Concept Graph Learning from Educational Data

Yiming Yang, Hanxiao Liu, Jaime Carbonell, Wanli Ma
Carnegie Mellon University
{yiming, hanxiaol, jgc, mawanli}@cs.cmu.edu

ABSTRACT

This paper addresses an open challenge in educational data mining, i.e., the problem of using observed prerequisite relations among courses to learn a directed universal concept graph, and using the induced graph to predict unobserved prerequisite relations among a broader range of courses. This is particularly useful to induce prerequisite relations among courses from different providers (universities, MOOCs, etc.). We propose a new framework for inference within and across two graphs—at the course level and at the induced concept level—which we call Concept Graph Learning (CGL). In the training phase, our system projects the course-level links onto the concept space to induce directed concept links; in the testing phase, the concept links are used to predict (unobserved) prerequisite links for test-set courses within the same institution or across institutions. The dual mappings enable our system to perform an interlingua-style transfer learning, e.g. treating the concept graph as the interlingua, and inducing prerequisite links in a transferable manner across different universities. Experiments on our newly collected data sets of courses from MIT, Caltech, Princeton and CMU show promising results, including the viability of CGL for transfer learning.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Concept learning; Induction; Parameter learning; I.5.2 [**Design Methodology**]: Pattern analysis; Classifier design and evaluation

General Terms

Algorithms; Performance; Design; Experimentation; Theory

Keywords

Multi-scale directed graph learning; online education; transfer learning; new inference algorithms

WSDM'15, February 2–6, 2015, Shanghai, China.

ACM 978-1-4503-3317-7/15/02 ...$15.00.
http://dx.doi.org/10.1145/2684822.2685292.

1. INTRODUCTION

The large and growing amounts of online education data present both open challenges and significant opportunities for machine learning research to enrich educational offerings. One of the most important challenges is to automatically detect the prerequisite dependencies among massive quantities of online courses, and to support decision making such as curricula planning for students, and to support course and curriculum design by teachers based on existing course offerings. One example is to find a coherent sequence of courses among MOOC offerings from different providers that respect implicit prerequisite relations. A more specific example would be a new student who just enters a university for a MS or PhD degree. She is interested in machine learning and data mining courses, but finds it difficult to choose among many courses which look similar or with ambiguous course titles to her, such as *Machine Learning, Statistical Machine Learning, Applied Machine Learning, Machine Learning with Large Datasets, Scalable Analytics, Advanced Data Analysis, Statistics: Data Mining, Intermediate Statistics, Statistical Computing*, and so on. Taking all the courses would imply taking forever to graduate, and possibly waste a big portion of her time due to the overlapping content. Alternately, if she wants to choose a small subset, which courses should she include? How should she order the courses without sufficient understanding about the prerequisite dependencies? Often prerequisites are explicit within an academic department but implicit across departments. Moreover, if she already took several courses in machine learning or data mining through Coursera or in her undergraduate education, how much do those courses overlap with the new ones? Without an accurate representation about how course contents overlap with each other, and how course ordering would affect the efficiency or difficulty of learning by students, it is difficult to answer her questions. Universities solve this problem in the old-fashioned way, via academic advisors, but it is not clear how to solve it in MOOC environments or cross-university offerings where courses do not have unique ID's and not described in a universally controlled vocabulary. Ideally, we would like to have a universal graph whose nodes are canonical and discriminant concepts (e.g. "*convexity*" or "*eigenvalues*") being taught in a broad range of courses, and whose links indicate pairwise preferences in sequencing the teaching of these concepts. For example, to learn the concepts of *PageRank* and *HITS*, students should have already learned the concepts of *eigenvectors, Markov matrices* and *irreducibility of matrices*. This means directed links from *eigenvectors, Markov*

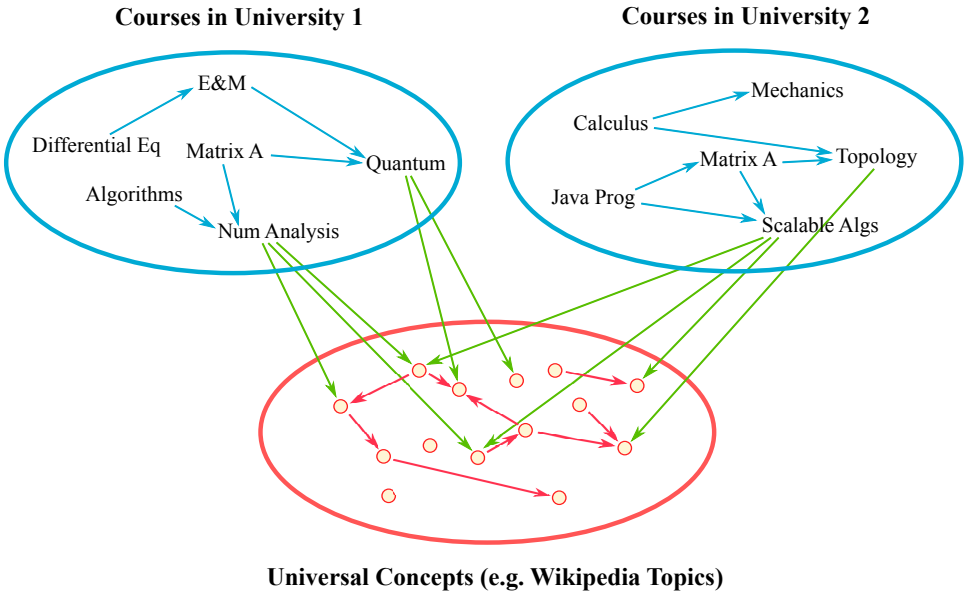

Figure 1: The framework of two-scale directed graphs: The higher-level graphs have courses (nodes) with perquisite relations (links). The lower-level graph consists of universal concepts (nodes) and pairwise preference in learning or teaching concepts. The links between the two levels are system-assigned weights of concepts to each course.

matrices and *irreducibility* to *PageRank* and *HITS* in the concept graph. To generalize this further, if there are many directed links from the concepts in one course (say *Matrix Algebra*) to the concepts in another course (say *Web Mining* with *Link Analysis* as a sub-topic), we may infer a prerequisite relation between the two courses. Clearly, having a directed graph with a broad coverage of universal concepts is crucial for reasoning about course content overlap and prerequisite relationship, and hence important for educational decision making, such as curriculum planning by students and modularization in course syllabus design by instructors.

How can we obtain such a knowledge-rich concept graph? Manual specification is obviously not scalable when the number of concepts reaches tens of thousands or larger. Using machine learning to automatically induce such a graph based on massive online course materials is an attractive alternative; however, no statistical learning techniques have been developed for this problem, to our knowledge. Addressing this open challenge with principled algorithmic solutions is the novel contribution we aim to accomplish in this paper. We call our new method Concept Graph Learning (CGL). Specifically, we propose a multi-scale inference framework as illustrated in Figure 1, which consists of two levels of graphs and cross-level links. Generally, a course would cover multiple concepts, and a concept may be covered by more than one course. Notice that the course-level graphs do not overlap because different universities do not have universal course IDs. However, the semantic concepts taught in different universities do overlap, and we want to learn the mappings between the non-universal courses and the universal concept space based on online course materials.

In this paper we investigate the problem of concept graph learning (CGL) with our new collections of syllabi (including course names, descriptions, listed lectures, prerequisite relations, etc.) from the Massachusetts Institute of Technology (MIT), the California Institute of Technology (Caltech), the Carnegie Mellon University (CMU) and Princeton. The syllabus data allow us to construct an initial course-level graph for each university, which may be further enriched by discovering latent pre-requisite links. As for representing the universal concept space, we study four representation schemes (Section 2.2), including 1) using the English words in course descriptions, 2) using sparse coding of English words, 3) using distributed word embedding of English words, and 4) using a large subset of Wikipedia categories. For each of these representation schemes, we provide algorithmic solutions to establish a mapping from courses to concepts, and to learn the concept-level dependencies based on observed prerequisite relations at the course level. The second part, i.e., the explicit learning of the directed graph for universal concepts, is the most unique part of our proposed framework. Once the concept graph is learned, we can predict *unobserved* prerequisite relations among any courses, including those not in the training set and by different universities. In other words, CGL enables an interlingua-style transfer learning as to train the models on the course materials of some universities and to predict the prerequisite relations for the courses in other universities. The universal transferability is particularly desirable in MOOC environments where courses are offered by different instructors in many universities. Since universities do not have unified course IDs, the course-level sub-graphs of different universities do not overlap with each other, and the prerequisite links are only local within each sub-graph. Thus to enable cross-university transfer, it is crucial to project course-level prerequisite links in different universities onto the directed links among universal concepts.

The bi-directional inference between the two directed graphs makes our CGL framework fundamentally different from existing approaches in graph-based link detection [16, 19,

20], matrix completion [4, 9, 13] and collaborative filtering [24]. That is, our approach requires explicit learning of the concept-level directed graph and optimal mapping between the two levels of links while other methods do not (see Section 4 for more discussion).

Our main contributions in this paper can be summarized as:

1. A novel framework for within- and cross-level inference of prerequisite relations at the course-level and the concept-level directed graphs;
2. New algorithmic solutions for scalable graph learning;
3. New data collections from multiple universities with syllabus descriptions, prerequisite links and lecture materials;
4. The first evaluation for prerequisite link prediction in within- and cross-university settings.

The rest of the paper is organized as follows: Section 2 introduces the formal definitions of our framework and inference algorithms; Section 3 describes the new data sets we collected for this study and future benchmark evaluations, and reports our empirical findings; Section 4 discusses related work; and Section 5 summarizes the main findings in this study.

2. FRAMEWORK & ALGORITHMS

2.1 Notation

Let us formally define our methods with the following notation.

- n is the number of courses in a training set;
- p is the dimension of the universal concept space (Section 2.2;
- $x_i \in \mathbb{R}^p$ for $i = 1, 2, \ldots, n$ are the bag-of-concepts representation of a course in the training set;
- X is an n-by-p matrix where each row is $x_i^\top$;
- Y is an n-by-n matrix where each cell is the binary indicator of the prerequisite relation between two courses, i.e., $y_{ij} = 1$ means that course j is a prerequisite of course i, and $y_{ij} = -1$ otherwise.
- A is a p-by-p matrix, whose elements are the weights of directed links among concepts. That is, A is the matrix of model parameters we want to optimize given the training data in X and Y.

2.2 Representation Schemes

What is the best way to represent the contents of courses to learn the universal concept space? We explore different answers with four alternate choices as follows:

1. ***Word-based Representation (Word)***: This method uses the vocabulary of course descriptions plus any listed keywords by the course providers (MIT, Caltech, CMU and Princeton) as the entire concept (feature) space. We applied standard procedures for text preprocessing, including stop-word removal, term-frequency (TF) based term weighting, and the removal of the rare words whose training-set frequency is one. We did not use TF-IDF weighting because the relative small number of "documents" (courses) in our data sets do not allow reliable estimates of the IDF part.

2. ***Sparse Coding of Words (SCW)***: This method projects the original n-dimensional vector representations of words (the columns in the course-by-word matrix) onto sparse vectors in a smaller k-dimensional space using Non-negative Matrix Factorization [18], where k is much smaller than n. One can view the lower dimensional components as the system-discovered latent concepts. Intrigued by the successful application of sparse coding in image processing [11], we explored its application to our graph-based inference problem. By applying the algorithm in [14] to our training sets we obtained a k-dimensional vector for each word; by taking the average of word vectors in each course we obtained the bag-of-concept representation of the course. This resulted in an n-by-k matrix X, representing all the training-set courses in the k-dimensional space of latent concepts. We set $k = 100$ in our experiments based on cross validation.

3. ***Distributed Word Embedding (DWE)***: This method also uses dimension-reduced vectors to represent both words and courses, similar to SCW. However, the lower dimensional vectors for words are discovered by multilayer neural networks through deep learning, based on word usage w.r.t contextual, syntactic and semantic information [17, 21]. Intrigued by the popularity of DWE in recent research in Natural Language Processing and other domains [6, 5], we explored its application to our graph-based inference problem. Specifically, we deploy DWE trained on Wikipedia articles by Rami et al. [1], in a domain which is semantically close to that of academic courses.

4. ***Category-based Representation (Cat)***: This method used a large subset of Wikipedia categories as the concept space. We selected the subset via a pooling strategy as follows: We used the words in our training-set courses to form 3509 queries (one query per course), and retrieved the top 100 documents per query based on cosine similarity. We then took the union of the Wikipedia category labels of these retrieved documents, and removed the categories which were retrieved by only three queries or less. This process resulted in a total of 10,051 categories in the concept space. The categorization of courses was based on the earlier highly scalable very-large category space work reported in [10]: the classifiers were trained on labeled Wikipedia articles and then applied to the word-based vector representation of each course for (weighted) category assignments.

Each of the above representation schemes may have its own strengths and weaknesses. *Word* is simple and natural but rather noisy, because semantically equivalent lexical variants are not unified into canonical concepts and there could be systematic vocabulary variation across universities. Also, this scheme will not work in cross-language settings, e.g., if courses descriptions are in English and Chinese. *Cat* would be less noisy and better in cross-language settings,

but the automated classification step will unavoidably introduce errors in category assignments. *SCW* (sparse coding of words) reduces the total number of model parameters via dimensionality reduction, which may lead to robust training (avoiding overfitting) and efficient computation, but at the risk of losing useful information in the projection from the original high-dimensional space to a lower dimensional space. *DWE* (distributed word embedding) deploys recent advances in deep learning of word meanings in context. However, reliable word embedding require the availability of large volumes of training text (e.g., Wikipedia articles); the potential mismatch between the training domain (for which large volumes of data can be obtained easily) and the test domain (for which large volumes of data are hard or costly to obtain) could be a serious issue. Yet another distinction among these representation schemes is that Word and Cat produce human-understandable concepts and links, while *SCW* and *DWE* produce latent factors which are harder to interpret by humans.

By exploring all the four representation schemes in our unified framework for two-level graph based inference, and by examining their effectiveness in the task of link prediction of prerequisite relations among courses, we aim to obtain a deeper understanding of the strengths and weaknesses of those representational choices.

2.3 The Optimization Methods

We define the problem of concept graph learning as a key part of learning-to-predict prerequisite relations among courses, i.e., for the two-layer statistical inference we introduced in Section 1 with Figure 1. Given a training set of courses with a bag-of-concepts representation per course as a row in matrix X, and a list of known prerequisite links per course as a row in matrix Y, we optimize matrix A whose elements specify both the direction (sign) and the strength (magnitudes) of each link between concepts. We propose two new approaches to this problem: a classification approach and a learning-to-rank approach. Both approaches deploy the same extended versions of SVM algorithms with squared hinge loss, but the objective functions for optimization are different. We also propose a nearest-neighbor approach for comparison, which predicts the course-level links (prerequisites) without learning the concept-level links.

2.3.1 The Classification Approach (CGL.Class)

In this method, we predict the score of the prerequisite link from course i to course j as:

$$\hat{y}_{ij} = f_A(x_i, x_j) = x_i^\top A x_j \tag{1}$$

The intuition behind this formula is shown in Figure 2. It can be easily verified that the quantity $x_i^\top A x_j$ is the summation of the weights of all the paths from node i to node j in this graph, where each path is weighted using the product of the corresponding x_{ik}, $A_{kk'}$ and $x_{jk'}$.

The criterion for optimizing matrix A given training data x_i for $i = 1, 2, \ldots, n$ and true labels y_{ij} for all course pairs is defined as:

$$\min_A \sum_{i,j} \left(1 - y_{ij}\left(x_i^\top A x_j\right)\right)_+^2 + \frac{\lambda}{2}\|A\|_{\mathrm{F}}^2 \tag{2}$$

where $(1-v)_+ = \max(0, 1-v)$ denotes the hinge function, and $\|\cdot\|_{\mathrm{F}}$ is the matrix Frobenius norm. The 1st term in formula 2 is the empirical loss; the 2nd term is the regularization term, controlling the model complexity based on the large margin principle. We choose to use the squared hinge loss $(1-v)_+^2$ the first term to gain first-order continuity of our objective function, enabling efficient computation using accelerated gradient descent [22, 23]. This efficiency improvement is crucial because we operate on pairs of courses, and thus have a much larger space than in normal classification (e.g. classifying individual courses).

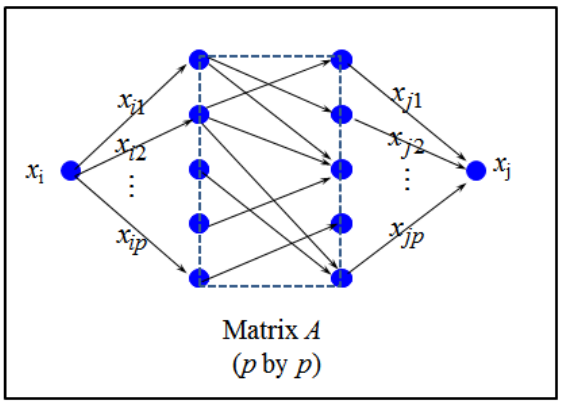

Figure 2: The weighted connections from course i to course j via matrix A which encodes the directed links between concepts

2.3.2 The Learning-to-Rank Approach (CGL.Rank)

Inspired by the learning-to-rank literature [12], we explored going beyond the binary classifier in the previous approach to one that essentially learns to rank prerequisite preferences. Let Ω_i be the set of course pairs with the true labels $y_{ij} = 1$ for different j's, and $\bar{\Omega}_i$ the pairs of courses with the true labels $y_{ij} = -1$ for different k's, we want our system to give all the pairs in Ω_i higher score than that of any pair in $\bar{\Omega}_i$. We call this the partial-order preference over links conditioned on course i. Let T be the union of tuple sets $\{(i,j,k)|(i,j) \in \Omega_i, (i,k) \in \bar{\Omega}_i\}$ for all i in $1, 2, \ldots, n$. We formulate our optimization problem as:

$$\min_A \sum_{(i,j,k)\in T} \left(1 - \left(x_i^\top A x_j - x_i^\top A x_k\right)\right)_+^2 + \frac{\lambda}{2}\|A\|_{\mathrm{F}}^2$$

Or equivalently, we can rewrite the objective as:

$$\min_A \sum_{(i,j,k)\in T} \left(1 - \left(\left(XAX^\top\right)_{ij} - \left(XAX^\top\right)_{ik}\right)\right)_+^2 + \frac{\lambda}{2}\|A\|_{\mathrm{F}}^2$$

Solving this optimization problem requires us to extend standard packages of SVM algorithms in order to improve computational efficiency because the number of model parameters (p^2) in our formulation is very large. For example, with the vocabulary size of 15,396 words in the MIT dataset, the number of model parameters is over 237 million. When p (the number of concepts) is much larger than n (the number of courses), one may consider solving this optimization problem in the dual space instead in the primal space. However, even in the dual space, the number of variables is still $O(n^3)$, corresponding to all the tuples of $i, j, k \in \{1, 2, \ldots, n\}$, and

the kernel matrix is on the order of of $O(n^6)$. We address these computational challenges in Section 2.4.

2.3.3 The Nearest-Neighbor Approach (kNN)

Different from the two approaches above where matrix A plays a central role, as a different baseline, we propose to predict the prerequisite relationship for any pair of courses based on matrix X and Y but not A. Let (u, v) be a new pair of courses in the test set, and x_u and x_v be the bag-of-concept representations of u and v, respectively. We score each pair (i, j) in the training set as:

$$\hat{y}_{ij} = \left(x_u^\top x_i\right) \times \left(x_v^\top x_j\right) \quad \forall i, j = 1, 2, \ldots, n$$

By taking the top pairs in the training set and by assigning the corresponding $y_{i,j}$ to the new pairs, we perform the k-nearest neighbor computation. If we normalize the vectors, the dot-products in the above formula become the cosine similarity. This approach requires nearest-neighbor search on-demand; when the number of course pairs in the test set is large, the online computation would be substantial.

Via cross validation, we found that k = 1 (1NN) works best for this problem on the current data sets.

2.4 Scalable Algorithms

As we have pointed out in Section 2.3.2 , the p^2 model parameters in matrix A and the n^3 constraints make the optimization of CGL.Rank computationally challenging. No standard optimization packages can be applied for large p and n. We address this problem by reformulating the objective in a way that the true number of model parameters reduces to $O(n^2)$.

Recall that our original objective (Section 2.3.2) is:

$$\min_{A \in \mathbb{R}^{p \times p}} \sum_{(i,j,k) \in T} \ell\left(\hat{Y}_{ij} - \hat{Y}_{ik}\right) + \frac{\lambda}{2}\|A\|_{\mathrm{F}}^2 \text{ s.t. } \hat{Y} = XAX^\top \tag{3}$$

where $\hat{Y}$ is n-by-n, the matrix of the predicted course-level links. A is a p-by-p matrix of the induced concept-level links (weighted). $\ell(v) = (1-v)_+^2$ is the squared hinge loss. Denote the optimal solution of (3) by A^*, we are going to show that this solution can always be written as $A^* = X^\top BX$ some n-by-n ($n \ll p$) matrix B.

We argue that A^* must be identical to the optimal solution of the following problem:

$$\min_{A \in \mathbb{R}^{p \times p}} \|A\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad XAX^\top = XA^*X^\top := \hat{Y}^* \tag{4}$$

Otherwise, the solution for (4) will lead to the same loss as A^* in (3) but a smaller regularization penalty, which is contradictory to our assumption that A^* is the optimal solution for (3). Notice that (4) is just the matrix version of the *minimum norm problem for an underdetermined linear system* (because $p^2 > n^2$), which has a closed-form solution [2, 3, 8] as below:

$$A^* = X^\top K^{-1} \hat{Y}^* K^{-1} X \tag{5}$$

where $K = XX^\top$. By defining $B := K^{-1}\hat{Y}^*K^{-1}$, we have $\hat{Y}^* = KBK$ and $A^* = X^\top BX$. Hence the regularization in (3) becomes

$$\begin{aligned}\|A\|_{\mathrm{F}}^2 = \mathrm{tr}\left(AA^\top\right) &= \mathrm{tr}\left(X^\top BXX^\top B^\top X\right) \\ &= \mathrm{tr}\left(XX^\top BXX^\top B^\top\right) = \mathrm{tr}\left(\hat{Y}B^\top\right)\end{aligned}$$

and (3) can be rewritten as an optimization w.r.t B which only involves $O\left(n^2\right)$ variables:

$$\min_{B \in \mathbb{R}^{n \times n}} \sum_{(i,j,k) \in T} \ell\left(\hat{Y}_{ij} - \hat{Y}_{ik}\right) + \frac{\lambda}{2}\mathrm{tr}\left(\hat{Y}B^\top\right) \text{ s.t. } \hat{Y} = KBK \tag{6}$$

The substantially reduced number of parameters in (6) allows us to efficiently compute the gradient even for extremely large p^2 (by reducing the per-iteration cost). Moreover, the smoothness of our objective function enables us to deploy Nestrerov's accelerated gradient descent [22, 23], ensuring a convergence rate in the order of t^{-2} where t is the number of steps, i.e., the reduced number of iterations.

2.5 Algorithm Implementation

The gradient for (6) w.r.t B is given by

$$-2K\left(\sum_{(i,j,k) \in T: \delta_{ijk} > 0} e_i \left(e_j - e_k\right)^\top \delta_{ijk}\right) K + \lambda \hat{Y}$$

here $\delta_{ijk} = 1 - \left(\hat{Y}_{ij} - \hat{Y}_{ik}\right)$, $\hat{Y} = KBK$ and e_i, e_j, e_k are all unit vectors. The detailed implementation of the accelerated gradient descent is summarized in Alg. 1.

Algorithm 1 Concept Graph Learning

```
procedure CGL.Rank(X, T, λ, η)
    K ← XX^⊤
    t ← 1, B ← 0, Q ← 0
    while not converge do
        Δ ← 0
        Ŷ ← KBK
        for (i, j, k) in T do
            δ_ijk ← 1 − (Ŷ_ij − Ŷ_ik)
            if δ_ijk > 0 then
                Δ_ij ← Δ_ij + δ_ijk
                Δ_ik ← Δ_ik − δ_ijk
        P ← B − η(λŶ − 2KΔK)
        B ← P + ((t−1)/(t+2))(P − Q)
        Q ← P
        t ← t + 1
    A ← X^⊤BX
    return Ŷ, A
```

3. EXPERIMENTS

3.1 Data and Metrics

We collected course listings, including course descriptions and available pre-requisite structure from MIT, Caltech, CMU and Princeton[1]. The first two were complete course catalogs, and the latter two required spidering and scraping, and hence we collected only Computer Science and Statistics for CMU, and Mathematics for Princeton. This implies that we can test within-university prerequisite discovery for all four—though MIT and Caltech will be most comprehensive—and cross university only pairs where the training university contains the disciplines in the test university.

Table 1 summarizes the datasets statistics.

[1]The datasets are available at `http://nyc.lti.cs.cmu.edu/teacher/dataset/`

Table 1: Datasets Statistics

University	# Courses	# Prerequisites	# Words
MIT	2322	1173	15396
Caltech	1048	761	5617
CMU	83	150	1955
Princeton	56	90	454

Table 2: Results of within-university prerequisite prediction

Algorithm	Data	AUC	MAP
CGL.Rank	MIT	0.96	0.46
CGL.Class	MIT	0.86	0.34
1NN	MIT	0.76	0.30
CGL.Rank	Caltech	0.95	0.33
CGL.Class	Caltech	0.86	0.27
1NN	Caltech	0.60	0.16
CGL.Rank	CMU	0.79	0.55
CGL.Class	CMU	0.70	0.38
1NN	CMU	0.75	0.43
CGL.Rank	Princeton	0.92	0.69
CGL.Class	Princeton	0.89	0.61
1NN	Princeton	0.82	0.58

To evaluate performance, we use the *Mean Average Precision* (MAP) which has been the preferred metric for in information retrieval for evaluating ranked lists, and the Area Under the Curve of ROC (ROC/AUC or simply AUC) which in popular in link detection evaluations.

3.2 Within-university Prerequisite Prediction

We tested all the methods on the dataset from each university. We used one third of the data for testing, and the remaining two thirds for training and validation. We conducted 5-fold cross validation on the training two-thirds, i.e., trained the model on 80% of the training/validation data set, and tuned extra parameters on the remaining 20%. We repeated this process 5 times with a different 80-20% spit in each run. The results of the 5 runs were averaged in reporting results. Figure 3 and Table 2 summarize the results of CGL.Rank, CGL.Class and 1NN. All the methods used the English words as the representation scheme in this first set of experiments. Notice that the AUC scores for all methods are much higher than the MAP scores. The high AUC scores derive in large part from the fact that AUC gives an equal weight to the system-predicted true positives, regardless of their positions in system-produced ranked lists. On the other hand, MAP weighs more heavily the true positives in higher positions of ranked lists. In other words, MAP measures the performance of a system in a harder task: Not only the system needs to find the true positives (along with false positives), it also needs to rank them higher than false positives as possible in order to obtain a high MAP score. Using a more concrete example, a totally useless system which makes positive or negative predictions at random with 50% of the chances will have an AUC score of 50%. But this system will have an extremely low score in MAP because the change for a true positive to randomly appear in the top of a ranked list will be low when true negatives dominate in the domain. Our data sets are from such a domain because each course only requires a very small number of other courses as prerequisites. Back to our original point, regardless the popularity of AUC in link detection evaluations, its limitation should be recognized: the relative performance among methods is more informative than the absolute values of AUC.

Table 3: CGL.Rank with four representations

Concept	Data	AUC	MAP
Word	MIT	0.96	0.46
Cat.	MIT	0.93	0.36
SCW	MIT	0.93	0.33
DWE	MIT	0.83	0.09
Word	Caltech	0.95	0.33
Cat.	Caltech	0.93	0.32
SCW	Caltech	0.91	0.22
DWE	Caltech	0.76	0.12
Word	CMU	0.79	0.55
Cat.	CMU	0.77	0.55
SCW	CMU	0.73	0.43
DWE	CMU	0.67	0.35
Word	Princeton	0.92	0.69
Cat.	Princeton	0.84	0.68
SCW	Princeton	0.82	0.60
DWE	Princeton	0.77	0.50

As we can see in Figure 3, the relative ordering of the methods in AUC and MAP are indeed highly correlated across all the data sets. MAP emphasizes positive instances in the top portion of each ranked list, and hence is more sensible for measuring the usefulness of the system where the user interacts with system-recommended prerequisite lists. Comparing the results on all the methods, we see that CGL.Rank dominates the others in both AUC and MAP on most data sets.

3.3 Effects of Representation Schemes

Figure 4 and Table 3 summarize the results of CGL.Rank with the four representation schemes, i.e., Word, Cat, SCW (sparse coding of words) and DWE (distributed word embedding), respectively. Again the scores of AUC and MAP are not on the same scale, but the relative performance suggest that Word and Cat are competitive with each other (or Word is slightly better one some data sets), followed by SCW and then DWE. In the rest of the empirical results reported in this paper, we focus more on the performance of CGL.Rank with Word as the representation scheme because they perform better, and the space limit does not allow us to present all the results for every possible permutation of method, scheme, dataset and metric.

3.4 Cross-university Prerequisite Prediction

In this set of experiments, we fixed the same test sets which were used in within-university evaluations, but we alter the training sets across universities, yielding transfer learning results where the models were trained with the data from a different university than those where they were tested. By fixing the test sets in both within- and cross-university evaluations we can compare the results on a common basis. The competitive performance of Cat in comparison with Word is encouraging, given that Wikipedia categories are defined as general knowledge, and the classifiers (SVM's) we used for category assignment to courses were

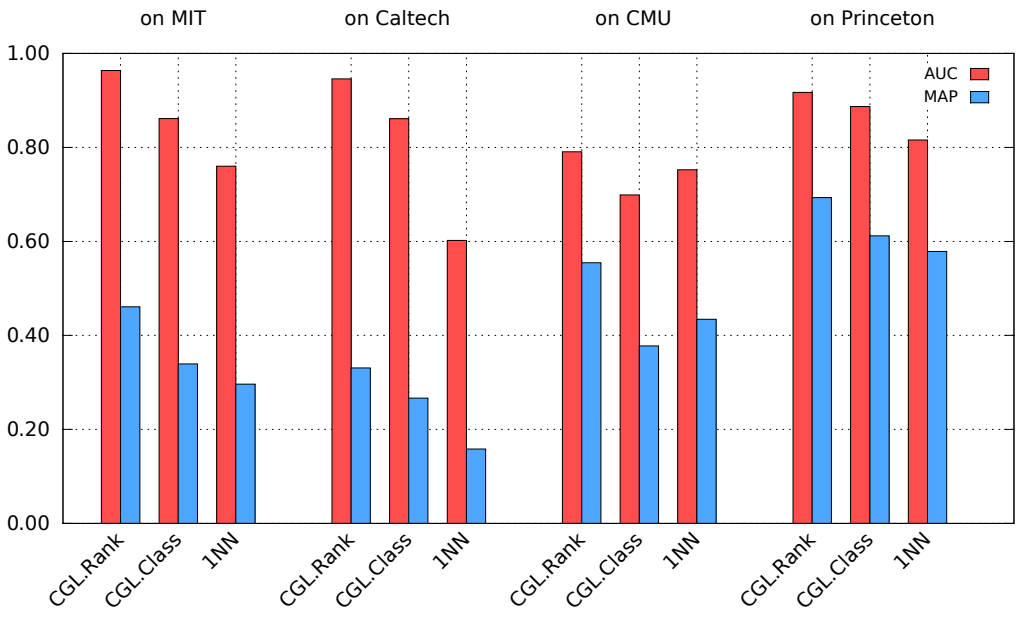

Figure 3: Different methods in within-university prerequisite prediction: All the methods used words as concepts.

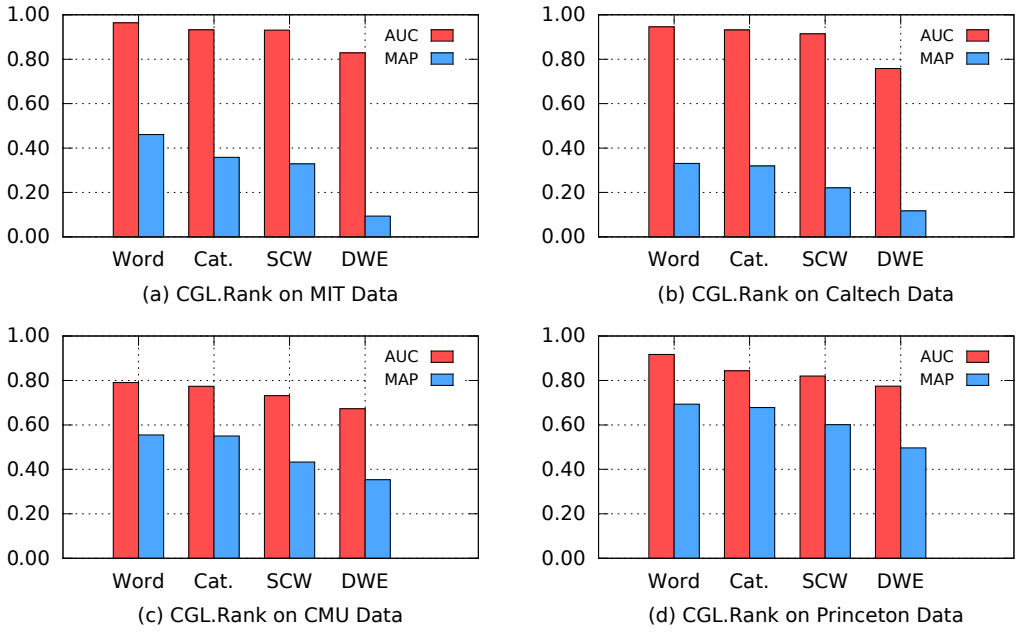

Figure 4: CGL.Rank with different representation schemes in within-university prerequisite prediction

trained on Wikipedia articles instead of the course materials (because we do not have human assigned Wikipedia category labels for courses). This means that the Wikipedia categories indeed have a good coverage on the concepts being taught in universities (and probably MOOC courses), and that our pooling strategy for selecting a relevant subset of Wikipedia categories is reasonably successful.

Table 4 and Figure 5 show the results of CGL.Rank (using words as concepts). Recall that the MIT and Caltech data cover the complete course catalogs, while the CMU data only cover the Computer Science and Statistics, and the Princeton data only over the Mathematics. This implies that we can only measure transfer learning on pairs where the training university contains the disciplines in the test university. By comparing the red bars (when the training university and the test university is the same) and the blue bars (when the training university is not the same as the test university), we see some performance loss in the transfer learning, which is expected. Nevertheless, we do get transfer, and this is the first report on successful transfer learning of educational knowledge, especially the prerequisite structures in disjoint graphs, across different universities through a unified concept graph. The results are therefore highly encouraging and suggest continued efforts to improve. Those results also suggest some interesting points, e.g., MIT might have a better coverage of the topics taught in Caltech, compared to the inverse. And, MIT courses seem to be closer to those in Princeton (Math) compared with those of CMU.

Table 4: CGL.Rank in within-university and cross-university settings

Training	Test	MAP	AUC
MIT	MIT	0.46	0.96
Caltech	MIT	0.13	0.88
Caltech	Caltech	0.33	0.95
MIT	Caltech	0.25	0.86
CMU	CMU	0.55	0.79
MIT	CMU	0.34	0.70
Caltech	CMU	0.28	0.62
Princeton	Princeton	0.69	0.92
MIT	Princeton	0.46	0.72
Caltech	Princeton	0.43	0.58

3.5 Experiment Details

We tested the efficiency of our proposed algorithms on a single machine with an Intel i7 8-core processor and 32GB RAM. On the largest MIT dataset with 981,009 training tuples and 490,505 test tuples, it took 3.08 minutes for CGL.Rank to reach a convergence rate of 1e-3 at 103 iterations with only 401MB memory. We also tested the effectiveness of using an accelerated version of gradient descent. Without the acceleration, that is, when using the ordinary gradient descent instead, it took 37.3 minutes and 1490 iterations to reach the same objective value. CGL.Class is equally efficient as CGL.Rank in terms of run time, though the latter is superior in terms of result quality. As for our 1NN baseline, it took 2.88 hours since a huge number ($2 \times 981,009 \times 490,505$) of dot-products need to be computed on the fly.

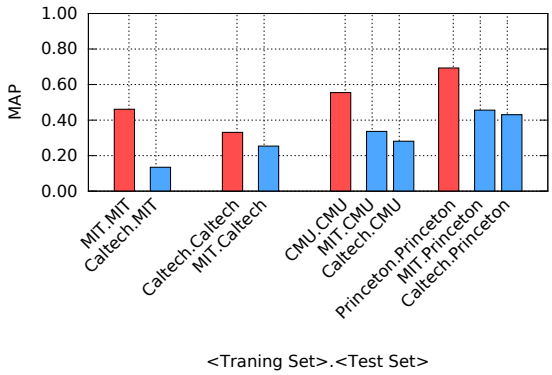

Figure 5: CGL.Rank results in within-university (red) and cross-university (blue) prerequisite prediction

4. DISCUSSION AND RELATED WORK

Whereas the task of inferring prerequisite relations among courses and concepts, and the task of inferring a concept network from a course network in order to transfer learned relations are both new, as are the extensions to the SVM algorithms presented, our work was inspired by methods in related fields, primarily:

In collaborative filtering via matrix completion the literature has focused on only one graph, such as a bipartite graph of u users and m preferences (e.g. for movies or products). Given some known values in the u-by-m matrix, the task is to estimate the remaining values (e.g. which other unseen movies or products would each user like) [24]. This is done via methods such as affine rank minimization [9] that reduce to convex optimization [3].

Another line of related research is transfer learning [7, 25, 26]. We seek to transfer prerequisite relations between pairs of courses within universities to other pairs also within universities, and to pairs that span universities. This is inspired by but different from the transfer learning literature. Transfer learning traditionally seeks to transfer informative features, priors, latent structures and more recently regularization penalties [15]. Instead, we transfer shared concepts in the mappings between course-space and concept-space to induce prerequisite relations.

Although our evaluations primarily focus on detecting prerequisite relations among courses, such a task is only one direct application of the automatically induced universal concept graph. Other important applications include automated or semi-automated curriculum planning for personal education goals based on different backgrounds of students, and modularization in course syllabus design by instructors. Both tasks require interaction between humans (students or teachers) and the system-induced concept graph, and exploring options and optimal paths. How well can our system support such a need? Although we do not have a formal evaluation in that aspect in this paper, we believe that of our system-induced links among concepts would be suggestive and useful. The fact that CAT (i.e., the category-based representation scheme) performs relatively well in our evaluations gives some comfort, and visual inspection of the inferred links at the concept level also supports that conclusion, though with occasional errors, such as the one high-

Table 5: Examples of linked Wikipedia categories in our system-induced concept graph

1st Wikipedia Category	Linked Wikipedia Category
"Partial differential equations"	"Ordinary differential eqns."
"Ethnography"	"Geography"
"Extrasolar Planets"	"Chaos Theory"
"British Industrial designers"	"Graphic Design"

lighted in red in Table 5 below. The first column in the table are examples of system-assigned Wikipedia categories to courses in our data sets; the second column consists of the linked Wikipedia categories with the higher weight (assigned by our system) given each category in the first column, which means that the concept in the 2nd-column should be taught before the corresponding concept in the first column.

5. CONCLUDING REMARKS

We conducted a new investigation on automatically inferring directed graphs at both the course level and the concept level, to enable prerequisite prediction for within-university and cross-university settings.

We proposed three approaches: a classification approach (CGL.Class), a learning to rank approach (CGL.Rank), and a nearest-neighbor search approach (kNN). Both CGL.Class and CGL.Rank (deploying adapted versions of SVM algorithms) explicitly model concept-level dependencies through a directed graph, and support an interlingua-style transfer learning across universities, while kNN makes simpler prediction without learning concept dependencies. To tackle the extremely high-dimensional optimization in our problems (e.g., 2×10^8 links in the concept graph for the MIT courses), our novel reformulation CGL.Rank enables the deployment of fast numerical solutions. On our newly collected datasets from MIT, Caltech, CMU and Princeton, CGL.Rank proved best under MAP and ROC/AUC, and computationally much more efficient than kNN.

We also tested four representation schemes for document content: using the original words, using Wikipedia categories as concepts, using a distributed word representation, and using sparse word encoding. The first two: original words and Wikipedia-derived concepts proved best. Our results in both the within-university and cross-university settings are highly encouraging.

We envision that the cross-university transfer learning of our approaches is particularly important for future application to MOOCs where courses come from different providers and across institutions, and there are seldom any explicit prerequisite links. A rich suite of future work includes:

- Testing on cross-university or cross-course-provider prerequisite links. We have tested cross-university transfer learning, but the inferred links are within each target university, rather than cross-institutional links. A natural extension of the current work is to predict cross-institutional prerequisites. For this kind of evaluation we will need labeled ground truth of cross-university prerequisites.
- Cross-language transfer. Using the Wikipedia categories and Wikipedia entries in different languages, it would be an interesting challenge to infer prerequisite relations for courses in different languages by mapping to the Wikipedia category/concept interlingua.
- Extensions of the inference from single source to multiple sources, from single media (text) to multiple media (including videos), and from single granularity level (courses) to multiple levels (including lectures).
- Deploying the induced concept graph for personalized curriculum planning by students and for syllabus design and course modularization by teachers.

6. ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation (NSF) under grant IIS 1350364.

7. REFERENCES

[1] R. Al-Rfou, B. Perozzi, and S. Skiena. Polyglot: Distributed word representations for multilingual nlp. *arXiv preprint arXiv:1307.1662*, 2013.

[2] A. Ben-Israel and T. N. Greville. *Generalized inverses*, volume 13. Springer, 2003.

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[5] Y. Chen, B. Perozzi, R. Al-Rfou, and S. Skiena. The expressive power of word embeddings. *arXiv preprint arXiv:1301.3226*, 2013.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[7] C. Do and A. Y. Ng. Transfer learning for text classification. In *NIPS*, 2005.

[8] D. W. Fausett and C. T. Fulton. Large least squares problems involving kronecker products. *SIAM Journal on Matrix Analysis and Applications*, 15(1):219–227, 1994.

[9] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.

[10] S. Gopal and Y. Yang. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 257–265. ACM, 2013.

[11] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[12] T. Joachims, H. Li, T.-Y. Liu, and C. Zhai. Learning to rank for information retrieval (lr4ir 2007). In *SIGIR Forum*, volume 41, pages 58–62, 2007.

[13] C. R. Johnson. Matrix completion problems: a survey. In *Proceedings of Symposia in Applied Mathematics*, volume 40, pages 171–198, 1990.

[14] H. Kim and H. Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, 30(2):713–730, 2008.

[15] M. Kshirsagar, J. Carbonell, and J. Klein-Seetharaman. Transfer learning based

methods towards the discovery of host-pathogen protein-protein interactions. In *Proc of ISMB*, volume 40, pages 171–198, 1990.

[16] J. Kunegis and A. Lommatzsch. Learning spectral graph transformations for link prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 561–568. ACM, 2009.

[17] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[18] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[19] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[20] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252. ACM, 2010.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[22] Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[23] Y. Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Mateaticheskie Metody*, 24:509–517, 1988.

[24] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[25] L. Yang, S. Hanneke, and J. Carbonell. A theory of transfer learning with applications to active learning. *Machine learning*, 90(2):161–189, 2013.

[26] J. Zhang, Z. Ghahramani, and Y. Yang. Flexible latent variable models for multi-task learning. *Machine Learning*, 73(3):221–242, 2008.