PATHFINDER: Graph-based Itemset Embedding for Learning Course Recommendation and Beyond

Jason (Jiasheng) Zhang Thai Le Yiming Liao Dongwon Lee

College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA, 16802, USA

{jpz5181, tql3, yiming, dongwon}@psu.edu

***Abstract*—We demonstrate a tool, named as PATHFINDER, that captures and visualizes rich latent relationships among courses as a graph, mines students' past course performance data, and recommends pathways or top-k courses most helpful to a given student, using an itemset embedding based learning model. With dedicated design for the asymmetric, non-additive and non-negative challenges specific to the problem, our model for helpfulness achieves the best performance among competing models. We demonstrate the visualization of four course relationships (e.g., mandatory, prerequisite, helpful, and top-k) in a graph. The PATHFINDER demo is publicly available at:** `http://pike.psu.edu/pathfinder/`

***Index Terms*—PATHFINDER, recommendation, course, graph**

I. INTRODUCTION

When today's students look for new courses to take, they often use two fundamental Information Retrieval functions–i.e., either *browsing* a list of courses in some sorted criteria or *searching* courses using some keywords and filtering interfaces.

To complement these effective but outdated functions, we present an AI-powered tool for education, named as PATHFINDER, that is novel in three aspects: (1) PATHFINDER captures rich latent relationships among courses (e.g., prerequisite, helpful, similar) as a graph and visualizes them intuitively; (2) PATHFINDER leverages students' past course performance data for more effective recommendation; and (3) PATHFINDER recommends the most useful course pathways for a student using the Itemset Embedding method.

A. Motivation

Consider two courses, MATH200 and PHYS210, at the same level and two groups of students, A and B, with similar characteristics (e.g., major, prior GPA). If A has taken MATH200 before PHYS210 and obtained better grades in both courses than B who took PHYS210 before MATH200, then it is reasonable to recommend a pathway of "MATH200 $\rightarrow$ PHYS210" to a new student whose characteristics are similar as A. However, in general, it is more challenging to derive such conclusion as A may have taken different courses before taking MATH200 that may have contributed to better grades in MATH200 and PHYS210.

To capture this complex intertwined *helpfulness* relationship among courses, we model the problem as the *itemset embedding learning problem* [1], [2]. The itemset in [1] is behavioral contexts, while it is a set of courses taken in the past in this work. This problem also relates to many of recommendation problem [3], or more specifically, the next-basket recommendation problem [4]–[7]. In addition, students' course enrollment data can also be modeled as a dynamic bipartite network, with students and courses as two sets of nodes and enrollment relationships as edges weighted by grades, related to the network embedding problem [8]–[10].

B. Challenges

However, our problem is also unique: (1) *Asymmetric*: The helpfulness relationship between two courses is asymmetric [11] in that MATH200 may be helpful to take first before PHYS210 but not vice versa. As a result, popular methods based on the notion of similarity (e.g., network embedding) are not appropriate; (2) *Non-additive*: The contributions of helpfulness from a set of courses are non-additive. For example, taking two very similar courses is not necessarily more helpful than taking only one. Therefore, the additive models such as [1], [5] are not applicable; and (3) *Non-negative*: The helpfulness relationship between two courses is non-negative. Therefore, taking more courses will not be less helpful than taking a subset of those courses.

II. AI IN PATHFINDER

A. Problem Formulation

Given a set of students $u \in U$, a set of courses $c \in C$, students' enrollment data is a set of tuples $E = \{\langle u, c, g, \tau \rangle\}$, where $g \in [0, 1]$ is the normalized grade and τ is the semester when a student s takes the course c. Then, our task is to extract the asymmetric relationship of a pair of courses (c_s, c_t), such that taking c_s before c_t will result in a better grade of c_t. In the itemset embedding learning framework [1], the enrollment data is reorganized as the set of tuples $E' = \{\langle C_s, G_s, c_t, g \rangle\}$, where C_s is a set of courses $c_s \in C_s$ that a student takes before the course c_t, G_s is the corresponding grade of the student on those courses in C_s, g is the student's grade in c_t, and $|E'| = |E|$.

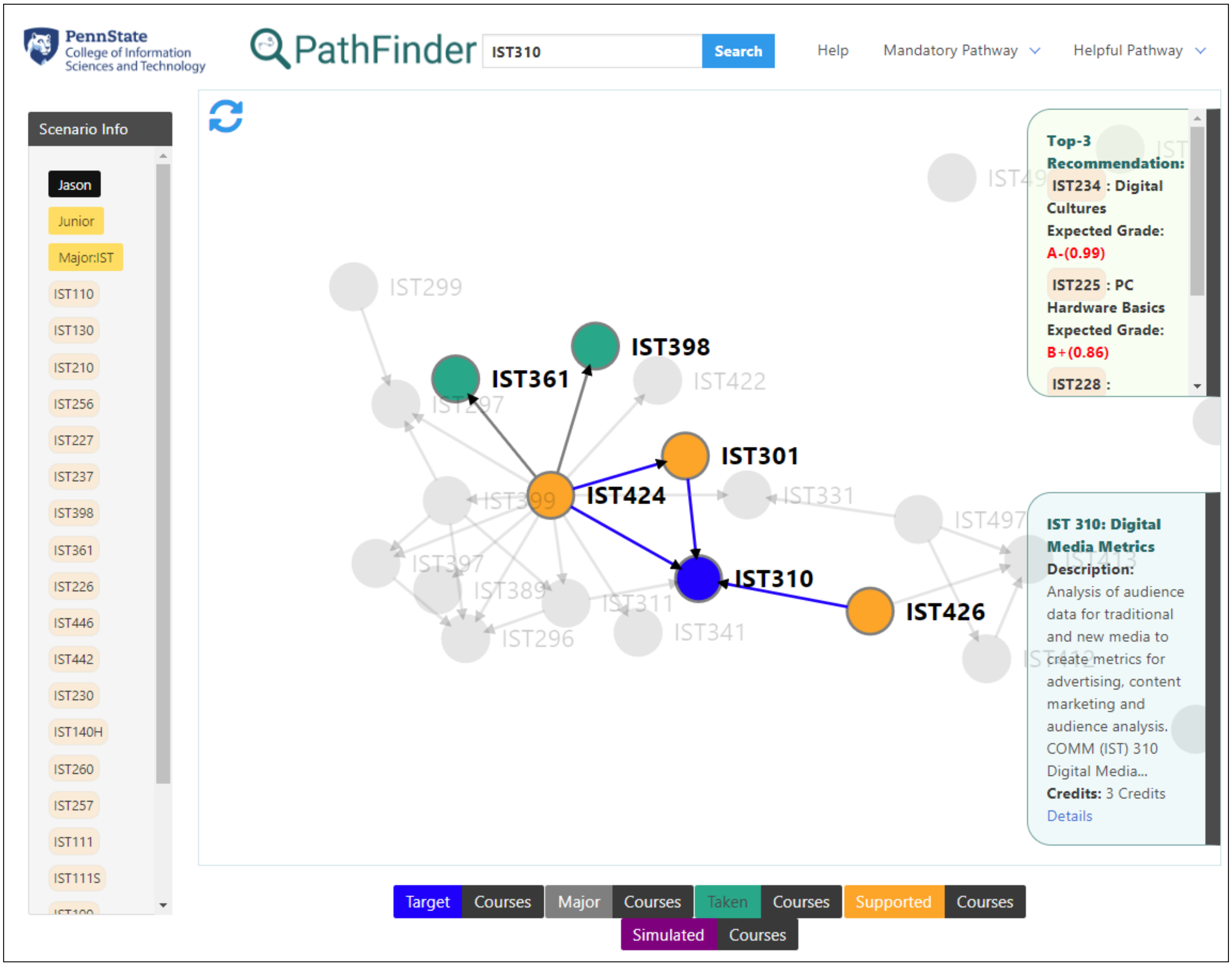

Fig. 1. With Green nodes indicating courses that a student has already taken, Orange nodes illustrate *helpful* or *required* courses for a target course in Blue. On the top right side shown top-3 recommended courses to the user with predicted grades. The bottom right shows the detail information of the selected course.

B. Model Design and Learning

In the PATHFINDER model, each course c is represented as an embedding vector $v_c \in \mathbf{R}^D$, which describes the latent skill requirement of the course. Upon finishing a course c, the student will obtain the skills of c, weighted by the grade g. In the itemset setting, given C_s and G_s, the student u's skill level is $v_u(C_s) \in \mathbf{R}^D$ with each dimension $d \in [D]$ calculated as:

$$v_u^d(C_s) = \max\{v_u^d(c_s) * g_{c_s} | c_s \in C_s\}, \tag{1}$$

where the max-pooling in Eq.1 ensures the consistency with the *non-additive* and *non-negative* properties of the helpfulness relationship. For the *non-additive*, adding a course c_s' to a taken course set C_s such that $\exists c_s \in C_s$, $v_{c_s'} = v_{c_s}$ makes no change to v_u. For the *non-negative*, it is easy to observe that $v_u^d(C_s) \geq v_u^d(C_s')$ for any $C_s' \subset C_s$.

Next, given the target course c_t, represented as v_{c_t}, the predicted grade $\widehat{g}$ of c_t is calculated as:

$$\widehat{g} = \sigma(-\sum_{d\in[D]} clip(v_{c_t}^d - v_u^d(C_s), 0)), \tag{2}$$

where σ is the Sigmoid function, $clip(x, 0)$ is the clip function that returns x if $x > 0$, otherwise 0. The Eq.2 compares the latent skill requirement v_{c_t} of the target course c_t and the student's skill level v_u to predict the student's performance in c_t. The clip function ensures that only insufficient skill dimensions ($d \in [D]$ s.t. $v_{c_t}^d \geq v_u^d(C_s)$) in the comparison will count, which is consistent with the intuition. For example, a student excellent in Mathematics does not necessarily perform well in a music course. In addition, the asymmetric relationship between c_s and c_t in Eq.2 ensures the consistency with *asymmetric* helpfulness relationships. The PATHFINDER model learns the parameters (embedding vectors v_c) by fitting the enrollment data as:

$$v = argmin_v \sum_{\langle C_s, G_s, c_t, g\rangle \in E'} (g - \widehat{g})^2. \qquad (3)$$

The optimization can be solved by Stochastic Gradient Descent (SGD). Given the course embedding vectors, the helpfulness relationship of an ordered pair of courses (c_s, c_t) can be calculated as $h(c_s, c_t) = \sigma(-\sum_{d\in[D]} clip(v_{c_t} - v_{c_s}, 0))$, which is simply the estimated grade of c_t when c_s is the only taken course. More naturally, one can also predict the grade of a target course c_t, given a set of taken courses C_s and their grades G_s.

III. Demonstration

To demonstrate PathFinder, shown in Figure 1, we have scraped $|C| = 8,620$ real course information from Penn State course catalog, and synthesized $|U| = 50,000$ students over 12 semesters and $|E| = 2,017,399$ enrollment data for $U \times C$, based on the randomly generated course embedding vectors v_c.

During the demonstration, we will show off the validity and usefulness of PathFinder using several mock-up but realistic scenarios–e.g., a Junior student “Jason” is scheduling courses to enroll.

- Jason is interested in taking a course IST310, and wants to verify whether he has fulfilled all (chains of) prerequisite courses. Using the pre-built prerequisite relationships among all courses, then, PathFinder can show the entire *prerequisite* sub-graph for the target course.
- Jason asks senior students what other courses have been useful for them to take before IST310. For this requirement, PathFinder aggregates the experiences from a large set of students (i.e., training set) who have taken IST310, and shows the *helpfulness* sub-graph, along with its prediction on Jason’s performance in IST310.
- Using Eq. 2, PathFinder can predict Jason’s performance in any target courses. Further, PathFinder enables Jason to do *what-if* type analysis for a selected set or chain of courses to take (e.g., comparing two scenarios between taking a pathway of IST310 $\rightarrow$ IST410 vs. IST310 $\rightarrow$ MATH250 $\rightarrow$ IST410). Figure 2 and 3 illustrate these functions.
- When Jason is not sure about what to take, PathFinder provides a lazy function that recommends *top-k* courses that Jason is likely to perform well based on Jason’s past performance as well as those of other students who have taken similar courses as Jason.
- For the detail information of a course Jason is interested in, he may hover the curse over the course node and the course information will be shown at the right corner of the page, as shown in Figure 1.

IV. Experiments

We prove the validity of PathFinder by comparison with state-of-the-art next-basket recommendation methods on grade prediction problem using the synthetic dataset.

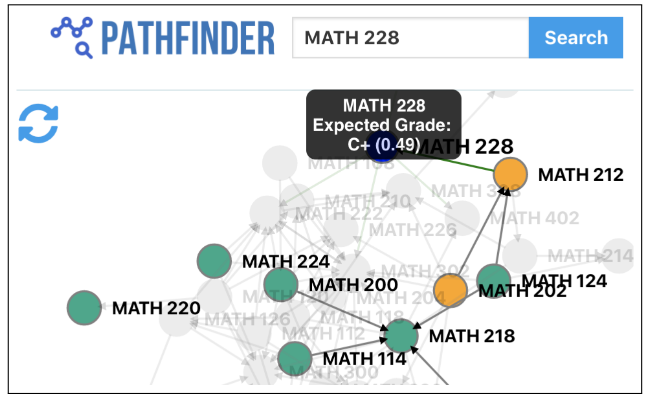

Fig. 2. GUI showing current predicted grade on target MATH 228 course in Blue is *”C+”* and recommending to take MATH 212 and MATH 202 courses in Orange.

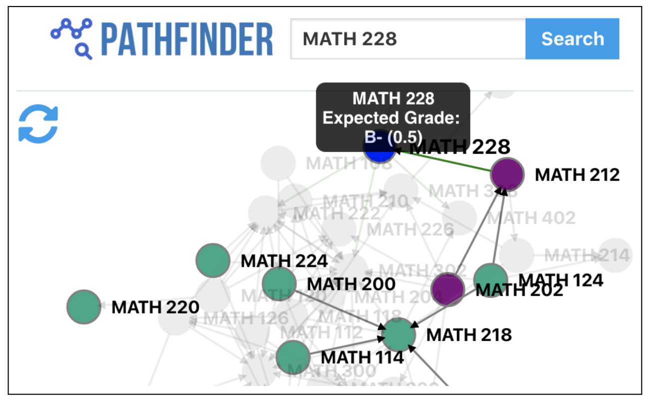

Fig. 3. GUI simulates *what-if* situation, predicting that the grade for target course MATH 288 in Blue *will* improve from *”C+”* to *”B-” if* the two suggested courses MATH 212 and MATH 202 are pre-taken (shown as Purple)

A. Setting

For enrollment data of each student, we uniformly chose the tuples E' with target courses c_t (and their grades g) in one semester e_{test} as test data; those with target courses in the $e_{valid} = e_{test} - 1$ semester are validation data, and those with target courses in e_{train}, $\forall e_{train} < e_{valid}$ semesters are training data. Mean-Square-Error (MSE) is used as performance measure.

B. Competing Methods

A list of next-basket recommendation methods are compared.

- FISM [12] (Factorized Item Similarity Model): It factorized item-item similarity matrix to predict new item for a user given her historical behavior. It does not explicitly consider the sequential nature of the problem, but stand as a state-of-the-art method of recommendation. We adopted it to include grades (rating in recommendation) as input.
- Fossil [5] (Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation): Based on FISM, Fossil explicitly consider the sequential nature of the problem by including order-specific weighted Markov

TABLE I
EXPERIMENT RESULTS

Method	MSE	Improvement % from RNN
FISM	1.41 $\times 10^{-3}$	-25.9%
Fossil	1.26 $\times 10^{-3}$	-12.5%
RNN	1.12 $\times 10^{-3}$	0%
PATHFINDER-a	4.2 $\times 10^{-4}$	62.5%
PATHFINDER-i	3.3 $\times 10^{-4}$	70.5%
PATHFINDER	**3.0** $\times 10^{-4}$	73.2%

chains for historical behavior. We adopted it to include grades as input.

- RNN [13] (Recurrent Neural Network): The sequential nature of the next-basket recommendation problem naturally brings RNN as a valid method. We adopt the same embedding design for source and target courses as FISM and Fossil. A GRU (Gated Recurrent Unit) structure is applied to encode the historical behavior. Average pooling is used to get the historical embedding as the sequence output. The inner product of the sequence output and target course embedding is the output.
- PATHFINDER: The method is proposed in this work.
- PATHFINDER-a: It replace the max-pooling aggregation of PATHFINDER in Eq.1 with summation, which lost the Non-additive and Non-negative conditions.
- PATHFINDER-i: It replace the prediction method of PATHFINDER in Eq.2 with sigmoid of inner product, which lost the Asymmetric condition.

C. Result

As the result shown in Table. I, PATHFINDER performs significantly better than existing methods. A closer comparison between FISM and Fossil shows the necessity of explicitly considering sequential nature of the problem. RNN, the most general model, did performs better than Fossil and FISM, but fell behind PATHFINDER. The reason is that no conditions of the problem (e.g., Asymmetric, Non-additive, and Non-negative) are utilized in RNN to restrict the model as PATHFINDER does, which makes learning from limited data set challenging. This can be further validated by the comparison between PATHFINDER and PATHFINDER-a, PATHFINDER and PATHFINDER-i, respectively.

V. FURTHER APPLICATIONS

In this paper, we have demonstrated that PATHFINDER model can accommodate three unique characteristics of an itemset embedding learning problem specifically in education domain. However, our model design and learning can also be adapted into a variety of similar recommendation problems in which there exists *Asymmetric, Non-additive, and Non-negative* constraints within relationships among recommended items.

We may take recommending tourism destinations as an example. Particularly, we want to recommend attractions for tourists by formulating a next-basket recommendation problem such that given a sequence of visited place, we want to recommend the next best places to visit. In this case, it is might helpful to recommend a place before (or after) the others (or the visited) due to user's personal priority or knowledge requisite, i.e. experience at one place (e.g. acquired historical background) is helpful to enjoy that of others (*Asymmetric*). Moreover, suggesting two very similar places (e.g. two different restaurants of a same chain of retail) might not bring more richness into the trip (*Non-additive*). Furthermore, giving more options (while maintaining other constraints) will not deteriorate the overall experience (*Non-negative*).

Last but not least, while we use an "offline" school setting to illustrate PATHFINDER, the same model can be customized to adopt in Massive Open Online Course (MOOC) domain as well. As MOOC is becoming more prevalent with thousands of skill sets being offered through several learning channels, PATHFINDER can help provide a better learning experience for the users as well.

VI. ACKNOWLEDGEMENT

This work was in part supported by NSF awards #1422215, #1742702, #1820609, #1915801, and #1934782. We appreciate reviewers for their careful and constructive comments.

REFERENCES

[1] D. Wang, M. Jiang, Q. Zeng, Z. Eberhart, and N. V. Chawla, "Multi-type itemset embedding for learning behavior success," in *KDD*, 2018, pp. 2397–2406.
[2] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, "Attention-based transactional context embedding for next-item recommendation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *WWW*, 2001, pp. 285–295.
[4] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.
[5] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *ICDM*, 2016, pp. 191–200.
[6] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 729–732.
[7] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 403–412.
[8] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, pp. 701–710.
[9] M. Gao, L. Chen, X. He, and A. Zhou, "Bine: Bipartite network embedding," in *SIGIR*, 2018, pp. 715–724.
[10] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Transactions on Knowledge and Data Engineering*, 2018.
[11] C. Zhou, Y. Liu, X. Liu, Z. Liu, and J. Gao, "Scalable graph embedding for asymmetric proximity." in *AAAI*, 2017, pp. 2942–2948.
[12] S. Kabbur, X. Ning, and G. Karypis, "Fism: factored item similarity models for top-n recommender systems," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 659–667.
[13] R. Devooght and H. Bersini, "Long and short-term recommendations with recurrent neural networks," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, ser. UMAP '17. ACM, 2017, pp. 13–21. [Online]. Available: http://doi.acm.org/10.1145/3079628.3079670