文章编号：2096-1472(2020)-02-39-04
DOI:10.19644/j.cnki.issn2096-1472.2020.02.011

基于sigmoid函数的题目难度值动态调整算法的设计

莫怀训[1]，刘晓瑞[2]

(1.广东省轻工职业技术学校，广东 广州 510310；
2.广州城市职业学院，广东 广州 510405)

摘　要：本文主要针对各级各类信息化教学或考试系统中的题库题目难度值进行讨论，分析如何进行分布计算并动态调整。本文将难度值定义为错误率，即错误次数/调用总次数，再进一步分析发现根据本定义直接进行计算的方法需要进行改进，可用sigmoid函数进行拟合，并在此基础上设计出分布计算算法和线性叠加算法。通过本方法，各终端无须额外存储其他数据，仅根据当前题目回答错误情况即可更新难度值；同时也无须连线中心服务器，可独立分布计算，然后线性叠加即可获得最终难度值。因此可在一定程度上降低整个题库系统设计的难度，同时也可以看出本方法对存储敏感型和去中心的任务具有一定参考价值。

关键词：题目难度值；sigmoid函数；分布计算；线性叠加
中图分类号：TP301　**文献标识码**：A

The Design of Dynamic Adjustment Algorithm of Difficulty Value Based on Sigmoid Function

MO Huaixun[1], LIU Xiaorui[2]

(1.*Guangdong Light Industry School*,*Guangzhou* 510310,*China*;
2.*Guangzhou City Polytechnic*,*Guangzhou* 510405,*China*)

Abstract: This paper mainly discusses the difficulty value of questions in various levels of information-based teaching or examination system,and analyzes how to calculate the distribution and adjust it dynamically.In this paper,the difficulty value is defined as the error rate,that is,the number of errors/the total number of calls.Further analysis shows that the method of calculation directly according to this definition can be improved.The sigmoid function is used for fitting,on this basis,the distribution calculation algorithm and the linear superposition algorithm are designed.Through this method,each terminal does not need to store additional data,and the difficulty value can be updated only based on the answer to the current question.At the same time,there is no need to connect to the central server.The calculation can be independently distributed,and then the final difficulty value linear can be obtained through linear superposition.Therefore,the difficulty of designing the entire test database system can be reduced to a certain extent.At the same time,it can also be seen that this method has certain reference value for storage-sensitive and decentralized tasks.

Keywords:difficulty value;sigmoid function;distribution calculation;linear superposition

1 引言(Introduction)

在信息化教学系统及目前各级各类等级考试系统中，题库是其中的重要内容，它承担着对学员训练、考核以及选拔等重要功能。题库中基本元素就是题目，而题目的形式、内容、难度等特征就是题库根据考核目标抽取题目组成测试卷需要考量的基础数据。其中试卷整体难度值和试题难度分布情况是组卷系统考虑的非常重要的两个指标，而试卷整体难度值、难度分布很明显是基于试卷各题目难度值来计算的[1]。

从实际情况来看，我们一般根据经验或过往的大量统计数据来确定一道题目的难度值，很明显这样效率低下而且往往并不一定符合实际情况。特别是一道题目面对不同学员群体时，难度值实际上是不同的，例如某一道电工计算题对于

基金项目：广东省教育科研“十三五”规划２０１８年度研究：基于“互联网＋教育”背景下的中职机电专业电类基础课程的知识图谱构建与应用研究；项目编号：２０１８ＹＱＪＫ３６５．

职业学校普通学生可能难度值较高，但对于经过培训的考证学员可能难度值中等。因此我们需要针对难度值设计一种计算方法，使得题目难度值在使用中动态调整，并逐渐符合实际情况[2]。

2 题目难度值的定义(Definition of difficulty value)

对于一道题目，其难度值可以从得分率、平均完成时长、错误率等维度去考虑。目前各级各类理论考试题库中，最常见的题型是选择题，实际上填空题、计算题、判断题等都可以变形为选择题而不减弱其测试效果，所以很多理论题库甚至只有选择题[3]。而对于选择题，一般不存在部分得分，因此使用错误率去代表难度值就比得分率合理。另外在一般考试过程中，试题作答没有顺序要求，考生可以根据实际情况灵活选择试卷前后的题目来作答，因此准确统计一道题目的平均完成时长较为困难。所以在本文中，题目难度值主要根据本题在实际使用中的错误率来代表[4]，公式如下：

难度值=错误率=错误次数/调用总次数 (1)

其中，总次数=错误次数+正确次数。错误率越高代表难度越高，调用的次数越多该值会越趋近于实际难度。从公式可以总结出难度值分布应该具备的基本规律如下：

难度值在0—1。

当错误次数趋近于总次数时，难度值趋高，为1时表示难度最高。

当正确次数趋近于总次数时，难度值趋低，为0时表示难度最低。

当正确次数=错误次数时，难度值=0.5，与总次数无关。

3 难度值计算方法的初步分析(A preliminary analysis of the calculation method of difficulty value)

对于一道未知难度的题目，最开始很自然可以假设学员做错的概率与正确的概率一样。即假设如已经调用次数为10000，则错误次数为5000，那么难度值初始值为0.5。

难度值随着错误次数增加的表达式$y=(5000+x)/(10000+x)$，如图1所示。难度值随着正确次数增加的表达式$y=5000/(10000+x)$，如图2所示。根据表达式易看出，调整次数初始假设值，对曲线的基本形态没有影响，但对难度值随着错误次数变化快慢有较大影响。

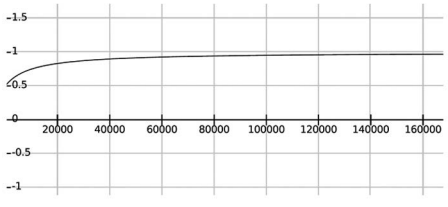


图1 $y=(5000+x)/(10000+x)$

Fig.1 $y=(5000+x)/(10000+x)$

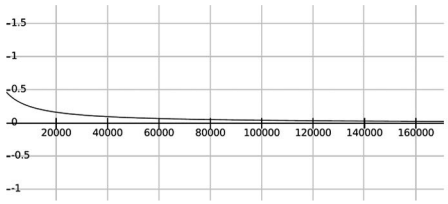


图2 $y=5000/(10000+x)$

Fig.2 $y=(5000+x)/(10000+x)$

基于公式1的计算方法在实际使用中有一些缺点。

首先我们需要针对每道题单独存储错误次数及调用次数，随着系统规模的扩大和用户使用人次的增加，最终可能会导致存储需求超出预算；其次错误次数增加及正确次数增加，实际表达式不一样，不利于将来对系统进一步的分析；最重要的是如果起始假设总调用次数太小，则难度值对错误次数太敏感，波动大，起始假设总调用次数过大，则难度值又变化不明显，两者都会导致动态调整时间过长，对系统使用造成不便，而应该假设为多少则难以进行合理分析。

4 难度值计算方法的优化(Optimization of calculation method)

4.1 基于sigmoid函数的拟合

仔细考察上面图1和图2。如果我们把错误次数定义为正值，正确次数定义为负值，即图2以y轴对称翻转，然后把它们连接在一起。我们可以很容易发现连接图形与sigmoid函数图像非常相似，如图3所示[5]。

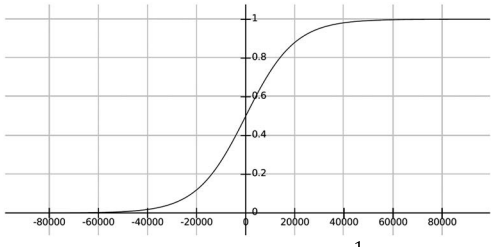


图3 Sigmoid函数 $y=\frac{1}{1+e^{-kx}}$

Fig.3 Sigmoid function $y=\frac{1}{1+e^{-kx}}$

因此我们采用sigmoid函数来代替上面原始定义的计算方法：

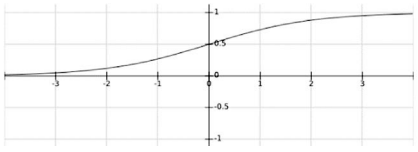
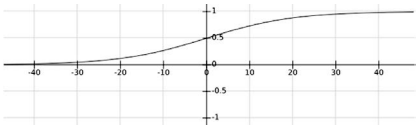
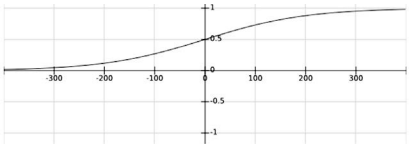
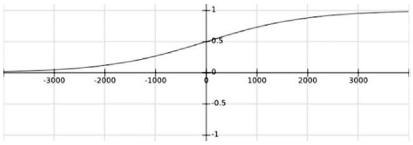
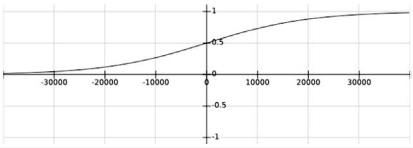
横坐标为错误净值即x=错误次数−正确次数，纵坐标$y=\frac{1}{1+e^{-kx}}$为难度值。

当$x=0$时，难度值$y=0.5$；当x趋近于$+\infty$时y趋近于1；当x趋近于$-\infty$时y趋近于0。

其中系数k可以调整曲线上升的快慢，也就是通过调整系数k可以让y更好拟合实际难度值的变化，该系数可命名为精度系数，对系统的影响如表1所示，可以看出一般情况下$0<k<1$。

表1 精度系数k的影响

Tab.1 The influence of precision coefficient k

精度系数k值	图形
1	
0.1	
0.001	
0.0001	
0.00001	

4.2 精度系数k的估算

精度系数主要根据题库面向对象的范围来确定。

例如本项目题库主要面对本校中职生源，而我校一届机电类中职生约300人，本题库题目生命周期暂定为10年，则对应人数约3000人，每人每学期针对某一道题可能会作答四次，即学习时一次，复习时一次，期中测试一次，期末考试一次。也就是每道题可能会被作答12000人次。假设对于某道题，有95%的同学人次错误，即大约有11000次错误，则错误净值x=11000−(12000−11000)=10000，难度值y应约等于1；同理如有95%的同学正确，则x=−10000，y约等于0。

查阅表1可知k取值范围在(0.00001—0.0001)。经过实验，k取0.0004，如图4所示。从图中可以看出，此时当$x\geqslant 10000$时，难度值$y\approx 1$；当$x\leqslant -10000$时，难度值$y\approx 0$；当x=0，即一半人错误一半人正确时，难度值y=0.5。曲线对难度值变化拟合度非常高，完全符合本文第2点对难度值总结的四条基本规律。

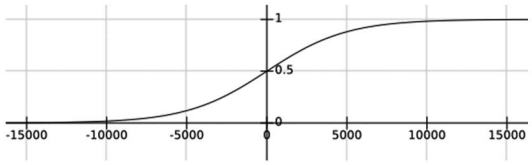


图4 k=0.004

Fig.4 k=0.004

为了更好更快的估算k值，我们需要进一步推导出其估算公式。推导过程如下：

假设有大于95%的同学人次错误时，此时难度值y应该大于95%。

设总次数为n，那么有错误净值$x=n*95\%-n(1-95\%)=n*90\%$。

$$\begin{aligned}&\text{难度值}y=\frac{1}{1+e^{-kx}}\geq 95\%\Rightarrow\frac{100}{95}\geq 1+e^{-kx}\\&\Rightarrow\frac{1}{19}\geq e^{-kx}\Rightarrow\ln\frac{1}{19}\geq -kx\\&\Rightarrow kx\geq\ln 19\Rightarrow 0.9nk\geq\ln 19\\&\Rightarrow k\geq\frac{\ln 19}{0.9n}\Rightarrow k\geq\frac{3.2716}{n}\end{aligned}\tag{2}$$

我们取$k=\frac{3.2716}{n}$，由于n通常很大，所以一般精确到第一位有效数字即可。

4.3 分布计算的合并处理

我们题目难度值可以在后台根据数据统一计算，这种方法比较简单，但需设置一个中心服务器来处理。

如果难度值与错误净值在一定的范围内呈线性关系，那么难度值可分布在各终端独立计算，需要合并时线性叠加即可。观察图5可发现难度值变化曲线是分3段近似线性的。因此我们可以将其线性化，得到叠加计算方法。

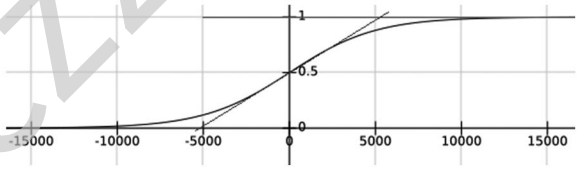


图5 线性化

Fig.5 Linearization

线性化及求叠加算法过程如下：

(1)难度值$y=\frac{1}{1+e^{-kx}}$曲线的拟合直线。

当x趋于∞，若lim$y$=1，则有水平渐近线y=1；

当x趋于−∞，若lim$y$=0，则有水平渐近线y=0。

对于点(0，0.5)两侧邻域的曲线，其一阶导数$\dot{y}=ky(1-y)=k\cdot 0.5(1-0.5)=\frac{k}{4}$，二阶导数$\ddot{y}=k^2y(1-y)(1-2y)=0$，也就是说对于该点两侧邻域的曲线等价于直线$y=\frac{k}{4}x+b$，将该点代入后可得$b=\frac{1}{2}$，所以我们采用直线$y=\frac{k}{4}x+\frac{1}{2}$来拟合点(0，0.5)两侧的曲线。

随着邻域的不断扩大，拟合误差也不断增大。当拟合直线$y=\frac{k}{4}x+\frac{1}{2}$与$y=1$、$y=0$分别相交后，拟合直线分别转变为$y=1$、$y=0$。交点分别为$(\frac{2}{k},1)$、$(-\frac{2}{k},0)$，因为$\Delta y=\left|\frac{k}{4}x+\frac{1}{2}-\frac{1}{1+e^{-kx}}\right|$，所以$\dot{\Delta y}=\frac{k}{4}-ky(1-y)=k(\frac{1}{2}-y)^2\geq 0$，故此时误差达到最大$\Delta y=\left|\frac{1}{1+e^2}\right|\approx 0.12$。如果我们对误差要求较高，则应适当调整拟合直线的斜率，使得最大误差符合我们的要求。

因此如果误差$\Delta y\approx 0.12$合适的话，我们可以将难度值$y=\frac{1}{1+e^{-kx}}$曲线简化为三段折线：以点(0，0.5)为中心，在$x\in(-\frac{2}{k},\frac{2}{k})$的范围内，近似为直线$y=\frac{k}{4}x+\frac{1}{2}$；在$x\in[\frac{2}{k},\infty)$的

范围内近似为直线y=1；在$x \in (-\infty, -\frac{2}{k}]$的范围内近似为直线$y$=0。

(2)叠加算法

假设目前对于某道题，我们从两个终端分别获得难度值y_1、y_2，那么叠加方法如下：

对于区间$(-\frac{2}{k}, \frac{2}{k}) \approx (-0.61132n, 0.61132n)$，其中$n$为可能的总人次数，而难度值函数自变量$x$是错误净值，也就是$x$=错误人次数−正确人次数，所以一般来说本区间应该覆盖了x的很大部分取值范围。因此我们首先考虑以直线$y = \frac{k}{4}x + \frac{1}{2}$来进行叠加。对于$y_1$，有$x_1 = \frac{4}{k}(y_1 - \frac{1}{2})$，对于$y_2$，有$x_2 = \frac{4}{k}(y_2 - \frac{1}{2})$，所以$y = \frac{k}{4}(x_1 + x_2) + \frac{1}{2} = \frac{k}{4}\left[\left(\frac{4}{k}\left(y_1 - \frac{1}{2}\right) + \frac{4}{k}\left(y_2 - \frac{1}{2}\right)\right)\right] + \frac{1}{2} = y_1 + y_2 - \frac{1}{2}$，从本式可以看出，拟合直线斜率的调整完全不影响叠加算法的结果。

如果$y = y_1 + y_2 - \frac{1}{2} \geq 1$，则表明难度值进入到拟合直线$y$=1的区域，故此时$y$=1；

如果$y = y_1 + y_2 - \frac{1}{2} \leq 0$，则表明难度值进入到拟合直线$y$=0的区域，故此时$y$=0。

5 算法设计(Design of algorithm)

5.1 递推计算方法的推导

如果考虑到存储优化及计算过程优化，我们往往希望不需存储所有次数的情况，只要存储最新的操作情况即可更新难度值，那么我们必须设计出递推算法。其推导过程如下：

容易导得难度值函数$y = \frac{1}{1+e^{-kx}}$的微分$\dot{y} = ky(1-y)$。

而$\dot{y} = \frac{y(x+\Delta x) - y(x)}{\Delta x}$，在本问题中自变量$x$每一次最小的变化为1，即$\Delta x = 1$。

所以$\dot{y} \approx y(x+1) - y(x) = y(x) - y(x-1)$，因此可得难度值递推公式如下：

$$y(x) - y(x-1) = y(x+1) - y(x) \approx ky(x)(1-y(x)) \quad (3)$$

其中，$y(x)$为当前难度值，$y(x+1)$为错误次数+1的难度值，$y(x-1)$为正确次数+1的难度值，k为精度系数。

因此本算法只需要存储当前难度值，根据当前操作情况即可更新。

5.2 伪代码

```
y=0.5;//预估难度系数初值为0.5
k=0.0004;//根据题库面对的对象人次数估算精度系数
function difficulty(x)//难度值计算函数
{
if x==-1 then//作答错误
                y=y+k*y*(1-y)
else
                y=y-k*y*(1-y)
end if
return y
                }
function superpose(y1,y2)//难度值叠加
{
y=y1+y2-1/2;
if y>1 then  y=1;
            if y<0 then  y=0;
return y
                }
```

6 结论(Conclusion)

本算法的优点在于动态调整算法简单可行，对于题目只需要存储当前难度值，然后根据当前作答的正确或错误情况动态调整即可；同时本算法可以使得难度值计算分布在各终端进行，当进行合并时只需线性叠加即可，特别的是，我们发现拟合直线根据精度要求调整斜率完全不影响叠加算法。另外给出了拟合实际数据的精度系数k的预估方法，并有较合理可信的解释。因此本算法的优化方法对一些存储敏感型或去中心化的任务具有一定的参考价值。

本算法的缺点在于主要只考虑了错误率对于题目难度值的影响。对于某些可能采用单题应答时间限制或者可以部分得分的考试形式，题目难度值还应考虑平均完成时间或者得分率。因此本算法还需进一步完善，以期适应更多不同的题库系统。

参考文献(References)

[1] 李俊杰,张建飞,胡杰,等.基于自适应题库的智能个性化语言学习平台的设计与应用[J].现代教育技术,2018,28(10):5-11.

[2] 王玥.自适应测验中题库的构建及其有效性检验[D].山东师范大学,2019.

[3] 刘宪爽,吴华明,肖文波,等.改进的双Sigmoid函数变步长自适应算法及在OCT中的应用[J].电子学报,2019,47(01):234-240.

[4] 张春霞.基于Matlab的自动组卷系统的设计与实现[D].内蒙古大学,2018.

[5] 叶勇,刘秀华,叶琰,等.基于LaTeX的题库管理与组卷系统设计[J].西南师范大学学报(自然科学版),2018,43(03):181-186.

作者简介：

莫怀训(1978-)，男，硕士，高级讲师.研究领域：电气自动化，计算机控制，智能教育技术.

刘晓瑞(1980-)，女，硕士，讲师.研究领域：前端设计技术，数据库，青少年计算机社区教育.