基于学生模型 *BOSM* 的智能组卷算法*

An Algorithm for Intelligent Test System Based on *BOSM*

冯志新 钟 诚 陈宁江
Feng Zhixin Zhong Cheng Chen Ningjiang

(广西大学计算机与信息工程学院 南宁 530004)
(College of Computer and Information Engineering, Guangxi University, Nanning, 530004)

摘要 为满足自适应性和个性化教学的要求，设计一个学生模型 *BOSM*，然后给出基于 *BOSM* 的智能组卷算法。算法根据知识点掌握程度值确定测试知识点权值以及难度系数等参数，实现根据测试知识点权值从题库中抽取满足要求的试题。试验表明该算法运行效果良好。

关键词 学生模型 掌握程度向量 自动组卷
中图法分类号 TP301.6 A

Abstract To meet the self-adaptability and personality, a student model named *BOSM* is developed. An algorithm for creating intelligently test paper based on *BOSM* is worked out. In terms of knowledge point proficiency, the algorithm can make a set of test questions for a quiz by computing the test parameters such as knowledge weight and difficulty coefficient, and extract topics. The experiment shows the algorithm performs well.

Key words student model, proficiency vector, intelligent test system

近年来，个性化的网上远程学习成为智能教学系统的一个研究热点。个性化网上学习模型的主要特点是实现针对每位学生的智能化、个别化教学，其中智能组卷系统是重要的组成部分，也是个难点。目前，大多数的智能组卷系统都是根据预先设定或人机对话的方式确定组卷要求，按一定的组卷策略从题库中抽取满足要求的试卷[1,2]。然而，这些传统的组卷算法并不能实现根据学生特定的学习情况抽取一份适合于每位学生的试卷。因此，为了适应智能化、个别化教学的要求，本文提出基于学生模型知识点掌握程度值的智能组卷算法。算法根据学生模型所反映的每位学生的知识点实际掌握程度，确定测试知识点权值以及难度系数范围等参数，然后以此为据从题库中抽取满足要求的试卷。

2002-06-16 收稿。
* 广西大学科研基金和国家高性能计算基金资助项目。

1　学生模型的设计

1.1　学生模型的概念

在计算机辅助教学系统中，我们把记录学生基本信息与学习状况的数据结构称为“学生模型”。根据学生知识表示方法的不同，目前常见的学生模型有3种[2]：

(1)覆盖模型，它把学生学习的知识描述成领域知识模块中专业知识的一个子集。

(2)偏差模型，它是通过把学生学习的错误概念，表示为领域专家知识的偏差而获得学生行为的学生模型。

(3)认知型模型，它是基于美国著名心理学家布卢姆的认知理论建立起来的学生模型，它是既能反映学生的知识水平，又能反映学生认知能力及心理因素的模型。

1.2　学生模型的建立

为了记录学生个性化信息以及对领域专家知识的掌握情况，建立了一个综合覆盖型和偏差型的学生模型 *BOSM*。在此模型中，通过建立知识点掌握情况表和学习进度表，可以视学生掌握的知识点为领域知识库的一个子集，由此分析学生当前学习状况、学习进度；通过建立知识错误概念集，即领域专家知识的一个偏差，可以记录学生学习过程中理解出错的知识点，并由此分析错误原因，提出针对性的学习建议。学生模型 *BOSM* 的结构及其与教学点播系统模块之间的关系如图1所示

BOSM 的定义为：$BOSM = \{SINFO, STUDYINFO, SPG, ERR\}$。

(1) 学生基本信息表 *SINFO*。该表给出学生的基本信息。$SINFO = \{sid, name, pwd, sex, age, hobby, sp\}$，其中 *sid* 为学生 *ID* 号，*name* 为学生姓名，*pwd* 为密码，*sex* 为性别，*age* 为年龄，*hobby* 为兴趣爱好，*sp* 为特长。

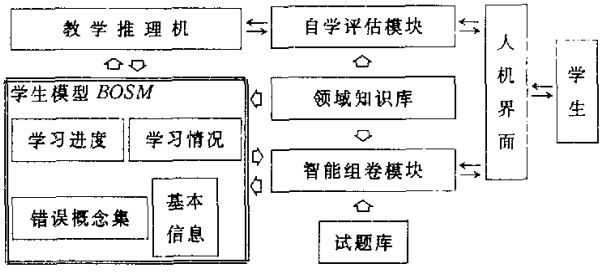

图1　教学点播系统及其学生模型

(2) 学习情况表 *STUDYINFO*。通过学习情况表，可以动态地描述学生对所学知识点的掌握情况。$STUDYINFO = \{KWD, KPV\}$。*KWD* 表示知识点，$KWD = \{sid, kid, sbid, target\}$，其中：*sid* 为学生 *ID* 号，*kid* 为知识点号，*sbid* 为学科号，*target* 为知识点的教学目标，教学目标即知识点要求达到的教学要求，可分为识记、理解、应用、分析、综合运用、评价[3] 等6个级别。*KPV* 表示各知识点对应的掌握程度向量，$KPV = \{sid\ kid\ lv1\ lv2\ lv3\ lv4\ lv5\ lv6\}$，其中 *sid* 为学生 *ID* 号，*kid* 为知识点号，*lv*1 ～ *lv*6 为 1 到 6 级知识点掌握级别的度量值。

为每个知识点设置一个知识点掌握程度向量 $KPV(x_1\ \ x_2\ \ x_3\ \ x_4\ \ x_5\ \ x_6)$，每个向量包括6个分量，分别对应知识认知领域的6级教学目标。各分量表示学生对该知识点的掌握水平处于各等级的概率，其值在0～1之间取值，所有分量值之和为1。向量表示的最小值为 *KPV* (1 0 0 0 0 0)，最大值为 *KPV* (0 0 0 0 0 1)。例如，向量 *KPV* (0.15 0.25 0.25 0.3 0.05 0)表示学

生对该知识点的掌握水平有15%的概率在第1级、25%的概率在第2级、25%的概率在第3级、30%的概率在第4级、5%的概率在第5级、在第6级的概率为0%。

(3)学习进度表 SPG。通过学习进度信息，系统可以分析学生的学习习惯，对学生进行教学推理以及给出进一步的学习建议。$SPG = \{sid, sbid, currkid, prekid, succkid\}$，其中 sid 为学生 ID 号，$sbid$ 为学科号，$currkid$ 为当前知识点号，$prekid$ 为前驱知识点号，表示本知识点之前学习的知识点，当本知识点为学习的起点时该值为空，$succkid$ 为后继知识点号，表示在本知识点之后即将学习的知识点，当本知识点为最后学习的知识点时该值为空。

(4) 错误概念表 ERR。错误概念表记录了学生自我测试过程中出错的知识点以及出错类型。保存这样的出错信息，系统可以分析学生学习的薄弱环节，以致给出有针对性的学习建议。$ERR = \{sid, sbid, kid, errc, errid\}$，其中 sid 为学生 ID 号，kid 为知识点号，$sbid$ 为学科号，$errc$ 为出错次数，$errid$ 为出错类别，取值 1 ～ 6，对应于试题的 6 级难度。

2 智能组卷算法设计

2.1 根据知识点掌握程度值确定知识点权值以及难度系数分布

为实现智能选取测试知识点的要求，组卷算法需要根据学生模型知识点掌握程度向量来设置知识点的权值。算法对综合掌握程度值低的知识点优先抽题，实现掌握能力差的知识点优先测试。学生模型 $BOSM$ 中，知识点的掌握程度用掌握程度向量 KPV 表示。不同的掌握程度应当代表不同的重要性，因而需要为每一个分量设置权值来反映之。组卷算法根据各向量分量值来计算学生对该知识点的综合掌握程度值，即将知识点的掌握程度向量与其权向量进行内积运算得到一个综合掌握程度值 c。一般地，教学目标级别越高，则说明对知识点的掌握要求也越高。因此权向量 P 各分量不妨设为其对应的索引，即 $P(1\ 2\ 3\ 4\ 5\ 6)$。这样综合掌握程度值 c 即为向量 K 与向量 P 的内积：$c = KPV \times P$。例如，设 $KPV = (0.15\ 0.25\ 0.25\ 0.3\ 0.05\ 0)$，则 $c = (0.15\ 0.25\ 0.25\ 0.3\ 0.05\ 0) \times (1\ 2\ 3\ 4\ 5\ 6)' = 2.85$。$c$ 即测试知识点权值。算法根据知识点权值从小到大的优先级依次抽取相应试题。

综合掌握程度值近似地反映了学生对知识点的理解程度，而向量的分量则较详细地说明了对知识点的掌握层次。因此，在智能组卷时，系统可根据向量各分量值来确定试题的难度系数。方法为：(1) 删除分量 $KPV_k \cdot x_6 = 1$ 的测试知识点，这样的知识点表示学生已经熟练掌握，无需再测试；(2) 确定测试该知识点的试题难度系数 D_k 的范围为 $i-1 \sim i+1$(其中，$KPV_k.x_i$ 值最大，i 取 1 ～ 6；$i = 1$ 时 D_k 为 1 ～ 3，$i = 6$ 时 D_k 为 4 ～ 6)，$KPV_k.x_i$ 分量值最大说明学生对该知识点的理解程度主要在第 i 级水平。

2.2 智能组卷算法实现

目前，许多组卷系统都是通过人机交互的方式由用户首先给出组卷要求，如测试知识点、题型、难度系统等，系统然后根据组卷要求产生组卷控制参数表[2]，并以此为基础，从试题库中自动抽取满足用户要求的试题。本文的组卷算法在组卷控制参数表的基础上，基于知识点优先的策略抽题。组卷控制参数包括题号、题型、知识点、难度系数、分值和完成标记。每道试题具有1条记录，每道试题的测试知识点、难度系数在组卷过程中确定，完成标记用于标记该题是否组卷完成(初值为0)，题量 n 由学生输入。组卷控制参数表在算法运行前通过预处理过程建立。

相应地，试题库中试题难度系数应划分为1～6级，而且题库中应有足够满足控制参数的试题。然而，算法在因满足要求的试题数量不足而失败时，应进行相应的参数放松，即将难度系数

范围放宽。在难度系数已经放宽的前提下，仍然组卷不成功，则说明试题库数量太少，这时应请示用户是继续组卷返回部分试题还是中断组卷以失败返回。算法具体步骤如下：

(1)如果测试知识点集 K 为空，则置 fail=1(初始为0)，退出。

(2)建立测试知识点链表 KL，置 KL 为 NULL。

①计算 k 的知识点综合掌握程度值 c_k，即 $c_k = KPV_k \times P_k (k \in K)$；

②按 c_k 升序将 k 插入线性链表 KL，置 KL[k].$k = k$，KL[k].$w = c_k$；从 K 中删除 k；

③ 如果 K 为空，则设置链表 KL 的表头 Head，转(3)；否则循环处理下一个知识点，转①；

(3)建立 KL 对应的难度系数范围二维数组 DA，置 DA 长度为 KL 元素个数，置 P = Head；

①根据 P 指向的知识点 $P.k$，查找其掌握程度向量 KPV_k；

② 如果 $KPV_k.x_i = 1$，则删除 KL 中该 k 节点；否则置 DA[k][0] = $i - 1$，DA[k][1] = $i + 1$($KPV_k.x_i$ 分量取值最大，其中，$i = 1$ 时 DA[k][0] = 1，DA[k][1] = 3；$i = 6$ 时 DA[k][0] = 4，DA[k][1] = 6)；

③P 指向链表下一节点；

④ 如果 P 指向 Null，则退出，转(4)；否则循环处理，转①；

(4)根据知识点链表 KL，循环从试题库中抽取试题匹配难度系数数组 DA。

①建立临时试卷表，初始化为空；建立临时试题库表，初始化为空；置试题计数 *count* = 1；

②设置知识点指针 PK =Head；建立知识点难度系数放松标记数组 lflag[n](n 为知识点数)；

③根据指针 PK 所指知识点 k 搜索试题库，查找与知识点 k 相匹配的试题，将这些试题存入临时试题库表，即知识点、题型、难度系数相匹配，并且没有与前次抽取试题重复，其中试题难度系数落在 DA[k][0]～DA[k][1]范围内即属匹配；

④若临时试题库为空则转⑧；否则调用随机函数，从中抽取一题添加到临时试卷表中，并置控制参数表第 *count* 条记录的知识点字段为 PK.k，难度系数为该试题实际难度系数，标记字段为 1；

⑤*count* ++；

⑥ PK 指向下一个知识节点；

⑦如果 PK 为 Null，则判断若 *count* 大于题量 n，则转 ⑨，若 *count* 小于等于 n，则重置知识点指针 PK=Head，转③；如果 PK 不为 NULL，则判断如果 *count* 大于题量 n，则转 ⑨，若 *count* 小于等于 n，则转 ③；

⑧如果放松标记数组 lflag[k] =1(初始为0)，则询问用户是否继续组卷，继续则转⑥，否则置 fail=1，转⑨；如果放松标记 lflag[k] =0，则置 lflag[k] =1，并放松难度系数范围，即 DA[k][0]－1和 DA[k][1]+1(特别地，当 DA[k][0]=1时，DA[k][0]不变；当 DA[k][1]=6时，DA[k][1]不变)，转③；

⑨退出。

算法(2)部分实现根据测试知识点建立测试知识点链表 KL，对每个知识点根据其掌握程度向量计算知识点综合掌握程度值，并作为知识点权值，按降序插入知识点链表 KL[4]。(3)部分实现根据链表 KL 计算对应知识点难度系数范围，即建立二维数组，保存知识点对应难度系数范围边界。(4)部分实现根据知识点优先的随机抽题算法，即根据 KL 节点的知识点，从试题

库中抽取满足难度系数等参数要求的试题，在试题多于一道时，随机抽取一道。算法结束后，若fail为1，则组卷失败，否则试题集保存在临时试卷表中。

3 结束语

与传统自动组卷算法不同，本文的算法把知识点的掌握程度值作为组卷的优先级，根据测试知识点掌握程度向量计算测试知识点权值以及确定测试知识点对应的难度系数范围，最后基于知识点循环抽取满足控制参数要求的试题。实验表明，在试题库设计合理、学生模型已建立的前提下，该算法运行稳定，效果良好。

参考文献

1 肖志辉，张祖荫，韩少杰. 智能出卷测试系统的设计与实现. 计算机工程与应用，2000，10：84～99.

2 林雪明，张均良，蒋伟钢. 基于知识点的试题组卷算法的建立. 微机发展，2001，2：77～79.

3 乐毓俊，刘占平，刘光然. 智能教学系统集成开发环境及其认知型学生模型的研究与实现. 宁夏大学学报（自然科学版），1996，16(4)：20～28.

4 苏德富，钟 诚. 计算机算法设计与分析. 北京：电子工业出版社，2001.

（责任编辑：蒋汉明）

（上接第255页）

表3 分词速度测试

文档	文档大小	识别词条数	分词时间	速度（词/秒）
D3	3.64	59	0.051	1157
D4	8.21	101	0.099	1020
D5	8.88	110	0.103	1068
D6	9.23	119	0.112	1062
D7	14.9	184	0.171	1076
D8	15.3	181	0.169	1071
D9	18.7	245	0.214	1144
总计	78.86	999	0.919	平均1087

3 结束语

当然，无词典分词法也有一定的局限性，会经常抽出一些共现频度高、但并不是词的常用字符串，如“这一”、“之一”以及“提供了”等等。在实际应用的统计分词系统中都要使用一部基本的分词词典（常用词词典）进行串匹配分词，即将字符串的词频统计和字符串匹配结合起来，既发挥匹配分词切分速度快、效率高的特点，又利用了无词典分词结合上下文识别生词、自动消除歧义的优点。

参考文献

1 Chien Lee-Feng. PAT-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval. Information Processing and Management，1999，35：501～521.

2 ZIPF H P. Human Behaviour and the Principle of Least Effort. Addison-wesley，Cambridge，Massachusetts，1949.

（责任编辑：邓大玉）