整体框架.md

# 整体框架

## 评价指标

## 应用方向和未来展望：

## 非常重要：：项目意义！！应该有一页纸：专门将项目意义，应用场景，解决什么问题，立项背景

在现在的学习环境中，有着非常多的小组合作项目，各位程度员也有着寻找编程伙伴的需要，我们的项目适用于学生在学校或院系中系统推荐 适合与自己合作学习的编程cp，以便在各类型的代码题目中取长补短，共同进步

我们的项目仅通过分析本院系学生的作业情况组配在本院系内的编程cp，这种研究方法也可以拓展到工作中，通过分析员工以往的项目完成情况，按照合适的组配方式组成工作小组，以高效地合作完成新的工作项目。

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

我觉得借鉴王者荣耀的匹配和等等竞技游戏的匹配机制：

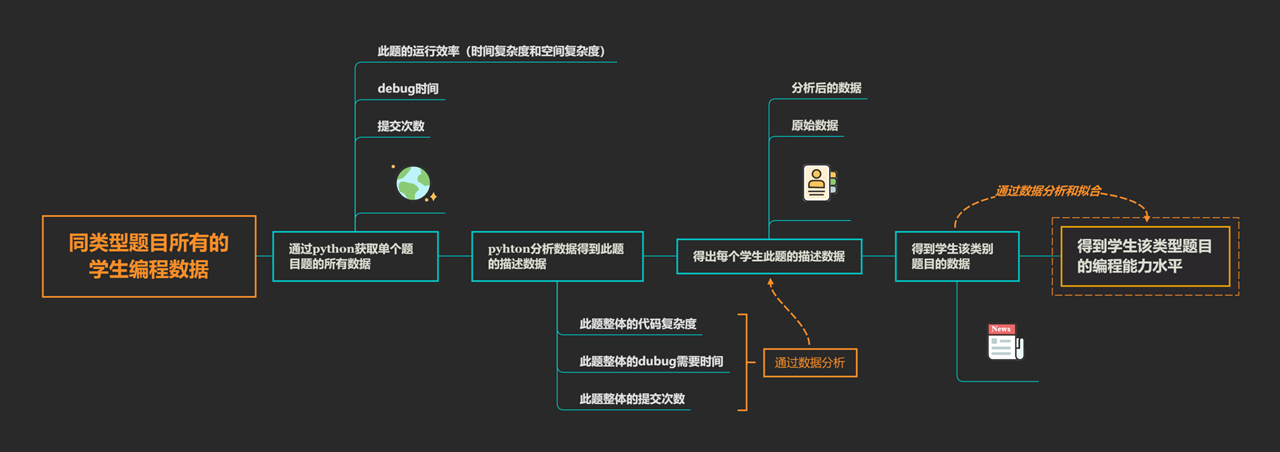
我们的目标应该是一个在线oj或者校内在线oj的刷题系统中的一个子刷题伙伴推荐系统->意味着是时时的，借鉴游戏中的隐藏分rank系统，每个人刷新题后会有个人能力数据更新，同时匹配系统会更新更符合的伙伴，（可以在线交流，如果长期都有同一个人为预期伙伴可以结交成长期伙伴这样子的。校内则更加方便交流。。），同时也可以通过不同的按钮->用户预期匹配方式：比如说喜欢旗鼓相当的伙伴或者互补型伙伴。。。得到不同的匹配结果。。。

。。补充，同时每个人也可以看自己的能力分布水平。。有图形化显示，借鉴lol等的个人战力系统。

### 题目数据得到：

#### 每个人编程值组：

每个个人数据都是，**该题总数据，综合出来该题整体数据**，再**个体题数据去比较得出，个体题的相对分值或绝对分值**



#### 

补充：目前思路：：每个题通过建模得出（通过该题总体数据）得出三项评价，并通过建模：：可以是直接叠加（这个模型我觉得不好建立：：直观上叠加。。。太简单）：得到学生该类题的三个能力值评价

可以对该类题的三个能力建模得出综合评价

最后匹配使用的数据。。可以所有debug叠加也可其他也可。无所谓。还没到那部分。。。。。

最终结果是：

。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

借鉴对战moba类游戏匹配算法，每个人每类题可以多一个波动水平，利用方差（最简单）

数据处理：

每个题：同类型题得出该学生该具体题的三项评价（目前的数学工具已经具备）：：：topsis法+熵权法

然后是，该学生某一具体类型如：图算法：旗下1到20题的三项评价数据（每项20个）得到该**学生该类题的 如：图算法的三项评价指标**

：：：熵权法

到这个层次，每个学生下面（如果）有6类题，每类 就有三个大的评价数据，可以用来匹配（见下：：组配）。。

然**后**数据综合处理一下，6类题18个数据得到**学生总的三项数据的能力值**

然后进行组配（待寻找）：：：RSR法

组配：可以通过推荐算法（交友网站或者刷题网站？）来进行借用（寻找again）

借鉴：对战moba类游戏匹配算法

目前想法：可以给每个人贴标签：debug能力强，或者数组能力差。

通过算法和机器学习给学生进行

（组配方式1）相似度匹配，越相似的可以进行结伴编程，

做一个程序。。输入数据，，得出每个学生擅长的题目类型做成标签（可以多个）（给每个学生贴标签：：比如擅长图算法。。）->如何判定擅长：：可以是。该学生类题的综合数据在前30%->复杂化的话。可以通过该类题的三个数据赋权重，重新排序，得出前30%为擅长该类题

（组配方式2）互补型匹配：：匹配出来的两个人应该尽可能使两方的最大边框补满六边形

等待补充其他匹配方式

。。。。。。

A：

图算法：

Debug能力（类型熟练能力）

构建代码速度。。（如果说熟练度的话。。岂不是跟上面重复）

该类题的算法构建能力

图综合评价

数组：

Debug能力（类型熟练能力）

构建代码速度。。（如果说熟练度的话。。岂不是跟上面重复）

该类题的算法构建能力

数组综合评价

。。。。

#### 对此题：debug能力

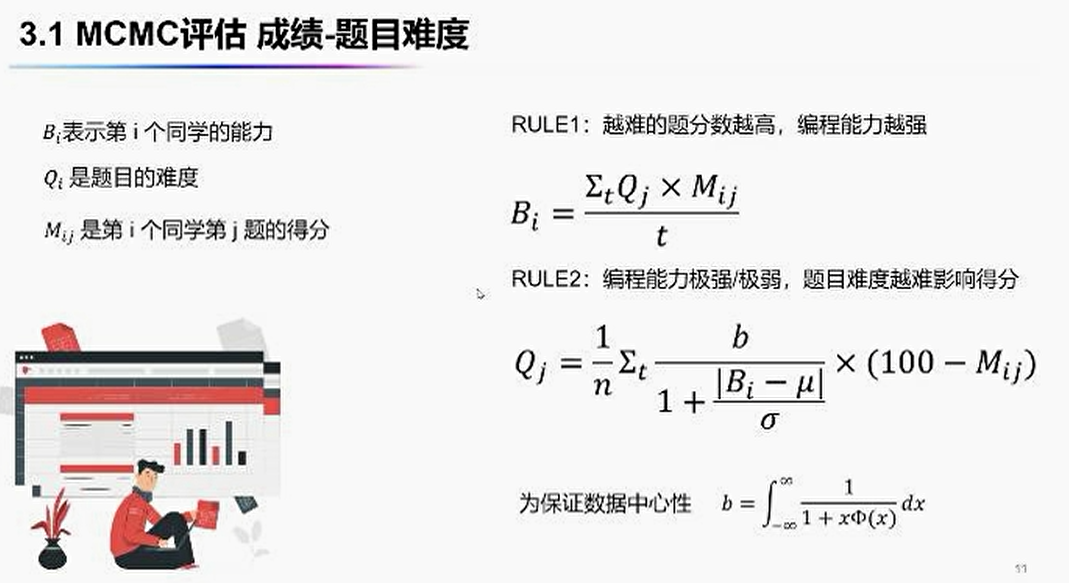
* debug得分能力值：提交次数和得分涨幅，综合上时间

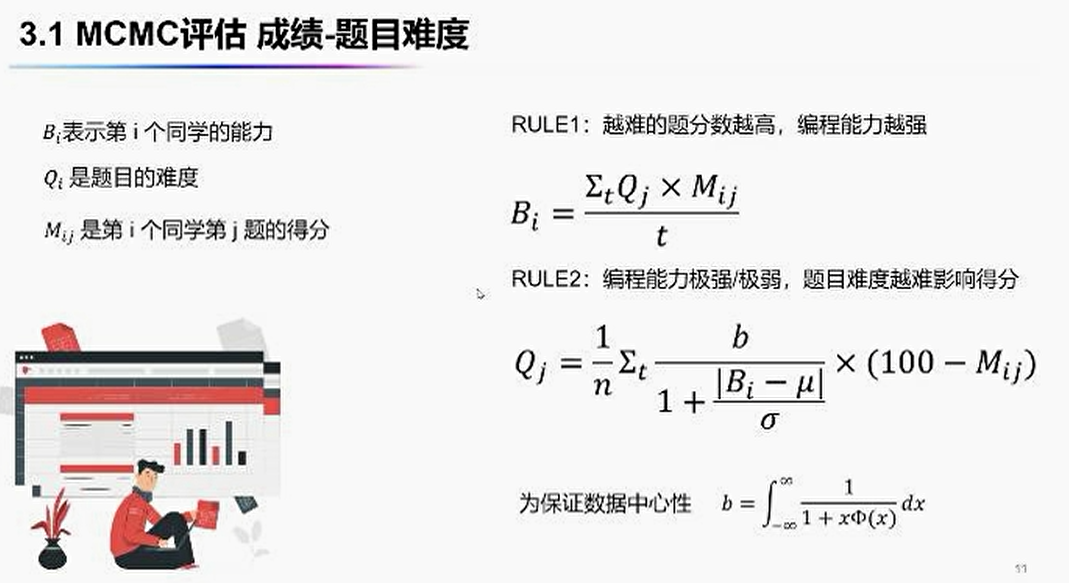
#### 第一次构建代码能力：

* 第一次提交时间 极小型
* 第一次提交得分 极大型
* 第一次提交行数 极小型

#### 算法构建能力：

* 假设会有写代码时间.
* 代码行数和运行时间两纬：可以作为该题效率的标准.
* 时间复杂度，空间复杂度。。。

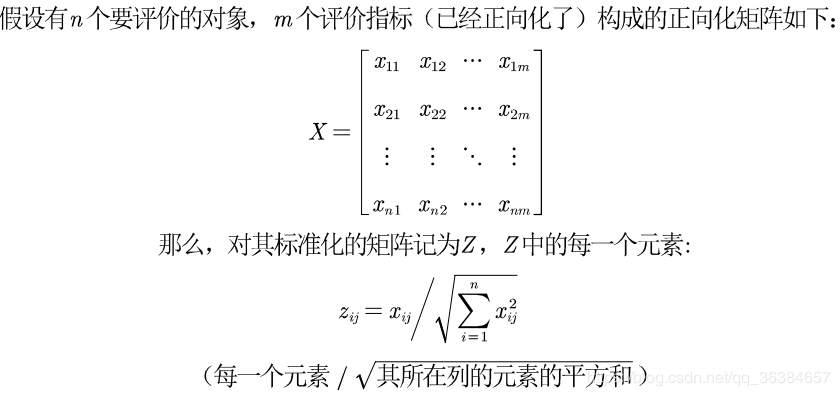


* + 难度建模
  + 视频6. 14:分51秒处!
  + 
  + 难度建模
* 最后是分值映射：：可有可无

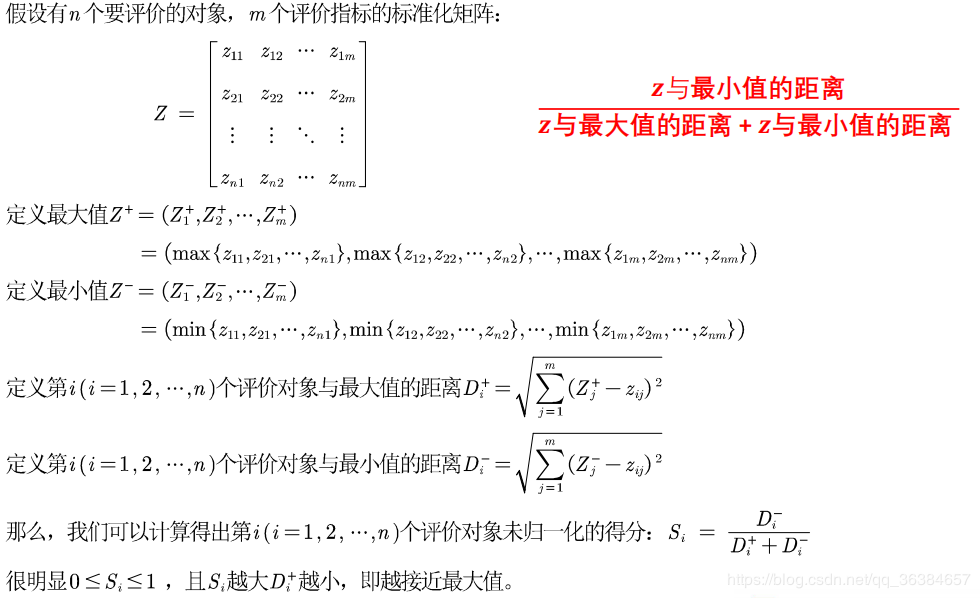
### 1、如果有多个指标且不全是极大型指标，则进行指标正向化。

### 2、之后对正向化矩阵进行标准化，目的是消除不同指标量纲的影响。

正向化方法：



### 3. 计算得分



### 4.归一化评分

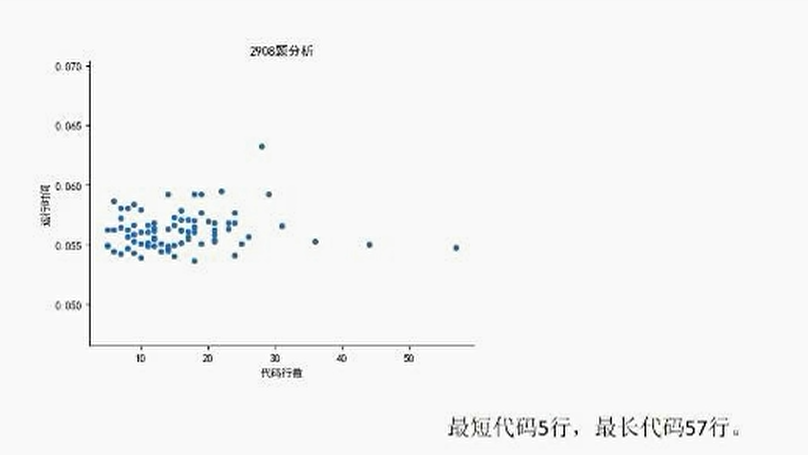
   即每个分数除以所有分数和

Topsis评价法：<https://blog.csdn.net/qq_36384657/article/details/98188769>

熵权法：<https://blog.csdn.net/luxurie/article/details/105877495>

#### 编程风格

## 使用方法：尚未分配的

* 总体能力曲线：：通过个体数据的折线：：拟合出来。。。方法未知，应用尚无
* 离散点图作为中间过程
* 
* 每一题的点阵图？？

## 改进部分

我觉得可以利用面向用例：：如果一道题面向用例，能力置一个低分，题目难度，该题总体上面向用例的数目，也可以作为题目难度的一个分值...改进点

* 每类题目的总体值：：类中每个题构成
  + 每类题目中每个题会有一个能力值：
    - 题目完成率 debug时间（第一次提交时间）
    - 提交次数 做题顺序
    - 原码时空损耗

### 得到组配：

#### 能力比对

#### 相关算法匹配

#### 组配方式：

**互补型匹配（双人）**

* 即两人编程水平总体相差不大，可以取长补短

**主程序员型匹配（多人）**

* 即组成的团队中一个人的编程能力很强，可以主导整个团队

**以强带弱性匹配（双人）**

* 即两人中一方编程水平较强，可以带动另一方

**民主互补型匹配（多人）**

* 即组成的团队成员实力相当，可以优势互补，合作学习

小组信息：人数 学号 姓名 邮箱 \_Python 练习完成题目数量 组员分工职责 1.

研究问题 你研究的问题，包括但不限于如何选择学生的编程能力评价指标； 如何在编程过程中推荐，学习代码； 如何自动生成编程学习路径等等 ，请简述问题的应用场景 2.

代码开源地址 给**出分析代码的地址，并且解释代码与研究问题的对应关系， 代码的实现逻辑 。** 3. 研究方法（重点） **包括你用到的数据分析方法、所使用的数据集，及其他你所用到的一切方法。这部分需要你的详细说明， 要 做到逻辑清晰且易理解。** 4. **案例分析 针对研究问题的分析，格式不作限制。** 5. 对这门课的意见和想对老师说的话， 也欢迎感兴趣的同学们 申请 加入实验室的研究。 6. 附录 你认为需要补充到研究报告中帮助读者理解的数据、图表等。 注：以上为研究报告的标准格式，如果你有特别的思路，也可以适当调整报告的结构。 提交 PDF 格式，宋体小四， 1.5 倍行距。