



Duke Undergraduate Academic Trajectories and Course Recommender System

Vanessa Tang, Sangseok Lee, and Yiran Chen

Abstract

Currently, universities rely on academic advisors and students' individual research to guide major and course selection. However, with thousands of possibilities, it is extremely time-consuming and difficult to search through all the options. Using historical enrollment and graduation data, we propose several tools to facilitate major and course selection for undergraduate students at Duke University. First, a series of dashboards aid students in major and course selection as well as course timeline planning. Second, network and alluvial plots show curriculum and course flows from semester to semester to help identify common academic trajectories. Third, a course recommender system provides personalized course recommendations to students. Through the use of these tools, both students and advisors can better plan majors and academic pathways.

Introduction

The abundance of courses available and the variety of academic requirements are often overwhelming for students. First, with hundreds of different majors to choose from, students often find it difficult to decide which major to pursue, especially when they have ill-defined or several different interests. Second, even within a major, students have another hundreds and even thousands of options of classes to take, varying in difficulty, specialty, timing, professor, and workload. Third, students must fulfill graduation requirements but can do so in a variety of ways. This leaves students with essentially an infinite number of options when it comes to choosing a major and classes. A student's final decisions for classes depend on the risk vs. reward trade-offs perceived by each student with his or her individual needs and objectives (Jiang, 2019). It is important to provide students with sufficient information about potential academic pathways to ensure that their class selections help guide them towards their goals.

To do so, we provide several deliverables to facilitate the course selection process for students, given historical data on student enrollment and graduation for undergraduate students at Duke University. These include an integrated planner for major and course selection, visualizations of academic trajectories, as well as a course recommender system. The two main goals of this project are the following:

- 1) to track academic pathway and
- 2) to develop a course recommendation system.

To accomplish the first goal of tracking academic pathway, we developed a series of dashboards to facilitate major and course selection as well as course planning, and constructed visualizations of academic trajectories based on network analysis. To accomplish the second goal of developing a course recommendation engine, we built and tested several recommendation algorithms to identify one model that would best predict the classes a student would take in a certain term of interest. This schema is detailed in figure 1.

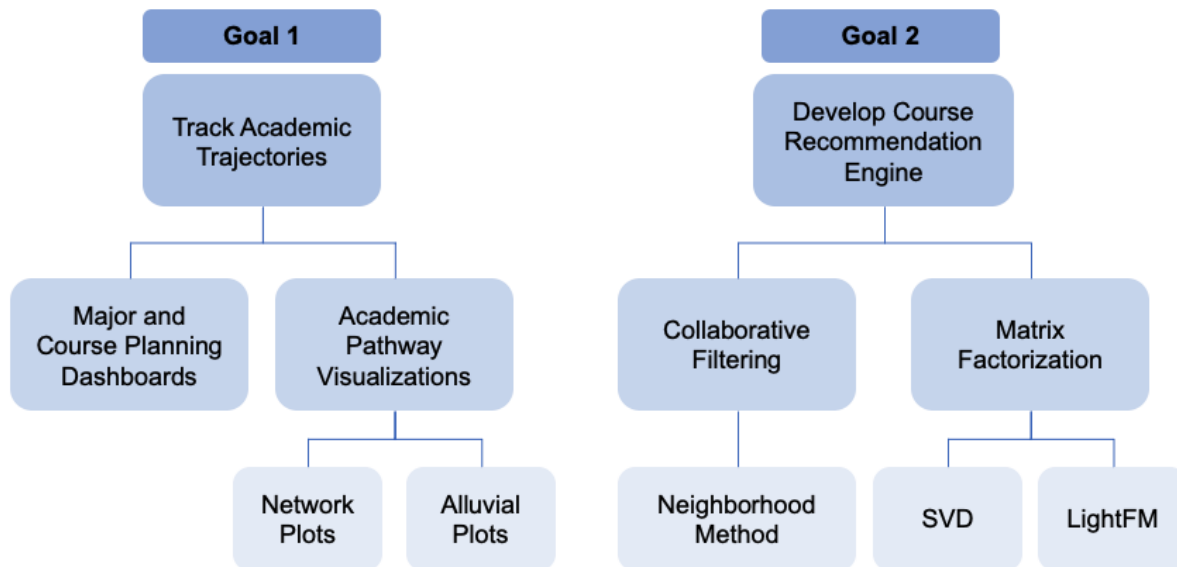


Figure 1. Overall schema of project, including primary goals and steps taken to work towards each goal.

Background


Currently, most universities, including Duke, rely on academic advisors to help students select classes. Students also spend a significant amount of time individually researching courses, majors, and professors to determine which classes to take. This process requires thorough research and thought about potential academic pathways. Despite help from advisors, individual research, and even suggestions from friends, students still have difficulty determining which classes to take. Furthermore, many first year students enter university not knowing what major they want to pursue. The current system relies on students deciding their major based on personal interests, potential career paths, or even pursuing the same major as friends. This issue underscores the importance of a tool to help facilitate this class and major selection process for students.

One proposed tool for facilitating course selection is ~~in the form of~~ a dashboard to support learning in the academic setting (Verbert, 2013). Most dashboards focus on graphical representations of historic data on students and courses to give professors a better understanding of course activities and teaching practices (Few, 2006). For this project, we share a similar idea of providing major and

course information given historic student data by exploring their course-taking behaviors across both time and major. The ~~planning~~ dashboards centralize information for both students and advisors and can help facilitate more informed and flexible decision making.

Academic advisors have also expressed interest in analyzing the structure and dynamics of the curriculum to better understand course-taking patterns, which enables visualization of popular course progressions and flows for each major (McFarland, 2006). In this project, we use visualizations of academic flows for courses within a major to identify the most popular pathways across each semester. These visualizations can help identify common course progressions and pathways for each major.

In addition to ~~using historical data to create~~ visualizations to ~~analyze academic trajectories~~, a course recommendation engine could also provide a more personalized approach to aiding students in the course selection process (Lee and Cho, 2011). Recommendation engines are commonly seen today in online platforms such as Netflix, Spotify, and Amazon, but are very uncommon in the academic setting. A course recommendation system could help manage the curriculum and keep track of students' academic progress (Lee and Cho, 2011).

There are two major branches of recommendation systems: content-based filtering and collaborative filtering (Charu, 2016). Content-based filtering uses item features to recommend other items similar to what the user previously liked. However, this model will only perform well given sufficient item features in the dataset (Charu, 2016). To address some of the limitations  content-based filtering, collaborative filtering uses similarities among both users and items to provide recommendations (Charu, 2016). This model recommends items to a user based on the preferences of another similar user (Charu, 2016). Similarity metrics are almost always problem-specific. One major advantage of collaborative filtering is that it can perform well without relying on item features. For our project, we implement collaborative filtering due to the absence of robust course-level information.

Overall, the development of a tool to facilitate course and major selection could be extremely beneficial for students. It could enable students to select a major that aligns more with their courses and interest, maximize efficiency in course-taking patterns, enable selection of courses that will fulfill requirements, as well as help students save time when looking for classes (Verbert, 2013). **Past work** includes implementation of dashboards and recommendation engines, and our project builds off these ideas to create a more tailored approach to undergraduate course and major planning at Duke University.

Data

From the Duke Registrar, we have two main datasets with degree and enrollment information split into two time periods: 2000 to 2012 and 2013 to 2020. Due to significant course renumbering and restructuring between Spring 2012 and Fall 2012, these two datasets cannot be explicitly linked, as there is no sure way to relate courses taken prior to 2012 to those taken after 2012. **Thus, all final deliverables are based on the 2013-2020** segment to enable visualization and model development based on the most recent data and the current course structure.

The first dataset contains graduation degree information, including graduated students' de-identified IDs, graduation year, and major, minor, certificate, or secondary degree descriptions. The second dataset contains all course enrollment information, including de-identified IDs corresponding to those in the degrees dataset in addition to course descriptions, grades received, and academic year and term for every course each student took at Duke. The two datasets were combined by merging on student ID (‘Calculation ID’) then cleaned and reorganized to create a dataset with one row for every course taken per student and separate columns for major, minor, certificate, and secondary degree information merged from the degrees dataset. The detailed data dictionary can be found in the appendix. The final dataset for the 2013 to 2020 segment has 325,107 rows with 13,513 students and 4,224 unique classes. There are 141 different primary majors.

There are approximately 6,000 to 7,000 undergraduate students enrolled at Duke every year. However, due to the structure of the datasets, the number of students incrementally decreases in the last few years as students “drop off” the dataset given the range of dates as students in 2010 graduate in 2014 but the dataset ends at 2012. These students are not included in this dataset. Thus, EDA and model building must account for potential inconsistencies at each end of the dataset’s time frame.



At Duke, every student graduates with a major, but students have the option of combining three of these options: major (MAJ), minor (MIN), secondary (SEC), and certificate (CER). As there are many different combinations of majors, minors, secondary, and certificates and all students must graduate with a major, we are focusing mainly on course planning for a student’s primary major in this project. Future work could help students identify secondary majors as well as minors, secondaries, or certificates.

Major and Course Planning Dashboards

The first goal of this project was to track academic pathways in order to facilitate the course selection process for Duke undergraduate students. To do so, we constructed a series of Tableau dashboards to aid in major and course selection as well as course planning. Tableau dashboards were chosen because they enable easy public access through Tableau Public, allow for extensive filtering for majors and/or classes, and are generally very user-friendly. The dashboards can be accessed through the following link:

[Duke University: Major and Course Planning](#)

The website consists of the following five main pages:



- 1. Home**

The Home page provides brief instructions on how to use the dashboards as well as a link to the feedback form.

- 2. Major Selection**

The Major Selection page enables students to search for classes they have *already taken*, then shows the top 10 most popular majors given the classes entered. The purpose of this



page is to help narrow down the search for majors based on classes a student has already taken. This can be especially helpful for first year students who may need more guidance in major selection.

3. Course Selection

The Course Selection page enables students to search for a major of interest, then select the term for which they are searching for classes. It then shows the top 10 most popular classes to take for the selected major and term as well as the total number of students who graduated with that major ~~between 2013 and 2020~~. The purpose of this page is to show students the common classes to take for their major and term. This can essentially identify requirements or even other non-major classes that many students within a major decide to take.

4. Course Planning

The Course Planning page enables students to search for classes they *want to take in the future*. Based on these classes, it shows the top 5 most common majors as well as a timeline of the popular terms to take each of the classes entered. By showing students popular majors given the courses entered, they can determine if these classes are “on track” with their major. The timeline feature can also help students balance their course load. For example, if it is common to take Molecular Biology in both the First Year Spring term and Second Year Fall term but the student’s First Year Spring term already has a heavy load, then he/she can choose to wait until the Second Year to take that class knowing that it is a common trend.



5. Feedback Survey

The Feedback page provides a link to a Google form where students and advisors can rate each page and provide comments. The purpose of this page is to enable us to receive feedback and continue making improvements on these dashboards.

Overall, this website consists of several dashboards that aim to facilitate major and course selection as well as course planning for students. ~~Though these dashboards do not incorporate traditional machine learning techniques,~~ they aim to be both easily understandable and helpful for the entire undergraduate student population as well as the academic advisors. These visualizations are based entirely on historical data of course enrollments and required no additional processing aside from the preprocessing and data cleaning previously detailed.

These dashboards help accomplish the first goal of this project of tracking academic pathways while creating a final end product usable by both students and advisors. We aim to continue improving these dashboards based on feedback from students and advisors. All in all, these dashboards serve to aid undergraduate students in selecting both majors and courses as well as planning course load.

Feedback and Future Work

We invited academic advisors of the Duke Registrar for a panel interview and gathered feedback regarding the dashboards. Overall, the feedback was extremely positive with a future direction focusing on integrating more information within the tool. Advisors were impressed with the synthesis of a large amount of information in a relatively ingestible way. Based on their feedback, we identified the following primary areas for future work for these dashboards:

- **Addition of more course data:** Future work could synthesize more data to allow students to see more course information, including locations, times, requirements, prerequisites, descriptions, and professors. These attributes often guide student course selection and could be a helpful addition to the dashboards.
- **Addition of major requirement data:** Currently, major requirement data is not available, as these often change from semester to semester. However, future work could synthesize course information with major requirement information to better guide students towards completing the necessary classes for their major.
- **Searching by course department and number:** The current implementation enables course searching by the name of the course. Searching by department and number (ex: Econ 101) would also be helpful to students. However, course numbering and/or names often change, so this would require contiguity across all data to ensure reliable results.
- **Filtering through Areas of Knowledge:** Within the Trinity College of Arts and Sciences at Duke, there are five areas of knowledge: Arts, Literatures, and Performance; Civilizations; Natural Sciences; Quantitative Studies; and Social Sciences. All students in Trinity must complete 2 credits within each area to graduate. Thus, this requirement guides the course selection process for many students, as Trinity is the largest college within Duke University. Based on feedback from advisors, it would be helpful to enable filtering classes by a selected Area of Knowledge, then show the most popular classes and/or the classes with the highest average grade. This could help students choose common classes that satisfy the Areas of Knowledge requirement as well as identify classes that are likely to be easier, as indicated by the higher average grade. Future work could integrate data with Areas of Knowledge information to combine with the current dataset on grades and enrollment.
- **Integration with DukeHub:** Ideally, this tool would be completely integrated with DukeHub, Duke's platform for course searching and registration. This would link the additional information of description, syllabus, timing, professor, etc. as well as enable students to add their courses to their cart then register.
- **Identifying patterns in combinations of classes:** Future work could identify classes that are commonly taken together to further help identify academic pathways. This could also identify combinations of courses where students are more likely to receive lower grades. Likewise, it could identify combinations of more difficult (lower average grade) courses taken with easier (higher average grade) courses to help students balance course loads and avoid taking too many difficult courses at the same time.

Note that providing grade information is controversial as it may discourage students from taking certain classes, even if they are major or graduation requirements (Main, 2014). Likewise, it may affect class enrollment and discourage advisors from recommending students to take a certain class knowing that historical data shows low average grades (Main, 2014). Thus, for the scope of this project, we were very wary of explicitly providing grade information. Though grade information is very valuable, it must be used cautiously due to its potential implications for guiding course selection.

Clearly, there is a wide variety of additions that can be made to this tool to further facilitate course and major selection for undergraduate students. While most of these suggestions are out of the scope of this project or not currently possible with the data provided, they can guide future work on this project to improve the tool for both advisors and students. Notably, almost all feedback from advisors were additions to the dashboard rather than revisions to the existing work. This emphasizes the effectiveness of our dashboards in providing useful information in an understandable way to help guide the course and major selection process for Duke undergraduate students.

Academic Trajectory Visualizations

While the major and course planning dashboards provide a deliverable usable for students and advisors, we also were interested in tracking academic pathways through other means. Thus, we implemented network and structure visualization tools based on social network analysis. In this case, *academic trajectories* can be defined as the progression of classes a student takes throughout their time at Duke. As stated by McFarland, academic trajectories show the “patterned flow of students across courses” (McFarland, 2006). This can be helpful in identifying common patterns of courses to take for each major, while highlighting prerequisites or courses that should be taken before proceeding to higher level courses. Furthermore, tracking pathways through visualizations can emphasize a common course-taking pattern for certain majors, thus identifying majors that may have a more straightforward structure than others.

Network Plots

To represent participant flows across courses, students’ schedules are used from two consecutive semesters (i.e. first year fall semester and first year spring semester) to develop large affiliation matrices (Friedkin and Thomas, 1997; Sorenson, 1987). In these affiliation matrices, rows are students, columns are courses, and cell values of 1 and 0 indicate membership in a particular semester. All observations where a student failed or dropped out mid-semester were excluded. After the affiliation networks are constructed (the enumeration of all students’ memberships in the fall semester (A_F) and spring semester (A_S)), matrix algebra is used to calculate participant flows across courses (Wasserman and Faust, 1994). To construct a mobility matrix, A_F and A_S must have the same number and order of students (rows) so that the transposed affiliation network of A_F (A_F^T) can be multiplied by the affiliation network of the spring semester (A_S). Through matrix multiplication, a mobility matrix of origins by destinations is created, $\mathbf{M} = A_F^T \times A_S$. The cell values in

this matrix indicate the number of students who transitioned from the Fall semester courses to the Spring semester courses. When presented in table form, the results are considered transition frequencies from origin states to destination states (Chase, 1991)

Transition frequencies show the volume of flows but do not test the likelihood that students will move from an origin state to a destination. Such a test can be constructed from the transition frequency tables by dividing the values in each row by the total number of students who are in the course (Chase, 1991; Sorenson, 1987). The new cell values represent maximum likelihood tests or transition probabilities that indicate the proportion of students from each course who move to the next position.

With the transition frequency table, we draw a network plot for the top 10 classes from each semester (Figure 2). Each node represents a class and each line indicates students' transition. The yellow line indicates course transition probabilities greater than 0.2. This plot shows a pattern in course transitions from semester to semester and can help guide students and advisors in choosing courses for the following semester.

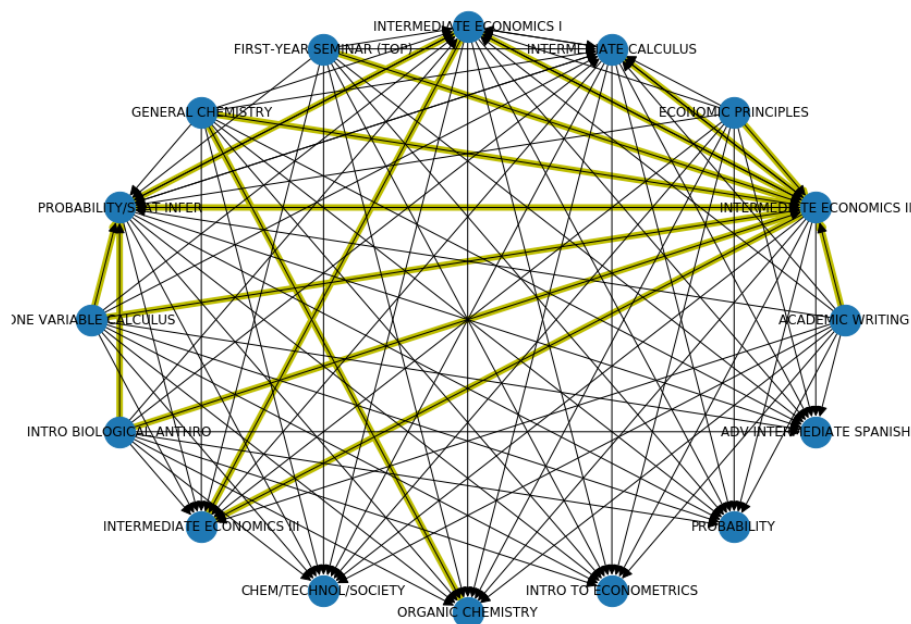


Figure 2. Network plot for economics majors from First Year Fall term to First Year Spring term. Each node is a class in the top 10 classes for each semester, directional lines indicate transitions from one class to the next, and yellow lines show transition probabilities greater than 0.2.

Alluvial Plots

While network plots help identify relationships among courses and common course-taking patterns, they do not indicate the popularity or number of students who take certain course progressions. Furthermore, they do not show as clear of a “pathway” from semester to semester. To help combat

some of the drawbacks of network plots, we implemented alluvial plots. In this case, alluvial plots show the flow of students from the popular classes in one semester on the left to the popular classes in the following semester on the right. The thickness of the bands indicate the number of students who progress from the class on the left to that on the right.

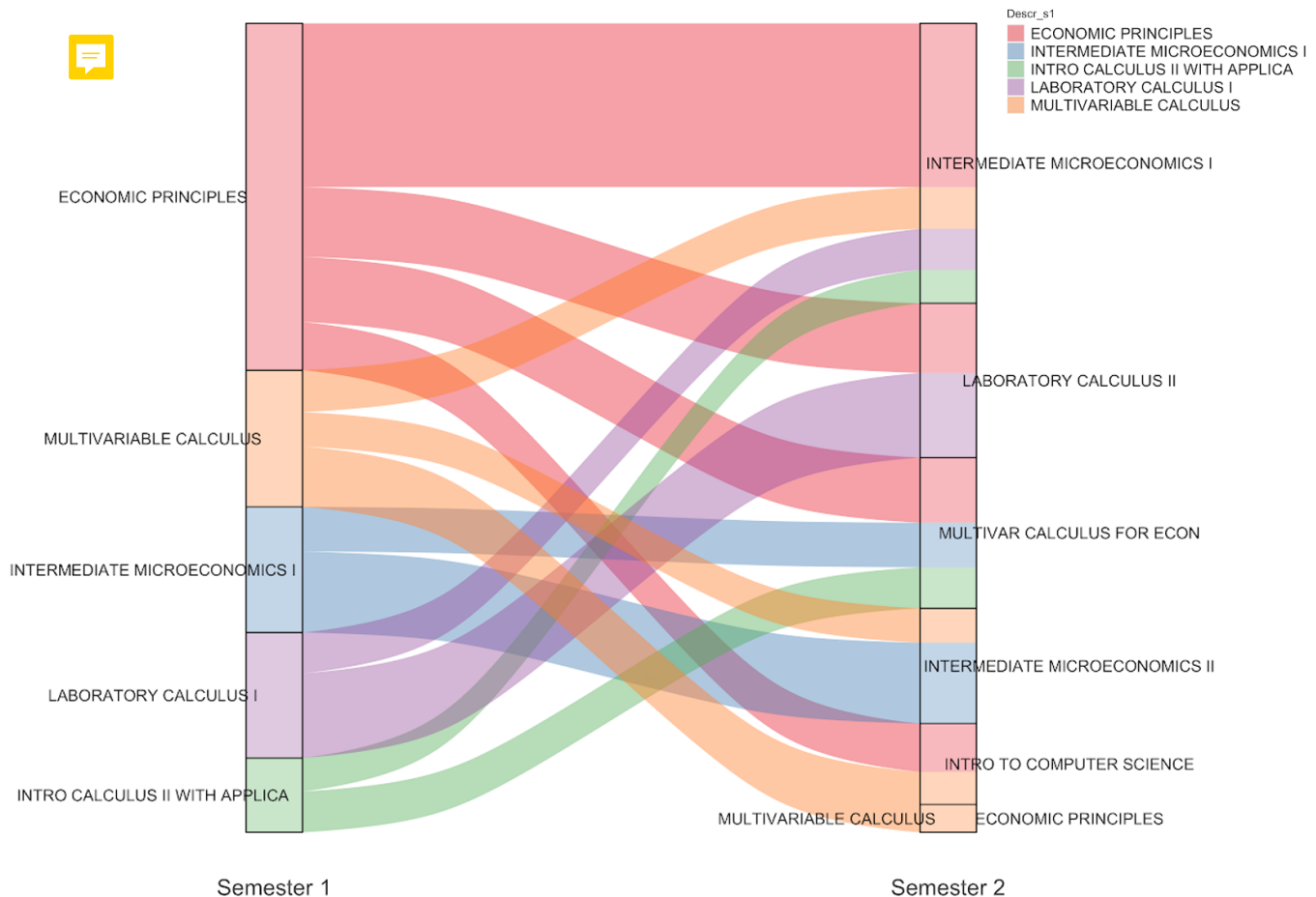


Figure 3. This alluvial plot for economics majors shows common course progressions from the first to the second semester. Though this is just an image for demonstration purposes, the actual implementation is more interactive and enables hovering for tooltips and selecting to highlight pathways. Based on the thickness of the pink bar, we can see that the most popular course for economics majors to take their first semester (First Year Fall term) is Economic Principles. Following the thickest bar, we can see that it is then most common to take Intermediate Microeconomics I during the second semester (First Year Spring term). On the other hand, some economics majors then take Intro to Computer Science their second term, but it is a much less popular pathway. Overall, the alluvial plot shows common course-taking patterns from semester to semester to highlight major-based academic trajectories.

One of the main advantages of alluvial plots is that they are more conducive to highlighting pathways over time. The flow of bars from left to right is more intuitive in terms of timeline in comparison to the circular pattern of network plots. Furthermore, they show the scale of each

course progression through the thickness of bars, providing a measure of popularity or commonness. This can help identify more common versus less common course progressions. For these reasons, it is helpful in identifying common academic pathways and is very beneficial for research purposes.

Despite some of these advantages, the primary disadvantage of alluvial plots is that they have a much less common or familiar plotting style, making them less interpretable to the average person. Thus, we chose not to include them in our final deliverable for advisors and students. In terms of making information ingestible and usable for advisors and students, feedback from advisors allowed us to conclude that this course progression information would be helpful but presented in a different manner. Future work could include the addition of a dashboard that would present course progression information in a more ingestible manner for students and advisors.

Neighborhood-based Recommendation Model

Recent Work

In the last decade, collaborative filtering (CF) has been an increasingly common method for developing recommendation systems. Clustering has also been one of the extensively used concepts in CF. B. M. Sarwar et al. (2002) argued that clustering improves the performance of recommendation, and P. Adamopoulos (2014) proposed a new probabilistic neighborhood-based approach as an improvement of the standard k-nearest neighbor algorithm. It is based on classical metrics of dispersion and diversity as well as on some newly proposed metrics. P. Knees et al. (2014) proposed a normalization technique called mutual proximity in the nearest neighbor selection phase to rescale the similarity space and symmetrize the nearest neighbor relation. They prove that incorporating normalized similarity values into the neighbor weighting step leads to increased rating prediction accuracy.

One of the major factors in CF that greatly influences the recommendation accuracy is the selected similarity measure. Almost all previous works are based on the well-known Pearson correlation measure. However, Pearson correlation does not take into account users' preferences. Sparsity is another common problem that contributes to generating incorrect recommendations. In addition, using a large dataset requires more time for computing similarities among users in order to build an effective neighborhood for the active user. Moreover, in our case, class recommendations must consider the temporal nature driven by the semester scheme. Consequently, combining these factors increases the prediction accuracy and finds the neighborhood where students share the same interests and take similar classes over time.

Proposed Approach

In order to overcome the problems of sparsity and temporal aspects, we propose a cosine similarity neighborhood method (Figure 4). This method consists of two primary parts, the identification of the class-to-class relationship followed by the student-to-class relationship, which are discussed in detail below.

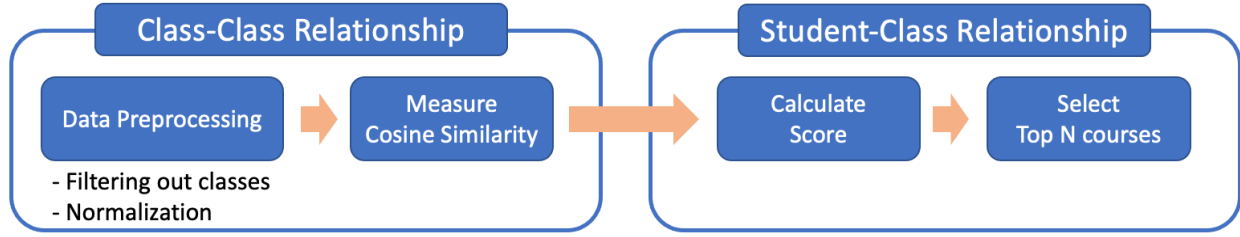


Figure 4. Brief overview of the cosine similarity neighborhood method for the course recommendation model.

Class-Class relationship: Before measuring the relationship between classes, we preprocess our data by filtering out classes taken less than N students and normalizing student's class selection. Then, we measure the relationship between classes using cosine similarity. With these two steps, we construct a new class by class matrix containing the weights (relationships) between each of our classes where a perfect correlation between classes equals 1 and no correlation at all equals 0.

- Data was preprocessed according to the following steps:
 - **Filtering out classes taken less than N students:** Selecting N depends on the size of data. In our dataset, when N is 200, 24 classes remained among the original 400 classes, and the recommendation model performed the best. Notably, the fewer classes, the better the performance will be because the search space is smaller.
 - **Normalizing student's class selection:** The magnitude of all the student's class selection was calculated by taking the square root of the sum of the squares of all the student's class selection. This is done for all classes a student takes, noted here as x, y, and z.

$$magnitude = \sqrt{\{(x^2 + y^2 + z^2 + \dots)\}}$$

We then create the new unit vector by dividing the class selection by the magnitude:

$$vector = \frac{x}{magnitude}, \frac{y}{magnitude}, \frac{z}{magnitude}'''$$

- **Measuring cosine similarity between classes:** The dot-product of the different class-vectors is then divided by the product of the normalized vectors from the previous step. We then calculate the normalized vector based on Euclidean distance (L2-norm) of that vector, which means the square root of the sum of the absolute values squared. This is detailed in the following equation:

$$\cos(\theta) = \frac{X \cdot Y}{||X|| \times ||Y||} = \frac{\sum_i X_i Y_i}{\sqrt{\sum_i X_i^2 \sum_i Y_i^2}}$$

Student-class relationship: Given that the similarity matrix represents the **class-class relationship**, we can calculate each student's likes and similarities by calculating a score for each class. Using these scores, recommendations can be generated by taking the top n courses with the highest score. To calculate the score, we use the following formula:

$$S(u, i) = \frac{\sum_{j \in N} W_{ij} r_{ui}}{\sum_j |W_{ij}|}$$

u = student
 i = course
 j = course taken
 W = weights (0 or 1)
 r = item-based matrix

The score for user u and item i is calculated by summing together all the weights for that item W_{ij} multiplied with the users rating for that item r_{ui} . We then divide by the sum of all the weights for that item W_{ij} . With this formula, we obtain a list of recommended classes. The highest score is the class that his/her most similar neighbors (students) have taken.

Accuracy: We define accuracy as the number of courses actually taken in the top K recommendations divided by the total number of courses taken during the selected semester. For example, if provided 10 recommended courses and the student took 2 of those recommendations out of 3 total classes, then the accuracy would be 2/3 or 67%.

$$Accuracy = \frac{Number\ of\ Courses\ Taken\ in\ Top\ K\ Recommendations}{Total\ Number\ of\ Courses\ Taken\ of\ the\ Academic\ Semester\ of\ Interest}$$

Modularity: In order to measure multiple connections among classes, we use a quality function known as *modularity* (Newman and Girvan, 2004), which is the most widely used to quantify the strength of community structures in the network literature (Fortunato, 2010). For a weighted and directed network and a cluster partition $C = \{C^{(1)}, C^{(2)}, \dots, C^{(K)}\}$ of the classes into K clusters, the modularity of the partition can be calculated by generalizing the formula given in Leicht and Newman (2008). This gives

$$QC = \frac{1}{w} \sum_{i \in V} \sum_{j \in V} (w_{ij} - \frac{w_i^{out} w_j^{in}}{w}) \delta(C_i, C_j)$$

where w_{ij} is the (i, j)th entry of the weighted adjacency matrix W , $w_i^{out} = \sum_j w_{ij}$ and $w_j^{in} = \sum_i w_{ij}$ are,

respectively, the weighted in- and out-degrees of class i and j , and $w = \sum_i \sum_j w_{ij}$ is the total weight

of the network. Finally, $\delta(C_i, C_j)$ is the Kronecker delta function, which is equal to 1 if both

i and j belong to the same cluster and zero otherwise. Intuitively, this equation might be understood as the sum over the proportion of within-cluster transitions minus what would be expected under a random mixing model, where students move randomly from class to class given the constraint that the weighted in- and out-degree of each class is preserved. A modularity of zero indicates that the classification scheme is no better than the random mixing model, while high values of Q indicate a tendency of transitions to be confined within the classes above what is expected by the random mixing model. In other words, networks with high modularity have dense connections between the nodes within clusters but sparse connections between nodes in different clusters.

Results

We compare the performance of the original dataset for economics majors ~~from 2013-2020~~ with the performance of the dataset after filtering out less popular classes. We do not consider recommending classes for the first year, because first year students have taken very few to no classes. Because the recommendation system requires past course-taking history, it is unfit for first year students. This is known as the “cold start problem,” where recommendation systems are unable to perform well because new users have little to no data (Charu, 2016). After filtering out less popular classes, this model performed an average of 26.8% better than generating recommendations from the original dataset (Table 1). In every semester except Fourth Year Spring term, the model performed significantly better with the filtered data than with the unfiltered data (Table 1).

k=10	2Y Fall	2Y Spring	3Y Fall	3Y Spring	4Y Fall	4Y Spring	Average
Original	25.4%	22.9%	9.1%	6.5%	14.4%	46.2%	20.8%
Filtering out less popular classes	60.8%	49.7%	51.1%	50.7%	38.6%	33.8%	47.6%
Change	+35.4%	+26.8%	+42%	+44.2%	24.2%	-12.4%	+26.8%

Table 1. Comparing filtered and unfiltered data results of the neighborhood-based recommendation model on the data subset with economics majors from 2013 to 2020 using 10 recommendations. The original data is unfiltered, while the filtered data indicates the subset where courses taken by

less than 200 students from 2013 to 2020 were removed. This helps decrease the search space for classes and thus improves model performance.

We also compare the average performance of four different but very common majors from 2013-2020, including Economics, Computer Science, Biomedical Engineering, and Political Science. In order to test model generalization performance to more majors, we also test performance on a dataset containing students in the top 5 majors: Biology (BS), Economics (BS), Public Policy (AB), Biomedical Engineering (BSE), Computer Science (BS). Comparing performance among the four different chosen majors, Computer Science performed highest with an average accuracy of 51.2%, and Political Science performed lowest at 45.3% (Table 2). However, overall performance for the four majors was relatively similar. Performance for the subset of data with the top 5 majors was slightly lower at 38.2% (Table 2). These results indicate that the model performs better within a specific major rather than generalizing to even a smaller subset of majors.



To further compare majors, we measure modularity, which measures how densely structured each network is. The modularity of Computer Science is 0.4 and the modularity of Political Science is 0.3 (Table 3). Though we cannot determine if there is a causal relationship between modularity and accuracy, we see a clear positive relationship between accuracy and modularity. This relationship may indicate that majors with higher modularity may have higher accuracy and be more fit for the recommender system. For example, Computer Science majors may have clearer academic trajectories than other majors, which leads to higher modularity and thus better prediction accuracy than other majors. This concept could be further tested in the future.

k=10	Economics	Computer Science	Biomedical Engineering	Political Science	Top 5
Accuracy Average	47.6%	51.2%	50.7%	45.3%	38.2%
Modularity	0.32	0.40	0.38	0.30	0.25

Table 3. Comparison of average accuracy and modularity for four majors of choice and the subset of top 5 most common majors.

Limitations

Though the cosine similarity neighborhood method has relatively higher prediction accuracy than other methods, there are several limitations. First, the cosine similarity neighborhood method cannot consider multiple features such as grade. Second, since it needs to compute a very large item-item matrix, it is very expensive and inefficient to compute and recommend courses in actual applications, especially if it is used for real-time recommendation.

In terms of measuring model performance for recommender systems, it is difficult to gauge which evaluation metrics are fit for the problem, and thus, which models perform better. In practice,

recommender systems can be assessed using A/B testing, but that is not feasible in our case. Furthermore, just because a student did not take a class, it does not mean that they did not like it. Likewise, if the student did end up taking the course, they still may not have liked it. These complications make it extremely difficult to determine if a recommendation engine is performing “well.” In our project, we chose to use a custom accuracy metric because we had data on the classes the students actually ended up taking. However, future work could implement other performance metrics or even A/B testing to better gauge model performance.

In addition, there is a limited number of courses students can take each semester. For example, the average student takes four courses each semester. Because our model attempts to accurately capture these 4 courses, it is relatively difficult to achieve a high accuracy given these minimal data points. Furthermore, students tend to lighten their load in their senior year, and frequently take fewer classes. This means that using our accuracy metric, we expect a low overall accuracy. Lastly, there is an extreme amount of variability in course-taking patterns even amongst students of the same major. This makes it difficult to accurately capture such a wide range of classes. However, many of these challenges are innate to the problem, and despite these limitations, this neighborhood-based recommendation engine consistently performed better than other recommendation models tested.

Other Recommendation Models



In addition to the neighborhood method, another common set of recommendation algorithms are latent factor models (LFMs). LFMs are a state-of-the-art method for model-based CF (Charu, 2016). These models assume that there is an unknown low-dimensional representation of users and items with which the user-item affinity can be accurately modeled (Melville, 2010). For example, the grade the student receives from a course might be assumed to depend on a few implicit factors such as the student's preference across various course subjects. Matrix factorization is a class of widely successful LFMs that attempt to find weighted low-rank approximations to the user-item matrix (Melville, 2010). In this project, we have attempted two types of matrix factorization based LFMs: singular value decomposition (SVD) and LightFM.

SVD

Singular value decomposition, SVD, is a very popular matrix factorization technique for recommender systems. However, it is most commonly used for explicit datasets where user feedback such as ratings is available (Melville, 2010). In the context of our problem, the student grades are treated as a proxy for explicit rating. Notably, grades are much different than an actual rating for the course, as they do not show a student's like or dislike of the class. However, in the absence of course ratings, we used grades instead. Thus, this model will tend to recommend courses where the students are most likely to receive a higher grade. SVD works according to the following equation: $R = U\Sigma V^T$, where R represents the predicted student grades, U consists of the current student grades, Σ is the diagonal matrix of singular values (weights), and V^T represents courses.

LightFM

LightFM is ~~one established~~ hybrid matrix factorization model. Hybrid recommendation algorithms combine both the concepts of content-based recommendation models and collaborative filtering (Melville, 2010). It learns **embeddings** for users and items to encode user preferences for items. Multiplying the encoded user matrix and item matrix together produces scores for every item for a given user. It is able to solve the “cold start problem”, where recommender systems struggle to make recommendations for new users due to the absence of past data. Hybrid models solve this issue by utilizing similarities based on user features to help generate recommendations for new users.

Comparison

The two LFM attempted both performed **consistently lower** than the neighborhood method. This may be due to the limited number of course or student features in the dataset. Alternatively, the accuracy metric used to gauge model performance does not cater well to LFM, resulting in low accuracy. Furthermore, it is likely that the use of grades as a proxy for ratings could have drastically skewed recommendations. Lastly, it was ~~essentially~~ impossible to analyze the vectors created in the latent space, making it difficult to determine why these models performed poorly.

A **comparison of the three primary models**, neighborhood method, SVD, and LightFM, is provided in table 4. Though SVD and LightFM present some advantages over neighborhood methods, actual implementation and testing of these models revealed that the neighborhood model with cosine similarity consistently performed much better than the other two models. Thus, we chose the neighborhood model as the final recommendation algorithm.

	Neighborhood method	SVD	LightFM
Pros	<ul style="list-style-type: none">• Much higher prediction accuracy• More consistent performance	<ul style="list-style-type: none">• Grade prediction• Scalable• Effective for sparse data	<ul style="list-style-type: none">• Incorporate metadata of courses and students• Able to solve cold start problem
Cons	<ul style="list-style-type: none">• Requires large offline storage	<ul style="list-style-type: none">• Very low prediction accuracy	<ul style="list-style-type: none">• Very low prediction accuracy

Table 4. Comparison of the three major recommendation engine models.

Future Work

To further aid undergraduate students in making more informed decisions for major and course selection, additional data, including course descriptions, syllabi, professors, prerequisites, major requirements, and timing should be gathered and integrated into the current dashboards. To better track academic trajectories, an additional page should be added to the dashboards that presents course progression information in a more ingestible manner for students and advisors. This could

show common courses to take next or courses that are commonly taken before higher level courses to help students plan course order. Future implementations could also identify combinations of classes that are commonly taken together leading to higher or lower grades, which could help with course load planning. Currently a prototype, the course and major planning dashboards can be further improved and fine tuned to be fully integrated with the current DukeHub registration system.

To improve the performance as well as the practical applications of the recommendation model, one potential direction is to investigate other representations of courses using advanced techniques, such as word embeddings, to identify cluster structures and similar classes across different domains. This could potentially provide new information that could improve model performance. To better evaluate model performance, other metrics such as relevance and diversity metrics in addition to our customized prediction accuracy could be used to better gauge the strengths and weaknesses of each model.

Overall, there are several improvements and additions that could be made to both the dashboards as well as the recommendation engine. However, our current work shows promising potential for the usability for a course planning tool, and future work can continue to build on these deliverables to develop a more thorough and robust tool.

Conclusion

The lack of recommendation systems in the academic space as well as the clear need for increased guidance in course and major planning for undergraduate students emphasizes the potential for a course and major planning tool. This tool enables students to easily filter through the massive amount of classes and majors, choose classes that align with their desired pathways, and better plan their undergraduate curriculum. Advisors could also benefit from these products to better understand course-taking patterns as well as guide students through their undergraduate careers. Overall, there are several clear benefits in developing a system to aid students in course and major selection.



Working towards the goals of tracking academic trajectories and developing a course recommender engine, we have successfully created three major deliverables. First, the integrated planner for major and course selection is easily accessible, user-friendly, and ingestible for both students and advisors. It allows for extensive filtering based on students' course history and personal preferences, which is especially useful for first year students who may not have a clear direction. Second, the visualizations of academic trajectories, including the network and alluvial plots, help identify curriculum flows for each major, while highlighting common prerequisites for courses between consecutive semesters. Lastly, the course recommender system provides more personalized suggestions by finding the neighborhood where students have taken similar classes over time. All of these products serve to provide more information to facilitate the major and course selection process for both the undergraduate students and the academic advisors at Duke University.

References

1. B. M. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering," in Proceedings of The fifth international conference on computer and information technology, 2002.
2. Charu, C. A. (2016). *Recommender Systems: The Textbook*.
3. Few, S. (2006). *Information dashboard design: The effective visual communication of data* (Vol. 2). Sebastopol, CA: O'reilly.
4. Jiang, W., Pardos, Z. A., & Wei, Q. (2019, March). Goal-based course recommendation. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge* (pp. 36-45).
5. Lee, Y., & Cho, J. (2011). An intelligent course recommendation system. *SmartCR*, 1(1), 69-84.
6. P. Adamopoulos, "On Discovering non-Obvious Recommendations: Using Unexpectedness and Neighborhood Selection Methods in Collaborative Filtering Systems," in Proceedings of the 7th ACM international conference on Web search and data mining, New York, NY, USA, 2014.
7. P. Knees, D. Schnitzer and A. Flexer, "Improving Neighborhood-Based Collaborative Filtering by Reducing Hubness," in Proceedings of International Conference on Multimedia Retrieval, New York, NY, USA, 2014
8. Main, J. B., & Ost, B. (2014, January 24). The impact of letter grades on Student Effort, COURSE selection, and Major choice: A Regression-Discontinuity Analysis.
9. McFarland, D. A. (2006). Curricular flows: Trajectories, turning points, and assignment criteria in high school math careers. *Sociology of Education*, 79(3), 177-205.
10. Melville, P., & Sindhvani, V. (2010). Recommender systems. *Encyclopedia of machine learning*, 1, 829-838.
11. Verbert, K., Duval, E., Klerkx, J., Govaerts, S., & Santos, J. L. (2013). Learning analytics dashboard applications. *American Behavioral Scientist*, 57(10), 1500-1509.
12. Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2), 026113.
13. Friedkin, N. E., & Thomas, S. L. (1997). Social positions in schooling. *Sociology of Education*, 239-255.
14. Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*.
15. Chase, I. D. (1991). Vacancy chains. *Annual review of sociology*, 133-154.
16. Sørensen, A. B. (1987). The organizational differentiation of students in schools as an opportunity structure. In *The social organization of schools* (pp. 103-129). Springer, Boston, MA.

Appendix

Data Dictionary

- Calculation ID: anonymized student ID
- Acad Plan: shorthand/abbreviation for major/minor/certificate/secondary
- Acad Plan Descr: description of acad plan (major/minor/certificate/secondary)
- Plan Type: denotes major/minor/certificate/secondary
- Degree: abbreviated type of degree (AB, BS, BSE)
- Descr_completions: type of degree (Bachelor of Arts, Bachelor of Science, Bachelor of Science in Engineering)
- Comp Term Descr: academic year and term of graduation
- Acad Year_completions: academic year of graduation
- Subject: subject of course
- Catalog: course number
- Descr_enrollment: name of course
- Grade: grade received in descr_enrollment
- Term Descr: academic year and term of enrollment in class
- Acad Year_enrollment: academic year of course enrollment
- Enrollment Start: first year student has registered for classes at Duke
- Term Year: year student is taking course
- Semester Term: term student is taking course (Fall, Spring, Summer 1, Summer 2)
- Class Year: indicates freshman through senior year (or more) and term (First Year Fall Term, First Year Spring Term, etc.)