# Contents

1 Introduction
   1.1 Integration of Adaptive Recommendation System . . . . . . 3
   1.2 Adaptive Hypermedia . . . . . . . . . . . . . . . . . . . . 4
      1.2.1 What can be adapted? . . . . . . . . . . . . . . . . 5
      1.2.2 How to represent knowledge? . . . . . . . . . . . . . 5
   1.3 Adaptive System Components . . . . . . . . . . . . . . . . 6
   1.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2 Learner Modeling - A Literature Survey 8
   2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   2.2 Role of User Model in Intelligent Tutoring Systems (ITS) . 8
   2.3 Information stored in User Model . . . . . . . . . . . . . . 10
   2.4 Techniques for user modeling . . . . . . . . . . . . . . . . 11
      2.4.1 Bayesian Belief Network . . . . . . . . . . . . . . . . 11
      2.4.2 Machine Learning Methods . . . . . . . . . . . . . . 11
      2.4.3 Fuzzy Logic based technique . . . . . . . . . . . . . 12
      2.4.4 Neural network based technique . . . . . . . . . . . 13
      2.4.5 Fuzzy clustering based techniques . . . . . . . . . . . 13
      2.4.6 Neuro-Fuzzy techniques . . . . . . . . . . . . . . . 14
   2.5 Comparative analysis of UM techniques for different appli-
      cations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
   2.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

3 Knowledge Representation - A Literature Survey 17
   3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . 17
   3.2 Literature survey on domain knowledge representation . . . 17
      3.2.1 Initially proposed system for knowledge representation
         (novice representation) . . . . . . . . . . . . . . . . 18
      3.2.2 Goal-oriented connectionist knowledge representation 18
      3.2.3 Hierarchy based knowledge representation . . . . . . . 20
      3.2.4 Fuzzy Cognitive Maps (FCM) . . . . . . . . . . . . . 21
   3.3 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Proposed System for Giving Personalized Feedbacks</td>
<td>24</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>24</td>
</tr>
<tr>
<td>4.2</td>
<td>4-layered architecture of the system</td>
<td>24</td>
</tr>
<tr>
<td>4.3</td>
<td>Knowledge Representation</td>
<td>25</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Example</td>
<td>26</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Rules for Recommendation</td>
<td>26</td>
</tr>
<tr>
<td>4.4</td>
<td>Modeling Learner</td>
<td>28</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Example Scenario</td>
<td>30</td>
</tr>
<tr>
<td>4.5</td>
<td>Low Level System Architecture</td>
<td>31</td>
</tr>
<tr>
<td>4.5.1</td>
<td>System Initialization</td>
<td>31</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Update the knowledge levels of related concepts</td>
<td>31</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Prioritization of Concepts</td>
<td>32</td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>33</td>
</tr>
<tr>
<td>5</td>
<td>Implementation</td>
<td>34</td>
</tr>
<tr>
<td>5.1</td>
<td>User interfaces</td>
<td>34</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Login page</td>
<td>34</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Learner’s page</td>
<td>34</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Login page</td>
<td>34</td>
</tr>
<tr>
<td>5.2</td>
<td>Use-case diagram</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion and Future Work</td>
<td>37</td>
</tr>
<tr>
<td>6.1</td>
<td>Conclusion</td>
<td>37</td>
</tr>
<tr>
<td>6.2</td>
<td>Looking ahead</td>
<td>37</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Learning workflow in MOOC system ........................................... 3
1.2 Personalized feedbacks in MOOC system ............................... 4
1.3 Methods and Techniques for adaptive hypermedia .................. 5
1.4 Components of an Adaptive System ........................................ 6

2.1 Modules in ITS [1] ................................................................. 9
2.2 Classification of user models ................................................ 10
2.3 Information stored in user model .......................................... 10

3.1 A three layered connectionist architecture for domain representation [2] .......................................................... 19
3.2 part of the concept hierarchy of Computer Organization and Architecture course [3] ...................................................... 20
3.3 Membership Functions for fuzzy sets of unknown (CU), known(CK), and learned (CL) [4] ................................................... 22

4.1 input data to the system ......................................................... 25
4.2 4-layered architecture of the system ...................................... 25
4.3 State diagram of objectives .................................................. 27
4.4 Relationships among objectives .......................................... 27
4.5 Domain model in FCM ......................................................... 28
4.6 knowledge representation ..................................................... 30
4.7 Initialization of concept nodes .............................................. 32

5.1 Login page ................................................................. 35
5.2 Learner’s page ............................................................... 35
5.3 Use case Diagram .......................................................... 36
Abstract

MOOC (Massive Open Online Course) provides a new way of learning, which is open, participatory, distributed and lifelong. Various premier universities of the world are now offering courses in the form of MOOC. Providing personalized feedbacks and recommendations to the participants is a challenge, as the number of participants is very large, which makes it difficult to monitor each participant. Presently, recommendations and feedbacks are static and do not match with the participants’ needs. There is a need of an automatic system, which can give personalized feedbacks and recommendations to the participants, as per their needs. This work is an attempt to integrate an adaptive recommendation system with MOOC, which can adapt the needs of the participants/learners and provide recommendations which are personalized to them.

In this work, we proposed an improvised adaptive system, in which adaptation is achieved by modeling the learners/participants, as per their knowledge levels. Modeling of learners also requires to represent the domain in a specific way, we used a combination of hierarchical and network based approaches to represent the domain. A basic programming course is considered for demonstration of the system.
Acknowledgment

I would like to express my deep gratitude to Prof. D.B. Phatak, who has always been making things simple to understand. Without his deep insight into this domain and his valuable time for this project, it would not have been possible for me to move ahead properly. I would also like to thank Mr. Nagesh Karmali and Mr. Bhairavnath for their support and guidance. They have been remarkable in their attempt to keep me motivated in this project and has always tried to improve me with proper feedback. I would like to thank Mr. Raj Agrawal for reviewing my report and giving me valuable suggestions.
Chapter 1

Introduction

Internet enabled learning gives a freedom of an effective navigation across the learning domain [5]. As compared to a conventional classroom, where students have the limitation on the number of times to listen a lecture finitely, Internet based learning gives the infinite access to the learning material with better navigation. Apart from better navigation and infinite access to learning material, there is a great opportunity of research to enhance Internet based learning. In the developing countries e.g. India, the magnificent increase in the number of engineering students is the aftermath of increased number of engineering institutes [6]. As a result of this increase in a short period of time, there is a lack of efficient infrastructure and very low faculty to student ratio [6]. Various premier institutes e.g. Indian Institute of Technology Bombay, have started various projects like Ekalavya for training faculties and students across the country [7].

MOOC (Massive Open Online Courses) is an emerging area of research which potentially can bridge the gap between good infrastructure and students. MOOC is an online course targeted for a very large scale participation of students and faculties. MOOC also provides an interactive discussion forum for interaction among participants. The number of participants in a course offered by a MOOC system, may vary from few thousands to millions. Unlike a conventional class-room, where number of students is very small, it is easy to monitor the individual performance, and give a personalized feedback to the students, lectures at such large scale, where the number of participants is diverse in terms of knowledge levels, culture, learning styles etc., it is very difficult to monitor each student and give feedback which is personalized to an individual.

Figure 1.1 shows the typical learning workflow in the MOOC environment. The course material consist of lecture videos, exercises, assignments etc., which is divided weekly (generally). At the start of the week, all the study material for the current week opens for the learner. Learner watch video lectures, attempts exercises, discuss in discussion forum provided, com-
plete assignments etc., after the deadline gets over, assignments of the learners undergo evaluation procedure. Evaluation procedure is mostly peer evaluation. The same procedure is repeated for the next week till the course is not completed.

In this implementation of MOOC system, Learner gets feedback only after the deadline is over, and based on the assignment evaluation. In this case, significance of the feedbacks is very less, as the learner is busy in doing the next week’s assignments and lectures. This feedback is also depends on the performance in assignments therefore not completely personalized to his/her learning.

1.1 Integration of Adaptive Recommendation System

Considering the above limitation in the MOOC system, there is a need of a system that can give the recommendation/feedback before the start of the next week/session, and the feedback should be the recommendation of some tasks to complete based on his/her learning in the current week. The primary advantage of this type of system is that, before starting the next week/session, learner is well aware of the concepts in which he/she is lacking and get some recommended tasks to complete which are personalized.
Figure 1.2: Personalized feedbacks in MOOC system

to him/her. To achieve this kind of adaptation, system needs to understand the user’s need and knowledge level.

Figure 1.2 shows the modified MOOC system, where the learner gets the feedbacks before deadline gets over. This feedback is given by system and is based on learner’s understanding about the concepts and performance in the exercises.

1.2 Adaptive Hypermedia

Adaptive Hypermedia is the collection of all techniques which can be used to enable adaptation in a web based application. Figure 1.3 shows the methods and techniques in adaptive hypermedia.

We aim to construct a system with the following goals in mind.

- The system should have information e.g. learner preferences, learning styles, knowledge levels, goals etc., of all the students, participating in the MOOC lecture.
- After the lecture, system should present the most relevant concepts to the learner, that s/he may be lacking and should prepare before the beginning of the next week/session.
The system should give a liberty to the students, to over-rule the recommendations from the system and study according to their individual goals.

1.2.1 What can be adapted?

Adaptation decisions are based on various characteristics of the users interacting with the system [8]. These characteristics together construct a user model in the system. These characteristics include user interests, user individual traits, user performance, etc. Using all the data about the user, the system should construct a learner model (user model), which represents a particular learner in the system. This model has to be updated during the course of interaction between the learner and the system. In chapter 2, we reviewed the existing techniques to construct user models in the learning environment. Chapter 2 also presents the type of information that should be stored in the user model in order to achieve adaptation.

1.2.2 How to represent knowledge?

Knowledge representation is also called Domain representation. Adaptation is largely depends on “How the domain knowledge is represented in the system”. Representation of the knowledge is specific to the learner’s model. Each learner modeling technique may have different ways to represent knowledge. Chapter 3 focused on a review of various domain representation techniques for different types of learner models. Chapter 3 also gives the mathematical formulation of domain (knowledge) representation which is used to construct our system.

Figure 1.2 shows the functionality of the system. The input to the system is the data from the week’s exercises, quizzes, etc.
1.3 Adaptive System Components

Figure 1.4 represents the components of a typical adaptive system. In the figure, it should be noted that, *Decision making engine* is taking the data from *User models* and *Hypermedia DB*. Decision making engine is responsible for making decisions about an individual learner. This tells the importance of modeling user and designing the database. Chapter 2 and Chapter 3 are focused on two important components of adaptive system i.e. *UM generator* and *Hypermedia DB*. Following list gives a brief introduction about various components of adaptive system.

- **Learner** refers to the user who attend the live lectures and interacts with the application for learning process. Learner may be in close or remote location e.g. teachers, students etc.

- **Designer** refers to the faculty or the individual who design the learning material for the domain.

- **UM generator** refers to the module which is responsible making model of a particular user based on their preferences and knowledge levels.

- **Hypermedia DB** is a database of all the learning material of a particular domain under consideration.

- **User models** are the user’s impression in the system based on individual traits such as knowledge level.
1.4 Summary

In this chapter, a brief introduction to MOOC system is presented. The possible enhancement to a MOOC system is proposed. The enhanced version of MOOC system requires an adaptive feedback system integration, for which high level requirements are presented. This chapter also presents the components of an adaptive system used in a learning environment. There are two important components namely UM generator and HypermediaDB that needs to be elaborated more. In the next chapters, the detailed architecture of adaptive application and its components are presented. Chapter 2, gives an overview of various existing efforts to model a learner in adaptive systems. Chapter 3, focused on the domain representation. The aim of Chapter 2 and Chapter 3, is to review various efforts in the direction of adaptation, in adaptive hypermedia.
Chapter 2

Learner Modeling - A Literature Survey

2.1 Introduction

Adaptive systems keep two types of information related to each user namely User profile/Learner profile and User model/Learner model. The two terms are apparently same but, in practical both are significantly different. The difference is in the amount of information that they possess. User Profile typically possesses the personal information about the user. Personal information contains preferences, the style of learning, intellectual abilities. User profile can be adjusted by the user. User model is the system generated model for each user based on some criteria. User profile serves as a precursor for User model. User profile helps the system to make User model by providing personal details about the user.

User model directly related to the adaptation property in the adaptive hypermedia systems. The adaptation is achieved by a three step process, 1. Capturing information about the user, 2. Processing the information to create or update user model and, 3. provide adaptation by using user model.

2.2 Role of User Model in Intelligent Tutoring Systems (ITS)

An Intelligent Tutoring System usually represented by four components namely 1. Student Module, 2. Tutoring Module, 3. Expertise Module, and 4. Interface Module. Out of these four components, first three components consist of specific type of user model. Figure 2.1 represents the four different modules of an ITS. Expertise Module provides all the necessary knowledge about the domain e.g. answers to the problems etc. Expertise
Module is the most important component to enable adaptive feedback. Student Module consists of the measurement of the user’s knowledge about the domain, user’s preferences and characteristics. Tutoring Module represents the rules which gives the adaptive system, a human tutor like perception [1]. Tutoring Module have the following functions in ITS:

- Domain Knowledge representation.
- Diagnosis of user’s Knowledge.
- Prediction of future actions.
- Knowledge development

The purpose of the user model is the Knowledge development of the user. Knowledge development is achieved in four stages namely what to teach, how to teach, and implementation of teaching actions [10].

- **What to teach:** To meet this purpose, tutoring module takes the current knowledge level about the user from the student module. With the help of knowledge level, tutoring module can decide the learning sequence of the domain for a particular user. Learning sequence consists of different tasks and related knowledge modules containing concepts, exercises, quizzes etc. This learning sequence helps the user to find proper and efficient path as per the user’s knowledge about the domain. This learning sequence can be of two types namely low level sequencing and high level sequencing. Low level sequencing depends only on user model whereas high level sequencing depends on domain knowledge along with user model [11].

- **How to teach:** To achieve this purpose, student model provides learning style and preferences to the tutoring module. With the help of this information, tutoring module modifies the tasks such as level of explanations, type of examples etc [11].
2.3 Information stored in User Model

There are broadly two types of information stored in user model namely Domain specific information and Domain independent information [1]. Figure 2.3 shows the types of information stored in user model.

Domain specific information contains knowledge state of user for the domain. There are different types of user models, domain specific information varies with the type of user model adopted. Domain specific information varies as the user interact with the system. Domain independent information contains all the information which is not dependent on a particular domain e.g. learning style, goals, preferences, background and experience etc [10].
2.4 Techniques for user modeling

A typical adaptive system is divided into two parts namely server side and client side [12]. The server side part is responsible for generating user model based on the user interaction and other information stored in the system e.g. user preferences. “Decision Making and Personalization Engine” takes the information from user model and makes the decision about adaptation. UM(user model) generation module is responsible for generating user model from the system [12]. Figure 1.4 represents the different components of adaptive system.

Focus of the literature survey is to find out the different methods adopted in the past for user modeling. Given below are some types of user modeling techniques in adaptive systems.

2.4.1 Bayesien Belief Network

Efforts have been made to use Bayesien Belief Network (BBN) to generate learning paths [13] in a particular domain. In [13], a method to select an optimal learning path among various candidate paths, is described. A Bayesien Belief Network is a directed graph in which each node represents an uncertain variable of interest and edges are causal or influential inks between the variables. Each node is associated with a probability table. This probability table is a set of conditional probability values that models the uncertain relationship between the node and its parents together [13]. [14] also attempts to use BBN to sequence the learning material for the learner. The knowledge material is divided into set of learning objectives. The actions corresponding to each learner can be of three types namely passive i.e. for which student do not require to use his/her knowledge (reading), Individual active i.e. for which student is required to use his/her knowledge, and collective actions i.e. actions related to group activities (discussions, projects etc.). The Utilities for each action can be calculated as:

\[ U = w_1.E_1 + w_2.E_2 + w_3.E_3 + w_4.E_4 \ldots \ldots + w_n.E_n \]

where \( w_i \) is the weight associated with each evidence.

Example

2.4.2 Machine Learning Methods

In most of the user modeling techniques listed in the next subsections, measurement of the user’s knowledge level can be done efficiently but they miss the assumptions about the interaction preferences or behavior pattern of the user. A user always leaves some pattern while interacting with the hypermedia system. Machine learning techniques can be applied to analyze the
regularities in these patterns and these regularities can later be integrated with user model [12]. Such a system becomes gradually more effective as it learns the users' interests, habits and preferences [15]. Machine learning is used to construct the behavior-oriented user models. Various efforts are made to use machine learning techniques to create user models that can be used in learning e.g. [15] presents a unique modeling technique called feature-based modeling technique that can be used in learning in tutoring systems. Various other sophisticated methods, which uses machine learning concepts, also employed in adaptive systems e.g. [16] attempts to use graph-based induction to process pre-defined domain knowledge and observations and find out some specific patterns. It is also seen that machine learning can be used for user classification [17] and plan recognition [18].

Limitations

The limitation of machine learning is the need of large amount of data [12] e.g. click streams and search logs etc. The purpose of this study is to make an integrated system for adaptation with MOOC, where this type of dataset is not available. System is provided with a user profile and some assumptions of the user's knowledge. In this case machine learning is not feasible to apply.

2.4.3 Fuzzy Logic based technique

The limitation of the traditional machine learning approach for user modeling can be overcome by using soft computing techniques. Fuzzy logic is an example of soft computing technique aimed to capture the ambiguity present in real information [12]. [19] gives the introduction to the fuzzy logic. Fuzzy logic is used in vast variety of applications e.g. [20] It is observed that fuzzy logic is used in recommendation system to recommend products in e-commerce sites. A fuzzy logic based user modeling typically models a stereotype that characterizes users, with the help of membership functions and recommendation is achieved using fuzzy AND operator. It is also noted that, fuzzy logic can also be used for filtering tasks [21]. The machine learning techniques are designed to make rules and inferences from the available data. But, in our case, the data is not available. So, rules have to be explicitly specified in the system.

Limitations

System cannot learn by applying fuzzy logic alone. Fuzzy logic along with machine learning should be used if the system needs to learn about the user’s
behavior. In our case, there is no explicit need to learn about the user’s. The system can adapt by specifying the rules explicitly.

2.4.4 Neural network based technique

Neural network can also infer a meaningful pattern form the set of large and imprecise data. [22] Gives a brief introduction to the neural networks based user modeling. For modeling human behavior neural networks proved to be the most efficient techniques [22]. For classification and recommendation tasks, neural networks systems found useful e.g. classification of user navigation paths [23], models student behavior in intelligent tutoring systems [24] [25].

Limitations

In [25], neural networks and machine learning techniques are used to construct students model in an intelligent tutoring system. In this paper, user model is generated by a “two phase learning” algorithm where in the first phase, training is done by exploiting the data from all the students interacting with the system. In the second phase learning about the users, is achieved. One need is common to most of the neural network technique i.e. learning [24] [25]. This learning requires a huge amount of data set. Considering the motivation behind this work, this technique is also not suitable for our system.

2.4.5 Fuzzy clustering based techniques

Clustering is the classification of a data point into different stereotypic clusters. There are two types of techniques for this type of classification, namely hard clustering (non-fuzzy clustering) and soft clustering (fuzzy clustering). In the hard clustering, a data point can belong to only one cluster at a time. There is very well defined boundaries of each cluster. A point can be inside the cluster or outside (binary classification) for each cluster whereas, in soft clustering (Fuzzy clustering), data points can be a part of more then one cluster at a given amount of time. A membership function is associated with every data point which specifies the degree of belonging of a particular data point to a given cluster. Fuzzy clustering can be used in measuring the knowledge level of the learner for a particular concept module, where it is very difficult to categorize the learner based on the score in the examination. Fuzzy classification technique fuzzifies the classification of learners in to different classes.
2.4.6 Neuro-Fuzzy techniques

It is the combination of neural networks and fuzzy logic for user modeling tasks. Like machine learning and neural networks technique, this technique also needs large amount of data to learn about the users.

2.5 Comparative analysis of UM techniques

Selecting a technique for the desired application is always been a challenge since every application of adaptive hypermedia is different in terms of the level of adaptation, rules for adaptation, training data available. [12] presents some guidelines to select the technique as per the need. The dimensions for comparison that are taken into considerations are listed below:

- **Computational complexity** refers to the time elapsed in off line processing.

- **Dynamic modeling** refers to the ability of the technique to alter the user model while the application is running.

- **Labeled/Unlabeled** refers to the type of data the given technique needs i.e. labeled data or unlabeled data.

- **Size of training data** refers to the amount of data the technique needs for adaptation purpose.

- **Uncertainty** refers to the ability of the technique to handle uncertainties.

- **Noisy data** refers to the fact that whether noisy data will affect user model that is produced by the technique.

It is also noticed that machine learning techniques are not suitable because in the case of first for dimensions it under-performs compared to the techniques listed in previous section [18]. Machine learning algorithm needs a large amount of off line processing in training to create user models [18], once the models are created it is very difficult to change on the fly [18], size of the data needed is also huge [18] and machine learning algorithm also needs labeled data for learning [18]. Table 2.1 compares many user modeling techniques [12]. Table 2.2 provides the recommendations for the selection of techniques based on the type of adaptive application.
<table>
<thead>
<tr>
<th>Technique</th>
<th>Complexity</th>
<th>Dynamic modeling</th>
<th>Labeled/Unlabeled</th>
<th>Size of training data</th>
<th>Uncertainty</th>
<th>Noisy data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy Logic</td>
<td>Med</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>High</td>
<td>Yes</td>
<td>Both</td>
<td>High</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Fuzzy clustering</td>
<td>High/Med</td>
<td>No</td>
<td>Both</td>
<td>Med/High</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Neuro-Fuzzy</td>
<td>High</td>
<td>Yes</td>
<td>Labeled</td>
<td>Med/High</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2.1: Characteristics of different techniques for user modeling [12]

<table>
<thead>
<tr>
<th>Task</th>
<th>Needed</th>
<th>Not Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction</td>
<td>NeuroFuzzy</td>
<td>Neural Networks</td>
</tr>
<tr>
<td>Recommendation</td>
<td>NeuroFuzzy</td>
<td>Neural Networks, Fuzzy clustering</td>
</tr>
<tr>
<td>Classification</td>
<td>Neuro Fuzzy</td>
<td>Neural Networks, Fuzzy Clustering</td>
</tr>
<tr>
<td>Filtering</td>
<td>Fuzzy Logic</td>
<td>Neural Networks</td>
</tr>
</tbody>
</table>

Table 2.2: Techniques recommended for adaptation tasks [12]
2.6 Summary

In this chapter, an overview of different user modeling techniques is given. User can be modeled by machine learning techniques but, in our case of MOOC implementation, pattern data of the users is not available. So, in this case, it become implicit to construct the rules for the system instead of learning by machine learning techniques. Table 2.2 clearly shows that, Fuzzy logic is best suited for the task of recommendations. The knowledge representation for the fuzzy logic technique needs to be reviewed. In the chapter 3, a review of various knowledge representation techniques, is given.
Chapter 3

Knowledge Representation - A Literature Survey

3.1 Introduction

Knowledge representation/ domain representation is one of the important component of any adaptive system [12]. During the actual operation, adaptive system should work similar to the human expert. To achieve this goal, the domain knowledge should be represented in a manner that there should be a mapping between the adaptive system and the human expert. In an adaptive system, adaptation depends on the type of knowledge representation technique adapted. In this chapter we present some basic ways to represent domain knowledge of the system.

3.2 Literature survey on domain knowledge representation

The knowledge of a domain can be represented in many ways and it varies as per the method to be adapted for user modeling. Our study focused on finding out the most efficient way for making an adaptive recommendation system for the learner (or user). From the section 1.5, it is clear that, for constructing an adaptive recommendation system, Fuzzy logic is the best suited technique. Therefore, we need to find out the knowledge representation techniques that are well suited for fuzzy logic.
3.2.1 Initially proposed system for knowledge representation (novice representation)

At the start of this study, we proposed a system for domain knowledge representation where the entire domain is divided into concepts.

\[ D_i = \{c_1, c_2, c_3, \ldots, c_n\} \]

\( D \) is the domain to be represented, \( D_i \) is to denote \( i \)th domain in case multi-domain system. Each concept \( c_m \) is associated with a weightage score \( w_m \) (weightage score of \( m \)th concept). The learner knowledge for each concept is recorded by taking quizzes containing the questions from different concepts. To learn the complete domain, the learner have to score 100%.

\[ \sum_{i=1}^{n} w_i = 100 \]

The quiz was structured so that each question can be mapped to one or more concepts. After the quiz, the system can measure the score related to every concept and recommend the appropriate knowledge module to study.

Limitations

In this approach the user is modeled just on the basis of the knowledge level of a particular concept, for example \((10, 2, 3, 5, 8, \ldots)\), this user possesses the knowledge level score of 10 for concept \( c_1 \). The problem with this technique is that, there is no relation among the concepts. In practical, there are many types of relationships among the concepts e.g. pre-requisite concepts, related concepts etc. The knowledge level of one concept affects the knowledge level of other concepts. In the further sections, we surveyed out some techniques which consider of the relationships among the concepts and making the adaptive system more effective.

3.2.2 Goal-oriented connectionist knowledge representation

There are attempts in the past to make systems to achieve a goal oriented teaching [2]. The system proposed uses connectionist-based distributed information for the selection of knowledge material to be presented to the learner based on his/her preferences, capabilities and needs. The system was build specifically to teach computer network course in the class.

[26] clearly says that, we should use layered architecture for domain, in which each layer should imply a different pedagogical information. Connectionist architecture is an example of layered representation of domain
Figure 3.1: A three layered connectionist architecture for domain representation [2]

knowledge. Figure 3.1 shows the three layered connectionist architecture for domain representation. The whole domain is divided into knowledge goals. Each knowledge goal is related to some concepts where each concept can be of different type e.g. for a particular concept, some concepts may be prerequisite concepts, some are related concepts. Domain in the connectionist architecture is divided into three layers where, all the knowledge goals represents the first layer. Second layer represents the the concepts associated with the knowledge goals. Each goal is associated with the actual knowledge module (set of exercises, theory, examples etc). Layer three represents the knowledge modules associated to each concept.

This approach uses the Relationship Storage Network (RSN). All the concepts in the second layer must map to all the nodes in the concept nodes(CN) of a dynamic associative memory.

\[ x(k + 1) = sat(Tx(k) + I) \]

The above equation is the RSN that represents the system. The variables used in the equation are described below:

- \( x \) is a real \( n \)-dimensional vector with components \( x_i \), where \( i = (1, 2, 3, 4....n) \), which denotes the states or activity of the \( ith \) concept node.
- \( T \) is an \( n \times n \) symmetric weight matrix with real components \( T(i, j) \).
I represents external inputs and it is a constant vector with real components $I(i)$.

sat is a saturation activation function.

Limitations

In this approach, there is only one type of relationship i.e. linguistic relationships among the concepts. The concepts are presented as outcome concepts, pre-requisite concepts. There is no quantitative relationship among the concepts. We have to build a system to teach programming course, in which there may be several concepts which are related to each other with different degrees. By assigning a quantitative relationships among concepts then the variation of knowledge level for a particular concept can be achieved with out explicitly learning that concept, which gives more intelligence to the system. Section 1.2.4 shows a techniques for domain representation with quantitative relationships among the concepts.

3.2.3 Hierarchy based knowledge representation

Efforts have been made to represent the knowledge hierarchically [27] [3]. In hierarchal representation of knowledge, the concepts are are arranged in the order of complexity. Figure [3.2] shows an example of the hierarchical representation of knowledge related to computer architecture. The advantage of representing knowledge in this form gives order of presentation of concepts.

Limitations

The type of relationship that hierarchical representation possesses is called part-of relationship where in general the concept at level $i+1$ is part-of the
concept at level \(i\). Hierarchical relationship does not specifies the dependency relationships among the concepts.

### 3.2.4 Fuzzy Cognitive Maps (FCM)

In the above mentioned techniques, the concept is considered to be learned, if the knowledge level for a particular concept is greater then a threshold value. But, in practical, there is always an uncertainty associated with the knowledge level e.g. at the same instance of time, a concept can be 70% learned and 20% known. This scenario cannot be captured by using above techniques.

Consider a circle on a plain, in fuzzy logic, a given point is neither completely inside nor outside the circle. It is inside the circle to some degree and outside to some degree. Each point in the fuzzy logic is associated with a membership function which gives the degree of membership to the circle. Formally, we call a set as fuzzy, if its membership function takes the value from \([0,1]\) not \(\{0,1\}\). If the membership function for all the elements in the universe is set to 0, then fuzzy set will always be empty.

\[
A = \phi : \forall x \in X : \mu_A(x) = 0
\]

\[4\] Considers two types of relationships among domain concepts namely essential pre-requisites and supportive pre-requisites. The set of pre-requisite relationship is the disjoint set of these two types of relationships.

\[
R = E \cup S \\
E \cap S = \phi
\]

- \(R\) represents the set of prerequisite relations.

- \(E\) represents the set of essential prerequisite relations.

- \(S\) represents the set of supportive prerequisite relations.

\[4\] defines the essential prerequisite as a fuzzy set i.e. for a concept \(c_j\) if the concept \(c_i\) is declared as essential prerequisite then this is may not be 100% essential prerequisite for the concept \(c_j\). There is some degree of membership associated with \(c_i\) to \(c_j\) which defines the degree of prerequisites. This membership function is represented by \(\mu_E(c_i, c_j)\).

\[
E \subseteq C \times C
\]  \hspace{1cm} (3.1)
\[ \mu_E : C \times C \rightarrow [0, 1] \]  
\[ E = \left\{ \frac{\mu_E(c_i, c_j)}{(c_i, c_j) \in C} \right\} \]

Each node is associated with a knowledge level 'l'. Membership functions are defined which value of fuzzifies the classification of learner on the basis of 'l'. Learner knowledge about a particular concept \( c \) is therefore expressed by providing the values of three fuzzy sets given below. Figure ?? gives the variation in the values of membership functions with the increase in knowledge level. (\( \mu_U, \mu_K, \mu_L \))

\[ \mu_U + \mu_K + \mu_L = 1 \]
\[ \mu_U > 0 \Rightarrow \mu_L = 0 \]
\[ \mu_L > 0 \Rightarrow \mu_U = 0 \]

Figure 3.3: Membership Functions for fuzzy sets of unknown (CU), known (CK), and learned (CL) [4]

3.3 Summary

Fuzzy Cognitive Map (FCM) is well suited to capture the dependencies among the concepts of a domain. We can represent each node as a concept module of a given domain and a membership function on knowledge level for determining the extent of information the learner may have for the particular concept. In this chapter, we reviewed several ways of representing
knowledge. Knowledge representation is specific to the user modeling technique adapted. In our application, we chose Fuzzy logic for user modeling and Fuzzy cognitive map for knowledge representation.
Chapter 4

Proposed System for Giving Personalized Feedbacks

4.1 Introduction

We are aimed to design a system that can be used with the MOOC system. The system should give the personalized feedbacks and provides personalized learning material. The system should adapt the needs of the users dynamically and update itself during the operation. In the context of user modeling, we surveyed out that, fuzzy logic is the best suited technique for our application because of the absence of training data (please refer chapter 2 for more details). The appropriate technique for knowledge representation is the fuzzy cognitive maps (FCMs) as per the needs of fuzzy logic while designing the user model.

4.2 4-layered architecture of the system

The system will to be used with the MOOC online courses, the information about the users’ preferences, information about the user’s knowledge about the particular concept etc. have to be captured online. After the completion of the week, the system gathers all the information about knowledge levels of all the students for all the concept taught in the week. The system must model each user and generate the personalized tasks to be completed before appearing for the next week. Figure 4.1 shows the input data to the system.

Figure 4.2 shows the 4-layered architecture of the system. Layer 1 among the four layers represents the set of lectures of the complete course (in this case lecture represents a week), Layer 2 represents the set of concepts associated with each lecture (related or independent), Layer 3 shows the set of objectives (hierarchically arranged) belong to each concept module, Layer 4
shows the knowledge modules e.g. theory, exercises, quizzes etc.

### 4.3 Knowledge Representation

We have surveyed out various techniques for knowledge representation in chapter 2. It is found out that, fuzzy cognitive maps are the most suited technique for knowledge representation. The technique is adopted from the existing systems in the field of intelligent tutoring system (ITS) of the same kind [28].

FCM divides the whole system into the combination of concept modules and relationships among those concepts modules. Each concept module corresponds to each node of the FCM. The relationships among different concepts are represented by directed arcs. Figure 4.5 shows a typical domain model in FCM.
4.3.1 Example
We are considering a basic programming course for our study. In this domain, we each week of MOOC course is defined as a set of concepts modules. Let’s consider for a given week, 4.1 shows the the concept modules that needs to be covered.

Each concept module contains various objectives which are hierarchically arranged on the basis of difficulty level of the objectives. The example objectives related to the given concepts are shown in Table 4.2. The first column specifies the concept ID, second column specifies objective ID and third column specifies objective description. \( obj_p \) represents the prerequisite objective that needs to be achieved in order to unlock the concept module. Note that all the objectives of a particular concept are hierarchically arranged on the basis of difficulty level. There is a relationship between the objectives of two different concepts for example - if a learner knows \((C_2, obj_3)\) then he/she must know about \((C_1, obj_3)\) (although its not guaranteed), so we can skip \((C_1, obj_3)\) in our recommendation. In this way, the recommendations can be achieved which is personalized to the learner. We need to declare the rules for recommendation.

4.3.2 Rules for Recommendation
In order to achieve the personalized recommendations, we need to exploit the relationship among the objectives of the concepts. Figure 4.4 shows the relationships among the objectives of the concepts. There are two types of relationships namely, hierarchical relationship and non-hierarchical relationship. Solid arrows shows the hierarchical relationship whereas dashed arrows shows non-hierarchical relationship. hierarchical relationship is the relationship among the objective of a given concept module whereas, hierarchical relationship is the relationship among the objectives of different concept modules. An objective is considered to be achieved, if a learner fulfills two conditions:

- viewed all the videos related to that objective.
- attempted all the exercises and quizzes with a minimum threshold.

Given below are the rules that needs to be implemented in order to achieve adaptive recommendation.

- **Rule 1:** If a learner achieves a particular objective, then all the objectives from root to this objective in hierarchical ordering, are considered to be achieved.
Rule 2: If the learner achieves an objective, then all the objectives related to it in a non-hierarchical relationship should be considered to be in ready state.

Rule 3: If a particular concept module has to be recommended to the user then the last concept that is in ready state should be shown to the learner.

Rule 4: If a student achieves an objective which was in ready state then Rule 1 should be applied.

Figure 4.3 shows the state diagram of objectives. In addition to the above rules, two concept modules are related if atleast one objective between two concepts are related by a non-hierarchical relationship. There is an arrow in the graphical representation of concepts \((C_i, C_j)\), if atleast one objective of \(C_i\) is related to \(C_j\).

There can be more than one relationship graphs possible for a given course module. Figure shows a possible relationship graph for our example table.
4.4 Modeling Learner

Learner modeling is the system’s assumption about the learner. In our application this assumption is largely focused on the measurement of the learner’s knowledge for a particular concept module. Learner’s knowledge can be represented by four tuples for a particular concept module given by:

\[(Un, Uk, K, L)\]

Elaboration for above four parameters are given below:

- **Unknown**(Un): the knowledge level associated with each concept module is from 0% to 60%.
- **Unsatisfactory known**(Uk): knowledge level is 55% to 75%.
- **Known** (K) : knowledge level is 70% to 90%
- **Learned**(L): knowledge level is 85% to 100%

These four parameters are basically four fuzzy sets based on the knowledge level for a particular concept. Let the knowledge level for a given concept is \( x \). The membership function for the above four fuzzy sets are given by:
<table>
<thead>
<tr>
<th>Concept id</th>
<th>Objective id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>obj$_p$</td>
<td>syntax of if-else statement</td>
</tr>
<tr>
<td>$C_1$</td>
<td>obj$_1$</td>
<td>check a given number is positive or negative</td>
</tr>
<tr>
<td>$C_1$</td>
<td>obj$_2$</td>
<td>check a given number is even or odd</td>
</tr>
<tr>
<td>$C_1$</td>
<td>obj$_3$</td>
<td>given two numbers, check which one is greater</td>
</tr>
<tr>
<td>$C_1$</td>
<td>obj$_4$</td>
<td>given three numbers, check which one is greater</td>
</tr>
<tr>
<td>$C_2$</td>
<td>obj$_p$</td>
<td>syntax of switch statement</td>
</tr>
<tr>
<td>$C_2$</td>
<td>obj$_1$</td>
<td>check a given number is positive or negative</td>
</tr>
<tr>
<td>$C_2$</td>
<td>obj$_2$</td>
<td>check a given number is even or odd</td>
</tr>
<tr>
<td>$C_2$</td>
<td>obj$_3$</td>
<td>given two numbers, check which one is greater</td>
</tr>
<tr>
<td>$C_2$</td>
<td>obj$_4$</td>
<td>given three numbers, check which one is greater</td>
</tr>
<tr>
<td>$C_3$</td>
<td>obj$_p$</td>
<td>syntax of the for statement</td>
</tr>
<tr>
<td>$C_3$</td>
<td>obj$_1$</td>
<td>print the first $n$ integers in a for loop</td>
</tr>
<tr>
<td>$C_3$</td>
<td>obj$_2$</td>
<td>counting in a for loop</td>
</tr>
<tr>
<td>$C_3$</td>
<td>obj$_3$</td>
<td>calculating sum in a for loop</td>
</tr>
<tr>
<td>$C_3$</td>
<td>obj$_4$</td>
<td>calculating average in a for loop</td>
</tr>
<tr>
<td>$C_3$</td>
<td>obj$_5$</td>
<td>calculating max/min in a for loop</td>
</tr>
<tr>
<td>$C_4$</td>
<td>obj$_p$</td>
<td>syntax of the while statement</td>
</tr>
<tr>
<td>$C_4$</td>
<td>obj$_1$</td>
<td>print the first $n$ integers in a while loop</td>
</tr>
<tr>
<td>$C_4$</td>
<td>obj$_2$</td>
<td>counting in a while loop</td>
</tr>
<tr>
<td>$C_4$</td>
<td>obj$_3$</td>
<td>calculating sum in a while loop</td>
</tr>
<tr>
<td>$C_4$</td>
<td>obj$_4$</td>
<td>calculating average in a while loop</td>
</tr>
<tr>
<td>$C_4$</td>
<td>obj$_5$</td>
<td>calculating max/min in a while loop</td>
</tr>
<tr>
<td>$C_5$</td>
<td>obj$_p$</td>
<td>syntax of the do-while statement</td>
</tr>
<tr>
<td>$C_5$</td>
<td>obj$_1$</td>
<td>print the first $n$ integers in a do-while loop</td>
</tr>
<tr>
<td>$C_5$</td>
<td>obj$_2$</td>
<td>counting in a do-while loop</td>
</tr>
<tr>
<td>$C_5$</td>
<td>obj$_3$</td>
<td>calculating sum in a do-while loop</td>
</tr>
<tr>
<td>$C_5$</td>
<td>obj$_4$</td>
<td>calculating average in a do-while loop</td>
</tr>
<tr>
<td>$C_5$</td>
<td>obj$_5$</td>
<td>calculating max/min in a do-while loop</td>
</tr>
<tr>
<td>$C_6$</td>
<td>obj$_1$</td>
<td>print all the even numbers between 1 to 100 using</td>
</tr>
<tr>
<td>$C_6$</td>
<td>obj$_2$</td>
<td>print all the odd numbers divisible by 7, between 1 to 100</td>
</tr>
<tr>
<td>$C_6$</td>
<td>obj$_3$</td>
<td>print all the numbers divisible by 3 or 5, between 1 to 100</td>
</tr>
<tr>
<td>$C_6$</td>
<td>obj$_4$</td>
<td>print all the odd numbers divisible 3 or 5 but not 3 and 5, between 1 to 100</td>
</tr>
</tbody>
</table>

Table 4.2: An example of concept to objective table
Learner is modeled using overlay modeling \[28\] where the learner's knowledge is the subset of the expert knowledge. The representation of learner knowledge level is taken from \[28\]. The values of the four above mentioned functions for a particular learner, models the knowledge level of that learner for a particular concept.

### 4.4.1 Example Scenario

If the learner's knowledge level $x$ is calculated as 89% for a particular concept module the the values of all the four membership is given as:

$$ (\mu_{Un}, \mu_{UK}, \mu_{K}, \mu_{L}) = (0, 0.2, 0.8) $$
The above equation represents that, the particular concept is 20% known and 80% learned. The knowledge level of the learner is calculated with the help of quizzes and exercises. One important factor also needs to be considered while calculating the learner’s knowledge level i.e. type of knowledge. In the programming tutorial, there are two types of knowledge associated with the concept i.e. syntax knowledge and semantic knowledge. Each test (quiz) must have some part of syntax knowledge and some part of semantic knowledge. Let \( \mu_{\text{syntax}} \) and \( \mu_{\text{semantic}} \) are the two membership functions for the two sets which represent the syntax knowledge and semantic knowledge.

\[
\mu_{\text{syntax}} + \mu_{\text{semantic}} = 1
\]

4.5 Low Level System Architecture

There are three steps that need to be followed in order to achieve adaptive recommendation for the learner.

- 1. System Initialization.
- 2. Update the system’s assumptions about the learner.
- 3. Prioritize the concept modules that need to be recommended.

4.5.1 System Initialization

Initialization of the system refers to capturing initial data about the user e.g. user preferences, learning style, knowledge level about all the concepts taught in the week. The initial parameter that we are focused on is knowledge level of all the concepts taught in the week. This knowledge level can be captured by using exercises and quizzes of the week. The structure of the quizzes and exercises should be pre-defined and the structure is such that each question must be mapped to a given objective of a given concept. The number of concepts are also finite for a given week.

There is a hierarchical relationship among the objectives. So, if a learner meets the lower level objective, the upper level objective will be treated as completed. After the objectives of a particular concept is achieved, the knowledge level of the concept module is calculated as:

\[
\text{Knowledge level}(C_i) = \frac{\text{Number of objectives achieved for } c_i}{\text{Total number of objectives in } c_i}
\]

4.5.2 Update the knowledge levels of related concepts

Figure 4.7 shows the state of each concept module after the first step. A concept may be related to other concepts of the domain. If an objective of
a particular concept is achieved, then it will also affect the knowledge levels of other concepts of the domain. After the achievement of an objective, the related objectives of the other concept module becomes \textit{ready}.

In this way, for a particular concept module, all the related concept modules’ objectives becomes \textit{ready}, if they area in a non-hierarchical relationship with the objectives of the given concept.

4.5.3 Prioritization of Concepts

After the concept nodes are initialized, we get the quadruple values of all the concepts related to a particular lecture. Now we need to recommend the objectives that needs to be achieved. The need of the system is such that, the learner should able to cover all the concepts in minimum amount of time. There can be many approaches to prioritize the concept module for the learner. Some of the approaches are given below:

- **Approach 1**: After the completion of a course module, all the related concepts have some objective in the \textit{ready} state. We can select the next concept on the basis of the distance of the objective in the \textit{ready} state from the last objective of that concept. The concept having the minimum distance between the last objective and the the objective having \textit{ready} state will be chosen for recommendation. The advantage of this approach is that, the chosen recommended concept module have a few objectives to be covered in order to complete the concept module.

  \textit{Limitation}: In this case we are not considering knowledge level of learner. If according to the membership function, the knowledge level
is learned and this concept have the minimum distance then it will be chosen over the concepts which are in the known state.

- **Approach 2:** After the completion of a course module, the next concept is selected such that the selected module have the maximum number of objectives that can make others objective to ready state. The advantage of this approach is that, everytime we select the concept that affects other concepts largely. This makes the faster completion of concept modules.

- **Approach 3:** In both of the above techniques, Learner’s knowledge level is not taken into consideration. After the first step, four tuples are associated with each concept module. We should try to make all concepts from unknown to unsatisfactory known then to known progressively. If all concepts are known then try to make all concepts as learned.

- **Approach 4:** In approach 3, we select all the concept having maximum unknown level for a particular concept. We can select the concept that are in known level and then proceed to unsatisfactory known and progressively to unknown. In this case, known and unsatisfactory known can make the concept that are in unknown state, more easily understandable if there is a relation among the concepts.

**4.6 Summary**

In this chapter, we presented the low level architecture of the system. Knowledge representation/domain representation is covered by considering a case of basic programming course. Fuzzy logic rules are presented for measuring the knowledge level of learner for a particular concept module. Adaptive recommendations can be achieved in three steps namely, initialization, updation and prioritization. There can be various approaches for concept prioritization, three of which are introduced in the chapter. The system’s success depends on the selection of the prioritization approach of concepts which is domain dependent. In the next chapter, the application details are presented.
Chapter 5

Implementation

In this chapter, we presented a rough implementation detail of the system. Since the requirement gathering phase is in the process, use case diagrams and User interfaces are updating daily. We are giving the latest user interfaces and use case diagrams.

5.1 User interfaces

We have designed two interfaces namely login page and learner page. Figure ?? shows the login page for the authentication of the users.

5.1.1 Login page

5.1.2 Learner’s page

Figure ?? shows the learner’s page. In the learner’s page, learner can view the objectives of all the weeks. In this page, we have arranged the objectives in the decreasing order of recommendation. Top objective is highly recommended for the learner. A green light means the objective is ready to be learned, A yellow signal means the objectives are moderately recommended to the learner and a red signal means the objective is not recommended to the learner.

5.1.3 Login page

5.2 Use-case diagram

Figure ?? gives the preliminary use case diagram based on the initial requirements. The requirements are still updating.
Figure 5.1: Login page

Figure 5.2: Learner’s page
Figure 5.3: Use case Diagram
Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this report, we presented the existing efforts in the direction of adaptive learning. We analysed that, recommendation system which should be adaptive, dynamic and self updative, and less complex, can be constructed using the Fuzzy logic approach. In this approach, we organized learning material in to a four layered architecture. The representation is the combination of hierarchical and network based approach. Unlike other approaches, we do not need any expert to fix the dependency values which makes it domain independent and easy to implement.

6.2 Looking ahead

We have constructed the basic implementation model but, the requirements are not fully captured in formal way. In the next stage, we are looking forward to implement this system with edx platform so that it can be used in july for teaching CS101. We need to implement this system in python because most part of edx platform is written in the same language. We also need to create test plans for our system before actual testing and deployment.
Bibliography


39


