Scholars Walk: A Markov Chain Framework for Course Recommendation

Agoritsa Polyzou
University of Minnesota
polyz001@umn.edu

Athanasios N. Nikolakopoulos
University of Minnesota
anikolak@umn.edu

George Karypis
University of Minnesota
karypis@umn.edu

ABSTRACT

Course selection is a crucial and challenging problem that students have to face while navigating through an undergraduate degree program. The decisions they make shape their future in ways that they cannot conceive in advance. Available departmental sample degree plans are not personalized for each student, and personal discussion time with an academic advisor is usually limited. Data-driven methods supporting decision making have gained importance to empower student choices and scale advice to large cohorts. We propose *Scholars Walk*, a random-walk-based approach that captures the sequential relationships between the different courses. Based on the "wisdom of the crowd" and the students' prior courses, we recommend a short list of courses for next semester. Our experimental evaluation illustrates that Scholars Walk outperforms other collaborative filtering and popularity-based approaches. At the same time, our framework is very efficient, easily interpretable, while also being able to take into consideration important aspects of the educational domain.

Keywords

course recommendation, Markov chains, random walks, sequential recommendation, higher education

1. INTRODUCTION

The general purpose of higher education is to offer programs, which will help learners to gain knowledge throughout their studies. Students enjoy a plethora of offerings. However, course selection can be "messy and unorganized" [3] as it depends on many factors that students need to consider. Students have to balance personal preferences (interests, objectives, and career goals) and general education and degree program requirements. As a result, course selection can be a non-trivial task.

Decisions can be made based on manual guides offered from each department, but these are not tailored to individual cases [7] in a higher education setting. Personalized assistance can be given by academic advisers, however this is not scalable with large cohorts with thousands of students. The ratio of student to advisor may be very high [14], limiting the adviser-advisee discussion time. Additionally, college students take on average up to 20% more courses than required [2]. Better advising can help alleviate these problems. We need predictive models that can be employed to enable strategic action and attain better results. In this paper, we focus on appropriately designing a course recommendation system (CRS) that could facilitate the conversation between advisors and students for future planning.

There are several existing approaches to generate a set of courses to recommend for next semester. Their majority suggest courses based on either the constraints and requirements that they satisfy or their expected grades. This paper introduces *Scholars Walk*, a random-walk based approach for the course selection problem. It describes a personalized model that takes advantage of the sequential nature of course selection. We assume that students' choices for the next term depend on the courses they have taken so far. In our approach, we build a Markov chain for each degree program over the courses taken consecutively. Then, we perform a random walk, starting from the courses that students took in the previous semester. We evaluated the proposed approach on a number of different departments with different subjects and characteristics. Scholars Walk overall outperforms other competing approaches in all the metrics considered in this paper.

2. RELATED WORK

Recommender systems have been broadly applied within the context of student learning [16]. We will further review the different approaches developed to help students select a subset of courses to register for an upcoming semester. The first course recommender systems are based on constraint satisfaction [22]. The sequence-based recommender [24] also considers complex constraints to improve the expected time-to-degree and GPA. A related body of work involves mining of association rules. Al-Badarenah et al. [1] cluster the students based on their grades first. Nguyen et al. [18] apply sequential rule mining in (course, grade) pairs and recommend the courses with the best performance. A different CSR was proposed by Esteban et al. [10], where there is available information about students' satisfaction after taking a course.

Agoritsa Polyzou, Athanasios N. Nikolakopoulos and George Karypis "Scholars Walk: A Markov Chain Framework for Course Recommendation" In: *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, Collin F. Lynch, Agathe Merceron, Michel Desmarais, & Roger Nkambou (eds.) 2019, pp. 396 - 401

Table 1: Statistics for each major.

Major	n	m	grades	%pop	flex
Accounting	846	53	22,524	45.9	0.28
Aerospace Engr	532	109	16,259	25.7	0.10
Biology	1,275	146	28,084	14.9	0.11
Biol Society Env	709	57	14,597	31.4	0.31
Biomedical Engr	644	131	19,748	23.8	0.16
Chemical Engr	826	108	24,825	26.3	0.11
Chemistry	724	145	18,292	17.4	0.14
Civil Engr	651	112	19,189	26.5	0.12
Communication	1,333	95	22,421	15.4	0.19
Computer Sc	998	161	24,899	13.7	0.11
Electrical Engr	740	164	22,191	17.1	0.12
Elementary Ed	770	49	16,527	40.4	0.31
English	1,176	153	17,736	9.5	0.11
Finance	1,234	83	32,255	29.7	0.20
Genetics Cell	680	93	15,385	23.1	0.19
Journalism	2,306	100	40,519	17.1	0.20
Kinesiology	1,176	164	33,622	14.8	0.16
Marketing	1,291	69	29,901	30.8	0.20
Mechanical Engr	1,369	132	39,436	18.9	0.11
Nursing	819	86	25,136	31.2	0.27
Nutrition	554	87	15,591	29.7	0.19
Political Science	1,307	171	19,260	8.1	0.12
Psychology	1,894	115	31,141	13.4	0.15

n, m are the number of students and courses.
%pop is the course popularity (percentage of students that took a course at least once).
The last column (flex) is the degree flexibility.

Recently, recurrent neural networks (RNNs) have been successfully applied within the educational domain. Long Short Term Memory (LSTM) networks have been used for grading prediction [13, 20]. In terms of course recommendation, a combination of LSTMs and skip-gram models has also developed to balance implicit and explicit student preferences [23]. Morsy et al. [17] have also used RNN to recommend courses which are expected to help maintain or improve students' GPA. Other approaches include a Markov-based model [15], that represents the sequence of courses taken as a stochastic process. Garner et al. [11] build a co-enrollment network and extract features for a network-based structural model. Finally Elbadrawy et al. [9] propose using the academic features to improve the recommendation performance.

3. DOMAIN & DATASET

This work focuses on the undergraduate students in a traditional educational institution. We used a dataset from the University of Minnesota that spans more than 10 years. The A–F grading scale (A, A-, B+, B, B-, C+, C, C-, D+, D, F) is used. Courses in which a student receives less than a C- do not count toward satisfying degree requirements.

We extracted the degree programs that have at least 500 graduated students from 23 different majors. We only kept students that actually received their degree and had at least three consecutive semesters with valid courses. We selected the 40 most frequent courses and the courses that belonged to frequent subjects. A subject is considered frequent if students have taken at least three courses that belong to that subject on average. We removed instances without an A–F grade, and non-academic courses, like independent/directed study or field study. We did not consider offerings in the summer semester. As these are less common, they would distort the course sequence of students not enrolled in summer. Basic statistics for each degree program are shown in Table 1. The average course popularity (%pop) for course i is the percentage of students that have taken i at least once during their studies. The degree flexibility (flex) is a measure of how different are the course selections that students make. It is one minus the average Jaccard similarity coefficient for every pair of students. The Jaccard similarity is computed as the number of courses that two students have in common divided by the minimum courses that student has taken them.

4. PROPOSED METHOD

4.1 Assumptions & Notation

In the context of course recommendation for higher education, we make the following assumptions:

1. Time is discrete and moves in steps, from one semester to the next.
2. There is a relative ordering of the courses in terms of course levels, difficulty or material covered.
3. Learning is seldom non-sequential; each course completed provides some knowledge and experience that can be used in future courses. As a consequence, sequence matters in course selection.
4. In the absence of enough domain experts, the order in which courses are taken by students historically can reveal useful information on the curriculum and degree requirements.
5. We know the number of courses that the student will take next semester.

For the rest of the paper we will adopt the following notation. When we use the word *target* we will refer to the student/course/semester for which we want to generate recommendation. Matrices are denoted with capital bold letters, while vectors are denoted with lower bold letters. Calligraphic letters will be used for sets.

The set of students is $\mathcal{S}$ and has size m. The set of all courses is denoted by $\mathcal{C}$, $|\mathcal{C}| = n$. Student j has an enrollment history $\mathcal{H}_j$, that is an ordered set of courses, $\{\mathcal{C}_{j,1}, \ldots, \mathcal{C}_{j,t}, \ldots, \mathcal{C}_{j,t_j}\}$, where $\mathcal{C}_{j,t}$ is the set of courses taken in semester t and t_j is the last semester that the student took courses. Table 2 presents the symbols we used.

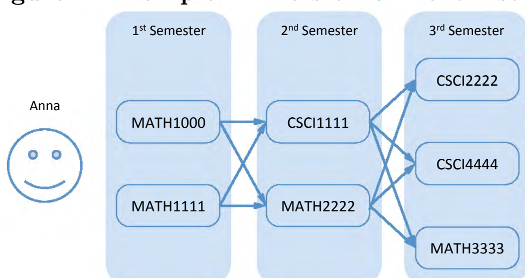
4.2 Building the Markov chain

Markov models satisfy the Markov property, i.e., the conditional probability distribution of future states depends only on the current state. In the simplest Markov model, known as first-order, each state is formed by a single action, i.e., a student took a course. In the case of K-th-order models, the state-space will correspond to all possible sequences of K actions. As the available data could not adequately support the number of states of higher-order chains, these models would suffer from reduced coverage and possibly worse overall performance [6]. Therefore, we adopted a first-order

Table 2: Notation.

n, m	number of courses, students
i, i'	indexes for courses
j, j'	indexes for students
t_j	number of semesters that j has taken courses
t	index for semesters
$\mathcal{C}$, $\mathcal{S}$	set of courses, courses
$\mathcal{A}$	set of states in Markov chain $\{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$
$\mathcal{H}_j$	enrollment history of student j
$\mathcal{C}_{j,t}$	course that student j took in semester t
$\mathbf{T}$, $\mathbf{F}$	matrices $(n \times n)$
$\mathbf{T}_{k,l}$	the (k, l) element of matrix $\mathbf{T}$
$\mathbf{T}_k$	the k-th row of matrix $\mathbf{T}$ $(1 \times n)$
$\mathbf{T}_{:,l}$	the l-th column of matrix $\mathbf{T}$ $(n \times 1)$
$\mathbf{u}$	personalization vector $(1 \times n)$
$\mathbf{p}^{(k)}$	state vector $(1 \times n)$ at timestep k

Figure 1: Example: Anna's enrollment history.



Markov chain. We assume that the next-semester courses depend only on the courses that the student is taking the current semester.

Markov models are represented by the parameters $\langle \mathcal{A}, \mathbf{T} \rangle$, where $\mathcal{A}$ is the set of states for which the Markov model is defined; and $\mathbf{T}$ is an $(n \times n)$ transition probability matrix (TPM), where n is the number of states (i.e., courses). In this context, state $\mathcal{A}_i$ is associated with the fact that the student took the course i. Each entry $\mathbf{T}_{i,i'}$ corresponds to the probability of moving to state $\mathcal{A}_{i'}$ when the process is in state $\mathcal{A}_i$, i.e., taking course i' after course i. Note that this matrix is not symmetric, i.e., $\mathbf{T}_{i,i'} \neq \mathbf{T}_{i',i}$, as the order in which the courses are taken matters.

Based on the historical enrollment information of the students, we first compute $\mathbf{F}$, an $(n \times n)$ matrix that holds the counts of every pair of consecutive courses. Every pair of courses (i, i') that a student has taken consecutively is used to estimate the entry $\mathbf{F}_{i',i}$, i.e., the frequency of the event that state $\mathcal{A}_{i'}$ follows the state $\mathcal{A}_i$. For example, consider student Anna in Fig. 1. The entry corresponding to the course pair of (MATH1000, CSCI1111) will be updated. Similarly, every line connecting two courses will equally contribute in the corresponding element of matrix $\mathbf{F}$.

After we compute the frequencies of matrix $\mathbf{F}$, we need to normalize it to get $\mathbf{T}$, a row stochastic matrix, so that the total transition probability from state i to any other state will sum up to 1:

$$\mathbf{T}_i = \mathbf{F}_i / \sum_{i'=1}^{n} \mathbf{F}_{i,i'}, \ \text{if} \sum_{i'=1}^{n} \mathbf{F}_{i,i'} > 0.$$

Additionally, it is possible that the sum of some rows to be zero. This occurs when a course is taken at the last semester of every student, so there are no courses after that to pair it with. In that case, we set the diagonal elements of the zero rows to one; $\mathbf{T}_{i,i} = 1$ and $\mathbf{T}_{i,i'} = 0$ for $i \neq i'$, if $\sum_{i'=0}^{n} \mathbf{F}_{i,i'} = 0$.

4.3 Walking over courses

We can view the Markov chain in the context of random walk on a course-to-course graph that is governed by the transition probability matrix. A random walk on a directed graph will form a path of vertices generated from a start vertex by selecting an edge, making a step by traversing the edge to a new vertex, and repeating the process [4]. This concept has been applied to many scientific fields. Closer to this work, random walks have recently been used for top-n item recommendation [19], and they are also known to empower systems used in production at major social media platforms [12, 8].

A random walk starts with any probability distribution $\mathbf{u} \in \mathcal{R}^{1 \times n}$. $\mathbf{u}_i$ is the probability of starting at vertex i. If one starts at a vertex i, then $\mathbf{u}_i = 1$, else $\mathbf{u}_{i'} = 0$ for $i' \neq i$. In our setting, the random walk for student j will equally start from any course in the student's last semester, so the personalization vector will be:

$$\mathbf{u}_i = \begin{cases} 1/|\mathcal{C}_{j,t_j}| & \text{if } i \in \mathcal{C}_{j,t_j}, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Let $\mathbf{p}^t \in \mathcal{R}^{1 \times n}$ be a row vector with an element for each vertex specifying the probability of being there at time t. Before we start the walk, $\mathbf{p}^0 = \mathbf{u}$. After the first step, the probability of being at vertex i' is the sum over each adjacent vertex i of starting at i and taking the transition from i to i'. In matrix notation, when we are at state k and we take a step, we will get the following probability distribution:

$$\mathbf{p}^{k+1} = \mathbf{p}^k \mathbf{T}, \quad (2)$$

where the i-th entry of the $\mathbf{p}^{k+1}$ is the probability of the walk after $k+1$ steps to land at vertex i. This can be written as a function of the starting probability vector as:

$$\mathbf{p}^{k+1} = \mathbf{u}\mathbf{T}^k. \quad (3)$$

The probability of the walker to reach the vertices after K steps provides an intuitive measure that can be used to rank the courses and offer personalized recommendations to the student accordingly.

Scholars Walk

To introduce an additional way for personalization in our model, we perform a *random walk with restarts* [21]. We introduce a parameter α, $0 < \alpha \leq 1$ that controls if the walk will take the step described above, or if the walk will restart. In the latter case, we use the personalized probability distribution as the restarting distribution. The probability dis-

Algorithm 1 Scholars Walk

Input: Model $\mathbf{T}$, student's personalization vector $\mathbf{u}$, parameters α, β, number of steps K.
Output: Recommendation vector $\mathbf{p}^{rec}$.

```
p^0 ← u, k ← 0
repeat
    k ← k + 1
    p^k ← α p^{k−1} T + (1 − α) u          ▷ Take a step.
    p^k ← p^k / ||p^k||_1                   ▷ Normalize p^k.
until ||p^k − p^{k−1}||_2 < tol or k ≥ K
for i ← 1 to n do
    p_i^k ← p_i^k * pop_i^{−β}              ▷ Penalize popular courses.
end for
p^rec ← p^k
```

tribution now is defined as:

$$\begin{aligned}\mathbf{p}^{k+1} &= \alpha \mathbf{p}^k \mathbf{T} + (1-\alpha)\mathbf{u} \\ &= \mathbf{p}^k(\alpha \mathbf{T} + (1-\alpha)\mathbb{1}\mathbf{u}) \\ &= \mathbf{u}(\alpha \mathbf{T} + (1-\alpha)\mathbb{1}\mathbf{u})^k,\end{aligned} \tag{4}$$

where $\mathbb{1}$ is a column vector $(n \times 1)$ of ones. The product of $\mathbb{1}\mathbf{u}$ will give us an $(n \times n)$ matrix where every row will have the probability that the walk will start at the corresponding course. Scholars Walk will perform a random walk governed by the matrix $\alpha\mathbf{T} + (1-\alpha)\mathbb{1}\mathbf{u}$.

The exact steps we followed are shown in Alg. 1. We can specify the number of steps to perform, or we can allow the algorithm to converge. If the number of steps is very small, the walk might not explore enough courses. If the number of steps is large, the walk might travel too far, and the recommendations might not be so relevant for the student. Additionally, to limit the domination of popular courses, we penalize the probabilities with the term $\text{pop}_i^{-\beta}$ [5], where pop_i is the popularity of the course. The parameter $\beta, 0 < \beta \leq 1$ shows how harsh we need to be with the penalty term.

Scholars Walk allows us to consider direct, as well as, transitive relations between the courses. It also provides a considerable degree of personalization, in order to recommend courses that are relevant to each particular student.

5. EXPERIMENTAL DESIGN

5.1 Competing approaches

The baselines are two group popularity approaches, on the department level (**Pop1**) and the academic level (**Pop2**) of the student measured by the number of years in the program [9]. For Pop1, we recommend the most popular courses in the major. For Pop2, we recommend the most common courses on the major and the academic level of the student ("freshmen", "sophomores", "juniors", and "seniors"). Students after their forth year are considered seniors.

We also compared against Basic Markov model (**Markov**) and Basic Markov model with skip (**MarkovSkip**) [15]. In these models, for a target student, the set of courses that other students have taken after taking a course that the target student took are the possible courses to recommend. We consider the combination of courses during the last two semesters to build and test the model. Each course is assigned a recommendation score that is the sum of all the conditional probabilities that lead to that course starting from the student's enrollment in the last semester. While the counts used in this case are the same with the ones computed in our matrix $\mathbf{F}$, the conditional probabilities are computed differently. In order to produce recommendations for students whose set of prior courses did not have a match, the skip model was introduced. In that case, we find other students that have similar course history with the target student, and weight their corresponding probabilities by a parameter λ.

Last, we train an **LSTM**-based course prediction model similar to [17, 23]. LSTMs can learn temporal dependencies with additional gates to retain and forget selected information. As input, we use a multi-hot representation of course enrollments per semester which are mapped to a predicted sequence of vectors. Once the LSTM has been learnt, we feed the network with a binary vector that indicates the courses that the target student has taken the past semester. The weights at the output of the model are used to rank the courses.

5.2 Evaluation metrics

Like in prior work [9, 15, 17, 23], we used **Recall@**n_s as the primary evaluation metric for the predictions, where n_s is the number of courses that the student took in the target semester. This is the percentage of actual enrolled courses that were contained in the recommendation list. The reported metrics are averaged out across all students predicted. Note that recall and precision are equivalent in our setting, since we recommend exactly as many courses as the student will take the upcoming semester.

We also compute the percentage of queries for which we were able to retrieve at least one of the courses that the student took in the target semester (**%rel**). It measures for how many cases we were able to recommend at least one course that was relevant.

5.3 Experimental setting

Model selection. Using the dataset described in Sect. 3, we split it into train, validation and test sets as follows. All semesters before 2013 (about 10 years) were used for training, courses taken during 2013 and in Spring 2014 were used for validation, and courses taken afterwards (Fall 2014 to Spring 2017) were used for test purposes, to report the results. The training set was used for building the models, whereas the validation set was used to select the best performing parameters in terms of the highest Recall@n_s. Based on the best set of parameters for the validation set, we computed the test set results in Sect. 6.

Parameters. For parameter α, we tried the following set of values: {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}. For parameter β, we tested values from 0 to 0.8, in increments of 0.025. In terms of the number of steps that we allowed for our walker, we tested the values 1, 3, and 1000. The last value corresponds to no limit for the number of steps.

Additional filtering. We build a different model for each

Table 3: Results for Scholars Walk w.r.t. K.

K	Recall@n_s	%rel	α	β	avg#steps
1	0.466	75.1	0.955	0.047	1
3	0.460	74.6	0.088	0.053	1.95
1000	0.461	74.6	0.075	0.051	2.32

K is the number of steps that we allow to our walker. α, β columns show the average values of these parameters over the models of all the majors.
The last column shows the actual average number of steps the Scholars Walk made before convergence.

Table 4: Performance comparison.

Model	Recall@n_s	%rel
Pop1	0.336	62.5
Pop2	0.338	64.6
Markov	0.456	73.0
MarkovSkip	0.400	69.6
LSTM	0.406	69.6
Scholars Walk	**0.466**	**75.1**

major for all the approaches we tested. After we generate a ranked list of the courses using any method, we filter out courses that are not offered the target semester. We also remove courses that the student has taken in the past and achieved a grade above C-, as they do not count towards any degree requirements, as mentioned in Sect. 3. In the end, we return a list with as many recommendations as the number of courses, n_s, that the student took next semester, based on assumption 5.

6. RESULTS

In this section, we will try to answer the following questions: 1) How do the parameters in our models affect the overall performance? Specifically, how does the number of steps affect recommendation performance? 2) What is the performance of our approach compared to the state-of-the-art approaches?

6.1 The effect of the number of steps

The performance of our models in terms of the metrics computed for different values of K is shown in 3. For each model and selection of K, we see the values of the parameters α and β that were used. These parameters were selected based on the recall on the validation set. The parameter α controls the restarting probabilities, while β is used to reweight the probability distribution before recommending its highest-weighted courses. The column avg#steps shows the average number of steps that the Scholars Walk actually made before convergence.

In this domain, we need only a few steps, as we can understand from Table 3: not only when we set $K = 1$ we get the best performance, but also, when we allow the walk to take many steps, the parameter α gets smaller values. This forces the walk to go back to the student's personalized starting vector with higher probability, indicating that the starting distribution is very important. Additionally, even if we do not put any constraints in K, the number of steps that the Scholars Walk takes is quite small. There is a small increase when increasing K from 1 to 3, but after that, the number of steps actually taken is not that high.

It is worth pointing out that, while setting $K = 1$ gives us the best overall performance, this is not the case for all the departments. The right value for K depends on the dataset used. In our data, there are four departments that need these extra steps. We observed that these departments have low average course popularity, which is average percentage of students that have taken a course at least once at some point during their studies, over all the courses. The average value for the departments with $K > 1$ was $16.7 \pm 9.7\%$, while for the rest of the models the corresponding number is $24.1 \pm 7.2\%$. A stronger signal is present in the metric of the degree flexibility, which is the average Jaccard distance between the courses that any pair of students took, as defined in the end of Sect. 3. The departments with $K > 1$ have 0.118 ± 0.005 degree flexibility against 0.184 ± 0.066 of the rest of the departments. This is an indicator that for stricter degrees, the walk depends on the extra steps to explore more courses. In these departments, students will take overall very similar sets of courses. On the other hand, if the degree program offers more freedom to the students, they select a wider range of courses, and there are more connections within courses.

6.2 Performance comparison

By comparing the best Scholars Walk model against five competing approaches, we get the results on Table 4. Our model performs the best, both in terms of recall, and in the percentage of cases for which it manages to be return some relevant recommendations.

Popularity approaches are having considerably satisfactory performance. However, specifying the academic level of the student does not help much. They can recommend relevant courses to more than 60% of the cases. The two Basic Markov models have quite different performance. The Markov model with skips performs poorly, compared to the Basic model. Additionally, it is worth mentioning that the Skip model was performing better and better as the parameter λ was getting smaller. The weight of the cases that do not completely match the target student's history, have as weight a power of λ. Consequently, when $\lambda \to 0$, the Skip model becomes the Basic Model. For that reason, the smaller value of λ that we report results for, is 0.4.

While comparing the Basic Markov model with Scholars Walk, it may seem that they have similar performance. However, that might be misleading, as the Basic Markov model utilizes longer course enrollment history than the Scholars Walk. It looks back two semesters on the student's courses, which corresponds to a second-order Markov chain. Moreover, the model uses data from two semesters not only for computing the associated probabilities, but also to make predictions. This leads to increased complexity because of the larger state-space with no benefit in recommendation quality. In the same boat are the LSTMs as well. Their increased complexity might lead to the overfitting of the model, when the data are not sufficient for training. Our approach, which is a first-order Markov chain, manages to perform better than the higher-order models and LSTMs.

Scholars Walk can accurately predict the course selection of the students, by taking advantage of the "breadth and depth" of the data. In terms of time complexity, once we build the transition probability matrix, walking through the courses is trivial. As a result, it scales well with the number of students, while providing them personalized recommendations. At the same time, it is a white-box model, where the recommendations are easily explainable.

7. CONCLUSION

In this paper we propose Scholars Walk, a novel method designed to harvest the sequential patterns arising from past course enrollment data in order to recommend a short list of personalized course suggestions for the next semester. The proposed method relies on a random walk-based scheme on a course-to-course graph and personalization is achieved by a student-adapted starting distribution reflecting the current student's enrollments. When compared with five competing models, from popularity-based to LSTMs and Basic Markov models, Scholars Walk achieves the best performance. It manages to be a successful, scalable approach that provides personalized recommendations for every student.

8. ACKNOWLEDGMENTS

This work was supported in part by NSF (1447788, 1704074, 1757916, 1834251), Army Research Office (W911NF1810344), Intel Corp, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

9. REFERENCES

[1] A. Al-Badarenah and J. Alsakran. An automated recommender system for course selection. *Intl. Journal of Advanced Computer Science and Applications*, 7(3):1166–1175, 2016.

[2] C. C. America. Time is the enemy, 2011.

[3] E. Babad and A. Tayeb. Experimental analysis of students' course selection. *British Journal of Educational Psychology*, 73(3):373–393, 2003.

[4] A. Blum, J. Hopcroft, and R. Kannan. Random walks and markov chains. In *Foundations of data science.* Vorabversion eines Lehrbuchs, 2016.

[5] F. Christoffel, B. Paudel, C. Newell, and A. Bernstein. Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks. In *9th ACM Conf. on Recommender Systems*, pages 163–170, New York, NY, USA, 2015. ACM.

[6] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Trans. on Internet technology (TOIT)*, 4(2):163–184, 2004.

[7] 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.

[8] C. Eksombatchai, P. Jindal, J. Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, and J. Leskovec. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *World Wide Web Conf.*, pages 1775–1784, 2018.

[9] A. Elbadrawy and G. Karypis. Domain-aware grade prediction and top-n course recommendation. In *10th ACM Conf. on RecSys*, pages 183–190, 2016.

[10] A. Esteban, A. Z. Gómez, and C. Romero. A hybrid multi-criteria approach using a genetic algorithm for recommending courses to university students. In *11th Intl. Conf. on Educational Data Mining*, 2018.

[11] J. P. Gardner, C. Brooks, and W. Li. Learn from your (markov) neighbor: Coenrollment, assortativity, and grade prediction in undergraduate courses. *Journal of Learning Analytics*, 5(3):42–59, 2018.

[12] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh. Wtf: The who to follow service at twitter. In *22Nd Intl. Conf. on World Wide Web*, pages 505–514, New York, NY, USA, 2013. ACM.

[13] Q. Hu and H. Rangwala. Course-specific markovian models for grade prediction. In *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 29–41. Springer, 2018.

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

[15] E. S. Khorasani, Z. Zhenge, and J. Champaign. A markov chain collaborative filtering model for course enrollment recommendations. In *Big Data (Big Data), IEEE Intl. Conf. on*, pages 3484–3490. IEEE, 2016.

[16] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper. Recommender systems in technology enhanced learning. In *Recommender systems handbook*, pages 387–415. Springer, 2011.

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

[18] H.-Q. Nguyen, T.-T. Pham, V. Vo, B. Vo, and T.-T. Quan. The predictive modeling for learning student results based on sequential rules. *Intl. Journal of Innovative Computing, Information and Control (IJICIC)*, 14(6):2129–2140, 2018.

[19] A. N. Nikolakopoulos and G. Karypis. Recwalk: Nearly uncoupled random walks for top-n recommendation. In *12th ACM Intl. Conf. on Web Search and Data Mining*, pages 150–158. ACM, 2019.

[20] F. Okubo, T. Yamashita, A. Shimada, and H. Ogata. A neural network approach for students' performance prediction. In *Seventh Intl. Learning Analytics & Knowledge Conf.*, pages 598–599. ACM, 2017.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

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

[23] 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*, pages 1–39, 2019.

[24] C. Wong. Sequence based course recommender for personalized curriculum planning. In *Intl. Conf. on Artificial Intelligence in Education*, 2018.