A COURSE RECOMMENDATION SYSTEM BASED ON LEARNER CAPACITY USING RESTRICTED BOLTZMANN MACHINE

PROJECT REPORT

Submitted by

DHANYA M

REG NO : TKM20CSCE04

In partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Under the guidance of
Prof. Thushara A

Thangal Kunju Musaliar College of Engineering
Kerala

JULY 2022

Thangal Kunju Musaliar College of Engineering
Dept. of Computer Science & Engineering

C E R T I F I C A T E

This is to certify that this report titled ***A Course Recommendation System Based on Learner Capacity Using Restricted Boltzmann Machine*** is a bonafide record of the **Project** presented by **DHANYA M (TKM20CSCE04)**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **M.Tech in Computer Science & Engineering** in **APJ Abdul Kalam Technological University** .

Coordinator	Head of the Department
Dr. Ansamma John	Dr. Dimple A Shajahan
Professor	Associate Professor
Computer Science & Engineering	Computer Science & Engineering

ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. Firstly I would like to thank the almighty for giving me the wisdom and grace for making my project a memorable one. I thank him for steering me to the shore of fulfillment under his protective wings.

I express my sincere gratitude to **Dr. T A Shahul Hameed**, Principal of T.K.M College of Engineering for giving me an opportunity to present my project. I would like to thank **Dr. Dimple A Shajahan**, Associate Professor and Head of the Department, CSE, TKMCE, for her constant support and encouragement throughout the work.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my guide **Prof. Thushara A**, Associate Professor, CSE, TKMCE, and project coordinator **Dr. Ansamma John**, Professor, CSE, TKMCE for their expert guidance, cooperation and immense encouragement. I also extend my thanks to the entire faculty members and staffs of the Department of Computer Science & Engineering, TKMCE, who has encouraged me throughout this work.

I also express my thanks to my loving parents and friends, for their support and encouragement in the successful completion of this project work.

DHANYA M

Abstract

The education field has undergone a rapid transformation with the advent of online courses, that aid in skill acquisition and life long learning. However, in the present scenario, the major limitation of these courses is the high dropout rate. This is due to the poor course recommendation and the low learner interaction associated with online courses. In order to reduce the dropout rate, course recommendation must be given based on the learning capacity of students. In this weak learner interaction scenario, cognitive diagnosis, which is a psychometric technique used to discover the proficiency level of students in specific knowledge components could not be used. Instead, a variant of this technique called Multi-dimensional Item Response Theory (MIRT) can be incorporated in course recommendation systems to suitably represent learner's learning state by obtaining implicit response on the followed course. In this work, course recommendation based on learner capacity is implemented by integrating MIRT into collaborative filtering, which is implemented using Restricted Boltzmann Machine. The Open University Analytics Dataset is used as the base experimental data. Experimental results and analysis show that this course recommendation system has better performance in terms of different quantitative metrics like precision and mean absolute error.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Online courses are getting immense popularity as an alternative mode of education with the advent of several online learning platforms like Coursera, Skillshare, Udemy, Codecademy, Edx, Pluralsight, Future Learn and Moodle. More and more people are utilising these learning platforms for skill acquisition, especially since the outbreak of Covid-19. A growing number of universities are making their programs available online at various levels and disciplines. In the present scenario, online learning is becoming a huge catalyzer for people to help them adopt the world's rapid change by acquisition of new skills.

The extremely high dropout rate[3] associated with online learning poses a challenge. This is due to the fact that learners find it difficult to choose a suitable course from the vast array of available options and mostly ends up with a wrong selection. To assist learners in course selection, recommendation systems are widely used in education sector. Content-based and session-based recommendations are the most commonly used recommendation services in online learning platforms today. A learner will only receive recommendations for courses with similar content to that he/she has already followed. The learner's learning capacity is not taken into consideration by these recommendation systems, which mostly suggest courses that are too challenging for him. The majority of research that has been conducted ([2], [6]–[8]) focus on the suggestion of certain objects (courses), similar to conventional recommendation systems that use a collaborative filtering framework [9], where each user is embedded into a fixed-length vector. However, these are insufficient to address the following two pressing issues:

- tracing learner's learning state.
- providing explainable tutorial information for a recommended course list.

A new course recommendation methodology is required to satisfy the actual needs of online learners in light of the aforementioned limitations. This new model should recommend courses by dynamically determining the learner's knowledge state or capacity. The major challenge, however, is determining how to explicitly quantify the learner's learning state.

For explicit quantification of the learner's learning state, a tool called Cognitive diagnosis[10] can be used. Cognitive diagnosis employs psychometrics, which is a specialised field within psychology and education, devoted to testing, measuring, assessment and related activities. This technique is widely used in intelligent tutoring systems (ITS) and other educational software containing an artificial intelligence component ([11]–[14]), to estimate the cognitive structure or knowledge state about specific subjects. Cognitive diagnosis requires a strong interactive environment in which the learner's response to a question can be instantly collected to assess the learning potential. Conventional online learning platforms have very little learner interaction. Therefore it is virtually impossible to get explicit feedback on whether or not a learner has achieved a particular course learning objective. Thus, it is difficult to incorporate cognitive diagnosis directly into online course recommendation systems.

Fortunately, there is a variant of cognitive diagnosis, called Multi-dimensional Item Response Theory (MIRT), which can be effectively used to collect implicit responses from learner's in a weak interaction environment. MIRT is a broad class of probabilistic models that characterise an individual's likelihood of responding to an item based on item parameters and multiple latent traits. It provides an ideal foundation for modelling performance in complex domains by accounting for multiple basic abilities at the same time and representing various combinations of the abilities required for different test items.

Recommendation systems need a lot of information about user's historical behaviour in order to make accurate predictions. To filter out useless and redundant information, this course recommendation system uses a technique called Collaborative Filtering. It is a method of making automatic predictions about a user's interests by collecting preferences or taste information from many users.

The objective of this work is to reduce the drop out rate in online learning platforms by providing better course recommendations. For that, a time effectiveness hypothesis is introduced by utilising MIRT to obtain the learner's response on a followed course. Based on the obtained response, the MIRT model is designed to estimate the learner capacity. Learner capacity estimation is treated as a user attribute and is integrated into a collaborative filtering framework implemented using Restricted Boltzmann Machine to provide better course recommendation.

The remainder of this report is organized as follows. Chapter 2 recalls some related works used as reference for completing this study. Chapter 3 includes the data collection section and all background information used to carry out this study. Chapter 4 presents the results of the study. Finally, conclusion and future works are provided in chapter 5.

Chapter 2

Related Works

Recommendation systems can be used in a variety of fields, including the educational sector. Such systems are primarily concerned with providing high educational standards and are attempting to improve the process of teaching and learning. Course recommendation systems recommend appropriate courses to learners based on prior knowledge of the learner's behaviour. Most works in this field are based on collaborative filtering, that use similarity between learners and courses simultaneously to provide recommendations.

Learners enrich their experiences through learning courses in online platform and thus their capacities should increase. This is demonstrated in the work 'Capacity Tracing-Enhanced course recommendation'[1], where the historical learning information of each learner obtained from a real world data set is divided into four parts in time dimensions and the average topic difficulty of followed courses in each part is computed. The results on some topics are displayed in Fig. 2.1, demonstrating that learners tend to follow more and more difficult courses in the learning process. Further, if the learner capacity is taken into account,the similarity measurement between learners may go beyond the followed courses. As shown in Fig. 2.2, based on the prerequisite relations and difficulties of courses, although user B and user A followed more common courses comparing with user C, user C is much more similar to user A because they are in the same capacity level (on Course 5).

In online education context, tutoring services can play a significant role. It ensures that the recommended course has explicit correspondence with user attributes to provide an explainable note. But existing solutions lack such explainable notes, and thus learners are easily confused about the recommended courses.

C. Piech et.al.[30] explored the utility of using Recurrent Neural Networks (RNNs) to model student learning. The RNN models donot require explicit encoding of human domain knowledge and is capable of capturing more complex represention of student knowledge. The learned model can be used for

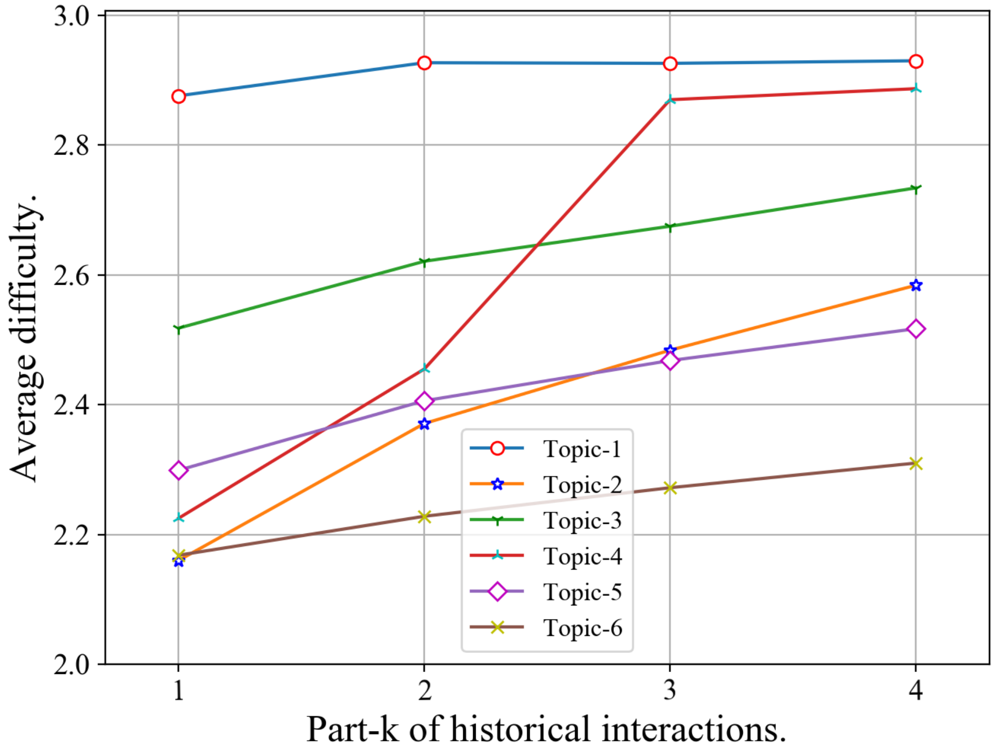

Figure 2.1: Demonstrating the importance of tracking learner capacity.

intelligent curriculum design and also allows straight forward interpretation and discovery of structures in student tasks.

B. Hidasi et.al.[34] proposed an RNN based approach for session based recommendations. The commonly used matrix factorisation approach, used in recommendation systems can work satisfactorily only on short-session based data. This work overcomes the limitations of matrix factorisation method by considering practical aspects of the task and introducing several modifications to classic RNNs like ranking loss function.

Y. Pang et.al.[6] presented a Collaborative Filtering Recommendation for MOOCs. In this work, they used a method called Multi Layer Bucketing Recommendation (MLBR). Here the learner vectors are scattered into buckets, such that each bucket contain similar learners with more courses in common. But this method does not consider relation among courses.

T. Zhu et.al.[13] proposed a three step personalised question recommendation method which combines the complementary advantages of Probabilistic Matrix Factorisation and Cognitive diagnosis. At first, the student's profi-

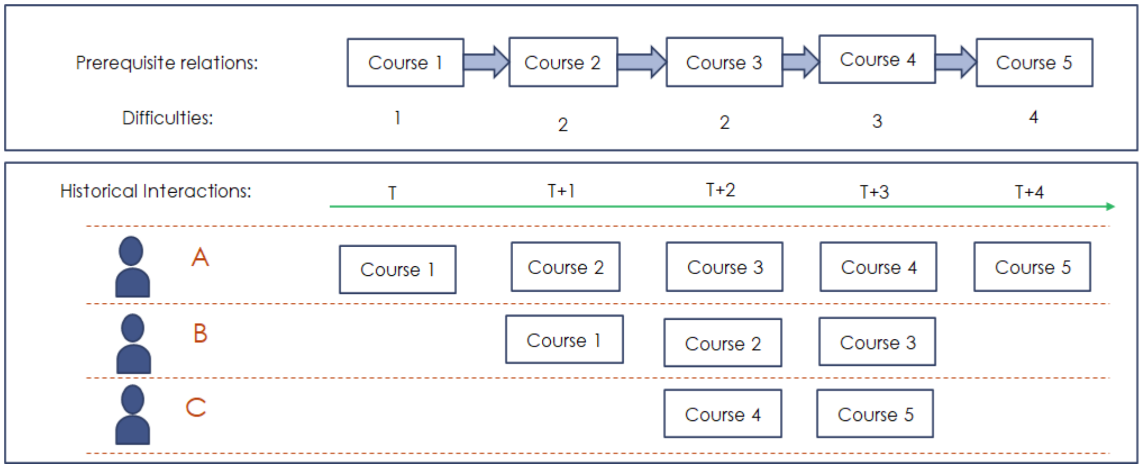

Figure 2.2: Learner similarity based on capacity.

ciency on each question is modelled using cognitive diagnosis depending on student scores and question-skill correlation matrix. Then the student's performance is predicted by using probabilistic matrix factorisation combined with student's question proficiency. Finally, questions are recommended based on predicted performance and difficulty of questions. This work succeeds in overcoming the limitation of collaborative filtering, which ignores the knowledge states of students; as well as that of cognitive diagnosis which can only model the knowledge state for a single student neglecting the common features of similar students.

Y. Cheng et.al.[16] devised an explanatory probabilistic approach to track the knowledge proficiency of students over time, leveraging educational priors. In order to track the mastery of knowledge over time, each student is represented as a knowledge vector at each time in a same knowledge space. Then two classical educational theories (Learning curve and Forgetting curve) are used as priors to capture the change of each student's proficiency over time. Finally, a probabilistic matrix factorisation framework is used to track the student's knowledge proficiency.

H.Zhang et.al.[8] proposed Personalized Course Recommendation System based on Deep Belief Network (DBF) in MOOCs environment to deal with sparse data and to handle high dimension attributes of online learning users. This work utilises the high performance of Deep Belief Network in function approximation, feature extraction and prediction classification.

Y. Pang et.al.[7] proposed Adaptive Recommendation for MOOCS (ARM) to provide a solution to the low learner satisfaction and loneliness feeling that tend to cause more dropouts. ARM combines Collaborative Filtering with time series to improve the recommendation accuracy.

L. Fang et.al.[18] presented a content based approach for personalised grammar question recommendation, which recommends similar grammatical structure and usage questions for further practicing. A structure called parse-key tree was used to capture the grammatical structure usage of grammar questions. Then similarity between the question query and database questions are computed to provide the recommendation. This work overcomes the inefficiency of content similarity calculation methods used in existing recommendation methods.

J. Zhang et.al.[2] attempted to solve the problem of course recommendation by a hierarchical reinforcement learning model. The model jointly trained a profile reviser and a basic recommendation model to enable the recommendation model being trained on user profiles revised by the profile reviser. This work addressed the limitations of attention based recommendation models especially when users enroll in many diverse courses. But this work cannot connect the courses to external entities like knowledge possessed by the user.

Bhaskar Mondal et.al.[36] proposed a machine learning approach to recommend suitable courses to learners based on their learning history as well as their past performance. This framework initially classifies learners based on their past performance using k-means clustering algorithm. Then collaborative filtering is applied in the clusters to recommend a few suitable courses.

Xuetao and Feng Liu[1] proposed a method to reduce the dropout rate in MOOCs by tracing the learners learning state before suggesting course recommendation. In this work, they had integrated Multi-dimensional item response Theory (MIRT) into Collaborative Filtering (CF). MIRT is extended to capacity tracing for course recommendation in MOOCs through the introduction of a concept called knowledge tracing.The capacity tracing enhanced course recommendation model is implemented using Neural Collaborative Filtering (NCF) and Gated Recurrent Unit (GRU).

Chapter 3

Methodology

Course recommendation based on learner capacity is provided by combining Multi-dimensional Item Response Theory (MIRT) with Collaborative Filtering. The MIRT tracks the learner capacity and provides capacity related input to the collaborative filtering framework. The collaborative filtering framework is implemented using Restricted Boltzmann Machine.

3.1 Multi-Dimensional Item Response Theory (MIRT)

Item Response Theory (IRT) [24] is a broad class of statistical models that track the probability of an individual choosing a particular item response. According to IRT, an individual's item response is based on the specific item and the characteristics of the individual. MIRT is derived from IRT. MIRT explains an item response according to an individual's standing across multiple latent dimensions. As per MIRT, each individual has multi-dimensional latent capacity θ_i and the response X_{ij} (0 or 1) to an item j by individual i is determined by θ_i and the characteristics of item j, d_j, together. Several MIRT models are proposed such as [25], [26]), and [27]. Equation (3.1) shows the probability that an individual i responds to an item j given his latent capacity in k-dimensions.

$$P(X_{ij} = 1|\theta_i, d_j) = \frac{\sum_{k=1}^{K} e^{\theta_{ik} - d_{jk}}}{1 + \sum_{k=1}^{K} e^{\theta_{ik} - d_{jk}}} \tag{3.1}$$

Utilising maximum likelihood estimation for $\Pi_{ij} P(X_{ij} = 1)$ based on some observed responses, we can simultaneously estimate θ_i and d_j and these estimated values can be identically distributed. MIRT models consider the capacity of a subject as static during a test. So it cannot work in a dynamic scenario.

3.2 Collaborative Filtering

Collaborative Filtering (CF) is a technique widely used in recommendation systems. CF makes predictions about the interest of an individual by collecting the interests of many other people. Here the assumption is that, if two learners, A and B has already followed a common course C, then there is a higher probability of A choosing another course followed by B than that of a randomly chosen learner. The predictions made by CF are specific to the user but the information used for prediction is extracted from many users. CF applications typically need very large data set.

There are two kinds of Collaborative Filtering techniques.

- Memory based approach
- Model based approach

In memory based approach, no training or optimisation is needed. Similarity between users or items are calculated using basic arithmetic operations like, cosine similarity or Pearson correlation coefficient. Even-though this approach is simple, its performance degrades if the dataset is sparse.

In model based approach, machine learning algorithms that predict the user's ratings of unrated items are used to develop CF models. Clustering based algorithms, Matrix Factorization based algorithms or Deep learning based algorithms can be used to implement model based Collaborative Filtering.

3.3 MIRT-CF Unified Framework

In order to improve the effectiveness of course recommendation, Multi dimensional Item Response Theory is integrated into collaborative filtering as shown in Fig. 3.1. In collaborative filtering framework, fixed length vectors called learner embedding and course embedding are respectively used to represent each learner and course. Moreover, learner embedding l_i is split into $g(\theta_i)$ and an implicit vector l'_i. θ_i is the multi dimensional capacities of l_i, and $g()$ is a linear transformation which is used to make $g(\theta_i)$ and l'_i on the same level. Similarly, course embedding c_j is split into $g(d_i)$ and c'_j. MIRT can be used to estimate θ_i and d_j explicitly, to represent multi-dimensional explainable human capacities in online learning scenario. l'_i and c'_j are utilised to model the information which are not related to $l_i's$ capacity and $c_j's$ difficulty, like $l_i's$ personal interests, $c_j's$ teaching style etc.

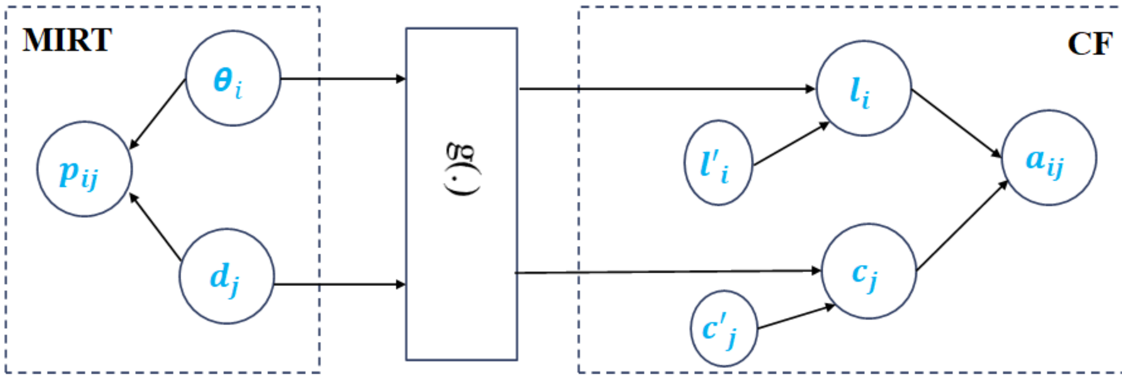

Figure 3.1: Unified framework to integrate MIRT into Collaborative Filtering.

Estimation of p_{ij}, which is the probability that l_i has mastered the knowledge components in course c_j is the key challenge in a weak-interaction scenario like the online learning platform. By using MIRT, the response of learner l_i to a Knowledge Component in course c_j can be obtained by using Equation 3.2.

$$p_{ij} = P(X_{ij} = 1) = \sqrt{(\alpha_{ij} + \eta) \times \frac{\tau_{ij} + \gamma}{D_{ij} + \gamma}} \tag{3.2}$$

The course completion ratio α_{ij} should be ideally equal to $\frac{\tau_{ij}}{D_{ij}}$, where τ_{ij} is the learning duration and D_{ij} is the course duration. Since different learners have different learning pace, the geometric mean of α_{ij} and $\frac{\tau_{ij}}{D_{ij}}$ is considered. The value of constants η and γ are selected according to the minimum units of α_{ij} and τ_{ij} (0.01 and 1 respectively) to relive the impact of missing data.

The MIRT model is designed to establish a connection between learner capacity, course difficulty and the learner's response. The capacity of learner l_i is represented by the vector $\theta_i = [\theta_{i1}, \theta_{i2}, ..., \theta_{iK}]$ where K is the number of known course topics. Similarly, the difficulty of course c_j is denoted by $d_j = [d_{j1}, d_{j2}, ..., d_{jK}]$ in K dimensions. Equation 3.3 is used by the MIRT model to estimate the ideal probability that learner l_i has mastered course c_j

based on l_i's capacity and c_j's difficulty.

$$q_{ij} = P(X_{ij} = 1|\theta_i, d_j) = \frac{\sum_{K=1}^{K} d_{jk} \times e^{\theta_{ik}-d_{jk}}}{1 + \sum_{K=1}^{K} d_{jk} \times e^{\theta_{ik}-d_{jk}}} \tag{3.3}$$

The learner capacity θ_i can be estimated and the course difficulty d_j can be updated by minimizing the KL divergence between the observed response p_{ij} and the ideal response $q_i j$ as per Equation 3.4.

$$D_{KL}(p||q) = \sum_{i,j} p_{ij} \times log\frac{p_{ij}}{q_{ij}} \tag{3.4}$$

The estimated learner capacity is treated as side information and considered as learner attributes in order to integrate capacity tracing into collaborative filtering for course recommendation. The recommendation model predicts the interaction between learner l_i and course c_j where the learner embedding l_i is obtained by concatenating the capacity related attributes of learner l_i with other attributes and course embedding c_j is obtained by concatenating difficulty related attributes of the course with other course attributes. Table 3.1 shows all the symbols used in this work along with their descriptions.

In this work, the collaborative filtering model is implemented using Restricted Deep Boltzmann Machine. The top-K courses with high probability are selected as the recommendation list.

3.4 Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM) is a machine learning algorithm used for dimensionality reduction, classification, regression, feature learning, topic modelling and collaborative filtering. Being a stochastic generative neural network RBM is capable of inferring the probability distribution from input dataset. Depending on the various tasks, RBMs can be trained using either supervised or unsupervised learning. RBM achieved the state of art performance in collaborative filtering recently, in the context of Netflix Prize.

RBMs are two layer neural networks.The architecture is shown in Fig. 3.2. The first layer is the visible layer and the second is the hidden layer. In RBM, the restriction is that intra-layer communication is not allowed. Each visible node processes an input and makes a stochastic decision about

Symbols	Descriptions
L	Learner set
l_i	i-th learner.
C	Course set
c_j	j-th course.
A	Learner course interaction set
a_{ij}	Interaction between i-th learner and j-th course
K	The number of known topics
D_j	The duration of j-th course
d_j, d_{jk}	Dificulty of j-th course and difficulty of j-th course on k-th topic
α_{ij}	Completed ratio in a_{ij}
τ_{ij}	Learning duration in a_{ij}
θ_i, θ_{ik}	Capacity of l_i on k-th topic
l_i', c_j'	Learner embedding and course embedding excluding capacity and difficulty.
X_{ij}	Item response of subject i on object j
p_{ij}	Probability of learner i having mastered course j
q_{ij}	Ideal probability obtained by MIRT model
$\theta_{i,t}, \theta_{i,t,k}$	Capacity of l_i at time t and capacity of l_i at time t on topic k
$c_t, d_{t,k}$	Followed course at time t and difficulty of c_t on topic k.

Table 3.1: All symbols and descriptions

whether to transmit the input or not. Due to the restriction, RBM has the following properties : The activation conditions of each hidden layer neurons are independent, given the state of the visible layer neurons and vice versa.

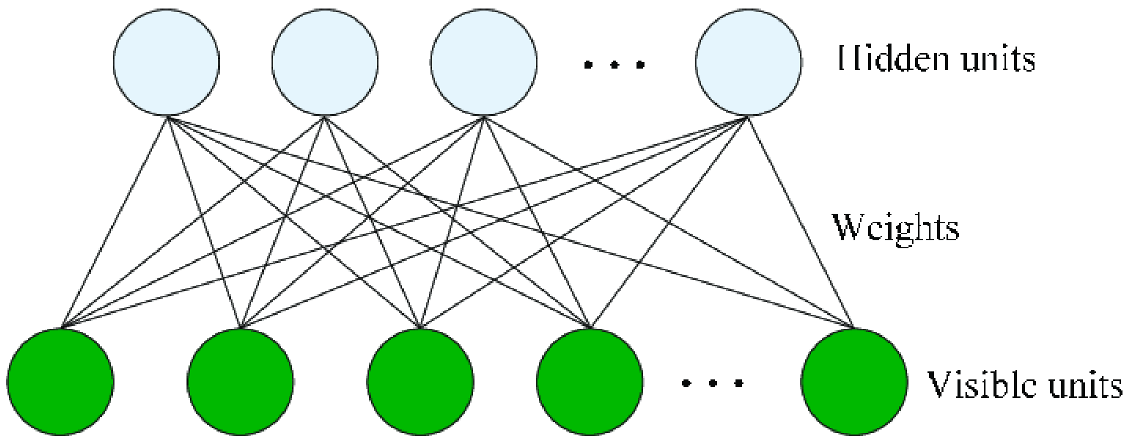

Figure 3.2: RBM architecture.

RBM works in two phases called Feed Forward Pass and Feed Backward Pass. In the Feed Forward Pass, the positive and negative associations between visible units and hidden units are identified. In the Feed Backward Pass, the input layer is reconstructed through the activated hidden neurons. After reconstruction, the error and weight change are calculated.

- Error = Reconstructed input - Actual input
- Weight change = Input * Error * Learning rate

After these two phases, the pattern that is responsible for the activation of hidden neurons can be identified. RBMs are energy based models and maximum likelihood is the learning goal. The Energy function of RBM is defined by Equation 3.5.

$$E(v,h;\theta) = -\sum_i b_i v_i - \sum_j a_j h_j - \sum_{ij} v_i h_j w_{ij} \tag{3.5}$$

In Equation 3.5, θ represents the RBM parameters { W, a, h }. W is the weight vector, a and b are biases of hidden and visible layer respectively. The joint probability of v and h can be obtained using Equation 3.6.

$$P_\theta(v,h) = \frac{1}{Z(\theta)} \exp\left(-E(v,h;\theta)\right) \tag{3.6}$$

$Z(\theta)$is the partition function which is given by Equation 3.7.

$$Z(\theta) = \sum_{v} \sum_{h} \exp\left(-E(v, h; \theta)\right) \tag{3.7}$$

By using the edge distribution of P(v, h), P(v) can be calculated as shown in Equation 3.8.

$$P_{\theta}(v) = \frac{1}{Z(\theta)} \sum_{h} \exp\left[v^T W h + a^T h + b^T v\right] \tag{3.8}$$

The maximum likelihood estimation method is used for parameter estimation. RBM parameters are obtained by maximizing P(v) as shown in Equation 3.9.

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} log P_{\theta}(v^{(n)}) \tag{3.9}$$

To find the extreme value solution the gradient decent algorithm can be used. Through Monte Carlo simulation the gradient is approximated.

$$\Delta a_i = v_i^{(0)} - v_i^{(k)}$$
$$\Delta b_i = P(h_j = 1|v^{(0)}) - P(h_j = 1|v^{(k)})$$
$$\Delta W_{ij} = P(h_j = 1|v^{(0)})v_i^{(0)} - P(h_j = 1|v^{(k)})v_i^{(k)}$$

where $v_i^{(0)}$ is the sample value and $v_i^{(k)}$ is the sample that satisfies the distribution P(v).

Now the parameters are updated as,

$$a_i = a_i + \Delta a_i$$
$$b_j = b_j + \Delta b_j$$
$$w_{ij} = W_{ij} + \Delta W_{ij}$$

In this work, the RBM is trained using Fast Gibbs Sampling algorithm[37]. This algorithm uses an accelerated weight called fastW and adjustment coefficient ζ. The accelerated weight is updated just like the traditional weight. The sum of the traditional and accelerated weights form the weight of the entire network. This ensures rapid increase in weight update in the early stage of training. The purpose of adjustment coefficient is to change the rate of accelerated weight update. The value of ζ ranges from 0 to 1.

At first, the input data v is sampled to obtain the hidden layer data.

$$b^{+} = P(h|v^{+}, W)$$
$$h^{-} = P(h|v^{-}, W + fastW)$$

v^{+} and v^{-} represent input data and $v^{+} = v^{-}$
After calculating the hidden layer data, positive and negative gradients are updated.

$$W^{+} = {v^{+}}^{T} h^{+}$$
$$W^{-} = {v^{-}}^{T} h^{-}$$

Now using the hidden layer data new input layer data is calculated.

$$v^{-} = P(v|h^{-}, W + fastW)$$

The gradient for weight updates is calculated as

$$\Delta W = W^{+} + W^{-}$$

The traditional and accelerated weight updates are done as follows

$$W = W + \Delta W$$
$$fastW = \zeta * fastW + \Delta W$$

Algorithm 1: FGS algorithm for RBM training.

Input: RBM **v, h**, training batch S
Output: gradient approximation $\Delta\mathbf{W}, \Delta\mathbf{a}_i, \Delta\mathbf{b}_j i = 1, ..., m, j = 1, ..., n$
1 : $Init\ \Delta\mathbf{W} = \Delta\mathbf{a}_i = \Delta\mathbf{b}_j = 0$
2: Initialize **W** to small regular values and fast**W** to all zeros
3: **For all v** ϵ S do
4: $\mathbf{v}^{(0)} \leftarrow \mathbf{v}$
5: **for** t = 0, . . ., k-1 do
6: **for** j = 1, . . ., n do sample
7: ${\mathbf{h}_{+}}^{(t)} \sim p(\mathbf{h}\ |\mathbf{v}^{(t)}\ , \mathbf{W})$,calculate: $\mathbf{W}^{+}\ = \mathbf{v}^{(0)T}\mathbf{h}_{+}$
8: **end**
9: **for** j = 1, . . ., n do sample
10: ${\mathbf{h}_{-}}^{(t)} \sim p(\mathbf{h}\ |\mathbf{v}^{(t)}\ , \mathbf{W} + \text{fast}\mathbf{W})$,calculate: $\mathbf{W}^{-}\ = \mathbf{v}^{(0)T}\mathbf{h}_{-}$
11: **end**
12: **for** i = 1, . . ., m do sample
13: ${\mathbf{v}_i}^{(t+1)} \sim p(\mathbf{v}_i\ |\ \mathbf{h}^{(t)}\ , \mathbf{W} + \text{fast}\mathbf{W})$
14: **end**

for j = 1, . . ., n, i = 1, . . ., m do
$\Delta\mathbf{W} \leftarrow \Delta\mathbf{W} + p(\mathbf{h}_j = 1 \mid \mathbf{v}^{(0)}) * \mathbf{v}_i{}^{(0)} - p(\mathbf{h}_j = 1 \mid \mathbf{v}^k) * \mathbf{v}_i{}^k$
$\mathbf{W} = \mathbf{W} + \Delta\mathbf{W}$
$\text{fast}\mathbf{W} = \zeta * \text{fast}\mathbf{W} + \Delta\mathbf{W}$
end
end
for i = 1, . . ., m do
$\Delta\mathbf{a}_i \leftarrow \Delta\mathbf{a}_i + \mathbf{v}_i^{(0)} - \mathbf{v}_i^{(k)}$
end
for j = 1, . . ., n do
$\Delta\mathbf{b}_j \leftarrow \Delta\mathbf{b}_j + p(\mathbf{h}_j = 1 \mid \mathbf{v}^{(0)}) - p(\mathbf{h}_j = 1 \mid \mathbf{v}^k)$
end
End

Chapter 4

Experimental Results and Discussions

4.1 Experimental Data

The Open University Learning Analytics dataset[31] is used as the base experimental data. The dataset contains information about courses, students and their interactions with virtual learning environment for twenty two selected courses. Presentation of courses start in February and October and are marked as 'B' and 'J' respectively. The dataset consists of tables connected using unique identifiers. The database schema is shown in Fig. 4.1.
The database consists of the following seven tables kept as csv files.
courses.csv
This file contains the list of all available courses and their presentations. The columns are :

- *code_module* : code name of the module that serves as the identifier.
- *code_presentation* : code name of the presentation. It consists of the course starting year and 'B' for presentation beginning in February and 'J' for that starting in October.
- length : Number of days needed to complete the course.

The structure of B and J presentations differ so they are analysed separately.
assessment.csv
This file contains information about assessments in module presentations.Each presentation has a set of assessments which are followed by the final examination. The columns of this file are :

- *code_module* : identification code of the module, to which the assessment belongs.
- *code_presentation* : identification code of the presentation, to which the assessment belongs.

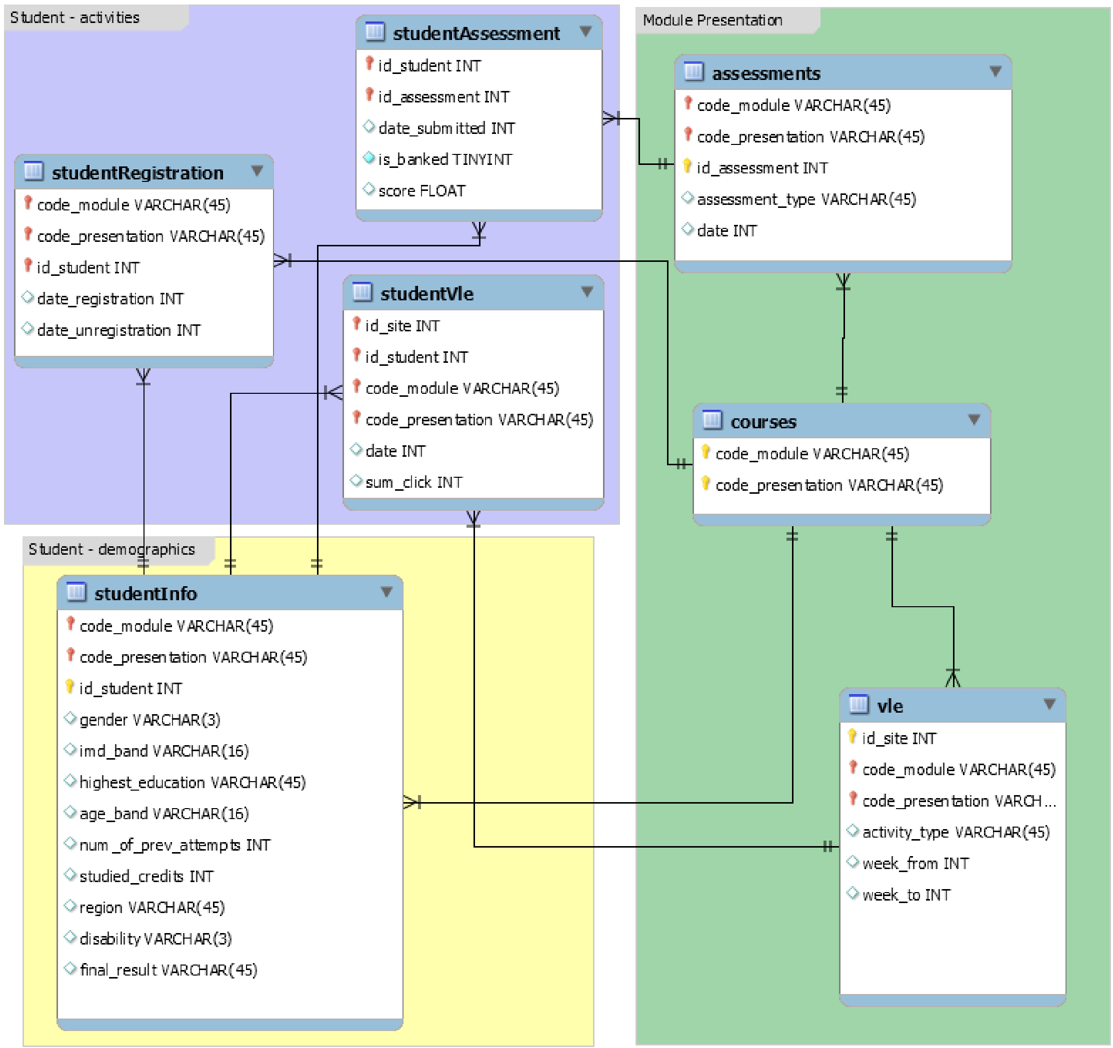

Figure 4.1: Schema of Open University Learning Analytics Dataset

- *id_assessment* : identification number of the assessment.
- *assessment_type* : type of assessment.Three types of assessments exist : Tutor Marked Assessment (TMA), Computer Marked Assessment (CMA) and Final Exam (Exam).
- date : the final submission date of the assessment estimated as the number of days since the beginning of the presentation. This begins from 0 (zero).
- weight : the weight of the assessment is expressed as %. Each exam

has a maximum weightage of 100%. The sum of all other assessments is 100%.

If the information about the final exam date is missing, it is at the end of the last presentation week.

vle.csv

This file contains information about the available learning materials in virtual learning environment. These are typically html pages, pdf files etc. The students can access these materials and their interactions are recorded. The columns in vle.csv are :

- *id_site* : an identification number of the material.
- *code_module* : an identification code for module.
- *code_presentation* : the identification code of presentation.
- *activity_type* : the role associated with the module material.
- *week_from* : the week from which the material is planned to be used.
- *week_to* : week until which the material is planned to be used.

studentInfo.csv

This file contains demographic information about the students along with their results. The columns are :

- *code_module* : identification code for a module in which the student is registered.
- *code_presentation* : the identification code of the presentation during which the student is registered on the module.
- *id_student* : a unique identification number for the student.
- gender : the student's gender.
- region : The geographic region where the student lives.
- *highest_education* : the highest educational qualification of the student on enrollment to the module presentation.
- *imd_band* : indicates the index of multiple depravation band of the place where the student lived during the module presentation.

- *age_band* : band of the student's age.
- *numof_prev_attempts* : the number times the student has attempted this module.
- *studied_credits* : the sum of credits obtained by the student for the currently attending modules.
- disability : This is a flag that indicates whether the student has some declared disability.
- *final_result* : student's final result in the module presentation.

studentRegistration.csv
This csv file contains information about the course registration details of the student. It has five columns.

- *code_module* : identification code for the module.
- *code_presentation* : the identification code of the presentation.
- *id_student* : unique identification number of the student.
- *date_registration* : The date on which the student registered for the course, this is measured relative to the beginning of the presentation.
- *date_unregistration* : date of student unregistration from the module presentation, this is the number of days measured relative to the start of the module presentation. Students, who completed the course have this field empty. Students who unregistered have 'withdrawal' as the value of the *final_result* column in the studentInfo.csv file.

studentAssessment.csv
This file contains results of the student's assessments. No result is recorded if the assessment is not submitted. The columns of this file are :

- *id_assessment* : identification number of the assessment.
- *id_student* : unique identification number for the student. item *date_submitted* : the assessment submission date of the student, relative to the start of the module.

- *is_banked* : a status flag indicating that the assessment result has been transferred from a previous presentation.
- score : the student's score in this assessment. This ranges from 0 to 100. A score below 40 is interpreted as Fail. The marks are in the range from 0 to 100.

studentVle.csv
This file shows the students interaction with materials available in the VLE. The columns are :

- *code_module* : identification code for a module.
- *code_presentation* : identification code of module presentation.
- *id_student* : unique identification number of student.
- *id_site* : identification number for VLE material.
- date: the date of student's interaction with the learning material. It is measured as the number of days since the start of module presentation.
- *sum_click* : the number of times the student interacts with the material in that day.

4.2 Settings

From the dataset, 80 % of the learner-course interactions are selected as the training set, and the remaining is the testing set. The overall distribution of learner-course interactions in Open University Analytic Dataset is shown in Fig. 4.1. The evaluation metrics selected are Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE),Precision and Recall as shown in Equations 4.1, 4.2, 4.3, 4.4 and 4.5.

$$MAE = \frac{\sum_{i=1}^{N} |pr_i - ar_i|}{N} \tag{4.1}$$

$$MSE = \frac{\sum_{i=1}^{N} (pr_i - ar_i)^2}{N} \tag{4.2}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (pr_i - ar_i)^2}{N}} \tag{4.3}$$

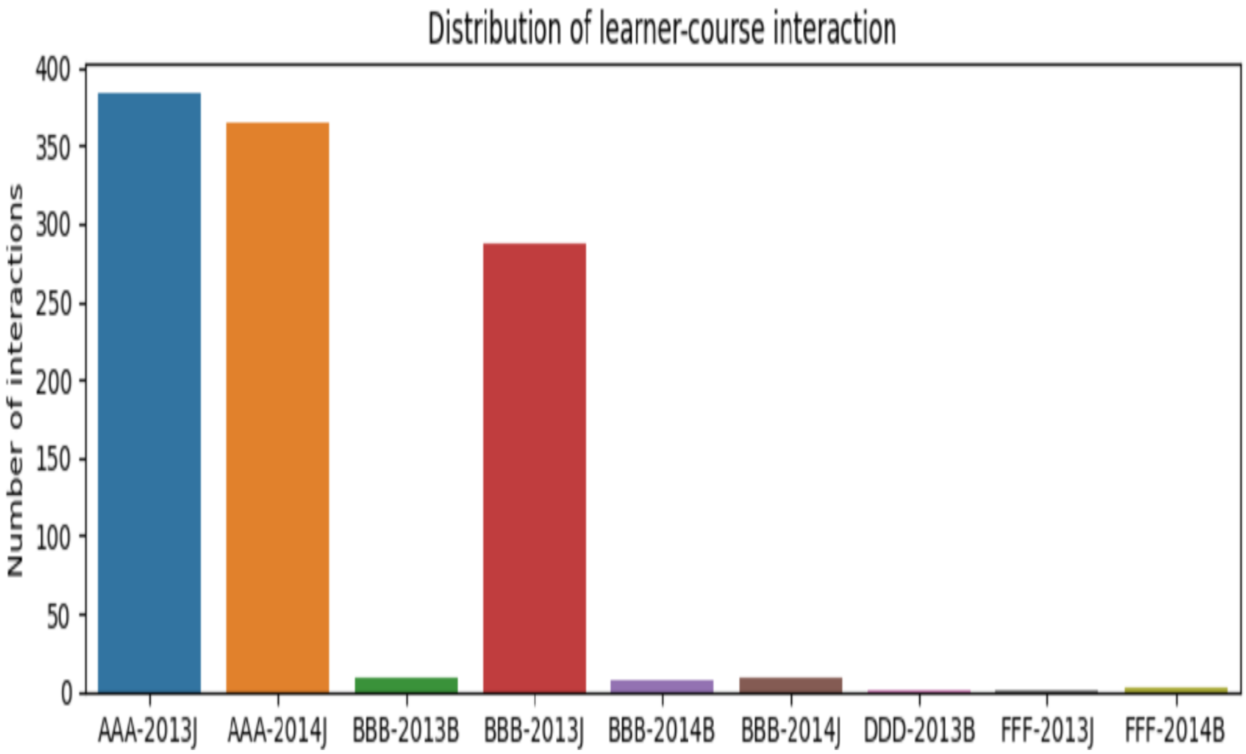

Figure 4.2: Learner-course interaction in Open University Analytic Dataset

$$Precision = \frac{|relevant\ items\ recommended|}{|items\ in\ the\ recommendation\ list|} \tag{4.4}$$

$$Recall = \frac{|relevant\ items\ recommended|}{|relevant\ items|} \tag{4.5}$$

4.3 Performance Comparison

To evaluate the effectiveness of the proposed course recommendation system, the capacity tracing-enhanced Neural Collaborative Filtering System (CT-E NCF) [1] is selected as the baseline model. In CT-E NCF, each learner and course are represented by fixed-length vectors.The interaction between learner and course are obtained by both element wise product and multi-layer perceptron on the learner and the course vector. The baseline and proposed models are trained on the training set and evaluated on the testing set. Figure 4.3. shows a comparison of both models with respect to the observed loss. The performance comparison is shown in Table 4.1. This

Metric	Base Model	Proposed Model
MSE	0.00306	0.00023
MAE	0.03479	0.01209
RMSE	0.05531	0.01506
Precision	66.667	80.0
Recall	33.333	66.667

Table 4.1: Performance comparison

shows that, the course recommendation system implemented using Restricted Boltzmann Machine has better performance in terms of all the five chosen metrics.

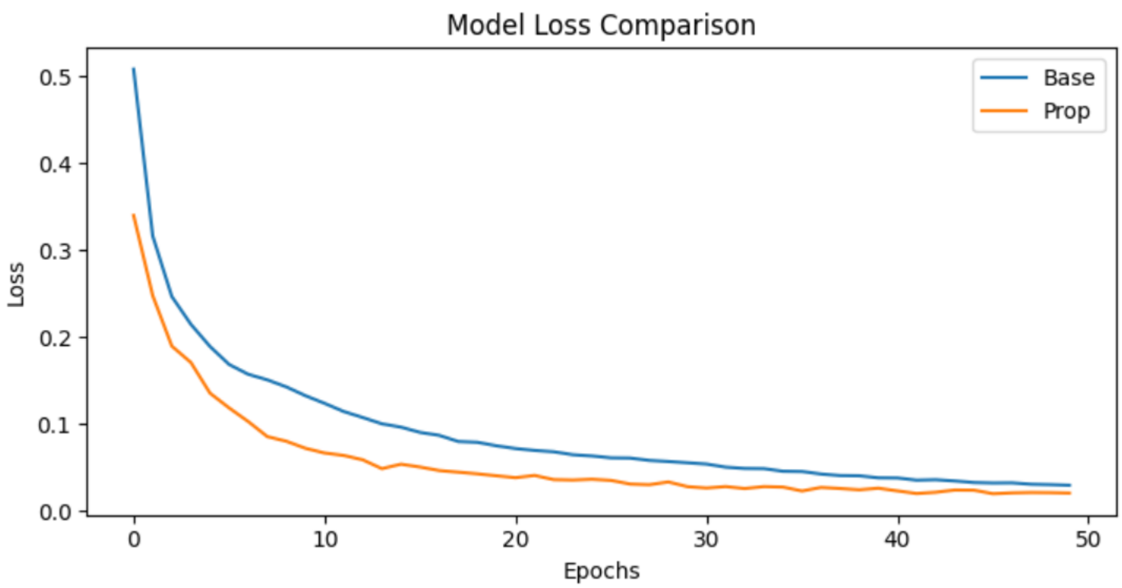

Figure 4.3: Loss Comparison

Chapter 5

Conclusion and Future Work

Course recommendation systems help students select courses of their interest and ability. The goal of this project is to provide course recommendation to learners based on their learning capacity so as to reduce the dropout rate. The learner capacity is estimated using Multi-dimensional Item Response Theory (MIRT), which is a variant of cognitive diagnosis. The capacity estimated by MIRT is treated as learner attribute and integrated into collaborative filtering framework. The collaborative filtering system is implemented using Restricted Boltzmann Machine, trained using Fast Gibbs Sampling algorithm. Extensive experiments were conducted on a real world dataset and the performance of the proposed system is compared with that of Capacity Tracing-Enhanced Neural Collaborative Filtering model chosen as baseline. The result obtained on the selected metrics: Precision, Recall, Mean Absolute Error, Mean Squared Error and Root Mean Square Error, shows that the proposed system has better performance.

Apart from the apparent future topics of improving accuracy or efficiency by trying and refining new algorithms, an idea called knowledge tracing[28] can be integrated into MIRT to facilitate dynamic updation of learner capacity in time dimension.

References

[1] Xuetao Tian and Feng Liu "Capacity Tracing-Enhanced Course Recommendation in MOOCs" IEEE Trans. Learn. Technol., vol: 14, Jun. 2021, doi: 10.1109/TLT.2021.3083180

[2] J. Zhang et al., "Hierarchical reinforcement learning for courserecommendation in MOOCs," in Proc. 33th AAAI Conf. Artificial Intelligence, Honolulu, HI, USA, Jan. 27 – Feb. 1, 2019, pp. 435–442, doi:10.1609/aaai.v33i01.3301435.

[3] F. Dalipi, A. S. Imran, and Z. Kastrati, "MOOC dropout prediction using machine learning techniques: Review and research challenges," in Proc. IEEE Global Engineering Education Conf., Tenerife, Spain, Apr. 17–20, 2018, pp. 1007–1014, doi: 10.1109/EDUCON.2018.8363340.

[4] Z. K. Papamitsiou and A. A. Economides, "Motivating students in collaborative activities with game-theoretic group recommendations," IEEE Trans. Learn. Technol., vol. 13, no. 2, pp. 374–386, Aug. 2020, doi:10.1109/TLT.2018.2869582.

[5] J. Cárdenas-Cobo, A. Puris, P. Novoa-Hernández, J. A. Galindo, and D.Benavides, "Recommender systems and Scratch: An integrated approach for enhancing computer programming learning," IEEE Trans. Learn.Technol., vol. 13, no. 2, pp. 387–403, Oct. 2020, doi: 10.1109/TLT.2019.2901457.

[6] Y. Pang, Y. Jin, Y. Zhang, and T. Zhu, "Collaborative filtering recommendation for MOOC application," Comput. Appl. Eng. Educ., vol. 25, no. 1, pp. 120–128, Jan. 2017, doi: 10.1002/cae.21785.

[7] Y. Pang et al., "Adaptive recommendation for MOOC with collaborative filtering and time series," Comput. Appl. Eng. Educ., vol. 26, no. 6, pp.2071–2083, Jun. 2018, doi: 10.1002/cae.21995.

[8] H. Zhang, H. Yang, T. Huang, and G. Zhan, "DBNCF: Personalized courses recommendation system based on DBN in MOOC environ-

ment," in Proc. Int. Symp. Educational Technology, Hong Kong, China, Jun. 27–29, 2017, pp. 106–108, doi: 10.1109/ISET.2017.33.

[9] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," ACM Comput. Surv., vol. 47, no. 1, pp. 1–45, Jul. 2014, doi: 10.1145/2556270.

[10] L. V. DiBello, L. A. Roussos, and W. Stout, "Review of cognitively diagnostic assessment and a summary of psychometric models," in Handbook of Statistics. American Elsevier, New York, NY, USA, 2006, vol. 26, pp. 979–1030, doi: 10.1016/S0169-7161(06)26031-0.

[11] M. C. Polson and R. J. Jeffrey, Foundations of Intelligent Tutoring Systems. Mahwah, NJ, USA: Erlbaum, 1988, doi: 10.4324/9780203761557.

[12] D. Selent, T. Patikorn, and N. Heffernan, "ASSISTments dataset from multiple randomized controlled experiments," in Proc. 3rd ACM Conf. Learning, Edinburgh, U.K., Apr. 25–26, 2016, pp. 181–184, doi: 10.1145/2876034.2893409.

[13] T. Zhu et al., "Cognitive diagnosis based personalized question recommendation," Chin. J. Comput., vol. 40, no. 1, pp. 176–191, Jan. 2017, doi: 10.11897/SP.J.1016.2017.00176

[14] Q. Liu et al., "Exploiting cognitive structure for adaptive learning," in Proc. 25th ACM SIGKDD Conf. Knowledge Discovery and Data Mining, Anchorage, AK, USA, Aug. 4–8, 2019, pp. 627–635, doi: 10.1145/3292500.3330922.

[15] P. Michlik and M. Bielikova, "Exercises recommending for limited time learning," Procedia Comput. Sci., vol. 1, no. 2, pp. 2821–2828, Dec. 2010, doi: 10.1016/j.procs.2010.08.007.

[16] Y. Chen et al., "Tracking knowledge proficiency of students with educational priors," in Proc. 26th ACM Conf. Information and Knowledge Management, Singapore, Nov. 6–10, 2017, pp. 989–998, doi: 10.1145/3132847.3132929.

[17] C. Jiang et al., "Personalized exercises recommendation algorithm based on knowledge hierarchical graph," Comput. Eng. & Appl., vol.

54, no. 10, pp. 229–235, Jun. 2018, doi: 10.3778/j.issn.1002-8331.1612-0019.

[18] L. Fang, L. A. Tuan, S. C. Hui, and L. Wu, "Personalized question recommendation for English grammar learning," Expert Syst., vol. 35, no. 2, p. e12244, Nov. 2018, doi: 10.1111/exsy.12244.

[19] S. E. Whitely, "Measuring aptitude processes with multicomponent latent trait models," J. Educ. Meas., vol. 18, no. 2, pp. 67–84, Sep. 2005, doi: 10.1111/j.1745-3984.1981.tb00843.x.

[20] K. K. Tatsuoka, "Rule space: An approach for dealing with misconceptions based on item response theory," J. Educ. Meas., vol. 20, no. 4, pp. 345–354, Sep. 2005, doi: 10.1111/j.1745-3984.1983.tb00212.x.

[21] B. W. Junker and K. Sijtsma, "Cognitive assessment models with few assumptions, and connections with nonparametric item response theory," Appl. Psychol. Meas., vol. 25, no. 3, pp. 258–272, Sep. 2001, doi: 10.1177/01466210122032064.

[22] J. P. Leighton and M. J. G. S. M. Hunka, "The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach," J. Educ. Meas., vol. 41, no. 3, pp. 205–237, Sep. 2004, doi: 10.1111/j.1745-3984.2004.tb01163.x.

[23] L. Pan, X. Wang, C. Li, J. Li, and J. Tang, "Course concept extraction in MOOCs via embedding-based graph propagation," in Proc. 8th Int. Joint Conf. Natural Language Processing, Taipei, Taiwan, Nov. 27 – Dec. 1, 2017, pp. 875–884.

[24] X. Fan, "Item response theory and classical test theory: An empirical comparison of their item/person statistics," Educ. Psychol. Meas., vol. 58, no. 3, pp. 357–381, Jun. 1998, doi: 10.1177/0013164498058003001.

[25] S. A. Mulaik, "A mathematical investigation of some multidimensional Rasch models for psychological tests," in Proc. Annu. Meeting Psychometrics Society, Princeton, NJ, USA, 1972.

[26] J. B. Sympson, "A model for testing with multidimensional items," in Proc. Computerized Adaptive Testing Conf., Minneapolis, MN, USA, Jan, 7, 1978.

[27] R. L. McKinley and M. D. Reckase, "The use of the general Rasch model with multidimensional item response data," The American College Testing Program, Resident Programs Department, Iowa City, IA, USA, Res. Rep. ONR 82-1, Aug. 1982.

[28] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," User Model. & User-Adapted Interact., vol. 4, no. 4, pp. 253–278, Dec. 1994, doi: 10.1007/BF01099821.

[29] B. S. Bloom, "Learning for mastery," Eval. Comment, vol. 1, no. 2, pp. 1–12, Jan. 1968.

[30] C. Piech et al., "Deep knowledge tracing," in Proc. 28th Int. Conf. Neural Information Processing Systems, Montreal, Canada, Dec. 7–12, 2015, pp. 505–513.

[31] https://analyse.kmi.open.ac.uk/opendataset

[32] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," IEEE Comput., vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.s

[33] X. He et al., "Neural collaborative filtering," in Proc. 26th Int. World Wide Web Conf., Perth, Australia, Apr. 3–7, 2017, pp. 173–182, doi:10.1145/3038912.3052569.

[34] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in Proc. 4th Int. Conf. Learning Representations, San Juan, Puerto Rico, May 2–4, 2016.

[35] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in Proc. 25th Conf. Uncertainty in Artificial Intelligence, Montreal, Canada, Jun. 18–21, 2009, pp. 452–461.

[36] Bhaskar Mondal, Om Patra, Sanket Mishra and Priyadarsan Patra, "A course recommendation system based on grades", in 2020 International conference on Computer Science, Engineering and Applications, March 2020, doi:10.1109/ICCSEA49143.2020.9132845.

[37] Qianglong Wang, Xiaoguang Gao, Kaifangwan, Fei Li and Zijian Hu, "A Novel Restricted Boltzmann Machine Training Algorithm with Fast Gibbs Sampling Policy", Hindawi, Mathematical Problems in Engineering, volume 2020,Article ID4206457.