Session-based Course Recommendation Frameworks using Deep Learning

Md Akib Zabed Khan
Florida International University
mkhan149@fiu.edu

Agoritsa Polyzou
Florida International University
apolyzou@fiu.edu

ABSTRACT

Academic advising plays an important role in students' decision-making in higher education. Data-driven methods provide useful recommendations to students to help them with degree completion. Several course recommendation models have been proposed in the literature to recommend courses for the next semester. One aspect of the data that has yet to be explored is the suitability of the recommended courses taken together in a semester. Students may face more difficulty coping with the workload of courses if there is no relationship among courses taken within a semester. To address this problem, we propose to employ session-based approaches to recommend a set of courses for the next semester. In particular, we test two session-based recommendation models, CourseBEACON and CourseDREAM. Our experimental evaluation shows that session-based methods outperform existing popularity-based, sequential, and non-sequential recommendation approaches. Accurate course recommendation can lead to better student advising, which, in turn, can lead to better student performance, lower dropout rates, and better overall student experience and satisfaction.

Keywords

session-based recommendation, course recommendation, deep learning

1. INTRODUCTION

In higher education, one of the challenges that students face is identifying the proper set of courses to register for every semester so that they will successfully progress with their degree. While selecting courses, students consider different factors, like a balance between their personal preferences (interests, goals, and career aspirations) and the requirements of their degree programs. Students usually need to register for some elective courses. Some courses have prerequisites. These decisions are hard to make, and consequently, appropriate course selection is a non-trivial task for the students. Courses can be selected based on example guides provided by each department, but these are not adapted to individual cases [6]. Students may get personalized assistance from academic advisors. However, the ratio of students to advisors is very high, which limits discussion time between an advisor and an advisee [11]. Lack of communication may result in unfavorable situations where students fail to cope with the course workload and become frustrated. As a matter of fact, the dropout rates at the undergraduate level of US colleges are alarming [10].

Data mining techniques and machine learning models can draw insights from historical data records and generate course recommendations to assist with student advising. Collaborative filtering algorithms and content filtering methods are the most common approaches in this field of research. Existing work has explored the sequential nature of course enrollment data, the words associated with course description data, and non-sequential approaches to prioritize students' preferences and analyze their characteristics and skills to recommend courses for a semester or even multiple consecutive semesters. However, no prior study in the literature considers how well the courses would be suited to be taken together within a semester. Some courses are usually taken together if they cover complementary concepts. Also, it is not a good idea for a student to take some very heavy courses in the same semester. For example, if a student registers for three or four difficult courses in the same semester, that could lead to poor performance in some or all of them, as the student will not have enough time to study for all the courses. In the end, their semester grade point average (GPA) might be low compared to their efforts.

Students might be more likely to perform well if their courses are related and not so difficult to study altogether within a semester. The set of courses taken in a semester by past students can provide impactful insights to measure the correlation, relationship, and compatibility of a pair of courses. These notions can be captured by session-based recommendation approaches. We propose to adapt such approaches to rank courses not only based on their suitability but based on their synergy as well. There are long short-term dependencies in the sequence of courses taken semester-by-semester, and we can capture them by using deep learning models.

We explore two different models that capture these dependencies. First, we propose CourseBEACON, where we calculate the co-occurrence rate between a pair of courses to

M. A. Z. Khan and A. Polyzou. Session-based course recommendation frameworks using deep learning. In M. Feng, T. Käser, and P. Talukdar, editors, *Proceedings of the 16th International Conference on Educational Data Mining*, pages 269–277, Bengaluru, India, July 2023. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.8115699

capture and estimate their relationship. Then, we incorporate this notion of course compatibility into sequential deep learning models (recurrent neural networks) to perform the recommendation task. Second, in CourseDREAM, we create latent vector representation for each basket of courses taken in a semester which is helpful to capture the courses that are historically considered to be suitable to take within a semester. Then, we use recurrent neural networks (RNNs) to capture the sequential transition of courses over the sequence. Using real historical data, we evaluate these proposed approaches. They outperform other baselines or existing state-of-the-art approaches we examined. Our course recommendation model will be impactful in academic advising to help students with decision-making and decrease the risk of dropout cases. The paper is organized as follows. Sect. 2 introduces the background, notation, and related work, while Sect. 3 delves deeper into the two proposed approaches. Sect. 4 presents our experimental setup in detail. Sect. 5 shows our results and Sect. 6 concludes the paper.

2. BACKGROUND

2.1 Problem

Some courses are needed to fulfill degree requirements, others are elective. Usually, it is up to the students to decide which courses to take within a semester and in which semester they will take any required courses. While selecting courses, students must remember degree requirements and several factors such as personal preferences, course prerequisites, career goals, and which courses are needed to build knowledge for future courses. Universities naturally collect information about the course registration history of students. Insightful patterns can be extracted by analyzing the historical information of past students to recommend courses to future students. *Course recommendation* is a systematic way to evaluate which courses are appropriate for a student with the goal of making advising easier. By inspecting the student-course interactions, and sequential transitions of courses over the semesters of past students, we recommend courses for other students by implementing various data-driven techniques.

2.2 Definition of Terms and Notations

Considering the terminology used in general recommendation literature, we can consider each student as a user, and each course as an item.

A **session** is a finite amount of time for a user to complete a set of activities together. In this paper, we consider a student's semester to be a session. A user can buy a set of items together in a session. A student can take a set of courses together within a semester. A basket is a similar notion to a session. In **session-based recommendation** models, we learn users' preferences from the sessions generated during the consumption process and pay increasing attention to recent sessions as users' preferences change and evolve dynamically. A session-based recommendation system recommends a set of items for the next session, for which we may or may not have some partial information (i.e., any items already present in the session). A **next-basket recommendation** is a special case when we generate a complete set of items for the next session (i.e., without any partial information). In this paper, we will recommend a complete set of courses for the next semester.

Table 1: Notations

$\mathcal{C}$, $\mathcal{S}$	set of courses and students, respectively
m, n	cardinality of $\mathcal{C}$, $\mathcal{S}$, $\lvert\mathcal{C}\rvert = m$ and $\lvert\mathcal{S}\rvert = n$
p, q	courses, $p, q \in \mathcal{C}$
u	student, $u \in \mathcal{S}$
i, j	index for student and course, respectively
t	index for semester
$\mathcal{B}_{i,t}$	set of courses i-th student took in semester t
H_i	course registration history of i-th student
t_i	total number of semesters for i-th student, $t_i = \lvert H_i\rvert$
k	total number of courses to recommend
$\mathbf{R}$	tensor $\mathbf{R} \in \mathbb{R}^{d\times d\times n}$ with d latent dimensions
$\mathbf{F}_{i,p,q}$	number of (i, p, q) triples for i-th student

We adopt the following notation. We will use calligraphic letters for sets, capital bold letters for matrices or tensors, and lowercase bold letters for vectors. $\mathcal{C}$ indicates the set of all courses ($|\mathcal{C}| = m$) and $\mathcal{S}$ denotes the set of all students ($|\mathcal{S}| = n$). $\mathcal{B}_{i,t}$ represents a set of courses that i-th student has taken in a semester t. H_i is the course registration history of the i-th student, $H_i = [\mathcal{B}_{i,1}, \mathcal{B}_{i,2}, ..., \mathcal{B}_{i,t_i}]$, where $t_i = |H_i|$ is the total number of semesters that the i-th student took courses. Additional notation is presented in Table 1. We will refer to the student and the semester we are trying to generate a recommendation for as the target student and semester, respectively.

2.3 Related Work

Researchers have analyzed different types of data to build course recommendation models using different techniques. In terms of *types of data*, many researchers used real-life course enrollment and course description datasets collected from universities' data warehouses [1, 8, 22, 23, 27, 32], and others used online datasets collected from different online course platforms [5, 19, 34]. Moreover, some of the researchers collected data by taking feedback from students conducting surveys [8, 22]. Some researchers considered the grades of students in each course as a useful feature to recommend courses that students were expected to perform well [8, 15, 17, 19, 32]. Others did not consider grades as an important feature [1, 21, 22, 23, 34]. Very few researchers considered the interests and skills of students to choose a course [14, 28] by collecting students' survey responses.

Different *course recommendation systems* have been proposed in the literature. Parameswaran et al. introduced the first course recommendation system based on constraint satisfaction [20]. Al-Badarenah et al. propose a collaborative filtering-based course selection system using a k-means clustering algorithm and an association rule mining method [1]. The Apriori algorithm (an association rule mining technique) has been used for a collaborative recommendation system for online courses [19]. There are some content-based filtering methods available in literature where researchers analyze course descriptions and course content to recommend courses [17, 18, 21, 22]. Pardos et al. propose a course2vec model (like word2vec model) for course recommendation using both course enrollment and course description data [22]. Students' preferences and student-course interactions are neglected in these methods. To prioritize students' preferences, a matrix factorization model has been proposed for course recommendation [29]. Pardos et al. propose a combination of long short-term memory (LSTM) networks and skip-gram models to recommend courses balancing explicit and implicit preferences of students [21]. RNN models have also been used to recommend courses that are expected to improve students' GPAs [16]. Shao et al. introduce a PLAN-BERT model to recommend multiple consecutive semesters toward degree completion [27]. Polyzou et al. present a random-walk-based approach, Scholars Walk, to capture sequential transitions of different courses semester-by-semester assuming that the past sequence of courses matters [23]. No prior study captures the relationship among courses taken in a semester considering each semester as a session to recommend correlated courses.

In commercial recommender systems, there are different types of *session-based recommendation systems* to recommend the next clicked item (next interaction), the next partial session (subsequent part) in the current session, and the next basket or complete session with respect to the previous sessions for a user [31]. For our problem setting, the last one is more appropriate. Next-basket recommendation approaches can be divided into two types: sequential and non-sequential approaches. Generally, sequential approaches capture the user-item interactions and sequential relationships of items, by constructing a transition matrix observing item transitions over sessions for a user. Rendle et al. introduced the first next-basket recommendation system by presenting a factorized personalized Markov chain (FPMC) model [26]. The FPMC model can capture the first-order dependency of items. Long short-term dependency of items over the sequence of baskets can be captured by recurrent neural networks. Yu et al. propose a dynamic recurrent basket model (DREAM) using LSTM networks [33]. Le et al. propose a correlation-sensitive next basket recommendation model named Beacon to recommend correlated items using transaction data of online grocery shops [13].

Non-sequential approaches prioritize users' preferences of items over the sequential transition of items. Matrix factorization and tensor decomposition techniques have been used to recommend the next item capturing users' preferences of choosing one item over another item [7, 25]. Wan et al. propose a representation learning model triple2vec to recommend complementary and compatible items in the next basket [30]. A tensor decomposition technique has been proposed to recommend complementary items in the next basket by using RESCAL decomposition [7].

In this paper, we explore both sequential and non-sequential approaches for session-based recommendation to recommend a set of synergistic courses for the next semester. Moreover, our course enrollment data is different from transaction data of items, i.e., one item can appear multiple times in different sessions in a sequence of baskets for a user, but one course is most likely to appear one time in a sequence of semesters for a student.

3. SESSION-BASED COURSE RECOMMENDATION

We propose to use a session-based, sequential course recommendation system to capture the synergy between courses taken in the same semester. Even though some courses might be worth equal credit hours, the required working time load varies based on the difficulty of subjects [4]. Students' course load can impact their performance [2]. A good combination of courses can balance the workload of courses. The courses well-suited to be taken together could cover similar topics, be correlated, or just not be overwhelming for the students. The influence of co-taken courses has been considered important for other educational tasks, i.e., grade prediction and designing an early warning system [3, 9, 24].

We analyze the relationship and correlation of courses by incorporating the concept of session-based recommendation (SBR) for the first time in course recommendation. We consider a set of courses taken in a semester as a session and inspect the session to understand the relationship among the courses. Let $\mathcal{B}_{i,t}$ be a set of courses of the i-th student at semester t. Given the courses for the first $t_i - 1$ semesters for the i-th student as input, our target is to recommend a set of correlated courses, $c_1, c_2..., c_k$ for the target semester t_i where k is the number of courses to be recommended. We extend two popular session-based models, the Beacon model [13] to CourseBEACON, and DREAM model [33] to CourseDREAM, for the purpose of course recommendation.

3.1 Assumptions

We make the following assumptions in the context of course recommendations in higher education.

1. Time is discrete and moves from one semester to the next.
2. The courses are ordered over the sequence of semesters, but there is no order in the courses within a semester.
3. Learning is sequential; each course taken in a semester provides some skills and knowledge that will help to learn future courses in the following semesters. So, the sequence of courses matters in course selection.
4. Course registration history of students might provide beneficial insight into the curriculum and degree requirements when sufficient domain experts are not available.
5. We know the number of courses the student will take in the target semester.

3.2 CourseBEACON

We adopt the Beacon model to CourseBEACON to generate correlation-sensitive course recommendations for next semester. The framework has three components: correlation-sensitive basket encoder, course basket sequence encoder, and correlation-sensitive score predictor. The basket encoder receives as input the sequence of courses taken in the previous semesters $[1, ..., t_i - 1]$ and the global correlation matrix, $\mathbf{M} \in \mathbb{R}^{m \times m}$, which captures the relationships among courses within a semester (basket). The encoder produces a sequence of basket representations as output for each prior basket of a student. We feed these representations into the course basket sequence encoder where LSTM networks capture the sequential association of courses over the sequence of semesters. Finally, we feed the output from the sequence encoder and the correlation matrix as inputs to the correlation-sensitive score predictor to produce the final scores for the candidate courses. We recommend the courses with the highest score for each student.

Building the Course Correlation Matrix We create the correlation matrix using all semesters available in training data. Let $\mathbf{F} \in \mathbb{R}^{m \times m}$ be the frequency matrix. For courses $p, q \in \mathcal{C}$, $\forall p \neq q$, we count $\mathbf{F}_{p,q}$, which is the number of times p, q co-occur together in a basket (i.e., taken in the same semester). We normalize $\mathbf{F}$ to generate the final correlation matrix $\mathbf{M}$ based on the Laplacian matrix $\mathbf{M} = \mathbf{D}^{-1/2}\mathbf{F}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ denotes the degree matrix, $D_{p,p} = \sum_{q \in \mathcal{C}} \mathbf{F}_{p,q}$ [12]. $\mathbf{F}$ and $\mathbf{M}$ are symmetric by definition.

Correlation-Sensitive Basket Encoder For each semester, we create a binary indicator vector for the set of courses that were taken by a student. We convert a basket of courses $\mathcal{B}_{i,t}$ to binary vector $\mathbf{b}_{i,t}$ of length m for the i-th student, where the j-th index is set to 1 if $c_j \in \mathcal{B}_{i,t}$; otherwise it is 0. We get an intermediate basket representation $\mathbf{z}_{i,t}$ for a basket $\mathcal{B}_{i,t}$ as follows:

$$\mathbf{z}_{i,t} = \mathbf{b}_{i,t} \odot \omega + \mathbf{b}_{i,t} * \mathbf{M}, \tag{1}$$

where ω is a learnable parameter that indicates the importance of the course basket representation, the circle-dot indicates element-wise product, and the asterisk indicates matrix multiplication. We feed $\mathbf{z}_{i,t}$ into a fully-connected layer and we get a latent basket representation $\mathbf{r}_{i,t}$ by applying the ReLU function in an element-wise manner:

$$\mathbf{r}_{i,t} = ReLU(\mathbf{z}_{i,t}\Phi + \phi), \tag{2}$$

where Φ, ϕ are the weight and bias parameters, respectively.

Course Basket Sequence Encoder We use the sequence of latent basket representations $\mathbf{r}_{i,t}, \forall t \in [1, \ldots, t_i - 1]$ for the i-th student as input in the sequence encoder. Each $\mathbf{r}_{i,t}$ is fed into an LSTM layer, along with the hidden output from the previous layer. We compute the hidden output $\mathbf{h}_{i,t}$ as:

$$\mathbf{h}_{i,t} = tanh(\mathbf{r}_{i,t}\Psi + \mathbf{h}_{\mathbf{i},(\mathbf{t}-\mathbf{1})}\mathbf{\Psi}' + \psi), \tag{3}$$

where Ψ, Ψ' and ψ are learnable weight and bias parameters.

Correlation-Sensitive Score Predictor We use the correlation matrix and the last hidden output as the input in the correlation-sensitive score predictor to derive a score for each candidate course. Let $\mathbf{h}_{i,t_i-1}$ be the last hidden output generated from the sequence encoder. First, we get a sequential signal $\mathbf{s}_i$ from the given sequence of baskets:

$$\mathbf{s}_i = \sigma(\mathbf{h}_{i,t_i-1}\mathbf{\Gamma}), \tag{4}$$

where $\mathbf{\Gamma}$ is a learnable weight matrix parameter and σ is the sigmoid function. Using the correlation matrix, we get the following predictor vector for the i-th student of length m:

$$\mathbf{y}_i = \alpha(\mathbf{s}_i \odot \omega + \mathbf{s}_i * \mathbf{M}) + (1 - \alpha)\mathbf{s}_i, \tag{5}$$

where $\alpha \in [0, 1]$ is a learnable parameter used to control the balance between intra-basket correlative and inter-basket sequential associations of courses. The j-th element of $\mathbf{y}_i$ indicates the recommendation score of course c_j to be in the target basket of i-th student.

3.3 CourseDREAM

We propose the Course Dynamic Recurrent Basket Model (CourseDREAM), based on DREAM [33], to recommend a set of courses for the target semester. We consider two different latent representation techniques, max pooling and average pooling, to create a representation of a semester of courses. We use long short-term memory networks (LSTM) to capture the sequential transition of courses over the sequence of semesters and a dynamic representation of students which captures the dynamic interests of a student throughout their studies. For prediction, we calculate the score for each course $\forall p \in \mathcal{C}$ based on the most recent basket's $\mathcal{B}_{i,(t_i-1)}$ dynamic representation. We recommend the courses with the highest scores for the target semester.

Latent Representation of Semester Each basket of the i-th student consists of one or more courses. The j-th course that i-th student took at semester t has the latent representation $\mathbf{c}_{i,j,t}$ with d latent dimensions. We create a latent vector representation $\mathbf{r}_{i,t}$ for the set of courses that the i-th student took in semester t by aggregating the vector representations of these courses, $\mathbf{c}_{i,j,t}$. We use two types of aggregation operations. First, in max pooling, we take the maximum value of every dimension over these vectors. The l-th element ($l \in [1, d]$) of $\mathbf{r}_{i,t}$ is created as:

$$\mathbf{r}_{i,t,l} = max(\mathbf{c}_{i,1,t,l}, \mathbf{c}_{i,2,t,l}, \ldots.), \tag{6}$$

where $\mathbf{c}_{i,j,t,l}$ is the l-th element of the course representation vector $\mathbf{c}_{i,j,t}$. Secondly, for the average pooling, we aggregate the courses' latent representations in semester t by taking the average value of every dimension, as follows:

$$\mathbf{r}_{i,t} = \frac{1}{|\mathcal{B}_{i,t}|} \sum_{j=1}^{|\mathcal{B}_{i,t}|} \mathbf{c}_{i,j,t}. \tag{7}$$

Next, these representations of the sequence of baskets are passed to the recurrent neural network (RNN) architecture.

Dynamic Representation of a Student We incorporate LSTM networks in the RNN architecture where the hidden layer $\mathbf{h}_{i,t}$ is the dynamic representation of i-th student at semester t. The recurrent connection weight matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$ is helpful to propagate sequential signals between two adjacent hidden states $\mathbf{h}_{i,t-1}$ and $\mathbf{h}_{i,t}$. We have a learnable transition matrix $\mathbf{T} \in \mathbb{R}^{t_m \times d}$ between the latent representation of basket $\mathbf{r}_{i,t}$ and a student's interests, where t_m is the maximum length of the sequence of baskets for any student. We compute the vector representation of the hidden layer as follows:

$$\mathbf{h}_{i,t} = f(\mathbf{T}\mathbf{r}_{i,t} + \mathbf{W}\mathbf{h}_{i,t-1}), \tag{8}$$

where $\mathbf{h}_{i,t-1}$ is the dynamic representation of the previous semester. $f(\cdot)$ is the sigmoid activation function, i.e., $f(x) = 1/(1 + e^{-x})$.

Score generation The model generates a score $\mathbf{y}_{i,t_i}$ for all available courses that the i-th student might take at target semester t_i by using the matrix $\mathbf{M}$ with the latent representation of all courses and the dynamic representation of the student $\mathbf{h}_{i,t}$ as follows:

$$\mathbf{y}_{i,t_i} = \mathbf{M}^T\mathbf{h}_{i,t}, \tag{9}$$

where the j-th element of $\mathbf{y}_{i,t_i}$, represents the recommendation score for the j-th course.

Table 2: Data statistics

	# students	# courses	# target baskets
Training	2973	618	14070
Validation	1231	540	2743
Test	657	494	1259

4. EXPERIMENTAL EVALUATION

4.1 Dataset

We used a real-world dataset from Florida International University, a public US university, that spans 9 years. Our dataset consists of the course registration history of undergraduate students in the Computer Science department. The grades follow the A–F grading scale (A, A-, B+, B, B-, C+, C, D, F). We only consider the data of students who have successfully graduated with a degree. We remove instances in which a grade less than C was earned because these do not (usually) count towards degree requirements [17]. We also remove an instance if a student drops a class in the middle of the semester. In this way, we keep course sequences and information that at least lead to successful graduation and may be considered good examples. We remove courses that appear less than three times in our dataset.

After preprocessing, we have the course registration history of 3328 students and there are 647 unique courses. We split the data into train, validation, and test set. We use the last three semesters (summer 2021, fall 2021, and spring 2022) for testing purposes and the previous 3 semesters (summer 2020, fall 2020, and spring 2021) for validation and model selection. The rest of the data before the summer 2020 term (almost seven years' course registration history) are kept in the training set.

From the validation and test sets, we remove the courses that do not appear in the training set. We also remove any instances from the training, validation, and test set where the length of the basket sequence is less than three for a student. In other words, to generate recommendations we need the history of at least two previous semesters. The statistics of training, validation, and test data are presented in Table 2. Each student might correspond to multiple instances, one for each semester that could be considered as a target semester. For example, if a student took courses from fall 2019 until Spring 2021, they are considered in two instances on the test set (we generate recommendations from summer 2020 and spring 2021) and with three recommendations in the validation (target semesters: summer 2020, fall 2020, and spring 2021).

4.2 Evaluation metrics

As in prior work [13, 21, 23, 33], we used **Recall@k** score as our main evaluation metric, where k is the number of courses that the student took at the target semester.

$$\text{Recall@}k = \frac{\text{\# of relevant recommendations}}{\text{\# of actual courses in target semester}} \quad (10)$$

We essentially calculate the fraction of courses in the target semester that we correctly recommended to a student. In the subsequent sections, we report the average score over all the recommendations, i.e., target baskets. Recall and precision scores are equal as we recommend as many courses as the target student will take in the target semester. We also calculate the percentage of students for who we offer at least one relevant recommendation (**%rel**) as:

$$\%\text{rel} = \frac{\text{\# instances with} \geq 1 \text{ relevant recommendation}}{\text{\# total instances}} \quad (11)$$

This metric captures the ability of our recommendations to retrieve at least one relevant course for each student.

4.3 Experimental setting

We use the training set to build the models, and we select the parameters of the model with the best performance, based on the Recall@k metric, on the validation set. For the model selected, we calculate the evaluation metrics on the test set.

For the CourseBEACON model, for parameter α, we have tested the values [0.1, 0.3, 0.5, 0.7, 0.9]. α balances the importance of intra-basket correlation and inter-basket sequential association of courses. The lower value of α prioritizes the sequential association more than the intra-basket correlation; a higher value prioritizes the correlation of courses within the basket more. We also examined embedding dimensions=[16, 32, 64], hidden units=[32, 64, 128] of LSTM networks, and dropout rates=[0.3, 0.4]. For the CourseDREAM model, we used both max pooling and average pooling, however, the outcomes were very similar. In this paper, the results are reported with the average pooling technique. We tried LSTM layers=[1, 2, 3], embedding dimensions=[8, 16, 32], and dropout rates=[0.3, 0.4, 0.5, 0.6] for the CourseDREAM model.

4.4 Competing approaches

4.4.1 *Non-sequential baseline*

We use a popularity-based approach as a non-sequential baseline for course recommendation. Incorporating the hashing technique, we create a dictionary for each semester, starting from the first semester of each student, and count how many students take course $p \in \mathcal{C}$ in that semester of their studies. The top courses with the highest frequency at the t-th semester are recommended for the t-th semester of a target student. For example, if we have a student, and the target semester is his fourth semester in the program, we will recommend the most popular courses that students in the training set take in their fourth semester.

4.4.2 *Sequential baseline*

We use a popular sequence-based approach as our sequential baseline for course recommendation. For each pair of courses $\forall p, q \in \mathcal{C}$, we check how many times students took course q after course p. We create a bipartite graph with pairs of courses on consecutive semesters available in the training data. Courses are nodes and the weight of a connecting edge increases by 1 if one course comes before another course (i.e., $\text{weight}(p, q) \mathrel{+}= 1$ if course p is taken just before course q by a student). We normalize the weights as follows:

$$\text{weight}(p, q) = \frac{\text{weight}(p, q)}{\sum_{\forall r \in \mathcal{C}} \text{weight}(p, r)} \quad (12)$$

Given the courses that a target student took the previous semester, $\mathcal{B}_{i,t_i-1}$, we can recommend a set of courses for the t_i-th (target) semester. The recommendation score of a course is measured based on the summation of the weights from all the courses of the previous semester to this course in our created bipartite graph.

4.4.3 *Tensor decomposition*

We re-implement the session-based tensor decomposition technique [7] to recommend courses for the upcoming semester. This model prioritizes users' preferences of items in a basket over the sequential nature of items in the basket sequence of users. Using the training data, we create tensor $\mathbf{F} \in \mathbb{R}^{n \times m \times m}$, where $\mathbf{F}_{i,p,q}$ stores the number of times that courses p, q are taken together in the same semester from the i-th student. This tensor is very sparse, as there are many pairs of courses p, q that are not taken together by each student.

So, we use the RESCAL tensor decomposition technique with factorization rank, d, to get the approximate value of $\mathbf{F}_{i,p,q}$. We calculate the matrix $\mathbf{A}$ (course embedding, $\mathbf{A} \in \mathbb{R}^{m \times d}$) and tensor $\mathbf{R}$ (user embedding, $\mathbf{R} \in \mathbb{R}^{d \times d \times n}$), and then, we calculate $\tilde{\mathbf{F}}_{i,p,q} = \mathbf{Q}_p * \mathbf{A}_q^T$ where $\mathbf{Q}_p$ is the query vector, i.e., the dot product of $\mathbf{A}_p \in \mathbb{R}^{1 \times d}$ for course p and $\mathbf{R}_i \in \mathbb{R}^{d \times d \times 1}$ for i-th student. To speed up the recommendation process, we implement a hashing technique by using the approximate nearest neighbor (ANN) indexing library, ANNOY. In this case, the query vector, $\mathbf{Q}_p$, is calculated for any course p taken by i-th student and we find the courses q which are nearest neighbors to the query vector using annoy indexing and calculate $\tilde{\mathbf{F}}_{i,p,q}$ for those p, q pairs. Then, for the i-th student, we recommend the courses q that have the highest $\tilde{\mathbf{F}}_{i,p,q}$ scores based on the courses p that the student has already taken.

We tried different rank values for factorization, d=[1, 2, 3, 4, 5, 8, 10, 15, 20, 50], and different numbers of nearest neighbors [5, 40, 100] for ANN indexing. However, we observe better results when we do not use ANN indexing which takes the maximum number of nearest neighbors (all available courses) into consideration.

4.4.4 *Scholars walk*

We use the Scholars walk model [23], a non-session-based approach to recommend a set of courses for the next session. First, we calculate matrix $\mathbf{F} \in \mathbb{R}^{m \times m}$ that contains the frequency $\mathbf{F}_{p,q}$ of every pair of consecutive courses $p, q \in \mathcal{C}$. We normalize $\mathbf{F}$ to get the transition probability matrix, $\mathbf{T}$. Then we perform a random walk on the course-to-course graph that is described by $\mathbf{T}$. The probability of the walker reaching the vertices after K steps gives an intuitive measure that is useful to rank the courses for a student to offer a personalized recommendation. We use the scholars walk algorithm to perform a random walk with restarts which guides us to consider direct and transitive relations between the courses.

We tried the following value for the parameters: the number of steps allowed=[1,2,3,4,5]; *alpha*=[1e-4, 1e-3, 1e-2, 1e-1, 0.2, 0.4, 0.6, 0.7, 0.8, 0.85, 0.9, 0.99, 0.999]; *beta* values from 0 to 1.6 with step 0.1.

Table 3: Performance comparison in terms of Recall@k.

Model	Validation	Test
Non Sequential baseline	0.1600	0.1039
Sequential baseline	0.2991	0.2470
Tensor Decomposition	0.1596	0.1309
Scholars Walk	0.3619	0.2679
CourseBEACON	**0.3859**	0.2948
CourseDREAM	0.3856	**0.3023**

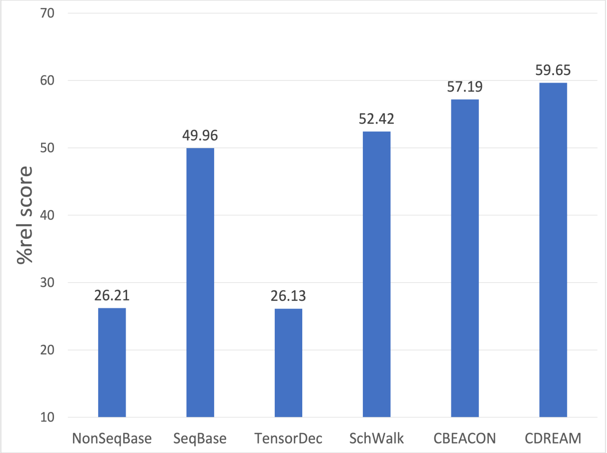

Figure 1: Percentage of instances with at least one relevant recommendation

4.5 Recommending courses

We recommend the top courses for the target semester $\mathcal{B}_{i,t_i}$ based on the predicted score $\mathbf{y}_{i,j}$ for course c_j for i-th student. The scores demonstrate how likely is for each course to be taken on the next semester with respect to both correlation of courses within the semester and sequential associations of courses over the semesters. We also create a list of courses for each semester t observing which courses are offered and available for all students. During post-processing, while recommending courses for a student, we filter out the courses which the student took in any previous semester and the courses which are not offered at that target semester [21, 23]. Then, we recommend the top $k = |\mathcal{B}_{i,t_i}|$ courses based on the highest scores for that student.

5. RESULTS

In this section, we will discuss the performance of our proposed approaches compared to the state-of-the-art sequential and non-sequential session-based or non-session-based approaches. We will also present how the hyperparameters affected the overall performance of our models.

5.1 Performance Comparison

The performance results of our proposed approaches and other competing approaches are shown in Table 3. We present Recall@k score for both the validation and test data. The percentage of relevant recommendations (%rel, percentage of at least one correct prediction for each instance) is presented in Figure 5.1.

First, if we only consider existing approaches, the best performing model is Scholars Walk. This model achieves recall performance around 26.79% and produces at least one relevant recommendation for 52.42% of the instances. On the opposite side, the non-sequential baseline performs particularly badly. The reason might be that in our school, we have a lot of transfer students, and the courses that they take in their second semester for example might be very different from traditional students.

Second, sequential approaches (sequential baseline, Scholars walk, CourseBEACON, CourseDREAM) outperform the non-sequential approaches (non-sequential baseline and tensor decomposition). This indicates that the sequence of courses over the semesters is an important factor in course selection. Sequential approaches can capture the student-course interaction along with the sequential transition of courses. On the contrary, non-sequential approaches just capture student-course interactions and students' preferences. We actually tested another non-sequential approach, BPR-MF, but the recall results were as low as the Tensor Decomposition, and we decided to only present one of these non-sequential approaches.

Third, our proposed session-based approaches outperform all the other competing approaches for course recommendation that we tested. We have the highest Recall@k and %rel with the CourseDREAM model. Between the two proposed models, CourseDREAM seems to be more stable, as there is a smaller difference between the validation and test performance. The fact that CourseDREAM behaves betters than CourseBEACON indicates that latent vector representation using the average pooling technique is more effective than creating the correlation matrix for the courses taken within a semester. An explanation could be that the correlation matrix may suffer from and be dominated by popular courses. We can also see that capturing the relationship of courses taken in a semester in the session-based approach is working better than other sequential approaches. The %rel scores of CourseDREAM demonstrate that we can recommend at least one correct recommendation for 59.65% of the instances.

Fourth, deep learning models (like LSTM networks) can capture the sequential transition of courses over the sequence of semesters. Incorporating the notion of the suitability of courses co-taken within a semester produces more accurate and useful recommendations.

5.2 The effect of different hyperparameters

First, we examine how the parameter α affects the results of the CourseBEACON model. In Fig. 5.2, we present the Recall@k of the validation and test set achieved for different values of α for the combination of parameters that have the best Recall@k (dropout rate = 0.4, embedding dimension d = 64, and hidden units = 128). We observe that lower values of α provide better performance, with the best model having $\alpha = 0.3$. This means that the sequential transition of courses plays more importance than the intra-basket correlation of courses within a semester. However, we still need to consider the relationship between courses taken in the same semester for course recommendation. This is what gives an advantage to these session-based models.

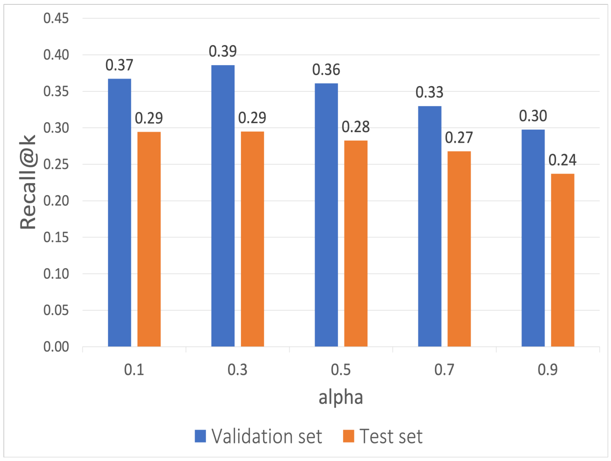

Figure 2: The effect of α in CourseBEACON model

Table 4: The effect of different hyperparameters in CourseDREAM model in terms of Recall@k (dropout rate = 0.4)

embedding dimensions	RNN layers	Validation	Test
32	1	0.3587	0.2782
32	2	0.3559	0.2792
32	3	**0.3856**	**0.3023**
16	1	0.3706	0.2882
16	2	0.3649	0.2943
16	3	0.3770	0.2990
8	1	0.3721	0.2958
8	2	0.3602	0.2826
8	3	0.3312	0.2659

Next, we examine the performance of the CourseDREAM model with respect to the parameters of embedding dimension and number of RNN layers in Table 4. Here, we have set dropout rates to 0.4, which gives us the best-performing model. We observe that the number of LSTM layers and the number of embedding dimensions do influence the results. Except for the case of 8 embedded dimensions, our models benefit from the increased number of RNN layers, which capture more complex patterns in the data.

6. CONCLUSION

We propose the use of session-based recommendation approaches for recommending suitable and complementary courses for the upcoming semester. In particular, we introduce CourseDREAM and CourseBEACON, two sequential session-based approaches that capture the relationship of the co-taken courses in different ways. Our experimental results show that our proposed models outperform all the sequential and non-sequential competing approaches. CourseDREAM can provide more relevant recommendations for the students so that recommended set of courses are related and more likely to be taken together within a semester. Our models will be helpful in advising students to achieve better performance overall and graduate on time.

7. REFERENCES

[1] A. Al-Badarenah and J. Alsakran. An automated recommender system for course selection. *International Journal of Advanced Computer Science and Applications*, 7(3):166–175, 2016.

[2] S. Boumi and A. E. Vela. Quantifying the impact of students' semester course load on their academic performance. In *2021 ASEE Virtual Annual Conference Content Access*, 2021.

[3] M. G. Brown, R. M. DeMonbrun, and S. D. Teasley. Conceptualizing co-enrollment: Accounting for student experiences across the curriculum. In *Proceedings of the 8th international conference on learning analytics and knowledge*, pages 305–309, 2018.

[4] S. Chockkalingam, R. Yu, and Z. A. Pardos. Which one's more work? predicting effective credit hours between courses. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 599–605, 2021.

[5] C. De Medio, C. Limongelli, F. Sciarrone, and M. Temperini. Moodlerec: A recommendation system for creating courses using the moodle e-learning platform. *Computers in Human Behavior*, 104:106168, 2020.

[6] A. Diamond, J. Roberts, T. Vorley, G. Birkin, J. Evans, J. Sheen, and T. Nathwani. Uk review of the provision of information about higher education: advisory study and literature review: report to the uk higher education funding bodies by cfe research. 2014.

[7] N. Entezari, E. E. Papalexakis, H. Wang, S. Rao, and S. K. Prasad. Tensor-based complementary product recommendation. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 409–415. IEEE, 2021.

[8] A. Esteban, A. Zafra, and C. Romero. A hybrid multi-criteria approach using a genetic algorithm for recommending courses to university students. *International educational data mining society*, 2018.

[9] J. Gardner and C. Brooks. Coenrollment networks and their relationship to grades in undergraduate education. In *Proceedings of the 8th international conference on learning analytics and knowledge*, pages 295–304, 2018.

[10] M. Hanson. *College Dropout Rates*. EducationData.org, June 17 2022. `https://educationdata.org/college-dropout-rates`.

[11] A. Kadlec, J. Immerwahr, and J. Gupta. Guided pathways to student success perspectives from indiana college students and advisors. *New York: Public Agenda*, 2014.

[12] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[13] D.-T. Le, H. W. Lauw, and Y. Fang. Correlation-sensitive next-basket recommendation. 2019.

[14] B. Ma, M. Lu, Y. Taniguchi, and S. Konomi. Exploration and explanation: An interactive course recommendation system for university environments. In *CEUR Workshop Proceedings*, volume 2903. CEUR-WS, 2021.

[15] B. Mondal, O. Patra, S. Mishra, and P. Patra. A course recommendation system based on grades. In *2020 international conference on computer science, engineering and applications (ICCSEA)*, pages 1–5. IEEE, 2020.

[16] S. Morsy and G. Karypis. Learning course sequencing for course recommendation. 2018.

[17] S. Morsy and G. Karypis. Will this course increase or decrease your gpa? towards grade-aware course recommendation. *arXiv preprint arXiv:1904.11798*, 2019.

[18] J. Naren, M. Z. Banu, and S. Lohavani. Recommendation system for students' course selection. In *Smart Systems and IoT: Innovations in Computing*, pages 825–834. Springer, 2020.

[19] R. Obeidat, R. Duwairi, and A. Al-Aiad. A collaborative recommendation system for online courses recommendations. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pages 49–54. IEEE, 2019.

[20] A. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS)*, 29(4):1–33, 2011.

[21] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User modeling and user-adapted interaction*, 29(2):487–525, 2019.

[22] Z. A. Pardos and W. Jiang. Designing for serendipity in a university course recommendation system. In *Proceedings of the tenth international conference on learning analytics & knowledge*, pages 350–359, 2020.

[23] A. Polyzou, A. N. Nikolakopoulos, and G. Karypis. Scholars walk: A markov chain framework for course recommendation. *International Educational Data Mining Society*, 2019.

[24] Z. Ren, X. Ning, A. S. Lan, and H. Rangwala. Grade prediction based on cumulative knowledge and co-taken courses. *International Educational Data Mining Society*, 2019.

[25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[26] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.

[27] E. Shao, S. Guo, and Z. A. Pardos. Degree planning with plan-bert: Multi-semester recommendation using future courses of interest. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14920–14929, 2021.

[28] M. S. Sulaiman, A. A. Tamizi, M. R. Shamsudin, and A. Azmi. Course recommendation system using fuzzy logic approach. *Indonesian Journal of Electrical Engineering and Computer Science*, 17(1):365–371, 2020.

[29] P. Symeonidis and D. Malakoudis. Multi-modal

matrix factorization with side information for recommending massive open online courses. *Expert Systems with Applications*, 118:261–271, 2019.

[30] M. Wan, D. Wang, J. Liu, P. Bennett, and J. McAuley. Representing and recommending shopping baskets with complementarity, compatibility and loyalty. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1133–1142, 2018.

[31] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)*, 54(7):1–38, 2021.

[32] C. Wong. Sequence based course recommender for personalized curriculum planning. In *International Conference on Artificial Intelligence in Education*, pages 531–534. Springer, 2018.

[33] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732, 2016.

[34] H. Zhang, T. Huang, Z. Lv, S. Liu, and Z. Zhou. Mcrs: A course recommendation system for moocs. *Multimedia Tools and Applications*, 77(6):7051–7069, 2018.