AMinerGNN: Heterogeneous Graph Neural Network for Paper Click-through Rate Prediction with Fusion Query

Zepeng Huai†‡∗
zepenghuai6@gmail.com
†School of Artificial Intelligence, UCAS
‡CASIA
Beijing, China
Yifan Zhu
zhuyifan@tsinghua.edu.cn
Tsinghua University
Beijing, China
Zhe Wang
zhe.wangz@bytedance.com
ByteDance Inc.
Mountain View, CA, United States
Peng Zhang
peng.zhang@aminer.cn
Zhipu AI Lab
Beijing, China

ABSTRACT
Paper recommendation with user-generated keyword is to suggest papers that simultaneously meet user’s interests and are relevant to the input keyword. This is a recommendation task with two queries, a.k.a. user ID and keyword. However, existing methods focus on recommendation according to one query, a.k.a. user ID, and are not applicable to solving this problem. In this paper, we propose a novel click-through rate (CTR) prediction model with heterogeneous graph neural network, called AMinerGNN, to recommend papers with two queries. Specifically, AMinerGNN constructs a heterogeneous graph to project user, paper, and keyword into the same embedding space by graph representation learning. To process two queries, a novel query attentive fusion layer is designed to recognize their importances dynamically and then fuse them as one query to build a unified and end-to-end recommender system. Experimental results on our proposed dataset and online A/B tests prove the superiority of AMinerGNN.

CCS CONCEPTS
• Information systems → Recommender systems.

KEYWORDS
Click-Through Rate Prediction, Graph Neural Network, Recommender System

ACM Reference Format:
Zepeng Huai†‡∗, Zhe Wang, Yifan Zhu, and Peng Zhang. 2022. AMinerGNN: Heterogeneous Graph Neural Network for Paper Click-through Rate Prediction with Fusion Query. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM ’22), October 17–21, 2022, Atlanta, GA, USA. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557544

1 INTRODUCTION
AMiner1 is an academic information retrieval website, which can provide paper recommendation to researchers via Keyword-RS as shown in Figure 1. When simultaneously given the user ID and a user-generated keyword (usually a research direction, such as graph neural network or pretraining), Keyword-RS suggests papers that not only are relevant to this keyword but also satisfy user potential interests.

According to the relation between the user-generated keyword and the research direction of the user based on his interacted papers, there are three segmentation scenarios in Keyword-RS: (1) both of them refer to the same research field (S1); (2) there exists the potential of interdisciplinary research between them (S2); (3) they are independent and have no relevance (S3). We collect some user feedbacks and have an important finding: the purpose that researchers use AMiner and the importance of these two queries vary according to scenarios, which are as follows: (1) In $S_1$, users aim to make a deeper literature review about this research field. At this time, Keyword-RS should utilize his interests to recommend and user ID is more important than keyword since user ID contains the personal information. (2) In

---

1Corresponding Author.
S2, users intend to know how this approach (user-generated keyword) can be integrated into his past research, like how contrastive learning can be used for recommendation. Two queries are both useful in this situation. (3) In S3, users prefer to improve the general understanding of this new field. Therefore, it is reasonable to recommend papers without personal preferences and show some typical or hot papers clicked by most researchers, which means the keyword is more important than user ID. To satisfy the above recommendation requirements, there are two problems: (1) how to model the relation between the user-generated keyword and his past research direction; (2) how to build a unified and end-to-end recommender system applicable in all scenarios.

To address the foregoing problems, we propose an end-to-end heterogeneous graph neural network for paper click-through rate (CTR) prediction, called AMinerGNN. It consists of two components: (1) Graph embedding layer. To model the semantic correlation, we construct a heterogeneous academic graph, which projects all users, papers and keywords into the same embedding space via graph based representation learning to further learn the relation between user-generated keyword and his past research direction. (2) Query attentive fusion layer: We employ an attention unit to automatically mine user scenario-specific purpose and dynamically adjust the importance of two queries, which are further combined as one query to build a unified recommender system.

2 RELATED WORKS

We briefly summarize two related subareas. (1) Click-through rate (CTR). Feature interaction [8, 10, 17, 18] is an early method to build a predictive model like FM [18] and DeepFM [8]. Recently, modeling user behaviors [3, 15, 16, 23, 24] is becoming an essential topic since they contain crucial patterns of user interests like DIN [24], DIEN [23], UBR [16] and SURGE [3]. However, their inputs are userID and target item, which can not directly resolve the CTR problem with fusion query. (2) Product search. This is a close solution by regarding the input keyword as the search query. A basic category of methods [4–6, 11] is to associate the free-form user queries with structured data stored in relational databases like MAM [5]. Recent works aim to study the user’s personal and temporal preferences [1, 2, 9, 13, 22] like HEM [1], ALSTP [9] and MGDSFR [13]. However, all of them are under-explored in the task where the search query and personal preferences have non-uniform relations.

3 PRELIMINARIES

3.1 Problem Setup

Paper CTR with two queries. Given a set < \mathcal{U}, \mathcal{K}, \mathcal{P} >, where \mathcal{U} = \{u_1, \ldots, u_M\} denotes M users, \mathcal{K} = \{k_1, \ldots, k_T\} denotes T keywords and \mathcal{P} = \{p_1, \ldots, p_N\} denotes N papers. In this problem, user and keyword are both regarded as queries, leading to three kinds of interaction data, which are user-papers, keyword-paper and user&keyword-paper interactions. The above three interactions are represented as \mathcal{Y} = \{y_{u_k}, y_{k_p}, y_{u_kp} | u \in \mathcal{U}, k \in \mathcal{K}, p \in \mathcal{P}\}, respectively. We also utilize two widely used information [12, 14, 21] to improve CTR performance: (1) query behaviors including two aspects: user behaviors \mathcal{H}_u = \{u | y_{u_p} = 1\} and keyword behaviors \mathcal{H}_k = \{p | y_{k_p} = 1\}, (2) item features \mathcal{F}_p = \{f_{q1}^p, \ldots, f_{qQ}^p\}, \ldots, \}.

\hat{y}_{ujkp}^\theta denotes q-th feature of paper p like citation number or publication year. When user u inputs a keyword k, the Keyword-RS is to predict whether u will click p as \hat{y}_{ujkp}^\theta = f(\mathcal{H}_u, \mathcal{H}_k, \mathcal{F}_p).

3.2 Graph Construction

We construct a heterogeneous graph \mathcal{G} to hold the semantic relatedness between different entities in Keyword-RS. Specifically, there are three types of nodes: user, paper, and keyword. Then three edge types are built from raw interactions between nodes as follows:

• user-paper. User u clicks or searches paper p.
• user-keyword: User u adds keyword k as shown in figure 1.
• paper-keyword: The title of paper p contains the keyword k.

Figure 2(a) shows an example of the above three situations.

4 METHODOLOGY

4.1 Motivation

To tackle the two problems mentioned in Section 1, a three-step solution is designed as follows: (1) representing user, paper, and keyword in the same embedding space \rightarrow (2) enhancing representations via raw interactions between entities \rightarrow (3) feeding user and keyword representations into a unified CTR model.

Note that GNN is of great significance on different entity sides: (1) On the paper side. Paper representations, which keywords are integrated into, can contain the information of the research direction of the paper. (2) On the user side. The interacted papers of a user represent his past research direction and preferences in viewing papers. What’s more, the keywords user adds directly reflect his concerned research field. Therefore, it is pivotal to enhance user representations with the above two entities. (3) On the keyword side. If two keywords co-occur in one paper, it shows the potential of interdisciplinary research between them. Thus, mutually enhanced representations of a pair of keywords can bring them closer in latent semantic space.

4.2 Graph embedding Layer

In each layer, we employ a widely used two-step scheme [7, 19, 20] to aggregate neighbors: (1) same relation-aware neighbors aggregation via average pooling; (2) relations combination via sum pooling. More formally, in the l-th layer, node representation is updated as

\[ e_i^{(l)} = \sum_{t \in O_i} \frac{1}{|N_t^i|} \sum_{j \in N_t^i} e_j^{(l-1)} \] (1)

where \[ e_i^{(l)} \] denotes the embedding of node i in the l-th layer and \[ e_i^{(0)} \] is randomly initialized. Here i can be one of the user, paper or keyword node. \[ N_t^i \] denotes the t-type neighbors for node i and \( t \in O_i \).

After performing L layers, we adopt sum pooling to integrate multi-hop information as

\[ e_i = \sum_{l=0}^{L} e_i^{(l)} \] (2)

where superscript * indicates that the representation is enhanced by GNN.
4.3 Query Attentive Fusion Layer

With all representations in the same embedding space, we calculate the correlation between a user and a keyword, denoted as $y_{uk}$, using distance correlation as

$$y_{uk} = d \text{Cor}(e_u^e, e_k^e) = \frac{d \text{Cov}(e_u^e, e_k^e)}{\sqrt{d \text{Cov}(e_u^e, e_u^e) \cdot d \text{Cov}(e_k^e, e_k^e)}}$$

where $d \text{Cov}(\cdot)$ is the distance covariance of two representations.

As aforementioned in Section 1, the relation between two queries is non-uniform for all scenarios. More specifically, the importance of user ID is in proportion to the correlation between two queries. Therefore, an intuitive idea is regarding correlation $y_{uk}$ as the attention weight to form a unified and fused query as

$$e_q^* = y_{uk} e_k^e + (1 - y_{uk}) e_k^e$$

Table 1: Statistics of the AMiner_KRS dataset. 'u', 'k' and 'p' denote user, keyword and paper, respectively.

| Graph $G$ | node | # users | # keywords | # papers |
|-----------|------|---------|------------|---------|
|           | edge | 67,806  | 398,001    | 200,837 |
| Interactions $Y$ | # yu-p | 568,424 | 5,030 | 2,781 |

4.4 Model Optimization

We use Logloss to capture divergence between two probability distributions as

$$L = - \sum_{(u,k,p) \in S} (y_{uk-p} \log \hat{y}_{uk-p} + (1 - y_{uk-p}) \log (1 - \hat{y}_{uk-p}))$$

5 EXPERIMENTS

5.1 Settings

5.1.1 Dataset Description. We propose a novel dataset, called AMiner_KRS dataset. We collect the interaction data in the past three months, from Feb. 1, 2022, to May 1, 2022. Finally, AMiner_KRS consists of 67806 users, 398001 papers, and 208837 keywords. Table 1 lists the statistics of the AMiner_KRS dataset.

5.1.2 Baselines & Metrics. To demonstrate the effectiveness, we compare AMinerGNN with two kinds of methods: (1) CTR models including DeepFM [8], DIN [24], DIEN [23], UBR [16], (2) Product search methods including HEM [1], ALSTP [9], MGDSPR [13]. We adopt the commonly used AUC and Log Loss to measure the performance.

5.1.3 Experimental Settings. The core contribution of AMinerGNN is generalizing traditional CTR models to the task with fusion query using two key designs: (1) GNN embedding layer and (2) query fusion layer. Therefore, our experiments aim to prove the effectiveness of these two parts. We compare each model with the following three variants: (1) using graph embedding layer, (2) using query attentive fusion layer, and (3) both adopting these two layers. We
As shown in Table 2, we have the following observations:

- For CTR baselines, there are similar performances and we use DIEN as an example to illustrate our findings. AMinerGNN and DIEn contribute up to 5.02% AUC and 4.11% AUC promotion compared to DIEn and DIEN, which proves the effectiveness of the query fusion layer. It has a similar result for GNN embedding layer. Moreover, the improvement brought by the GNN layer is more significant than that by the query fusion layer. The reason is that the GNN layer is the prerequisite to accurately modeling the semantic relation between two queries using distance covariance.

- For product search baselines, all of them underperform than DIEn and UBR. The main reason is that they insufficiently recognize the pattern that the user attention towards the input keyword and personalization are both important in all scenarios. A detailed comparison is reported in Section 5.3.

- Note that we adopt DIEN as our base CTR model rather than UBR. UBRGf slightly outperforms DIEnGf. However, the online service time of UBRGf is much bigger than that of DIEnGf, because UBR uses the search engine approach, leading to an increase of the online inferring time. We use Locust2 to simulate 200 simultaneous users and average the response time of CTR-based models3 shown in Table 3. Finally, DIEnGf is deployed as a trade-off between performance and speed.

5.3 Scenario Test

In this section, we investigate whether AMinerGNN adaptively fits for different scenarios. We calculate \( y_u k \) via equation 3 for each testing batch. \( y_u k \in [0.5, 1], [0.1, 0.5], [0, 0.1] \) indicate S1, S2, and S3, respectively. The results are shown in Figure 3 and the major findings are as below:

- CTR-based methods perform stably in three scenarios, which proves that two key components of AMinerGNN can generalize traditional CTR models to a unified recommender system applicable in all scenarios.

- Product search-based models perform better in S2, while the performances in the other two scenarios sharply decrease. The reason is the input keyword and personalization are both important in S2, which can realize their potentials best. While in the other two scenarios, only one query dominates influence, which is less applicable for product search-based models.

5.4 Online A/B Test

We report the online A/B tests of AMinerGNN for a duration of 14 days, from May 17, 2022, to May 30, 2022. About 30 thousand users participated in this A/B test. Two kinds of click-through rate (CTR) on the homepage of AMiner are chosen as metrics. ‘UV-CTR’ indicates whether a user clicks the recommended paper, irrespective of the number of clicks. ‘PV-CTR’ is the ratio of clicked papers to all exposed papers. Table 4 shows AMinerGNN achieves 9.6% and 3.7% UV-CTR improvement relative to DIEN and MGDSRP, proving AMinerGNN works better in this recommendation task.

6 CONCLUSION

In this paper, we study the recommendation task with two queries. The proposed AMinerGNN can automatically mine user scenario-specific purposes, dynamically recognize the importance of two queries, and attentively fuse two queries to construct a unified and end-to-end recommender system. Compared with the traditional CTR model given one query and product search methods, AMinerGNN performs better and achieves a higher CTR on the AMiner homepage.
REFERENCES

[1] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 645–654.

[2] Keping Bi, Qingyao Ai, and W Bruce Croft. 2020. A transformer-based embedding model for personalized product search. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 1521–1524.

[3] Jianxin Chang, Chen Gao, Yu Zheng, Yopun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In SIGIR. 378–387.

[4] Huizhong Duan and ChengXiang Zhai. 2013. Mining coordinated intent representation for entity search and recommendation. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. 333–342.

[5] Huizhong Duan, ChengXiang Zhai, Jinjing Cheng, and Abhishek Gattani. 2013. A probabilistic mixture model for mining and analyzing product search log. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management. 2179–2188.

[6] Huizhong Duan, ChengXiang Zhai, Jinjing Cheng, and Abhishek Gattani. 2013. Supporting keyword search in product database: a probabilistic approach. Proceedings of the VLDB Endowment 6, 14 (2013), 1786–1797.

[7] Xinyu Fu, Jiarui Jin, Ziqiao Meng, and Irwin King. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In WWW. 2331–2341.

[8] Huiyong Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, Xiuqiang He, and Zhenhua Dong. 2018. Deepfm: An end-to-end wide & deep learning framework for CTR prediction. arXiv preprint arXiv:1804.04950 (2018).

[9] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2021. Attentive long short-term preference modeling for personalized product search. ACM Transactions on Information Systems (TOIS) 37, 2 (2019), 1–27.

[10] Yuchin Juan, Yang Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In Proceedings of the 10th ACM conference on recommender systems. 43–50.

[11] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. 2017. On application of learning to rank for e-commerce search. In Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. 475–484.

[12] Pan Li. 2021. Leveraging Multi-Faceted User Preferences for Improving Click-Through Rate Predictions. In RecSys. 864–868.

[13] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based Product Retrieval in Taobao Search. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 3181–3189.

[14] Bin Liu, Ruiming Tang, Yingzhai Chen, Jinkai Yu, Huiyong Guo, and Yuzhou Zhang. 2019. Feature generation by convolutional neural network for click-through rate prediction. In WWW. 1119–1129.

[15] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Leijian Ren, Ying Fan, Xiaoliang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In CIKM. 2685–2692.

[16] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User behavior retrieval for click-through rate prediction. In SIGIR. 2347–2356.

[17] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE. 1149–1154.

[18] Steffen Rendle. 2010. Factorization machines. In 2010 IEEE International conference on data mining. IEEE. 995–1000.

[19] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In WWW. 2022–2032.

[20] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In KDD. 793–803.

[21] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep learning for click-through rate estimation. IJCAI (2021).

[22] Yuan Zhang, Dong Wang, and Yan Zhang. 2019. Neural IR meets graph embedding: A ranking model for product search. In The World Wide Web Conference. 2390–2400.

[23] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoliang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In AAAI. Vol. 33. 5941–5948.

[24] Guorui Zhou, Xiaoliang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yinghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1059–1068.