INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 18, Issue: 4, Month: August, Year: 2023
Article Number: 5173, https://doi.org/10.15837/ijccc.2023.4.5173

A Project Recommender Based on Customized Graph Neural Networks in Online Labor Markets

Y. Ma, Z. Ma, Y. Li, H. Gao, Y. Xue

Yixuan Ma*
School of Software Engineering
Beijing Jiaotong University, China
*Corresponding author: mayx@bjtu.edu

Zeyao Ma
School of Computer Science
Beijing University of Posts and Telecommunications, China
mazeyao@bupt.edu.cn

Yankai Li
School of Information Engineering
China University of Geosciences (Beijing), China
1006192217@cugb.edu.cn

Haoyu Gao
School of Software Engineering
Beijing Jiaotong University, China
19301113@cugb.edu.cn

Yukai Xue
School of Software Engineering
Beijing Jiaotong University, China
19301104@bjtu.edu.cn

Abstract

Project recommendation is a crucial task in the online labor markets, where freelancers seek projects that match their skills and preferences. The challenge in project recommendation is to effectively capture the complex relationships between projects, skills, and freelancers in a high-dimensional space. Traditional recommender systems, such as matrix factorization, often suffer from sparsity, making it difficult to accurately recommend projects to freelancers. To address the challenges, this research proposes a customized graph neural network approach, called GCNRec, specifically using Light Graph Convolution Network (LightGCN) to effectively capture both explicit and implicit relationships between projects, skills, and freelancers, and make personalized project recommendations. The experiment results on real-world dataset demonstrate the effectiveness of the proposed method in solving the challenges in project recommendation in the online labor market.

Keywords: project recommendation, online labor markets, graph neural networks, LightGCN.

1 Introduction

The online labor market is a thriving platform that links clients seeking services with freelancers possessing the necessary skills and knowledge. This industry has experienced substantial growth in recent years and is favored for its convenience and flexibility, enabling individuals to monetize their abilities. The largest freelancing platforms, including Upwork and Freelancer, have attracted millions of clients and freelancers from across the globe. As of 2022, the global online freelance market had surpassed $1.5 trillion and is projected to continue growing in the future. Nevertheless, freelancers face the challenge of identifying suitable projects within a diverse and expansive market. To address this problem, project recommendation systems have been designed to furnish personalized and effective suggestions based on freelancers' abilities and preferences.

Recommender systems have been developed to assist individuals in making personalized and effective decisions in various fields, including e-commerce[13], entertainment[4], and news[18]. The objective of a recommender system is to predict the preferences of a user based on their past behaviors and preferences, as well as the behaviors and preferences of similar users. In the case of project recommendation in the online labor market, the challenge lies in accurately capturing the complex relationships between projects, skills, and freelancers in a high-dimensional space. The uniqueness of project recommendations compared to other recommendation systems lies in the need to consider multiple factors such as the skill required for a project, the preferences of the freelancer, and the project budget and deadline. These factors can interact in complex ways, making it difficult to make accurate project recommendations using traditional recommendation methods, such as matrix factorization.

To address these challenges, recent research has explored the use of graph neural networks for project recommendation. Graph neural networks are a powerful tool for learning from graph-structured data[23], and have been successfully applied to a wide range of tasks, including node classification[21], link prediction[27], and graph generation[26]. By treating the project recommendation problem as a graph-based learning task, graph neural networks can effectively capture the relationships between projects, skills, and freelancers, and make personalized project recommendations. Light Graph Convolution Network (LightGCN) [8], a type of graph convolutional network is designed to handle large-scale graphs with a lightweight architecture, making it well-suited for the online labor market, where the number of projects, skills, and freelancers can be very large. The model constructs the graph structure for the project recommendation task, where the nodes in the graph represent projects, skills, and freelancers, and the edges represent the relationships between these entities.

However, traditional LightGCN only focuses on explicit connections of users, such as defined skills required for the project, which can limit the performance of recommendations. This is because explicit connections may not capture all of the relationships between entities, such as indirect or latent relationships. For example, two freelancers who lack direct connections may still have comparable project preferences. Furthermore, the traditional LightGCN technique is not adept at capturing long-distant dependencies in the graph structure, which are crucial for comprehending the intricate relationships between workers and projects.

In this paper, we propose a novel project recommendation framework based on customized graph neural networks, called GCNRec, in the online labor market. Specifically, we customized LightGCN by incorporating both explicit and implicit interaction between workers and projects and capturing long-distance dependencies between entities. We evaluate the proposed method on a real-world dataset and show that it outperforms existing recommendation methods in terms of accuracy and efficiency.

The rest of the paper is organized as follows. Section 2 provides a background on project recommendation and graph neural networks. Section 3 describes the proposed method in detail. Section 4 presents the experimental results and evaluation. Section 5 concludes the paper and discusses future work.

2 Related Work

Project recommendation in the online labor markets has been a long-standing challenge in the field of recommender systems. The traditional approach to project recommendation has been collaborative filtering (CF), which was introduced by Goldberg in 1992 through the creation of Tapestry [7]. This approach relied on detecting similarities in user behaviors and item preferences. However, this type of recommendation system has too many requirements for users and is inconvenient in certain scenarios such as inputting query sentences. Resnick proposed the first automatic recommender system, GroupLens, in 1994 which utilized rating methods (from 1 to 5) to measure user preferences [12]. This traditional approach of CF suffers from the "cold start" problem, where the algorithm is limited by the user-item data and thus results in lower-quality recommendations [5].

Recent advances in graph neural networks (GNNs) have provided solutions to overcome the limitations of traditional CF. Graph Convolutional Networks (GCNs) have been proposed to model the interaction between users and projects as a binary graph, with the edges representing the interactions and the weights of the relationships being expressed through edge properties [15]. GCNs can extract feature vectors of different dimensions for different modeling purposes and has higher data access efficiency than adjacency matrices.

Several variations of GCN have been proposed, such as LightGCN[8], which is a simplified version that reduces training difficulty, and Knowledge Graph Attention Networks (KGAT)[16], which links items with their attributes. Graph Heterogeneous Collaborative Filtering (GHCF)[2] adds both node and relation representations for multi-relational prediction, and performs well in dealing with the cold-start problem. Wu et al.[22] proposed a model that considers the impact of the user on user iteration to address the data sparsity and cold start issues in traditional algorithms. Sheu and Li et al. proposed the Context-Aware Graph Embedding (CAGE) [14] framework to enrich the semantics of news articles and refine their embeddings through GCN. Khatua et al.[9] used GCN-based Graph Convolution Network (GUN) to match recruiters and job seekers through Twitter data.

In the field of project recommendation, our work is inspired by the LightGCN approach. In the online labor market, the relationship between users, items, and skills is not reciprocal. We introduce an attention mechanism to focus on projects that match user skills to effectively improve the accuracy of recommendations. This is based on the idea that a user may pay more attention to projects that match their skills, not just because they are interested, but because it increases their completion rate. This work differs from previous efforts in the field as it considers the influence of skills on project recommendation, while previous methods have focused on social recommendation and the influence of users on users.

3 Methodology

In this section, we provide an overview of LightGCN architecture, including the embedding layer, light graph convolution, and layer combination. Then, we describe our proposed recommendation framework GCNRec based on LightGCN and present the construction of graphs that include both explicit and implicit interactions. Thereafter, we explain the model training process and the regularization approach included in our framework.

3.1 LightGCN Architecture

Light Graph Convolution Network (LightGCN) is a graph convolutional network architecture for graph-based recommendation systems[8]. It is designed to handle large-scale graph data, such as the data found in online labor markets, by using a scalable and efficient graph convolution operation.

The LightGCN consists of multiple graph convolutional layers that aggregate information from the graph's neighbors to update the representations of the nodes (e.g., workers and projects). In each layer, the representation of each node is updated by taking a weighted sum of the representations of its neighbors, where the weights are learned by the model. This process is repeated multiple times to aggregate information from multiple hops in the graph, allowing LightGCN to capture complex relationships between nodes. Finally, the node representations learned by LightGCN can be used for recommendation by computing a score for each user-project pair, indicating the likelihood of the user being interested in the project. These scores can be used to rank the projects for each user and make recommendations.

The architecture of LightGCN consists of the embedding layer, the design of light graph convolution, and layer combination.

3.1.1 Embedding Layer

The vast majority of recommendation systems learn user and item representations to estimate user's preference on any item[20]. In online labor markets, workers and projects are described with an embedding vector $h_w \in R^d, h_p \in R^d$, respectively. The embedding matrix can be built as follows:

$$H = \begin{bmatrix} h_{w_1} \\ \vdots \\ h_{w_n} \\ h_{p_1} \\ \vdots \\ h_{p_m} \end{bmatrix} = \begin{bmatrix} h_{w_{11}} & \dots & h_{w_{1d}} \\ \vdots & \ddots & \vdots \\ h_{w_{N1}} & \dots & h_{w_{Nd}} \\ h_{p_{11}} & \dots & h_{p_{1d}} \\ \vdots & \ddots & \vdots \\ h_{p_{M1}} & \dots & h_{p_{Md}} \end{bmatrix} \tag{1}$$

where N and M are the numbers of users and items. h_w and h_p are ID embeddings that will be fed into worker-project interaction graphs.

3.1.2 Light Graph Convolution

LightGCN implements the concept of Graph Convolutional Network and acquires node representations through the smoothing of features across nodes that are connected and at a maximum of K steps away from the central node. This method has been presented in previous research such as [10, 19]. LightGCN simplifies the traditional graph convolutional layer by eliminating feature transformation and nonlinear activation. The only operation maintained in this model is a straightforward weighted sum aggregation, which is defined as:

$$h_w^{(k+1)} = \sum_{p \in N_w} \frac{1}{\sqrt{|N_w|}\sqrt{|N_p|}} h_p^k \tag{2}$$

$$h_p^{(k+1)} = \sum_{w \in N_i} \frac{1}{\sqrt{|N_p|}\sqrt{|N_w|}} h_w^k \tag{3}$$

where h_w^k,h_p^k is the embedding of worker and project in k LGC layer. N_w,N_p denote the direct neighbor of the target worker and project. $|N_w|$ and $|N_p|$ denote the neighbor amount. The symmetric normalization term $\frac{1}{\sqrt{|N_p|}\sqrt{|N_w|}}$ follows the design of standard GCN[10], which helps prevent the growth of the scale of embeddings during graph convolution operations. Other normalization options, such as the L1 norm, can also be considered, but through empirical testing, the use of symmetric normalization has been found to yield favorable results [8].

3.1.3 Layer Combination

In LightGCN, the final representation of the node is computed by combining all embeddings obtained in each LGC layer. It's worth noticing that the origin embedding at 0-th layer, i.e., $h_w^{(0)}$ for the worker and $h_p^{(0)}$ for the project, is as well needed to be combined. Thus, the final representation of a worker or a project is form by combining the embeddings obtained at each layer:

$$h_w = \sum_{k=0}^{K} \alpha_k h_w^{(k)} \tag{4}$$

$$h_p = \sum_{k=0}^{K} \alpha_k h_p^{(k)} \tag{5}$$

where $\alpha_k \geq 0$ is the significance of the k-th layer embedding in forming the final embedding represented by a weighting factor, which can either be manually tuned as a hyperparameter or learned as a model parameter. In this paper, we closely follow the value of α_k in LightGCN work [8], which sets α_k uniformly as $1/(K+1)$ to maintain the simplicity of LightGCN and avoid overcomplicating the model.

3.2 Proposed Framework

3.2.1 Overview of the Proposed Framework

Conventional methods focus on the user's behavior data and construct a user-item graph for recommendation purposes. However, this approach is limited as it ignores the attributes of users and items and their impact on the recommendation. Behavioral data is important but not the only consideration in the recommendation system, especially in the case of the labor market. Conventional graph neural network methods such as lightGCN only take into account explicit connections and extract information from a user's point of view. However, there are also crucial implicit connections between users and items in the labor market recommendation.

Therefore, this paper argues that the recommendation problems in the labor market are mutual selection problems that require considering both the user's preferences and their abilities. However, most current recommendation strategies are user-oriented, neglecting the important factor of item matching, leading to suboptimal recommendation results.

The proposed recommendation framework incorporates attribute features derived from regular worker behavior and fuses them with explicit features to build a comprehensive worker-project network structure. Skill is chosen as the bridge for implicit feature construction, and the model establishes a graph structure of worker-skill-item. The framework also includes a matching degree matrix to extract skill information between workers and projects, providing a bidirectional recommendation that takes into account both worker preferences and abilities. The overview of our proposed GCNRec framework is shown in Figure 1.

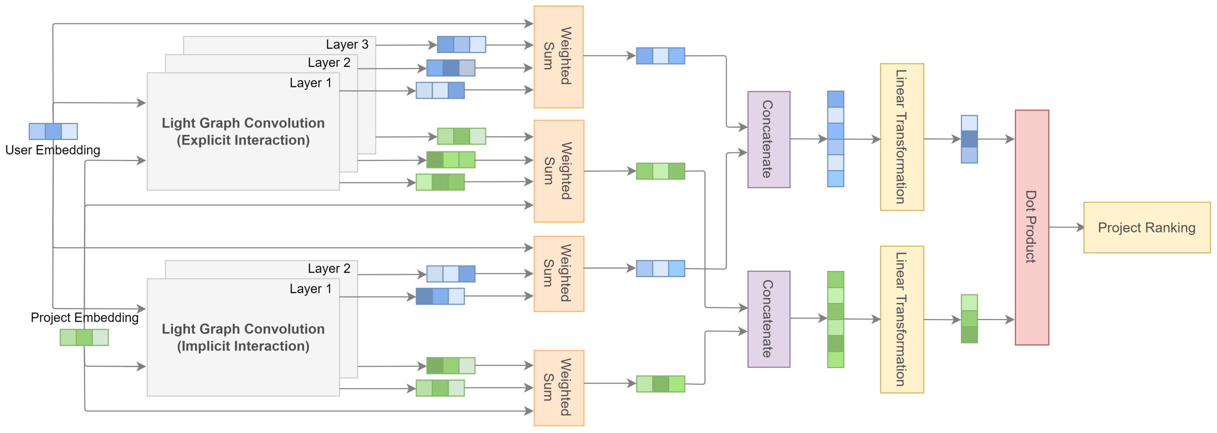

Figure 1: An overview of our proposed GCNRec framework. In this model, the number of LGC layers on explicit interaction and implicit interaction is set to 3 and 2, respectively, which leads to good performance in general.

3.2.2 Graph Construction and LGC Transformation

Our initial goal is to create a graph that incorporates both explicit and implicit interactions in the online labor market field, as mentioned earlier. We extract interactions from dataset and form two symmetrically normalized adjacency matrix $\widetilde{A}_1$, $\widetilde{A}_2$ same as LightGCN[8],$\widetilde{A}_1$ includes the worker-project explicit interaction and $\widetilde{A}_2$ includes the worker-project implicit interaction. Based on different kinds of interaction, we can employ different numbers of LGC layers as Figure 1 uses 3 LGC layers and 2 LGC layers, respectively. The result of LGC transformation can be computed as follows:

$$H_1 = \frac{1}{K_1+1}(H_1^{(0)} + \widetilde{A}_1 H_1^{(0)} + \cdots + \widetilde{A}_1^{K_1} H_1^{(0)}) \tag{6}$$

$$H_2 = \frac{1}{K_2+1}(H_2^{(0)} + \widetilde{A_2}H_2^{(0)} + \cdots + \widetilde{A_2}^{K_1}H_2^{(0)}) \tag{7}$$

where K_1 and K_2 denotes the number of LGC layer of two interactions, $H_1^{(0)} \in \mathbb{R}^{(N+M)\times D}$ denotes the original embedding of user and project in which D is the size of the embedding dimension.

3.2.3 Embedding Fusion

Given that explicit and implicit features may have varying levels of influence on the recommendation, simply combining their feature matrices is not appropriate. To incorporate the impact of implicit features without compromising the influence of dominant features, we train the weights of both dominant and implicit features through backpropagation through a linear layer. This linear layer adjusts the relative importance of each feature, leading to a more effective final recommendation result.

After the LGC transformation, we need to split the result H_1 to obtain the worker embedding W_1 and project embedding P_1. We do the same operation on P_2. Then, we concatenate W_1 and W_2 as well as P_1 and P_2, which is the most straightforward way to combine embeddings[24]:

$$W = W_1 \mid\mid W_2 \tag{8}$$

$$P = P_1 \mid\mid P_2 \tag{9}$$

where $W \in \mathbb{R}^{(N\times 2D)}$,$P \in \mathbb{R}^{(M\times 2D)}$. After embedding concatenation, we do linear transformation to adjust the size of embedding:

$$E_w = WM_w \tag{10}$$

$$E_p = PM_p \tag{11}$$

where $W_u, W_p \in \mathbb{R}^{(2D\times D)}$ is a trainable matrix whose value can be adjusted automatically through backpropagation. According to the nature of matrix multiplication, the linear transformation can act as a distributor to adaptively distribute the amount of influence of explicit interaction and implicit interaction in the final representation, which guarantees the expressive power of our framework.

3.2.4 Project Ranking

Finally, to obtain the rank of each project to target user, we conduct dot product:

$$\widehat{y}(w,p) = e_w^T e_p \tag{12}$$

where $\widehat{y}$ is the prediction score of project p to user u. $e_u \in E_u$ is the final embedding of a single user and $e_p \in E_p$ is the final embedding of a target project. The dot product is a widely used approach to ranking project [20].

3.3 Model Training and Regularization

We train our recommendation framework by optimizing Bayesian Personalized Ranking (BPR) loss[11], which are widely used in various kinds of recommendations[3, 17]. BPR loss encourages the model to compare the relative order between observed worker-project interactions and their counterparts:

$$L_{BPR} = -\sum_{w=1}^{M}\sum_{i\in N_w}\sum_{j\notin N_w} \ln\sigma(\widehat{y}_{wi} - \widehat{y}_{wj}) + \lambda \mid\mid E^{(0)} \mid\mid^2 \tag{13}$$

where λ denotes L2 coefficient used for regularization. We employ Adam optimizer[6] and divide the dataset into mini-batch.

4 Experiments

4.1 Dataset

Data used in this research was collected from Freelancer.com, a major online labor marketplace, through the website's open API. The data covers the period between January 2000 and March 2017 and contains details about users, projects, bids, and project reviews.

To guarantee the dataset's quality, we selected only those workers who had interacted with at least 20 projects. To tackle the cold-start issue, we partitioned each worker's projects into training, validation, and test sets. We randomly selected 80% of the projects for training and kept 10% as a validation set to fine-tune the parameters. This approach ensured that all workers appeared in both the training and test sets. The data after preprocessed is available in Table 1. The workers' explicit interactions with projects refer to their bidding operation in the original dataset, while implicit interactions are those where a worker's skill tags match the project's category tags. We obtained the implicit interactions by extracting the skills tags of each worker and matching them with the project's category tags.

Table 1: Dataset after Preprocessing

Attributes	Amount
Number of workers	10,692
Number of projects	46,534
Number of total categories	989
Number of worker-project explicit interaction	1,683,062
Number of worker-project implicit interaction	3,170,345

4.2 Experimental Settings

4.2.1 Baseline Models

To illustrate the improvements of our framework in recommendation in the online labor market. We compare our framework with the following models:

- MF[11]: Matrix Factorization(MF) is a basic recommendation model in collaborative filtering. By utilizing Bayesian personalized ranking (BPR) loss, we streamline the model optimization process to solely consider the user-project interaction.
- NGCF[17]: Neural Graph Collaborative Filtering (NGCF) is a model that leverages a graph convolutional network and surpasses other GCN-based models such as GC-MC [1] and PinSage [25]. NGCF's approach entails utilizing graph convolutional layer propagation to investigate high-order connectivity in the user-item graph.

4.2.2 Evaluation Metrics

In order to assess the effectiveness of the recommendation model, we utilize four popular evaluation methods: Precision@K, Recall@K, F1@K, and NDCG@K. For our study, we set K to a default value of 20. The Precision, Recall, and F1 metrics are utilized to determine the accuracy of the recommendation. Precision measures the ratio of user bids for a project within the recommended K items. Recall measures the fraction of user bids for a project among the complete project set. The aforementioned metrics can be calculated using the following equations:

$$Precision@K(u) = \frac{\mid R^k(u) \cap T(u) \mid}{K} \tag{14}$$

$$Recall@K(u) = \frac{\mid R^k(u) \cap T(u) \mid}{\mid T(u) \mid} \tag{15}$$

$$F1@K(u) = \frac{2 \times Precision@K(u) \times Recall@K(u)}{Precision@K(u) + Recall@K(u)} \tag{16}$$

where $T(u)$ denotes the ground truth project set, $K(u)$ denotes the top-K recommended project set. NDCG takes the project ranking position into account, which is not considered in the previous three metrics.

$$NDCG@K = \frac{1}{\mid U \mid} \sum_{u \in U} \frac{\sum_{k=1}^{K} \frac{I(R_k^K(u) \in T(u))}{\log(k+1)}}{\sum_{k=1}^{K} \frac{1}{\log(k+1)}} \tag{17}$$

where $R^k(u)$ is the k^{th} project in top-K recommended project set $R^k(u)$.

4.3 Experiment Results

4.3.1 Model Training Process

In all models, the embedding size for both the worker and project are fixed to 64, and parameters are initialized utilizing the Glorot & Bengio method [6]. The learning rate is set to 0.001, and the mini-batch size is set to 2048. We search for the L2 regularization coefficient λ in the range of $1e^{-6}, 1e^{-5}, \ldots, 1e^{-2}$. The number of layers for the two graphs is explored within the ranges of $1, 2, 3, 4$ and $1, 2, \ldots, 10$, respectively. The models are trained for 1000 epochs with an early stopping mechanism of 100. The optimal performance is achieved with λ=0.001, K_1=2, and K_2=3.

4.3.2 Model Performance Comparison

Table 2 shows a comparison of the overall performance of three different models: Matrix Factorization (MF), NGCF (Neural Graph Collaborative Filtering), and GCNRec (Graph Convolutional Networks for Recommendation). The performance measures are Precision, Recall, F1, and NDCG (Normalized Discounted Cumulative Gain).

Table 2: Overall Performance Comparison

Models	Precision	Recall	F1	NDCG
MF	0.0363	0.0373	0.0368	0.0469
NGCF	0.0434	0.0476	0.0454	0.0572
GCNRec	0.0458	0.0516	0.0485	0.0610
%Improv.	26.17%	12.83%	31.79%	30.06%

The GCNRec model showed the highest improvement over the MF model with a 26.17% increase in Precision, a 12.83% increase in Recall, a 31.79% increase in F1, and a 30.06% increase in NDCG. It demonstrates the inability of the MF model to aggregate high-hop information and the strong capabilities of GNN-based models. This result highlights the importance of indirect interactions between a project/worker and the target worker in the online labor market. Specifically, GCNRec achieves 5.53% higher Precision, 8.4% higher Recall, 6.83% higher F1, and 6.64% higher NDCG compared to NGCF, emphasizing the crucial role of project and worker properties in project recommendation in the online labor market.

4.4 The Influence of LGC layer

In Table 3, a comparison is presented among various combinations of explicit and implicit layers employed in a recommendation system, evaluated based on four performance metrics, namely, Precision, Recall, F1, and NDCG. The explicit interaction is modeled using 1 to 4 LGC layers, while the implicit interaction is modeled using 1 to 4 LGC layers as well. For every combination of explicit and implicit layers, the table displays the corresponding metric values.

Table 3: Performance Comparison of Different Layers

Explicit Layer	Implicit Layer	Precision	Recall	F1	NDCG
1	1	0.0449	0.0501	0.0474	0.0594
	2	0.0445	0.0497	0.0469	0.0591
	3	0.0438	0.0487	0.0461	0.0584
	4	0.0453	0.0506	0.0478	0.0608
2	1	0.0451	0.0503	0.0476	0.0601
	2	0.0449	0.0502	0.0474	0.0601
	3	0.0448	0.0502	0.0473	0.0599
	4	0.0455	0.0514	0.0483	0.0607
3	1	0.0457	0.0510	0.0483	0.0608
	2	*0.0458	*0.0516	*0.0485	*0.0610
	3	0.0453	0.0510	0.0480	0.0607
	4	0.0453	0.0513	0.0482	0.0602
4	1	0.0449	0.0500	0.0473	0.0599
	2	0.0443	0.0508	0.0479	0.0608
	3	0.0458	0.0505	0.0480	0.0609
	4	0.0455	0.0515	0.0483	0.0608

The results show that the best overall performance is achieved with 3 LGC layers for explicit interaction and 2 LGC layers for implicit interaction, as indicated by the asterisks (*), as the precision, F1 and NDCG values are the highest. When the number of layers is greater than 3, the performance drops quickly. The reason is that too many layers can lead to an over-fitting problem, which makes the node embedding indistinguishable. The table also shows that the values of precision, recall, F1, and NDCG become more stable as the number of layers increases, and the average value of each metric is higher than the two baselines, indicating the ability of the model to reduce the over-fitting problem and the stability of the framework.

4.5 User Case Analysis

History skills and profile skills are important factors that could be used to identify a suitable project for a worker. The profile skills represent the skills that the worker has in their profile, and these skills are usually used to describe their expertise and areas of interest. The history skills, on the other hand, are the skills that the worker has utilized in the past and could indicate the types of projects that they have worked on and the areas where they have experience. By considering both the profile skills and the history skills of a worker, it is possible to determine the types of projects that would be best suited to their expertise and experience, increasing the chances of project success. In this section, we choose a specific worker with ID 6611. Table 4 provides detailed skill information about this worker.

Table 4: Detailed Skill Information of User 6611

Skill Type	Skill Name
Profile Skill	Mobile Phone
	iPhone
	Objective C
	iPad
	Swift
	Apple Watch
History Skill	iPhone
	Mobile Phone
	Objective C
	iPad
	Swift

We made a comparison of our proposed model GCNRec with BPR and NGCF in recommending projects to worker 6611. Each model recommends 20 projects, which are listed in Table 5 along with a ranking (number in parentheses) and a star (*) indicating if the project is deemed as the most suitable for the worker.

Table 5: Top 20 Recommended Projects of User 6611

Model	Top 20 Projects				
BPR	24169(2)	23660(1)	23732(2)	23656(2)	23729(2)
	24106(1)	23670(1)	24019(2)	23673(1)	24117(1)
	23645(1)	24128(1)	24167(1)	23657(1)	*24114(1)
	24094(1)	23304(1)	23727(1)	23983(1)	24298(1)
NGCF	24221(1)	23732(2)	*23731(2)	23645(1)	24117(1)
	23983(1)	24120(1)	24169(2)	23977(2)	23660(1)
	23725(1)	*23985(1)	23676(1)	23304(1)	23331(1)
	23648(1)	23729(2)	24271(2)	24109(2)	24158(1)
GCNRec	23670(1)	23657(1)	23729(2)	23983(1)	24280(2)
	23981(2)	44379(2)	*23730(2)	23645(1)	23732(2)
	*23731(2)	24094(1)	23977(2)	*24114(1)	*23329(1)
	23996(1)	24271(2)	23649(2)	23993(2)	24217(2)

In order to demonstrate the importance of incorporating domain knowledge in the online labor market, we have calculated the overlap between the skill tags associated with a worker and the recommended projects. The worker's tags are skills that we have previously mentioned, while the tags for projects are required skills provided by the employer. Both the worker and projects can have multiple skill tags. Table 5 shows that the average number of overlapping tags between the worker and projects recommended by BPR, NGCF, and GCNRec is 1.25, 1.35, and 1.5, respectively. The highest value of average overlap indicates that GCNRec is able to effectively uncover the relationship between the worker's skills and project categories. This demonstrates the usefulness of this capability, as the number of accurately recommended projects by BPR, NGCF, and GCNRec is 1, 2, and 4, respectively.

5 Conclusion

In this paper, we present a novel project recommendation system for the online labor market, called GCNRec, that leverages the power of graph convolutional networks and unique features of the online labor market. Our framework focuses on matching the skills of freelancers with the requirements of projects by considering both explicit interactions, such as the match between a worker's skills and a project's requirements, and implicit interactions, such as the worker's bidding history. The results of our experiments on a large real-world dataset demonstrate the outstanding performance of GCNRec compared to other recommendation systems. Our model has the potential to improve the work efficiency and income of freelancers by assisting them in finding the right projects for their skills. In future work, we plan to incorporate additional relevant information from the online labor market into our recommendation framework.

Funding

The work is supported by the Young Scientists Fund of the National Natural Science Foundation of China (Grant No.72101023) and the Beijing Social Science Foundation (Grant No. 21JCC092), and in part by Beijing Logistics Informatics Research Base, China.

References

[1] Berg, R., Kipf, T. & Welling, M. (2017). Graph convolutional matrix completion. *ArXiv Preprint ArXiv:1706.02263.* 2017.

[2] Chen, C., Ma, W., Zhang, M., Wang, Z., He, X., Wang, C., Liu, Y. & Ma, S. (2021). Graph Heterogeneous Multi-Relational Recommendation. *Proceedings Of The AAAI Conference On Artificial Intelligence.*35, 3958-3966, 2021.

[3] Chen, J., Zhang, H., He, X., Nie, L., Liu, W. & Chua, T. (2017). Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. *Proceedings Of The 40th International ACM SIGIR Conference On Research And Development In Information Retrieval.* pp. 335-344, 2017.

[4] Christensen, I. & Schiaffino, S. (2011). Entertainment recommender systems for group of users. *Expert Systems With Applications.* 38, 14127–14135, 2011.

[5] Dave, V., Zhang, B., Al Hasan, M., AlJadda, K. & Korayem, M. (2018). A combined representation learning approach for better job and skill recommendation. *Proceedings Of The 27th ACM International Conference On Information And Knowledge Management.* pp. 1997-2005, 2018.

[6] Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings Of The Thirteenth International Conference On Artificial Intelligence And Statistics.* pp. 249-256, 2010.

[7] Goldberg, D., Nichols, D., Oki, B. & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications Of The ACM.* **35**, 61-70, 1992.

[8] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y. & Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. *Proceedings Of The 43rd International ACM SIGIR Conference On Research And Development In Information Retrieval.* pp. 639-648, 2020.

[9] Khatua, A. & Nejdl, W. (2020). Matching recruiters and jobseekers on twitter. *2020 IEEE/ACM International Conference On Advances In Social Networks Analysis And Mining (ASONAM).* pp. 266-269, 2020.

[10] Kipf, T. & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *ArXiv Preprint ArXiv:1609.02907.* 2016

[11] Rendle, S., Freudenthaler, C., Gantner, Z. & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. *ArXiv Preprint ArXiv:1205.2618.* 2012.

[12] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. & Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. *Proceedings Of The 1994 ACM Conference On Computer Supported Cooperative Work.* pp. 175-186, 1994.

[13] Schafer, J., Konstan, J. & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining And Knowledge Discovery.* 5, 115—153, 2001.

[14] Sheu, H. & Li, S. (2020). Context-aware graph embedding for session-based news recommendation. *Fourteenth ACM Conference On Recommender Systems.* pp. 657-662, 2020.

[15] Sun, J., Zhang, Y., Guo, W., Guo, H., Tang, R., He, X., Ma, C. & Coates, M. (2020). Neighbor interaction aware graph convolution networks for recommendation. *Proceedings Of The 43rd International ACM SIGIR Conference On Research And Development In Information Retrieval.* pp. 1289-1298, 2020.

[16] Wang, X., He, X., Cao, Y., Liu, M. & Chua, T. (2019). Kgat: Knowledge graph attention network for recommendation. *Proceedings Of The 25th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining.* pp. 950-958, 2019.

[17] Wang, X., He, X., Wang, M., Feng, F. & Chua, T. (2019). Neural graph collaborative filtering. *Proceedings Of The 42nd International ACM SIGIR Conference On Research And Development In Information Retrieval.* pp. 165-174, 2019.

[18] Wu, C., Wu, F., An, M., Huang, J., Huang, Y. & Xie, X. (2019). NPA: neural news recommendation with personalized attention. *Proceedings Of The 25th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining.* pp. 2576-2584, 2019.

[19] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T. & Weinberger, K. (2019). Simplifying graph convolutional networks. *International Conference On Machine Learning.* pp. 6861-6871, 2019.

[20] Wu, S., Sun, F., Zhang, W., Xie, X. & Cui, B. (2020). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys (CSUR).* 2020

[21] Wu, J., He, J. & Xu, J. (2019). Net: Degree-specific graph neural networks for node and graph classification. *Proceedings Of The 25th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining.* pp. 406-415, 2019.

[22] Wu, L., Sun, P., Fu, Y., Hong, R., Wang, X. & Wang, M. (2019). A neural influence diffusion model for social recommendation. *Proceedings Of The 42nd International ACM SIGIR Conference On Research And Development In Information Retrieval.* pp. 235-244, 2019.

[23] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. & Philip, S. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions On Neural Networks And Learning Systems.* 32, 4—24, 2020.

[24] Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K. & Jegelka, S. (2018). Representation learning on graphs with jumping knowledge networks. *International Conference On Machine Learning.* pp. 5453-5462, 2018.

[25] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. *Proceedings Of The 24th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining.* pp. 974-983, 2018.

[26] Ying, Z., Bourgeois, D., You, J., Zitnik, M. & Leskovec, J. (2019). Gnnexplainer: Generating explanations for graph neural networks. *Advances In Neural Information Processing Systems.* 32, 2019.

[27] Zhang, M. & Chen, Y. (2018). Link prediction based on graph neural networks. *Advances In Neural Information Processing Systems.* 31, 2018.

Cite this paper as:
Ma, Y.; Ma, Z; Li Y.; Gao H.; Xue, Y. (2023). A Project Recommender Based on Customized Graph Neural Networks in Online Labor Market, *International Journal of Computers Communications & Control*, 18(4), 5173, 2023.
https://doi.org/10.15837/ijccc.2023.4.5173