A Dynamic Tutoring Planning Approach for Intelligent Tutoring Systems Based on Fuzzy Computing

3Q−8

Haodong Wu Teiji Furugori
University of Electro-Communications

1 Introduction

A major problem to design intelligent tutoring systems (ITSs) is how to individualize the instruction for each learner. The material and style for teaching, the learner's role in on-line learning should depend not only on pedagogical knowledge and the learning history recorded in student model, but also on the learner's needs and demands. In this paper, we present a multiple layer decision- making model based on fuzzy computing to generate non-linear tutoring flow.

2 Decision-making Model

The tutoring planning in ITSs is normally divided into the categories "what to teach," "when to teach it," and "how to teach it." To pursue flexibility and efficieny, we divide the planning into four layers: topic decision level, unit decision level, presentation decision level and interactive decision level.

The topic decision level chooses a topic to teach next. The unit decision level determines which path will be taken to traverse the unit sub-network related to the topic selected. The presentation decision level deals with the selection, sequencing and synthesis of presentations that take forms of samples, graphics, simulations, tests and descriptions. The interactive decision level deals with how to lead a dialogue with the learner and correspond to his/her actions.

Decisions between adjacent levels are constrained mutually. The decision in the upper constrains the extent of decision in the lower, and the result or response in the lower will be passed to the upper to regulate the decision-making in the subsequent step.

We view the teaching material stored in knowledge base as a collection of units that include concepts, facts, procedures and rules, and then suppose that the domain knowledge is represented by semantic networks, where each node in a network represents a unit and each directed arc represents a relation between units. We introduce four types of relations as follows to describe logic inherent relationship or pedagogical linkage between concepts both in topic network and unit network.

Part-of relation (C_i, C_j) means that concept j is a sub-concept of concept i.

Prerequisite relation (C_i, C_j) defines that concept j is a prerequisite of concept i, which requires that concept j must be learned before concept i being selected. The prerequisite relation makes the selection obey the inherent order or human-determined order.

Easy-difficult relation (C_i, C_j) means that concept i is easier to learn or contains less material than concept j. It is utilized to implement some instructional strateges or tactics like "from easy to difficult," or "from deep to shallow".

Comparative relation (C_i, C_j) implies that the two concepts are similar or reverse in meaning. According to the comparative relation the tutor can generate a number of equivalent classes for comparative instruction.

There may be multiple relations between two concepts, and each relation has different strength to link the concepts in different pedagogical stratege. A tuple (fac_1 fac_2 fac_3 fac_4) is introduced to represent the weights of relations being set by pedagogical strateges, where the values of fac_1,fac_3 and fac_4 are within [0, 1] and the value of fac_2 is within [-1, 1]. When the value is negative, it means to take the reverse of the determined condition.

3 Unit Selection Algorithm

The procedure to select a topic is same as that to select a unit, so we only discuss unit selection algorithm here. The operators and data structures used in the algorithm are defined as follows:

Definition: $\bigvee, \bigwedge$ are operators. Suppose X and Y are fuzzy variables:

$$X \bigvee Y = max(X, Y),\ X \bigwedge Y = min(X, Y)$$

A Dynamic Tutoring Planning Approach for Intelligent Tutoring Systems Based on Fuzzy Computing
Haodong Wu, Teiji Furugori
University of Electro-Communications

$\bigvee$ and $\bigwedge$ satisfy relexivity, symmetricity and transitivity.

Definition: $\bigodot$ is a matrix operator, for $B(n \times k)$, $C(k \times p)$ set $D = B \bigodot C$, where

$d_{ij}=(b_{i1} \bigwedge c_{1j}) \bigvee (b_{i2} \bigwedge c_{2j}) \bigvee \cdots \bigvee (b_{ik} \bigwedge c_{kj})$

$(1 \le i \le n,\ 1 \le j \le p)$

Definition: A is a matrix to describe a network with m units. For any $a_{ij} \in A$ $(1 \le i, j \le m)$, $a_{ij}=1$ means that there is a relation (not a comparative relation) relating unit i to unit j, and 0 means the contrary.

Definition: A^* is transitive closure of A.

$$A^* = \bigvee_{i=1}^{k} \underbrace{(A \odot A \odot \cdots \odot A)}_{\text{i times of A}}$$

$$\text{and } \bigvee_{i=1}^{k+1} \underbrace{(A \odot A \odot \cdots \odot A)}_{\text{i times of A}} = \bigvee_{j=1}^{k} \underbrace{(A \odot A \odot \cdots \odot A)}_{\text{j times of A}}$$

Suppose $a'_{ij} \in A^*$ $(1 \le i, j \le m)$, $a'_{ij}=1$ means that there is a transitive relation relating unit i to unit j, and 0 means the contrary.

. Sel is a flag array to identify each unit in the network whether or not it has been selected. Sel[i]=0 means that unit i has been selected, 1 means contrary.

. At is an array with m elements, At[i] represents the number of unselected units depended by unit i.

At = Sel $\times$ A^*

Definition: Unit i is selectable iff At[i]=0 $(1 \le i \le m)$

. ctrl is a tag to identify the control mode, 1 represents system control mode and 0 represents learner control mode.

$Su = Sel \times [(fac_1\ fac_2\ fac_3\ fac_4) \cdot (N_1\ N_2\ N_3\ N_4)^{-1} \bigvee$
$0.5 \times ctrl \times G]$

In above formula, N_1,N_2,N_3,N_4 are connective matrixs (m×m) corresponding to four types of relations in the network and G is a matrix (m×m) which records a optimal path in history. Each value of Su is defined as selectability of unit.

Definition: Unit i is called an optimal selection, if for any unit j

$Su[i] \ge Su[j]$ $(1 \le i, j \le m)$.

Unit Select Algorithm

Step 1: Compute A^* and initialize crtl.

Step 2: Check the student model for searching all the unlearned concepts or concepts with low master degrees, if no such unit, exit the selection procedure, else initialize Sel.

Step 3: Inquire the learner whether or not he/she wants to select a unit. If the reply is yes, convert the input into the inner number SR.

Step 4: Compute At: At = Sel$\times A^*$ and storing all unit i where At[i] equals to 0 into queue LIST.

Step 5: Compute the selectability Su.

Step 6: Sort the units in LIST by their selectabilities in descending order, and store the sorting results into queue Goodlist.

Step 7: If SR is not null, it may be taken as the selection result and jump to step 9 in two cases: one is under system control mode and SR is in Goodlist; another is under learn control mode and At[SR]<2; otherwise SR will be refused.

Step 8: Fetch the first unit as the result and delete it from Goodlist.

Step 9: Find out the unit's number p and set Sel[p]=0.

The topic and unit selection algorithm have realized high non-linear planning for instruction. Because all concepts are stored and selected hierarchically, it satisfies the principle of instruction: "The best way to organize teaching material should be constructing a new concept based on the concepts learned, and describing it in a uniform description."

4 Conclusion

The approach we presented has several features different from other methods[1][2]. First, it satifies multiple targets or constriants such as pedagogical stratege, knowledge inner relatiopship, learner's ability and demands. Second, the algorithm reduces its time complexity by dividing the planning space into multiple layers of sub-space and using the principle of "local selecting" to constraint the planning space. Finally, the tutor can regulate the planning dynamically by modifying the values of factors such as G (the historical optimal learning path) and relation weight (fac_1 fac_2 fac_3 fac_4).

References

[1] T. Murry, B.P. Woolf, "A Knowledge Acquision Tool for Intelligent Computer Tutors", SIGART Bulletion, Vol.2, No.2, 1992.

[2] N.J. Rushby (editor), "Computer-Based Learning", Pergamon Infotech. Limited, 1983.