基于D3.js的高校排名可视化系统

傅嘉豪　江京　Group 4
Astronaut Center of China (Auditors in PKU)

***Abstract*—本文介绍了利用D3.js开发中国（不含港澳台）高校排名的数据可视化系统。从数据集出发，在分析了数据的结构的基础上，我们以高校所在地区属性为入口，并巧妙地利用了平行坐标轴处理高维数据的优势，使复杂的高维数据变得条理清晰。在界面中我们利用大量的关联和链接，展现不同图表数据之间的联系，改善了读者与界面交互时的体验。同时，通过开发出的可视化系统，我们很容易地发现了中国内地高校的一些模式，具有很好的现实意义。**

I. Introduction

作为培养一流人才的高等院校，因其得天独厚的资源优势，不仅仅被学生、学者和科研工作人员等关心，在社会上也受到了越来越广泛的关注。衡量一个高校的质量，必须多方面地考虑多个因素。每个学校都有自己的优势学科与独有特色，而如何将各个大学、各个学科等这些人们感兴趣的数据，以一个更直观地方式呈现出来，数据可视化自然是不二之选。

A. 数据描述和分析

本文以US News University ranking[1]所提供的中国内地综合排名前20所高校的数据为例，设计了一个中国大学排名的可交互的在线可视化系统。数据集总共有7个维度，分别为School Name, Location, Global Ranking, Global Scores, Subject Name,Subject Ranking, 一共210条数据。显然，数据的各个维度之间存在着依赖关系，比如一个Location对应多个School Name，而每一个School Name对应着一个Location。很明显这些数据之间存在着依赖关系，可以用图1来表示这些数据之间的联系。

根据依赖关系，可以将原始数据表分割为两个满足4NF范式的数据集：(SchoolName, Location, GlobalRanking, GlobalScores)、(SchoolName, SubjectName, SubjectRanking, SubjectScores)。

综合分析七个维度的数据是及其困难的。为选取合适的几个维度综合分析，首先考虑各个唯独数据间的相关性，如图2所示。据图2，显然，(3,4)和（6,7），即(GlobalRanking, GlobalScores)和(SubjectRanking, SubjectScores)间存在极强的负相关性，相关系数分别为$r_{3,4} = -0.9841$, $r_{6,7} = -0.8064$。考虑数据所代表的意义，排名越高（数值越小），分数越高（数值越大），二者存在负相关，这与图2结果一致。故在可视化分析中，我们可以不必将分数和排名同时表现，用户根据所感兴趣的进行切换即可。

[1]http://www.usnews.com/education/best-global-universities/search?region=asia&

B. 可视化方法介绍

对高维数据的可视化处理有很多方法，老师在课堂上主要讲了散点矩阵、极坐标轴图、地形图、星形图、平行坐标轴等等，每种方法都有各自的优劣。由于这次数据涉及到大学和学科的分数和排名，需要对比数据之间的高低，所以我们想到用平行坐标轴去可视化这些数据。对于Location这个属性我们通过把一个圆划分成不同的区域来表示。因此我们的关注点就是学校所处的位置，也就是说先查看位置然后再去关注学校和对应的学科。

平行坐标轴有以下几个优点：

- 能够在平面上展现高于三维的数据；
- 符合人们对图案的一般认知，人们可以很方便的比较不同维度之间的数据；
- 有很好的可伸缩性，每个坐标轴都有着各自的取值范围；
- 可以高效的选择或者删除数据集。

不足之处在用数据太多时，坐标轴会被挤得很近，让人难以观察和理解图案，所以对于过多的数据集处理效果不是很好，同时坐标轴的顺序也不是很好确定，各个维度之间的相关性也不容易展现。

C. *D3.js*简介

D3.js是一个JavaScript库，它可以通过数据来操作文档。D3可以通过使用HTML、SVG和CSS把数据鲜活形象地展现出来。D3严格遵循Web标准，因而可以让你的程序

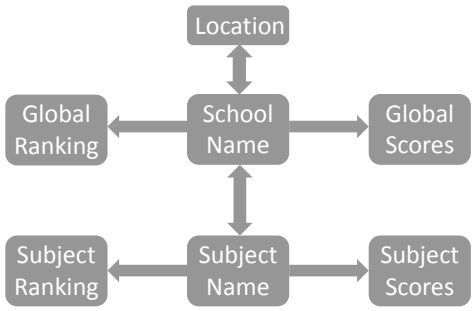


Fig. 1. 数据依赖关系

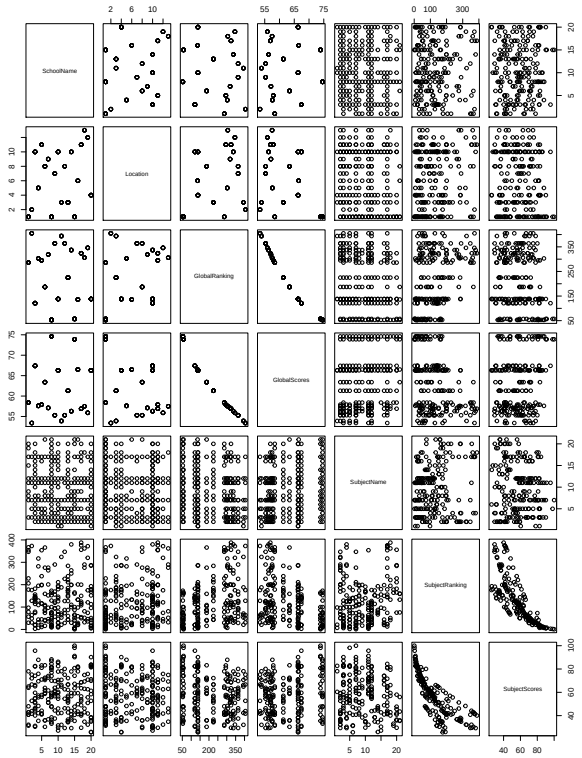


Fig. 2. 数据相关性

轻松兼容现代主流浏览器并避免对特定框架的依赖。同时，它提供了强大的可视化组件，可以让使用者以数据驱动的方式去操作DOM。D3在数据可视化过程中得到广泛运用，特别是各种炫酷的图表，比如折线图、弦图、力导向图等。

II. Method

A. 界面设计

从图1可以看出，对数据的分析一般有三个切入点，分别是Location、School Name和Subject Name，该系统应该从这三个特征中的任何一个出发都可以遍历所有数据。而且这三个角度也是符合人们对大学的一般认知的，比如对于一个刚高考完的考生填报志愿的过程:

- 有的人更看重大学的地理位置，觉得只要大学所处的城市比较好，学校或者学科差点也没事；
- 有的人更看重大学的名气，认为一定要上北大清华交大北航等国内一流名校；
- 有的人优选考虑学科，根据自己的兴趣去选择学科，再去挑选这个学科对应的大学。

所以对于本次数据的分析也可以从这三个角度构建视图。

考虑学校和地域之间的联系，可以用一个视图来表示高校的地域分布情况。这时最佳的可视化方案是地图，在地图上标记出各个学校所在位置，或不同地区包含的高校情况。考虑实际情况，我们没有任何Html开发基础，地图表现难度较大；另一方面，人们对高校所在地域的分布情况是比较熟悉的，而在上海、北京等高校密集的地方，地图上对应区域反而较少，不利于表现高校数目。故采用扇形图（饼图）的形式，展现这20所高校在各个省份的分布情况。

如I-A中所述，在分数和排名中选择一个表现即可。如此以来，去除Location属性后，仍有4个属性需要展现，以排名为例，分别为(SchoolName, GlobalRanking, SubjectName, Subject Ranking)。可讲排名（Ranking）信息归为一类，此时可以采用平行坐标轴(Parallel Coordinates)的方式呈现。一组纵向坐标轴，自左向右依次展示GlobalRanking和各个科目的排名信息，不同的学校可以用不同的颜色表示，以增强区分度。

平行坐标轴的可视化设计，可以很清楚的看到所感兴趣的学校，不同科目间的强弱对比情况。而同一学科间，各个学校的排名情况则不够清楚。为了解决这一问题，针对某个选定坐标轴，再增加一幅直方图，更清楚地表示这个坐标轴上数据，即同一学科，各个学校见的优劣对比。

考虑到各个学校开设的科目不同，一所高校难以涵盖所有的科目，故在平行坐标轴图上一定会有数据缺失。为此，在数据缺失的坐标轴上，采取“过线不描点”的策略，在最邻近的两个数据点之间采用直线连接。另一方面，在用户选择感兴趣的学校后，可以针对这些学校开设的科目，在平行坐标图上用更大的空间来表示。

B. 交互设计

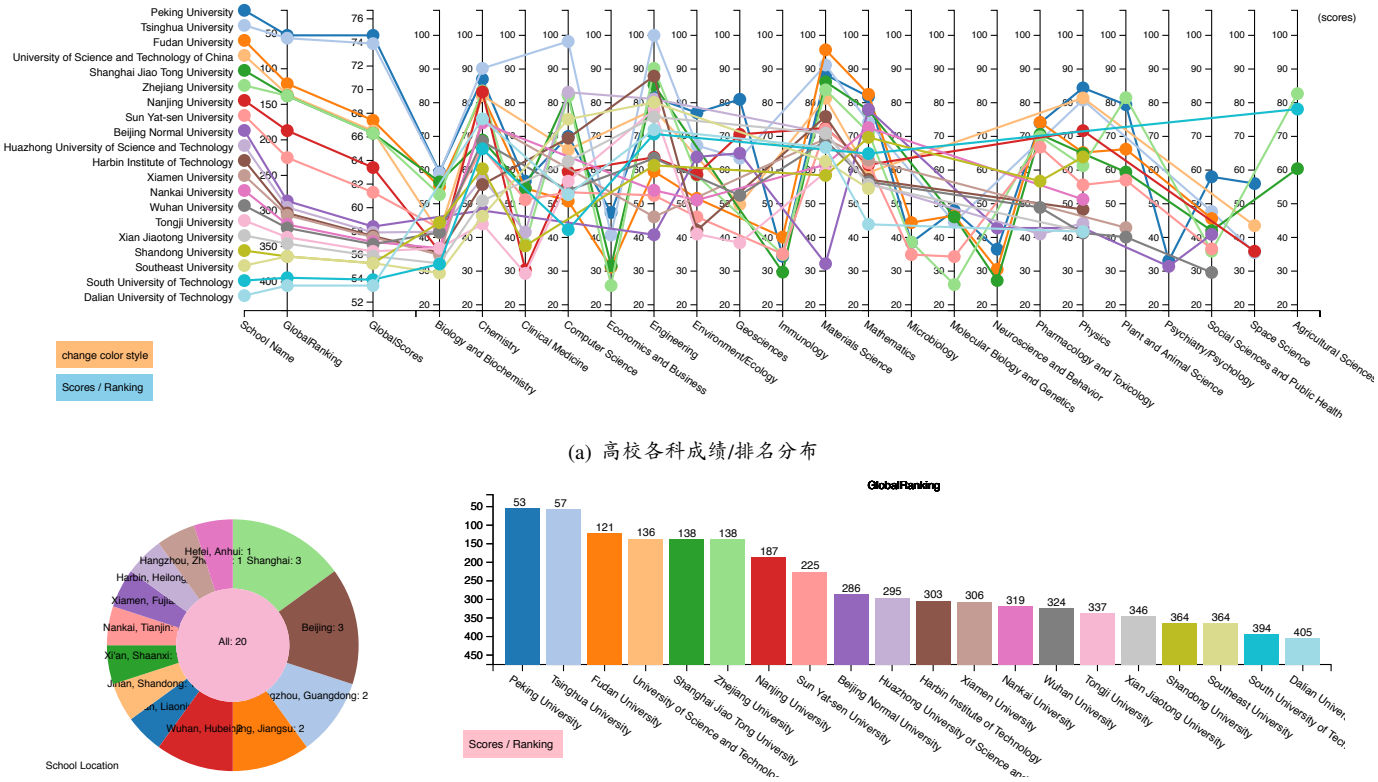
根据实际需求，数据含有20个高校信息，而用户一般只对某几个高校感兴趣。故必须允许用户通过点击选择部分高校，重点表现用户所感兴趣的数据，适当隐藏其他数据。另一方面，三个视图之间存在对应关系，为方便用户，可以在用户将鼠标移动至某一感兴趣数据时，将三个视图中的相关信息同时高亮显示。如用户将鼠标放在扇形图上北京地区对应区域时，该区域高亮，另外两幅视图中所有位于北京的高校也同时高亮。

为清楚地展现平行坐标视图中同一坐标轴的信息，该视图需要实时切换，根据用户点选不断更新。另外，还必须设计两个按钮，用于切换两幅图表内的排名、分数显示。

C. 数据预处理

遍历原始数据文件，发现西安交通大学对应的名称为“Xi’an Jiaotong University”，名称中的单引号(')被程序读入后容易误解析，故预先更改为“Xian Jiaotong University”。

除此以外，为增强该应用的可拓展性，对原始数据未做其他更改。



(a) 高校各科成绩/排名分布

(b) 高校地域分布图

(c) 某一学科（综合）成绩/排名分布

Fig. 3. 应用界面

D. 程序实现

本应用共包含三个视图，故必须创建cake(data), subjectRanking(data), parallelCoordinate(data) 三个函数，用以绘制各个视图。考虑到用户交互，必须在定义各个视图的刷新函数，并通过updateGraphics() 统一调用。

用户在移动/点选鼠标时，被相应的对象捕获，该对象将信息传递给selectedSchool, selectingSchool, selectedProvince, selectingProvince 四个堆栈，随后调用updateGraphics() 函数更新视图。

开发过程中，遇到的一个问题是：平行坐标图中的折线不能直接绑定数据。为解决这一问题，我们将数据简化为一个数字编号，用以代表对应的学校，并存储在每个path 对象的textContent 属性里，并通过this 指针调用。

通过一系列的堆栈，呈递用户交互信息，再进一步开发升级时可以不必关心点选细节，直接从这四个堆栈中调用信息，极大程度地简化了拓展难度。

III. Result

A. 界面和功能

进入浏览器后，该系统的界面如图3所示。

界面上方是一个平行坐标轴图，如图3(a)所示。这张图包含的信息最多，实现起来也最复杂。前两个轴分别是Global Scores和Global Ranking，后面是所有的学科。用户将鼠标移动至学校标签或对应点、折线时，所选的学校会在各个视图中同时高亮显示；此时用户若单击鼠标，则对应学校会在已选学校中添加/去除，对应在所有视图中变亮/变暗。单击坐标轴标签或坐标轴上所有点时，右下视图则自动切换到对应科目，左下角的“chang color style”按钮用来随机切换颜色样式，scores/ranking按钮来切换分数和排名的属性。总之这幅图除了坐标轴不能点击以外，其他各个部分都可以点击交互，并且实现相互关联部分的实时更新。

界面左下角的扇形图如图3(b)所示。这张图是按照数据的Location属性来划分圆的，每个城市所占的扇形的大小是根据所包含的大学的数量来划分的，中间的圆是一个汇总。用不同颜色来区分不同的城市，点击扇形或中间的圆可以向上传递对应的城市所包含的学校。左下角注明了这张图用的是学校位置属性。鼠标移植各个扇形上时，会在各个视图中高亮显示该区域包含的所有学校，单击时则对应学校会在已选学校中添加/去除，对应在所有视图中变亮/变暗。

界面右下如图3(c)所示，显示某一个学科（或综合成绩/排名）对应的所有学校的分数或排名。鼠标移植各个矩形上时，会在各个视图中高亮显示该矩形对应的高校，

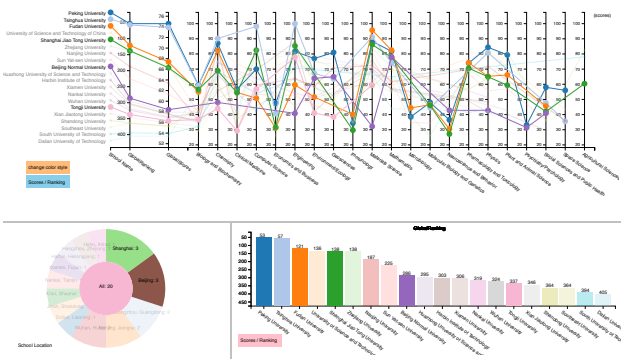


Fig. 4. 结论样例

单击时则对应学校会在已选学校中添加/去除，对应在所有视图中变亮/变暗。点击左下角的“scores/ranking”可以切换分数和排名。

除此以外，再界面右上角增加了“打印”按钮，用户可以将自己所感兴趣的信息通过打印机打印，或通过系统的虚拟打印机存储为pdf文件。

B. 发现的结论

通过设计出来的图形，我们发现了一些有意思的结论。列举部分如下：

- 20所高校中，明显的分为三个层次：前两名北大清华为第一梯队，全球排名在50 60之间；3~9名共6所高校为第二梯队；后11所高校为第三梯队，排名均在270名以后且较为集中（3）；
- 在扇形图中，可见北京和上海两座城市共有六所学校上榜，超过1/4比例。点选北京和上海，发现其学校综合排名整体较高，前五名中有四个学校都位于北京/上海（如图4）；
- 在学科排名上，高校之间出现了一定的共性：如Immunology, Economics and Business, Neuroscience and Behavior等学科得分普遍较低，而Materials Science 得分普遍较高(如图3(a))；

IV. Summary

我们用D3.js画了圆、平行坐标轴和直方图来展示所有的属性下的数据以及数据之间的相互联系，使复杂的数据变得条理清晰。并且巧妙地利用了平行坐标轴处理高维数据的优势，使得在一幅图中就能对比不同学科以及不同学校之间的差异，并用大量的关联和链接展现数据之间的联系，也提升了读者与界面交互时的体验。

A. 优点和不足

本系统的一大亮点，是设计了大量的视图间数据关联交互，强有力地展现了数据各个纬度间的联系。根据用户选择自己感兴趣的学校，重新对视图进行绘制和突出显示，力图增强数据的表现效果。

虽然我们尽了最大的努力来完成这次作业，但是由于能力有限，还与很多地方可以改进。对于地点这个属性，我们用的是一个圆环来表示，这样读者可以获得的信息有限，如果能在一个小地图中显示然后再与其他属性相互关联就会更好地展示地点属性。另一个需要改进的地方就是细节上很多地方不是那么完美，一个突出的问题是文字换行的问题，这样各个图之间就不会显得那么拥挤；学科的分数或者排名直方图到其它图的交互不是那么充分，如果能够点击某个学科的分数之后，平行坐标轴就只高亮显示这个学校的所有学科的折线。

程序方面，因为现在都没有Html, CSS 和JavaScript 开发经验，尤其是对D3.js语法的理解不够，在程序架构设计方便存在很多不足，随着开发的深入，代码结构较为混乱，尤其是在视图绘制的相关函数上。另一方面，后续的视图更新部分，是在部分掌握D3.js和JavaScript基础上进行的，结构性好，易于维护和进一步开发。

B. 收获

整个过程最大的难点在于编程，思路出来了，在实现的过程发现很多问题，但是一旦在过程中去修改一些设计或者增加一些功能，程序就会比较混乱，所以应该在开始的时候尽可能地扩展思路，并预留好相应的程序接口，这样才能更好的完成。同时团队的合作也很重要，团队中的每个人都要尽自己最大的努力，保持好沟通。

Acknowledgements

感谢袁老师的每次授课，让我们对“数据可视化”这一问题有了更深入的了解。感谢助教刘强强和冯璐同学，在本次作业中给我们答疑解惑，帮助我们顺利完成。最后感谢队友间的密切配合，希望我们以后可以做得更好。

References

[1] PKU VisWiki, http://vis.pku.edu.cn/wiki/public_course/datavis_f16/start
[2] D3 API Reference, https://github.com/d3/d3/blob/master/API.md
[3] Learning D3.JS, http://d3.decembercafe.org/