A Markov Chain Collaborative Filtering Model for Course Enrollment Recommendations

Elham S.Khorasani, Zhao Zhenge, John Champaign
Department of Computer Sciencce
University of Illinois at Springfield
Springfield,IL, USA
esahe2@uis.edu, zzhao43@uis.edu, john.champaign@gmail.com

***Abstract*—In this paper we detail our initial approach and early results in examining the efficacy of a Markov-based stochastic model to course enrollment recommendations. We outline a Markov-based collaborative filtering model to recommend courses to students at each semester based on the sequence of courses they have taken in the previous semesters. The proposed model is based on the enrollment data and no prior knowledge of the institution, course prerequisites, curriculum or degree requirement is assumed. Using enrollment data from a research university in Canada, we evaluate and compare the Markov model with traditional collaborative filtering approaches for course recommendation. Our initial results show that the Markov-based model significantly outperforms traditional collaborative filtering models when applied to course enrollment recommendation.**

***Keywords*-course recommender system, collaborative filtering, Markov-based collaborative filtering, academic planning**

I. Introduction

pre-requisites and choosing between course options has been a feature of the educational environment dating back to Platos inscription ”Let no one ignorant of geometry enter!” inscribed at the entrance of his academy [1]. In efforts to remain competitive, course options at institutes of higher learning have exploded, often offering hundreds of course possibilities that satisfy their general education requirements [2]. From personal experience, the authors can confirm that when changing academic policies are included, faculty advisors find course selection equally daunting.

Several studies in the past have addressed the problem of recommending appropriate courses to the students [3]. O'Mahony and Smyth [4] used a variation of the item-based collaborating filtering to recommend elective courses to University College Dublin students based on the core modules taken by students. Tai et. al. [5] used clustering-based collaborative filtering to group students with similar interest and recommend courses for each group. A decision tree is used in [6] on historical transcript data to predict the grade of a student if enrolled in a certain course. Aher and Lobo [7] used clustering to group students based on the categories of the courses they enrolled in the past and then applied association rule mining to each group to find the courses that have been frequently taken together. The result is then used to recommend courses to an active student in each group. Koutrika et. al. [8] discussed an implementation of a closed community based course recommender system where students' comments and ratings for courses are used in addition to users' logs to recommend courses to students. The quality of the recommendations produced by such system depends heavily on the number of ratings provided by a community of students. Various methods are proposed in [9] to encourage students' participation to increase course ratings and the quality of recommendations.

Existing works in the course enrollment recommendation systems do not consider the sequence of courses taken by each individual student over his/her course of study. The order in which courses have been taken by students can provide valuable insight into institutional curriculum and degree requirements and can be used to enhance recommendations.

In this study, we use a Markov based model, where a sequence of courses taken by students are represented as stochastic processes and the transition probabilities are estimated from the data [10]. Two approaches for estimating transition probabilities are examined: 1-basic Maximum Likelihood Estimation (MLE) and 2- enhanced MLE estimation based on skip-gram modeling[11].

A dataset containing historical enrollment data of ten years (2001-2011) obtained from a Canadian research university was used to evaluate the accuracy of the recommendations produced by the Markov model. Apache Spark is used to implement and evaluate the Markov model on this dataset and to compare its accuracy with traditional collaborative filtering approaches..

The rest of the article is organized as follows. Next section provides a brief background on traditional collaborative filtering techniques. Section 3 presents Markov-based collaborative filtering models for course recommendations. Section 4 evaluates the efficacy of the Markov models on the course enrollment data and compares its accuracy with other common collaborative filtering methods. Section 5 concludes the study and present future research directions.

II. Collaborative Filtering-based Course Recommender

Collaborative-filtering is a recommendation approach that uses similarity between users and the benefit they have received from items in the past to make recommendations. People are most familiar with this style of recommendation from purchases on Amazon.com, an online retailer, which makes recommendations based on both a particular user's purchase history and on the basis of other users' common purchases.

Three main approaches to collaborative filtering are memory-based, model-based and hybrid methods [12]. Memory-based approaches use a user-item rating matrix to compute pairwise similarities between users or items and use the similarity values to predict ratings and make recommendations. Several measures of similarity have been used, including Correlation-based metrics, vector cosine similarity, Jaccard set similarity, conditional probability-based similarity, etc. Typical examples of memory-based approaches are item-based and user-based collaborative filtering methods. Typical examples of model-based approaches are matrix-factorization based recommender systems.

In the context of course enrollment recommendations, we want to recommend courses for a particular student to enroll in on the basis of courses that similar students have enrolled in the past. Students do not explicitly rate a course at the time of enrollment; therefore, similarity measures which do not rely on explicit ratings, such as Jaccard set similarity, are more appropriate to use in this context [13]. Jaccord set similarity measures the similarity between two courses as the ratio of the number of users who took both courses over the number of users who took either course.

An item-based course recommender measures the similarity between two courses based on the students that took both courses. To predict a recommendation score for a course c, given student s, the item-based recommender computes the average similarity of c to other courses taken by s.

In contrast, a user-based course recommender measures the similarity between two students based on their common courses. To predict a recommendation score for a course c, given student s, a user-based course recommender first picks a neighborhood of top n most similar students to s who took c. It then computes the recommendation score as the average similarity of s to the users in this neighborhood.

In general, the item-based collaborative filtering is a better choice in the context of course recommendation for two reasons: first, the number of students taking courses is typically higher than the number of courses taken by the students. Therefore, it is more efficient to compute pairwise course similarities as opposed to pair-wise student similarities making the item-based course recommender more computationally efficient. Second, the courses in the system are less likely to change over time as opposed to the students who take those courses. Hence, the pairwise course similarity matrix can be computed off-line.

Matrix factorization based techniques for recommender systems identify a set of latent features from the item-rating pattern and factorize the user-rating matrix to two lower dimensional user-feature and feature-item matrices. Matrix factorization methods have proven effective to address some of the challenges faced by the memory-based methods such as scalability and sparsity of the ratings matrix [14].

III. Markov-based Collaborating Filtering Models For Course Enrollment Recommendation

The traditional memory-based and model-based course recommender systems do not consider the sequence of courses taken by the students in each semester. In the context of course enrollment recommender, the order in which courses are taken plays an important part in advising and developing a student's academic plan. For example, an academic advisor might advise a student to take ”statistical Inference” course before ”Data Science” or he may advise students not to take two more involved courses concurrently . In addition, in the absence of institutional knowledge, the order in which courses are taken by students can provide useful information on the curriculum and degree requirement and increase the accuracy of the recommender systems.

A. Basic Markov Model

To account for the order of courses in the collaborative filtering method, we model the sequence of courses that a student takes as a Markov process in which the courses a student will take in semester $k+1$ depend only on the courses that he/she has taken in the previous k semesters.

A state in the basic Markov model is represented as a set of k courses taken in k consecutive semesters: $s = \{c1, c2, c3, ..., ck\}$

The transition probability of going from state $s_1 = \{c_1, c_2, c_3, ..., c_k\}$ to state $s_2 = \{c_1, c_2, c_3, c_4, ..., c_{k+1}\}$ can be estimated from the enrollment data using the Maximum Likelihood Estimation (MLE) [15] as follows:

$$p(s_2 = \{c_1, c_2, ..., c_k, c_{k+1}\}|s_1 = \{c_1, c_2, ..., c_k\}) = \frac{count_{st}(\{c_1, c_2, c_3, ...c_k\} \rightarrow c_{k+1})}{count_{st}(\{c_1, c_2, ..., c_k\})} \quad (1)$$

Where $count_{st}(\{c_1, c_2, c_3, ...c_k\} \rightarrow c_{k+1})$ represents the number of students who took c_{k+1} after taking the consecutive courses $\{c_1, c_2, c_3, ..., c_k\}$ in k previous semesters and $count_{st}(\{c_1, c_2, ..., c_k\})$ is the total number of students who took $\{c_1, c_2, c_3, ..., c_k\}$ in k consecutive semesters.

Since students typically take more than one course per semester, every student is mapped to several states in the state space corresponding to various combinations of courses she has taken in k (or $k+1$) consecutive semesters.

We calculate a recommendation score $r(st, c_j, j)$ for each course c_j that a student st is likely to take in semester j, given his enrollments in k previous semesters, $s_1 = \{c_{j-1}, c_{j-2}, ..., c_{j-k}\}$, as follows:

$$r(st, c_j, j) = \sum_{s_1=\{c_{j-1},c_{j-2},\ldots,c_{j-k}\}} p(s_1 \cup \{c_j\}|s1) \tag{2}$$

The above formula simply adds up all transition probabilities for $s_1 \rightarrow s_1 \cup \{c_j\}$ where s_1 is a sequence of consecutive courses taken by the student in k previous semesters.

To illustrate the implementation of the Markov model, let's suppose that we have a sample enrollment data set of four students in three consecutive semesters as described in table I.

Stundent	Semester	Courses
st_1	1	CSC385, CSC302
	2	CSC388, CSC389
	3	CSC478, CSC387
st_2	1	CSC 225, CSC442
	2	CSC 385, CSC302, CSC 275
	3	CSC 388, CSC389
st_3	1	CSC275, CSC385
	2	CSC472, CSC388
	3	CSC 442, CSC 378
st_4	1	CSC388, CSC387
	2	CSC389, CSC385
	3	CSC378, CSC472

Table I
SAMPLE ENROLLMENT DATA OF FOUR STUDENTS IN THREE CONSECUTIVE SEMESTERS.

Student	Chain of two courses	chain of three courses
st_1	CSC385→ CSC388	CSC385→CSC388→CSC478
	CSC385→CSC389	CSC385→CSC388→CSC387
	CSC302→CSC388	CSC385→CSC389→CSC478
	CSC302→CSC389	CSC385→CSC389→CSC387
	CSC388→CSC478	CSC302→CSC388→CSC478
	CSC388→CSC387	CSC302→CSC388→CSC387
	CSC389→CSC478	CSC302→CSC389→CSC478
	CSC389→CSC387	CSC302→CSC389→CSC387
st_2	CSC225→CSC275	CSC225→CSC275→CSC389
	CSC225→CSC302	CSC225→CSC275→CSC388
	CSC225→CSC385	CSC225→CSC302→CSC389
	CSC442→CSC275	CSC225→CSC302→CSC388
	CSC442→CSC302	CSC225→CSC385→CSC389
	CSC442→CSC385	CSC225→CSC385→CSC388
	CSC385→CSC388	CSC442→CSC275→CSC389
	CSC385→CSC389	CSC442→CSC275→CSC388
	CSC302→CSC388	CSC442→CSC302→CSC389
	CSC302→CSC389	CSC442→CSC302→CSC388
	CSC275→CSC388	CSC442→CSC385→CSC389
	CSC275→CSC389	CSC442→CSC385→CSC388
st_3	CSC275→CSC472	CSC275→CSC472→CSC442
	CSC275→CSC388	CSC275→CSC472→CSC378
	CSC385→CSC472	CSC275→CSC388→CSC442
	CSC385→CSC388	CSC275→CSC388→CSC378
	CSC472→CSC442	CSC385→CSC472→CSC442
	CSC472→CSC378	CSC385→CSC472→CSC378
	CSC388→CSC442	CSC385→CSC388→CSC442
	CSC388→CSC478	CSC385→CSC388→CSC378
st_4	CSC388→CSC389	CSC388→CSC389→CSC378
	CSC388→CSC385	CSC388→CSC389→CSC472
	CSC387→CSC389	CSC388→CSC385→CSC378
	CSC387→CSC385	CSC388→CSC385→CSC472
	CSC389→CSC378	CSC387→CSC389→CSC378
	CSC389→CSC472	CSC387→CSC389→CSC472
	CSC385→CSC378	CSC387→CSC385→CSC378
	CSC385→CSC472	CSC387→CSC385→CSC472

Table II
CHAINS OF TWO AND THREE CONSECUTIVE COURSES TAKEN BY EACH STUDENT

The following steps outlines the implementation of the Markov model to predict the courses that each student is most likely to take in their 4th semester.

Step 1–Building the States: Build two sets of chains per student:

1) **k-chains:**$\{c_1 \rightarrow c_2 \rightarrow c_3 \cdots \rightarrow c_k\}$ where c_2 is taken after c_1, c_3 is taken after c_2, and so on.
2) **k+1 chains:**$\{c_1 \rightarrow c_2 \rightarrow c_3 \cdots \rightarrow c_k \rightarrow c_{k+1}\}$ where c_{k+1} is taken after c_k.

For simplicity, let's assume that a history of two consecutive semesters (k=2) is considered in the Markov model for the dataset in table I. This assumes that the courses a student enrolls in each semester depends on the courses he/she took in the previous two semesters. The chains of two and three consecutive courses for each student are listed in table II.

Step 2–Finding Transitions: To make recommendations for a student, we need to take the set of k consecutive courses taken by the student in k previous semesters and for each set c, find all k+1 chains in the dataset starting with c. This step builds the state transitions by identifying the next courses that a student is likely to take based on the course he/she has taken in previous k semesters. Table III shows the result of this step for each student in the example dataset. The second column lists a set of two consecutive courses taken by each student in semesters 2 and 3 $\{c_2 \rightarrow c_3\}$. The third column shows all transitions from column 2; that is, all chains of three consecutive courses in the dataset starting with $\{c_2, c_3\}$.

Step 3–computing Recommendation Scores The recommendation score for each course that a student is likely to take next semester is computed using formulas 1 and 2.

For instance, to compute the probability that st_2 will take $CSC442$ in her fourth semester, one needs to find the transitions leading to $CSC442$ and compute their probabilities using MLE (1. There are two such transitions for st_2:

1) $\{385, 388\} \rightarrow \{385, 388, 442\}$, and
2) $\{275, 388\} \rightarrow \{275, 388, 442\}$

Stundent	sets of consecutive courses taken in the last two semesters	all unique courses in dataset taken after the two courses in column 2
st_1	{CSC388,CSC387}	–
	{CSC388,CSC472}	–
	{CC389,CSC472}	–
	{CSC389,CSC387}	{CSC387,CSC389}→ CSC472
		{CSC387,CSC389}→CSC378
st_2	{CSC385,CSC388}	{CSC385,CSC388}→CSC478
		{CSC385,CSC388}→CSC387
		{CSC385,CSC388}→CSC442
		{CSC385,CSC388}→CSC378
		{CSC388,CSC385}→CSC472
	{CSC385,CSC389}	{CSC385,CSC389}→ CSC478
		{CSC385,CSC389}→ CSC387
	{CC302,CSC388}	{CC302,CSC388}→CSC478
		{CC302,CSC388}→CSC387
	{CSC302,CSC389}	{CC302,CSC389}→CSC478
		{CC302,CSC389}→CSC387
	{CSC275,CSC388}	{CSC275,CSC388}→CSC442
		{CSC275,CSC388}→CSC378
	{CSC275,CSC389}	–
st_3	{CSC472,CSC442}	–
	{CSC472,CSC378}	–
	{CC388,CSC442}	–
	{CSC388,CSC378}	–
st_4	{CSC389,CSC378}	–
	{CSC389,CSC472}	–
	{CC385,CSC378}	–
	{CSC385,CSC472}	{CSC385,CSC472}→CSC442
		{CSC385,CSC472}→ CSC378

Table III
CHAINS OF TWO AND THREE CONSECUTIVE COURSES TAKEN BY EACH STUDENT

The probability of the first transition is 1/3 which is equal to the number of students who took CSC442 after taking {CSC385,CSC388} over the number of students who took {CSC385,CSC388}in two consecutive semesters. Similarly, the probability of the second transition is 1 which is equal to the number of students who took CSC442 after {CSC275,CSC388} over the number of students who took {CSC275,CSC388} in two consecutive semesters. Consequently, the recommendation score for st_2 taking CSC442 in her fourth semester is:

$$r(st_2, CSC442) = \frac{1}{3} + \frac{1}{1} = 1.33 \tag{3}$$

Similarly, table IVshows the courses that each student is likely to take next semester together with their recommendation scores. The courses which have already been taken by each student are excluded from the third column.

B. Skip Model

One of the problems with the Markov model described above is data sparsity. This means if the set of consecutive courses taken by a student in k previous semesters does not match those of any other students, then there will be no recommendation for this student. For instance, st_3 in table IV does not get any course recommendation for semester 4 because the courses she took in semesters 2 and 3 does not match any course sequence in the dataset.

Stundent	likely courses for next semester	recommendation score
st_1	CSC378	1
st_2	CSC478	1/3+1+1+1=3.33
	CSC387	1/3+1+1+1=3.33
	CSC442	1/3+1=1.33
	CSC378	2/3+1=1.66
	CSC	1/3=0.33
st_3	No recommendation	–
st_4	CSC442	1
	CSC378	1

Table IV
COURSE RECOMMENDATIONS FOR EACH STUDENT

Inspired by language modeling community, several methods have been developed to address data sparsity in markov chains representing sequential data [16], [17], [18]. The model used here is a simple skipping model which assumes that the courses a student takes in semester $k+1$ does not depend only on k previous semesters but can also depend on the semesters before that. To maintain the memory-less property of the Markov model, this model allows some semesters to be skipped when building the state space. For example, if we allow one semester to be skipped in building the chains of courses, then st_3 will have the following additional chains of two courses:

$$\begin{array}{l} CSC275 \to CSC378 \\ CSC275 \to CSC442 \\ CSC385 \to CSC442 \\ CSC385 \to CSC378 \end{array}$$

Weights are assigned to each state to differentiate between the states that are built with and without skipping. The more semesters skipped in a state, the less the state should weigh in predicting courses for the future semester. If a chain of courses is built by skipping n consecutive semesters, then the chain will be assigned a weight equal to λ^n where λ is a coefficient between 0 and 1 . For example, if one semester is skipped in building a chain its weight is equal to λ, if two consecutive semesters are skipped in the chain then its weight is equal to λ^2 and so on. If no semester is skipped in a chain its weight will be equal to one. The transitional probabilities are then computed as follows:

$$p(s_1 = \{c_1, c_2, ..., c_k\} \to s_2 = \{c_1, c_2, ..., c_k, c_{k+1}\}) = \frac{\sum_{st} W(st, \{c_1, c_2, ...c_k, c_{k+1}\})}{\sum_{st} W(st, \{c_1, c_2, ..., c_k\})} \tag{4}$$

Where $W(st, \{c_1, c_2, ..., c_k\})$ is the weight assigned to state $\{c_1, c_2, ..., c_k\}$ and $W(st, \{c_1, c_2, ...c_k, c_{k+1}\})$ is the weight assigned to state $\{c_1, c_2, ..., c_{k+1}\}$ for student st.

The recommendation score $r(st, c_j, j)$ for each course c_j that a student st is likely to take in semester j, given his enrollments in the previous k semesters, $s_1 = \{c_{j-1}, c_{j-2}, ..., c_{j-k}\}$ in the skip model is computed as follows:

$$r(st, c_j, j) = \sum_{s_1=\{c_{j-1},c_{j-2},...,c_{j-k}\}} W(st, s_1)p(s_1 \rightarrow s_1 \cup \{c_j\}) \quad (5)$$

Where $W(st, s_1)$ is the weight of state $\{c_{j-1}, c_{j-2}, ..., c_{j-k}\}$ for st. If no semester is skipped in building the states, all states have weights equal to one and the model will be reduced to the basic Markov model.

For instance, for st_3, the skip chain CSC275→CSC442 matches with transitions {CSC442,CSC275}→CSC389 and {CSC442,CSC275}→CSC388 in the dataset. Similarly, the skip chain CSC385→CSC442 for st_3 matches with transitions {CSC442,CSC385}→ CSC389 and {CSC442,CSC385}→CSC388 in the dataset. CSC388 has already bee n taken by this student;hence, we only compute the recommendation score for CSC389. Assumming $\lambda = \frac{1}{2}$ we have:

$$\begin{aligned} r(st_3, 389) &= W(st_3, \{275, 442\}) * P(\{275, 442\} \rightarrow 389) \\ &+ W(st_3, \{385, 442\}) * P(\{385, 442\} \rightarrow 389) \\ &= \frac{1}{2} * 1 + \frac{1}{2} * 1 = 1 \end{aligned}$$

IV. Evaluation

To evaluate the performance of the Markov model in course enrollment recommendation, we used an enrollment dataset from a Canadian research university and compared the accuracy of the Markov-based model with item-based and matrix factorization-based course recommenders. The enrollment dataset used included all students who had taken a computer science course at that university between September 2001 and December 2011.

The data was in a comma-separated file made up of rows with 6 fields: a unique, anonymized student identifier, the term (which could be Spring, Summer or Fall and the year), the subject ID (such as CS or ENGL), the course code (such as 101), the percentile grade received and the students major. The data needed to be preprocessed in preparation for implementing the Markov model. If a student enrolled in the same course more than once (for example, due to failing the course), we only retained the latest enrollment with maximum mark. The courses which appeared less than six times in the dataset were removed on the basis that these courses which are seldom registered by students were either unpopular courses or cancelled by the university. Furthermore, the students who had less than two semesters worth of data were removed from the dataset.

The enrollment data was split into training and testing sets. For each student, her latest semester in the dataset was put into the testing set and the rest of her enrollment data was added to the training set. After these adjustments, the training set consisted of 37,392 students with data about 468,632 courses they took. There were 2,326 unique courses in the training set. A small spark cluster of 12 cores, 48 GB memory, and 2 TB of disk was used to implement the basic and skip Markov models described in the previous section. We measured the precision and recall for the top 12 recommendations returned for each student by basic and skipping Markov models and compared them to item-based and matrix factorization-based recommenders. The rational for returning top 12 recommendations was to give on average three recommendations for every course that a student intends to take in the new semester with the assumption that a full time student takes, on average, four classes per semester.

Precision and recall metrics are borrowed from information retrieval and rely on the separation of relevant items from irrelevant items. Within the recommender system community the definition of relevancy is very subjective. In this work we consider a recommendation to be relevant if it appears for a specific student in the test data. With this assumption, recall would be the fraction of courses in the test set that are generated by the course recommender for each student. That is, what percentage of the courses that students took in their latest semester was recommended by the system? Precision measures the fraction of the course recommendations that are relevant, meaning, what percentage of the recommended courses were taken by the students in the test data. Since the test set only included a single semester for each student and the number of courses taken by each student in the test set was very small, we expected the precision to be very low. Therefore, we mostly relied on recall to measure the performance of the course recommender systems. Nevertheless, precision is still used for benchmarking and comparison of recommenders.

The performance of Markov recommenders is compared to three other recommender systems in table V. These include: 1- a random recommender which randomly recommends 12 courses to every student out of the courses that the he/she has not already taken, 2- an item-based recommender with Jaccord set Similarity to measure item-similarity, and 3- an alternating least square (ALS) matrix factorization based recommender system with implicit feedback [13]. For ALS recommender, we experimented with various numbers of features and iterations and reported the best values for precision and recall.

Table V shows that the precision and recall of the basic and skip Markov models are higher than both ALS matrix factorization based and item-based recommenders. The basic and skip Markov based models could predict roughly 73 and 78 percent of the courses students took in their latest semester, respectively. The recall of the skip Markov model was about 15% higher than that of ALS based recommender and 28% higher than the item-based

Recommender	Recall	Precision
Skip Markov Model	78%	23%
Basic Markov Recommender	73%	20%
ALS Recommender with Implicit Feedback	63%	11%
Item-Based Recommender with Jaccard Similarity	50%	0.8%
Random Recommnder	1%	0.6%

Table V
PERFORMANCE OF BASIC AND SKIP MARKOV MODELS COMPARED TO RANDOM AND ITEM-BASED RECOMMENDERS

recommender. In addition, the precisions of the Markov based models were about 10% higher than those of ALS and item-based recommenders. This confirms the fact that a student enrollments in a future semester can be predicted not only by the set of courses taken by other students, but also the order in which those courses have been taken.

While 78% may not be enough accuracy to rely solely on the course recommender system for automated advising, the authors would like to emphasize that no prior institutional knowledge was used in building the recommender systems. The accuracy can be boosted significantly if additional institutional knowledge, such as course prerequisites, core and elective courses and patterns of offering, is used to postprocess and enhance the recommendations generated by the system.

V. Conclusion and Future Work

In this work we examined, using historical data from students studying computer science, the effectiveness of a Markov based collaborative filtering course enrollment recommender. We argued that the order in which courses taken by students plays an important role in recommending new courses to students to take in their future semester. We showed that the precision and recall of the recommendations returned by the Markov model on this dataset outperforms those of item-based and matrix factorization-based recommender systems.

We consider this work as a pilot study to test our early expectations about how to preprocess and analyze the enrollment data. Moving forward we are interested in examining finer grain recommendations for students.

In this work the accuracy of the recommendations is measured by taking the courses that students ultimately enrolled in as ground truth. Instead, a comparison of the recommendations made by a recommender system to recommendations made by experienced advisors is a worthwhile evaluation of the system that should be examined.

Instead of predicting the courses that students actually enrolled in, we ultimately want to help students make better course selections than they would have made themselves. To this end, we would like to compare students success, measured by their GPA at graduation, to their fidelity to the recommended course of study. This fidelity can be calculated by determining the ratio of courses taken to courses recommended. Our hypothesis is that students who chose a course of study closer to what would have been recommended will have greater success than students who did not.

When data becomes available about student employment and salary after graduation, we are interested in examining correlations between transcript data and career success. One possible outcome would be to identify course results that are strongly predictive of long term success. Another useful outcome would be comparing students with a similar background to a student seeking course recommendations and base the recommendations on those similar students who were most successful after graduation.

In summary, we would like to develop a course recommender that will make suggestions to students based on their transcript. We would like to then examine the long term success of student who had access to these recommendation compared to a control group that did not.

Acknowledgment

The authors would like to thank... more thanks here

References

[1] W. Anglin, *Mathematics: A Concise History and Philosophy*. NY: Springer-Verlag, 1994.

[2] B. Schwartz, *The Paradox of Choice*. HarperCollins, 2009. [Online]. Available: https://books.google.com/books?id=g422yyua-P8C

[3] H. Drachsler, K. Verbert, O. C. Santos, and N. Manouselis, "Panorama of recommender systems to support learning," in *Recommender systems handbook*. Springer, 2015, pp. 421–451.

[4] M. P. O'Mahony and B. Smyth, "A recommender system for on-line course enrolment: An initial study," in *Proceedings of the 2007 ACM Conference on Recommender Systems*, ser. RecSys '07. New York, NY, USA: ACM, 2007, pp. 133–136. [Online]. Available: http://doi.acm.org/10.1145/1297231.1297254

[5] D. Wen-Shung Tai, H.-J. Wu, and P.-H. Li, "Effective e-learning recommendation system based on self-organizing maps and association mining," *The Electronic Library*, vol. 26, no. 3, pp. 329–344, 2008.

[6] C. Vialardi, J. Bravo Agapito, L. S. Shafti, and A. Ortigosa, *Recommendation in higher education using data mining techniques*. Barnes, T., Desmarais, M., Romero, C., & Ventura, S., 2009.

[7] S. B. Aher and L. Lobo, "Combination of machine learning algorithms for recommendation of courses in e-learning system based on historical data," *Knowledge-Based Systems*, vol. 51, pp. 1–14, 2013.

[8] G. Koutrika, B. Bercovitz, F. Kaliszan, H. Liou, and H. Garcia-molina, "Courserank: A closed-community social system through the magnifying glass," in *In The 3rd Intl AAAI Conference on Weblogs and Social Media (ICWSM*, 2009.

[9] R. Farzan and P. Brusilovsky, "Encouraging user participation in a course recommender system: An impact on user behavior," *Computers in Human Behavior*, vol. 27, no. 1, pp. 276 – 284, 2011, current Research Topics in Cognitive Load TheoryThird International Cognitive Load Theory Conference. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747563210002487

[10] G. Shani, R. I. Brafman, and D. Heckerman, "An mdp-based recommender system," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 453–460.

[11] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks, "A closer look at skip-gram modelling," in *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, 2006, pp. 1–4.

[12] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. in Artif. Intell.*, vol. 2009, pp. 4:2–4:2, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/421425

[13] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*. Greenwich, CT, USA: Manning Publications Co., 2011.

[14] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[15] T. W. Anderson and L. A. Goodman, "Statistical inference about markov chains," *The Annals of Mathematical Statistics*, pp. 89–110, 1957.

[16] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 310–318.

[17] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Comput. Linguist.*, vol. 18, no. 4, pp. 467–479, Dec. 1992. [Online]. Available: http://dl.acm.org/citation.cfm?id=176313.176316

[18] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech & Language*, vol. 15, no. 4, pp. 403–434, 2001.