结果
y = np.array([1, 1, 0, 0, 0, 1, 0, 0, 0, 0])

# 分割数据集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

# 初始化模型
models = {
    "Logistic Regression": LogisticRegression(),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(),
    "Gradient Boosting": GradientBoostingClassifier()
}

# 训练并评估模型
for name, model in models.items():
    model.fit(X_train, y_train)
    predictions = model.predict(X_test)
    accuracy = accuracy_score(y_test, predictions)
    print(f"{name} Accuracy: {accuracy:.2f}")
```

3. 代码

1）随机选择 300 个学生为测试集，估计分类树模型：

```
import pandas as pd
```

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
import matplotlib.pyplot as plt

# 加载数据集
df = pd.read_csv('student-por.csv')

# 编码 higher 变量
df['higher']=df['higher'].apply(lambda x:0 if x=='no' else 1)

# 选择特征值和响应变量
features = ['sex', 'famsize', 'Medu', 'Fedu', 'studytime', 'failures', 'schoolsup',
            'famsup', 'paid', 'activities', 'absences', 'G3', 'internet', 'Dalc', 'Walc']
X = df[features]
y = df['higher']  # 响应变量

# 对分类变量进行编码
for i in X.columns:
    if X[i].dtype == "O":
        dummies = pd.get_dummies(X[i], prefix = i, drop_first = True)
        X = pd.concat([X, dummies], axis = 1)
        X = X.drop(i, axis = 1)

# 分割数据集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=300, random_state=1)

# 训练分类树模型
tree = DecisionTreeClassifier(random_state=1)
tree.fit(X_train, y_train)

# 展示模型拟合及预测情况
print("训练集 R 方： ", tree.score(X_train, y_train))
print("测试集 R 方： ", tree.score(X_test, y_test))
```

2）通过交叉验证，画图并确定最优复杂性参数：

```
# 设置参数范围
param_grid = {'max_depth': range(1, 20),
              'min_samples_split': range(2, 20),
              'min_samples_leaf': range(1, 20)}

# 使用网格搜索和交叉验证
grid_search = GridSearchCV(DecisionTreeClassifier(random_state=1), param_grid, cv=5)
```

```
grid_search.fit(X_train, y_train)

#打印最优参数
print("Best parameters: ", grid_search.best_params_)
```

3）在 2）基础上画分类树：

```
#使用最优参数重新训练决策树
optimal_tree = DecisionTreeClassifier(random_state=1, **grid_search.best_params_)
optimal_tree.fit(X_train, y_train)

#画出分类树
plt.figure(figsize=(20, 10))
plot_tree(optimal_tree, filled=True, feature_names=X.columns, class_names=['no', 'yes'], rounded=True, fontsize=14)
plt.show()
```

4）在测试集中预测，评估模型准确率：

```
#预测测试集
y_pred = optimal_tree.predict(X_test)

#计算并展示准确率
accuracy = accuracy_score(y_test, y_pred)
print(f"Model accuracy on test set: {accuracy}")
```

注：本题由苏振昊贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

第五章 习题

1. **在本章案例 1 中，选择了 CLDS2016 村庄数据中的经济发展水平与基础设施建设相关的变量作为分类标准，你认为还有哪些变量可以进一步作为村庄分类的标准？原因是什么？**
2. **使用不同的聚类算法、不同的参数设置对你感兴趣的群体进行划分，选出你认为最好的模型，对各个类别进行进一步描述和解读。**
3. **请使用聚类分析方法，尝试对文本数据（如人民日报文本数据、中国知网上的文献摘要数据等非结构化数据）进行分析，并比较上述聚类方法在分析结构化数据和非结构化数据中的异同。**

答案

1. 在本章的案例中，我们结合理论与计算方法，提供了中国村庄在基础设施与经济发展间不平衡的类型学证据。其实，诸如村庄的文化特征（如有无宗族、有无图书馆等）、人口结构（如老中青占比、不同姓氏的占比、外出务工人口占比等）、地理特征（距离县城的距离、是否山区等）等因素均可以作为新的变量，进行进一步的聚类分析，以更深入的了解中国农村之间的异质性与同质性。关于分类的标准，除了上述例子外，还有非常多的变量可以加入，同学们可以发挥自己的社会学想象力，结合聚类分析方法，探析中国的村庄结构及其背后的社会意涵。

2. 我们再次回顾聚类分析的流程，并给出常见的思路。首先，我们拿到了一份新颖的社会数据，并希望从中获取某种社会事实的类型学分析。为此，应首先考察数据的基本特征，它所包含的维度数量，稀疏性，噪声强度，变量分布以及样本数量。在这一阶段，结合降维方法对高维空间进行可视化非常有益，如 t-sne、umap 的算法可以很好地概括数据的局部与全局特征，这些对数据的直观印象可以指导我们选用的适合的距离度量与聚类算法。在对数据整体的状况又所了解后，我们应结合降维的结果对异常数据进行标注或剔除，并考虑使用 PCA 等方法对数据进行提炼。但应该注意，对数据的简化可能损失有益的信息，应针对我们的提问进行考量。

下一步我们需要选择聚类方法，并结合不同的超参数（组合）进行实验，做出性能评估，以选择最佳的超参数。这包括常见聚类指标的计算，对聚类结果的可视化。需要注意的是，大多数聚类算法都具有初始敏感性，因此很多时候应该进行重复实验，并考虑这些性能指标的均值与标准差。最终，结合性能指标、可视化结果以及理论上的可解释性，我们选出恰当的模型，并做出解释，进而发展类型学理论。

读者可循着上述思路，自己选择有趣的数据与议题进行聚类分析的尝试。

3. 对于文本数据的聚类分析，我们仍然要参照习题 2 中的思路，但要额外考虑文本数据的表征问题。对于文本而言，我们尤其关注其背后的语义信息，并需要选择的表征模型，对这些语义进行提取，进而转化为特征向量。表征模型的发展对于文本分析非常重要，这意味着模型能反映原始文本更多的语义细节，按照模型的发展历程排序，主要有词频模型、潜在语义空间模型、主题模型以及词嵌入模型。当然，越精细而准确的文本表征通常依赖于更高维的向量空间用于语义映射。因此，基于表征模型的文本聚类，通常具有稀疏性以及高维度的特征。

针对文本聚类的这些特质，我们有几种主要的策略。首先是放弃一定的细节，使用更为精简的表征模型，或者进行进一步的降维后再聚类，这种方法对于期望的聚类粒度不高的任务已经完全够用了。其次，对于数据量过大的任务，在计算经济性的考虑下，

我们应尽量选用一些针对大规模数据优化的模型，例如 KNN、minibatch-Kmeans 等方法最后，要求更为精细且数据量适中的任务，我们可以使用谱聚类、HDBSCAN 等近年来被证明性能良好的聚类方法，这些方法在大量任务中都有更好的表现，但意味着更高的计算复杂度。

最后，我们会发现，结构化数据与非结构化数据聚类的本质区别是由于数据表征带来的，进而对后续的算法选择带来了整体性影响。

注：本题由刘河庆和刘太石贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

第六章 习题

1. 请回忆卷积神经网络中卷积层和池化层的作用，动手实践案例代码，并回答以下问题：

1) 在某一图片数据集中，图片大小为 500*500，如果使用 5*5 的卷积核进行一次卷积操作，得到图片的大小是多少？在此基础上，如果再隔行进行一次池化操作，得到的图片大小是？

2) 请尝试修改手写汉字识别案例中的 lr 参数，体会学习率的变动对梯度下降和神经网络最终性能的影响。请问过高或过低的学习率分别可能导致模型训练过程和模型结果出现什么问题？

2. 请将文本自动生成案例中使用的模板文本替换成你喜欢的任意文本，并回答以下问题：

1) 请更改模型的最大生成文本长度参数（max_len），你认为文本长度对 GPT2 模型生成的性能有影响吗？如果没有的话，尝试提出一套客观评价文本“好坏”的评价体系。

2) GPT2 模型的核心在于根据统计概率选择有最大可能“合适”的词汇，这类基于庞大统计结果的方法的优缺点在哪里？（提示，可以参考自然语言处理章节进行解答）

答案

1. 对于卷积操作，假设使用 5x5 的卷积核进行卷积操作，不使用 padding，步长为 1，则卷积后图像的大小为： 248x248。

学习率（lr）是神经网络中的一个重要超参数，它控制着参数更新的步长。过高或过低的学习率都可能导致训练过程和模型结果出现问题：过高的学习率会导致参数更新的幅度过大，可能导致无法收敛，甚至发散。在训练过程中，损失函数的值可能会剧烈波动，甚至无法减小。这种情况下，模型无法收敛到合适的解，训练过程可能会非常不稳定。过低的学习率会导致参数更新的步幅过小，训练过程变得非常缓慢，需要更多的迭代次数才能收敛到一个合适的解。如果学习率过低，模型可能会陷入局部最优解，而无法达到更好的全局最优解。此外，训练时间也会因为过低的学习率而大大增加。

因此，选择合适的学习率是非常重要的。通常，可以通过尝试不同的学习率，并观察训练过程中损失函数的变化情况来确定最佳的学习率。

2. 我们将模型的最大生成文本长度参数（max_len）设定为 500。对于 GPT-2 模型来说，文本长度的设定会对生成的性能产生一定影响。较长的文本长度可能会增加模型生成时的计算成本和时间，同时也增加了模型在记忆和逻辑连贯性方面的挑战。较短的文本长度则可能限制了模型生成的表达能力和语境的丰富性。因此，适当地调整文本长度参数可以更好地平衡生成文本的质量和效率。

针对文本生成质量的评价体系可以包括以下几个方面：

- 语法正确性： 生成的文本应该符合语法规则，不应该出现拼写错误、语法错误等问题。
- 语义连贯性： 生成的文本内容应该在逻辑上连贯，不应该出现逻辑矛盾或不相关的内容。
- 信息准确性： 生成的文本内容应该准确反映输入文本的含义或主题，不应该产生误导或错误的信息。
- 语言风格： 根据具体应用场景和需求，生成的文本应该符合相应的语言风格，如正式、轻松、学术等。
- 语境适应性： 根据上下文环境，生成的文本应该能够合理地理解和适应不同的语境和话题。

通过综合考虑以上各方面的评价指标，可以对生成的文本质量进行客观评估，并根据具体需求和应用场景来调整模型参数和优化生成结果。

基于统计概率的方法是自然语言处理中常用的方法之一，其优缺点如下：

1) 优点：

- 语言模型自然性： 基于统计概率的方法能够捕捉到自然语言中词汇之间的相关性和语言结构，使得生成的文本更加自然流畅。
- 灵活性： 这类方法通常能够适应不同领域和语境下的语言特点，具有一定的泛化能力。
- 可解释性： 由于是基于统计概率进行选择，因此生成的结果通常是可以解释的，能够直观地理解生成的原因。

2) 缺点：

- 数据依赖性： 基于统计概率的方法对大量的数据依赖性较强，需要大规模的语料库进行训练，才能够获得准确的统计信息，这限制了模型在特定领域或稀缺数据情况下的应用。
- 词汇限制性： 这类方法通常基于预先定义的词汇表进行生成，无法处理未见过的词汇或者专有名词等，因此在处理一些特定领域的文本时可能会出现词汇缺失的问题。
- 过度拟合： 如果训练数据不够多样化或者模型复杂度过高，容易出现过度拟合的问题，导致模型在生成新文本时缺乏多样性和创造性。
- 无法考虑上下文： 基于统计概率的方法通常只考虑局部的词汇概率，无法很好地捕捉到长距离依赖关系，导致生成结果可能缺乏逻辑连贯性或上下文相关性。

综上所述，基于统计概率的方法在自然语言处理中具有一定的优势，但也存在一些局限性，需要在实际应用中进行权衡和取舍。

注：本题由蒋欣兰和李凌浩贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

第七章 习题

1. 请解释中文词汇处理与英文词汇处理的主要区别，以及中文自然语言处理可能面临的困难。

2. 相较于基于数理统计的词汇处理方法，基于字典与语法规则的词汇处理方法有何优点和局限性?

3. 使用百度自然语言处理 API 对选定文本进行情感分析，计算综合情感倾向和情感得分。

4. 选择几段不同主题的文本，使用文本分类功能将它们分类，并比较不同类别文本的平均情感得分。

答案

1. 中英文词汇处理的主要区别在于中文缺乏明显的词间界限，如空格，这使得中文分词更为复杂。此外，中文中的同一个字在不同上下文中可能有不同的意义，增加了词义消歧的难度。中文自然语言处理可能面临的困难包括歧义词的处理、成语和俗语的理解、以及复杂句子结构的语义分析。

2. 基于字典与语法规则的方法能够精确地识别和解析符合规则的语言结构，适合处理结构严谨的文本。然而，这种方法的局限性在于它依赖于完整且准确的规则库，对于非标准用法或新兴用法的适应性较差，且维护更新规则库的成本较高。

3. Python 代码示例

```
import requests
import json

# 设置 API 访问密钥和 URL
API_KEY = '你的 API 密钥'
ENDPOINT = 'https://api.baidu.com/rpc/2.0/nlp/v1/sentiment_classify?charset=UTF-8&access_token=' + API_KEY

# 定义文本
text = "这是一个非常棒的产品，我非常喜欢"

# 准备请求数据
headers = {'Content-Type': 'application/json'}
data = {
    "text": text
}

# 发送请求
response = requests.post(ENDPOINT, headers=headers, data=json.dumps(data))
result = response.json()
```

```
# 打印情感分析结果
print(result)
```

4. 使用 IMDb 电影评论数据集作为例子。这个数据集含有大量的电影评论及其对应的情感标签（正面或负面）。主要分成以下步骤：

1.数据预处理

加载数据集，并将其分为训练集和测试集。

2.模型训练

使用 scikit-learn 的 MultinomialNB（多项式朴素贝叶斯分类器）训练一个文本分类模型。

3.文本分类

使用训练好的模型对测试集中的文本进行分类。

4.情感分析

对分类后的文本使用 nltk 的 SentimentIntensityAnalyzer 进行情感分析。

5.计算平均情感得分

6.对每个类别的文本计算平均情感得分，并进行比较。

完整的 Python 代码示例：

```
import nltk
from nltk.corpus import movie_reviews
from nltk.sentiment import SentimentIntensityAnalyzer
from sklearn.datasets import load_files
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score

# 下载并加载 IMDb 电影评论数据集
nltk.download('movie_reviews')
movie_reviews_data_folder = nltk.data.find('corpora/movie_reviews')
dataset = load_files(movie_reviews_data_folder, shuffle=False)

# 数据预处理和分割
X_train, X_test, y_train, y_test = train_test_split(
    dataset.data, dataset.target, test_size=0.25, random_state=42)

# 文本分类模型训练
vectorizer = CountVectorizer()
X_train_counts = vectorizer.fit_transform(X_train)
classifier = MultinomialNB()
classifier.fit(X_train_counts, y_train)

# 测试模型性能
X_test_counts = vectorizer.transform(X_test)
y_pred = classifier.predict(X_test_counts)
print(f"Classification Accuracy: {accuracy_score(y_test, y_pred)}")
```

```
# 情感分析
nltk.download('vader_lexicon')
sia = SentimentIntensityAnalyzer()

# 计算每个分类的平均情感得分
sentiment_scores = {category: [] for category in dataset.target_names}
for text, label in zip(X_test, y_pred):
    scores = sia.polarity_scores(text.decode('utf-8'))
    sentiment_scores[dataset.target_names[label]].append(scores['compound'])

average_sentiments = {category: sum(scores) / len(scores)
                      for category, scores in sentiment_scores.items()}
print(f"Average Sentiment Scores: {average_sentiments}")
```

注：本题由蒋欣兰和李凌浩贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

第八章 习题

1. **计算机视觉在经济生产乃至日常生活中有越来越重要的应用，请尝试探索应用计算机视觉技术的场景，并指出可能面临的困难。**

2. **下图是赵孟頫所作《人骑图》（局部）1，历经收藏，已有丰富的题字。请调整汉字书法识别代码中文本框置信度与文本置信度阈值，让 OCR 程序能尽量全面而准确地识别出画面内的落款与题字，并思考：两种置信度在不同场景中应该如何权衡）**

3. **请仿照图片相似度案例的代码，使用 SIFT 算法在图片集 A 和图片集 B 中找出同一个人的照片。（左侧为图片集 A，右侧为图片集 B，图片来自百度 AI 平台）。**

4. **请使用百度 API 对上一题中的人脸进行表情分析，并思考：类似 SIFT 的关键点识别方法能否用于表情分析中，为什么？（提示：请从 SIFT 产生特征关键点的过程思考）。**

答案

1. 计算机视觉（Computer Vision）在经济生产和日常生活中的应用正在日益增加，其应用场景包括但不限于以下几个方面：

- 工业自动化：计算机视觉被广泛应用于工业生产线，用于检测产品质量、执行自动化装配和包装等任务。通过计算机视觉系统，可以实现对生产过程的实时监控和质量控制，提高生产效率和产品质量。
- 智能交通：计算机视觉技术被用于交通监控、智能交通信号灯控制、车辆识别和追踪等方面。例如，交通摄像头可以通过计算机视觉技术实现车辆违章检测、拥堵监测和智能导航等功能，提升交通管理效率。
- 医疗影像分析：医学影像识别技术是计算机视觉在医疗领域的重要应用之一。计算机视觉可以帮助医生识别和分析医学影像，辅助诊断疾病、制定治疗方案，并提高医疗诊断的准确性和效率。
- 智能家居：计算机视觉技术可以应用于智能家居系统，实现人机交互、姿态识别、行为监测等功能。例如，智能摄像头可以通过计算机视觉技术实现人脸识别、姿态检测和智能家居设备控制。
- 零售业：计算机视觉在零售业中的应用也日益增多，包括商品识别、货架监控、购物体验改进等方面。通过计算机视觉技术，零售商可以实现自动化收银、商品库存管理和消费者行为分析，提升营销效果和服务质量。
- 虽然计算机视觉在各个领域都有着广泛的应用，但同时也面临一些困难和挑战：
- 数据质量和标注：计算机视觉的性能很大程度上依赖于数据的质量和标注准确性。然而，获取高质量的大规模数据并进行准确标注是一项耗时耗力的工作，且在某些领域可能存在数据稀缺的问题。
- 复杂环境下的识别：在复杂多变的环境下，例如恶劣天气、光照不足或者背景杂乱的情

[1]https://bkimg.cdn.bcebos.com/pic/95eef01f3a292df50a2f34f6bc315c6035a87361?x-bce-process=image/watermark,image_d2F0ZXIvYmFpa2UyMjA=,g_7,xp_5,yp_5/format,f_auto

况下，计算机视觉系统的性能可能会受到影响，识别准确率下降。

- 隐私和安全问题：随着计算机视觉技术的普及，隐私和安全问题也备受关注。例如，智能监控系统可能会引发个人隐私泄露的担忧，需要制定相应的法律法规和技术手段来保护用户隐私。
- 算法的鲁棒性和可解释性：计算机视觉算法的鲁棒性和可解释性也是一个挑战。在某些情况下，算法可能对输入数据的微小变化非常敏感，导致性能不稳定；此外，黑盒式的深度学习模型也难以解释其决策过程，限制了其在某些关键领域的应用。
- 虽然计算机视觉面临着一些挑战，但随着技术的不断发展和创新，相信这些问题将会逐渐得到解决，计算机视觉技术将会在更多领域发挥重要作用。

2. 在 OCR 程序中，文本框置信度阈值是指系统对检测到的文本框的置信程度，而文本置信度阈值则是系统对文本框内识别出的具体文本内容的置信程度。调整这两个置信度阈值可以影响 OCR 识别结果的全面性与准确性。文本框置信度阈值：如果你发现漏检了一些文本框，你可以尝试降低这个阈值。这样，OCR 程序会倾向于包括那些置信度不是很高的文本框。但这也可能会导致一些非文本区域被错误地标记为文本。文本置信度阈值：如果 OCR 识别出的文本中包含了很多错误，你可以尝试提高这个阈值。这样，只有当系统对识别出的文本非常确信时，它才会将其包括在最终结果中。这通常会提高准确性，但也可能会错过一些置信度较低的正确文本。

在不同场景中权衡这两种置信度时，你应该考虑以下因素：

- 文档质量：如果文档图像清晰，字迹鲜明，你可能会选择更高的置信度阈值，以确保准确性。如果文档质量较差，降低阈值可能更合适，以便尽可能多地检测到文本。
- 文本重要性：如果文本的每一个字都很重要（如法律文件），你可能会希望提高文本置信度阈值，即使这意味着一些文本可能不会被识别。在较少关键的文档中，如草图或脑图，你可以降低这个阈值。
- 后续处理步骤：如果 OCR 后有进一步的错误校正步骤，那么你可以降低置信度阈值，因为后续步骤可以纠正一些错误。
- 处理时间和资源：更低的置信度阈值可能会导致更多的文本被识别，这可能需要更多的处理时间和资源来分析。
- 特定的 OCR 工具和算法：不同的 OCR 工具和算法对置信度的处理可能不同，因此你可能需要根据你所使用的具体工具和算法来调整这些阈值。

在实践中，你可能需要多次尝试不同的阈值，观察结果的变化，并找到最适合你特定场景的平衡点。

4. 代码

```
# 以下示例代码给出了将图片集 A 中的图片与图片集 B 中所有图片进行对比的过程。
!pip install opencv-python==3.4.2.17
!pip install opencv-contrib-python==3.4.2.17
import cv2
from matplotlib import pyplot as plt
import numpy as np
import os
import math

```

```
10.
11.  def getMatchNum(matches,ratio):
12.      """返回特征点匹配数量和匹配掩码"""
13.      matchesMask=[[0,0] for i in range(len(matches))]
14.      matchNum=0
15.      for i,(m,n) in enumerate(matches):
16.          if m.distance<ratio*n.distance: #将距离比率小于 ratio 的匹配点删选出来
17.              matchesMask[i]=[1,0]
18.              matchNum+=1
19.      return (matchNum,matchesMask)
20.
21.  path_a='./data/picture_A/'    # 图片集 A 的路径
22.  path_b='./data/picture_B/'    # 图片集 B 的路径
23.  queryPath_a=path_a
24.  gueryPath_b=patb_b
25.  samplePath_a=path_a+'reference_1.jpg' #样本图片，此处以图片集 A 中的图 1 为例。
26.  comparisonImageList=[] #记录比较结果
27.  #建立 SIFT 特征提取器
28.  sift = cv2.xfeatures2d.SIFT_create()
29.  #建立 FLANN 匹配物件
30.  FLANN_INDEX_KDTREE=0
31.  indexParams=dict(algorithm=FLANN_INDEX_KDTREE,trees=5)
32.  searchParams=dict(checks=50)
33.  flann=cv2.FlannBasedMatcher(indexParams,searchParams)
34.
35.  sampleImage=cv2.imread(samplePath,0)
36.
37.  kp1, des1 = sift.detectAndCompute(sampleImage, None) #提取样本图片的特征
38.  for parent,dirnames,filenames in os.walk(queryPath_b):    # 比较图片集 B 中的图片
39.      for p in filenames:
40.          p=queryPath_b+p
41.          queryImage=cv2.imread(p,0)
42.          kp2, des2 = sift.detectAndCompute(queryImage, None) #提取比对图片的特征
43.          matches=flann.knnMatch(des1,des2,k=2) #匹配特征点，为了删选匹配点，指定 k 为 2，这样对样本图的每个特征点，返回两个匹配
44.          (matchNum,matchesMask)=getMatchNum(matches,0.9) #通过比率条件，计算出匹配程度
45.          matchRatio=matchNum*100/len(matches)
46.          drawParams=dict(matchColor=(0,255,0),
47.                  singlePointColor=(255,0,0),
48.                  matchesMask=matchesMask,
```

```
                flags=0)
            comparisonImage=cv2.drawMatchesKnn(sampleImage,kp1,queryImage,kp2,matches,None,**drawParams)
            comparisonImageList.append((comparisonImage,matchRatio)) #记录结果
    comparisonImageList.sort(key=lambda x:x[1],reverse=True) #按照匹配度排序
    count=len(comparisonImageList)
    column=4
    row=math.ceil(count/column)
    #绘图展示
    figure,ax=plt.subplots(row,column)
    for index,(image,ratio) in enumerate(comparisonImageList):
        ax[int(index/column)][index%column].set_title('Similiarity %.2f%%' % ratio)
        ax[int(index/column)][index%column].imshow(image)
    plt.show()
```

5. 代码

```
import requests
import base64
with open('work/example.png', 'rb') as f:
    img_base64 = base64.b64encode(f.read()).decode()


# client_id 为官网获取的 AK,  client_secret 为官网获取的 SK
host = 'https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id=EhbGU71P4TyBLwfZnnejXvlq&client_secret=XLvcDbbEsmZMGXzCBKSwcOkxNhPySRuI'
response = requests.get(host)
if response:
    access_token = response.json()['access_token']
print(access_token)

"""
人脸检测与属性分析
'''

request_url = "https://aip.baidubce.com/rest/2.0/face/v3/detect"

params = "{\"image\":\""+str(img_base64)+"\",\"\
        "\"image_type\":\"BASE64\",\"face_field\":\"age,beauty,expression,face_shape,gender,glasses,quality,eye_status,emotion,face_type,mask,emotion\"}"
request_url = request_url + "?access_token=" + access_token
headers = {'content-type': 'application/json'}
response = requests.post(request_url, data=params, headers=headers)
if response:
```

```
    print (response.json())
import json
result = response.json()
from PIL import Image
import matplotlib.pyplot as plt
img=Image.open('data/picture_A/sample.jpg')  # 读取图片集 A 中的图片
plt.figure('example')
plt.imshow(img)
plt.show()

if result['error_code'] != 0:
    print("Request Error! Code", result['error_code'], result['error_msg'])
else:
    result = result['result']
    print("人脸个数： ", result['face_num'])
    face_list = result['face_list']
    face_count = 1
    for result in face_list:
        print("人脸 No."+str(face_count))
        print("人脸置信度： ",result['face_probability'])
        print("年龄： ",result['age'])
        print("颜值： ",result['beauty'])
        print("微笑： ",result['expression']['type'], "， 置信度： ", result['expression']['probability'])
        print("表情： ",result['emotion']['type'])
        print("脸型： ",result['face_shape']['type'], "， 置信度： ", result['face_shape']['probability'])
        print("性别",result['gender']['type'], "， 置信度： ", result['gender']['probability'])
        print("是否佩戴眼镜？ ",result['glasses']['type'])
        print("是否佩戴口罩？ ",result['mask']['type'])
        print('---'*10)
```

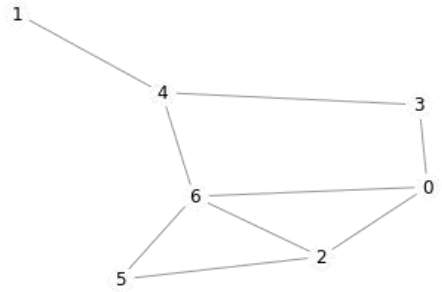
SIFT 较难用于表情识别。首先，SIFT 算法的输入是一对需要对比的图片，输出是图片经过算法比对后的相似度，但表情识别任务中往往没有可供比对的表情。其次，SIFT 提取的关键点是计算机认为特征突出的关键点，而既有表情识别方式（比如百度提供的 API）认为专门定位人脸关键点是表情识别更高效的方式。单纯使用 SIFT 方法生成的关键点不一定具有表情上的内涵，也很难有助于表情识别任务。

注：本题由蒋欣兰、李凌浩、刘金龙贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

第九章 习题

1. 计算中心度

❖ 写出上图对应的邻边矩阵，以及点 6 的邻边列表；计算图的直径；计算每个节点的点度、接近、和中介中心度，并进行标准化。

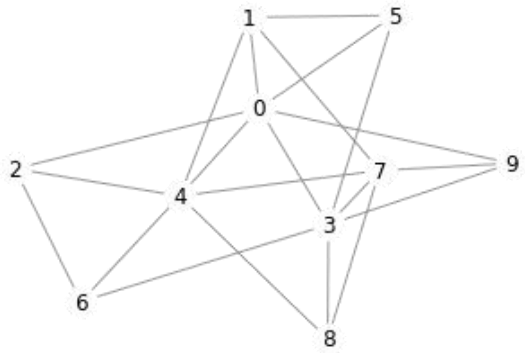


2. 计算密度和集聚系数。

(1) 计算下图的密度。

(2) 以点 0 为顶点的开放三元组、闭合三元组各有几个？计算点 0 的局部集聚系数。

(3) 在阅读 13 章的编程案例之后，用 Networkx 中的函数验证你的答案。



3. 结合所学的专业知识，请问中心度是否能作为一种度量权力的方式?

(1) 如果是，你认为该用哪一种中心度，是否需要对原有算法进行改良，该如何解释这种中心度所测量的权力在现实中的意义?

(2) 如果不是，请说明理由。在权力这个概念中，什么是中心度无法测量的部分?

答案

1. 邻边矩阵：

```
matrix([[0, 0, 1, 1, 0, 0, 1],
        [0, 0, 0, 0, 1, 0, 0],
        [1, 0, 0, 0, 0, 1, 1],
        [1, 0, 0, 0, 1, 0, 0],
        [0, 1, 0, 1, 0, 0, 1],
        [0, 0, 1, 0, 0, 0, 1],
        [1, 0, 1, 0, 1, 1, 0]])
```

邻边列表： [(0, 6), (2, 6), (4, 6), (5, 6)]

图的直径：3

	点度中心度-值	点度中心度-标准化	接近中心度-值	接近中心度-标准化	中介中心度-值	中介中心度-标准化(networkx)	中介中心度-标准化（原定义）
0	3	0.500000	0.100000	0.600000	2.166667	0.144444	0.371429
1	1	0.166667	0.071429	0.428571	0.000000	0.000000	0.000000
2	3	0.500000	0.100000	0.600000	0.833333	0.055556	0.142857
3	2	0.333333	0.090909	0.545455	1.000000	0.066667	0.171429
4	3	0.500000	0.111111	0.666667	5.833333	0.388889	1.000000
5	2	0.333333	0.083333	0.500000	0.000000	0.000000	0.000000
6	4	0.666667	0.125000	0.750000	6.166667	0.411111	1.057143

假设这个图代表的是一个传递信息的网络。谁是网络中对信息的掌控力最强 (一旦移除, 对信息流通影响最大) 的节点, 为什么？谁是网络中能最有效率地获取信息的节点, 为什么？

如上所示，点 6 中介中心度最高，因此对信息的掌控力最强。点 6 的接近中心度也是最高的，因此能最有效率地获取信息。

2. （1）0.47

（2）点 0 的邻边：(0,1), (0,2), (0,3), (0,4), (0,5), (0,9)

开放三元组有 10 个：(0,1,2), (0,1,3), (0,1,9), (0,2,3), (0,2,5), (0,2,9), (0,3,4), (0,4,5), (0,4,9), (0,5,9)

闭合三元组有 5 个：(0,1,4), (0,1,5), (0,2,4), (0,3,5), (0,3,9)

局部集聚系数 = 5/15 = 0.3333

（3）答案略（见 https://github.com/bjcliangMIT/CSS_textbook_SNA）

注：本题由梁晨贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式至以下二维码。

3. 没有确定的答案，但 Bonacich (1987)是一个值得参考的综述。我们需要思考的理论问题是，对于某个研究来说，权力究竟是什么？比如，如果我们认为它是一种地位(status)或者阶级，那或许我们可以像 Podolny (2001)那样，假设认识地位高的人也倾向于有更高的地位，并用类似特征向量中心度的方式计算权力。如果我们认为它是一种仲裁的能力，让人能在鹬蚌相争时渔翁得利, 那或许我们可以用 Burt (1992)的结构洞来测量权力。当你不确定的时候，可以先问自己：在你的网络图中，两点之间的连边有什么现实意义？在这个图中可以顺着点和边流动的是什么——是命令，信息，还是情感?

参考文献；

[1]. Bonacich, P. (1987). Power and Centrality: A Family of Measures. American Journal of Sociology, 92(5), 1170 – 1182. http://www.jstor.org/stable/2780000

[2]. Podolny, J. M. (2001). Networks as the Pipes and Prisms of the Market. American Journal of Sociology, 107(1), 33 – 60. https://doi.org/10.1086/323038

[3]. Burt, R. S. (1992). Structural Holes: The Social Structure of Competition. Harvard University Press. http://www.jstor.org/stable/j.ctv1kz4h78

注：本题由梁晨贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

第十章 习题

1. 如果你要调查所在班级的同学的社交情况，并分析这个社交网络对学习或者找工作的影响。请问：

(1) 你认为如何取样比较好，为什么？

(2) 设计 3-5 个调查问卷访题。结合实际，解释设计的理念，并分析这几个访题分别可能会造成什么潜在的遗漏。

2. 我们会对网络生成和度分布分析的相关概念进行考察和延展。

(1) 在 networkx 的 random_graphs 包中，找到 Watts-Strogatz 小世界网络、Erdos-Renyi 网络、以及 Barabási-Albert 网络的生成函数。

(2) 令 $n = 100,\ k = 4,\ p = 0.3$，生成一个 Watts-Strogatz 网络。阅读 networkx 中相关的文档，解释其中参数 n, k, p 的含义，并汇报这个网络的邻边数量（E_{WS}）。

(3) 根据 Erdos-Renyi 函数在 networkx 中的定义，解释其中参数 p 的含义。计算：如果我们希望生成一个 $n = 100$ 的 Erdos-Renyi 网络，令其邻边数量的期望值为E_{WS}，那么 p 应被设定成什么值？重复 1000 次生成 $n = 100$ 的 Erdos-Renyi 网络并计算邻边数量，验证这个结论。

(4) 给定一个 $n = 100,\ m = 2$ 的 Barabási-Albert 网络，不通过编程的方式，根据其函数定义直接计算出这个网络的邻边数量（提示：这个数量应该是定值）。生成这个网络，验证结论。

(5) 分别用 Shell 和 Spring Layout 画出上述三个网络进行比较，你有什么发现？解释这些发现。提示：因为节点 ID 都是 0-99，我们可以尝试先固定点的位置，再绘制不同网络图的边进行比较。

(6) 创建一个 $n = 3000,\ k = 4,\ p = 0.3$的 Watts-Strogatz 网络$WS_{1000}$。根据③中我们发现的$p$和$E_{WS}$的关系，以及④中我们发现的，Barabási-Albert 网络中m和邻边数量的关系，计算所需的p和m值，以得到邻边数量与WS_{1000}接近的 Erdos-Renyi 网络和 Barabási-Albert 网络。

(7) 用散点图画出三个网络图对应的点度分布图，其中 X 轴为点度，Y 轴为频率。提示：可以对 X 轴 Y 轴分别取 log 再作图。对比三张图，描述和解释它们之间的区别和联系。

3. 除了 NetworkX 以外, SNAP 也是一个常用的网络分析包, 它没有 NetworkX 那么丰富的函数, 但更适用于一些大型网络的分析。在这道题中，我们尝试使用 SNAP 分析包做如下分析:

(1) 从 SNAP-DBLP[2]下载“com-dblp.ungraph.txt.gz”和“com-dblp.top5000.cmty.txt.gz”两个数据包。DBLP 是一个关于计算机领域论文的大型数据库, 我们所下载的这个网络则是基于 DBLP 所建立的一个关于作者的合著（co-authorship）网络[3]。

(2) 解码两个数据包，用“com-dblp.ungraph.txt”生成网络，这个网络 G 应该包含 317, 080 个节点和 1, 049, 866 个邻边。计算 G 的平均集聚系数、闭合三元组（triangles 数量），以及闭合三元组的比例（Fraction of Closed Triangles），并与上述网站进行核对。

(3) 读入“com-dblp.top5000.cmty.txt”文件，这个文件包含了整个网络中最大的 5000 个社区，我们从中得到最大的社区, 并用 subgraph 函数从 G 中提取对应的子图, 命名为 sub_G (n=7556)。然后，我们用 networkx 包中的 greedy modularity 函数挖掘 sub_G 中的社区，并再次提取其中最大的社区，称为 sub2G。提示：因为 greedy modularity 函数假设要图中节点 ID 为 0-N, 而我们的子图中节点 ID 并不是连续的，所以我们需要先用

---

[2] https://snap.stanford.edu/data/com-DBLP.html

[3] J. Yang and J. Leskovec. Defining and Evaluating Network Communities based on Ground-truth. ICDM, 2012.

nx.relabel.convert_node_labels_to_integers 函数重命名 sub_G 中的节点，才能进行社区挖掘。

(4) 在最后得到的 sub2G ($n \approx 1200$) 中，发挥你的创意和审美能力，尝试不同的可视化函数（比如不同的 layout）和其中的参数组合（比如不同的 iteration 值）。图 10.16 是用 greedy modularity 进行了一次社区挖掘并根据社区来涂色的例子。

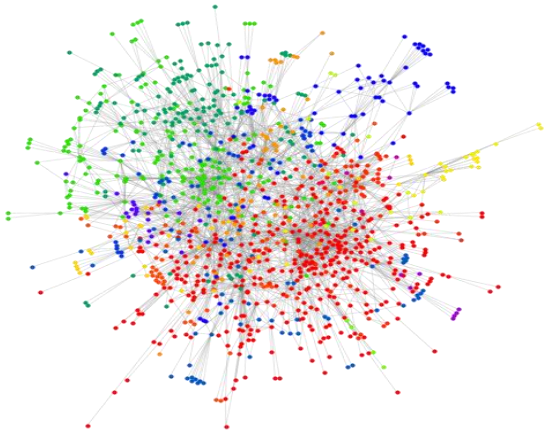


图 10.16 社区挖掘的结果示例

答案

1. 因为一个班人并不多，而且人都比较好找，所以比起要一步一步去问的滚雪球，直接发调查问卷给每个人，问他们跟谁关系比较好更方便。通常来说，我们可以在问卷中得到多个网络的信息，但重要的是问题需要比较具体，以确保回答者对于问题的理解是基本一致的。比如可以问：

1）请列出你平时最经常一起讨论学习的五名同学

2）请列出你觉得在找工作的时候最可能会能帮上忙的五名同学

3）请列出你平时会与其分享个人生活的五名同学。人们讨论重要人生问题的时候，和讨论感情和八卦的时候，经常找的是不同的朋友，因此区分网络关系中承载的各种不同的信息和感情是设计调查问卷中很重要的一环。

注：本题由梁晨贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式至以下二维码。

2. （1）代码

```
import networkx as nx
nx.random_graphs.watts_strogatz_graph(n,k,p)   #Watts-Strogatz
nx.random_graphs.erdos_renyi_graph(n,p) #Erdos-Renyi
nx.random_graphs.barabasi_albert_graph(n,m) #Barabási-Albert
```

（2）答案

- n：网络节点总数
- k：把节点排成一个环形，每个节点与 k 个最近邻居相连
- p：对每个邻边，有 p 的概率改变其所连接的节点
- 因为是无向图，邻边数量为 $\mathrm{n} \times \frac{\mathrm{k}}{2} = 200$

（3）答案

- p: 在所有的节点中任取两点，它们有 p 的概率能组成邻边。这意味着任何两点间的邻边，其生成的概率都是独立的。
- 通过 Erdos-Renyi 网络的生成机制我们知道，对于无向图来说，$E_{WS} = |V|(|V|-1) \times \frac{p}{2}$，即 $p = \frac{2E_{WS}}{|V|(|V|-1)}$。
- 比如我们希望$E_{WS} = 200$,那么 $p = \frac{2\times200}{(100\times99)} = 4/99$

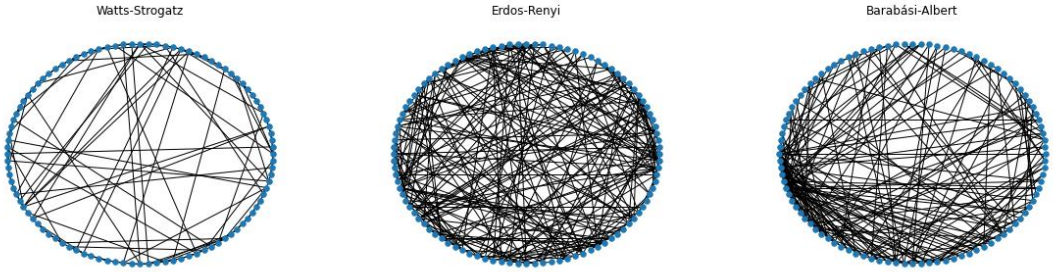
（4）答案

- Barabási-Albert 网络的生成机制以一个邻边数为 0 的网络开始。对每个新加的节点，算法将其与网络中既存的 m 个节点相连，连接概率与既存节点的邻边数成正比。因为这样的生成机制，邻边高的点会有更高的概率与新节点相连。故而会形成“富者越富”的马太效应。
- 以一个邻边数为 0, 节点数为 m 的网络开始, 每次添加一个点就为其连接 m 条边, 重复 n-m 次，直到得到 n 的节点的网络图。故而邻边有 $(n-m) \times m = (100-2) \times 2 = 196$ 条。

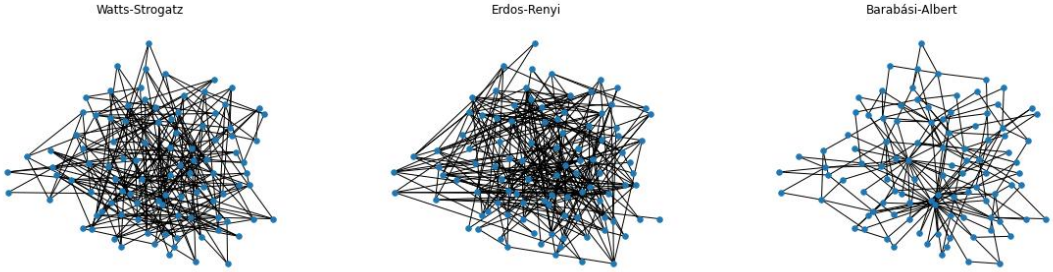
（5）代码

```
WS = nx.random_graphs.watts_strogatz_graph(n=100,k=4,p=0.3)
BA = nx.random_graphs.barabasi_albert_graph(n=100,m=2)

# Shell
fig, axs = plt.subplots(1,3, figsize=(20,5))
pos = nx.shell_layout(BA)
nx.draw(WS, pos, with_labels=False,node_size = 30, ax=axs[0])
nx.draw(ER, pos, with_labels=False,node_size = 30, ax=axs[1])
nx.draw(BA, pos, with_labels=False,node_size = 30, ax=axs[2])
axs[0].set_title('Watts-Strogatz')
axs[1].set_title('Erdos-Renyi')
axs[2].set_title('Barabási-Albert')
plt.show()

# Spring Layout
fig, axs = plt.subplots(1,3, figsize=(20,5))
pos = nx.spring_layout(BA, seed=100)
nx.draw(WS, pos, with_labels=False, node_size = 30, ax=axs[0])
nx.draw(ER, pos, with_labels=False, node_size = 30, ax=axs[1])
nx.draw(BA, pos, with_labels=False, node_size = 30, ax=axs[2])
axs[0].set_title('Watts-Strogatz')
axs[1].set_title('Erdos-Renyi')
axs[2].set_title('Barabási-Albert')
plt.show()
```

Shell Layout



Spring Layout



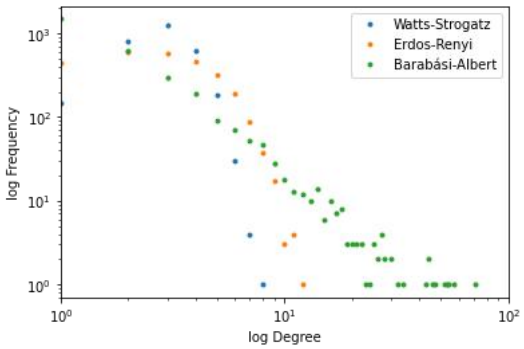
（6）代码

```
WS_new = nx.random_graphs.watts_strogatz_graph(n=3000,k=4,p=0.3)
ER_new = nx.random_graphs.erdos_renyi_graph(n=3000,p=(2*6000)/(3000*2999))
root = math.sqrt(1500**2-6000)
BA_new = nx.random_graphs.barabasi_albert_graph(n=3000,m=round(root+1500))
```

（7）代码

```
counts = Counter(d for n, d in np.array(nx.degree(WS_new)))
b = [counts.get(i, 0) for i in range(1,max(counts)+1)]
x = range(len(b)); y = [z for z in b]
plt.xlabel("log Degree"); plt.ylabel("log Frequency")
plt.loglog(x, y, '.', label='Watts-Strogatz')

counts = Counter(d for n, d in np.array(nx.degree(ER_new)))
b = [counts.get(i, 0) for i in range(1,max(counts)+1)]
x = range(len(b)); y = [z for z in b]
plt.xlabel("log Degree"); plt.ylabel("log Frequency")
plt.loglog(x, y, '.', label='Erdos-Renyi')

counts = Counter(d for n, d in np.array(nx.degree(BA_new)))
b = [counts.get(i, 0) for i in range(1,max(counts)+1)]
x = range(len(b)); y = [z for z in b]
plt.xlabel("log Degree"); plt.ylabel("log Frequency")
plt.loglog(x, y, '.', label='Barabási-Albert')

plt.xlim(1, 100);
plt.legend();plt.show()
```

答案重点在于 Barabási-Albert 图的马太效应：有极少的一些节点，其点度中心度极大



3. （2）

- 平均集聚系数 0.6324
- 闭合三元组数量 2224385
- 闭合三元组的比例 0.1283

代码

```
f = open('/Users/bjcliang/Downloads/com-dblp.ungraph.txt')
[(next(f)) for i in range(4)]
edgelist = [tuple(map(int, sent.split())) for sent in f]

G_nx = nx.Graph()
G_nx.add_edges_from(edgelist)
nx.info(G_nx)

import time
time0=time.time()
print("平均集聚系数:", round(nx.average_clustering(G_nx)), 4)
print("用时", round(time.time()-time0, 4), "秒")
print()
time0=time.time()
triangles = nx.triangles(G_nx)
print("闭合三元组:", sum(triangles.values())/3)
print("用时", round(time.time()-time0, 4), "秒")

#!pip install snap-stanford
import snap
G_snap = snap.TUNGraph.New()
for v in G_nx.nodes(): G_snap.AddNode(v)
for v1,v2 in G_nx.edges(): G_snap.AddEdge(v1,v2)

time0=time.time()
print("平均集聚系数:", round(G_snap.GetClustCf()), 4)
print("用时", round(time.time()-time0, 4), "秒")
print()
```

```
time0=time.time()
result = G_snap.GetTriadsAll()
print("闭合三元组:", result[0])
print("闭合三元组的比例:", round(result[0]/(result[0]+result[2])), 4)
print("用时", round(time.time()-time0, 4), "秒")

f = open('/Users/bjcliang/Downloads/com-dblp.top5000.cmty.txt')
communities = [tuple(map(int, sent.split())) for sent in f]
sorted_communities = sorted(communities, key = lambda x: -len(x))
sub_G = G_nx.subgraph(sorted_communities[0])
nx.info(sub_G)

sub_G_renamed = nx.relabel.convert_node_labels_to_integers(sub_G)
community = nx.algorithms.community.greedy_modularity_communities(sub_G_renamed)
largest_community = sorted(community, key = lambda x: len(x))[-1]
sub2G = sub_G_renamed.subgraph(largest_community)
nx.info(sub2G)

sub2G_renamed = nx.relabel.convert_node_labels_to_integers(sub2G)
community = nx.algorithms.community.greedy_modularity_communities(sub2G_renamed)

import matplotlib.cm as cm #用来获得色卡
plt.figure(figsize=(10, 10)) #  控制图片的大小
pos = nx.spring_layout(sub2G_renamed,iterations=300, seed=100)
cmap = cm.get_cmap('prism')
for i, group in enumerate(community):
    color = np.array([cmap(i/len(community))])
    nx.draw_networkx_nodes(sub2G_renamed, pos,
                           group, node_size=10,
                           node_color=color)
nx.draw_networkx_edges(sub2G_renamed, pos, alpha=0.2, edge_color = 'grey')
plt.axis('off'); plt.show()
```

注：本题由梁晨贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式至以下二维码。更多参见（ai.baiji.org.cn 计算社会科学板块）或 https://github.com/bjcliangMIT/CSS_textbook_SNA?

第十一章 习题

1 请总结本节 11.2.2 的谢林隔离模型中有哪些影响个体搬迁的因素？你认为还有哪些案例中没有提到的因素也会影响人们的搬家决策？请对这些因素进行解释，如果在 ABM 中加入这些因素，你认为可以怎样测量？

2 如果 11.5.4 案例模型中加入现代媒体信息影响，你认为可以是哪些因素？这些因素在 ABM 中如何测量？

3 利用 Python 以下面两个场景构建 ABM 模型，确定模型的主体参数、环境参数和规则

（1）消费者在大型商业广场中选择去什么餐厅就餐；

（2）大一新生第一次参加校园英语角。

提醒：

① 参数应该尽可能多元，体现异质性；

② 特别注意不同主体之间的相互作用规则。

4 与传统的实证社会科学相比，你认为 ABM 方法有哪些优缺点？针对这些缺点，你认为如何弥补？

答案

1. 习题答案

(1) 谢林隔离模型中影响个体搬迁的因素主要包括相似性阈值、城市空置率、居民搬家行为总轮次、城市中居民来源地和城市规模。

(2) 经济因素、教育和医疗资源、社会网络以及政策因素也可能影响人们的搬家决策。

(3) 答案

① 经济因素包括收入、房价、租金、生活成本、就业机会等。这些因素直接影响个人的财务状况和搬迁的决策。

- ✓ ABM 测量：在 ABM 中，可以为每个主体设定一个经济属性，如收入、储蓄等。个体的搬迁决策可以基于其经济状况和目标地区的经济吸引力（如房价、租金、就业机会等）来进行。

② 教育和医疗资源主要指学校质量、医院设施等公共资源的可获得性，家庭可能会为了子女的教育或家庭成员的健康考虑搬迁。

- ✓ ABM 测量：在模型中，可以为每个地区或网格设定教育和医疗资源的属性，如学校评级、医院设施数量等。个体的搬迁决策可以基于这些资源的可用性和其需求来进行。

③ 社会网络涉及到与邻居和社区成员的关系。紧密的社会联系可能会使居民更倾向于留在当前居住地。

- ✓ ABM 测量：在模型中，可以设定主体之间的社交关系，如朋友、邻居等。个体的搬迁决策可以基于其社交网络的强度和稳定性来进行。

2. 答案

（1）可能的影响因素：

①媒体报道信息的及时性和更新频率；

②媒体报道的偏向性；

③公众对媒体发布信息的信任程度。

(2)
①媒体报道信息的及时性和更新频率的测量: 可以根据从事件发生到报道发布的时间间隔设定一个参数表示媒体报道信息的时效性。
②媒体报道的偏向性的测量: 可以基于媒体的历史报道数据或专家评估为每个媒体设定一个偏向性参数，如焦虑、宽慰或中立。
③公众对媒体发布信息的信任程度: 可以基于公众对媒体的信任调查或媒体的历史准确性来为媒体设定一个可信度参数。
3. 答案
(1) 消费者在大型商业广场中选择去什么餐厅就餐
①主体参数：消费者的口味偏好、预算、饥饿程度以及目前所处位置
②环境参数：商场的空间结构和餐厅的位置、菜品类型、价格、口碑
③规则: 消费者根据当前位置和饥饿程度在商场内移动筛选餐厅, 然后消费者根据口味偏好、预算和餐厅的口碑及价格选择餐厅。
(2) 大一新生第一次参加校园英语角
①主体参数：大一新生的英语水平、话题兴趣程度、性格
②环境参数：英语角的主题、氛围、参与者类型、活动的时间
③规则：新生根据对话体的感兴趣程度和英语水平选择是否参加英语角，然后根据英语角活动的氛围、主题、时间和参与者类型决定是否持续参与。
3. 习题答案
(1) 优点
①ABM 具有灵活性，可以模拟复杂的社会现象，考虑多种因素之间的相互作用。
②ABM 具有可重复性，可以多次运行，以检验结果的稳健性。
③ABM 方法允许研究者从微观个体行为出发，通过模拟个体的互动和决策过程，来观察和理解宏观层面的社会现象和规律, 传统实证社会科学方法只能基于现实世界的观察或实验进行研究。
(2) 缺点
① ABM 的模拟结果往往难以通过现实的数据来直接验证, 因为模拟的假设和参数化设置可能与现实存在偏差。
② 复杂的 ABM 模型需要大量的计算资源来运行，这可能限制了模型的规模和复杂性。
③ 模型的构建和参数化过程中存在大量的主观决策和不确定性，这可能影响模拟结果的准确性和可靠性。
(3) 弥补方法：
① 通过融合现实世界的数据，如社会调查、实验数据等，来校准和验证模型参数，提高模型的现实性和准确性。
② 结合多种数据来源来设定模型参数，减少主观性。
③ 通过简化模型结构、优化算法和利用高性能计算资源来降低 ABM 的计算复杂性和资源需求，使模型能够在更大规模和更复杂的环境中运行。

注：本题由陈忱和喻晨玲贡献诸多解答方式的其中一种，欢迎补充、讨论和添加更多解答方式参见（ai.baiji.org.cn 计算社会科学板块）。

参与撰写人员：范晓光（浙江大学）、吕鹏（中国社会科学院大学）
方琦（中国人民大学）、徐圆（中国人民大学）、李瑞（哈尔滨工程大学）
陈心想（中央民族大学）、董书昊（中国社会科学院大学）
陈忱（浙江工业大学）、平玉丽（浙江工业大学）
刘河庆（华中科技大学）、刘太石（华中科技大学）
蒋欣兰（中国社会科学院大学）、李凌浩（中国人民大学）
刘金龙（中央民族大学）、孙宇飞（清华大学）
梁晨（麻省理工大学）、喻晨玲（浙江工业大学