基于美国大学排名的数据挖掘

辛极 余海林

一. 摘要

随着中国经济的快速发展，越来越多的人感觉中国教育跟不上经济发展。于是最近这几年，各大高校同学纷纷选择出国，导致出国留学人数一直处于快速增长的态势。在出国的国家选择上，几乎没有异议，均以美国为首。但是当具体到某个学校时，似乎就比较困难了，除了前几名的学校，后面一些的学校排名总是不停变化，让学生们很难抉择。所以，我们想了解这些排名究竟是如何产生的，与哪些因素相关，与哪些因素无关，从而分析出各类排名是否准确。

二. 背景介绍

随着中国经济的快速发展，越来越多的人感觉中国教育跟不上经济发展。于是最近这几年，各大高校同学纷纷选择出国，导致出国留学人数一直处于快速增长的态势。特别是近年来，人数更是飞速增长，预计今年将要突破 60 万大关。这与一届毕业生的人数（700 万）比，已经是非常巨大的一个比例了。

在出国的国家选择上，几乎没有异议，均以美国为首。但是当具体到某个学校时，似乎就比较困难了，除了前几名的学校，后面一些的学校排名总是不停变化，让学生们很难抉择。即使目前世界上比较公认的四种大学排名（USNEWS、ARWU、THE、QS），差别也很明显。这不只是是指他们互相之间差别明显，还是指不同年份他们各自的排名也同样差别明显。

所以，我们想了解这些排名究竟是如何产生的，与哪些因素相关，与哪些因素无关，从而分析出各类排名是否准确。当然，我们还希望能够通过输入不同的因素来向同学们推荐适合的学校。

三. 研发现状

目前我们主要实现了关于排名与各因素之间的相关性的一个排序。也就是说，我们可以根据一个排名，分析出这个排名最相关的若干个因素，从而了解这个排名是如何产生的。

未来我们可以实现根据不同的需求，加大相关因素的比例，从而做出我们自己独有的排名，也就是说，根据同学们的需求，对学校进行排名，从而给出推荐意见。

四. 设计

基本设计思路：利用排名，给每一个因素学习出一个系数，通过系数的绝对值大小来显示其相关性。

核心算法：采用线性 SVM 的 rank 功能 3—svm.sh ，实现的源代码网址：https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html。

实现思路：

1. 数据获取和预处理

取自 https://catalog.data.gov/dataset/college—scorecard 的关于美国大学的各项属性表格，更新日期是 March 8, 2017，其中包括 7703 所美国大学，以及其对应的各种属性共 1743 维以及 2017 年 USNews 美国最佳大学排名——综合大学排名，来自 https://wenku.baidu.com/view/a77a4134cdbff121dd36a32d7375a417866fc1bc.html，对

美国前 231 名的大学做了排名。
因为两个来源的数据中大学名字常不匹配（例如 University of California−Berkeley 和 University of California−−Berkeley，或者 University of Alabama 和 The University of Alabama），所以要进行基于经验的消歧。

```
def rep(i, pat1, pat2, head, tail):
    temp = head + rank[i, 1].replace(pat1, pat2) + tail
    if temp in name_list:
        rank[i, 1] = temp

for i in range(len(rank)):
    rep(i, "St. ", "St ", "", "")
    rep(i, "St. ", "Saint ", "", "")
    rep(i, "--", "-", "", "")
    rep(i, "--", " at ", "", "")
    rep(i, "--", "-", "The ", "")
    rep(i, "--", " at ", "The ", "")
    rep(i, "--", " in ", "", "")
    rep(i, "--", " in ", "The ", "")
    rep(i, "--", " ", "", "")
    rep(i, "--", " ", "The ", "")
    rep(i, "", "", "The ", "")
    rep(i, "&", "and", "", "")
    rep(i, "", "", "", "-Main Campus")
```

以上就是我们进行命名匹配的代码片段，rep 函数的任务就是在大学的前面、后面加上字符串，再进行字符串替换，如果这样能够找到对应的名字，就认为匹配成功。如果某个属性在这 210 个学校当中，有不少于 150 个都是数（而不是字符串或者 NULL），那么就把这个属性保留下来。一共 1743 个属性，由此选出 499 个属性。这样做，一方面是为了避开全都是字符串或者 NULL 的无用属性，另一方面是为了不让少量的缺失浪费整个有用属性。缺失的数据由全局平均值补齐。

2. 训练和测试

每个大学对应一个属性向量 x，SVM 初始化一个参数向量 w，两者内积即是该大学的得分。以拿到的排名为依据，优化的目标是排名高的大学得分更高，以此训练 SVM。

此后在测试集上测试效果。测试集包括 6 间学校，是从一开始名字无法匹配的学校里手动抽出来并且匹配名字的，抽的时候基本上是每隔四五十的名次抽一间。结果如下图

The original rank	The predicted score	The predicted rank	University name
5	76.00011164	2	Columbia University in the City of New York
55	53.02081989	50	University of Washington-Seattle Campus
115	41.40878801	118	University of Oklahoma-Norman Campus
150	37.50624646	152	SUNY at Albany
200	26.8471974	220	University of Colorado Denver/Anschutz Medical Campus
228	26.18748815	220	University of Missouri-St Louis

五. 应用实现

主要实现了 svm 的 rank 功能对各因素的赋值 w 向量

分析结果：

最相关因素：

No.1 学科占比（神学和宗教学）（负）

No.2 在 4 年时间内完成学位的学生的比例（正）

No.3 学生上完第一年学之后继续上的比例（正）

还有一些比较重要的因素是：家长的最高学历（正）、教师平均工资（正）、学校在每个学生身上的花费（正）、纬度（正）等等学生上完第一年学之后继续上的比例（正），其中（正）表示系数为正，也就是此项得分越高，总体得分越高，（负）则正好相反。

不相关因素

No.1 学生个人收入

No.2 学生平均负债

No.3 SAT 平均成绩

六. 操作说明

如何运行代码：

```
$ mkdir data
$ python 1-0-parse_name.py
$ python 2-preprocess.py
$ ./3-svm.sh
$ python 4-analyse.py
```

其中 1-0-parse_name.py 为匹配后的大学名称，2-preprocess.py 为选择后的属性，4-analyse.py 为训练出的模型。