数据科学研究——Python 编程行为分析

摘要：2020 年春季学期，陈振宇老师使用其团队自主开发的慕测平台（Mooctest）为 2018 级本科生提供了在线 Python 编程练习的机会。本组通过陈振宇老师提供的数据进行了 Python 编程行为的研究与分析，包括使用熵值法和 TOPSIS 方法数据化题目难度、使用层次分析法和 TOPSIS 方法探究个人能力、使用 Rasch Model 可视化题目难度、题目质量和学生能力的总体分布情况。

关键词：熵值法、TOPSIS、层次分析法、Rasch Model、Python。

小组成员：

成员	学号	邮箱	完成题目	任务
朱金宇	181250211	181250211@smail.nju.edu.cn	199	个人编程能力分析
奚志恒	181250162	181250162@smail.nju.edu.cn	200	题目质量、学生能力分布分析
魏荣来	181250152	181250152@smail.nju.edu.cn	200	难度分析

2020 年 7 月 23 日

目录

概述

研究背景

疫情冲击下，南京大学软件学院“数据科学基础”课程采取线上教学模式，陈振宇老师使用其团队自主开发的慕测平台（Mooctest）为 2018 级本科生提供了在线 Python 编程练习的机会。

陈振宇老师及其团队为同学们准备了八类题目，包括数字操作、字符串、数组、树结构、图结构、查找算法、排序算法、线性表，难度各有不同，具有一定的参考价值。本门课程中共有 262 人参与了学习，分成 5 组。

在实施在线编程练习后，题目的难度分布如何？题目质量分布如何？学生能力分布如何？如何数据化和可视化题目难度？每个学生对不同类型题目的掌握程度如何？本次研究，本小组成员希望通过慕测系统产生的 Python 练习数据进行分析，期望通过研究学习者与题目两个方面解答上述问题。

研究问题与目标

1. 在学生提交记录的基础上，数据化题目难度。
2. 在已获得的题目难度基础上，探究个人能力。
3. 可视化题目难度、题目质量和学生能力的总体分布情况。

研究数据源

学生提交到慕测平台的数据，由陈振宇老师团队提供。

数据源处理见附录。

研究方法

1. 使用熵值法和 TOPSIS 方法数据化题目难度。

2. 使用层次分析法和 TOPSIS 方法探究个人能力。

3. 使用 Rasch Model 可视化题目难度、题目质量和学生能力的总体分布情况。

开源地址

https://github.com/JinyuChata/datasci-coursework

文件结构见附录。

分析题目难度

熵值法简介

在信息论中，熵是对不确定性的一种度量。信息量越大，不确定性就越小，熵也就越小；信息量越小，不确定性越大，熵也越大。根据熵的特性，我们可以通过计算熵值来判断一个事件的随机性及无序程度，也可以用熵值来判断某个指标的离散程度，指标的离散程度越大，该指标对综合评价的影响越大。

因此，可根据各项指标的变异程度，利用信息熵这个工具，计算出各个指标的权重，为多指标综合评价提供依据。

TOPSIS 简介

TOPSIS （Technique for Order Preference by Similarity to an Ideal Solution ）法是 C.L.Hwang 和 K.Yoon 于 1981 年首次提出，TOPSIS 法根据有限个评价对象与理想化目标的接近程度进行排序的方法，是在现有的对象中进行相对优劣的评价。TOPSIS 法是一种逼近于理想解的排序法，该方法只要求各效用函数具有单调递增（或递减）性就行。TOPSIS 法是多目标决策分析中一种常用的有效方法，又称为优劣解距离法。

其基本原理，是通过检测评价对象与最优解、最劣解的距离来进行排序，若评价对象最靠近最优解同时又最远离最劣解，则为最好；否则不为最优。其中最优解的各指标值都达到各评价指标的最优值。最劣解的各指标值都达到各评价指标的最差值。

原因

方便操作

该模型可以在已知数据情况下，估计出题目的难度系数，方法简单易操作，同时有较大的参考价值，很适合 OJ 系统的题目难度评定系统。如果能提供更多的参考角度，同时提供更大的数据，模型的准确性会提升很多。

面向测试用例不影响判断

上交的题目答案有许多面向测试用例的编程，表明题目太难，只能想到这个做法，但是，面向测试用例编程只会影响均分和提交次数，而面向测试用力的提交，使得均分降低，提交次数提升，这两个方面在模型中都会提升题目的难度，所以面向用例并不会对实际的题目难度有太大影响。

实践

问题假设

假设一

题目都是互联网上的题目，没有原创题，即可以通过搜索引擎找到原题及答案，但是搜索难度未知。

假设二

提交时间是写代码的风格习惯问题，所以提交时间并不考虑在题目的难度分析中。

假设三

题目答案包含 Python 及 C++，都是用较为标准的答案格式，所以答案代码行数在一定程度上可以反映题目本身的难度，即答案代码越长，题目难度越高。

假设四

分析出的题目难度，属于题目在允许使用搜索引擎的情况搜索答案的情况下，得出的难度，所以难度中不仅包含题目本身的难度，也包含答案是否易搜索等因素。

模型的建立与求解

熵值法赋权

Step1 对原始数据矩阵按列进行归一化处理，归一化方法不唯一，在此使用比值归一化。

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}}$$

Step2 计算各指标的熵值：

$$e_j = -k \sum_{i=1}^{n} p_{ij} \ln p_{ij}, (j = 1,2,\ldots,m)$$

其中 **k** 与样本数量有关，常取$k = \frac{1}{\ln n}$此外，补充定义：若 $p_{ij} = 0$，则令$p_{ij}\ln p_{ij} = 0$

Step3 计算各指标的权系数：

$$h_j = \frac{1 - e_j}{\sum_{k=1}^{m} 1 - e_k}, (j = 1,2,\ldots,m)$$

熵权系数h_j越大，则该指标代表的信息量越大，表示其对综合评价的作用越大。

熵值法分析结果

详细代码和文件见附件，此处只展示结果

	均分占比	提交次数占比	测试用例占比	答案行数占比

全部	0.03649	0.14904	0.15127	0.66318

由此可见，均分和提交次数所占权重很小，而测试用例个数和代码行数所占权重极大，这也是比较符合显示数据的。

一方面，提交的记录在分数上，有的同学喜欢先测试调试代码，等到满分之后再提交，这样导致有的同学只有 100 分的提交，造成题目的均分差别不大，所以在题目难度方面，这个指标的信息含量较小，较难真正反映题目的解决难度。

另一方面，答案代码行数权重较大，答案的长短属于题目本身的属性，不会受其他因素干扰，同时，答案的长短离散程度较大，能更好的区分不同的题目难度。

剩余两个指标的权重较小，较为均匀，因为这两个指标的人为因素较为明显，能较为片面的反映题目的解决难度，比如，测试用例考察代码复杂度，从而反映题目的解决难度，提交次数考察题目的边界值，编程细节，从而反映题目的解决难度。

综上，使用熵值法测定的权值较为准确。

TOPSIS 求解

step1 指标正向化

TOPSIS 法使用距离尺度来度量样本差距，使用距离尺度就需要对指标属性进行同向化处理（若一个维度的数据越大越好，另一个维度的数据越小越好，会造成尺度混乱）。通常采用成本型指标向效益型指标转化（即数值越大评价越高，事实上几乎所有的评价方法都需要进行转化），将极小型指标，中间型指标，区间型指标都转化为极大型指标。

本题只有一个极小型指标，均分，其余均为极大型指标，故在此只展示极小型指标正向化的方法

$$x' = M - x$$

其中，M 表示 x 可能取得的最大值

step2 构造归一化初始矩阵

共有 n 个评价对象，每个对象有 m 个指标，则原始数据矩阵构造为

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{21} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

构造加权规范矩阵，属性进行向量规范化，即每一列元素都除以当前列向量的范数（使用余弦距离度量）

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}}$$

由此得到归一化处理后的标准化矩阵 Z:

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{21} & \cdots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nm} \end{bmatrix}$$

step3 确定最优方案和最劣方案

最优方案Z^+ 由 Z 中每列元素的最大值构成：

$$Z^+ = (max\{z_{11}, z_{21}, \cdots, z_{n1}\}, max\{z_{12}, z_{22}, \cdots, z_{n2}\}, \cdots, max\{z_{1m}, z_{2m}, \cdots, z_{nm}\})$$

最劣方案由Z^-由 Z 中每列元素的最小值构成：

$$Z^- = (min\{z_{11}, z_{21}, \cdots, z_{n1}\}, min\{z_{12}, z_{22}, \cdots, z_{n2}\}, \cdots, min\{z_{1m}, z_{2m}, \cdots, z_{nm}\})$$

step4 计算各评价对象与最优，最劣方案的接近程度

$$D_i^+ = \sqrt{\sum_{j=1}^{m} w_j (Z_j^+ - z_{ij})^2}$$

$$D_i^- = \sqrt{\sum_{j=1}^{m} w_j (Z_j^- - z_{ij})^2}$$

其中w_j为第 j 个属性的权重（重要程度），由熵值法获得。

D_i^+，D_i^-分别表示最优和最劣的接近程度

step5 计算各评价对象与最优方案的贴近程度

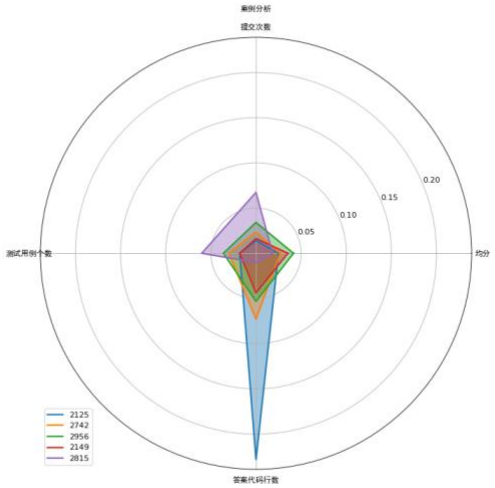
$$C_i = \frac{D_i^+}{D_i^+ + D_i^-}$$

其中$0 \le C_i \le 1, C_i$越接近 1 表明评价对象越优秀

step6 排序，得出结论

TOPSIS 分析结果

案例分析



挑选排名第 1，50，100，150，200 的题目作为分析样例

排名第 1 的题目，id 为 2125，难度系数 0.756415，名称 Vera 与道路建设，属于图结构，提交次数 43 次，满分提交 26 次，测试用例 3 个，标准答案有 360 行，没有显示题目来源，所以搜索难度提高，同时在雷达图上可以明显的看出代码行数超出一般题目许多，又因为代码行数权值大，从而综合得分高，难度排名第一。经过本组成员查证，题目来源 CCO2017，加拿大信息竞赛，属于国赛难度，所以，这道题目较难情有可原。

排名第 50 的题目，id 为 2742，难度系数 0.303247，名称可持久化平衡树，属于树结构，提交次数 72 次，满分提交 43 次，测试用例 5 个，标准答案 115 行，题目来源洛谷，在洛谷网站中属于省选/NOI-，属于省级竞赛难度。

排名第 100 的题目，id 为 2956，难度系数 0.233548，名称 A 先生的字符串，属于字符串，提交次数 105 次，满分提交 39 次，测试用例 6 个，标准答案 84 行，没有显示题目来源。经本小组成员查证，来源 TJOI，属于省级竞赛难度。

排名第 150 的题目，id 为 2149，难度系数 0.182964，名称寻找 LCT，属于图结构，总提交 106 次，满分提交 47 次，测试用例 3 个，标准答案 69 行，没有显示题目来源。经本小组成员查证，来源 libreoj，省赛简单题。

排名第 200 的题目，id 为 2815，难度系数 0.147915，名称乘积为一，属于数组，总提交 207，满分提交 95 次，测试用例 10 个，标准答案 16 行，题目来源 CodeForces，属于竞赛题目难度之下，但是仍需要较高的算法水平。

综合上述题目，可以发现，随着题目难度的降低，题目的答案行数降低，同时，同学们的总提交次数也有所上升，在雷达图中也有较好的表现，符合假设以及常识，说明模型对于题目难度的分析较为准确，能针对同学们的提交数据，区分出不同难度的题目，并将难度数据化，为后续分析提供基础。

局限

数据单一

本题的数据均来自 mooccode 平台的练习，但是数据只有每人提交的代码，而缺少运行的代码，从而缺少很多中间数据。如果能提供每个人提交的中间代码，使用该模型效果应该会好很多。

数据不统一

题目提供的答案有 C++和 Python 两种，所以使用代码行数作为指标的时候，会对最后的数据产生较大的影响，因为同一道题目，使用 Python 行数要小于 C++。如果将答案统一为 C++或者 Python 代码，使用该模型效果会提升很多。

测试用例过少

用例个数大多数都小于 10 个，导致每道题的分数不分散，产生杠铃型分布，如果能提升测试用例个数，能进一步提升测试的准确性，同时也能更全面的测试出算法的复杂度，边界情况，去除面向测试用例编程的可能性。

答案显示用例

题目的答案只有固定的用例，大大降低题目难度，学生可以通过相同的操作完成题目，建议以后将答案换成动态答案或者题目的标准解法，从而使得题目难度分析更加精确。

使用层次分析法分析个人编程能力

层次分析法简介

层次分析法（Analytic Hierarchy Process, AHP）为 1971 年 Thomas L. Saaty （匹兹堡大学教授）所发展出来，主要应用在不确定情况下及具有多数个评估准则的决策问题上。

层次分析法发展的目的是将复杂的问题系统化，由不同层面给予层级分解，并通过量化的运算，找到脉络后加以综合评估。

使用层次分析法的原因

适合评价类问题

层次分析法的实质，即为经过调查研究后处理数据，以得到影响评价体系的多个因素的权重。层次分析法较为适合评价类问题的分析，对于学生测评等尤其有效。

引入主观因素

引入相对主观的、经调查得到的数据作为分配权重的依据，一定程度上弥补了熵值法仅仅通过信息熵定义权重的弊端，以人为本，按照被调查者对能力分布的期望分配权重。

实践

选取评价指标

基于已有数据对被试者的能力进行评价，最重要的是根据被试者的提交次数、每次提交成绩、最终提交成绩与提交代码质量进行分析。根据概述部分我们已经对总计 300 余名被试者的分组情况进行确定。

显然，最终提交得分越高、提交题目占为被试者提供题目比例越大，则被试者在此方面的能力越强。然而，多次提交等刷分行为、直接面向测试用例编程的作弊行为等，也均会影响被试者能力评价体系的建立。

当被试者存在面向用例的作弊行为时，在最终提交得分处获得的高评价一定会被否决。但是，由于慕测平台提供的数据仅限于***提交记录***、而非***测试用例运行记录***，依据不同被试者做题习惯不同（部分运行到满分再提交，部分以提交代替运行）所以对题均提交次数的研究存在一定的局限性

我们在理论分析的基础性上，结合了已知数据的可靠性，确定了以下指标及处理方法：

1. 被试者需要根据所做题目不同划分为不同组别，不同组别被试者不具有可比较性；
2. 最终成绩选取一组统计量：
   - 平均最高得分：对于每道题目，我们均将最高提交得分作为最终得分指标，最高得分越高，说明对该题目的解答正确性越高；
3. 解题过程选取两组统计量：
   - 解题过程平均分：对于每次提交，我们将某类型下已做题目的所有提交计算算术平均分，该项分值越高，说明此被试者在本类题目中表现越好；
   - 完成题目的题均作答次数：对于所有尝试过的题目，计算每个类型该题目的平均作答次数，平均作答次数越低，说明本类题目被试者掌握程度越好
4. 作答完整度选用两组数据统计量：
   - 作答题目数与全部题目数比：该项比例越高，说明被试者在本类题目中作答越完整；
   - 带难度的作答完整度比：由于前序研究已获得了题目的难度，所以可以将难度引入到层次分析中，计算带难度的作答完整度比方法如下：

$$Res = \frac{\sum_{S_{submit}} score}{C_{all} * 100}$$

其中：

S_{submit}:被试者最终提交的本组本类型题目集合

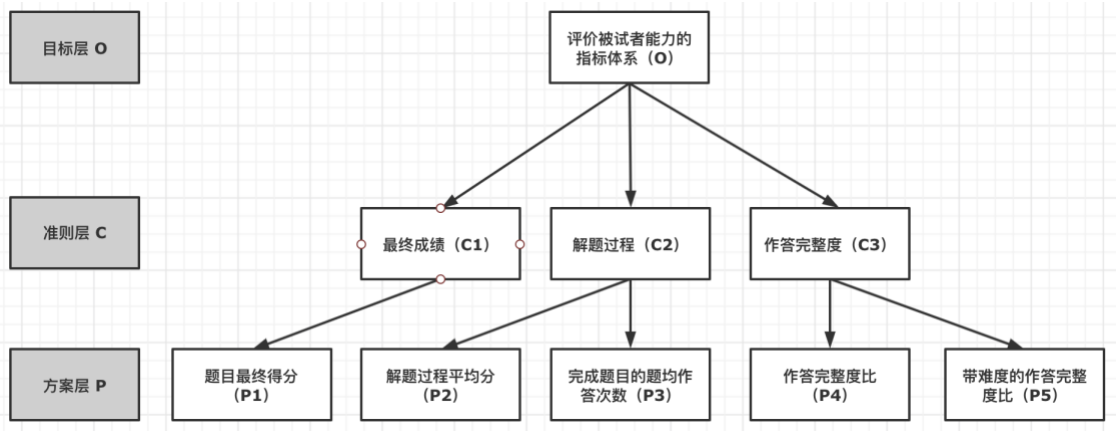
C_{all}:本组本类型的题目数

我们采用如图所示的方法对两两指标的重要性进行检验

等级	同样重要	稍微重要	明显重要	强烈重要	极端重要
量化指标	1	3	5	7	9

层次分析法建立评价模型

为确保上述指标的适用性，我们使用层次分析法进行检验



由于我们需要对 P1-P8 所有 8 个方案（或指标）计算其权值，则先分为三个准则，对每个准则下的一个或多个方案进行重要性评定：

最终成绩 C1 仅存在一个指标，故无重要性差异

解题过程 C2 存在 P2、P3 两个指标，分析后重要性如下：

	C2 解题过程	
	P2	P3
P2	1	3
P3	1/3	1

作答完整度存在 P4、P5 两个指标，分析后重要性如下：

由于我们已有了难度数据，所以杂合难度后的作答完整度比无难度下的作答完整度重要

	C3 作答完整度	
	P4	P5
P4	1	1/5
P5	5	1

得到三个准则 C1-C3，继续对它们进行重要性评定：

	O 评价体系		
	C1	C2	C3
C1	1	7	3
C2	1/7	1	1/5
C3	1/3	5	1

由于 O:C 的重要性表格存在 3 行 3 列，有必要进行一致性检验：

一致性指标：

$$CI = \frac{\lambda_{max} - n}{n - 1} = 0.0324$$

一致性比例：

$$CR = \frac{CI}{RI_n} = \frac{0.0324}{0.52} = 0.0624 < 0.10$$

O:C 矩阵的一致性可以接受

方案层对准则层的权重：

- C1:{P1}：

$$(1)$$

- C2:{P2, P3}：

$$\begin{pmatrix} 0.75 \\ 0.25 \end{pmatrix}$$

- C3:{P4, P5}：

$$\begin{pmatrix} 0.17 \\ 0.83 \end{pmatrix}$$

- 使用特征值法，求得准则层对目标层的权重：
  - O:{C1, C2, C3}：

$$\begin{pmatrix} 0.6491 \\ 0.0719 \\ 0.2790 \end{pmatrix}$$

- 最终计算得到五种准则对目标层的权重为：

$$\begin{pmatrix} 0.6491 \\ 0.0539 \\ 0.0178 \\ 0.0474 \\ 0.2316 \end{pmatrix}$$

TOPSIS 分析结果

算法过程

1. 将原始数据正向化

我们选用的某些指标，指标值越高、显示被试者能力越强；而亦存在一些指标，其值越高、显示被试者能力越弱。有必要将所有指标进行统一的正向化，方便运算。

指标名称	指标类型
最终得分	极大型指标
题目提交均分	极大型指标
作答题目比	极大型指标
提交次数比	极小型指标
面向用例比	极小型指标

此次分析中，仅存在极小型指标需要正向化，使用极小型转极大型的简单公式：

$$x' = x_{max} - x$$

即可完成转换

2. 正向化矩阵标准化

   略，见上

3. 计算得分

   略，见上

   最终，可以算出第 i 个评价对象未归一化的得分：

$$S_i = \frac{D_i^-}{D_i^+ + D_i^-}$$

4. 对 Si 的解释

   使用层次分析法结果加权后，Si 即可看做被试者的平均得分，越高则能力越强

数据处理

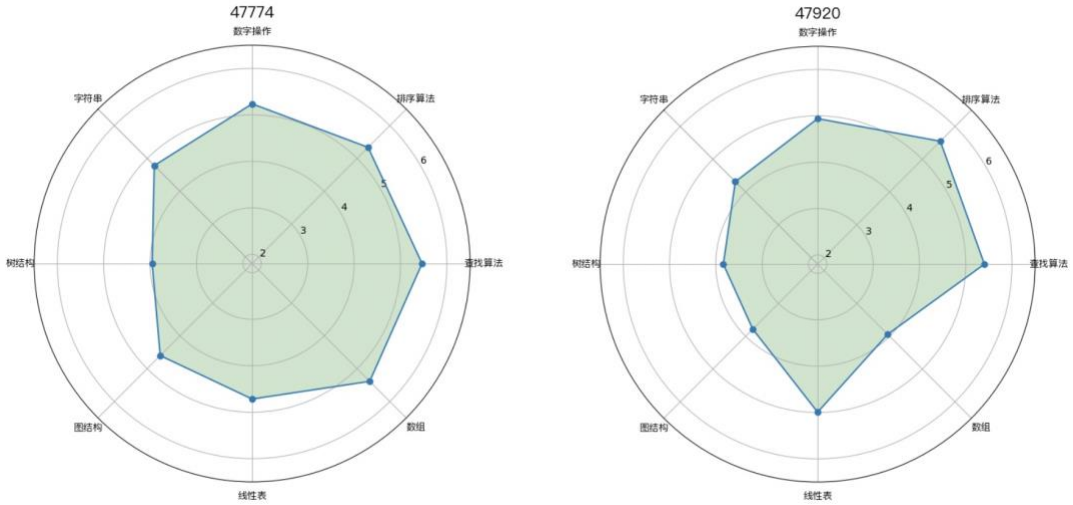
依据评价体系与 2.2 中算法，被试者最终得分越高则其能力越强

为了能够统一被试者的能力标准，依据 1.0 中*组间题目不同但难度相似*的前提，我们在分组得到基本统计量后，将不同组别合并处理，依据 2.2 中算法，得到最终分数。下图为部分同学在*查找算法*项目中的得分情况：

S score 项为 TOPSIS 中所得 *Si* 同倍数放大后的结果

id	Mean sore committed	Mean score submitted	Submit per commit	Commit ratio	Commit diff ratio	S score
47329	0	0	2	0	0	1.81249262
8160	100	87.65444444	1.285714286	1	1	5.56824346
8246	94.04761905	85.86956522	1.095238095	1	0.941216206	5.43427969
8317	94.04761905	94.04761905	1	1	0.941216206	5.49534482
8318	94.04761905	94.04761905	1	1	0.941216206	5.49534482
16304	94.04761905	94.04761905	1	1	0.941216206	5.49534482

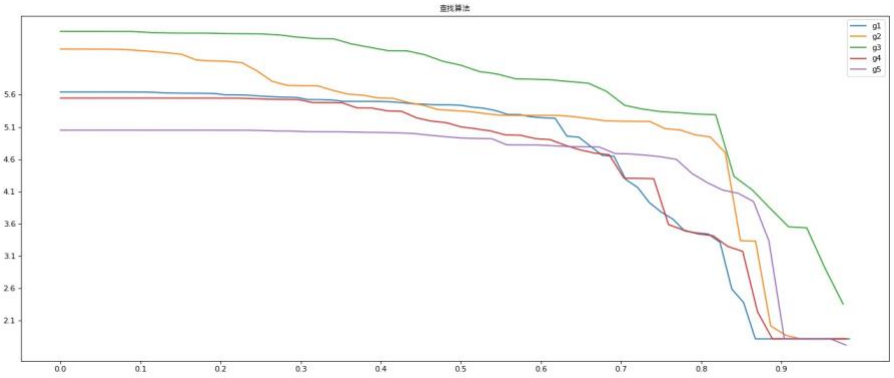
依据上述数据制图，得到每位同学的能力分布雷达图，下面为 id 为 47774，47920 两位同学的能力分布雷达图：



分析雷达图后，我们发现完成率高的同学们，能力分布大体形状一致，但存在少许差别；

即采用 TOPSIS 法分析同学们的能力分布基本合理。

依据上述数据制图，我们还可以得到各组同学之间的能力分布差距（基于各组题目难度大致相同的前提）

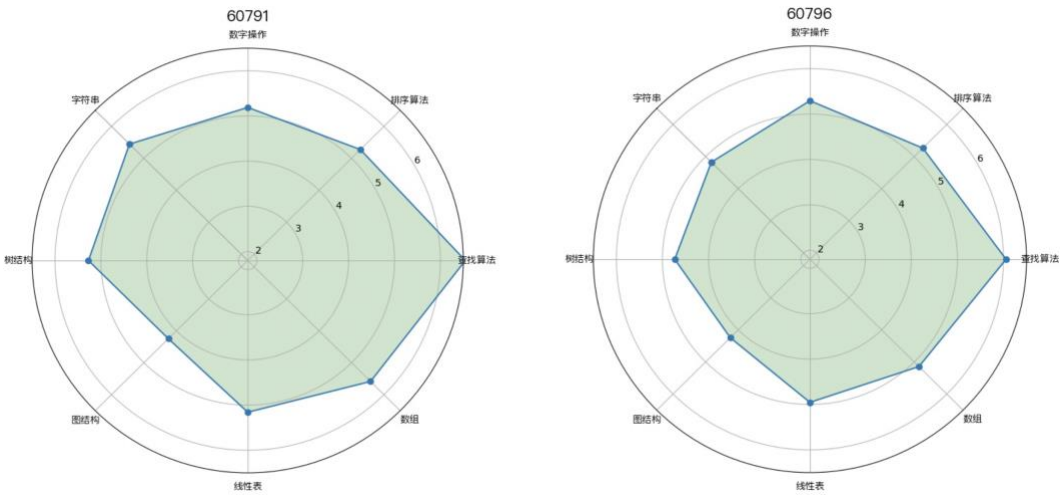


小结

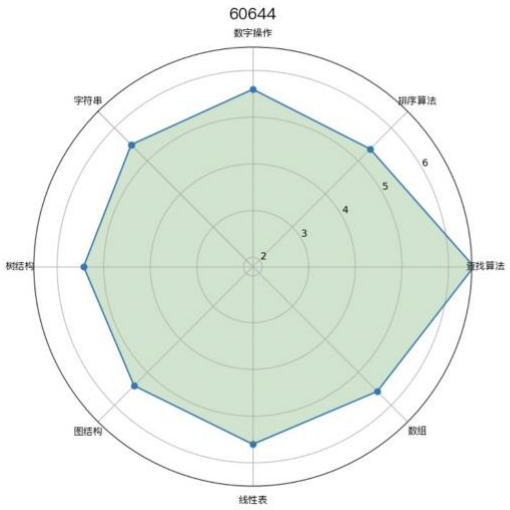
被试者能力分布

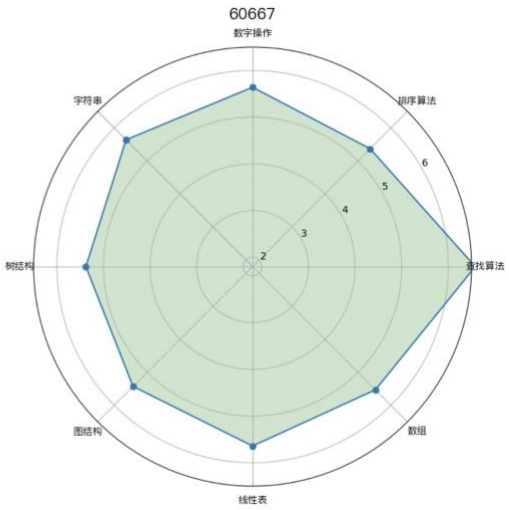
在不同组别题目不同但难度大致相同的假设下，将不同组别的同种类型题目杂糅计算，获得同学们的能力雷达图，发现如下几类代表性情况：

绝大多数同学的情况，以第三组为例：图结构、树结构、字符串相较于查找算法等类型难度较大，同学们在这些难度较大的题目中得分相对不高，但题目完成度较高，没有明显短板

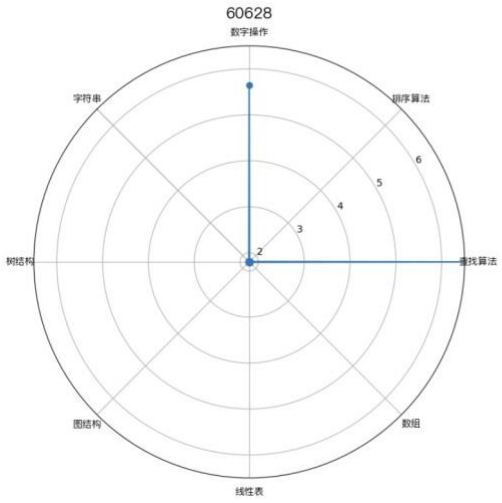


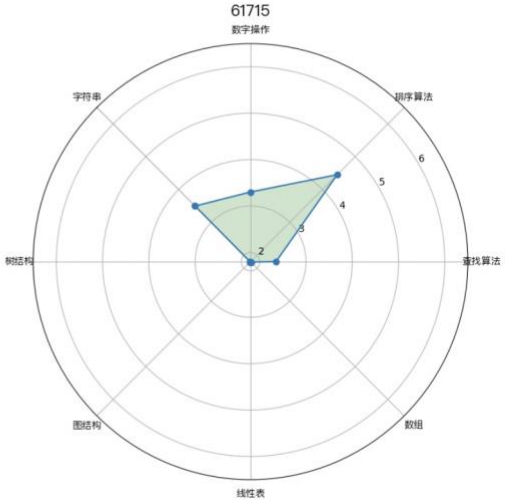
以第三组为例：题目完成率高，完成所有题目，在每种类型上都能取得优异的成绩



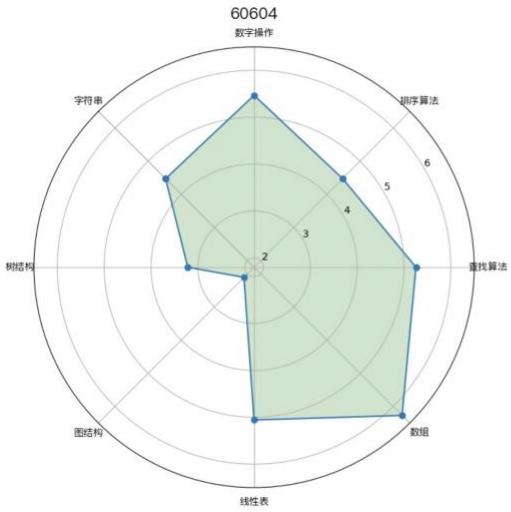


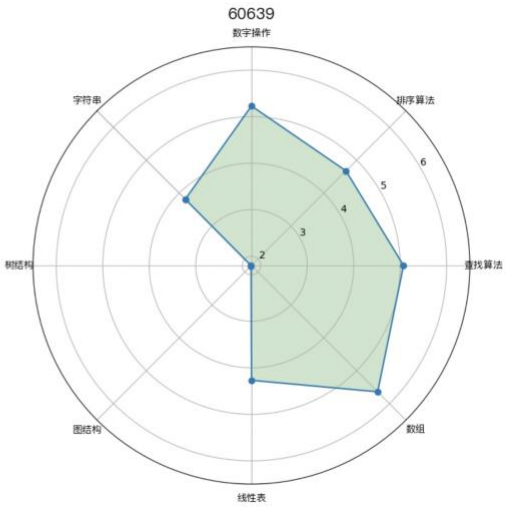
以第三组为例：选择性完成部分题目，仅在最初做题兴致较高，而随着时间的推移逐渐陷入*放弃*的被试者



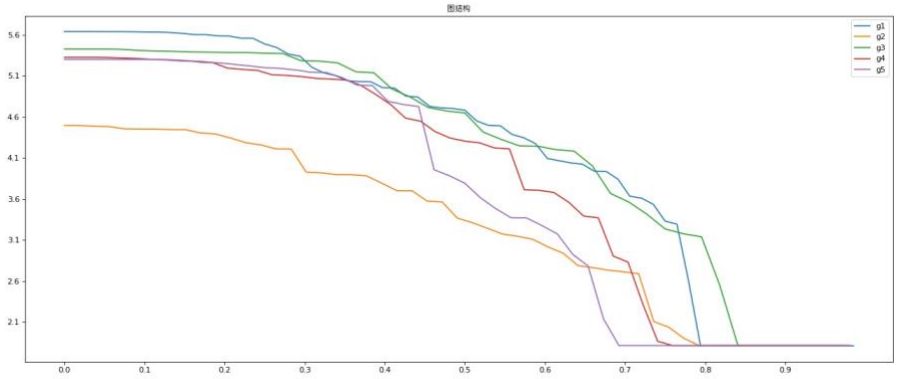


以第二组为例：最终放弃题目较难、占比不大的图结构、树结构等，但较好地完成了其他项目的被试者

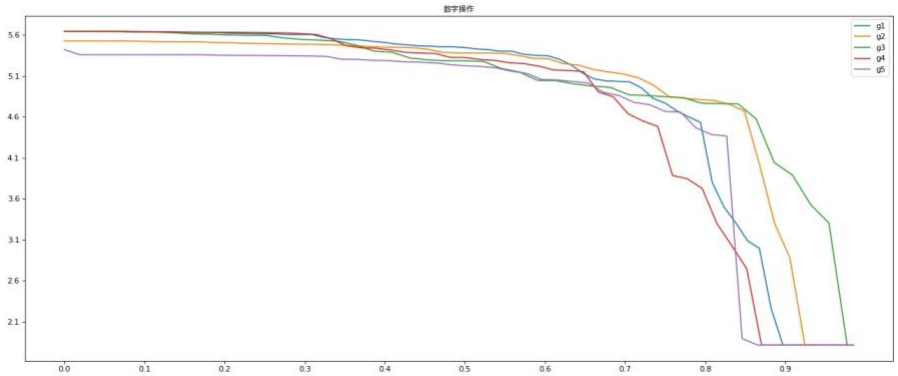


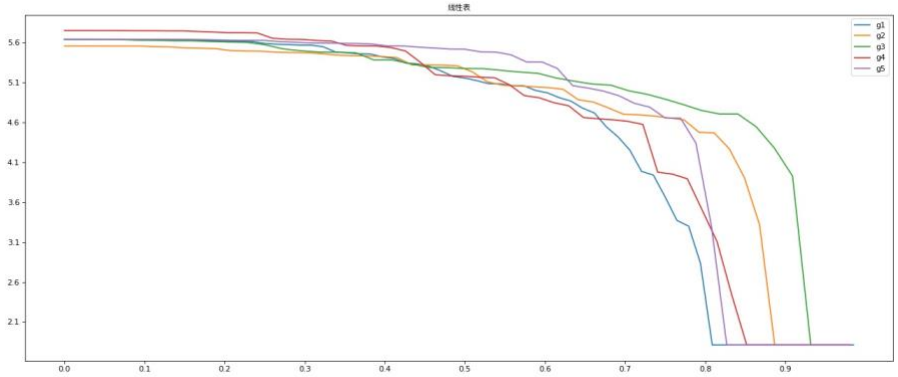


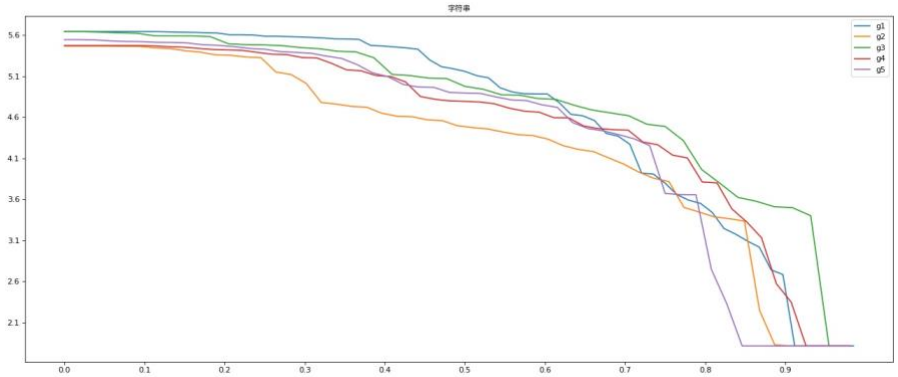
组间分类别能力比较



以最具代表性的图结构为例，图结构被试者的分数分布曲线明显低于其他的题目，且在图结构组间分析中，第二组曲线明显较为弱势，说明第二组的图结构试题可能相对其他组别较难，或第二组同学在图结构题目中与其他组别差距较大。







而在数字操作、线性表、字符串这类大家掌握较好、题目也不易出难的类别中，五个组别的得分差距不大，且绝大多数同学的得分均较为理想。

研究局限性

1. 受制于 OJ 系统提交的特点，同学们可以在一次提交前进行多次运行操作，但运行操作不计入提交次数、亦不存在所谓*运行记录*可供参考。因此，必然大量存在多次运行满意后再提交的受试者。必然降低平均提交分数、提交次数等作为参考数据的可信度。
2. 由于线上 OJ 与 Mooctest 系统的局限性，导致大量同学在其他线上 OJ 平台中查询题解，可能会使部分被试者的成绩虚高，无法客观反映同学的能力水平。
3. Mooctest 系统每次提交显示正确与错误输出的性质，致使部分同学直接“面向用例”套出答案，也会导致最终能力水平的分析结果不够客观。
4. 层次分析法的实质即为主观因素在最终评价体系的建立中，占据主导地位。受限于时间因素，我们仅对自己周边的几位同学进行调查，得到指标权重，可能无法全面的反映出被试者、教师等其他群体的意愿。

可视化题目难度、题目质量和学生能力的总体分布情况

Rasch Model 简介

项目反应理论（IRT）

在了解 Rasch 模型之前,先介绍一下项目反应理论(Item Response Theory, IRT)。

> *项目反应理论是一系列心理统计学模型的总称。IRT 是用来分析考试成绩或者问卷调查数据的数学模型。这些模型的目标是来确定的潜在特征（latent trait）是否可以通过测试题被反应出来，以及测试题和被测试者之间的互动关系。——维基百科*

简单来说，项目反应理论是用来分析试题难度、人的解题能力及其相互关系的一系列理论，它的应用很广泛，SAT、TOFEL、GRE 等考试都是以项目反应理论为基础构建的测验。而我们所使用的 Rasch 模型就是 IRT 系列模型中最基本的一个。

Rasch 模型的假设

Rasch 模型有四个假设：

1. 单维性，测验中的每一个项目（item）都测量到一种共同的潜在特质。
2. 局部独立性，被测者在每一个项目（item）上的反应都是独立的。
3. 非速度测验假设，测验的进行是没有时间限制。
4. 知道——正确假设，如果被测者知道一道题目的正确答案，则必然答对。

Rasch 模型公式与试题特征曲线（ICC）

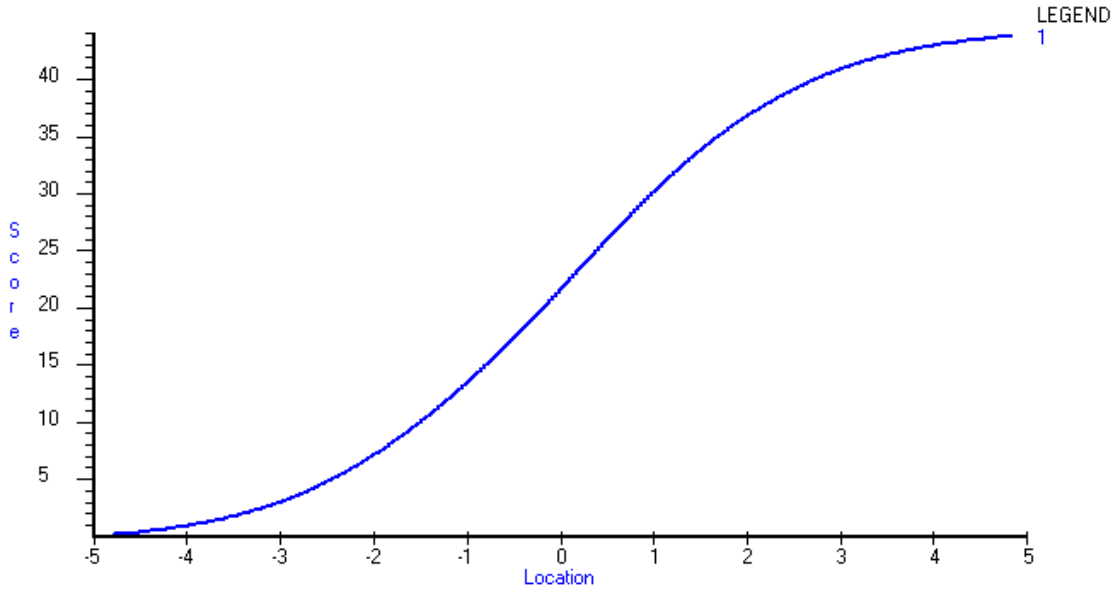
$$P(U_i = 1|\theta, b_i) = P_i(\theta) = \frac{e^{(\theta - b_i)}}{1 + e^{(\theta - b_i)}} \quad (1)$$

b_i: 试题难度参数(item difficulty parameter)

θ: 能力参数(ability)

公式(1)即 Rasch 模型的核心公式，它表示能力参数为θ的人在做一道难度参数为b_i的题目的时候答对的概率。

该公式的图像是一条逻辑斯蒂曲线(Logistic Curve),在此处被称为试题特征曲线(Item Characteristic Curve, ICC)



ICC Example

在 ICC 图中, $\theta = b_i$时,被试者答对的概率刚好是 0.5; $\theta - b_i > 0$,则答对几率大于 0.5; $\theta - b_i < 0$,则答对几率小于 0.5。所以我们看到 ICC 曲线的理想状态是 S 形，当人的能力θ增加时，他所能够应对的试题难度会逐渐提升,但是一开始这种提升是不明显的(图中曲线斜率较小);之后再中期这种提升变得明显(图中曲线斜率较大);当他的能力超过一个较大值时,这种难度的提升再次变得不明显(图中曲线斜率较小)。

Rasch Model 的估计原理

Rasch Model 使用极大似然估计的方法。我们以估计学生能力来解释其估计方法，估计试题难度的部分也是类似的方法。

首先需要有一个前提，那就是各个试题互不影响，各个学生之间也互不影响。

暂时先假设每道题的难度系数b_i是已知的。

考虑公式：

$$P_i(\theta) = \frac{e^{(\theta - b_i)}}{1 + e^{(\theta - b_i)}} \quad (2)$$

使用u_i来表示一个人是否答对一道题：

$$U_i = u_i = \begin{cases} 1, correct\ \ response \\ 0, incorrect\ response \end{cases} \quad i = 1,2,\ldots,n \quad (3)$$

使用Q_i来表示一个人答错一道题的概率,显然：

$$Q_i(\theta) = 1 - P_i(\theta) = \frac{1}{1 + e^{(\theta - b_i)}} \quad (4)$$

于是,在某个学生的能力θ的条件下,他答对第i道试题的概率可以如下表示：

$$\begin{aligned} P(U_i|\theta) &= \begin{cases} P_i(\theta), correct\ \ response(u_i = 1) \\ 1 - P_i(\theta), incorrect\ \ response(u_i = 0) \end{cases} \\ &= P_i^{u_i} Q_i^{1-u_i}, i = 1,2,\ldots,n \end{aligned} \quad (5)$$

因为试题并不会互相影响,所以：

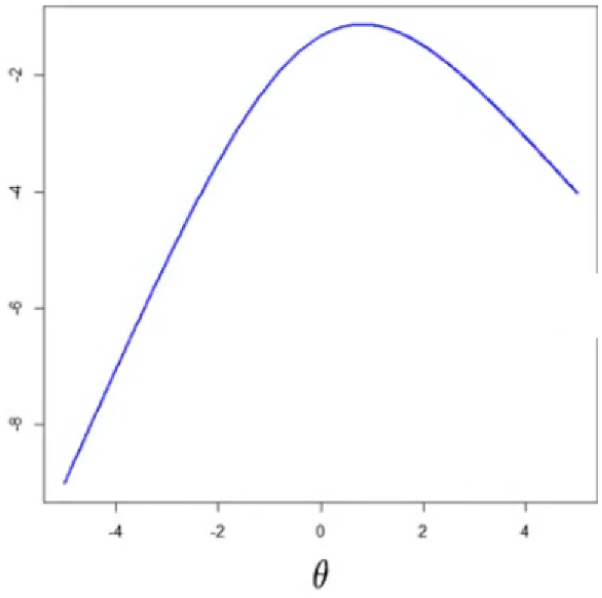
$$\begin{aligned} P(U_1, U_2, \ldots, U_n|\theta) &= P(U_1|\theta)P(U_2|\theta)\ldots P(U_n|\theta) \\ &= \prod_{i=1}^{n} P_i^{u_i} Q_i^{1-u_i} \end{aligned} \quad (6)$$

接下来，取对数似然函数：

$$
\begin{aligned}
L &= \ln P(U_1, U_2, \ldots, U_n|\theta) \\
&= \ln P(U_1|\theta) + \ln P(U_2|\theta) + \ldots + \ln P(U_n|\theta) \\
&= \sum_{i=1}^{n} (\ln P_i^{u_i} + \ln Q_i^{1-u_i}) \\
&= \sum_{i=1}^{n} (u_i \ln P_i + (1-u_i)\ln Q_i)
\end{aligned} \tag{7}
$$

接下去,我们只要把该同学做每道题得到的u_i与b_i代入公式（7）的右边，就可以发现 L 只与θ有关，他们的关系及关系图象如下：

$$\hat{\theta} = argmax\ L \tag{8}$$



L与θ关系

于是我们只要找到上图最高点，他对应的横坐标就是学生能力θ的估计值。

使用 Rasch Model 的原因

Rasch Model 本身的优势

Rasch Model 应用如此广泛，是因为其本身有很多优势：

1. 数据有线性的特质，Rasch 模型可以把非线性的数据转换成为具有等距意义的“logit scale”数据，从而使客观的测量成为可能。
2. 参数分离，客观测量的参数分离要求“题目难度的标定必定独立于被试样本的分布，对个体能力的测量必须独立于题目的难度分布”。而由 Rasch model 的估计原理可知，这两个要求是可以被满足的。
3. 个体和题目公用一把尺，Rasch 模型通过对数转换，将个体和题目在同一单维度尺上进行标定。因此个体与个体之间、题目与题目之间、个体与题目之间可以方便地进行比较。

我们获得数据样本的适用性

小组从老师那里拿到地数据是符合 Rasch 模型的要求的。

首先，每一个同学做一道题是否通过是最核心的一个数据，这个可以由原始数据中$final_score$来展示。如果$final_score$为 100 则视为通过该道题目测试；反之则视为未通过。

第二，尽管所有的同学被分成了五个组,每个组有 50-60 个人,我们在分析的时候不能进行总体年级的分析，但是我们分析的时候也可以根据这样的分组来做。

第三，可以假设每个同学答题是互不影响的，每道试题也是互不影响的。尽管这可能过于理想化，但是我们相信不合要求的数据是极少的，而我们要研究总体的情况，因此可以认为极少数的不合规不会影响总体的分析与判断。

第四，我们不仅可以提取处分组的数据，还可以提取出分类的数据（如数组，字符串，图结构等），这样我们可以不仅仅分析所有类别题目总体的情况，还可以分析每个类别的情况。

实践

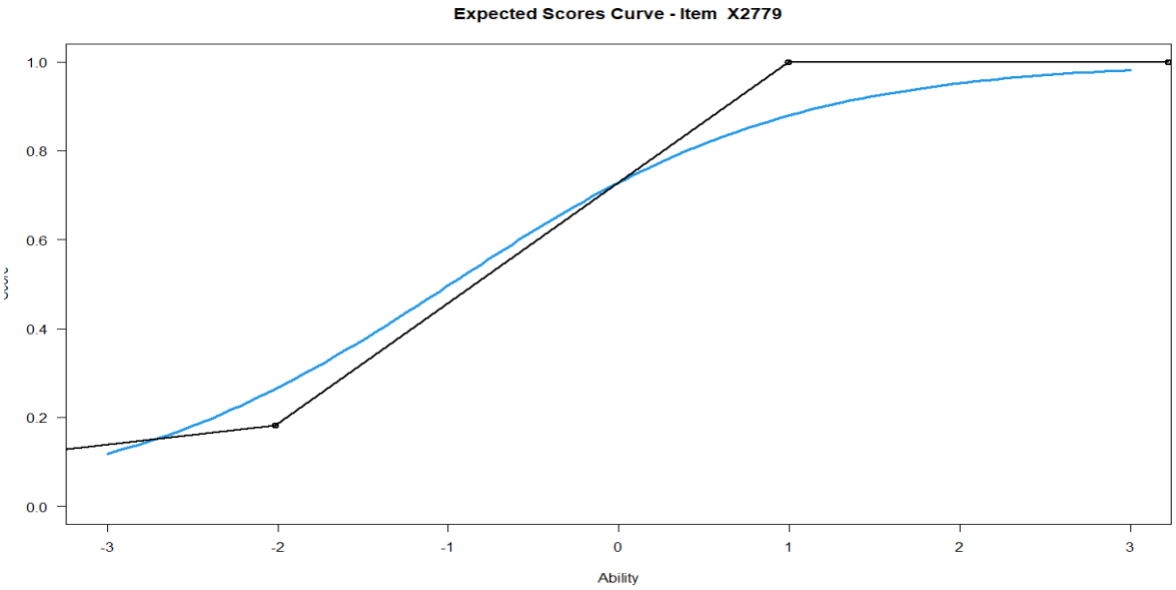
我们使用 R 语言来计算 Rasch Model 的参数，并绘制图表。因为分成了五组数据，并且数据量比较大，所以不会在文中列出全部的参数和图表，我们仅仅拿出一些典型的数据与图表来进行研究分析。

ICC

单条 ICC 图

先来看一些第一组中题目的单条 ICC 图，可以观察出一些有意思的结论：

- 题目一：

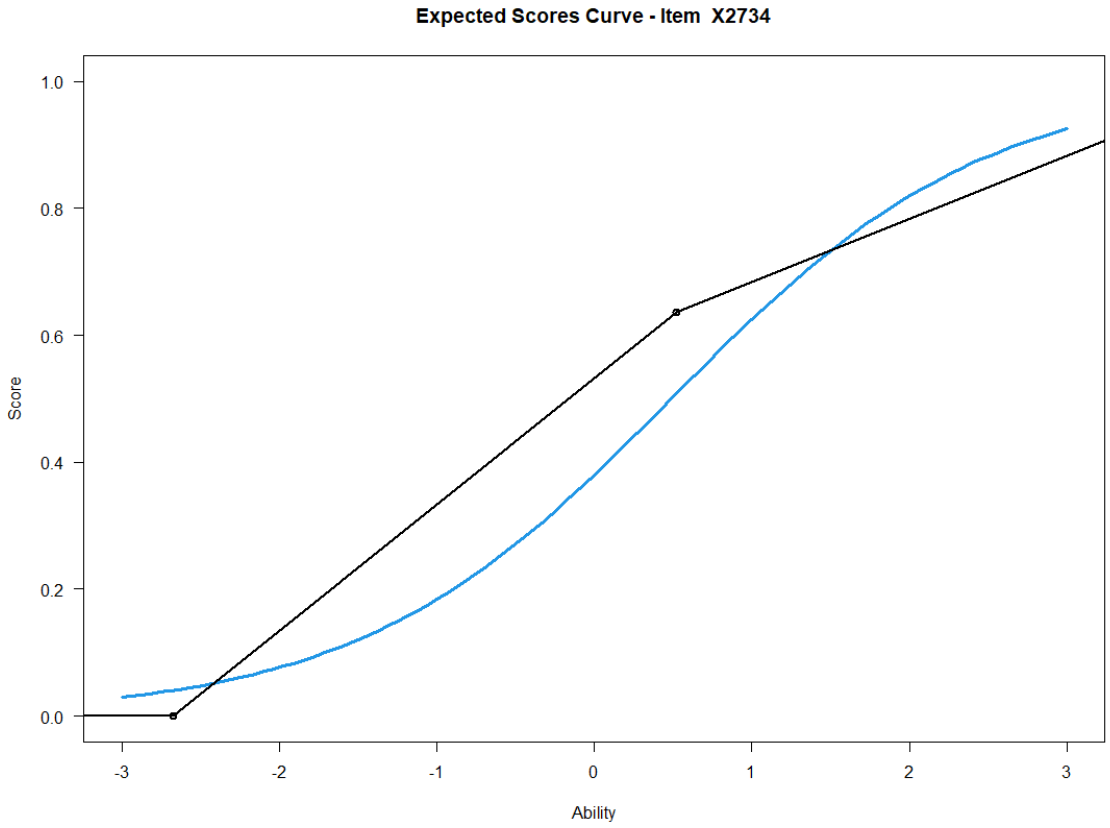


这是第一组的一道题目，题目编号是 2779，类别是线性表。

图中的黑色折线是同学们实际做题的情况，蓝色曲线是通过 Rasch Model 拟合出来的曲线。我们可以看到黑色和蓝色的线总体上是比较接近的，这说明我们的分析是比较有效的。另外，拟合曲线也很接近于一条 S 形的逻辑斯蒂曲线，这说明这道题是质量比较高的，有一定的区分度。我们也可以看出当能力参数比较小时，黑色折现在蓝色曲线下面，这说明此时 Rasch 模型是高估了学生的能力；而当能力参数变大时，Rasch 模型却低估了学生能力。

而在模型中，我们计算得到的这道题的难度参数为-0.9881，是一道中等难度的题目。

- 题目二：

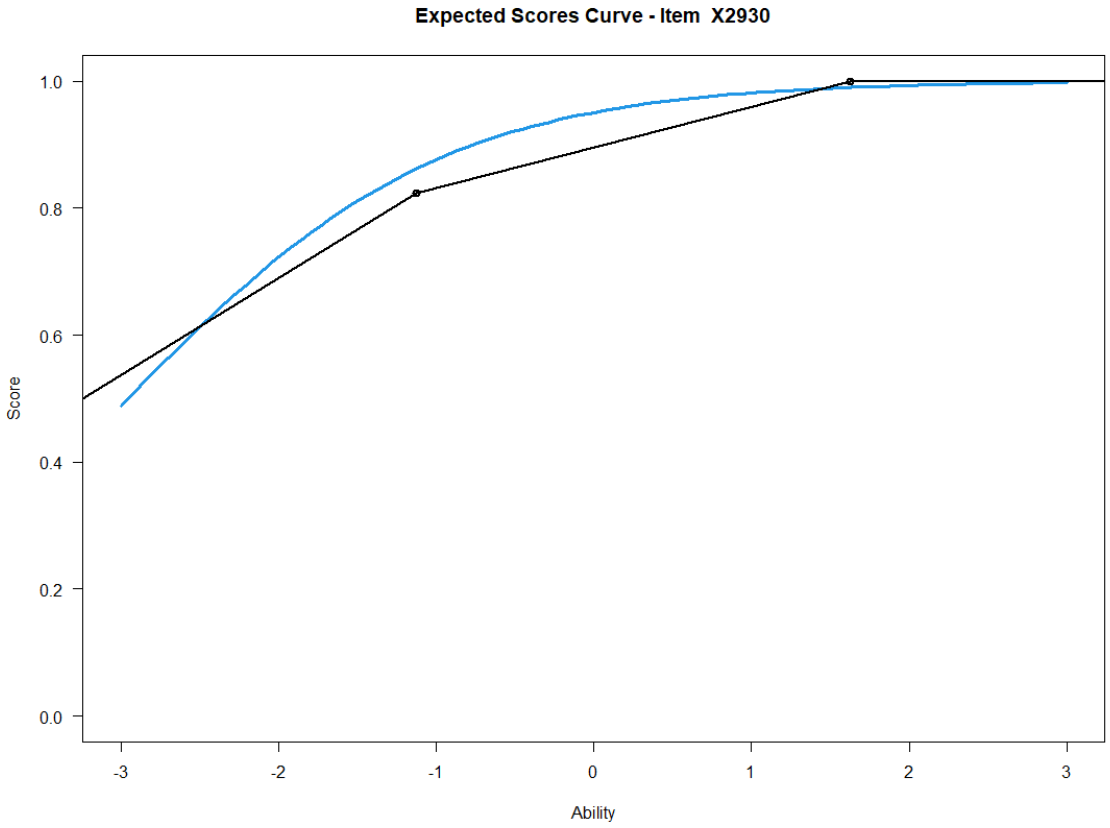


这依然是第一组的一道题目，题目编号是 2734，类别是树结构。

可以明显地看到，表示实际的黑色折线和拟合的蓝色曲线并不是特别接近，说明 rasch model 的拟合在这道题并不是特别准确，但是我们依然可以通过这张图得到很多信息。拟合曲线依然很像一条逻辑斯蒂曲线，这道题依然是区分度比较高的，因此质量比较不错。我们也可以看到，当能力参数小于 0 时，它的增加并不能明显提高学生答出此题的可能性（当$ability$较小时，曲线斜率增长较慢）。

总体上面这张图中的拟合曲线明显低于题目一的拟合曲线，说明总体上学生做出这道题的概率低于做出题目一的概率。而在模型中，我们计算出的难度系数为 0.4901，明显大于题目一的难度系数，因而我们从图上观察出的结论时正确的。

- 题目三：



这是第一组的一道题目,编号是 2930,类别是数组。

我们可以发现，这道题并没有很高的区分度，因为能力比较低的学生已经可以有接近 0.5 的概率答对该题，而能力参数为 0 的同学已经有接近 1 的概率去完成作答，因此这道题质量不是很高。

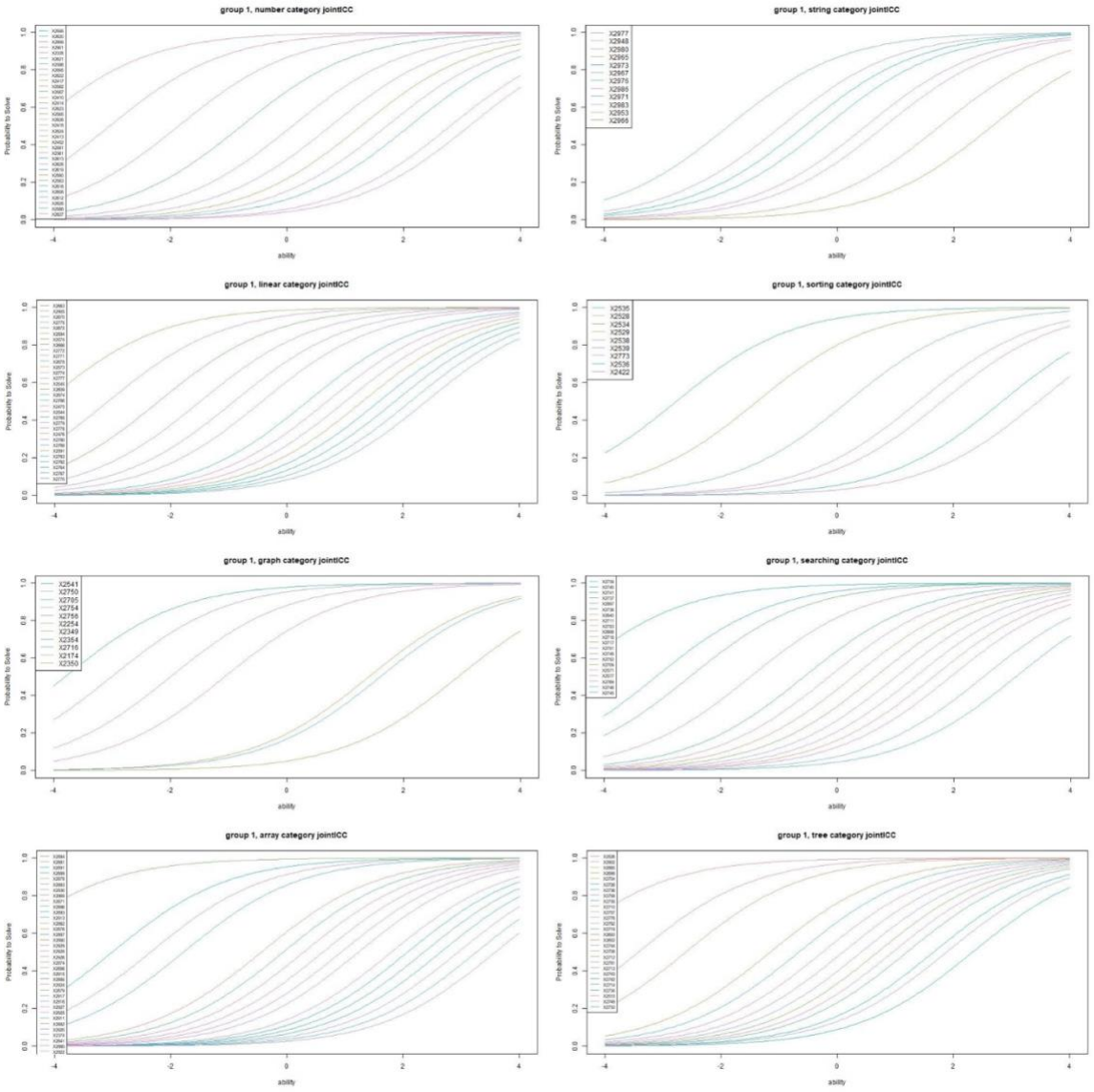
另外，与上面两张图比较来看，题目三的曲线明显是高于题目一和题目二的，因此我们认为这道题是相对简单的。事实上，我们得出的难度系数为-2.9545，这说明题目三是一道很简单的题目。

本组的 R 语言代码通过 tam 包可以获得每道题目的单条 ICC 图，发现大多数题目是符合逻辑斯蒂曲线的，它们有较高的质量，但是限于篇幅，在这里就不放过多图片。

联合 ICC 图（jointICC）

在画图的时候可以把多条 ICC 曲线放到同一张图里面，这样观察题目集合的特征时，可以更加直观。这种图片我们称之为联合 ICC 图（jointICC）。

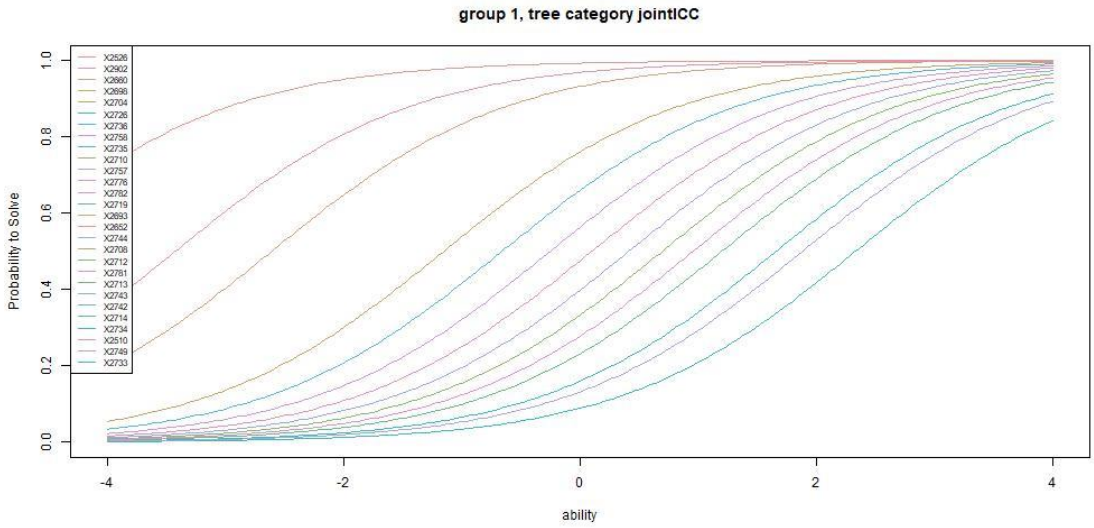
依然以第一组为例。先把第一组八类题目各自的联合 ICC 图拼接到一起,如下所示:



可以发现,每种类别的题目难度分布都很完整(曲线接近 x 轴说明题目难，曲线远离 x 轴说明题目简单),也就是说,从简单到困难的题目在每种类别里面都有体现。这说明我们题库中的题目设计是很良好的，没有出现不合理的题目设计（比如图结构的题目都特别难，数字操作的题目都特别简单这样的不合理）。

另外，我们可以发现每种类别的题目分布都是类橄榄型，简单和困难题较少，中等难度的题比较多，这样可以增加学生能力的区分度，因为简单的题目可能所有人都会答对，困难的题目可能所有人都会答错，而中等难度的题目可以拉开差距，形成区分。而如果只比较简单题与难题的数量，可以看出简单题比难题多。

接下来，可以再聚焦看第一组树类别的联合 ICC 图。



可以发现，第一组的树类别的题目题量是足够的，大部分题目都是中等难度，他们的曲线是 S 形的逻辑斯蒂曲线。而有三道题比较偏上（远离 x 轴），这三道题我们可以归结为“简单题”或者说“送分题”，他们相对来说是比较容易被解答的。综合来看，第一组树类别的题目质量是很高的。

其他各组的题目联合分布图大致情况也类似第一组，由于篇幅原因就不放在此处。

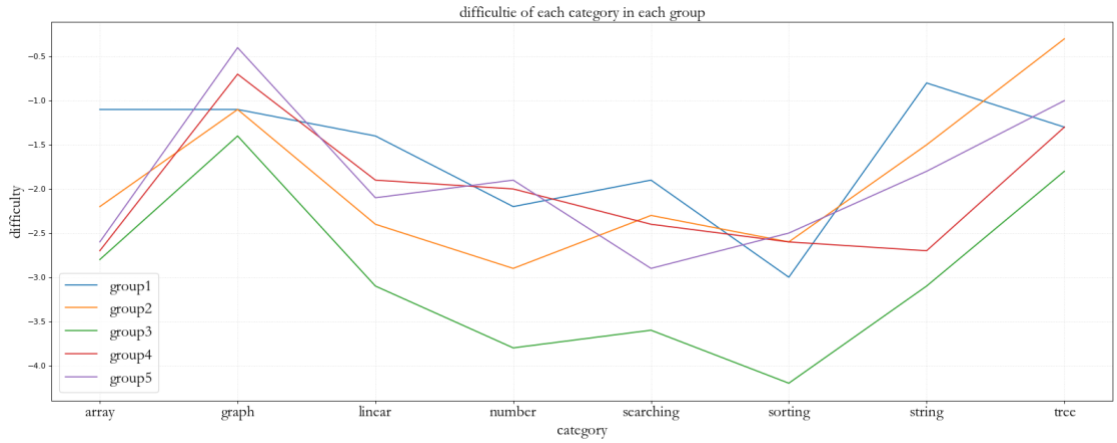
难度系数

除了图片上通过视觉直观的观察之外，还需要查看计算得到的参数来进行题目难度与质量的分析。

下面是一张本组使用 rasch 模型计算得到的平均难度系数表格。

	all	array	Graph	Linear	Number	Search	Sort	string	tree
group1	-1.3	-1.1	-1.1	-1.4	-2.2	-1.9	-3.0	-0.8	-1.3
group2	-1.2	-2.2	-1.1	-2.4	-2.9	-2.3	-2.6	-1.5	-0.3
group3	-2.6	-2.8	-1.4	-3.1	-3.8	-3.6	-4.2	-3.1	-1.8
group4	-.17	-2.7	-0.7	-1.9	-2.0	-2.4	-2.6	-2.7	-1.3
group5	-1.7	-2.6	-0.4	-2.1	-1.9	-2.9	-2.5	-1.8	-1.0

接下来是本组根据难度系数表格画出来的折线图(没有 all,只有八大类别)：



前提假设

假设每个组的同学的实力都是差不多的。在分组的时候是按照学号来分的，学号的分布是比较均匀的，因而有理由假设每个组的同学实力差不多。

组内比较

首先进行组内的比较。

第一组：可以看到数组类别、图类别、字符串类别、和树类别的题目总体是比较困难的（也可以认为是第一组同学在这些方面不太擅长），而数字类别和排序算法比较简单（也可以认为是第一组同学在这些方面比较擅长）。

第二组：图类别和树类别的题目总体是比较困难的（也可以认为是第二组同学在这些方面不太擅长），而数字类别和排序算法比较简单（也可以认为是第二组同学在这些方面比较擅长）。

第三组：图类别和树类别的题目总体是比较困难的（也可以认为是第三组同学在这些方面不太擅长），而数字类别和排序算法比较简单（也可以认为是第三组同学在这些方面比较擅长）。

第四组：可以看到图类别和树类别的题目总体是比较困难的（也可以认为是第四组同学在这些方面不太擅长），而数组类别和字符串类别比较简单（也可以认为是第四组同学在这些方面比较擅长）。

第五组：可以看到图类别和树类别的题目总体是比较困难的（也可以认为是第四组同学在这些方面不太擅长），而数组类别和查找算法比较简单（也可以认为是第四组同学在这些方面比较擅长）。

每一组组内的分析来看，每个组题目难度分布并不是完全相同的，比如第一组的数组类别比较困难，而第四组和第五组的数组类别却比较简单。笔者认为，这种差异更应该被归结为客观因素，即题目难度的区别，而非主观的同学能力的不同，因为我们之前假定每个组同学的实力是差不多的。

组间分析

接下来进行组间的分析研究。

可以明显地看到，第三组的折线是低于其他四组的。这说明第三组的题目相对而言是比较简单的（当然我们可以理解为第三组的同学实力比较强，完成度比较高。但

这是主观的因素，笔者同样更倾向于相信客观因素，即题目难度）。而其他四组的折线是差不多的、有交错的，这可以理解为其他四组题目的难度差不太多。

整体来看，树结构和图结构的题目是相对比较困难的，同学们很难解答这些题目。我们会建议加强同学们在这方面的练习。而数组、数字操作的题目相对而言比较简单，同学们较为容易解答这些题目。

WrightMap (Person-Item Map)

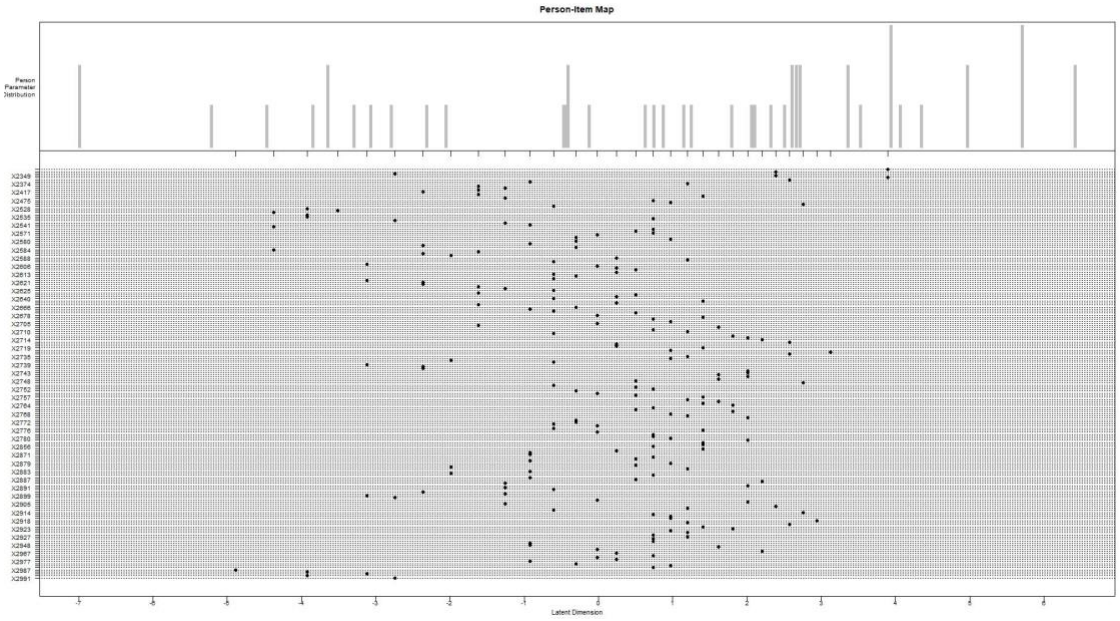
介绍 WrightMap

WrightMap （怀特图），也叫 Person-Item Map(试题-人图)，是为了纪念在 Rasch 模型测量中做出卓越贡献的本怀特（Ben Wright）而命名的。这种图可以按照比例显示被试者的能力和试题的难度，被试者能力和试题难度被（水平或者垂直地）分隔开，它被广泛应用于各种响应模型中。

实践中使用 WrightMap

本研究小组使用 R 语言来绘制 WrightMap。

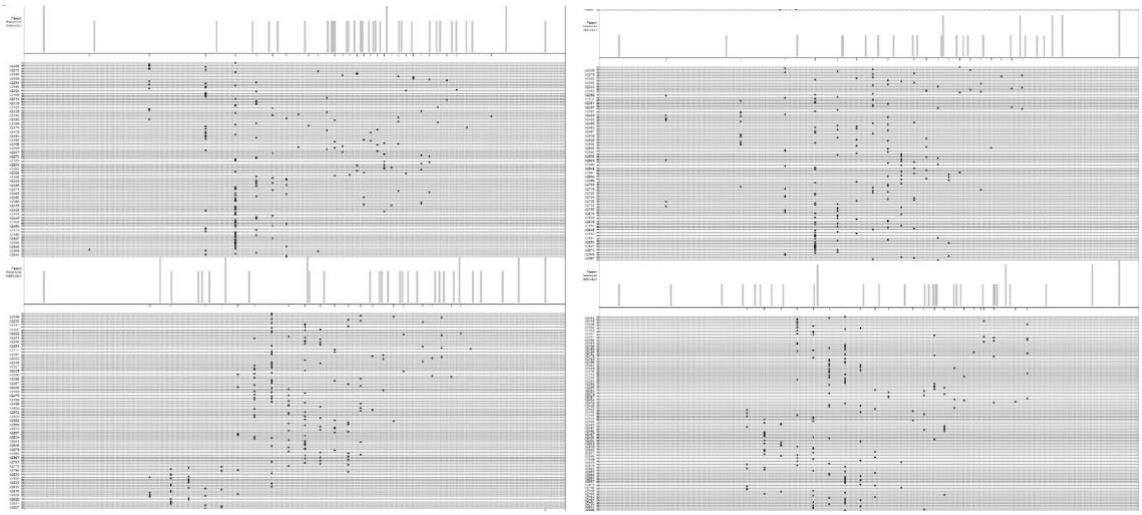
下面的图片是第一组被试者和第一组所有题目的怀特图：



上面的柱形是被试者能力的分布，下面网格线中的一系列点是试题难度的分布。点或者柱形越靠右边，说明试题难度越大或者被试者能力越高。如果一个柱形和一系列点在同一条竖直的铅垂线上，说明柱形对应的被试者做出那些点对应的试题的概率为 0.5。通过分析第一组所有题目的 WrightMap，可以发现：

1. 试题分布为橄榄型，以中等难度试题为主，困难和简单题较少，这也与我们通过前面的分析得出的结论一致。
2. 在模型估算中，许多试题难度相近。
3. 同学能力分布也呈现橄榄型，大部分同学的能力都在中等偏上的程度，他们有大于 0.5 的概率能答出大多数题目；有一小部分同学的能力特别差，连相对简单的题目也无法解答；还有一部分同学的能力特别强，有超过 0.5 的概率能答出所有题目。
4. 柱形分步较广，说明测试出来的学生能力区分度很高，这说明试题总体质量较高。这也与我们之前得出的结论一致。

其他小组的情况与第一组相类似，我们在这里只放出 WrightMap，不再赘述观察到的结论。如下（分别为第二组，第三组，第四组）：



小结

1. 通过观察每道题目各自的 ICC 曲线，发现大多数题目符合逻辑斯蒂曲线，有很高的题目质量，区分度高。
2. 通过观察联合 ICC 图（jointICC），发现每组每种类别的题目难度分布都很完整，从简单到困难的题目在每种类别里面都有体现；每组每种类别的题目分布都呈现橄榄型，简单和困难题目较少，中等难度的题目比较多，这有助于拉开差距，形成区分。
3. 使用 rasch 模型计算出每个组每种类别的题目难度参数，形成一个表格，并根据表格画出折线图，发现每个组的题目难度分布并不完全相同，比如第一组的数组类别比较困难，而第四组和第五组的数组类别却比较简单。我们把这中差异归结为客观因素，即题目难度。我们还发现第三组的题目整体难度低于其他四组。
4. 同样使用第三点得到的表格和折线图，我们发现树结构和图结构的题目相对比较困难，同学们很难解答这些题目。而数组、数字操作的题目相对而言比较简单，同学们较为容易解答这些题目。
5. 使用 R 语言得到了每组的 WrightMap。我们发现一些已经被印证过的结论：试题分布为橄榄型，以中等难度试题为主，困难和简单题较少；WrightMao 中

代表学生的柱形分步较广，说明测试出来的学生能力区分度很高，这说明试题总体质量较高。

6. 通过第五点中的 WrightMap，发现同学能力分布也呈现橄榄型，大部分同学的能力都在中等偏上的程度，他们有大于 0.5 的概率能答出大多数题目；有一小部分同学的能力特别差，连相对简单的题目也无法解答；还有一部分同学的能力特别强，有超过 0.5 的概率能答出所有题目。

总结

结论

在这次研究中，本小组首先在学生提交记录的基础上，使用熵值法和 TOPSIS 法进行了题目难度的数据化；然后在获得题目难度的基础上，使用层次分析法和 TOPSIS 法探究了个人能力；最后使用 Rasch Model 进行了题目难度、题目质量和学生能力的总体分布情况的可视化。

应用场景

本次研究成果可以使同学们在进行 Python 在线编程练习的时候根据我们研究得到的题目难度和题目质量来进行相应练习；也可以使老师直观地看到题目难度、质量的情况和学生能力的情况，便于开展后续教学；也为慕测平台提供了一个参考，便于继续完善平台。

之后如果有更多更详细的数据，也可以用本次研究同样的方法来进行分析，得到个人能力、题目难度、题目质量的相关信息。

建议

1. 保持题目难度和质量的分布，适当增加树结构、图结构等较难类别题目的题量。
2. 把运行结果也记录下来，作为参考依据。
3. 统一规范所有题目的输入输出。
4. 增加测试用例，或者使用随机用例。
5. 统一规范答案所使用的编程语言和风格，提高标准答案质量。

参考文献

[1] 余民宁(2011).试题反应理论（IRT）及其应用.台北市：心理出版社。

[2] 黄丽婷,刘驰.应用层次分析法的半身裙结构设计影响因素权重分析[J].毛纺科技,2020,48(07):67-70.

[3] 杨威.试题的难度和区分度检验与试题库建设──公共《教育学》考试抽样分析[J].肇庆学院学报,2001(02):102-106+110.

[4] Yoon, K. (1987). "A reconciliation among discrete compromise situations". Journal of the Operational Research Society. 38 (3): 277–286. doi:10.1057/jors.1987.44.

[5] Wikipedia contributors, 'Rasch model', *Wikipedia*, *The Free Encyclopedia*, 27 May 2020, 17:45 UTC, <en.wikipedia.org/wiki/Rasch_model>

[6] Wikipedia contributors, 'Rasch model', *Wikipedia*, *The Free Encyclopedia*,19 July 2020, at 13:48 UTC, <en.wikipedia.org/wiki/Item*response*theory>

[7] Hwang, C.L.; Yoon, K. (1981). Multiple Attribute Decision Making: Methods and Applications. New York: Springer-Verlag.

附录

开源地址

https://github.com/JinyuChata/datasci-coursework

数据处理

题目基础数据处理

依据所提供的题目提交数据，我们小组将其按题目分离，获取到题目基本特征作为基础数据：

cases_analysis_source.json // 题目基本数据

```
{
  "2908": {
      "题目名称": "单词分类",
      "题目类别": "字符串",
      "总提交次数": 487,
      "有效提交次数": 99,
      "平均得分": 31.498973305954827,
      "方差": 1643.5436334428252
  },
  "2172": {
      "题目名称": "后缀转中缀",
      "题目类别": "线性表",
      "总提交次数": 145,
      "有效提交次数": 52,
      "平均得分": 47.86206896551724,
      "方差": 1855.4292508917933
  },
  // ...
}
```

cases_test.json // 题目测试用例信息

```
{
  "2061": {
      "测试用例个数": 1,
      "脚本行数": 90
  },
  "2063": {
      "测试用例个数": 3,
      "脚本行数": 17
  },
  // ...
```

```
}
```

小组数据分离处理

依据所提供的题目提交数据，我们将其尝试按组别分离，得到各组同学名录表与各组题目表：

group_count.json // 小组人数分布表

```
{
    "g1": 68,
    "g2": 53,
    "g3": 44,
    "g4": 54,
    "g5": 52
}
```

group_results.json // 小组 - 成员对应表

```
{
    "2843": "g5",
    "3544": "g2",
    "8160": "g1",
    "8246": "g1",
    "8317": "g1",
  // ...
}
```

group_cases.json // 小组 - 题目对应表

```
{
 "g1": [
     "2174",
     "2254",
     "2335",
     "2349",
     "2350",
     "2354",
     "2361",
     "2374",
     // ... 更多题目
 ],
 // ... 更多小组
}
```

目录结构

```
datasci-coursework
├─period0
│   ├─cases
│   ├─data
│   ├─data retrieval
│   ├─group partition
│   │   └─分组
│   └─初期报告
├─period1_wrl
│   ├─Codes
│   ├─Datas
│   └─Plots
├─period1_xzh
│   ├─pca
│   │   ├─Codes
│   │   ├─Datas
│   │   └─Plots
│   └─rasch
│       ├─Codes-all
│       ├─Datas-all
│       ├─group_1
│       │   ├─Codes
│       │   ├─Datas
│       │   └─Plots
│       │       ├─ICCs
│       │       └─wrightmap
│       ├─group_2
│       │   ├─Codes
│       │   ├─Datas
│       │   └─Plots
│       ├─group_3
│       │   ├─Codes
│       │   ├─Datas
│       │   └─Plots
│       ├─group_4
│       │   ├─Codes
│       │   ├─Datas
│       │   └─Plots
│       └─group_5
│           ├─Codes
│           ├─Datas
│           └─Plots
└─period1_zjy
    ├─Codes
    ├─Datas
    └─Plots
```