ADAPTIVE PERSONALIZED E – LEARNING USING WEIGHTED FUZZY & RECOMMENDATION

Mr. B. Karthikeyan 1
Assistant Professor
Department of Computer Science
Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore, Tamil Nadu, India

Mr. N. Kamalraj 2
Head & Associate Professor
Department of Information Technology
Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore, Tamil Nadu, India

Abstract:
Fuzzy logic has a number of properties that make it suitable for describing e learning events: (i) it can tolerate the unreliable and imprecise datasets; (ii) it is much closer to the way of thinking than crisp logic. (iii) Compared to other classification algorithms based on probability theory, fuzzy logic is much more intuitive and easier to use. A disadvantage of using fuzzy logic is that storing the rule base might require a significant amount of storage. The number of rules grows exponentially to the number of variables. With n variables each of which can take m values, the number of rules in the rule-base is mn.

Introduction
In this chapter, it is proposed to mine e learning data using personalization techniques to extract useful patterns and support to the learners. Fuzzy logic is used with knowledge definer system for effective personalization of rules from quantitative datasets. The proposed method uses synthetic dataset showed in chapter 4 consisting of 10 numbers of domains and learner details.

Weighted Fuzzy Intelligent Computing in Knowledge Learning System
(WFR_IEM_MCF) phase of a web-based educational application is created for individualized instruction on the computer science domains.

The proposed system effectively identifies the individual students skills based on their objective questions, practical results, and the time based evaluations.

The time analysis is one of the factor included to define the students proficiency in each domain.

WFR_IEM_MCF is the basis for providing personalized tutoring, which identifies and updates the student’s knowledge level.

The WFR_IEM_MCF have the ability to recognize the alterations of the student’s learning states and dynamically adapt the presentation of the learning material accordingly.

Advantages of the proposed system:

1. This approach improves the efficiency of the adaptively of the instructional process.
2. WFR_IEM_MCF module, which identifies the alterations on the state of students’ knowledge level.
3. Student can read the learning material more times, until they have learned the domain knowledge Fully. The goal of this proposal was to personalize the e-learning procedure using rule based fuzzy sets. In order to enhance individual proficiency and adaptive learning process, WFR_IEM_MCF has been proposed.

It combined the user skill state and the domain concepts to promote personalization in educational applications. It sequentially verifies the student’s knowledge and concept depth from various events with their personal choices. The process has divided into three steps known as domain knowledge; proficiency evaluation and course prediction based on the skill state and concept matching. Initially the chapter discusses about fuzzy logic implementation in e learning.
Weighted Fuzzy Rule Implementation:

Weighted Fuzzy Rule (FL) is a multi valued paradigm which allows intermediate values to be defined between conventional evaluations like true/false, yes/no, pass/fail, etc. Notions like rather tall or very fast can be formulated mathematically and processed by computers, in order to apply a more human-like way of thinking in the programming of computers. The Fuzzy systems are an alternative to traditional notions of set membership and logic on e learning.

The simplest fuzzy model consists of a set of rules with an “if – then” structure:

If < condition 1 > and … and > condition n >
then < conclusion >

Where, condition i is a statement of type “Zi is Yij”. In this statement Zi represents the actual value of some i-th real world variable meanwhile Yij is a flexible predicate naming the j-th linguistic term of the corresponding i-th Linguistic Variable. Yij given by a fuzzy set which represents the use of the flexible predicate on the domain of Zi. Statements of this kind are called “premises”. The conclusion is also a fuzzy set, which represents the linguistic term expressing a flexible predicate, which characterizes the output behaviour of the system if all conditions are satisfied. Notice that “if – then” rules may be used both to model the state of a system (see Rule 1 and ambiguous. Hence, membership of an element from the universe in this set is measured by a function that attempts to describe vagueness and ambiguity.

A fuzzy set, then, is a set containing elements that have varying degrees of membership in the set. This idea is in contrast with classical, or crisp, sets because members of a crisp set would not be members unless their membership was full, or complete, in that set (i.e., their membership is assigned a value of 1. Elements in a fuzzy set, because their membership need not be complete, can also be members of other fuzzy sets on the same universe. Elements of a fuzzy set are mapped to a universe of membership values using a function-theoretic form. This function maps elements of a fuzzy set A~ to a real numbered value on the interval 0 to 1.

When a simple database T with two attributes (X1 and X2 as level 1 and level 2 respectively) and (three below) and to take a decision to control the system (see Rule 2).

Rule 1: If knowledge level (KL) of (level1) is high and the KL(level2) is high and KL( level 3) is low then the user need to concentrate on level 3.

Rule 2: If knowledge level (KL) of (level1) is medium and KL(level2) is low. Fuzzy control is then closely related to fuzzy decision making. Take as example the following set of “if – then” rules constituting a fuzzy control model for an extreme simple e learning system for a particular learner:

R1 : If knowledge level (KL) of (level1) is low and the KL (level2) is low and KL (level 3) is low then the user need to concentrate on level 1.

R2 : If knowledge level (KL) of (level1) is medium and the KL (level2) is high and KL (level 3) is low then the user need to concentrate on level 1.

R3 : If knowledge level (KL) of (level1) is high and the KL (level2) is high and KL (level 3) is high then the user need to go on level 4.

In classical or crisp, sets the transition for an element in the universe between membership and non-membership in a given set is abrupt and well-defined. For an element in a universe that contains fuzzy sets, this transition can be gradual. This transition among various degrees of membership can be thought of as conforming to the fact that the boundaries of the fuzzy sets are vague linguistic terms and associated MFs (see Figure 3.1) is considered.

![Figure 3.1 Attributes and linguistic terms for the attributes X1 and X2](Image)

As per the E-learning dataset, the above attributes and linguistic terms are defined as the value of individual scores. If the value for X1 is defined as 5, then the score will be modified into the linguistic terms (High, medium, Low). This is calculated from the total questions and time taken for every question.

If total question is 10 → the score is 8 → the time take for each question is 2 ms approximately → then the linguistic score will be converted as “High”.
3.3 WFR_IEM_MCFProcess:

A. Domain Knowledge Representation using Fuzzy

Domain concepts are the learning material, and directed arcs, which represent relations between the concepts of the learning material. The relations that exist between the concepts of the learning material depict so the order in which the domain concepts have to be delivered and the structure of the learning material, as the knowledge dependencies. In particular, there are three types of relations between the concepts: “precedes” that declares the order in which each domain concept of the learning material has to be taught. A domain concept is affected regarding the knowledge level of its related domain concepts. In other words, they depict the “strength of impact” of a domain concept on a related concept. The particular numbers are only positive [14].

**Fig 3.2 concept similarity score calculation in every domain**

The above Figure 3.2 represents the concept similarities between different programming languages. Using this, the learner can choose the language for further study. This is happened due to the fact that the increase of the knowledge level of a domain concept leads to the increase of the knowledge level of a depended domain concept, and the decrease of the knowledge level of a domain concept leads to the decrease of the knowledge level of a depended domain concept. Therefore, the numbers of the directed arcs that depict the knowledge dependencies belong to the interval (0, 1).

One important module of the WFR_IEM_MCF is the domain module, which contains a description of the knowledge or behaviors that represent expertise in the subject-matter domain of the system. To enable communication between system and learner at a content level, the domain model of the system has to be adequate with respect to inferences and relations of domain entities and compatible.

**Fig. 3.3. Concept Hierarchy**

The hierarchy of the domain concepts, as well as the dependence relations among them must be represented. Therefore, the domain knowledge of the presented system is organized into a hierarchical structure in combination with FCMs.

**Fig. 3.3.1 domain for level 1 Hierarchy**

The hierarchical structure shown in Fig 3.3 and 3.3.1 depicts the difficulty level of the domain topics and the order in which each topic must be taught.
Students Knowledge in every domain

Knowledge representation in E-learning

In other words, the knowledge domain module is responsible for the representation of the subject matter taking into account the course modules, which absorb domain concepts. The meticulous module has been introduced in ITS but its use has been extended to most current educational software applications that aim to be adaptive and/or personalized [13]. To enable communication between system and learner at content level, the domain model of the system has to be adequate with respect to inferences and relations of domain entities with the mental domain of a human expert. This technique makes easier the assortment of the appropriate educational material satisfying the student’s learning needs.

Domain of the Programming Language

This model is used for Learners of programming languages have different backgrounds and their knowledge of a concept of the programming language, which they are taught, is subject to change. A new concept may be completely unknown to the learner but in other circumstances it may be partly or completely known due to previous related knowledge of the learner. For example, if a learner already knows an algorithm (e.g., calculating the sum of integers in a ‘for’ loop), there is no need to learn another similar algorithm (e.g., counting in a ‘for’ loop). Similarly, if a learner knows programming structure (e.g., one-dimensional arrays), it is easier to understand another programming structure (e.g., multidimensional arrays), so this new structure should not be considered as being completely unknown to the learner.

Weighted Fuzzy intelligence computing for automatically learning Process of a Student

The Weighted Fuzzy intelligence computing system is used to model the cognitive states of learners of the programming language. This system for modeling automatically the learning or forgetting...
process of a student is presented. Fuzzy sets are finding very wide applicability in Weighted Fuzzy intelligence computing systems (WFICs). The operation of WFICs is based on rules. The rules are expressed as a collection of IF-THEN statements. The Weighted Fuzzy intelligence computing module is responsible for identifying and updating the student’s knowledge level of all the concepts of the knowledge domain.

### Initial process

**Step 1:** Read the pattern and its target value.  
**Step 2:** Create Fuzzy membership function.  
**Step 3:** Create clustering of domain levels using scores.  
**Step 4:** Process with target values.  
**Step 5:** Obtain final weights.

### Proposed Algorithm

**Initial process**

**Step 1:** Read the data D  
**Step 2:** For each attribute Ai do linguistic function L and make fuzzy set Fs.  
**Step 3:** Create Weighted Fuzzy membership function. \( M_F(Score) = \{\text{High, Medium, Low}\} \)  
Support: elements having non-zero degree of membership.  
Core: set with elements having degree of 1  
-Cut: set with elements with degree \( \geq \alpha \).  
Height: maximum degree of membership.  
\( y_1 = M_F(Score)(x, [2379]) \);

**Step 4:** Create clustering of domain levels using scores.

1. Let \( n \) is the total number of clusters; \( n_1 \) is the length of D.  
2. Define Centroid C from lowest and highest set  
   a. \( C = (d_{\text{min}}(i), d_{\text{max}}(j)) \)  
3. Calculate distance \( y_1 = \text{dist}(D, C.get()); \)  
4. D.cluster(cluster);  
5. Return cluster
**Step 5: formulate rules**
Select rules according to the WFM

**Step 6: de-fuzzification**

**Step 7: Testing process and return KL**

**Testing process**

**Step 1:** Input a pattern.

**Step 2:** Process with Fuzzy membership function.

**Step 3:** Find the cluster to which the pattern belongs.

<table>
<thead>
<tr>
<th>Dataset information’s</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of programming Languages selected</td>
<td>4</td>
</tr>
<tr>
<td>Total number of learners</td>
<td>100</td>
</tr>
<tr>
<td>Total number of domains used</td>
<td>28</td>
</tr>
<tr>
<td>Stereo types</td>
<td>8</td>
</tr>
<tr>
<td>Types of knowledge level</td>
<td>4</td>
</tr>
</tbody>
</table>

**Step 4:** Obtain estimated target values.
**Step 5:** Classify the status (KL)

The proposed algorithm takes less number of iterations to reach a stable state. The time taken by this algorithm is in between the time taken by the existing fuzKSD and time taken by ITS.

**IMPLEMENTATION AND RESULT**

This section describes the implementation process. Implementation is the realization of an application, or execution of plan, idea, model, design of a research. This section explains the software, datasets and modules which are used to develop the research.

**DATASETS**

The system used a synthetic dataset, which can be any number of e learners created by the proposed website. Data collection is the first step of our proposed system, the data authenticity and integrality will directly affect the following works smoothly carrying on and the final recommendation of characteristic’s quality. Therefore it must use scientific, reasonable and advanced technology to gather various data.

At present, towards e learning technology, the main data origin has three kinds: server data, client data and middle data. The first step in the Web usage mining process consists of collecting the relevant learning data, which will be analyzed to provide useful information about the learners’ proficiency.

**Table 4.2 dataset descriptions**

The above table describes the dataset description used for the experiments. Where the dataset contains calculated result is multiplied by a special factor that determines the difficulty level of the current topic. This factor can be calculated by referring to the results achieved by all the learners that studied this topic before. The better the achieved results the higher the difficulty factor value is. This process is illustrated through the following formula:

\[
E_T = \frac{\sum_{p \in P_T} (W_p \times R_p)}{\sum_{p \in P_T} W_p} \times D_T
\]

Where \(E_T\) is the expected result of topic \(T\), \(P_T\) is the set of all the prerequisites of topic \(T\), \(W_p\) is the relevance importance value of topic \(P\), \(R_p\) is the mastering level of topic \(P\) and \(D_T\) is the difficulty factor of topic \(T\). The skill analyzer module has to compare the expected result \(E_T\) to the achieved result \(R\) and modify the plan accordingly.
8 number of stereotype learners which are described below.

<table>
<thead>
<tr>
<th>Stereotype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stereo1</td>
<td>Novice learners;</td>
</tr>
<tr>
<td>Stereo2</td>
<td>learners that know about the foundations of programming (like variables and constants, operands, expressions);</td>
</tr>
<tr>
<td>Stereo3</td>
<td>learners that know about the selection statement (if, if-else if, nested if);</td>
</tr>
<tr>
<td>Stereo4</td>
<td>learners that know about how to built programs using the “for” statement;</td>
</tr>
<tr>
<td>Stereo5</td>
<td>learners that know about the iteration statements “while” and “do..while;”</td>
</tr>
<tr>
<td>Stereo6</td>
<td>multidimensional arrays;</td>
</tr>
<tr>
<td>Stereo7</td>
<td>Expert learners (they know, also, about the process of sub programming).</td>
</tr>
</tbody>
</table>

**Table 4.3 stereotype**

The following four fuzzy sets for representing students’ knowledge level of a domain concept are defined.

<table>
<thead>
<tr>
<th>Knowledge Levels</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown (Un)</td>
<td>the degree of success is from 0–60%;</td>
</tr>
<tr>
<td>Insufficiently Known (InK)</td>
<td>the degree of success is from 55–75%;</td>
</tr>
<tr>
<td>Known (K)</td>
<td>the degree of success is from 70–90%;</td>
</tr>
<tr>
<td>Learned (L)</td>
<td>the degree of success is from 85–100%.</td>
</tr>
</tbody>
</table>

**Table 4.4 knowledge levels**

From the above set of learners, 12 learners are based on stereotype1, 15 are stereotype2, 23 are stereotype3, 23, 27 are stereotype4 respectively. The details are shown in the following table. The table has been generated from the above dataset D.

<table>
<thead>
<tr>
<th>Type</th>
<th>Learners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stereo1</td>
<td>12</td>
</tr>
<tr>
<td>Stereo2</td>
<td>15</td>
</tr>
<tr>
<td>Stereo3</td>
<td>23</td>
</tr>
<tr>
<td>Stereo4</td>
<td>4</td>
</tr>
<tr>
<td>Stereo5</td>
<td>6</td>
</tr>
<tr>
<td>Stereo6</td>
<td>20</td>
</tr>
<tr>
<td>Stereo7</td>
<td>18</td>
</tr>
<tr>
<td>Stereo8</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table 4.5 Students count in different stereotype**

**Users count in different stereotype Type**

The above chart shows, the stereotype 3 has more number of learners that others, finally the stereotype 8 has only 4 learners in the domain.

The next process is analyzing the learner knowledge type. From the 100 learners, 31% learners are under knowledge level 1, 38% are under knowledge level 2, 29% under level3, 2% are level 4 respectively. The details are shown in the following table. The table has been generated from the above dataset D.

<table>
<thead>
<tr>
<th>Type</th>
<th>Students (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown(UK)</td>
<td>31</td>
</tr>
<tr>
<td>Insufficiently Known (InK)</td>
<td>38</td>
</tr>
<tr>
<td>Known(K)</td>
<td>29</td>
</tr>
<tr>
<td>Learned (L)</td>
<td>2</td>
</tr>
</tbody>
</table>

**Table 4.5.1 Students count in different knowledge level**
Students in different Knowledge level

![Students in different Knowledge level](image)

**Fig 4.2 chart for total learners percentage in different knowledge levels**

The above chart shows, the level 2 has more number of learners that others, finally the level 4 ie K4 has only 2% learners in the domain.

**Step 1: Learner Level**

Researchers in the e-learning community have always considered it important to develop a model of the novice learner. The adaptive and collaborative capabilities of the system are mainly based on this model. Along with many characteristics that define the learner model there are characteristics that have a great influence on the learning process such are:

- **Prior knowledge of the learner** that determines the concepts that he can learn.
- **Learner’s skills and learning styles** that determines the teaching approach that best suits the learner.
- **Learner’s emotions and motivations** is another important characteristic that is usually forgotten in learner modelling research.

Thus, the system proposes a multi-level comprehensive learner model that is based on the Novice learner Information Package (NIP). The proposed novice learner model consists of three levels as follows.

1. **Personality level**: This level consists of information related to the learner type and any extensions to it, along with information about the learning style and skills of the novice learner.
2. **Behaviour Level**: This level is responsible for recording information about learner activities in the system. For example, test and time taken for each test has been analyzed.

- **Knowledge Level**: This level includes information about novice learner current knowledge as opposed to a Knowledge sharing systems.

**Step 2: Domain Knowledge Model**

Knowledge of the domain can be seen as interconnection of two networks which are the network of concepts known as knowledge space and the network of educational material. Knowledge space is a structured representation of the domain knowledge, in our proposal the score has been used for knowledge representation. It consists of interconnected nodes where each node represents a small piece of knowledge (i.e. a concept).

**Course Model**

The adaptation capability of the proposed system highly depends on the past successful course plans stored in the database along with the scores. The course model is a structured case-based model, where each case represents an experience of course delivery.

Every learner can view their domain knowledge on each domain. Finally, the proposed system will provide appropriate tutorials and notes to improve their knowledge.

**Step 3: Evaluation model**

Each training state was designed with specific issues in mind. The evaluation of past training cases includes calculating the similarity measures between the state of current learner and the previous training cases. This process should consider all the issues that affect the construction of the previous training state. These issues include the training objectives, learner’s knowledge mastering levels, his skills appraisal levels, his learning style along with any possible effective environmental issues. Moreover, looking at past learners’ achieved results may be helpful in choosing the best previous training experience.

It is worth noting that each of the above issues has e-learning own relative impact on the training experience. The relative importance of each issue can be set by the system administrator.

The problem is how to evaluate previous training cases in order to choose the training state that has the highest degree of similarity to novice learner. First, let us suppose that $E = \{e_1, e_2, \ldots, e_n\}$ is a set of n-effective issues. Second, assume that $L = \{l_1, l_2, \ldots, l_k\}$ is a set of k-previous learning cases, while \( l \) by e-learning will be used to indicate the current learning state. Moreover, the similarity between the
effect of $e_l$ on the current learning state and e-
learning effect on previous learning state $l_p$ is $S(l, l_p)$.

$R=\{r_1, r_2, ..., r_n\}$ is a set of relative importance values corresponding to different effective issues in $E$; so that $r_j$ is the relative importance of issue $e_j$; where $(r_j < 1)$ and.

Based on the above assumptions, the following formula can be used to evaluate the similarity between the current learning state $l$ and previous learning state $l_p$.

$$S(l, l_p) = \sum_{i=1}^{n} r_i \cdot S_i(l, l_p)$$

\[ \sum_{j=1}^{n} r_j = 1 \]

Definition 1: \(N(A)\) represents the number of elements of the set $A$.

Next, we will shed lights on some practical ways to calculate $S_l(l, l_p)$. Putting in mind that $S_l(l, l_p) = 1$ which implies that $S(l, l_p) \leq 1$. First, recall that the cornerstone in building a course is the course objectives. So, we will start by proposing a way to calculate $S_o(l, l_p)$ which is the similarity between objectives of novice learner $l$ and objectives in the learning state $l_p$. To do so, assume that the set of objectives required by the current learning state is $O=\{o_1, o_2, ..., o_m\}$ and the set of objectives in the learning state $l_p$ is $O_p=\{o_{p1}, o_{p2}, ..., o_{pf}\}$ then:

\[ N(O) \]

Assuming that every novice learner has at least one learning objective, then we can derive that $S_o(l, l_p) \leq 1$. Next, $S_o(l, l_p)$ can be substituted in formula 1 for $S_h(l, l_p)$ where $h \in \{1, ..., n\}$ and $e_h$ is the label for the objectives issue which is a element in $E$.

$$S_l(l, l_p) = \begin{cases} 1 & \text{if } N(T) = N(T_p) = 0 \\ \frac{1}{N(T)} N(T_p) & \text{if } N(T) = 0 \text{ & } N(T_p) \neq 0 \\ \frac{N(T \cap T_p)}{N(T)} & \text{otherwise} \end{cases}$$

Note that if the current learner needs no topics to be added to his training plan (i.e. $N(T)= 0$) then the most suitable previous plan could be the least tough (i.e. with the least number of added topics); meaning that $1/N(T_p)$ will yield smaller values if $N(T_p)$ is large. In the same way, $S_l(l, l_p)$ can be substituted in formula 1.

Third step is, referring to the four dimensions of Felder’s learning styles model, we can define $f_{ld} \in \{-1, 0, +1\}$; Where $F_{ld}$ is the learning style of novice learner $l$ in dimension $d$ which can take one of three values. Similarly, $f_{pd} \in \{-1, 0, +1\}$ is the learning style of novice learner $l_p$ in dimension $d$.

Second steps is, let us suppose that the set of topics needed to be added to the course plan due to lack in learner $l$ background knowledge or skills is $T = \{t_1, t_2, ..., t_m\}$ and the set of topics in the learning state $l_p$ that has no direct links to the objectives of that course (rather, it was added to compensate a lack of learner’s background knowledge or skills) is $T_p =\{t_{p1}, t_{p2}, ..., t_{pf}\}$. Then:

$$F_{ld} \in \{-1, 0, +1\}$$

Note that if the current learner needs no topics to be added to his training plan (i.e. $N(T)= 0$) then the most suitable previous plan could be the least tough (i.e. with the least number of added topics); meaning that $1/N(T_p)$ will yield smaller values if $N(T_p)$ is large. In the same way, $S_l(l, l_p)$ can be substituted in formula 1.
References:
15. Tung-Cheng Hsieh et al, “Designing and implementing a personalized remedial learning system for enhancing the programming learning”.