为了更好的阅读体验，请访问我的github主页：
https://github.com/knowsomestringtheory/My-Projects

英雄数据爬虫

by 中国科学技术大学 王宇佳

前言

本程序爬取18183游戏网的王者荣耀板块中的一共101位英雄的数据，最后将爬到的数据整理切割成三大板块：初始属性板块、铭文板块、装备板块以便后续的聚类分析。

```
import re
import os
import time
import random
import pandas as pd
import requests
from bs4 import BeautifulSoup
from sklearn.preprocessing import MultiLabelBinarizer
from urllib.request import urljoin
```

爬取英雄的初始属性以及铭文出装

```
base_url = 'http://db.18183.com/wzry/'
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/83.0.4103.116'}
# 本爬虫爬取的页面为18183游戏网站的王者荣耀板块

def rollarticle(url1):
    html1 = requests.get(url1,headers=headers).text
    soup1 = BeautifulSoup(html1, 'lxml')
    hero_urls = ['http://db.18183.com' + i.find_all('a')[0]['href'] for i in \
                 soup1.find_all('li', {"class": "mod-iconitem"})]
    return hero_urls
#定义rollarticle函数，用来爬取王者荣耀板块所有英雄的详细资料链接

def get_zhuangbei(soup1):
    zhuangbei0 = soup1.find_all('div',{'class':'collocation-list'})[0].find_all('p',{'class':'mod-iconitem-
tit'})
    zhuangbei1 = [i.text for i in zhuangbei0]
    zb_by_who = soup1.find_all('div',{'class':'collocation-box mod-bg'})[0].find_all('div',{'class':'title'})
[0].text
    return zb_by_who, zhuangbei1
#定义get_zhuangbei函数，用来爬取英雄的推荐出装以及推荐出装人

def get_minwen(soup1):
    minwen0 = soup1.find_all('ul',{'class':'glyph-list'})[0].find_all('p')
    minwen = [minwen0[i].text for i in range(len(minwen0)) if i%(len(minwen0)/3)==1]
    return minwen
#定义get_minwen函数，用来爬取英雄的推荐铭文搭配
```

接下来通过rollarticle函数遍历所有英雄页面，通过正则表达式来匹配英雄的各个初始属性，并通过get_zhuangbei和get_minwen函数来匹配英雄的出装和铭文搭配：

```
urls = rollarticle(base_url); results = [];
for jj in range(len(urls)):
    html = requests.get(urls[jj],headers=headers).text
    soup = BeautifulSoup(html, 'lxml')
    hero_name = soup.h1.text
    position = re.findall(r'(.+)s2=(.+)>(\w+)<',\
                          ''.join(str(soup.find_all('a',{'target':'_blank'})))[0][2]
    info = soup.find_all('div',{'class':'otherinfo-datapanel'})[0]
    info1 = [i.text for i in info.find_all('p')];info2 = {};
    info2['英雄'] = hero_name;  info2['位置'] = position
    for i in info1:
        try:
            info2[re.findall('(.+)：(.*)',i)[0][0].strip()]=\
                 re.findall('(.+)：(.*)',i)[0][1].replace('%%','%').strip()
        except:
            info2[re.findall('(.+)：(.*)',i)[0][0].strip()] = None
    results.append(info2)
    info2['出装'] = get_zhuangbei(soup)[1]; info2['出装人'] = get_zhuangbei(soup)[0];
    info2['铭文'] = get_minwen(soup)
    print(jj+1,end=" ")
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108
109 110 111
```

```
hero_info = pd.DataFrame(results)
hero_info.head()
```

	英雄	位置	最大生命	最大法力	物理攻击	法术攻击	物理防御	物理减伤率	法术防御	法术减伤率	...	物理吸血	法术吸血	冷却缩减	攻击范围	韧性	生命回复	法力回复	出装	出
0	蒙恬	战士	3079	1000	168	0	115	16%	50	7.6%	...	0	0	0%	近程	0	76	0	[影忍之足...]	国
1	镜	刺客	3264	460	173	0	87	12.6%	50	7.6%	...	0	0	0%	近程	0	52	16	[追击刀锋...]	C
2	李信	战士	3418	0	176	0	106	15%	50	7.6%	...	0	0	0%	近程	0	52	0	[冷静之靴...]	国
3	蒙犽	射手	3235	0	160	0	90	13%	50	7.6%	...	0	0	0%	远程	0	40	0	[暗影战斧...]	国
4	鲁班大师	坦克	3287	420	166	0	127	17.4%	50	7.6%	...	0	0	0%	近程	0	83	15	[极影...]	国

5 rows × 26 columns

最后，将它输出成csv作为我们的原始英雄信息表：

```
hero_info.to_csv('hero_info.csv', encoding='utf_8_sig')
```

爬取所有铭文的加成效果

```
base_url1 = 'http://db.18183.com/wzry/rune/'
html0 = requests.get(base_url1, headers=headers).text
soup0 = BeautifulSoup(html0, 'lxml')
#铭文信息页面解析

etc_dic = {}; etc_dic1 = {}; etc_dic2 = {};  min_dic = {};
for i in range(len(soup0.find_all('dl'))):
    en_attri = soup0.find_all('dl')[i]['class'][1]
    ch_attri = soup0.find_all('dl')[i].dt.text
    etc_dic[en_attri] = ch_attri
etc_dic['attack_speed'] = '攻速加成'; etc_dic['critical_strike_chance'] = '暴击几率'
etc_dic['mana_regen'] = '法力回复'; etc_dic['health_regen'] = '生命回复'
etc_dic1['point'] = ''; etc_dic1['percent'] = '%';
etc_dic2['red'] = '红色'; etc_dic2['green'] = '绿色'; etc_dic2['blue'] = '蓝色';
for j in etc_dic.values():
    min_dic[j] = 0
#构建字典，以便和前面的英雄初始属性的column names一致

base_url2 = 'http://db.18183.com/api/rune/'
html1 = requests.get(base_url2, headers=headers).text
soup1 = BeautifulSoup(html1, 'lxml')
#解析出铭文页面的json文件

tem = re.findall(re.compile(r'\"\d{4}\":.+?}]}},'),html1); results = [];
for i in range(len(tem)):
    minwen = min_dic.copy()
    min_name = re.findall(re.compile(r'\"name\":\"(.+?)\"'),tem[i])[0].encode('utf-8').decode('unicode_escape')
    min_color = re.findall(re.compile(r'\"color\":\"(.+?)\"'),tem[i])[0].encode('utf-8').decode('unicode_escape')
    min_level = re.findall(re.compile(r'\"tier\":\"(.+?)\"'),tem[i])[0]
    minwen0 = ''.join(re.findall(re.compile(r'\"attribute\":\"[^}]+?\"normal\"}'),tem[i]))
    attri0 = re.findall(re.compile(r'\"attribute\":\"(.+?)\",\"value\":\"(.+?)\",\"unit\":\"(.+?)\"'),minwen0)
```

```
    minwen['铭文'] = min_name; minwen['颜色'] = etc_dic2[min_color]; minwen['等级'] = min_level;
    for j in range(len(attri0)):
        attri = etc_dic[attri0[j][0]]; attri_value = attri0[j][1] + etc_dic1[attri0[j][2]];
        minwen[attri] = attri_value
    results.append(minwen)
#利用正则表达式匹配铭文属性
```

```
minwen_info = pd.DataFrame(results)
minwen_info.sort_values(by=['等级']).head()
```

	物理攻击	物理护甲穿透	攻速加成	暴击几率	暴击效果	物理吸血	法术攻击	法术护甲穿透	最大法力	法力回复	冷却缩减	法术吸血	最大生命	物理防御	法术防御	生命回复	移速	铭文	颜色	等级
24	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	勇气	红色	1
21	0	0	0	0	0	0	1.1	0	0	0	0	0	0	0	0	0	0	斗志	红色	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.0	0	治疗	蓝色	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3.0%	疾行	蓝色	1
15	0	0	0	0	0	0	0	0	0	0	0.2%	0	0	0	0	0	0	应激	绿色	1

最后，将它输出成csv作为我们的铭文原始信息表：

```
minwen_info.to_csv('minwen_info.csv', encoding='utf_8_sig')
```

数据整理

为了方便后续分析，我们需要对爬到的原始数据进行整理切割成三个板块的数据：初始属性板块、铭文板块、装备板块。

初始属性板块

```
ori_hero = pd.read_csv('hero_info.csv',index_col=False)
hero0 = ori_hero._get_numeric_data() #获取纯数值的数据
for i in hero0.columns:
    if len(hero0[i].unique())<=1:
        hero0.drop(i,axis=1,inplace=True)
hero0.drop('暴击几率',axis=1,inplace=True)
#去掉所有英雄的数值全部一样的特征。注意到在初始属性中，孙悟空的被动属性导致其初始暴击几率为20%，
#而其他英雄全部为0。为了不给后续的K-means带来干扰，我们把“暴击几率”这个特征也去掉。
hero0.to_csv('hero0.csv', encoding='utf_8_sig',index=False)
hero0.head()
```

	最大生命	最大法力	物理攻击	物理防御	物理减伤率	移速	暴击效果	生命回复	法力回复
0	3225	3	149	94	0.14	370	2.0	49	0
1	3235	100	151	90	0.13	360	2.0	40	0
2	3178	200	151	86	0.14	370	2.0	45	50
3	3320	420	151	97	0.14	380	2.0	80	15
4	3105	100	152	88	0.13	370	2.0	44	0

铭文板块

```
ori_hero = pd.read_csv('hero_info.csv',index_col=False)
minwen_attri = pd.read_csv('minwen_info.csv',index_col=False)
minwen_numeric = minwen_attri._get_numeric_data()
index = ori_hero._get_numeric_data().index;
columns = ori_hero._get_numeric_data().columns;
minwen = pd.DataFrame(0, index=index, columns=columns)

for i in range(101):
    m1 = ori_hero.iloc[i,-1][2:4]; m2 = ori_hero.iloc[i,-1][8:10]; m3 = ori_hero.iloc[i,-1][14:16];
    for j in minwen_numeric.columns[:-1]:
        try:
            minwen.loc[i,j] = 10*minwen_attri[minwen_attri['铭文']==m1][j].values[0]
            minwen.loc[i,j] = 10*minwen_attri[minwen_attri['铭文']==m2][j].values[0]
            minwen.loc[i,j] = 10*minwen_attri[minwen_attri['铭文']==m3][j].values[0]
```

```
        except:
            pass
# 提取出10个5级铭文提供的属性加成

for i in minwen.columns:
    if len(minwen[i].unique())<=1:
        minwen.drop(i,axis=1,inplace=True)
#去掉所有英雄的数值全部一样的特征。

minwen.to_csv('minwen.csv', encoding='utf_8_sig',index=False)
minwen.head()
```

	最大生命	物理攻击	法术攻击	物理防御	物理护甲穿透	法术护甲穿透	攻速加成	暴击几率	暴击效果	物理吸血
0	0.0	20.0	0.0	0.0	36.0	0.0	0.0	0.0	0.0	0.0
1	0.0	20.0	0.0	0.0	36.0	0.0	0.0	0.0	0.0	0.0
2	0.0	20.0	0.0	0.0	36.0	0.0	0.0	0.0	0.0	0.0
3	0.0	20.0	0.0	0.0	36.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	42.0	0.0	0.0	24.0	0.0	0.0	0.0	0.0

装备板块

由于有的装备含有主动属性（例如很多辅助装备可以主动提升队友的属性）或者提供很多需要配合召唤师使用的增益（例如打野刀的属性与击杀的野怪数量有关），因此不适合将其出装转换成英雄的属性增益。更好的做法是，直接将出装信息作为类别变量保存，然后通过one-hot编码转换成0-1型数据：

```
ori_hero = pd.read_csv('hero_info.csv',index_col=False)
ori_hero['出装'] = [eval(i) for i in ori_hero['出装']]
mlb = MultiLabelBinarizer()
res = pd.DataFrame(mlb.fit_transform(ori_hero['出装']),columns=mlb.classes_,index=ori_hero['英雄'])
#将出装转化为0-1型数据

zhuangbei = res.reset_index(drop=True)
zhuangbei.to_csv('zhuangbei.csv', encoding='utf_8_sig',index=False)
zhuangbei.head()
```

	不死鸟之眼	不祥征兆	冰痕之握	冰霜法杖	冷静之靴	博学者之怒	反伤刺甲	名刀·司命	噬神之书	回响之杖	...	贤者的庇护	贪婪之噬	辉月	近卫荣耀	追击刀锋	逐日之弓	闪电匕首	霸者重装	风灵纹章	魔女斗篷
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0	0	1	1	...	0	0	0	0	0	0	0	0	0	0

5 rows × 54 columns

综合属性

将上述的铭文属性提供的增益加到初始属性中，然后与装备属性合并就得到了我们爬到的所有属性特征的综合表：

```
ori_hero = pd.read_csv('hero_info.csv',index_col=False)
minwen_attri = pd.read_csv('minwen_info.csv',index_col=False)
# 读取原始数据

for i in range(101):
    m1 = ori_hero.iloc[i,-1][2:4]; m2 = ori_hero.iloc[i,-1][8:10]; m3 = ori_hero.iloc[i,-1][14:16];
    for j in minwen_attri.columns:
        if j == '移速':
            ori_hero.loc[i,j] = ori_hero.loc[i,j]*(1 + 10*minwen_attri[minwen_attri['铭文']==m1][j].values[0])
            ori_hero.loc[i,j] = ori_hero.loc[i,j]*(1 + 10*minwen_attri[minwen_attri['铭文']==m2][j].values[0])
            ori_hero.loc[i,j] = ori_hero.loc[i,j]*(1 + 10*minwen_attri[minwen_attri['铭文']==m3][j].values[0])
        else:
            try:
                ori_hero.loc[i,j] = ori_hero.loc[i,j] + 10*minwen_attri[minwen_attri['铭文']==m1][j].values[0]
                ori_hero.loc[i,j] = ori_hero.loc[i,j] + 10*minwen_attri[minwen_attri['铭文']==m2][j].values[0]
                ori_hero.loc[i,j] = ori_hero.loc[i,j] + 10*minwen_attri[minwen_attri['铭文']==m3][j].values[0]
            except:
                pass
#将推荐铭文提供的属性增益添加到英雄的初始属性中

ori_hero['出装'] = [eval(i) for i in ori_hero['出装']]
```

```
mlb = MultiLabelBinarizer()
res = pd.DataFrame(mlb.fit_transform(ori_hero['出装']),columns=mlb.classes_,index=ori_hero['英雄'])
#将出装转化为0-1型数据

hero_final = pd.concat([ori_hero, res.reset_index()], axis=1, sort=False)
hero_final = hero_final._get_numeric_data()
for i in hero_final.columns:
    if len(hero_final[i].unique())<=1:
        hero_final.drop(i,axis=1,inplace=True)
# 去除常数特征

a = list(hero_final.columns[0:17])
b = list(hero_final.columns[17:])
hero_final.insert(loc=0, column='英雄', value=ori_hero.iloc[:,0])
hero_final.insert(loc=1, column='定位', value=ori_hero.iloc[:,1])
display(hero_final.head())
print('王者荣耀总共含有{}个英雄：\n{}'.format(len(hero_final),', '.join(list(ori_hero.iloc[:,0]))))
print('\n这{}个英雄有5个定位：{}'.format(len(hero_final),list(ori_hero.iloc[:,1].unique())))
print('\n英雄一共含有17个数值属性：\n{}'.format(', '.join(a)))
print('\n英雄所有推荐出装中一共包含了{}件装备：\n{}'.format(hero_final.columns.size-19,', '.join(b)))
hero_final.to_csv('hero_final.csv', encoding='utf_8_sig',index=False)
```

	英雄	定位	最大生命	最大法力	物理攻击	法术攻击	物理防御	物理减伤率	移速	物理护甲穿透	...	贤者的庇护	贪婪之噬	辉月	近卫荣耀	追击刀锋	逐日之弓	闪电匕首	霸者重装	风灵纹章	魔女斗篷
0	曜	刺客	3225.0	3.0	194.0	0.0	94.0	0.14	407.0	100.0	...	0	0	0	0	0	0	0	0	0	0
1	蒙犴	射手	3235.0	100.0	196.0	0.0	90.0	0.13	396.0	100.0	...	0	0	0	0	0	0	0	0	0	0
2	裴擒虎	刺客	3178.0	200.0	196.0	0.0	86.0	0.14	407.0	100.0	...	0	1	0	0	0	0	0	0	0	0
3	孙策	战士	3320.0	420.0	180.0	0.0	97.0	0.14	418.0	100.0	...	0	0	0	0	0	0	0	0	0	0
4	芈月	法师	3105.0	100.0	152.0	66.0	88.0	0.13	370.0	0.0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 73 columns

```
王者荣耀总共含有101个英雄：
曜，蒙犴，裴擒虎，孙策，芈月，元歌，张飞，小乔，太乙真人，花木兰，女娲，牛魔，上官婉儿，诸葛亮，宫本武藏，嬴政，哪吒，蔡文姬，干将莫邪，盘古，周瑜，艾琳，后羿，李元芳，雅典娜，明世隐，瑶，鬼谷子，杨玉环，刘备，露娜，东皇太一，高渐离，亚瑟，西施，橘右京，虞姬，米莱狄，王昭君，甄姬，狂铁，云中君，鲁班大师，杨戬，孙悟空，貂蝉，嫦娥，马超，典韦，扁鹊，大乔，姜子牙，蒙恬，狄仁杰，李白，关羽，不知火舞，庄周，安琪拉，妲己，张良，梦奇，达摩，沈梦溪，兰陵王，弈星，韩信，孙尚香，赵云，公孙离，娜可露露，镜，武则天，曹操，鲁班七号，马可波罗，钟无艳，李信，猪八戒，铠，孙膑，阿轲，老夫子，刘禅，百里玄策，司马懿，白起，刘邦，钟馗，吕布，夏侯惇，廉颇，成吉思汗，伽罗，程咬金，黄忠，百里守约，墨子，苏烈，项羽，盾山

这101个英雄有5个定位：['刺客', '射手', '战士', '法师', '辅助']

英雄一共含有17个数值属性：
最大生命，最大法力，物理攻击，法术攻击，物理防御，物理减伤率，移速，物理护甲穿透，法术护甲穿透，攻速加成，暴击几率，暴击效果，物理吸血，法术吸血，冷却缩减，生命回复，法力回复

英雄所有推荐出装中一共包含了54件装备：
不死鸟之眼，不祥征兆，冰痕之握，冰霜法杖，冷静之靴，博学者之怒，反伤刺甲，名刀·司命，噬神之书，回响之杖，圣杯，圣者法典，奔狼纹章，守护者之铠，宗师之力，巨人之握，巫术法杖，影刃，影忍之足，急速战靴，抵抗之靴，救赎之翼，无尽战刃，时之预言，暗影战斧，暴烈之甲，末世，极寒风暴，极影，泣血之刃，炽热支配者，疾步之靴，痛苦面具，破军，破晓，破魔刀，碎星锤，神隐斗篷，秘法之靴，红莲斗篷，纯净苍穹，虚无法杖，血魔之怒，贤者之书，贤者的庇护，贪婪之噬，辉月，近卫荣耀，追击刀锋，逐日之弓，闪电匕首，霸者重装，风灵纹章，魔女斗篷
```

结尾

该程序在18183游戏网上爬到了四份数据：

- hero0.csv：英雄的初始属性。
- minwen.csv：各个英雄搭配推荐铭文时获得的属性增益。
- zhuangbei.csv：各个英雄的推荐出装，以one-hot码（0-1数据）表示。
- hero_final.csv: 上面三个表的汇总表。

接下来，在下一个程序（聚类分析.ipynb）中：我们将针对这四个表做聚类分析，并且通过聚类质量的好坏来分别评估王者荣耀的设计师在如下三个板块中是否对英雄的定位有比较好的区分：英雄的初始属性板块、铭文板块、装备板块。