PM K-LightGCN: Optimizing for Accuracy and Popularity Match in Course Recommendation

YIDING RAN, HENGCHANG HU, and MIN-YEN KAN, National University of Singapore

A growing body of literature on educational recommenders focuses on accuracy but neglects how it can marginalize user experience. Accuracy optimization fits the interaction data, whereas user experience optimization recognizes students' limited knowledge and recommends better alternatives. We propose a multi-objective course recommender that balances the optimization of both objectives: 1) accuracy, and 2) student experience. For the first objective, we take inspiration from K-Nearest Neighbors (KNN) model's success in course recommendation, even outperforming contemporary neural network based models. KNN's focus on the pairwise relation between close neighbors aligns with the nature of course consumption. Hence, we propose K-LightGCN which uses KNN models to supervise embedding learning in state-of-the-art LightGCN and achieves a 12.8% accuracy improvement relative to LightGCN. For the second objective, we introduce metric PP-Mismatch@K to quantify user experience. We propose PM K-LightGCN which post-filters K-LightGCN's outputs to optimize PP-Mismatch@K and achieve a 17% improvement in student experience with minimal drop in accuracy.

Additional Key Words and Phrases: Multi-Objective Recommender, Course Recommender, Course Popularity

1 INTRODUCTION

Course selection in tertiary education are high-stakes decisions that influences both the student's subsequent academic plan and the future career path [8]. The development of artificial intelligence (AI) inspires exploration of AI-enabled course recommender to assist students' decision making. Existing studies largely focuses on accuracy optimization [9, 17], where the sole pursuit is fitting the underlying **student selections**; i.e., course enrollment records. In contrast, *student-centric* course recommenders should also optimize user experience, which is not guaranteed by high accuracy. Pardos and Jiang proposed a course recommender designed for serendipity that recommends previously unknown but relevant courses [25]. Their experiments show that when presented with such new courses, students opt to change to take these courses instead. Hence, accuracy is not the best metric for user experience, as when given more information, students improve their experience by adjusting selections. Thus, accuracy optimization trained using student selections — the ground truth — is insufficient to optimize such high-stakes student experiences.

Students share similar course selection criteria like course content and future career value [6]. But how they assess one course in these aspects differs. Formulating student experience optimization as an objective requires measuring the factors leading to student satisfaction. In a small-scale user study conducted at our institution, students demonstrated disparate interests towards elective courses: when shown detailed course information, some students consistently preferred popular courses whereas others favored niche ones. Hence, student satisfaction is related to whether course popularity matches their interests. Thus, we propose a proxy for student experience, **preference-popularity match**, that describes the ideal situation where students are recommended with courses of their desired popularity.

Both accuracy optimization and preference-popularity match are important objectives of course recommendation. However, one often comes at the cost of another. The former closely fits the data whereas the latter believes students are better off with alternative course plans. To optimize for both and achieve balance, we present a multi-objective course recommender with accuracy optimization as its primary objective and preference-popularity match as a secondary one.

We first conduct a preliminary study that explores the performance of traditional recommenders and modern neural network based models on course recommendation. We then observe the unusual superiority of ItemKNN in accuracy optimization, which is explained by the tight alignment between its focus on pairwise relations among close neighbors and the unique characteristics of course consumption. Inspired by this, we propose a revised version of the state-of-the-art recommendation algorithm, LightGCN. In our revision, pairwise similarity computed by KNN models are used to supervise embedding learning in LightGCN. Then, predictions by KNN models and LightGCN are combined as the final recommendation. The resultant K-LightGCN model optimizes for our first objective — accuracy. We then introduce a metric for student experience, PP-Mismatch@K, which quantifies the level of preference-popularity mismatch in recommendations. To achieve our second objective — preference-popularity match — we propose PM K-LightGCN, which takes in recommendations of K-LightGCN and selects 10 courses that best fit the defined objective. Through evaluation, our multi-objective course recommender demonstrates a good balance between both objectives.

2 RELATED WORK

Educational Recommendation Systems. The education domain is relatively underexplored in recommendation research. Only 44 works were associated with this field in a 2018 mapping study [28]. The unique characteristics of course-taking behavior also add complications to the course recommendation task. First, in institutes of higher learning (hereafter, IHLs), students only take 4 to 6 courses per semester, worsening sparsity issues already challenging recommender implementation [13]. Also, students' course selection is influenced by both intrinsic and extrinsic motivations which can differ across individuals [16, 22]. Additionally, IHL courses are designed to be complementary and often have complex relations such as prerequisites and preclusions [10]. An ideal recommender should learn students' varied interests and consider the underlying constraints in course choice [13].

Approaches to course recommendation include content-based, collaborative filtering and knowledge-based techniques [13]. Content-based approaches learn courses' characteristics from their description and recommend those similar to students' previous enrollments [5, 24, 29]. Collaborative filtering is the most common approach that uses only the enrollment history, represented as an interaction matrix, from which student and course vector representations are learned [7, 12, 21, 23]. Knowledge-based methods use domain knowledge like students' faculty and grades and course prerequisite relationships. Jiang et al. proposed a Goal-Based Course Recommender that uses past semester grades [18].

Use of Popularity in Recommendation Systems. In this paper, we introduce the novel concept of preference-popularity match, which aligns popularity of recommended courses with student preference. One scenario that violates this ideal is recommending popular courses to students preferring niche ones. This *popularity bias* arises when popular items are recommended with higher accuracy and frequency, but less popular items are recommended with lower accuracy [3]. Abdollahpouri et al. show that the 3% items involved in the top 20% of interactions, i.e., the popular items, make up 60–100% of recommended items sourced from different recommendation algorithms [3]. Recommending popular items caters to the needs of the majority, leading to high accuracy. However this leads to popularity bias where fewer users consume less popular items, as many potentially interesting items are not exposed to users [1], which artificially exacerbates lopsided demands in a vicious cycle. Besides, users enjoy popular items to different extents, and those preferring niche ones would suffer from inaccurate recommendations [3]. Existing popularity bias mitigation includes unbiased learning [4, 31] which adjusts the data distribution by re-weighting interaction samples or uses bias-free uniform data to learn unbiased embedding and ranking adjustment [1, 2, 32] which includes a regularization term to ensure exposure of less popular items or re-ranks recommendation lists based on certain conditions, similar to our approach. However, popularity bias mitigation neglects the undesirable scenario where niche courses are recommended to students enjoying popular ones, which motivates our proposal of a more comprehensive concept — preference-popularity match.

Course ID	wing_modede529dfcbb2907e9760eea0875cdd12
Student ID	wing_mod412b5c6d4a88a03e91dfc16dd4d494ff
Faculty	School of Computing
Interaction Semester	1910
Enrollment Semester	1710

Table 1. Enrollment Record Sample

3 EXPERIMENT SETTINGS

We use course enrollment records provided by a leading tertiary educational institution to explore both objectives with accuracy optimization in Section 4 and preference-popularity match in Section 5. Due to privacy concerns, IHLs normally do not disclose their students' enrollment records, making it difficult to test our algorithm on multiple datasets.

3.1 Dataset and Data Split

Our dataset is an anonymized, institutional review board (IRB) approved listing of per-semester course-taking histories of graduated undergraduates from the year 2010 to 2020. There are 41,304 unique students and 5,179 unique courses, which together generate 1.4 million enrollments. Due to privacy considerations, each student and course is assigned a unique masked ID to anonymize every student and course. For each enrollment record, we also have information on the student's faculty, the semester when this enrollment occurred (interaction semester) and the semester when the student matriculated (enrollment semester). A sample of this dataset is shown in Table 1.

The dataset is split into train, validation and test sets mimicking real-life scenario. Course recommendations are usually based on students' past enrollment with reference to the records of graduated students. Hence, we randomly select 35% of students to form Group A (the graduated seniors), and the remaining form Group B (students seeking recommendations). All of Group A's records are allocated for training, whereas Group B's are split into train, validation and test sets, based on the interaction semester. The resultant train–validation–test ratio is approximately 7:1:2.

3.2 Evaluation

The recommendation algorithms are implemented using the RecBole library [30], which provides a unified framework to develop recommendation algorithms. We evaluate recommender performance on two objectives: accuracy by metrics HR@K and NDCG@K as well as preference-popularity match using our self-proposed metric Preference-Popularity Mismatch@K, explained in the upcoming sections. We set $K = 10$ to evaluate the top 10 recommendations.

4 OBJECTIVE 1: ACCURACY OPTIMIZATION

We first optimize the recommender for accuracy. We conduct a preliminary study to identify the unique nature of course consumption which motivates our K-LightGCN proposal. We first lay out notations we use: individual students s_i and courses c_j are referred to by subscripts. The set of all students and courses are of size x and y, respectively. The symmetric item–item (user–user) similarity matrix generated by ItemKNN (UserKNN) is represented by w^{item} (w^{user}).

4.1 Preliminary Study

We shortlist the following non-domain specific models as baselines. Existing course recommendation algorithms are not selected as they do not share standardized input. Many require course description or grade information that are absent from this dataset. The following algorithms take in only course enrollment records.

ItemKNN is an item-based, KNN model [11] which computes recommendation scores based on pairwise item–item similarity. Courses the most similar to those taken by the student are recommended. **UserKNN** is a user-based model that recommends courses taken by students similar to the target student.

Multi-Layer Perceptron Model (MLP) employs a basic neural network structure and represents students and courses as latent vectors for non-linear relation capture [15].

Neural Matrix Factorization (NeuMF) is a specific implementation of Neural Collaborative Filtering (NCF [15]) which fuses Matrix Factorization (MF) and MLP methods so that they reinforce each other.

Light Graph Convolution Network (LightGCN) simplifies Graph Convolution Network (GCN) model to best serve the recommendation task by including only its essential component: neighborhood aggregation [14]. In LightGCN, the embeddings for each student and course is learned by aggregating that of its neighbors. The prediction for the interaction between Student s_i and Course c_j, is the dot product of the embeddings for the student and course, $\hat{y_{s_i,c_j}}$. LightGCN uses Bayesian Personalized Ranking (BPR) loss [26], a pairwise loss that encourages the prediction of an observed interaction to be higher than unobserved ones and penalizes otherwise. The BPR loss is defined in Equation 1 where n_{s_i} represents courses Student s_i has taken previously and $\left\|\mathbf{E}^{(0)}\right\|^2$ is the L_2 regularization term to prevent overfitting.

$$L_{BPR} = -\sum_{i=1}^{x} \sum_{j \in n_{s_i}} \sum_{p \notin n_{s_i}} \ln \sigma(\hat{y}_{s_i,c_j} - \hat{y}_{s_i,c_p}) + \lambda \left\|\mathbf{E}^{(0)}\right\|^2 \tag{1}$$

4.1.1 Results. Baseline performance is shown in Table 2a. Both KNN models outperform the rest despite their much simpler structure, highly uncommon in other recommendation domains [15, 19, 20]. It is possible the nature of course consumption aligns with the underlying inductive bias of KNN models — identifying neighbors of students and courses. Among the deep models, only LightGCN incorporates neighborhood information, outperforming others. This suggests that localized relations captured in neighborhoods are more important than global relations in course recommendation.

	Model	HR@10	NDCG@10
a)	**ItemKNN**	**0.7762**	**0.3337**
	UserKNN	0.7294	0.2521
	MLP	0.6013	0.1946
	NeuMF	0.6458	0.2156
	LightGCN	*0.7008*	*0.2542*
b)	**CN-LightGCN**	*0.7287 (+3.98%))*	*0.2896 (+13.9%)*
c)	**K-LightGCN**	**0.7905 (+1.84%)**	**0.3346 (+0.27%)**

Table 2. Accuracy Optimization Results: a) Baseline Performance (§ 4.1.2), b) Revised LightGCN Performance (performance relative to LightGCN) (§ 4.1.3) and c) K-LightGCN Performance (performance relative to ItemKNN) (§ 4.2).

# layers	HR@10	NDCG@10
1	0.6588	0.2260
2	**0.6950**	**0.2502**
3	0.6938	0.2491
4	0.6897	0.2434
6	0.6798	0.2365

Table 3. LightGCN with Different Layer Numbers. Best performance is bolded.

4.1.2 Superiority of ItemKNN. Accurately explaining such observed superiority of ItemKNN in course recommendation assists in understanding the course recommendation task and designing models accordingly. ItemKNN computes the pairwise item–item similarity for all available courses using cosine similarity. For each course, only the top k highest similarity values are kept, which correspond to its k nearest neighbors. We thus compare the structure of the best performing baseline ItemKNN and the best performing deep model LightGCN. By checking whether the identified structural differences contribute to ItemKNN's success, we gain a deeper understanding on the ItemKNN's strength as applied to course recommendation.

Importance of Pairwise Relation. Neighborhood information is incorporated in LightGCN through neighbor convolution. Each layer propagates information of neighbors from varied distances away: the first layer passes information only to those adjacent; i.e., for each course, only information of students who have taken the course is passed. With two layers, additional information transmitted includes that of students sharing interaction history overlap with the target student. Hence, at layer two, LightGCN learns pairwise relations, equivalent to ItemKNN's pairwise similarity.

ItemKNN's high accuracy suggests the possibility that pairwise relation captures the essential patterns for accurate recommendation. Hence, we hypothesize that in course recommendation, adding more layers over-smooths the learned representations. To validate this, we experiment with LightGCN of different layers. Table 3's results confirm that 2-layer LightGCN focusing on pairwise relation is most effective in learning the underlying relations and generating accurate recommendations, which differs from the e-commerce and social network setting studied previously [14]. This relates to the unique nature of course consumption. Students who took the same courses are likely from the same academic program; hence, they share common curriculum requirements and similar interaction patterns [16]. For a target student, what others from same program have taken previously largely indicates the courses (s)he will take next. By learning pairwise information, we are capturing these hidden relations among students and courses.

Focus on Close Neighbors. We also observe that ItemKNN only considers the top k neighbors whereas in LightGCN, information from all neighbors contributes to embedding learning and the final recommendation. It is possible that considering more than necessary neighbors brings in noise and lowers accuracy. To test this, we restrain LightGCN at the second layer to only perform neighbor propagation using the closest k neighbors where k is the number of neighbors used in KNN models. This revised LightGCN is called Constrain-Neighbor LightGCN (CN-LightGCN). Table 2b's results demonstrate the improvement in CN-LightGCN's performance relative to LightGCN, validating our claim. For a target student, (s)he may have taken the same course with many others. Among them, only a few are from the same academic program, whose course-taking histories are of key importance in suggesting what the target student would take next. Hence, focusing on these strong neighbors and discarding noise from weak ones is essential to course recommendation.

4.2 K-LightGCN: Accuracy Optimization For Course Recommendation

The close alignment between ItemKNN's focus on pairwise relations between close neighbors and the nature of course consumption partially explains the superiority of ItemKNN. Despite its overall higher accuracy, LightGCN provides accurate recommendations for some cases that ItemKNN fails. Hence, we propose K-LightGCN which combines ItemKNN and LightGCN and modifies it to reinforce the two features identified as key to accurate course recommendation.

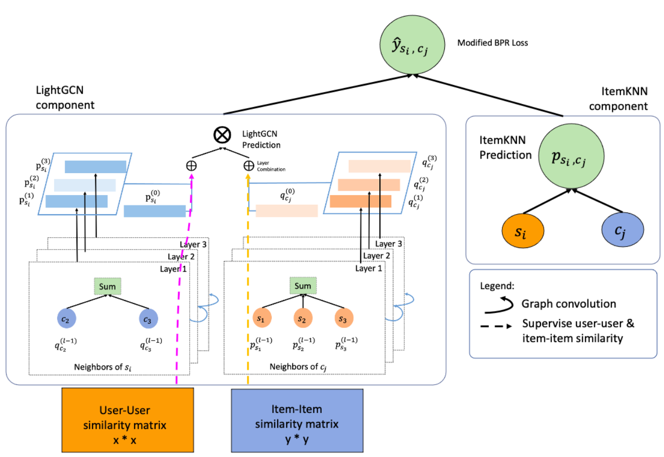

Fig. 1. An illustration of K-LightGCN model architecture. Pairwise similarity matrices computed by ItemKNN are used to supervise the embedding learning in LightGCN. Predictions generated by LightGCN and ItemKNN are combined as final recommendation.

4.2.1 *K-LightGCN Structure.*
K-LightGCN consists of paired LightGCN and ItemKNN components (Fig. 1). K-LightGCN enhances the strength of KNN by having the LightGCN component learn student and course embeddings with similarity

values close to that computed in KNN models. Thus, we introduce a new term to the original BPR loss. At each training batch, user–user cosine similarity matrix $w^{user}_{LightGCN}$ and item–item cosine similarity matrix $w^{item}_{LightGCN}$ are computed respectively using the learned student and course embedding in LightGCN. The squared differences between the similarity matrices computed by LightGCN and KNN are added to the loss. The resultant loss function in K-LightGCN is defined in Equation 2 where L_{BPR} is the original BPR loss defined in Equation 1. The final prediction is the μ-weighted sum of LightGCN and ItemKNN predictions, with different weights learned for each student.

$$L'_{BPR} = L_{BPR} + \mu\left(\sum_{i=1}^{y} \frac{1}{y} \sum_{j=1}^{y} (w^{item}_{LightGCN_{i,j}} - w^{item}_{i,j})^2 + \sum_{i=1}^{x} \frac{1}{x} \sum_{j=1}^{x} (w^{user}_{LightGCN_{i,j}} - w^{user}_{i,j})^2\right) \tag{2}$$

4.2.2 K-LightGCN Performance. The experiment results of K-LightGCN is shown in Table 2c. As K-LightGCN outperforms both ItemKNN and LightGCN, the effectiveness of the underlying design for course recommendation is validated. By learning embeddings with pairwise similarity close to that computed by KNN models, LightGCN is sharper in capturing the essential pairwise relations. For example, for the test case between Student ***s37*** and Course ***4f6***, ItemKNN recommends accurately but LightGCN misses the hit. Previously, Student ***s37*** has interacted with eight courses and identified by ItemKNN, the target Course ***4f6*** is highly similar to three of them, including Courses ***0d***, ***f5***, and ***z9***. Hence, ItemKNN successfully recommends the target course. However, LightGCN misses this relation as the cosine similarity of the LightGCN embedding between the four courses is close to zero. With ItemKNN supervising the embedding learning, K-LightGCN learns similar embeddings for the four courses identified, leading more accurate recommendation.

5 OBJECTIVE 2: PREFERENCE-POPULARITY MATCH

Accuracy solely learns student selections. However, with more information, students can adjust their plans to improve experience. There are multiple ways to quantify student experience. Our user study identified that students vary in preference in taking popular or niche courses. Hence, we measure student preference and course popularity to further optimize student experience by matching the two values, inspiring our Popularity-Match K-LightGCN model.

5.1 Preliminary Study

We propose a continuous measure for course popularity and student preference. Course popularity is computed as $pop = log(\frac{\#interactions}{\#semesters})$ which is the average number of enrollments when it is offered. Higher *pop* indicates on average, more students take the course, implying it is more highly desired in terms of course selection criteria like course content, instructor quality, and course difficulty [6]. Student preference is computed as $pref = \frac{\sum_{i \in I} pop(i)}{|I|}$, where I represents the set of courses (s)he has interacted with previously. Higher *pref* indicates the student tends to take more popular courses. Fig. 2 and Fig. 3 show the distribution for course popularity and student preference in our dataset, respectively. In both plots, wider sections indicate most courses (students) have *pop* (*pref*) within that range and those beyond the 75th percentile are in orange. For courses, a limited few have *pop* above 4 but as they dominate the enrollment records, they are recommended more, even when the target student prefers niche courses. As required by IHLs, a typical student takes several extremely popular core courses and some relatively niche electives, resulting in most with *pref* above 4.

5.1.1 Preference-Popularity Match Definition and Evaluation. The distribution of student preference demonstrates large variation: some prefer popular courses whereas others enjoy niche ones. Hence, to provide quality user experience for all, our goal is to expose students with low *pref* to niche courses as well as recommending popular courses that better fit

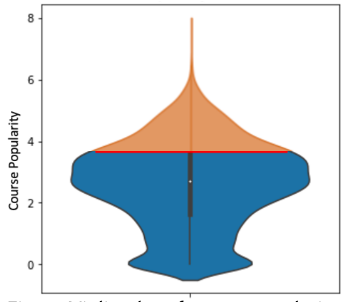

Fig. 2. Violin plot of course popularity.

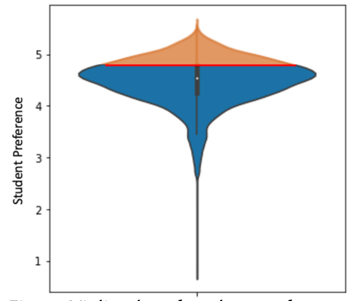

Fig. 3. Violin plot of student preference.

the interests of students with high *pref*; i.e., **preference-popularity matching**. In contrast, **preference-popularity mismatch** is a loss: a mismatch between student preference and popularity of recommended courses. This occurs when popular (niche) courses are recommended to students with low (high) *pref*. With the former, students are not exposed to the niche courses they are looking for; with the latter, students are likely recommended with courses they already know of or those that only loosely fit their preference. Such recommendations do not value add to students' decision making.

We propose Preference-Popularity Mismatch@K (PP-Mismatch@K) loss to measure preference-popularity match performance, defined as $\frac{1}{z}\sum_{r_{s_i}\in R}\frac{1}{|K|}\sum_{j\in r_{s_i}}\left|pref(s_i)-pop(c_j)\right|$. For each test case, it takes in top $K = 10$ recommendation list r_{s_i}, computing the absolute difference between student's preference and recommendations' average popularity. PP-Mismatch@10 is the average mismatch of all z test cases and higher values indicate poorer performance.

5.2 PM K-LightGCN: Multi-Objective Course Recommender

Matching popularity of recommended courses with student preference is key to meeting students' varied needs and providing quality user experience. We propose Popularity-Match K-LightGCN (PM K-LightGCN) to explicitly capture this criterion by aligning model recommendations with students' preference on top of accuracy optimization. PM K-LightGCN takes the top 50 recommendations by K-LightGCN and selects 10 courses that minimize the mismatch defined as $\frac{1}{|K|}\sum_{j\in r_{s_i}}\left|pref(s_i)-pop(c_j)\right|$ by choosing those with popularity closest to the target student's preference.

We do not perform another round of optimization using all available courses to achieve the second objective as we believe accuracy is the primary criterion to be fulfilled. After K-LightGCN generates accurate recommendations, we select the 10 courses that fit the second objective as the final recommendations.

5.2.1 Experiment Results. Recommenders' performance on both objectives indicate that recommendations by PM K-LightGCN better match students' preference (a mismatch reduction of 17%; Table 4). The only difference between PM K-LightGCN and K-LightGCN is the filtering component that selects the $K = 10$ courses out of the top recommendations by K-LightGCN. PM K-LightGCN successfully mitigates preference-popularity mismatch, but K-LightGCN generates recommendations with the highest mismatch. To optimize solely for accuracy, K-LightGCN picks up the pattern that most students enjoy popular courses as indicated by the wider sections between $pref = 4 - 5$ in the violin plot for student preference (Fig. 3) and recommends popular courses more often to cater to the majority.

Model	PP-Mismatch@10	HR@10	NDCG@10
ItemKNN	1.050	0.7762	0.3337
UserKNN	1.071	0.7294	0.2521
LightGCN	1.077	0.7008	0.2542
K-LightGCN	**1.109**	**0.7905**	**0.3346**
PM K-LightGCN	**0.920 (-17.0%)†**	**0.7570 (-4.23%)***	**0.3000 (-10.3%)***

Table 4. Performance in Two Objectives (relative to K-LightGCN). †: For PP-Mismatch, lower values are desirable and negative percentage change indicates better performance. *: For HR and NDCG, negative percentage change indicates worse performance.

5.2.2 Case Studies. We select three test cases, listed in Table 5 to examine preference-popularity mismatch mitigation in PM K-LightGCN. For simplicity, we list the first five courses in the top 10 recommendations. In the first case study, both PM K-LightGCN and K-LightGCN recommend accurately. In addition to accuracy optimization, with PM K-LightGCN filtering the outputs of K-LightGCN to minimize preference-popularity mismatch, the popularity of its recommendations all align with the student's preference. Hence, the student is likely to be inspired by PM K-LightGCN's recommendations as they additionally have the relevant characteristics considered by the student.

In the second case study, the connection between target course and the student's preference is missed by K-LightGCN, which is corrected with the additional filtering process in PM K-LightGCN. However, both models fail in the third case study. Despite this, PM K-LightGCN recommends several niche courses that are of similar popularity to the student's preference. This offers the student possibilities to make a more informed decision.

Case Study	Student Preference	Target Course Popularity	Popularity of Top 5 Courses Recommended									
			PM K-LightGCN					K-LightGCN				
			#1	#2	#3	#4	#5	#1	#2	#3	#4	#5
1	4.381	3.883	3.946	**3.883**	3.876	3.588	3.419	3.946	3.377	**3.883**	3.411	2.508
2	3.343	3.434	3.332	3.377	**3.434**	3.583	3.588	3.876	1.872	3.030	4.200	2.085
3	1.749	1.398	1.621	1.477	1.407	1.376	1.301	3.030	3.168	1.477	2.347	2.684

Table 5. Case Study of PM K-LightGCN (accurate recommendations are bolded, #k represents the k_{th} ranked course)

6 CONCLUSION

We propose a multi-objective course recommender that optimizes for accuracy and preference-popularity match. For accuracy optimization, we propose K-LightGCN, which combines KNN models and LightGCN and uses the former to supervise embedding learning in LightGCN. For preference-popularity match, we post-filter top recommendations by K-LightGCN to select $K = 10$ courses of popularity closest to the target student's preference. Typical IHL curriculum requires students to read core courses in junior years and electives to be chosen from the available ones in senior years. Hence, as students mature, they will benefit more from PM K-LightGCN which matches their personalized preference. In our study, the PP-Mismatch@10 improvement in PM K-LightGCN mainly comes from recommendation in later years.

Course recommendation remains a developing area. This project sheds lights on the importance of neighborhood information and multi-objective recommender development. Future studies could improve on PM K-LightGCN with alternative student preference measure. Currently, it is computed as the average popularity of courses taken previously which neglects the popularity variations in the enrollment history. Such limitation will result in poor performance when the student takes extremely popular and niche courses together. It could be mediated with next-basket recommendation [27] which matches the popularity distribution of the recommended basket with that of courses taken previously.

Though PM K-LightGCN's post-filtering step does not involve intricate machine learning, it is sufficiently lightweight and saves space and time when deployed. Besides, this lightweight design is both effective in preference-popularity mismatch mitigation and balancing both objectives. Although the pursuit of preference-popularity match lowers accuracy, compared to K-LightGCN, PM K-LightGCN achieves a 17% improvement in preference-popularity match with the sacrifice of only 4% in accuracy. As enrollment records may not optimize student experience due to their limited knowledge [25], the fall in accuracy can improve user satisfaction. However, due to the inherent constraints of our dataset not including actual course information, it is infeasible to conduct a user study for better model evaluation from user's perspective. Besides, the design of PM K-LightGCN allows scaling to additional criteria. When other needs like the promotion of certain courses to better prepare students for the workplace arise, they can be formulated as additional loss functions or post-filtering criteria to be added to the recommender in the same lightweight and efficient manner.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 42–46. https://doi.org/10.1145/3109859.3109912

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking. https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS19/paper/view/18199/17319

[3] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, Bamshad Mobasher, and Edward Malthouse. 2021. User-centered Evaluation of Popularity Bias in Recommender Systems. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. 119–129.

[4] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 385–394.

[5] Rel Guzman Apaza, Elizabeth Veronica Vera Cervantes, Laura Vanessa Cruz Quispe, and José Eduardo Ochoa Luna. 2014. Online Courses Recommendation based on LDA. In *SIMBig*.

[6] Elisha Babad. 2001. Students' Course Selection: Differential Considerations for First and Last Course. *Research in Higher Education* 42, 4 (2001), 469–492. http://www.jstor.org/stable/40196437

[7] Alican Bozyiğit, Fatma Bozyiğit, Deniz Kilinç, and Efendi Nasiboğlu. 2018. Collaborative Filtering based Course Recommender using OWA operators. In *2018 International Symposium on Computers in Education (SIIE)*. 1–5. https://doi.org/10.1109/SIIE.2018.8586681

[8] Hui-Tzu Chang, Chia-Yu Lin, Li-Chun Wang, and Fang-Ching Tseng. 2022. How Students can Effectively Choose the Right Courses: Building a Recommendation System to Assist Students in Choosing Courses Adaptively. *Educational Technology Society* 25, 1 (2022), 61–74. https://www.jstor.org/stable/48647030

[9] Pei-Chann Chang, Cheng-Hui Lin, and Meng-Hui Chen. 2016. A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms* 9, 3 (2016), 47.

[10] Glenn M Davis, Abdallah A AbuHashem, David Lang, and Mitchell L Stevens. 2020. Identifying Preparatory Courses that Predict Student Success in Quantitative Subjects. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*. 337–340.

[11] Mukund Deshpande and George Karypis. 2004. Item-Based Top-<i>N</i> Recommendation Algorithms. *ACM Trans. Inf. Syst.* 22, 1 (jan 2004), 143–177. https://doi.org/10.1145/963770.963776

[12] Asmaa Elbadrawy and George Karypis. 2016. Domain-Aware Grade Prediction and Top-n Course Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems* (Boston, Massachusetts, USA) *(RecSys '16)*. Association for Computing Machinery, New York, NY, USA, 183–190. https://doi.org/10.1145/2959100.2959133

[13] Deepani B. Guruge, Rajan Kadel, and Sharly J. Halder. 2021. The State of the Art in Methodologies of Course Recommender Systems,ÄîA Review of Recent Research. *Data* 6, 2 (2021). https://doi.org/10.3390/data6020018

[14] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. 639–648. https://doi.org/10.1145/3397271.3401063

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[16] Mary R. Hedges, Gail A. Pacheco, and Don J. Webber. 2014. What determines students,Äô choices of elective modules? *International Review of Economics Education* 17 (2014), 39–54. https://doi.org/10.1016/j.iree.2014.05.003

[17] Chung-Yi Huang, Rung-Ching Chen, and Long-Sheng Chen. 2013. Course-recommendation system based on ontology. In *2013 International Conference on Machine Learning and Cybernetics*, Vol. 3. IEEE, 1168–1173.

[18] Weijie Jiang, Zachary A Pardos, and Qiang Wei. 2019. Goal-based course recommendation. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. 36–45.

[19] Santosh Kabbur, Xia Ning, and George Karypis. 2013. 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*. 659–667.

[20] Giannis Karamanolakis, Kevin Raji Cherian, Ananth Ravi Narayan, Jie Yuan, Da Tang, and Tony Jebara. 2018. Item recommendation with variational autoencoders and heterogeneous priors. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*. 10–14.

[21] Yehuda Koren, Steffen Rendle, and Robert Bell. 2022. *Advances in Collaborative Filtering*. Springer US, New York, NY, 91–142. https://doi.org/10.1007/978-1-0716-2197-4_3

[22] Y.-M Lee and Sung-Hsien Sun. 2010. The relationship between autonomous motivation of course-taking and learning engagement on college students. 55 (03 2010), 155–182.

[23] Bhaskar Mondal, Om Patra, Sanket Mishra, and Priyadarshan Patra. 2020. A course recommendation system based on grades. In *2020 international conference on computer science, engineering and applications (ICCSEA)*. IEEE, 1–5.

[24] Raphaël Morsomme and Sofia Vazquez Alferez. 2019. Content-Based Course Recommender System for Liberal Arts Education. *International educational data mining society* (2019).

[25] Zachary A. Pardos and Weijie Jiang. 2020. Designing for Serendipity in a University Course Recommendation System. In *Proceedings of the Tenth International Conference on Learning Analytics amp; Knowledge* (Frankfurt, Germany) *(LAK '20)*. Association for Computing Machinery, New York, NY, USA, 350–359. https://doi.org/10.1145/3375462.3375524

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[27] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-Basket Recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) *(WWW '10)*. Association for Computing Machinery, New York, NY, USA, 811–820. https://doi.org/10.1145/1772690.1772773

[28] Abdon Rivera, Mariela Tapia-Leon, and Sergio Luján-Mora. 2018. *Recommendation Systems in Education: A Systematic Mapping Study.* 937–947. https://doi.org/10.1007/978-3-319-73450-7_89

[29] S. Venugopalan, M. V. Srinath, and Paul Rodrigues. 2016. Recommender System for E-Learning through Content and Profile Based Approach. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies* (Udaipur, India) *(ICTCS '16)*. Association for Computing Machinery, New York, NY, USA, Article 45, 5 pages. https://doi.org/10.1145/2905055.2905103

[30] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2020. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. *arXiv preprint arXiv:2011.01731* (2020).

[31] Ziwei Zhu, Yun He, Yin Zhang, and James Caverlee. 2020. Unbiased implicit recommendation and propensity estimation via combinational joint learning. In *Fourteenth ACM Conference on Recommender Systems*. 551–556.

[32] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (Virtual Event, Israel) *(WSDM '21)*. Association for Computing Machinery, New York, NY, USA, 85–93. https://doi.org/10.1145/3437963.3441820