

Project 设计报告

课程名称: 数值分析

学 院: 信息与电子工程学院

专 业: 信息工程

班 级: 信息工程 1602

姓 名: 钱泽诚 学 号: 3160102418

指导教师: 余官定

学年学期: 2017 ~ 2018 学年 第 1 学期

2017 年 11 月 15 日

目 录

1	设计目标	错误!未定义书签。
2	设计思路	错误!未定义书签。
3	算法描述	错误!未定义书签。
4	界面效果图	错误!未定义书签。
5	应用的性能描述	10

设计目标：

根据国家卫生部在 2009 年 6 月 2 日公布的《中国 7 岁以下儿童生长发育参照标准》，设计一个能够查询年龄在 0~84 月之间小孩标准身高和体重的应用。根据用户输入的小孩生日以及性别信息，反馈给用户相应年龄小孩的标准身高和体重。若用户希望能测试身高和体重是否标准，应用也可以提供测试的模块，根据用户输入的身高和体重，与《参照标准》中所提供的 SD 偏差进行比对，反馈给用户身高和体重是否正常。

设计思路：

根鉴于《参照标准》中提供的身高和体重样本数据较多，利用高次多项式做拟合可能会出现不可预料的偏差，高次多项式函数在相邻数据点之间的图像波动可能会比较大，不能准确反映相应年龄小孩的身高和体重。因此我选用了三次样条的方法来拟合数据。基本思路是：对表格中给定数据的每一栏（中位数、+1SD、-1SD……）分别用自然三次样条拟合，对每个表格再做三次样条拟合。所以共有 $4 \times 35 \times 7 = 980$ 个三次函数。然后根据用户的输入去计算多项式的结果，进行比对并反馈给用户。

我用到的编程语言是 python，版本是 3.6.2,主要用到了符号计算模块 sympy(1.1.1)，以及内置的图形界面模块 tkinter，正则表达式模块 re。需要说明的是，sympy 模块不是内置的，因此需要在其它的计算机上运行脚本，需要手动安装此模块。

设计主要分为两块，其一是多项式的拟合，其二是图形界面的设计。我在 `coefficient.py` 脚本中完成多项式的拟合，每次拟合的结果都是一个表中某一列对应的三次样条，输出结果为一个符号多项式的列表。由于项目提供的数据是图片的形式，因此无法直接导入，每次做拟合的时候需要手动输入对应列的数据，附件中“男童”、“女童”文件夹即为对应的脚本。然而就像神经网络的训练一样，拟合的过程只需要进行一次，我们需要的只是拟合出来的结果。拟合完成成，我把相关的各类多项式都存入了 `function.py` 文件中。然后是图形界面

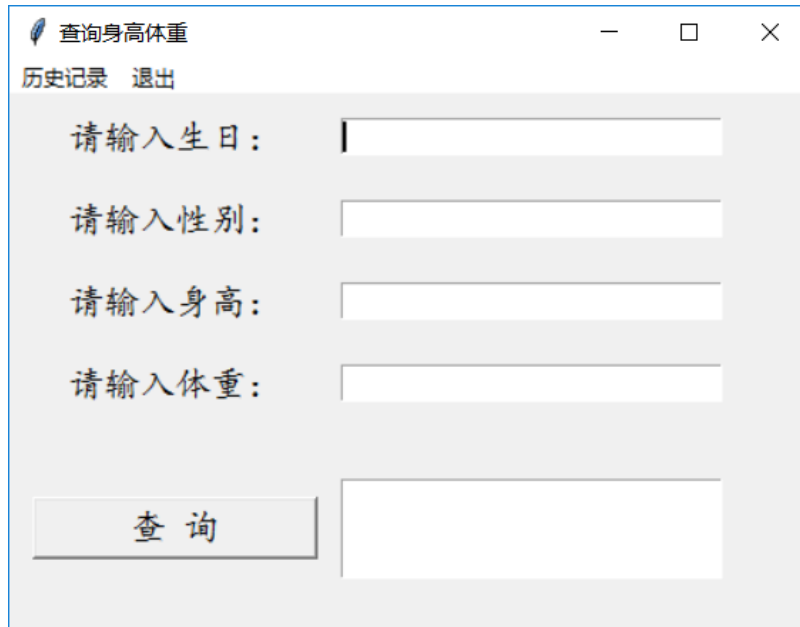
的设计，这个主要是在 `main.py` 函数中完成，我用 `tkinter` 来进行图形界面的设计，界面是一个窗口，上面的菜单栏“历史记录”可以用来查询用户曾经的查询记录，退出菜单将结束脚本的运行。主要界面由标签、输入框、按钮和文本框组成，接受的输入信息有：生日、小孩的性别、身高和体重。生日的输入格式例如‘2015/5/12’ 性别为男或女，身高体重输入数字即可(合法输入为正数)。然后点击查询按钮可以对于生日的输入，我先用正则表达式来匹配用户的输入格式是否合法，若合法，再检测用户输入生日对应的年龄范围是否在 0~84 之间，若所有输入都合法，才会再查询按钮右侧的文本框处显示查询结果。点击菜单栏上的历史记录后，也将再文本框显示用户的查询记录（若存在）。

算法描述：

算法主要用到的数据结构是 `python` 的列表。在函数的拟合模块（`coefficient.py`）中，通过解自然三次样条的线性方程组，获得拟合多项式的系数并存放到另一个列表中。4 个表格的所有拟合结果都被存放到 `function.py` 脚本中，然后在 `main.py` 文件（图形界面设计）脚本中，导入 `function` 模块就可以进行查询，查询至少需要两个数据，即生日和性别。将用户输入的生日转化为月龄，然后用 `function.py` 中的拟合结果查询标准的身高和体重，输出到文本框中反馈给用户，同时也存放到历史记录的列表中，如果用户输入了身高或体重，则再进行查询和比对，并把比对结果也反馈给用户。历史记录列表可供用户随时访问曾经的查询历史。在图形界面的设计中，我用到了标签组件提示用户输入，文本框打印查询结果和历史记录，查询按钮进行查询，并在文本框显示打印结果的时候设置为只读模式，防止用户随意更改打印出来的内容。用到了正则表达式匹配用户输入的生日格式。

界面效果图：

1、脚本运行时最初的界面：



The image shows the initial state of a Windows application titled '查询身高体重' (Query Height and Weight). The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar, there are two links: '历史记录' (History) and '退出' (Exit). The main area contains four input fields with labels: '请输入生日:' (Please enter birthday), '请输入性别:' (Please enter gender), '请输入身高:' (Please enter height), and '请输入体重:' (Please enter weight). At the bottom left is a '查询' (Query) button, and at the bottom right is a large empty rectangular box for displaying results.

2、输入生日和性别并点击查询按钮：



The image shows the application after the user has entered '2015/5/12' for the birthday and '男' (Male) for the gender, and clicked the '查询' (Query) button. The '标准身高为: 93.6' (Standard height is: 93.6) and '标准体重为: 13.7' (Standard weight is: 13.7) are now displayed in the result box at the bottom right. The other input fields (height and weight) remain empty.

3、输入身高信息并点击查询按钮：

查询身高体重

历史记录 退出

请输入生日：

2015/5/12

请输入性别：

男

请输入身高：

92

请输入体重：

查 询

标准身高为：93.6
标准体重为：13.7
身高属于正常范围

查询身高体重

历史记录 退出

请输入生日：

2015/5/12

请输入性别：

男

请输入身高：

89

请输入体重：

查 询

标准身高为：93.6
标准体重为：13.7
偏矮

4、输入体重信息并点击查询按钮：

查询身高体重

历史记录 退出

请输入生日：

2015/5/12

请输入性别：

男

请输入身高：

89

请输入体重：

13

查 询

标准身高为：93.6
标准体重为：13.7
偏矮
体重属于正常范围

查询身高体重

历史记录 退出

请输入生日：

2015/5/12

请输入性别：

男

请输入身高：

89

请输入体重：

14

查 询

标准身高为：93.6
标准体重为：13.7
偏矮
偏重

5、点击菜单栏上的历史记录：

应用的性能描述：

性能分析主要分为两块，一块是拟合函数时的算法性能分析，另一块是图形界面算法调度的性能分析。

关于准确度，由于拟合多项式的系数都是用 python 的 float 型存储的，在 python 的 IDLE 中用 sys 包的 getsizeof 方法查看得 float 型用 24 位存储，

```
>>> import sys
>>> sys.getsizeof(1.0)
24
>>>
```

因此拟合结果毫无疑问精度是非常高的。在实际的输出中，由于接受的用户输入身高保留一位小数，体重保留两位小数，因此我用四舍五入(round)的方法让输出结果也相应地保留了一位、两位小数。如果需要更高的精度，可以在 main.py 脚本文件中找到调用 round 函数的地方修改格式化输出的精度，这是可由用户自定义的。另一方面，由于采用的拟合方法

是自然三次样条，因此在表格的给定数据点处的结果是完全精确的，而在其它数据点处，也能够起到一定的预测作用，而在边界处，由于是自然三次样条，满足边界点的一阶导数为 0，因此在边界外（81 个月 < 月龄 ≤ 84 个月），预测结果可能与期望值有一定的偏差，因为 81 个月 < 月龄 ≤ 84 个月同样用的是最后一段三次样条。如果容许误差较大，结果是可以接受的，如果有更进一步的需求，可以考虑对最后一段的结果做一些优化，或者在均衡算法性能和准确性的基础上采用其他的拟合方法。

关于做三次样条拟合的模块(coefficient.py 脚本)主要用了矩阵求逆(求解线性方程组)的方法，时间复杂度为 $O(n^3)$ ，如果做矩阵 LU 分解的优化，可以把时间复杂度降为 $O(n^2)$ ，但这个不是我们需要关心的，因此所有的数据只需要拟合一次而已，而且表格中数据的规模也是确定的，无论是否做优化，算法都可以在用户可接受的运行时间内给出运算的结果，由以上分析，在用户真正接触到的图形界面中，只需要进行简单的索引即可，因此时间复杂度是在 $O(n)$ 级别的，用户体验是非常好的。我在 coefficient.py 脚本中用 time 模块测试了拟合的运行时间，下面几张图是测试的结果（单位为秒）：

```
///
= RESTART: C:\Users\Dell\OneDrive\课程\数值分析\project\Py program\coefficient.p
y =
11.953259111111112

///
= RESTART: C:\Users\Dell\OneDrive\课程\数值分析\project\Py program\coefficient.p
y =
14.470109777777777
\\ \

///
= RESTART: C:\Users\Dell\OneDrive\课程\数值分析\project\Py program\coefficient.p
y =
14.100972888888889

>>>
= RESTART: C:\Users\Dell\OneDrive\课程\数值分析\project\Py program\coefficient.p
y =
12.705480444444445
\\ \

///
= RESTART: C:\Users\Dell\OneDrive\课程\数值分析\project\Py program\coefficient.p
y =
14.241858666666667
>>>
```

Project 设计总结

在此次 project 的设计中，我选用了 python 这门语言，因为 python 有其他编程语言不可比拟的优点，它拥有强大的可扩展性，非常灵活，有各种编写好的数据结构（列表、元组、字典），索引起来也十分方便。在图形界面的设计环节，我选用了 tkinter 包，但用到后来才方面 tkinter 这个包属于比较初级的图形界面设计包，pyqt 的可定制性会更强一些，但由于时间所限，就没有再改用 pyqt。总的来说，由于测试数据量不是很大，因此对算法的性能要求不是特别高，如前所述，只需要拟合出多项式即可，在用户接触的图形界面中，只需要进行相应的算法调度，而相应的函数已经在之前的拟合中已经得到了，因此用户体验到的程序运行时间实际上是非常短的，只需要简单地索引即可，没有其他的高额时间花费，用户的体验是非常良好的。