

# Adaptive Educational Testing System

摘要: 本文讨论了一个自适应教育测试系统的建立,分析了LOGISTIC模型在自适应测试中的应用和测试系统的体系结构,最后对自适应测试的实现进行了详细的说明。

关键词: 自适应测试系统 LOGISTIC模型 COM

## 自适应教育测试系统

赵若斌 (北京 中国科学技术大学研究生院 100039)

史忠植 (北京中国科学院计算机技术研究所 100080)

### 1 引言

考试系统在教育测试中的应用越来越普遍,但是目前我国开发的测试系统主要是用以下两种方式出题:一种是组卷考试系统,不论考生水平的高低都使用相同的试题进行测试,这就造成难题对于低水平的考生根本无法应答形同虚设,而容易的试题对于高水平的考生也起不到测试的作用,因此测试效果不理想;另一种系统出题采用随机选题的方式,显然这种方式出题也缺乏针对性,为了克服以上的不足,自适应教育测试软件成为了当前研究与发展的一个方向,在自适应测试过程中系统对考生的水平进行实时评估,根据考生的水平按一定规则选择具有较高分辨率的试题进行测试,这样不但测试更有针对性,提高了测试的效率,而且还能降低高水平考生疏忽或低水平考生猜测造成的测试误差,提高了测试的质量。已有国外测试机构开发出这类系统应用于实际的测试,例如:ETS的TOEF考试系统。下面就一个网上自适应测试系统的建立进行说明。

### 2 测试系统的分析与设计

#### 2.1 测试系统的功能分析

##### 2.1.1 系统的主要功能模块

管理模块:负责题库的管理与维护,测试系统参数的设置,用户资料的维护;

自适应测试模块:实施自适应测试;

数据存储模块:负责测试题,用户信息,测试信息的存储。

##### 2.1.2 自适应测试功能分析

自适应测试系统在测试过程中需要不断的评估考生的水平,根据考生的水平按照一定的规则选择测试题,因此科学的评估考生的水平是实施自适应测试的基础。在教育测量中的题目反应理论下,用不同难度的试题对考生进行测试,测试的结果可以相互比较,十分适合作为自适应测试系统的测试理论,因此本文应用该理论中的单参数LOGISTIC模型评估考生的水平。

(1) 测试模型的应用:单参数LOGISTIC模型适用于每一道题的答题结果只有对或错两种情况(不包含部分正确)的考试中。模型的表达式:  $P(\theta) = 1 / [1 + \exp\{-D(\theta - b)\}]$  (式1);其中  $D=1.7$  为常数;  $\theta$  是能力值(表示考生的水平);  $b$  表示试题难度;  $P(\theta)$  表示正确应答概率;应用该模型可以完成以下功能:

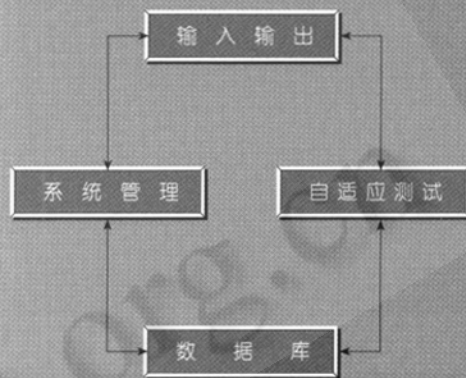


图1 系统的主要功能模块

① 建立能力值计算函数:根据试题的难度值  $b$  和考生的应答对错(用1,0表示),应用能力值计算函数可以求出表示考生水平的能力值  $\theta$ 。

② 建立测试误差计算函数:由考生能力值  $\theta$  和所做试题的难度值  $b$ ,应用误差计算函数可以求出测试的误差。

(2) 测试中选择试题的规则:应用单参数LOGISTIC模型,选择难度等于考生的能力值(即  $b = \theta$ )的试题,试题对考生有最大的分辨率,测试效果最好。

(3) 分数计算:在实际应用中一般把表示考生水平的能力值  $\theta$ ,通过线性变换转化为便于理解的考试分数。转换公式  $T = L\theta + K$  (式2);其中  $T$  表示考试分数,  $L$ ,  $K$  分别为常数,可根据需要设定,例如:TOEF考试系统中  $L=70$ ,  $K=500$ ,

(4) 本文自适应测试的基本步骤如下:

- ① 检查考生返回答案的对错;
- ② 调用能力值计算函数评估考生的水平;
- ③ 调用误差计算函数计算到目前为止的测试误差;
- ④ 判断是否达到测试的误差要求, 如果达到要求, 转到步骤 6);
- ⑤ 选择合适的试题提供给考生做进一步测试, 返回步骤 1);
- ⑥ 将能力值转换成分数输出。

## 2.2 测试系统的结构

测试系统采用三层 Web 结构: 第一层是用

表 1 试题表

列名	说明
itemID	试题号, 标识试题;
item	试题内容;
answer	试题答案;
difficulty	试题难度;

表 2 测试信息表

列名	说明
UserID	考生号, 标识考生;
ItemID	试题 ID, 记录考生所做的试题;
truefalse	答题情况, 答对赋值 1, 答错了赋值 0;

户层, 作为用户与系统的接口; 第二层是测试业务层, 实现自适应测试功能, 并在数据层和用户层之间传递数据; 第三层是数据层, 存储测试所需的试题和各种考试信息。应用三层结构使系统具有很大的灵活性和可扩展性。当不同的学科使用系统进行测试时, 主要工作是把相应的学科试题加入数据层的题库中; 如果要增加新的测试模型或升级测试业务, 主要对中间层做出变动, 对其他层影响不大; 此外在需要大量计算的业务层, 可以方便的增加设备数量或使用配置高的设备来提高系统的负载能力。因此这种结构便于系统的升级与维护。

## 3 系统的实现

### 3.1 测试系统实现的总体说明

(1) 客户层: 客户层应用 Web 浏览器, 用户与系统通过浏览器进行交互。用户通过浏览器输入用户名、密码、试题的答案以及新试题等信息。系统通过浏览器显示考试题目、考试成绩或管理员查询的试题等信息。

(2) 业务层: 业务层使用 Windows2000 上的 IIS5.0 Web 服务器。业务层有两个模块管理模块和自适应测试模块。管理模块负责试题信息的建立与维护, 用户信息的管理。自适应测试模块负责完成自适应测试工作, 包括在测试过程中检验答案、评估考生的水平, 选择合适的测试题目,

计算测试误差, 计算考试分数。这些功能用 ASP 编程实现。

(3) 数据层: 数据层用 SQL Server2000 为服务器, 这有利于与 Windows 2000 上的 IIS5.0 Web 服务器无缝连接。数据层存储测试所需的试题、用户信息以及一些测试信息。其中记录试题信息的试题表 ItemInfoT 和用来记录考生考试中所做的试题以及答题对错的测试信息表 TestInfoT 的字段如下。

试题表 ItemInfoT 包括以下字段 (见表 1):

测试信息表 TestInfoT 包括以下字段 (见表 2):

(4) 测试系统的体系结构图

### 3.2 自适应测试模块说明

(1) 能力值计算函数和误差计算函数在自适应测试中起着重要作用, 说明如下:

① 能力值计算函数 float CalculateCap (float fIniCap, float\*arrDiff, bool\*arrCorrect); 其中 fIniCap 为考生初始能力值; arrDiff 为所作试题的难度值数组的指针; arrCorrect 为应答结果数组的指针; 函数返回值是新的能力值。

② 误差计算函数 float Miscalculation (float fCap, float\*arrDiff); 其中 fCap 为考生能力值; arrDiff 为考生所作试题难度值数组的指针; 函数的返回值为测试的误差。

由于这两个函数应用频繁, 实现时将它们封装在一个 COM 组件 ModelCalculation 中, 这样可以加快系统的响应速度, 提高系统的性能, 还与测试模型相关的计算模块化, 便于系统的集成与维护。

(2) 为了方便系统的应用设立了以下几个 Application 变量: 变量 L、K 分别存储分数计算 (式 2) 中常量 L、K 的值; 变量 E 作为测试结束条件的测试误差值; 变量 M 保存最大测试长度, 控制测试的题目数; 此外还设置了以下几个 session 变量: 变量  $\theta$  保存测试过程中能力值计算函数求出的考生能力值, 用考生所做的第一道试题

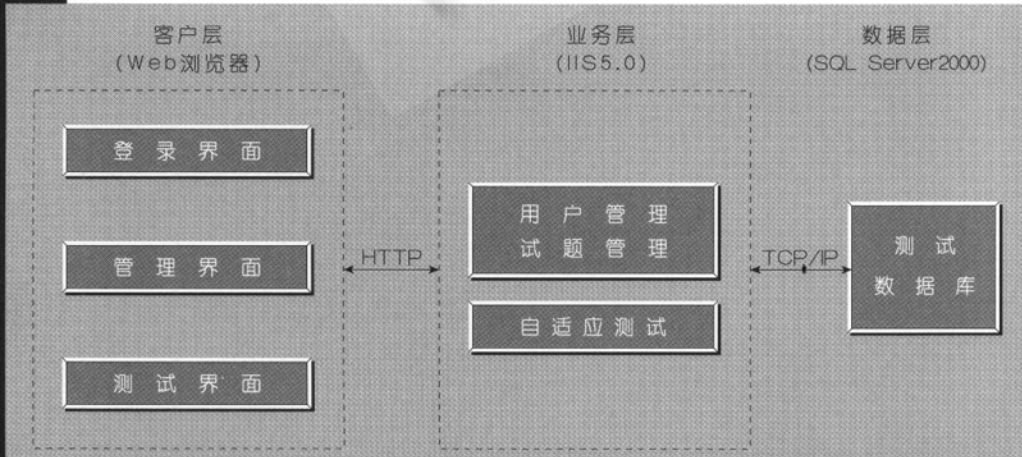


图 2 测试系统的体系结构图

的难度值给变量  $\theta$  赋初值; 变量 ItemID 保存考生当前正在做的试题的试题号。

(3) 当系统接收到考生提交的答案时, 首先检查答案的对错, 然后将考生号、试题号、应答应错写入测试信息表 TestInfoT 中。接着调用能力值计算函数, 分别从 TestInfoT 表和 TestInfoT 提取该生做的所有试题的难度值及相应的应答应错 (0 或 1), 以上一次估计的能力值 (存在 session 中) 为初值求出新的能力值, 更新 session 中的旧值, 然后以新的能力值以及所做试题的难度值为参数, 调用误差计算函数求出到现在为止的测试误差。如果达到测试所需的误差精度 E, 计算考生成绩结束测试, 否则选择难度值与考生能力值相等的试题继续测试。

(4) 特别指出在自适应测试中, 如果考生到目前为止做的试题不是有对有错, 不调用能力值计算函数估计考生的水平。如果考生应答应对, 按一定的步长增加下一道试题的难度值; 如果考生应答应全错, 按一定的步长减小下一道试题的难度值; 并且直接用试题的难度值更新考生的能力值, 直到考生所做的试题既有对的

又有错的, 再调用能力值计算函数估计考生的水平。如果达到测试做题数上限 M 时, 考生应答应还是全对或全错, 则给考生的能力值赋极值, 计算考生成绩结束测试。

(5) 为了避免考生长时间没有应答无法继续进行测试, 对所有的试题设定一个最长的应答时间限制, 如果在规定的时间内没有应答, 就认为答错并给出下一道试题。由服务器下载的 JavaScript 程序在客户端应用本地时钟控制试题应答时间, 这样可以减轻服务器端的负担, 增加测试系统的灵活性。

### 3.3 系统的安全控制

测试系统建立了系统本身的用户系统, 用户名称和密码存在于数据库中。用户只有输入正确用户名和密码才可以登录系统中。在 IIS 服务器上, 结合 WINDOWS 自身的用户安全机制给予管理员和考生 Web 服务器上目录不同的访问权限。连接 SQL 服务器时, 在 ASP 代码中只给出 SQL 的普通用户的代码对数据库通过 ODBC 的 DSN 进行连接, 这样可以减少对 WINDOWS 操作系统的干扰。

## 4 结束语

本文对应用单参数 LOGISTIC 模型建立自适应测试系统进行了讨论。详细说明了测试模型的应用, 测试系统的体系结构及自适应测试实现的关键问题。单参数 LOGISTIC 模型虽然是一种应用广泛的成熟模型, 但是它无法应用于多值记分等一些复杂的测试情况, 为了扩大测试系统的应用范围, 还需要对系统的测试模型进行扩充。 ■

### 参考文献

- 1 Betty A. Bergstrom, David L. Blits. An IRT Application. Computer Adaptive Technologies [DB/OL]. ETS Research Report [RR-01-011], 2001.
- 2 [美] Matt J. Crouch 著, 康博创作室译. 怎样用 ASP 和 COM 进行 Web 编程. 人民邮电出版社, 2001.
- 3 漆书清, 戴海崎, 丁树良. 现代教育与心理测量理论 [M]. 江西出版社, 1998.

