

Via: A Prerequisite Projection Graph for Course Sequence Discovery

Richard Diehl Martinez
Computer Science
Stanford University
Email: rdm@stanford.edu

Geoffrey Angus
Computer Science
Stanford University
Email: gangus@stanford.edu

***Index Terms*—Causal Inference Sequence, Temporal Networks, Bipartite Graphs, Network Analysis.**

I. INTRODUCTION

Every year, more than 1400 students enter Stanford University as first-year undergraduates. To help these students, Stanford provides an array of tools to advise students on how to effectively structure their course progression. Since its inception in 2014, Carta¹ has emerged as one of the primary tools at students' disposals to structure their academic pathway through Stanford. The goal of Carta, as expressed in their mission statement, is to help students make informed decisions about their classes by evaluating how students consider, choose, and sequence courses. However, few of Stanford's academic resources give students access the academic pathway information necessary to contextualize local decisions at the course enrollment level. By analyzing recurrent patterns in the historical enrollment data of students throughout their academic careers, we propose a graphical representation of course sequences that enable us to extract patterns in student behavior regarding course enrollment decisions.

II. PROBLEM FORMULATION

The goal of our project is to analyze historical student behavior in course enrollment in order to generate a directed graph that accurately models how courses are typically sequenced. To do this, we must demonstrate that our graphical representation of the sequential relationship between classes is representative of actual observable dynamics at Stanford. This assumption will enable us to further explore patterns in the course data uncover latent relationships between courses in our so-called course sequence graph. Moreover, we will demonstrate that the course sequence graph enables us to recover and study global properties of course sequences such as academic majors, and knowledge specializations.

III. RELATED WORK

Our literature review centered primarily on unsupervised learning algorithms to detect latent structures in networks. Part of our goal in creating a course sequence graph is the ability to detect the presence of community structures between

courses. Karrer et. al's *Stochastic blockmodels and community structure in networks* propose a degree-corrected stochastic block model that outperformed those block models that did not account for degree [1]. They propose this algorithm in order to correct for the inaccuracies that, at time of writing, plagued the performance of stochastic block models on real world networks. We posit that mandatory courses taken in the first two years at Stanford will present problems similar to those found in the the real world networks described in the paper.

Another look into the augmentation of traditional stochastic block models can be found in Aicher et. al's paper *Learning latent block structure in weighted networks*. This paper proposed a Weighted Stochastic Block Model, for the task of discovering structure in information that would otherwise be lost when weights in a real world network are discarded or thresholded. The projection task will rely on co-enrollment frequency in order to classify prerequisite relationships, therefore there will likely be a weighted signal produced for each class relationship that signifies a confidence in prerequisite relationship.

In addition to finding community blocks within course sequences, we are interested in leveraging the temporal nature of our dataset. That is, given that a set of students that have taken a set of class a certain number of quarters apart, what is the probability that there exists some prerequisite relationship between those classes. Chang et al. have explored a similar problem in predicting temporal bipartite graph projections for online social networks [2]. The authors use a method known as relational topic modeling to draw edges between groups of nodes that share similar local structures. Chaturvedi et al. frame the problem of link prediction in a similar manner, but employ a kernalized SVM through which they pipe information about a node's local structure [4].

In addition, we reviewed literature regarding the topic of algorithmic course sequencing in higher education. Doing so enabled us to cross examine our methodology and gain a more holistic insight into the field. One of these papers was Xu et. al's Personalized Course Sequence Recommendations [5]. The authors of this paper frame the task of developing an optimal sequence of courses as a task of constructing an optimal policy such that a students time until graduation is minimized and their GPA t is maximized. The optimization algorithm used in order to do this is a variation of a Forward-Search Backward-

¹<https://www.cartastanford.edu>

Induction algorithm.

A. Enrollment Data

1) *Data Description*: The dataset we use to build our graph for mapping sequential relationships between classes was obtained directly from the Stanford Carta Lab. This dataset contains anonymized class enrollment data for over 52,000 students (both graduates and undergraduates) who were enrolled at Stanford during any time between Fall 2000 and Fall 2018. Most importantly for our purposes, the enrollment data gives us individualized student data, allowing us to aggregate the course sequence history of each student in the dataset.

2) *Data Preprocessing*: Data preprocessing primarily involved the separation and extraction of the information required from the Carta dataset—that is to say, for each student, all classes paired with their quarters of enrollment. Student metadata was omitted and only classes in our dataset marked completed (classes not dropped by the student during the quarter) were included. We then built a sequence matrix representation of what we will call the Carta Network. This can be interpreted as a bipartite, multi-graph that expresses student enrollment in courses over time. This sequence matrix, S , has the dimensionality $|M| \times |C|$, where M is the set of all students and C is the set of all classes. An entry into the matrix indicates which student took which class at which quarter in their Stanford career. For instance if $S_{i,j} = 5$ this means that student i took class j during their 5th quarter at Stanford. This sequence matrix enables us to discern the classes frequently taken sequentially by students.

Abstraction and its relation to programming. Software engineering principles of data abstraction and modularity. Object-oriented programming, fundamental data structures (such as stacks, queues, sets) and data-directed design. Recursion and recursive data structures (linked lists, trees, graphs). Introduction to time and space complexity analysis. Uses the programming language C++ covering its basic facilities. Prerequisite: 106A or equivalent. Summer quarter enrollment is limited.

Fig. 1. Course Description for CS106B: Programming Abstraction.

B. Course Description Data

1) *Data Description*: In addition to course enrollment information, the Carta Team provided us access to a scraped dataset of course descriptions from explore courses. In total, this dataset contained roughly 29,000 course descriptions across all majors. An example of an entry from the dataset is provided in fig. 1, for the course CS106B: Programming Abstractions.

2) *Data Preprocessing*: In our later analysis we will leverage the course descriptions in order to generate a 'ground-truth' for course sequences. Our insight for doing so was that several course descriptions explicitly list prerequisites for taking a particular class. Since these requirements are explicitly enumerated by a teaching team, we can assume that any courses listed in the prerequisite section of a course description are accurate. In order to extract the prerequisite section and the courses listed under this header we make use of regular expressions. The regular expression extraction was made difficult by the varying ways in which courses can be referred to (i.e. CS106A, 106A...). As a result, we note that even though this data provides us a ground-truth these values may be noisy.

IV. GRAPH CONSTRUCTION

In the following subsections we introduce several graphical models that illustrate common course sequence structures. Each of the subsections below are ordered by increasing complexity in their representation of course relationships. We constructed these baseline models in order to gain insight and confirm some intuitions about the nature of the Carta network.

For the construction of these graphs we are guided by the following first-principle assumptions regarding the nature of classes and their content:

- 1) Knowledge can both be learned and lost.
- 2) The probability of losing knowledge increases the more time passes since that knowledge was gained.
- 3) A class A teaches a finite set of concepts to students. This set of concepts collate into what we know to be knowledge.
- 4) A class A may require students to have a prior understanding of another finite set of concepts.
- 5) For many pairs of classes A and B , there exist overlapping concepts that are taught or required in both of these classes.
- 6) For many pairs of classes A and B , the concepts covered in class A are required to complete class B .
- 7) If assumption 2 holds for a pair of classes A and B , it is very unlikely that the inverse relationship holds as well: that is to say that the concepts covered in class B is not required to complete class A .
- 8) Given assumption 6, we can reasonably assume that a class B will be taken after a class A for which the relationship in assumption 2 holds.

From these first principles we can reasonably assume that all classes either have or do not have some prerequisite relationship. We model this relationship between all classes offered at Stanford, by denoting each class as a node in a graph and each edge as a directed prerequisite relationship between two classes.

A. Carta Network

The Carta Network can be defined as a bipartite multi-graph that models each student as a node and each class as a node. A student node, s , and a class node, c , are then linked via an edge

$e^{(t)}$ if student s took a class c at timestep t of their Stanford career. This bipartite graph is a graphical representation of the sequence matrix which will be used to generate basic visualizations of the data in our final report.

B. Prerequisite Graph (Baseline)

After building a graph that maps the relationship between student and classes directly, we explore how to model a projection of this graph to establish relationships between classes. For this task, we are guided by the aforementioned first-principles. In particular, we make use of the sequence matrix to establish a new, prerequisite score graph. We define G to be a symmetric graph with dimensions $|C| \times |C|$, where C is the set of classes present in the dataset. Each entry in the graph then establishes how often a class j was taken after a class i in a directly preceding quarter. That is to say $G_{i,j} = 2000$ means that 2000 students took class j exactly one quarter after taking class i . We can derive this information quickly from the sequence matrix. In order to then establish a graph from this information, we can extract the top n entries from the matrix G . This will return a list of n tuples that represent the top n pairs of classes taken exactly one quarter after the other.

C. Prerequisite Graph (Discounting)

One of the problems we encounter in our previous construction of a prerequisite graph is that this model does not account for any relationship between courses that have been taken more than one quarter apart. That is to say, we would like to account for a sequential relationship between classes that have been taken more than one quarter apart. However, we posit that the more time has elapsed between a student taking two classes, the less likely it is that the two have a prerequisite relationship. This insight follows fairly simply from our assumptions 2 and 8 gathered from first-principles. That is to say, if a class B directly requires information from a class A then class A and B should be taken within a close time interval of each other. Otherwise, the information from class A may be lost over time and be of less use to class B . These guiding principles thus lead us to conclude that we would like each entry in our graph G to also include a discounted factor for classes i and j that have been taken sequentially, but more than a quarter apart². In our implementation, we use a discounting factor of 0.9 (found to work best empirically) that is exponentially discounted as the time between taking class j after i increases. Thus each entry $G_{i,j}$ is computed as

$$G_{i,j} = \sum_{s \in S} (0.9)^{t_s(i,j)} I_{s:i \rightarrow j}.$$

Where we let S be all sets of students, t_s be a function that computes the numbers of quarters between student s 's enrollment in class j after i , and $I_{s:i \rightarrow j}$ be an indicator variable for whether a student took class j after class i . Having made these changes, however, we note that we have yet to account

for an additional factor. Given our assumption 7, we should not expect to see any bi-directional relationship between two classes. That is to say, a class i and j should not list each other as prerequisites. We would thus ideally like to penalize the sequential relationship between a class i and j whenever we observe a large number of relationships between j and i as well. We can also make the inverse argument as we did before, that the further apart a sequential relationship is observed between a class j to i the less it should penalize the sequential relationship between i and j . We can now make this simple change, to obtain our final graph G , where each entry in G is now given by:

$$G_{i,j} = \sum_{s \in S} (0.9)^{t_s(i,j)} I_{s:i \rightarrow j} - \sum_{s \in S} (0.9)^{t_s(j,i)} I_{s:j \rightarrow i}.$$

As before, in order to now establish a graph from this information, we can extract the top n entries from the matrix G . This model would seem to capture the semantics of prerequisite relationships more soundly than our baseline.

D. Prerequisite Graph (Discounting, Normalized)

Although, the prerequisite graph established in the previous section successfully accounts for dynamic sequential class relationships, it fails to normalize for the number of students that have taken a particular class. Intuitively, our current model fails to accurately balance out the sequential influences between classes, since classes with large enrollments, like CS 106A, will naturally always yield the most sequential relationships with any number of different classes, no matter if they should be related sequentially or not. To account for this normalization, we divide each entry in a certain row i of the 'prerequisite scores matrix' G by the square root of the enrollment of course i . That is what $G_{\text{normalized}}[i, j] = G[i, j] / \sqrt{\text{enrollment}(i)}$. We find empirically that dividing by the absolute value of the enrollment of class i penalized large classes too much. Mathematically, we can now express our normalized matrix G , $G_{\text{normalized}}$, as simply $G_{\text{normalized}} = D^{-1/2} G I_{\text{diag}}$. Where $D^{-1/2}$ is a diagonal matrix with the enrollment data of each class i on the i^{th} entry along the diagonal, G is the previous discounted scores matrix and I_{diag} is an identity matrix.

E. Discounting, Learned

After running experiments on the model described above, we determined that a learned decay could lead to even more promising results. We propose an algorithm that uses automatic differentiation via backpropagation to learn some scaling coefficient for the sum of enrollments for each timestep. In our proposed Learned Discount model, we calculate scores for each of the class pairings i, j in prerequisite adjacency matrix G with the following formula:

$$G_{ij} = \sum_{i=-T}^T \theta_i (|\text{enrolled after } i \text{ timesteps}|)$$

²<https://link.springer.com/article/10.1007/s00712-013-0363-3>

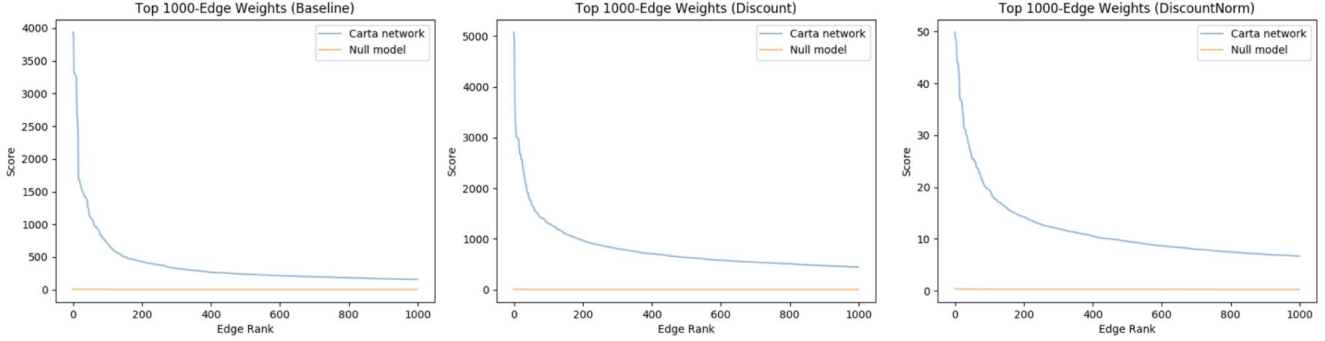


Fig. 2. A comparison of the weights of the top one thousand of one hundred million edges generated by each of the scoring mechanisms. When the scoring mechanisms are applied to both the Carta network and the null model, it is evident that there are strong behavioral patterns among students that imply prerequisite relationships between courses. The exponential relationship evident across all 3 models tells us that prerequisite relationships only exist between a small subset of class pairs, confirming our intuition.

We derive this formula from the notion that we can re-express the score calculation presented for Normalized Discount as the following:

$$G'_{ij} = \sum_{k=1}^T (0.9)^k (|\text{enrolled after } k \text{ timesteps}|) - \sum_{k=-T}^{-1} (0.9)^k (|\text{enrolled after } k \text{ timesteps}|)$$

By learning parameter vector θ , we replace the decay factor used above with a set of coefficients learned from features of the input tensor. In order to do this, several modifications had to be made to the process of graph generation. Specifically, we reformulated the problem as a regression task that trained these weights to find a best fit between our predicted prerequisite adjacency matrix and a ground truth adjacency matrix.

First, we created a non-discounting 3-D tensor M of the shape $|C| \times |C| \times T$, where C is the set of classes and T is the maximum delta in time between the enrollment in two classes of interest. That is to say, an entry M_{ijk} is the number of students who enrolled in class j after class i with a timestep delta k . This allows us to record counts of enrollment as well the amount of time between enrollments without scaling. We then learn some set of parameters $\theta \in \mathbb{R}^{2T+1}$. Note that we learn $2T + 1$ parameters in order to accommodate for the sometimes relationships penalized in the Normalized Discount algorithm presented in section IV, part D.

A loss function and some notion of ground truth was necessary in order to learn these parameters. We constructed an adjacency matrix using the ExploreCourses course descriptions as the ground truth (see section III, part 2), utilized Mean Squared Error as the loss function, and utilized the Adam optimization function in order to train our model as proposed by Kingma et. al [6].

We refrained from training more parameters to limit the degrees of freedom of our model and ensure that the model does not simply overfit to the ground truth results. We deviate from the traditional machine learning paradigms because we are solely interested in the objective of fitting the model to our dataset and not in the objective of generalizing to unseen data.

F. Null Model

The null model is a graph built by randomly generating edges for the Carta network. The underlying sequence matrix used to build this graph is replaced by a randomly sampled series of data. For each student we assume that the student selects four classes at random for each quarter they attended Stanford. The null model also makes the assumption that all students attend Stanford for 12-quarters (4 years) which is a reasonable approximation of the length of time an undergraduate student attends Stanford. Additionally, we assume that for each quarter a student takes exactly 4 classes which are sampled randomly without replacement from all of the available courses at Stanford. The sample probability of a class is directly proportional to the total number of students that have enrolled in that course since Fall of 2000. To obtain a probability we then divide this total count by the total number of students that have been enrolled in any Stanford class since Fall of 2000. By randomly assigning classes to students over 12-quarters, we can simulate a series of student enrollment data with which we will build our null model. This simulated data will be stored in a null sequence matrix, S_{null} , which will be passed to the algorithms that build the prerequisite projection graphs. As we do before, we can now plot the sequence scores to determine the optimal number of nodes to include in our graph. Running the DiscountNorm algorithm on both the Carta network and the null model and observing the top 5000 edges ranked by generated prerequisite score, we observed that $n = 1000$ would provide us with the most interesting results of both projections.

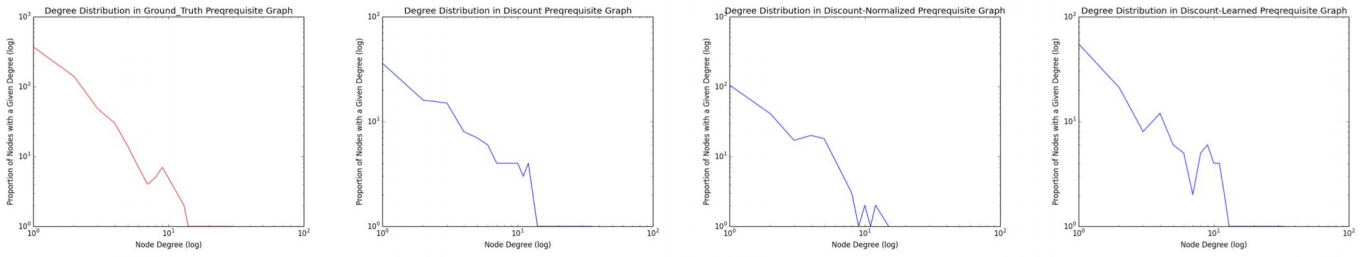


Fig. 3. The degree distributions found in the ground truth prerequisite graph (shown here in red) and three of the proposed graph projection algorithms—Discount, Normalized Discount, and Learned Discount. We note here that they have similar forms, revealing that there are some courses that act as prerequisites to many other courses.

V. RESULTS & DISCUSSION

We partition our results section into two parts, the first section being a discussion about the experiments run to confirm a network exists in the sequence data and the second section being a discussion of our proposed projection algorithms relative to our notion of ground truth, the prerequisite graph generated from the ExploreCourses course descriptions.

A. Null Model Comparison

We confirmed a set of intuitions about the structure of the real-world network before transitioning into evaluating the accuracy of our prerequisite projection graph.

1) *Edge Score Distribution*: One task was to discern whether or not clear temporal structure existed in the real-world network. Specifically, we looked to observe the distribution of edge scores. We hypothesized that the real-world network would produce few, high-scoring directed edges and a vast number of low-scoring directed edges. This would confirm that, in the Carta network, there are strong relationships between courses with prerequisite relationships and no relationship between courses otherwise. We see in Fig. 2 a comparison between the preliminary scoring mechanisms as they are applied to the real-world network and its null model equivalent. The structure of the network is evident we see here that when applied to the real-world network, each of the three scoring mechanisms demonstrate a clear exponential relationship between the score and ranking. When the mechanisms are applied to the null model, we observe a near uniform distribution of the scores between the edges. This is a strong signal that temporal dependencies exist in the real-world network, which confirm our intuition that a prerequisite structure can be drawn out of the Carta network in the way that we have defined and constructed it.

2) *Academic Major Modularity*: Another feature which further reaffirms our belief that a latent prerequisite structure in the Carta network exists is that we can observe higher modularity of sets within academic majors. We hypothesized that prerequisite projections created with the scoring mechanisms defined above would lead to the observation of communities. Intuitively, this can be interpreted as students taking courses in some sequence in order to complete academic major requirements. In Fig. 4, we show that this

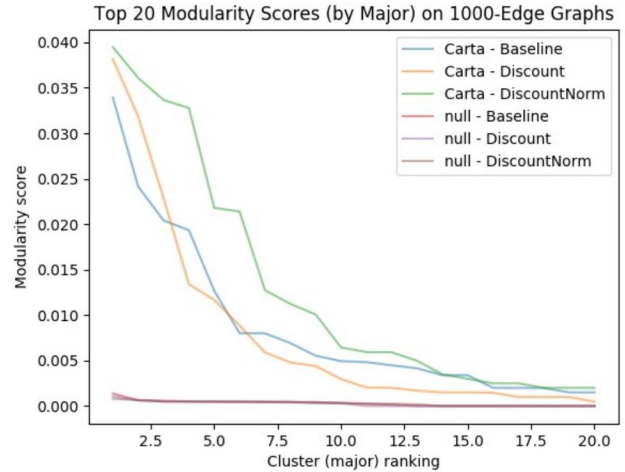


Fig. 4. The modularity scores of the top 20 most connected majors in the networks generated by our three preliminary prerequisite projection algorithms on both the Carta network and the null model. High major modularity confirms that students complete major-specific sequences during their time at Stanford. Of the top 20 sampled from the Carta Network, approximately two-thirds across all projections were STEM majors.

phenomenon does indeed express itself in our projections across the first three graph generation algorithms proposed—the baseline, Discount, and Normalized Discount. The chart plots the modularity of the most modular sets where the nodes of each set are grouped solely by academic major. The edges used were simply the top-1000 edges as determined by the scoring mechanisms, here evaluated in an unweighted manner.

The two experiments explained above showed strong signal that there indeed existed latent structures within the Carta enrollment data that signified the possibility of crafting a prerequisite projection graph. Therefore, we moved forward with two more experiments that incorporated a notion of ground truth to our problem formulation in order to guide our decision making process to create a more accurate representation of the relationships found in the dataset.

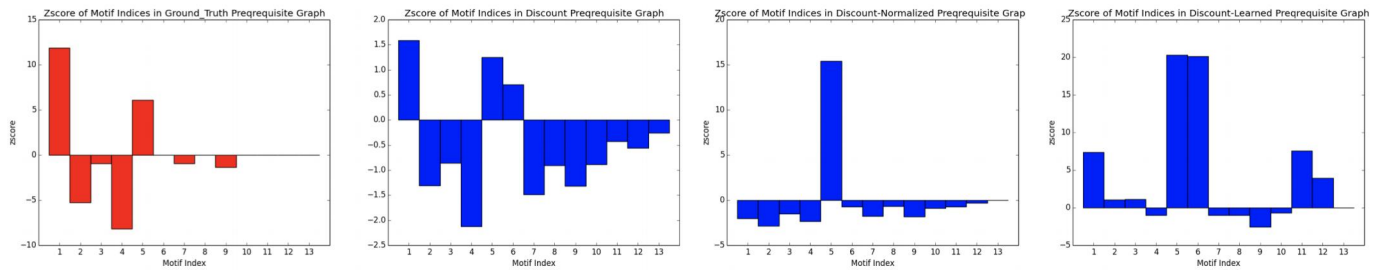


Fig. 5. The motif distribution for the ground truth prerequisite graph and three of the proposed graph projection algorithms— Discount, Normalized Discount, and Learned Discount. Note that the motifs are similarly over and under represented in both of the graphs.

B. Performance Evaluation

In the previous subsection we conducted a preliminary analysis of our predicted subgraphs against our null model. In this subsection and in the following, we will conduct further analysis of the local and global structural similarities between our generated predicted course sequence model and our generated ground-truth sequence data. Using the course description data (see section III, part B), we can create a ‘ground-truth’ course sequence graph. As before, each node in this graph represents a particular course, and an edge is drawn between courses for courses listed as prerequisites in the course description.

1) *Degree Distribution Comparison:* We first explore global structural similarities by analyzing the node degree distributions of our generated models against the distribution of node degrees in the ground-truth graph. In figure 3 we illustrate the varying degree distributions between the different types of graph projections we generate against the distribution of the ground truth model. In each of these we observe a roughly exponential relationship between the node degree and the count of nodes with this degree. Moreover, we observe that the highest node degree is roughly the same across the models. This suggests courses acting as prerequisites for many courses are present in both our predicted projections and in the ground truth projection. As a result, on a global scale both the ground-truth graph and our predicted models show similar structures.

2) *3-Motif Distribution Comparison:* In this section, we analyze local structural similarities between our generated course sequence graph and the ground truth graph. We do so by enumerating the occurrence of motifs of size 3 in each of our generated graphs. Refer to figure in our appendix for the indices which correspond to the observed motifs of size 3. In figure 7 we illustrate the varying motif distributions between the different types of graph projections we generate. Notice that the ground-truth model lists a particularly high incidence of motifs 1 and 5. Qualitatively, this observation seems reasonable if we consider how courses sequences are structured. Motif 1 illustrates the example of a course being required as a prerequisite to two others. This makes sense since a large amount of courses list introductory classes as prerequisite (e.g. many courses require students to have taken MATH 51 as a

basic linear algebra course). Motif 5 similarly makes intuitive sense, since there exist courses (particularly in the School of Engineering) that have several requirements which themselves have some prerequisite structure. For instance, CS 161 lists both CS 109 and CS 106 as prerequisites, but CS 109 also requires CS 106 to have been taken beforehand.

With the introduction of discounting into our course sequence model, we observe that both motifs 1 and 5 are detected in large proportion by our model. Notice, however, that this model also has a considerable occurrence of motif 6. Intuitively, motifs 5 and 6 are very similar, the only difference resulting from the fact that in motif 6 both of the classes which are required by a third class have some mutual requirement. In the context of course sequences, however, we would hope to not see this sort of relationship frequently since there should not exist mutual prerequisites between courses. Fortunately, the discount normalized model no longer shows the occurrence of motif 6, but still a very pronounced occurrence of motif 5. Unfortunately, we lose the recovery of motif 1. It makes sense that we do not observe motif 6 in this model, since we explicitly penalize bi-directional edges (i.e. courses that mutually list each other as requirements). In our final model, where the discounted factor has been learned by an attention-based neural network, we are again able to recover both the common presence of motif 1 and 5. Interestingly, however, this model also shows a large degree of motifs 6, 11 and 12. The frequency of these later motifs may be due to the fact that the neural network approach does not have a notion to penalize the co-occurrence of bi-directional edges. Instead, the network has to learn these intuitive heuristics. In our case, it is not able to do so, perhaps because it is limited by the amount of data.

3) *Qualitative Analysis:* In addition to providing the aforementioned quantitative arguments, we have provided samples of the top 16 sequential course relationships returned to us by our final course sequence graph, ordered in decreasing order of sequence score in Fig. 6. Notice that we are able to recover some well-known course sequences offered at Stanford, such as the introductory CS sequence (CS 106A → CS 106B), the introductory economics sequence (ECON 50 → ECON 51) and the introductory physics sequence (PHYSICS 41 → PHYSICS 43). Note that there are several entries that show an almost “interchangeable” prerequisite relationship between certain courses, particularly in the HUMBIO department. We

Previous Course	Next Course	Sequence Score
PWR1	PWR2	49.86
HUMBIO3A	HUMBIO4A	49.06
HUMBIO3B	HUMBIO4B	48.67
HUMBIO3B	HUMBIO4A	48.56
HUMBIO3A	HUMBIO4B	48.36
CHEM33	CHEM35	46.71
CHEM31A	CHEM31B	45.28
HUMBIO2B	HUMBIO3B	44.11
HUMBIO2A	HUMBIO3A	44.02
HUMBIO2A	HUMBIO3B	43.99
HUMBIO2B	HUMBIO3A	43.58
CS106A	CS106B	42.80
CHEM35	CHEM131	41.85
CHEM31B	CHEM33	41.11
ECON50	ECON51	41.06
PHYSICS41	PHYSICS43	37.74

Fig. 6. The top 16 edges ranked by the score calculated by the DiscountNorm prerequisite projection algorithm on the Carta network. We observe that this algorithm captures relationships in course sequences taken heavily by undergrads.

discovered that this was because of high co-enrollment at the instructors' behest.

VI. CONCLUSION

We have demonstrated that there are latent structures embedded in the Carta enrollment data. We first generated a bipartite graph from the raw data and then constructed a projection that captured prerequisite relationships between classes using student enrollment behavior during their time at Stanford. Our report also outlines a method with which neural networks can learn to recover course sequences from temporal data via an attention layer over the temporal dimension. These results were validated by comparing the predicted sequence structure of our models with existing, ground-truth course sequences. Future work may want to explore this later analysis in more depth, by leveraging more complex networks like LSTMs and Transformer Networks which incorporate notations of temporal sequences.

VII. BIBLIOGRAPHY

- [1] Brian Karrer and Mark EJ Newman. 2011. Stochastic blockmodels and community structure in networks. *Physical Review E*.
- [2] Chang, J., and Blei, D. M. 2010. Hierarchical relational models for document networks. *The Annals of Applied Statistics* 124150.
- [3] Christopher Aicher, Abigail Z. Jacobs, and Aaron Clauset. 2014. Learning latent block structure in weighted networks. *Journal of Complex Networks*.
- [4] Snigdha Chaturvedi, Hal Daume III, Taesun Moon, and Shashank Srivastava. 2012. A topical graph kernel for link prediction in labeled graphs. In *Proceedings of the International Conference of Machine Learning*.

[5] Xu, Jie, Tianwei Xing, and Mihaela Van Der Schaar. "Personalized course sequence recommendations." *IEEE Transactions on Signal Processing* 64.20 (2016): 5340-5352.

[6] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).

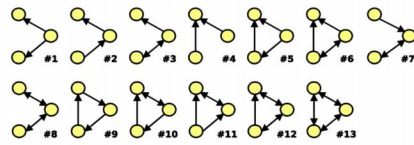


Fig. 7.