ABSTRACT
The processes through which course selections accumulate into college pathways in US higher education is poorly instrumented for observation at scale. We offer an analytic toolkit, called Via, which transforms commonly available enrollment data into formal graphs that are amenable to interactive visualizations and computational exploration. We explain the procedures required to project enrollment records onto graphs, and then demonstrate the toolkit utilizing eighteen years of enrollment data at a large private research university. Findings complement prior research on academic search and offer powerful new means for making pathway navigation more efficient.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
Course Sequences; Academic Pathways; Graph Visualization

INTRODUCTION
US higher education is unique in the world in the extent to which schools expect undergraduates to explore a variety of courses before committing to a field of study. In contrast with virtually all other national postsecondary systems, in which students enter schools and programs with relatively structured curricula, US undergraduates often are encouraged to explore a variety of academic options through an iterative course search and selection process. Following prior authors [1, 3], we call a student’s eventual sequence of course elections a pathway. At present, pathways are poorly instrumented for observation by students, educators, administrators and researchers [11].

While enriching when successful, the contingencies of course selection are not well understood and are almost always fateful. Ethnographic research suggests that students tend to select courses based on partial information inflected by logistical constraints and personal preferences that have little directly to do with academics [29, 34, 35]. Absent conscientious design and signposting, students can easily spend time, credit hours, and tuition accumulating courses that do not lead efficiently to majors and completion [3]. The work of instructors, department chairs, deans, and budget officers is similarly constrained by lack of clarity about how pathways accumulate.

Fortunately the information necessary to observe pathways systematically and at scale is present in all colleges and universities in the form of academic transcripts. Transcripts are the official records documenting courses accumulated by each student as he or she makes academic progress. Yet transcripts typically are housed in tables of databases to which few have access, and in their “raw” form exceptionally opaque to interpretation. Most schools have specialized staff who generate specifically requested reports for individuals in high-level academic positions1, but these personnel typically cannot interact directly with all the parties in need of insight at varying levels of detail.

This paper presents Via, an analytic toolkit we have built to observe and understand undergraduate pathways utilizing de-identified transcript data held by a large private research university. Our approach is to provide a zoomable, investigative instrument for large-scale, qualitative and quantitative investigations of pathways. Built on graph theory, Via provides a visual interface for observing the course sequences embedded in tens of thousands of transcripts. Via’s grounding in graphs allows us to bring powerful mathematical computations to bear on the problem of pathway evolution.

Graph approaches have been applied to a wide variety of other tasks, such as detecting communities [17], collecting materials for survey articles [25], augmenting collaborative recommendation records [22], predicting future collaborations between

1We acknowledge here our own version of such a unit, which has provided us with numerous insights into both the information needs, and data semantics at our institution.
After related work we introduce Via and examples of its application. We then explain the underlying computations that make the visualizations possible. Section Advanced Applications presents applications that rely on Via’s formal graph underpinnings.

RELATED WORK
The dangers of choice overload are well documented. Research in marketing [9] and psychology [36, 23] has demonstrated both the stated preference for choice, and improved factual success when choice is limited. In higher education, Baker et al. [5] find that simply reducing choice is not an acceptable answer. While students do not have strong reactions to increased guidance, they react negatively to a reduction in choice. Thus enriching guidance is one promising approach. The author of [41] shows that simulations and predictions based on student models can successfully identify points where advisors could intervene to improve graduation outcomes.

Educational outcomes in community colleges are negatively affected by “chaotic enrollment patterns” [12, 4, 37]. A number of these institutions have explored options for improved support by providing “guided pathways” [24], but without the benefit of graph visualization and other computational techniques. A few comprehensive universities have built carefully instrumented “early alert” and other advising tools utilizing institutional data [15], but the computational techniques underlying these tools remain opaque.

Across the social and physical sciences, graphs are used to visualize, simplify and facilitate computational analysis of complex dynamic systems. Graphs have been applied to model a diverse set of networks such as food chains [20], the human genome [32] and ecological systems [16]. Social scientists have long used graph methods to study systems-level phenomena [8]. The use of graphs within the social sciences was particularly spurred on by the insight that human societies could be structured like biological systems. The nineteenth century French philosopher, Emile Durkheim, argued for instance that social regularities could be found in the structure of social environments in which they were embedded [13]. By studying systemic regularities it is possible to derive macro-level insights about the structure of many micro-level interactions.

Since the mid-1950s, graphs have been applied to model the flow of information in social and professional networks. As part of the MIT Group Networks Laboratory, Leavitt et al. observed how the structure of interpersonal relationships between groups of coworkers facilitated the spread of information throughout a team of colleagues [27]. More recently, similar methods have been applied to study how workplace professional networks influence the spread of information through company email chains [14]. The modeling of knowledge transmission parallels closely how graph-theoretic methods have been applied to understand social dynamics in the field of education. Studies of citation networks, for instance, describe how intellectual advances spread through academic space [6, 18].

A number of the challenges posed in representing citation networks, such as learning optimal edge weights, are directly applicable to the problem of modeling course sequences. For citation networks, it is useful to modulate the edge strength between two papers in order to represent the relative influence of a cited paper. Batagelj proposes solutions to this problem by introducing SPC weights on each edge of the network to capture the incoming and outgoing “flow of information” for a given paper [6]. Hajra et al. also observe aging phenomena among papers in which the probability that a paper is cited decays with time at an exponential rate proportional to \( t^{0.9} \) up to ten years after its publication date [19]. Course selection may display a similar exponential time-dependent probability. Moreover, just as citation networks provide a framework for modeling the flow of knowledge within academia, course sequence networks imply shared and prerequisite knowledge between courses.

Also analogous to our effort is recent work on social connectivity in massive open online courses (MOOCs). Large online education providers, such as Coursera and EdX, use thread messaging boards to facilitate collaboration and exchange among students. In these systems, typically called "forums," learners can post questions or comments to start a thread or add to existing ones. Data gathered on such forums enables researchers to study such things as exchange of information among learners [10], the influence of particular learners on the course participation of others [40], and patterns of civic discourse within and across forums [33].

Sinha et al. model each participant within a particular Coursera class as a node and draw a directed edge from a thread or sub-thread initiator to any of the people who engaged in that discussion [39]. In doing so the authors are able to represent the flow of topics and engagement within the social dynamics of a certain course. Sinha et al. were able to analyze measures of degree and betweenness centrality on this model of learner engagement to understand the influence of discussion initiators on the overall collaboration of learners in the class. The parallels are clear if we view discussion topics as self-contained units of knowledge that are part of a course.

Similarly to our network construction, Sinha et al. use a gradient coloring of nodes to indicate its relative betweenness centrality. Zhu et al. also model the engagement of learners in online forums on a week-by-week basis using Exponential Graph Models (EGM) [43]. The use of EGMs allows the authors to model a learner’s participation for a given week by incorporating their performance for both the previous and subsequent weeks. While inspired by EGMs, our model does not directly include aggregate network statistics, in order to compute node and edge properties. Instead, we model the probability of two courses being taken sequentially given data in which the courses may have been taken several academic terms apart. Finally, NetworkSeer uses a similar modeling
framework as in the previous papers but additionally models individual learners’ demographic information within the course discussion thread [42]. Although we do not have direct access to learner demographic data, our model uses final major to observe differences in course-taking behavior.

The social dynamics of learner participation in MOOCs and the spread of knowledge through citation networks are the closest parallels to modeling undergraduate course election in US universities. Courses can be observed as distinct units of information that share overlap and prerequisite knowledge with other courses. Academic publications and forum discussions are similarly self-contained units of knowledge that build upon and interact with other publications and discussion threads, respectively. Given a dearth of direct research in course sequence networks, our Via toolkit builds upon the modeling frameworks of citation graphs and MOOC social dynamic networks. In contrast, course embeddings [31] represent courses and enrollments as vectors, which are then clustered and otherwise manipulated.

INTRODUCING VIA

Here we introduce the minimum of underlying computation that is needed to understand Via graphs. Details follow in Section Methodology.

We use student data from a large US private research university. This dataset contains the anonymized enrollment data of over 52,000 students who were enrolled at the university during any time between Fall 2000 and Fall 2018. Each of the roughly two million rows in this table consists of a unique student identifier, a course in which they enrolled, and the academic term during which they enrolled in the class. The dataset also contains supplementary information, including each student’s major during time of enrollment in a course and each student’s major upon graduation.

Depending on the class of problems of interest, we filter along these additional values. Similar filters could be used with Via over demographic data that are available to university administrators. We did not have such data available, but hope that our examples below suggest the significant additional potential for Via to inform understanding of gender, first-generation, ethno-racial minority status or other relevant subgroup characteristics.

Via converts course enrollment data to directed graphs. Nodes represent courses, directed links represent sequential enrollment in the courses at the link endpoints. Link weights model conditional probabilities of enrolling in the link destination course, given prior enrollment in the course at the link origin. Link weights are visualized as line opacity.

Via partitions the graphs by the departments with which courses are associated.

Tool users can zoom to select particular courses or departments. The selections can then be extracted as a separate graph. These interactions are implemented by Cytoscape, which we use as the engine behind Via [38].

Cytoscape was developed to model biological systems and interactions between cellular organisms. Yet the tool is well suited for representing our academic pathway data.

Cytoscape uses a spring-embedder system for optimally spacing and aggregating nodes, and additionally allows the tool operator to cluster nodes by attributes [7]. For our purposes, we display courses within departments as ring structures. For example, the large ring in Figure 1 represents the Computer Science department in a graph generated from enrollment data filtered to include only CS majors. Each of the dots that comprise the circumference of the ring is one course in that department. The interior connecting arrows display the flow of students from one CS course to another. Links between rings are enrollments of CS majors in other departments’ courses.

Via assigns unique colorings for both courses (nodes) and enrollment probabilities (edges) to visualize course and enrollment-level attributes. In all of the figures below, we color edges between courses in the same department orange, and all edges between courses in different departments grey. We further modulate the opacity of an edge from course \(i\) to course \(j\) to represent the conditional probability of taking course \(j\) after course \(i\). Finally, each node is colored in a blue-gradient, representing the prior probability of a student enrolling in the respective course. For example, a deep blue node would be a course that many students are highly likely to enroll in during their undergraduate years.

Note that while a Via graph aggregates all students—no single student’s information is represented—we can choose to make available to Via only students with certain characteristics. In Figure 1 this attribute is major. We could as easily construct such overviews that distinguish between, for example, gender or minority status.

Beyond visualizations Via provides a second, more formal avenue for analysis. Because the tool’s principle visual element is a graph, many graph related mathematical approaches are available. These include variations of centrality and between-
ness, clustering, Page Rank, and others. Applications in the following section will exemplify the use of the visual methods. Section Advanced Applications will add applications that use graph computations.

VISUAL APPLICATIONS

The following examples illustrate how Via can help student, instructor, and administrator stakeholders make sense of student data.

Student Stakeholders

Departments differ in the degree to which requirements oblige students to enroll in the department’s own courses. A department with fewer formal intra-departmental requirements allows students to more easily sample other areas of interest. Gaining an overview of such differences among many departments is tedious. Prior student pathways tell the story more directly.

In Figure 2 we compare the interconnectivity of the History and Electrical Engineering majors, limited to course enrollments of one academic year apart. The Electrical Engineering department clearly has students take numerous internal courses in sequence, as indicated by the many orange links. History, in contrast, allows students more freedom to take outside courses with few intra-departmental sequence requirements. (Those outlinks are excluded from the Figure.)

Note that this intra-cluster connectivity is equivalent to the graph computational modularity. This metric represents the connectivity of clusters within a graph, by calculating the over-representation of edges among groups of nodes.

As a second use case, we illustrate how Via can be leveraged to discover “try-me” courses. Departments offer such service courses for students majoring in unrelated fields, but who are interested in exploring other areas of study. Via enables quick discovery of such courses.

We find the “try-me” courses by creating a Via graph over students of a single major. We then identify the most popular courses within this graph that are not in that major. For instance, by filtering on the History major, we observe that of the History majors, nearly 28% take CS105, and 21% take CS106A. These trends are visualized through variations in node coloration in Figure 3.

Instructors comprise a second group of stakeholders that can benefit from the easy interpretability of our course sequence visualizations. It is of particular interest to instructors to gain an overview of the sorts of students enrolling in their courses, in order to customize and improve lesson plans [26].

Babad et al. report evidence that the primary reasons students decide to enroll in particular courses are the learning value of a course followed closely by instructional style [2]. Particularly difficult courses are avoided unless they are required. Our assessment of the Computer Science “try-me” courses for History majors corroborates these results. Although CS106A is widely recognized on campus as the most highly-enrolled introductory Computer Science course at our case university, the Computer Science department presents CS105 as a course for non-majors, and also has a reputation on campus as being more accessible. Our Via toolkit is thus able to scaffold more personalized course discovery for particular kinds of students.

Instructor Stakeholders

Instructors comprise a second group of stakeholders that can benefit from the easy interpretability of our course sequence visualizations. It is of particular interest to instructors to gain an overview of the sorts of students enrolling in their courses, in order to customize and improve lesson plans [26].

As an example, we illustrate how Via can be leveraged to discover “try-me” courses. Departments offer such service courses for students majoring in unrelated fields, but who are interested in exploring other areas of study. Via enables quick discovery of such courses.

We find the “try-me” courses by creating a Via graph over students of a single major. We then identify the most popular courses within this graph that are not in that major. For instance, by filtering on the History major, we observe that of the History majors, nearly 28% take CS105, and 21% take CS106A. These trends are visualized through variations in node coloration in Figure 3.

In Figure 2 we compare the interconnectivity of the History and Electrical Engineering majors, limited to course enrollments of one academic year apart. The Electrical Engineering department clearly has students take numerous internal courses in sequence, as indicated by the many orange links. History, in contrast, allows students more freedom to take outside courses with few intra-departmental sequence requirements. (Those outlinks are excluded from the Figure.)

Note that this intra-cluster connectivity is equivalent to the graph computational modularity. This metric represents the connectivity of clusters within a graph, by calculating the over-representation of edges among groups of nodes.

As a second use case, we illustrate how Via can be leveraged to discover “try-me” courses. Departments offer such service courses for students majoring in unrelated fields, but who are interested in exploring other areas of study. Via enables quick discovery of such courses.

We find the “try-me” courses by creating a Via graph over students of a single major. We then identify the most popular courses within this graph that are not in that major. For instance, by filtering on the History major, we observe that of the History majors, nearly 28% take CS105, and 21% take CS106A. These trends are visualized through variations in node coloration in Figure 3.
work that represents courses as nodes and course relationships as weighted edges.

Figure 5 illustrates the process. We begin with a matrix $M$ representing student enrollment in courses over time. We can interpret this matrix $M$ as the adjacency matrix of a bipartite, labeled graph documenting the interaction of student nodes with course nodes. We end with a matrix $P$ that defines a projection of the courses in $M$, where edge weights signify how strongly two courses are linked. This projection can be visualized as a network such as the one found in Figure 1, which maps all course interactions in a way that describes the aggregate behavior of student enrollment patterns. In order to preserve the normal, sequential nature of courses across academic years, we omit the Summer Period from all experiments.

Combined with an existing graph manipulation kit—in our case Cytoscape [38]—the graph described by $P$ serves as our target toolkit for university stakeholders. We now show how we endow the construction process of $P$ with flexibility that can tune the resulting graph for visualizations that help answer a variety of questions.

As per the overview above, the first step is to prepare a sequence matrix. The second involves the calculation of the projection. The following sections merely formalize the overview.

**Sequence Matrix Generation**

Via receives data in the form of tabular student enrollment records. The input to the projection algorithm is a matrix $M$ of shape $(|S|, |C|)$, where $S$ is the set of all students and $C$ is the set of all courses. A given entry $M_{ij}$ represents the time when a student $i$ enrolls in some course $j$. For example, an entry $M_{ij} = 1$ signifies that student $i$ enrolled in course $j$ during their first academic term, generally quarter or semester. An entry $M_{ij} = 0$ implies that $i$ never enrolled in $j$. This thus implies that each row $m_i$ represents the entire course enrollment history of some student $i$. We generate various forms of matrix $M$ from the raw enrollment data by filtering on different student attributes. The nature of the filter depends on the questions...
being asked. For example, if we are only interested in those who majored in the Humanities, we would limit \( M \) to this student subset. Another filter one might apply at this step is year of enrollment, if only that time slice is of interest. The more filtering is applied to \( M \) the faster the subsequent processing, and visual interaction response.

Our data provides information on each enrollment’s time, the enrolling student’s major at the time of enrollment, and the student’s final major. Depending on data availability, filters based on gender, status as underrepresented minority, or college entrance scores could be applied as well.

**Graph Projection**

Next, given the sequence matrix \( M \), we generate the matrix \( P \) of shape \((|C|, |C|)\). This matrix represents the adjacency matrix for a one-mode projection of the bipartite network specified by matrix \( M \). We weight each edge in \( P \) using conditional probabilities that describe how likely one is to take one course given another course. Thus, entry \( P_{ij} \) represents the conditional probability of taking course \( j \) given that one has taken course \( i \). \( P \) aggregates students, losing some information, but gaining the probability of moving from one course to another.

Matrix \( P \) is thus a representation of how even non-adjacent course nodes in \( M \) interact, based on the nature of student enrollment. We use this matrix \( P \) as the basis of all calculations and visualizations. The parameters for the conditional probabilities are fit based on counts determined in the calculation of intermediary matrix \( \tilde{P} \) of shape \((|C|, |C|)\). We calculate each entry \( \tilde{P}_{ij} \) by accumulating the occurrences of course \( i \) taken at some point before course \( j \) across all students in set \( S \):

\[
\tilde{P}_{ij} = \sum_{s=1}^{[S]} \mathbb{1}\{M_{si} - M_{sj} \geq 0\} \ast \lambda^{M_{si} - M_{sj}}
\]

where \( M_{si} - M_{sj} \) is the academic timestep delta (commonly measured in semesters, or quarters) between a student \( s \) ’s taking course \( i \) and course \( j \). Function \( \lambda \) is a second point in the visualization construction where flexibility is provided. The function may be chosen to de-emphasize the connection between subsequent courses.

For example, in an analysis of degree completion we may be interested in cases when course offerings were taken as closely together as possible. In contrast, when planning curricula we may wish simply to learn the probabilities of two courses taken in sequence, even if several academic terms apart.

Function \( \lambda \) may be chosen to be continuous or discrete. For example, the following choice attenuates the relationship between two courses through an exponential decay over enrollment term distance:

\[
\tilde{P}_{ij} = \sum_{s=1}^{[S]} \mathbb{1}\{M_{si} - M_{sj} \geq 0\} \ast \lambda^{M_{si} - M_{sj}}
\]

where, \( \lambda \) is a constant that controls decay rate. Intuitively, this type of function gives more weight to close course-pair enrollments than temporally distant course-pair enrollments.

Alternatively, we may choose a function \( d \) that tallies a relationship between two courses only when course \( j \) immediately follows course \( i \). In the following example, elapsed time between two courses beyond one term would sever the relationship:

\[
d = \begin{cases} 1, & M_{si} - M_{sj} = 1 \\ 0, & M_{si} - M_{sj} > 1 \end{cases}
\]

More elaborate discrete functions could amplify course relationships of particular time distances.

Using \( \tilde{P} \), we calculate the final projection matrix \( P \). In some experiments, we set \( P = \tilde{P} \), in order to gain access to a raw count projection matrix. In other experiments, we set \( P \) to be a matrix of conditional probabilities between courses. In this case, each entry \( P_{ij} \) is a conditional probability \( p(j|i) \) whose parameters are computed using the following closed form expression:

\[
P_{ij} = \frac{\tilde{P}_{ij}}{\sum_{i=1}^{[S]} \mathbb{1}\{M_{si} > 0\}}
\]

\( P_{ij} \) thus represents the proportion of students who take the course sequence: course \( i \) followed by \( j \) out of the total number of students who take course \( i \) at any point. One may note that this bears a resemblance to a Bayesian Network; however, we make the assumption that the process of transitioning from course node to course node is Markovian in nature. This eases implementation but precludes our model from being a true Bayesian Network due to the emergence of cycles.

In the following section we describe how we combine the projection matrix with an existing graph visualization tool. For simplicity we choose the timestep delta function \( \lambda \) to be discrete, although we can alter the nature of the function for different examples.

**ADVANCED APPLICATIONS**

With the underlying data structures explained, it is now clear how Via builds special subgraphs by filtering early, when the student level information is still available. The filtering occurs as matrix \( M \) is constructed. Similarly, the discount function is used to control how closely together we wish enrollments to have occurred in time. Several of the applications above used these techniques.

We next introduce applications that rely on the graph data structure less as a visualization method. Instead curriculum insights are gained from mathematical analyses of the graph that is described by the projection matrix \( P \).

**Course Discovery**

An important use case of our model is predicting prerequisites for courses which have no explicitly listed required classes. Because Via models the interactions between courses as a graph, we can leverage the PageRank algorithm for this task [30]. PageRank measures the relative importance between nodes in a graph by conducting random walks through a network.

\(^2\)At the time of this writing we have not yet provided access to these methods through Via’s user interface.
Yet aggregated data on student behavior suggests otherwise. For the first example, let us demonstrate two examples of how the PageRank algorithm will take subsequently in their college careers: it illuminates pathways based on prior behavior. The random walk in this context is an imaginary new student who is enrolled in a seed course. From there the student randomly enroll in successor courses, guided by the probabilities in the projection matrix, until the student reaches a pre-determined target course $C$.

Importantly, PageRank goes beyond counting enrollments—the incidence of enrollment links into a course node $C$. The algorithm mirrors the probability that a student arrives at $C$ from other courses, which might be topically distant. The algorithm thus takes into account path lengths, not just the immediate enrollment history prior to $C$.

We demonstrate two examples of how the PageRank algorithm grants us insights into implicit prerequisite relationships that exist between courses.

**Which math alternative to choose:** For the first example, let us imagine a student who hopes to take CS200, a course that covers a broad introduction to Machine Learning methods. The course is offered by the Computer Science department in the School of Engineering. In order to take this class, students are required to understand matrix calculus and linear algebra. The two introductory mathematics courses offered to first-year undergraduates are ENGR1 and MATH1. Among these two courses, ENGR1 is offered by the School of Engineering, and is branded as an introductory mathematics course with an engineering focus. Based on the course catalog alone, ENGR1 is ostensibly the introductory math course to take if one is interested in engineering. MATH1 is offered as the more general introductory math course, often taken by students from outside the School of Engineering. Taking ENGR1 over MATH1 would seem to be a reasonable choice for a student seeking to take CS200.

Yet aggregated data on student behavior suggests otherwise. Running PageRank twice, once with the seed course set as MATH1 and once with ENGR1, we find that a student is more likely to complete CS200 after completing MATH1. Contrary to our premonition, this result indicates a stronger sequential relationship between MATH1 and CS200 than between ENGR1 and CS200. We suspect that students, instructors, and administrators all might benefit from illuminations of this sort.

One might argue that such insights could be gleaned from simple database queries over enrollment data to show that MATH1 and CS200 have a higher joint enrollment count than ENGR1 and CS200. When analyses are even mildly more complex than that, however, the approach becomes intractable.

Consider the case in which we are conditioning on more than one course—say a student’s entire course history. The intersection of students with this precise set of courses is small, perhaps too small to derive any meaningful statistical analysis of future behavior. In its use of random walks, PageRank gives us a simple, yet efficient method of approximating common student behaviors given a set of courses already taken. The following illustrates this point.

**De facto Prerequisites:** Consider the common scenario in which a first-year undergraduate finds an upper level course with few or no listed prerequisites. In the following example the course is POLSCI11D, Democracy, Development, and the Rule of Law. Its student composition skews heavily towards third- and fourth-year undergraduates and graduate students. The student wishes to find a course that will prepare her for this upper level offering.

We seed the PageRank algorithm with the four courses that the student has already taken. We set POLSCI11D to be the target of all random walks (enrollments).

We then iterate through the course catalog: at each iteration, we take one course as the candidate for best preparatory course. We append it to the initial four-class seed, ‘pretending’ the student had taken this course. We then run PageRank from the augmented seed set, and note the PageRank score of the target course. Upon loop termination, we select as good candidates the top $k$ courses that led to the highest target course PageRank. See Figure 7 for a pseudo-code implementation of the algorithm for $k = 1$.

In this example, the top candidate course discovered by our algorithm is POLSCI11L, Modern Political Thought: Machiavelli to Marx and Mill. The catalog excerpts in Figure 6 confirm that this course is a good choice. Unlike the target course, 71% of students take the course in their first two years. This example shows how PageRank reaches beyond analyses derived solely from enrollment data queries.

The algorithm features several tunable parameters. For example, a student can weigh their seed set such that some courses have higher influence over the final decision than others. A student who may have recently switched majors may want their queries to skew more heavily towards their new major. In these cases, a student may decide to weigh their seed set such that some courses have higher influence over the final decision than others. Students can also constrain the can-
let max_score = 0
let best_course = None

let initial_classes = [ 'POLISCI1',
  'STATS1',
  'POLISCI2',
  'POLISCI3']

let target_class = 'POLISCI1D'

for course in courses:
  if course is in initial_classes:
    continue
  else:
    initial_classes.append(course)
    score = page_rank(initial_classes, target_class)
    if score > max_score:
      max_score = score
      best_course = course

initial_classes.remove(course)

return (max_score, best_course)

Figure 7. Pseudo-code implementation of a course candidate search algorithm based on PageRank scores.

didate course set to courses they have already researched, as opposed to performing a search over the entire course catalog.

Curricular Course Roles

Courses often play particular roles in their department, or across a university. We already mentioned service courses that provide non majors with a taste of the department’s discipline. Personnel within a department generally understand these roles. But department outsiders do not possess such knowledge. Researchers such as education or sociology scholars studying universities and colleges other than their own have even less access to role information.

We illustrate here how Via can recover course “roles.” The application of more sophisticated algorithms rooted in graph theory allows Via users to detect both roles and communities within course networks. To this end we applied a RolX [21] analysis, which requires a graph representation of the enrollment data. Intuitively, the analysis attempts to identify sets of course nodes across a graph that are similar in enrollment history, and position in students’ pathways. The algorithm then attempts to identify families of such courses, analogously to how k-means defines clusters. As with clustering, RolX does not provide semantics for discovered clusters. Those must be provided by human interpretation.

Concretely in terms of graph analysis concepts, we represent each course as a six-dimensional vector of its:

- In-degree.
- Out-degree.
- Betweenness centrality: Sum of the fraction of all course pairs whose shortest enrollment paths pass through the course node.
- Closeness centrality: closeness to all other courses enrolled in after the course.
- Closeness centrality, reversed: closeness to all other courses enrolled in before the course.
- PageRank

The RolX analysis clusters these vectors, resulting in three course roles.

Figure 8. Overview of courses that fill one of three discovered RolX roles. Red: introductory; green: intermediate; blue: enrichment. (Elision and cropping to fit publication.)

Figure 8 shows a layout of courses by role. We manually—therefore subjectively—examined catalog descriptions of courses in the three roles. The red courses are introductory into their department’s fields, such as Introductory Fluids Engineering. The green courses are intermediate offerings, which would usually be taken by majors in the field. Another example from this roles is Computer Vision: From 3D Reconstruction to Recognition. The final role, blue, comprises enrichment courses of general interest and accessibility, and seminars.

In Via the roles turn into course attributes, by which we can filter, color, and layout courses and enrollments. Figure 9 is the result of extracting the subgraph in the center of Figure 8, filtering to include only introductory-role courses, coloring course nodes by discipline, and automatically having them laid out in circles. The task of finding introductory courses in the sciences now simply involves clicking on any of the yellow nodes to see the course name, or zooming in on the Science circle to see all the course name labels.

A final task that exploits roles is creation of an overview that shows how many courses of each role the departments of the university contribute to the curriculum. Figure 10 shows the result. Each circle’s rim contains all courses that serve one role. The circles were automatically constructed by requesting a circular layout of all courses by the course role attribute. Courses of a particular department were then selected by specifying that the names of courses begin with, for example, physics.
We explained the process through which standard enrollment data can be transformed into graph structures that may be tuned to particular investigative goals. This flexibility arises both during graph construction, and during interactive manipulation of the graphs. We deploy the existing tool Cytoscape for such manipulations, but other tools may be just as appropriate, once graphs are constructed through the algorithm we have presented.

We will continue our exploration by investigating networks with multiple node types: one to model students, another to model courses. The answers to other types of questions will be found through this different family of graphs.

We further plan to extend Via to include support for simulations and what-if analyses. Further effort will also need to be invested into making the graph construction process easy to use.

Many questions may be answered by skillful SQL queries over institutional datasets, but such approaches fall short when questions are not yet clearly defined and when large data corpora beg more powerful analytic techniques. As colleges and universities seek to better understand their own internal operations and increasingly are held accountable for their performance by others, toolkits such as the one presented here will prove invaluable going forward.

REFERENCES
