Rank Aggregation for Course Sequence Discovery

Mihai Cucuringu, Charles Z. Marshak, Dillon Montag, Puck Rombach

Abstract This work extends the rank aggregation framework for the setting of discovering optimal course sequences at the university level, and contributes to the literature on educational applications of network analysis. Each student provides a partial ranking of the courses taken throughout her or his undergraduate career. We build a network of courses by computing pairwise rank comparisons between courses based on the order students typically take them, and aggregate the results over the entire student population, to obtain a proxy for the rank offset between pairs of courses. We extract a global ranking of the courses via several state-of-the art algorithms for ranking with pairwise noisy information, including SerialRank, Rank Centrality, and the recent SyncRank based on the group synchronization problem. We test this application of rank aggregation on 15 years of student data from the Department of Mathematics at the University of California, Los Angeles (UCLA). Furthermore, we experiment with the above approach on different subsets of the student population conditioned on final GPA, and highlight several differences in the obtained rankings that uncover potential hidden pre-requisites in the Mathematics curriculum.

Mihai Cucuringu
Department of Statistics and the Mathematical Institute, University of Oxford, Oxford, UK. The Alan Turing Institute, London, UK, e-mail: mihai.cucuringu@stats.ox.ac.uk

Charles Z. Marshak
Department of Mathematics, University of California at Los Angeles, Los Angeles, CA, e-mail: cmarshak@math.ucla.edu

Dillon Montag
Department of Computer Science, Westmont College, Santa Barbara, CA e-mail: dmontag23@gmail.com

Puck Rombach
Department of Mathematics, University of California at Los Angeles, Los Angeles, CA e-mail: rombach@math.ucla.edu

1 Introduction

College enrollment is at an all-time high at American universities [14], and this generation of college students is choosing to focus on science, technology, engineering, and mathematics (STEM) courses [2]. University faculty and administrators must create systems to effectively and efficiently train this burgeoning population of STEM students. The goal of this paper is to address one aspect of this broad issue through network-based techniques and statistical learning, namely the design of course sequences that can benefit the student population. We apply rank aggregation to obtain course sequences adhered to by UCLA mathematics students to infer hidden dependencies between mathematics courses, and to better understand how different types of students navigate their coursework. By comparing the course sequences of grade *A* and grade *C* students, we provide insights on the optimal course sequences for the mathematics major. More generally, the application of rank aggregation to order temporal events and discover patterns in sequences has not been investigated in the literature. As such, we also hope this work may be a proof of concept and spur such techniques for more similar data driven investigations. Our work contributes to the growing body of educational applications of network analysis, a relevant recent line of work in this direction being [12], whose authors explore the interplay between community detection and minimum spanning tress to reveal typical flows of curricular contents.

The remainder of this paper is organized as follows. In Section 2, we review related machine learning techniques designed for course sequence discovery. In Section 3, we review the technical aspects of the rank aggregation methods that will be used. In Section 4, we apply these methods to analyze sequences of mathematics courses at UCLA. We use student data from the UCLA Department of Mathematics between 2000-2015, and interpret our findings to infer course sequences from these records and latent dependencies between them. We also compare the performance of each rank aggregation method to demonstrate the robustness of this particular framework. Our final Section 5 reviews our findings and indicates future directions.

2 Related Work

Academic data mining has become a valuable tool for assisting university students select their coursework. A popular approach has been to adapt eCommerce recommendation systems to the academic space [7, 16, 15]. Here, sequences are determined incrementally by comparing student's records to others with similar coursework and grades. In [17], the authors construct an intricate system that describes how students can move through the web of course dependencies. The authors use their model to extract sequences that minimize the expected time to graduation within their system. Our goal in this work, unlike those above, is to *learn* course sequences by studying the network of courses, namely the flow of students from course to course. Our approach adapts well-known rank aggregation techniques to extract a complete sequence of all courses based on the partial sequences of courses that students have pursued. In our approach, the extracted course sequence does not provide any indication on how many courses should be taken per quarter nor what courses can be taken simultaneously. However, the extracted sequences can in turn be used as inputs for more personalized recommendation later on. Moreover, these extracted course sequences can then be used to understand how different types of students navigate through their major. By comparing high and low performing students, we infer hidden dependencies that could be missed in probabilistic models.

Rank aggregation has been a powerful tool in web search [5], sport rankings [4, 3], and more recently, grading schemes [11]. However, it has not been used to infer trends in ordering temporal data, and we believe that the data mining of such temporal sequences [19, 10, 18] may benefit from this approach. We now review the state-of-the-art rank aggregation literature, to which a subset of the authors has made a recent contribution [4], and discuss their applicability to extracting global course sequences that are most consistent with the given data.

3 Methods for Rank Aggregation

Rank aggregation is the process of obtaining a global ranking from incomplete and noisy pairwise comparisons [5]. For this academic application, we view each student as providing a set of pairwise comparisons between courses based on the order they finished their coursework. These comparisons are often incomplete (students usually do not take all available courses) and noisy (obstacles may force students to take classes out of order). We implicitly assume that there is an underlying, optimal course sequence that informs the way students navigate through their courses. We extract this course sequence using rank aggregation methods that are discussed carefully in Section 3. A review of many of the methods used here can be found in [4]. We will apply six different methods to extract course sequences: PageRank [9], Rank Centrality [8], SerialRank [6], SyncRank [4], SVD-Rank [4], and also a Least-Squares based method. We first describe the network models that will be used for all rank aggregation methods, and then summarize several state-of-the-art ranking methods as applied in our context.

3.1 Network Model and Ranking Constructions

For our academic application of rank aggregation, we wish to extract a global ranking of all courses that is most consistent with the order in which they are taken by the students. Here, our comparisons stem from the frequency with which course i was taken before course j. The first step to extract course sequences will be to translate these comparisons into a network model. We will have two network models, each quantifying the flow of students from course to course.

In the first network model, each node will represent a course, and our edges represent the flow of students between two courses. Let $k = 1, \ldots, n_s$ be our enumeration of students and similarly i, j be courses with $i, j = 1, \ldots, n_c$. We first define the variable I_{ij}^k as a binary indicator of whether student k took course i before course j. The count matrix $\mathbf{C}$ is then defined to be $C_{ij} = \sum_{k=1}^{n_s} I_{ij}^k$ when $i \neq j$, and $C_{ij} = 0$ when $i = j$. We define the $n_c \times n_c$ transition matrix $\mathbf{P}$ to have entries representing the percentage of students having taken both course i and j, but with course i before j then $P_{ij} = C_{ij}/(C_{ij} + C_{ji})$. If $C_{ij} = C_{ji} = 0$, then we define $P_{ij} = P_{ji} = 0$. By construction, $P_{ij} + P_{ji} = 1$ when course i and course j are compared at least once. The matrix $\mathbf{P}$ can be used to define a directed multigraph in which an edge weight P_{ij} approximates the flow of students moving from course i to course j. We have only considered Applied Mathematics students to construct the edge weights of the network. We note that of those students that take Real Analysis I and Linear Algebra I in different quarters, very few take Real Analysis first, as somewhat expected.

We now define a second related network model in which we ensure the net flow of students between two courses is 0, which is needed for some of the ranking methods we will consider. We define the skew-symmetric matrix $\mathbf{F}$ of size $n_c \times n_c$, with $|F_{ij}| \in [0.5, 1]$, which encodes the frequency with which course i is taken before j

$$F_{ij} = \begin{cases} P_{ij} \text{ if } P_{ij} \geq 0.5 \\ P_{ij} - 1 \text{ if } P_{ij} < 0.5. \end{cases} \tag{1}$$

In large scale applications, the measurement matrices $\mathbf{F}$ and $\mathbf{P}$ will most likely be sparse, with only a subset of the available pairwise comparisons available. We capture the existing pairwise measurements in a graph $G(V,E)$, where the node set V denotes the courses, with $|V| = n_c$. We add an edge between course i and j, that is $(i, j) \in E$ whenever $P_{ij} \neq 0$ (equivalently, $F_{ij} \neq 0$), and let $m = |E|$.

3.2 Rank Centrality

Rank Centrality was conceived as a way to discover rankings generated by the Bradley-Terry model [8]. This model assumes players i and j have latent real-valued weights w_i, w_j assigned to them so that $\mathbb{P}(i \text{ beats } j) = w_i/(w_i + w_j)$. The authors used this method to in turn rank NASCAR drivers and Indian cricket teams with great success [8]. Again, for our application, we compare courses i and j, in which "beat" means course i came before course j in sequence of courses. Let $\mathbf{P}$ be the matrix defined in the previous section whose entries P_{ij} denote the number of students that took course i before j, amongst all those students that took i and j in different quarters. Rank Centrality defines a Markov chain on the n_c courses with the following stochastic matrix $\mathbf{S}^{\text{rc}} = \mathbf{P}/d_{\max} + (\mathbf{I} - \mathbf{D}/d_{\max})$, where $d_{\max}$ is the maximum out degree in the network, $\mathbf{I}$ is the $n_c \times n_c$ identity matrix, and $\mathbf{D}$ is the diagonal matrix of out degrees with $d_{ii} = \sum_j P_{ij}$. This differs in two important ways from the popular PageRank algorithm [9]. First, there is no teleportation term for Rank Centrality. Second, from the latter term in the sum, the random walker can remain at course i with probablity $1 - 1/d_{\max} \sum_j P_{ij}$. It also means that courses with smaller total out-degree will have an added self-loop of nontrivial weight. This means less-popular courses tend to higher marginal values in the stationary distribution.

3.3 SerialRank

SerialRank [6] adapts the seriation problem proposed in [1] to determine a global ranking of players. The authors define a similarity function determined by the outcomes two players have with common opponents, and they study the similarity graph rather than a Markov chain. For our academic application, we interpret links as the likelihood two courses are taken at a similar time.

To construct the similarity matrix, we recall the matrix $\mathbf{P}$ from the previous sections whose entries P_{ij} that counts the percentage of students taking course i and then course j. As in [6], we construct the comparison matrix $\mathbf{A}_k$ for course k using

$$(A_k)_{ij} = 1 - \frac{|P_{ik} - P_{jk}|}{2}$$

whenever both course i and course j have been taken in a sequence with course k. If either course i or course j has not been taken in sequence k, we define $(A_k)_{ij} = \frac{1}{2}$. In other words, if course i and course j have a similar percentage of students that are taking course k afterwards, then i and j must be more similar themselves. The similarity matrix $\mathbf{S}^{\mathrm{sr}}$ is then determined by summing over the comparison matrices $\mathbf{S}^{\mathrm{sr}} = \sum_{k=1}^{n_c} \mathbf{A}_k$. We can then rescale $\mathbf{S}^{\mathrm{sr}}$ by subtracting the minimum value of $\mathbf{S}^{\mathrm{sr}}$ so that course i and j with the smallest similarity now have similarity of 0. To determine a ranking from $\mathbf{S}^{\mathrm{sr}}$, we form the combinatorial laplacian $\mathbf{L}$ and rank the courses using the components of the Fiedler vector [6], where $\mathbf{L}$ is given by $\mathbf{L} = \mathbf{D} - \mathbf{S}^{\mathrm{sr}}$. The justification for inspecting the Fiedler vector is proposed in [1] as a relaxation of an NP-hard problem. Specifically, let us assume that $\mathbf{q}$ is a vector with the ith component representing the ranking of the ith course. We can see that the following energy will be minimized for an optimal ranking $\mathbf{q} = \arg\min_{\mathbf{q}'} \sum_{i,j} s_{ij}^{\mathrm{sr}} (q'_i - q'_j)^2$. Observering that $\sum_{i,j} s_{ij}^{\mathrm{sr}} (q'_i - q'_j)^2 = (\mathbf{q}')^T \mathbf{L} \mathbf{q}'$, the problem can be relaxed into the well-studied eigenvector problem $\mathbf{q} = \arg\min_{\mathbf{q}'} (\mathbf{q}')^T \mathbf{L} \mathbf{q}'$ such that $||\mathbf{q}'||_2 = 1$ and $\mathbf{1}_{n_c}^T \mathbf{q}' = 0$, where $\mathbf{1}_{n_c}$ is the vector of length n_c comprised entirely of 1's. The minimum of the above optimization problem is attained by the eigenvector corresponding to the smallest non-trivial eigenvalue of $\mathbf{L}$. After ordering the n_c courses using the components of $\mathbf{q}$, we obtain a global ranking.

3.4 SyncRank: Synchronization based ranking

In very recent work by one of the authors [4], the problem of ranking with incomplete noisy information was formulated as an instance of the group synchronization problem over the special orthogonal group SO(2), which we briefly detail. Determining individual group elements from the measurement of their pairwise ratios is known as the *group synchronization* problem [13]. The seminal paper of Singer [13] considered the angular synchronization problem over SO(2), where the goal is to recover n unknown ground truth angles $\theta_1, \ldots, \theta_n \in [0, 2\pi)$, given m noisy pairwise angle offsets captured in a matrix Θ of size $n \times n$ $\Theta_{ij} = (\theta_i - \theta_j + \text{Noise}) \bmod 2\pi$. Spectral and semidefinite programming (SDP) relaxations for this problem (followed by a rounding procedure) were originally introduced and analyzed by Singer [13]. The difficulty of the problem is amplified on one hand by the amount of noise in the offset measurements, and on the other hand by the fact that $m \ll \binom{n}{2}$, i.e., only a very small subset of all possible pairwise offsets are measured.

The connection between ranking and angular synchronization can be summarized as follows. Denote the true ranking of the n_c courses by r_i, assuming without loss of generality that $r_i = i$, i.e., the rank of the i^{th} course is i. Imagine the ranks to lie on a one-dimensional line, with the pairwise rank comparisons given, in the noiseless case, by $R_{ij} = r_i - r_j$. Note that this matrix has a similar interpretation to the matrix F defined in (1), in the sense that both are skew-symmetric matrices that capture the outcome of the pairwise comparison between i and j.

Next, we consider the angular embedding given by mapping the ranks of the courses to the unit circle, say fixing r_1 to have a zero angle with the x-axis, and the last player r_{n_c} corresponding to an angle equal to π. We interpret the given pairwise measurements F_{ij} as a proxy for the rank offsets, and map them to $\Theta_{ij} \in [0,\pi)$ via the transformation $R_{ij} \mapsto \Theta_{ij} := \pi F_{ij}$. In other words, we imagine the n_c courses wrapped around a fraction of the circle, interpret the available rank-offset measurements as angle-offsets in the angular space, and thus arrive at the setup of the angular synchronization problem. We then build the $n_c \times n_c$ Hermitian matrix $\mathbf{H}$ with $H_{ij} = e^{\imath \theta_{ij}}$, if $(i,j) \in E$, and $H_{ij} = 0$ otherwise, with $\imath = \sqrt{-1}$. We solve the angular synchronization problem via its spectral relaxation, which considers the top eigenvector v_1 of $\mathbf{H}$, and denote the recovered solution by $\widehat{r}_i = e^{\imath \widehat{\theta}_i} = \frac{v_1(i)}{|v_1(i)|}, \quad i = 1,2,\dots,n_c$. We extract the corresponding set of angles $\widehat{\theta}_1,\dots,\widehat{\theta}_{n_c}$ from $\widehat{r}_1,\dots,\widehat{r}_{n_c}$, which induces the final ranking solution after modding out the best circular permutation. This last step is due to the fact that the estimation of the rotation angles is up to an additive phase, since $e^{\imath\alpha} v_1$ is also an eigenvector of $\mathbf{H}$ for any $\alpha \in \mathbb{R}$.

4 Course Sequence Discovery

Next, we apply various ranking techniques to data collected by the UCLA Department of Mathematics[1] between Fall 2000 and Spring 2015.

4.1 Cleaning the Data

The data is comprised of individual student records indicating each student's quarter by quarter math coursework, the student's grade in the course, and the major declared each quarter. To apply the rank aggregation methods above, we must construct the matrix $\mathbf{P}$ or $\mathbf{F}$ defined in Section 3.1. We restrict our attention to a particular major, each major having their very own set of requirements and objectives. At UCLA, the Department of Mathematics has 7 unique majors ranging from Pure Mathematics to Mathematics & Economics to Mathematics of Computation. Since a student can change their major from quarter to quarter, we group students based on the major they declare in their last quarter. We also exclude all community college transfer students as they often exhibit very different trends compared to students that were admitted as Freshman. Moreover, if a student retook a class, we look only at the grade and the quarter the class was *last* taken. Lastly, once a suitable population of students is selected, we only consider those math classes that at least 10 percent of this population takes. In particular, if we consider only a particular major, we exclude classes that less than 10% of this major has taken.

[1] The research was reviewed and approved by the Chair of the UCLA Department of Mathematics, on fully anonymized data, allowing for the dissemination of our results in a scholarly publication.

4.2 Assessing Rank Aggregation

We analyze the sequences obtained by rank aggregation, and compare in Table 3 the output sequences for all six methods on Applied Mathematics majors who have an A-range GPA. All methods were able to correctly order the calculus sequences (31AB, 32AB, 33AB) and are thus removed from the table. Moreover, all methods place Linear Algebra I and Real Analysis I fairly early in the sequence as these are courses required for all majors. All the methods consistently placed Real Analysis I before Real Analysis II and III and similarly for other courses taught in sequences. These basic dependencies were captured by rank aggregation across the board. We did notice that Rank Centrality had the greatest disagreement when compared to the rest of the methods. Rank Centrality placed Applied Algebra and Partial Differential Equations much earlier than the other methods. We suspect this particular disagreement with the other methods is due to the relatively low enrollment in these courses. About 10% of students in the Applied Mathematics major took these classes, which was the minimum for a class to be considered in our course sequence. We suspect Rank Centrality to be affected in certain cases by the popularity of classes.

Figure 1 is the heat-map corresponding to matrix **P**, and shows a more granulated view of the flow of students. The names of the course numbers that label the axes are shown in Table 4. We only consider Pure Math students with A-range GPAs, and observe in the same Figure 1 that the Calculus sequence (31A, 31B, 32A, 32B) is taken earliest in the expected sequence. We observe $P_{ij} + P_{ji} = 1$ when both i and j have been taken by at least one student and are taken in different quarts. There are some pairs of classes for which this is not the case. For instance, we can see from Figure 1 that no single student took Math 61 (Discrete Structures) and Math 133 (Fourier Series) in different quarters. Our assessments above indicate that indeed this approach to course sequence discovery produces meaningful results. Finally, we remark that our approach is making the implicit assumption that the coursework flow can be represented as a linear sequential ordering, which may not be appropriate in certain cases, especially when partial orderings may better fit the data (e.g., there could be three courses that clearly occupy positions 7-9, but in no particular order).

4.3 Inferring Hidden Prerequisites

One of the original motivations for this work was to learn optimal course sequences through the math major. Even though our network model is not dependent on performance, we can still condition the students we consider in the network's construction by their overall GPA. Comparing the extracted sequences of high and low performing students, we can infer hidden prerequisites and optimal course sequences. In Table 1, we investigate this for three different majors using SyncRank [4]. We compare the first 11 non-Calculus courses for A-range students and C-range students. For pure mathematics students, we see that A students tend to take Discrete Structures before Linear Algebra I, the latter course being the first upper division proof-based course a math student takes at UCLA. As such, we might infer that Discrete Structures, which covers the basics of functions, sets, and combinatorics to be helpful to Pure Mathematics students entering a proof-based curriculum. Another

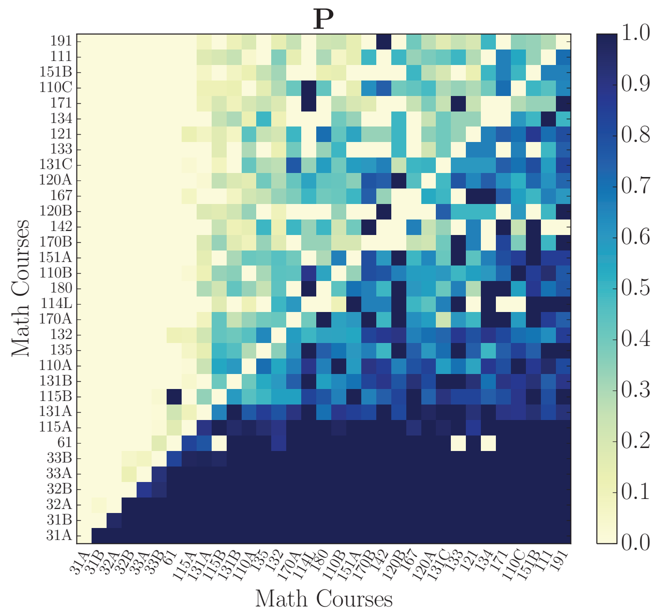

Fig. 1 The **P** matrix for A students with a Pure Mathematics focus (courses ordered by PageRank).

Table 1 Comparing the A and C students in 3 majors using SyncRank.

Applied Mathematics		**Applied Sciecnces**		**Pure Mathematics**	
A ($n_s = 140$)	**C** ($n_s = 198$)	**A** ($n_s = 75$)	**C** ($n_s = 162$)	**A** ($n_s = 86$)	**C** ($n_s = 95$)
Lin. Algebra I	Lin. Algebra I	Lin. Algebra I	Lin. Algebra I	Discr. Struct.	Lin. Algebra I
Discr. Struct.	Discr. Struct.	Probability I	Discr. Struct.	Lin. Algebra I	Hist. of Math
Real Analysis I	Probability I	Discr. Struct.	Probability I	Real Analysis I	Real Analysis I
Probability I	Real Analysis I	Real Analysis I	Real Analysis I	Lin. Algebra II	Discr. Struct.
Complex Analysis	Algebra I	Act. Math	Nonlin. Syst.	Algebra I	Algebra I
Nonlin. Syst.	Num. Analysis I	Num. Analysis I	Math Modeling	Real Analysis II	Ord. Diff. Eqn.'s
Num. Analysis I	Graph Theory	Probability II	Graph Theory	Ord. Diff. Eqn.'s	Complex Analysis
Math Modeling	Real Analysis II	Graph Theory	Game Theory	Complex Analysis	Game Theory
Real Analysis II	Act. Math	Act. Models II	Num. Analysis I	Probability I	Probability I
Algebra I	Nonlin. Syst.	Act. Models II	Optimization	Algebra II	Graph Theory
Graph Theory	Math Modeling	Ord. Diff. Eqn.'s	Ord. Diff. Eqn.'s	Graph Theory	Num. Analysis I
Ord. Diff. Eqn.'s	Hist. of Math	Num. Analysis II	Act. Math	Real Analysis III	Optimization
Game Theory	Complex Analysis	Optimization	Probability II	Num. Analysis I	Number Theory
Research Seminar	Probability II	Math Econ.	Act. Models II	Logic	Algebra II

interesting feature for Pure Mathematics students is that A students tend to take Real Analysis I and Real Analysis II fairly close to each other while C students do not. The Real Analysis coursework is known to be a conceptually challenging university course, both at UCLA and elsewhere. More statistical analysis is needed to ascertain whether strong performance is correlated to taking these course sequences in close succession, but rank aggregation allows us to quickly identify such trends.

For Applied Mathematics students, we see that Probability I is frequently taken after Real Analysis I for A students, but not for C students. As such, we infer that Real Analysis I is a prerequisite for Probability I when considering such students. Contrary to this ordering, Applied Science students take Probability before Real Analysis I whether they are A or C students. One possible explanation for the differences between these applied majors is that many Applied Science students are pre-actuarial and are very familiar with the material in Probability I. As such, the ordering of Probability I and Real Analysis I is not as pertinent for Applied Science students. Assuming A students navigate through the major best, we can also use rank aggregations method to study differences in major course sequences as in Table 2. Here, we apply SerialRank to 6 different math majors, again only considering A-range students. We see that the Applied math disciplines (Applied Science, Mathematics & Economics, Mathematics for Computation, and Applied Mathematics) all take Probability I early, while Pure Mathematics students generally do not. We also noticed that Applied Science and Mathematics for Computation students take at least one pre-actuarial course, while other majors do not. Of course, these findings do not indicate the correlation of these course sequences with performance, but certainly this methodology quickly illuminates possible trends, and represents an interesting possible future research direction to explore.

5 Conclusions and Future Work

We demonstrated that with an appropriate network model, rank aggregation techniques can extract course sequences and infer latent course dependencies. Our findings captured easily verifiable dependencies such as the completion of lower division calculus courses, and then proceeding to Linear Algebra I and Real Analysis I, two gateway classes central to the math curriculum across all majors. We were able to inspect the differences between the various math majors at UCLA and the differences between grade A, B, and C students within these majors. Lastly, we were able to infer that there are some hidden prerequisites of courses that emerge based on the trends observed in high and low performing students. Rank aggregation has typically been used to rank sports teams, athletes, or other competitive endeavors. The application in our present work adapts this methodology for temporal orderings of university course data. The crucial ingredient was an appropriate network model that captured how students navigate through courses, which rendered the problem suitable for state-of-the-art algorithms for ranking with noisy pairwise measurements.

6 Acknowledgements

This work was supported by NSF grant DMS-1045536, UC Lab Fees Research Grant 12-LR-236660, ARO MURI grant W911NF-11-1-0332, AFOSR MURI grant

Table 2 Course Sequences for Applied Mathematics Majors obtained by the six methods.

PageRank	**Rank Centrality**	**SerialRank**
Linear Algebra I	Linear Algebra I	Linear Algebra I
Discr. Struct.	Discr. Struct.	Discr. Struct.
Real Analysis I	Applied Algebra	Real Analysis I
Probability I	Partial Diff. Eqn.'s	Probability I
Num. Analysis I	Math Modeling	Num. Analysis I
Nonlin. Syst.	Real Analysis I	Nonlin. Syst.
Complex Analysis	Abstract Algebra I	Abstract Algebra I
Abstract Algebra I	Game Theory	Complex Analysis
Real Analysis II	Complex Analysis	Real Analysis II
Graph Theory	Num. Analysis II	Graph Theory
Math Modeling	Num. Analysis I	Math Modeling
Act. Math	Graph Theory	Act. Math
Ord. Diff. Eqn.'s	Probability I	Ord. Diff. Eqn.'s
Applied Algebra	Nonlin. Syst.	Applied Algebra
Optimization	Hist. of Math	Hist. of Math
Hist. of Math	Probability II	Probability II
Probability II	Act. Math	Optimization
Num. Analysis II	Real Analysis II	Game Theory
Game Theory	Ord. Diff. Eqn.'s	Num. Analysis II
Partial Diff. Eqn.'s	Optimization	Partial Diff. Eqn.'s

SVD	**Least Squares**	**SyncRank**
Linear Algebra I	Linear Algebra I	Linear Algebra I
Discr. Struct.	Discr. Struct.	Discr. Struct.
Real Analysis I	Real Analysis I	Real Analysis I
Probability I	Probability I	Probability I
Nonlin. Syst.	Num. Analysis I	Num. Analysis I
Num. Analysis I	Nonlin. Syst.	Nonlin. Syst.
Complex Analysis	Complex Analysis	Complex Analysis
Abstract Algebra I	Abstract Algebra I	Abstract Algebra I
Real Analysis II	Real Analysis II	Act. Math
Graph Theory	Graph Theory	Graph Theory
Act. Math	Math Modeling	Real Analysis II
Math Modeling	Act. Math	Applied Algebra
Applied Algebra	Applied Algebra	Math Modeling
Ord. Diff. Eqn.'s	Ord. Diff. Eqn.'s	Ord. Diff. Eqn.'s
Hist. of Math	Hist. of Math	Hist. of Math
Probability II	Optimization	Probability II
Optimization	Probability II	Optimization
Num. Analysis II	Num. Analysis II	Num. Analysis II
Game Theory	Game Theory	Game Theory
Partial Diff. Eqn.'s	Partial Diff. Eqn.'s	Partial Diff. Eqn.'s

Table 3 First 11 courses for A students in 5 different majors using SyncRank.

Applied Mathematics $n_s = 140$	**Pure Mathematics** $n_s = 86$	**Applied Sciences** $n_s = 75$	**Mathematics & Economics** $n_s = 101$	**Mathematics for Computation** $n_s = 20$
Lin. Algebra I	Discr. Struct.	Lin. Algebra I	Discr. Struct.	Lin. Algebra I
Discr. Struct.	Lin. Algebra I	Probability I	Lin. Algebra I	Probability I
Real Analysis I	Real Analysis I	Discr. Struct.	Real Analysis I	Real Analysis I
Probability I	Lin. Algebra II	Real Analysis I	Probability I	Discr. Struct.
Complex Analysis	Algebra I	Act. Math	Applied Algebra	Probability II
Nonlinear Systems	Real Analysis II	Num.Analysis I	Optimization	Act. Math
Num.Analysis I	Ord. Diff. Eqn.'s	Probability II	Real Analysis II	Math Modeling
Math Modeling	Complex Analysis	Graph Theory	Probability II	Math Econ.
Real Analysis II	Probability I	Act. Models II	Num.Analysis I	Act. Models II
Algebra I	Algebra II	Act. Models II	Game Theory	Num.Analysis I
Graph Theory	Graph Theory	Ord. Diff. Eqn.'s	Math Econ.	Loss Models I

Table 4 Course names and numbers.

#	Course Name	#	Course Name	#	Course Name
31A	Calculus I	120A	Differential Geometry I	164	Optimization
31B	Calculus II	120B	Differential Geometry II	167	Game Theory
32A	Multi. Calculus I	121	Topology	170A	Probability I
32B	Multi. Calculus II	123	Axiomatic Geometry	170B	Probability II
33A	Applied Linear Algebra	131A	Real Analysis I	172A	Actuarial Mathematics
33B	Applied ODE's	131B	Real Analysis II	172B	Actuarial Models II
61	Discrete Structures	131C	Real Analysis III	172C	Actuarial Models II
106	History of Mathematics	132	Complex Analysis	173A	Casualty Loss Models I
110A	Abstract Algebra I	133	Fourier Analysis	173B	Casualty Loss Models II
110B	Abstract Algebra II	134	Nonlinear Systems	174	Math. Economics
110C	Abstract Algebra III	135	ODE's	180	Graph Theory
111	Number Theory	136	PDE's	182	Algorithms
115A	Linear Algebra I	142	Mathematical Modeling	184	Combinatorics
115B	Linear Algebra II	151A	Numerical Analysis I	191	Research Seminar
117	Applied Algebra	151B	Numerical Analysis II	199	Individual Research

FA9550-10-1-0569, NSF grant DMS-1417674, and ONR grant N-0001-4121-0838, and EPSRC grant EP/N510129/1.

References

[1] Atkins, J.E., Boman, E.G., Hendrickson, B.: A spectral algorithm for seriation and the consecutive ones problem. SIAM Journal on Computing **28**(1), 297–310 (1998)

[2] Bowen, R.M.: Science and engineering indicators 2012. Online, Arlington VA: National Science Foundation (2012)

[3] Callaghan, T., Mucha, P.J., Porter, M.A.: Random walker ranking for NCAA division IA football. American Math. Monthly **114**(9), 761–777 (2007)
[4] Cucuringu, M.: Sync-Rank: Robust Ranking, Constrained Ranking and Rank Aggregation via Eigenvector and SDP Synchronization. IEEE Transactions on Network Science and Engineering **3**(1), 58–79 (2016)
[5] Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proceedings of the 10th international conference on World Wide Web, pp. 613–622. ACM (2001)
[6] Fogel, F., d'Aspremont, A., Vojnovic, M.: Serialrank: Spectral ranking using seriation. In: Advances in Neural Information Processing Systems, pp. 900–908 (2014)
[7] Ghauth, K.I., Abdullah, N.A.: Learning materials recommendation using good learners' ratings and content-based filtering. Educational Technology Research and Development **58**(6), 711–727 (2010)
[8] Negahban, S., Oh, S., Shah, D.: Iterative ranking from pair-wise comparisons. In: NIPS, pp. 2474–2482 (2012)
[9] Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. (1999)
[10] Pei, J., Han, J., Wang, W.: Mining sequential patterns with constraints in large databases. In: Proceedings of the 11th International Conference on Information and Knowledge Management, pp. 18–25. ACM (2002)
[11] Raman, K., Joachims, T.: Methods for ordinal peer grading. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1037–1046. ACM (2014)
[12] Sayama, H.: Mapping the curricular structure and contents of network science courses. CoRR **abs/1707.09570** (2017). URL http://arxiv.org/abs/1707.09570
[13] Singer, A.: Angular synchronization by eigenvectors and semidefinite programming. Appl. Comput. Harmon. Anal. **30**(1), 20–36 (2011)
[14] Snyder, T.D., Dillow, S.A.: Digest of education statistics, 2012. National Center for Education Statistics (2013)
[15] Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., Schmidt-Thieme, L.: Recommender system for predicting student performance. Procedia Computer Science **1**(2), 2811–2819 (2010)
[16] Wen-Shung Tai, D., Wu, H.J., Li, P.H.: Effective e-learning recommendation system based on self-organizing maps and association mining. The Electronic Library **26**(3), 329–344 (2008)
[17] Xu, J., Xing, T., Van Der Schaar, M.: Personalized course sequence recommendations. IEEE Transactions on Signal Processing **64**(20), 5340–5352
[18] Yang, J., Wang, W., Yu, P.S.: Mining asynchronous periodic patterns in time series data. IEEE Transactions on Knowledge and Data Engineering **15**(3), 613–628 (2003)
[19] Yang, J., Wang, W., Yu, P.S., Han, J.: Mining long sequential patterns in a noisy environment. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, pp. 406–417. ACM (2002)