**文献阅读报告**

Visualization Analysis and Design / Tamara Munzner / CRC Press / 2014:第 2 章 What: Data  Abstraction、第 3 章 Why: Task Abstraction

  第2章总结，第二章主要讲了数据可视化的几种常见数据类型，强调了数据类型及其所代表的语义的重要性。通过对数据集的五种基本数据类型：items, attributes, links, positions, and grids，的介绍，进一步说明了我们在对各类数据进行可视化的同时应该多关注如何将数据的含义传达给读者。无论是什么数据类型的数据都不是简单的拼凑，除了表面上的数学含义外，还要关注物理含义，比如：1.3，单纯的从数据角度看这就是一个小数而已很普通，其背后的物理意义可能有很多比如：1.3千克，1.3元，1.3公里等；所以在对数据做可视化的时候，应当多关注其背后的物理含义，比如如果是1.3公里是否可以结合地图去展示，当然这只是我简单的一个思考，这样的案例还有很多。所以我们先了解数据的基本类型和含义对我们在做数据可视化时的一个基础。

那么第二章就针对五种基本数据类型和四种基本的数据集类型（tables, networks, fields, geometry ）展开了介绍，对于基本数据类型和数据集类型的关系文章中给出了图1所示的关系表。

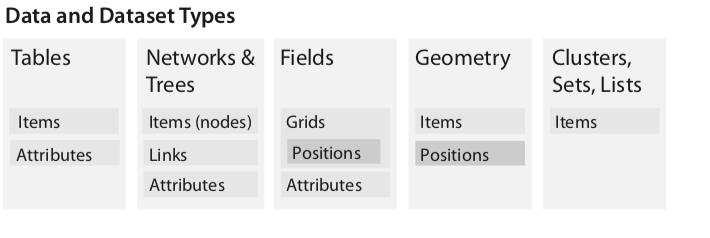


图1:Data and Dataset Types

文中对这个表进行了简单的解释，但是缺少实际的案例，因为我自己根据实际自己日常做项目所接触到的数据，结合着该关系表给出若干解释。

对于Tables这个数据集类型是表，看起来看简单，其实说复杂也是很复杂，看读者怎么将理论去联系实际做延伸。比如我可能想到的就是我们实际项目接触的表，如表1所示，这个表是我们这次大作业要要用到的表，每一个表的属性不同，而且可以看出表的第一列代表的是公司的名称，第二列代表的是公司所在的地理位置，第三列是我们在csv文件或者数据库中给的一个ID，结合着这个表去看图1关系表。除了这个表外，我还查看了很多表，以加深我对文章中简短描述的深入理解，图2是另一个表的截图（由于不在word中画此表，故以截图示意）。对于表1我们可以看出，图2的表复杂了很多，这种复杂不单单是Items和Attributes的增加，更多的是数据的含义更多，所代表的信息也更宽泛，甚至我们可以去看看每个列（属性）之间的关系，比如第p列的学历和第k列的收入，一般情况下肯定是学历高收入高，数据行列变多的同时，数据行和列之间的关系也随之变得更加复杂，我们在进行数据可视化的时候，多留意其背后的物理含义，更能帮助我们达到可视化的目的，做到“信，达，雅”的将信息传递给读者。

表1:Tables示例1

|  |  |  |
| --- | --- | --- |
| 宏东资本 | 安徽 | 1 |
| 翼龙贷 | 北京 | 2 |
| 靠谱鸟 | 北京 | 3 |
| 宜贷网 | 北京 | 4 |
| 爱钱进 | 北京 | 5 |
| 恒易融 | 北京 | 6 |
| 投哪网 | 广东 | 7 |
| ppmoney | 广东 | 8 |
| 钱爸爸 | 广东 | 9 |
| 永银贷 | 河北 | 10 |
| 丰泽财富 | 江苏 | 11 |
| 陆金服 | 上海 | 12 |
| 拍拍贷 | 上海 | 13 |
| 恒大金服 | 深圳 | 14 |
| 蜀丰财富 | 四川 | 15 |
| 台州贷 | 浙江 | 16 |

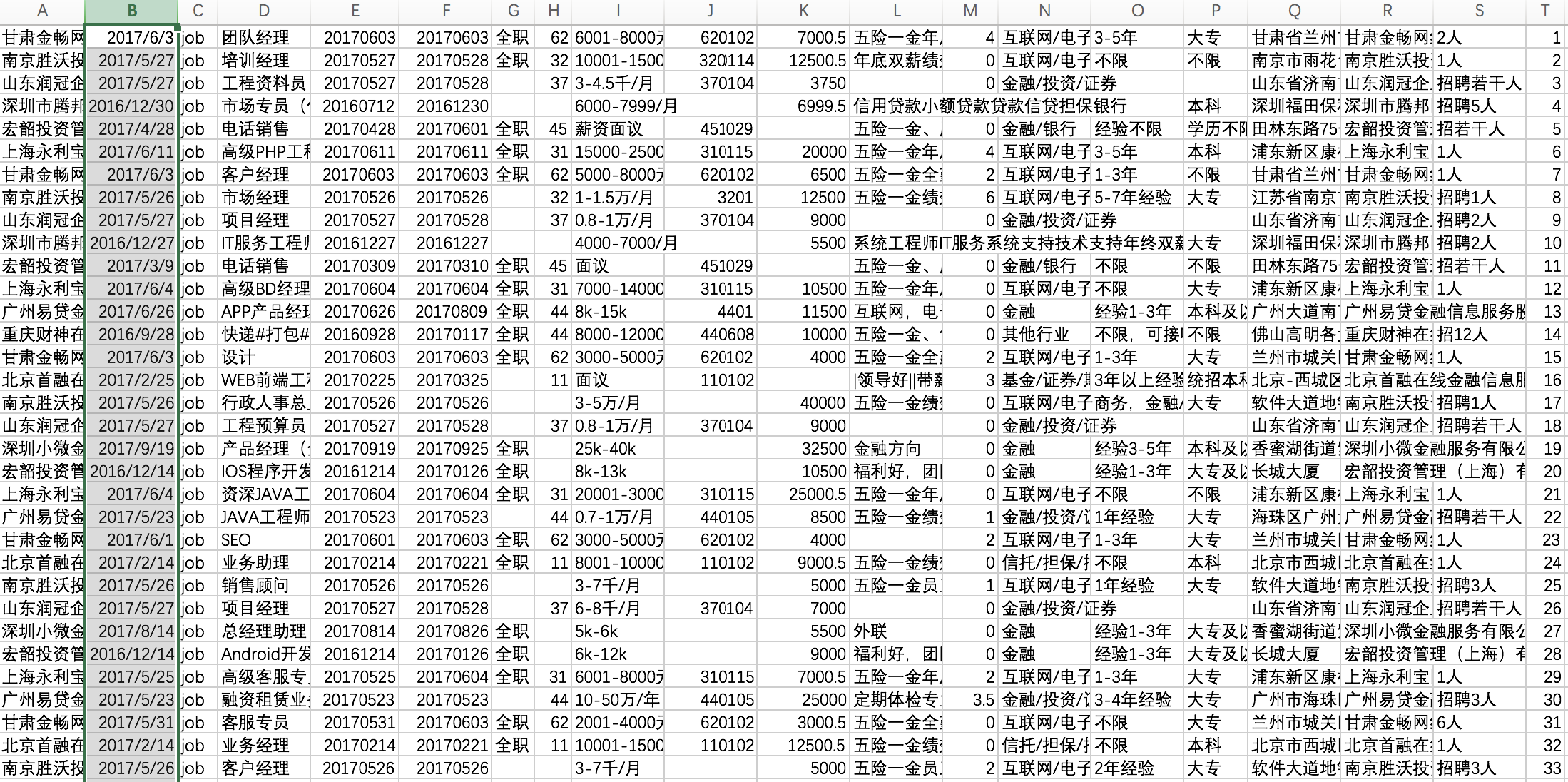


图2: Tables示例2

那么对于网络结构和树结构我理解其实树结构在某种意义上是网络结构的简化结构，是一种特殊结构。树结构我们再熟悉不过，之前我们在修数据结构课程中有很多案例，比如图3所示的就是一个典型的树结构的数据集，我们可以赋予它的含义有很多，比如：顶点是家谱中最长，下面分别是后代，还可以是上面是某个商品的集中加工地，后面均是他们的各种级别分销渠道。在看到这张图时，可以想到的关系有很多，当然这种简单点数据集展现可以传达给读者很多信息，可见我们在做数据可视化时要思考的、留意的、着重展现的有很多。

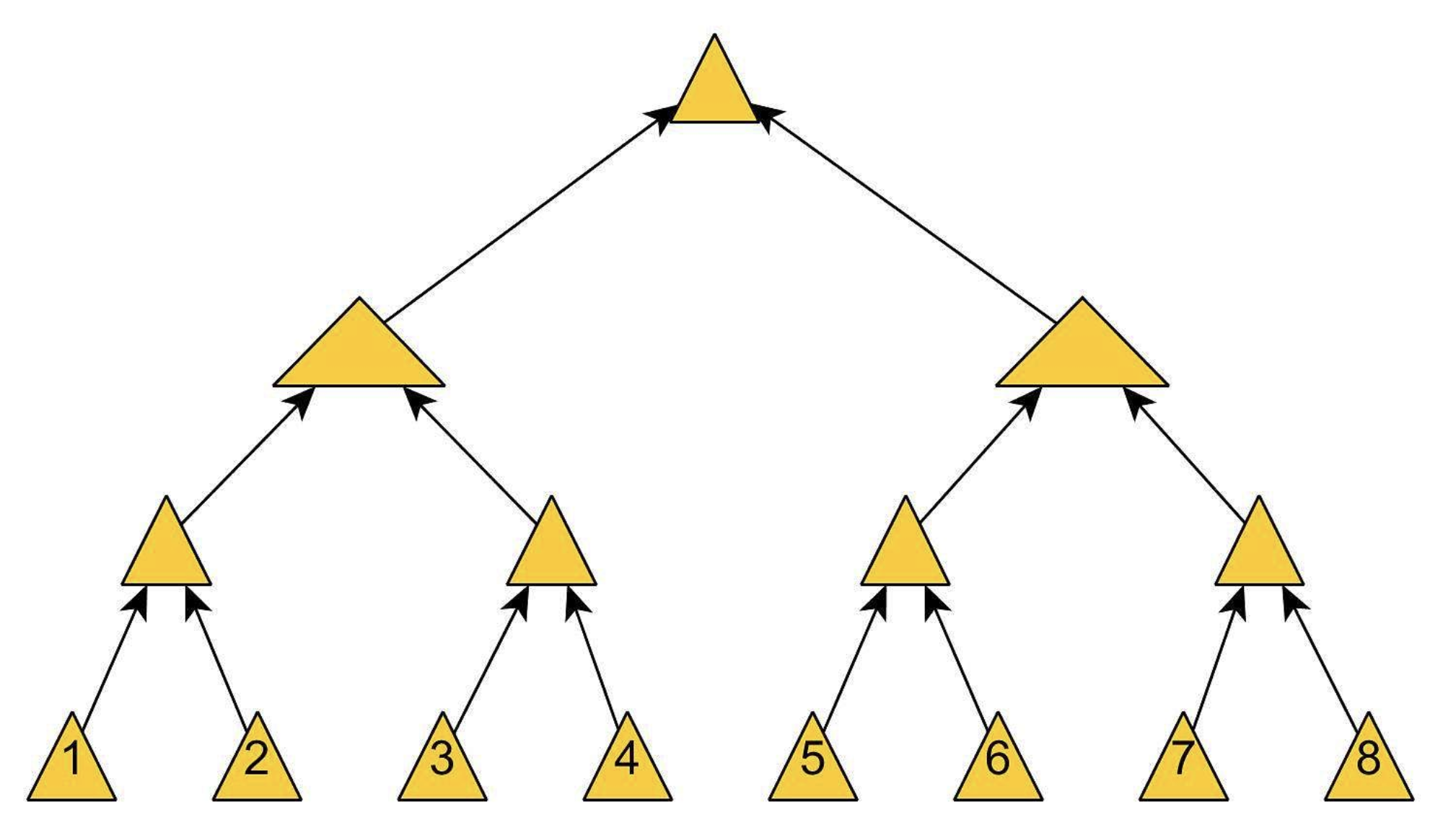


图3:Tree示例图（图片来源于网络）

理解完树结构我又找了一个网络结构的数据集，现在网络结构在深度学习领域可以应用很广泛，图4所示的就是一个很典型的网络结构，我们将可视化的方法进一步扩展到这种模型的结构上，并赋予其不同的含义，以达到将复杂的甚至是无法解释其背后原理的模型很鲜明的可视化出来，这比那些复杂的公示看起来更容易接受，也更能清晰的将网络的搭建展示清楚。

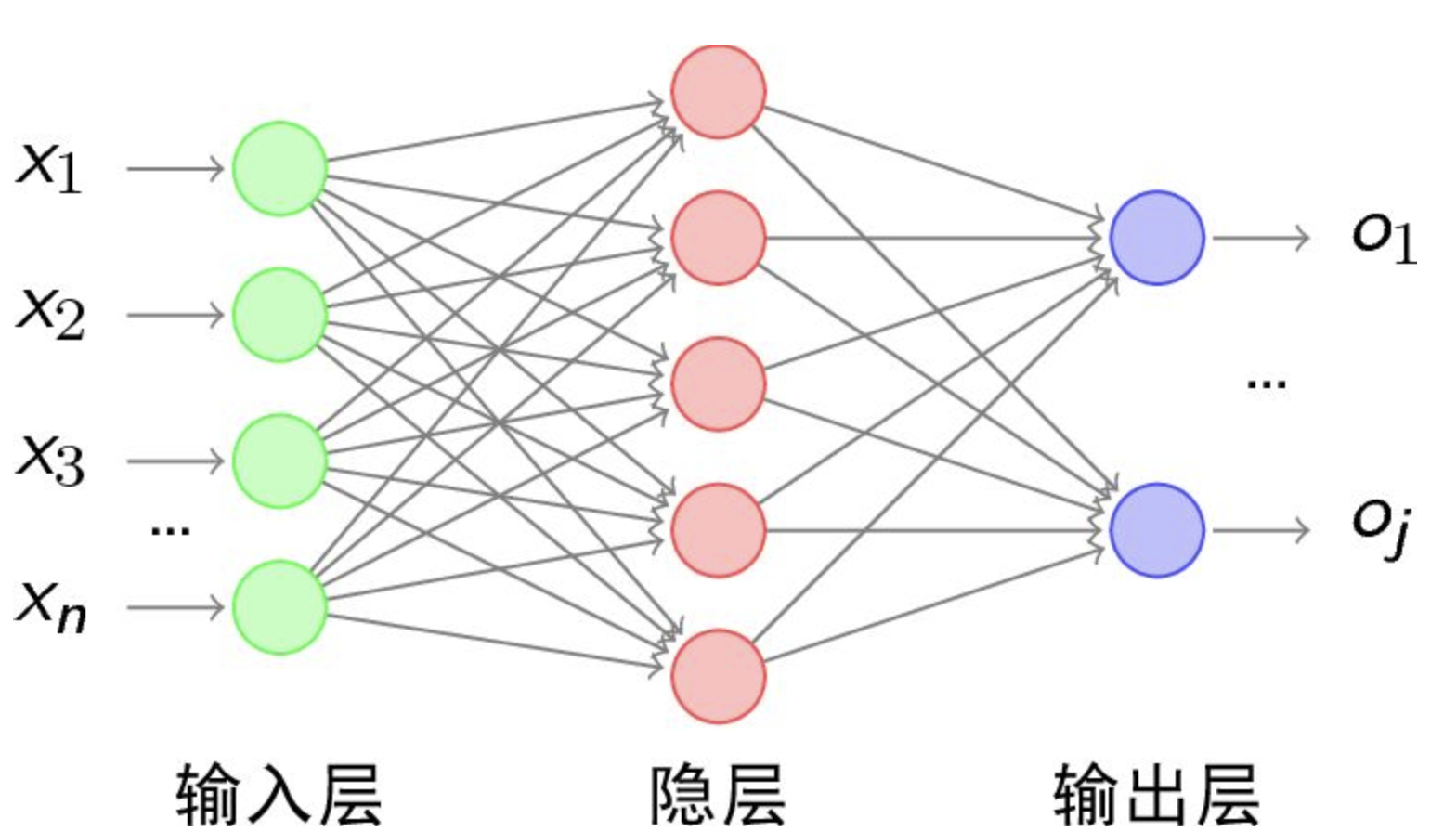


图4:网络结构示例（图片来源于网络）

后面的fields, geometry，我也和我实际项目中的各种可视化相关联进行了思考，在这里不在一一阐述，不过我想说下我对clusters的可视化的理解。因为机器学习的火热，带来了各种各样的聚类分类方法，在算法对数据进行分类或者聚类结束后往往需要可视化出其结果。我们课题组就经常使用一些聚类的算法，当然这里不对算法做讨论，思考的更多的是对分类或者聚类结果的可视化。如图5所示是DBSCAN算法对一堆数据聚类后的可视化图，我们在经过一番调参后，终于得到了好的结果，那么将这些结果如何做到信达雅的传递读者又是一门学问。以前看到这张图可能没太在意其背后是如何进行可视化的，重点都放在了聚类算法上。现在看到这张图，是否要考虑下它是如何可视化的，每个点的半径怎么设置？每个类的颜色怎么分配？

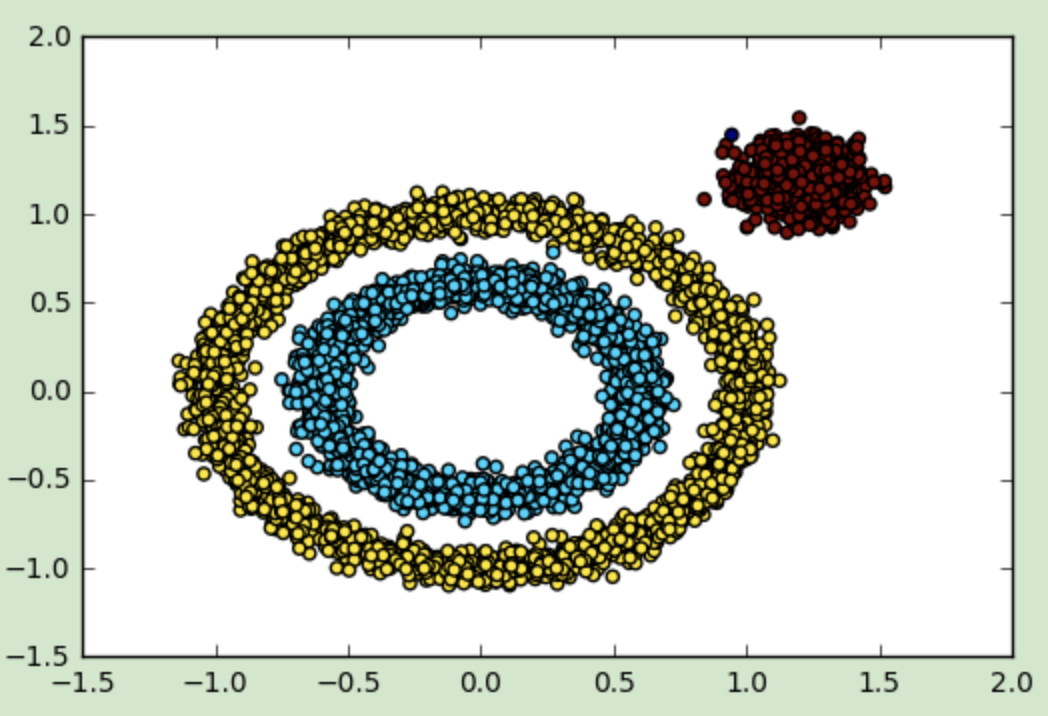


图5:聚类可视化示例图（图片来源于网络）

在第三章中，作者主要在讲述在设计可视化工具时所要注意的具体细节，有很多看起来略微学术的名次，可能需要进一步的理解和解释，最好还是结合具体的案例或者工具去实践理解作者的阐述。

作者说明我们在科学可视化领域为什么要Task Abstractly，对于文章的表述我不在做简单的翻译阐述，下面总结下我读后的理解。其实我们在做可视化工具或设计时像解决很多问题一样，都是问题驱动，然后进一步的去分析、探究最后给出解决方案或者合理的工具设计。文章将这种解决问题的结果分为三个层次，阐述了什么时高级水平的可视化，什么时中级水平的可视化，什么是低级水平的可视化，我个人觉得可以结合刘世霞老师在可视化第一次课时说的“信、达、雅”，去理解，将信息进行转化（可视化）最终要明确需求是什么。针对vis工具的水平标准，文章给出了它的判断理由。最高级别的工具更多的是使用vis consume或者produce信息；中等级别是用户可以根据目标的身份和位置是否已知来对搜索进行分类；低级别是做查询，又可以分为识别目标、比较目标和总结目标。文中所提到的设计者和使用者的关系，简单理解就是设计者多站在使用者的角度考虑问题，毕竟是工具类产品。像TABLEAU就是一款很好的可视化工具，同时也比较通用，像文章提到的同一个问题在不用的领域有不同的表达，在这可以得到不错的解决，因为它就是通过用户交互来决定输入是什么，输出是什么，最终的展示是什么等等。这也是目前比较常用的一款数据可视化软件，我觉得文章不止这些观点可以结合这个具体的工具去理机，还有很多，比如提到的交互设计、数据产生数据等等，它给出的示例图就有很多是TABLEAU做出来的，可见这个工具在一些方面时文章第三章某些理论的实践。

下面我对文中使用的TABLEAU案例做些补充，理论联系实践，补充最新的一个TABLEAU PREP可视化工具案例。Tableau Prep 是一款直观、简洁和智能的工具，可有效清理和组织数据以供分析。像文中所说，此工具很多数据分析都是手动去做枯燥的操作，给出了一个功能，以设计者的角度来解决使用者的问题。通过flow -> step(第一个:input step) input格式：excel 储存flow会直接储存成为flow文件，不会产生真正的数据；在输入端input：添加表格：Clean\_data/ ABA Bestsellers 02-28-18.xlsx -> handcover Nonfiction 浏览表格 添加filter；profile: 打开profile\_pane.tflx,选择其中一个值，这个值对应的其他field被点亮。 outlier detection：从视图中可以容易地被看出; clean: 打开the\_cleaning\_step.tflx 重命名、删除card、更改数据类型 custom split：file path板块 -> custom split -> 用"."split clean:选择新产生的card -> clean -> remove letters -> trim spaces filter: 更改table names为list　->　calculation -> CONTAINS([list], "Nonfiction") -> undo(向后箭头) Add calculated field：Author -> create calculated field -> Author: REGEXP\_REPLACE([Author], "&", "and") (在原有的基础上改，不新建field);group 和replace：1.打开group\_and\_replace.tflx -> 对 Airbnb listings这个field：group and replace -> manual selection -> 将所有bath相关的选中成为bath组 -> 对bedroom做相同操作；2.根据单词拼写自动group：对misspellings实施操作pronunciation 3.Common Character: 通过公共字符来辨别是否被group：对第三个field实施common characters，姓名顺序不同的被group；Aggregate: 打开the\_aggregate\_step.tflx -> 橘色添加aggregate field -> 目的：现有交易信息和书的best seller信息，希望能得到book-level的平均折扣和交易次数 -> 拖动ISBN到grouped fields -> 拖动transaction Id 和 discount到aggregate fields -> 修改discount为average -> 添加clean step -> 下面部分是表格;join: 打开the\_join\_step.tflx (电影和书) -> add join -> 将两个都连接到join上 -> 添加条件（Author - Author / Title = Title） -> 在join type中选择left或是right join -> summary of join result在变化 -> 红色没有match 黑色match;Union: 打开the\_union\_step\_finished.tflx -> 选择union -> 数据来源于四个表 -> 查看下方的表格看细节 -> 如果想要删除，不能在union里删除，要新建一个clean step。

以上是我补充的案例，从这个工具来看，都很符合文章中的观点，由于这些功能的截图太多，所以我只给出了文字，具体可以参考网上很多视频。