Research on Machine Vision Effect Based on Graph Neural Network Decision

Jide Huo1, *, Weining Sun2, Huaixuan Dai2

1Hefei University of Technology, Hefei, Anhui, 230009
2College of Mechanical and Electrical Engineering, Hohai University, Changzhou, Jiangsu, 213022

*Corresponding author: jidehuo2020@hfut.edu.cn

Abstract. With the continuous expansion of the scale of the network, modeling the complex graph structure is a major challenge for the recommendation task. There are conflicts between these complex information, which will directly affect the recommendation results. For this reason, a graph neural network recommendation method based on multi-branch decision is proposed. The algorithm models the social network graph and the user project graph through the neural network, connects the two graphs intrinsically, and learns the feature vectors of the target users in the social space and the project space. Then two feature vectors are connected in series by MLP to extract the potential feature vector of the user. Finally, the prediction score is generated by integrating the probability matrix decomposition model. In this method, the graph neural network is used to model the program data and dependence graph, and the effective program features are automatically extracted from the source code, and then the extracted features are input into the downstream model for loop vectorization parameter prediction. Finally, the attention mechanism is used for information fusion to get the final session representation and predict the next interactive item. Comparative experiments are carried out under the two scenarios of e-commerce and civil aviation respectively. The experimental results show that, compared with the optimal benchmark model, the improvement of MGSP model on each index of e-commerce data set is more than 1%, and that of civil aviation data set is about 3%, which verifies the effectiveness of MGSP model. On the LLVM cyclic vector test set, the proposed method achieves a speedup of 2.08 times and improves the performance by 12% compared with the existing methods. In order to solve the problem that the existing deep learning model considers the program code as a serial sequence and misses a large performance optimization space, a new program heuristic method based on depth map network is proposed to achieve optimization.

Keywords: GNN, MLP, Information fusion, Multi-branch decision.

1. Introduction
The existing deep learning-based heuristic methods for building compilers without feature engineering usually use cyclic neural networks, such as its variant long-term and short-term memory networks, to
construct compiler heuristics [1]. This method usually serializes the structure of the program after mixed coding, which only represents the sequence relationship before and after the program, and cannot contain the rich grammatical and semantic relations of the program itself. In recent years, graph neural network has achieved great success in many fields, such as computer code recognition, link prediction, entity classification and so on. Compared with the traditional neural network, which can only update the weight, the learning of graph neural network takes place in every link, including the update of edge state, node state and global information [2].

In order to solve the above problems, a session-aware recommendation model based on multi-graph neural network is proposed in this paper. The model is mainly composed of four parts:

1) Composition module: the object transfer diagram and collaborative correlation diagram are constructed according to the target session and all the conversations in the training set [3].

2) Graph neural network module: GNN, is applied to the constructed ITG and CRG to aggregate node information, and two kinds of node representations are generated.

3) Double-layer attention module: considering the implicit dependency relationship between the two types of node representations captured, that is, the implicit relationship between personalized preference and collaborative information, and the self-attention mechanism can capture the implicit dependency [4] relationship between node representations, it has a strong ability to extract features, so this module first uses the self-attention mechanism to model the two types of node representation at the same time.

4) Prediction module: considering the different importance of different features in the prediction task, this module uses the attention mechanism for information fusion to get the final session representation and predict the next interactive item [5].

We discussed how to use GNN modeling to learn the potential characteristics of users. The main contributions to this work are as follows. 1. A neural network architecture is proposed to model and learn the potential characteristics of users, which is integrated into the probability matrix decomposition to form recommendations through the internal relationship between the user project graph and the social network graph. Two [6]. This paper proposes a method of linking user opinions and project interactions in the user project diagram. 3. A wide range of experimental data sets are carried out in two real data sets to prove the effectiveness and feasibility of the combination of GNN and MF.

2. Graph neural network model

2.1. Model building

Graphvector-gnn recursively updates the features of nodes according to the types of edges in the graph by taking the adjacency matrix and node feature matrix of the graph data as input. After several recursions, the Graphvector-GNN updates a new feature representation for each vertex, which can update similar nodes to the neighboring vector space considering the relationship between node features and graph features. In recursive updating, the embedding of all vertices in the previous round is completed before the embedding of all vertices in the new round begins. Finally, feature vectors of fixed size are extracted by node aggregation.

The innovation of Graphvector-GNN layer lies in its ability to model multiple relationships and complete information transfer and aggregation across multiple graphs. In this paper, the graph data is expressed as, where V and E are the set of vertices and edges in the graph respectively. The initialization task of node feature matrix X is obtained by Word2vec model. The unique node features of each edge are generated through edge type and feature matrix X. Finally, feature extraction of edge type is completed through aggregation. The calculation process of eigenmatrix is:

$$\mu = G_a * X + G_b * X + \ldots + G_c * X$$  \hspace{1cm} (1)
Then the graph data $G$ and the updated feature matrix are input into the embedded layer successively. After the update, the features of each node are obtained. Finally, the feature matrices are added to one row to complete the generation process of the embedded vector. The feedforward neural network is defined as

$$\mu_v^{i+1} = \tanh(W_0 X_v + \sigma(\sum_{u \in N(v)}(\mu_u^i)))$$ (2)

$$\sigma(l) = W_1^* \text{ReLU}(W_2^* \text{ReLU}(W_n^* l))$$ (3)

As the number of iterations increases, node features can more effectively abstract the global program representation.

![Diagram](image)

**Fig. 1** Embed the vector update process

### 2.2. Collaborative correlation module

ITG is a directed weighted graph, represented by $G_{itg} = (V, E_{itg})$, where $V$ represents the collection of items that have appeared in the target session $S$, and $E_{itg}$ represents the collection of directed edges between items. $G_{itg}$ is constructed as follows: In session $S$, if the user interacts with the item $v_i$ and then interacts with the item $v_j$, then the edge weight corresponding to $(v_i, v_j)$ is added by 1.

The gating unit is introduced into GNN, and the gating chart sequence neural network is proposed, which greatly enhances the ability of GNN to deal with sequence data. Taking ITG as an example, this paper introduces how to learn the representation of nodes through the sequential neural network of gating chart.

$$a'_i = \text{Concat}(A_i^{in} \left[ s_1^{i-1}, s_2^{i-1}, \ldots, s_n^{i-1} \right]^T W_a^{in} + b^{in})$$

$$A_i^{out} \left[ s_1^{i-1}, s_2^{i-1}, \ldots, s_n^{i-1} \right]^T W_a^{out} + b^{out})$$

$$z'_i = \sigma(W_a a'_i + P_s s_i^{i-1})$$ (5)

$$r'_i = \sigma(W_a a'_i + P_s s_i^{i-1})$$ (6)

$$s'_i = \tanh(W_a a'_i + P_s (r'_i \oplus s_i^{i-1}))$$ (7)
\[ s'_i = (1 - z'_i) s_{i-1}' + z'_i s'_i \]  

Doing the same with CRG yields an alternative representation of each item in the session, denoised as \( S_{\text{crg}} = [s_{\text{crg,1}}, s_{\text{crg,2}}, \ldots, s_{\text{crg,n}}] \). Since the two types of representations contain personalized preference information and collaborative information respectively, the two types of representations are respectively called personalized preference item representation and collaborative information item representation in this paper.

3. GNN_MF framework

The goal of GNN_MF is to find the underlying model of the user and the project \((U \in R^{f \times n} \text{ and } V \in R^{f \times m})\) and to reconstruct the rating matrix \( R \) by \((UV)\) to predict the score. Its conditional distribution can be defined as:

\[ p(R | U, V, \sigma_R^2) = \prod_i \prod_j N(\sigma_{ij}^2) \]

4. GNN Results

SR-GNN and GC-SAN are selected as the comparison model. Set the value range of embedded dimension \( d \) from 10 to 120, select evaluation indicators, and experiment on two data sets. The experimental results are shown in the figure. It can be seen from figure that the performance of the
three models improves with the increase of embedded dimension \( d \), but when embedded dimension \( d \) reaches 60, the performance of the three models no longer improves with the increase of embedded dimension \( d \). In addition, it is obvious that the performance of MGSP model is better than that of SR-GNN and GC-SAN in all embedded dimensions.

![Fig. 3 Performance comparison of different models under different embedding dimensions](image)

Increasing the number of embedded layers (that is, increasing the number of neighborhood aggregation iterations) can improve performance. However, when the number of embedded layers is more than 5 layers, the performance tends to decline steadily, which will lead to the over-fitting phenomenon and the decrease of the prediction accuracy of the verification set.

![Fig. 4 The effect of embedding layers](image)

In order to verify the effect of TWD-GNN algorithm under sparse data, a part of data is deleted from the training set, and GNN with different sparsity is obtained. As shown in the figure. With the continuous increase of data sparsity, the accuracy of CF algorithm decreases the fastest.

![Fig. 5 RMSE comparison](image)
The performance of the model is tested in different situations. Because the two models perform best in the comparative model, SR-GNN and GC-SAN are chosen as the comparative model in this experiment. Set the session length from 1 to 20, select the evaluation index, and experiment on two data sets. The experimental results are shown in figure 6. Comparing the performance of the three models in long sessions and short sessions, we can find that the prediction effect is the best when the session length is 2 in Diginetica dataset and 3 in 1 game 64 dataset. With the increase of session length, the prediction effect of the three models shows a downward trend, but the downward trend of MGSP model is the slowest, and the effect is obviously better than SR-GNN and GC-SAN when the session is longer.

![Fig. 6 MAE comparison](image)

5. Conclusion
In this paper, a graph neural network recommendation method based on multi-branch decision is proposed. In order to solve the problem of information conflict in complex networks, the domain space of uncertain problems can be effectively reduced to a limited set by dividing large-scale data into three decision-making branches. Social graphs are included in recommendations, while many real-world industries have a wealth of other faceted information about users and products. For example, users and projects are associated with rich properties. Therefore, exploring the recommendation of graph neural networks with attributes will be an interesting direction in the future. The MGSP model in this paper is better than the benchmark models such as SR-GNN and GC-SAN in all indexes. In addition, the effectiveness of each component of the model is verified by ablation experiments.

The results show that the performance of GraphVector is 12% higher than that of the latest prediction model NeuroVector. This paper introduces GraphVector, which is an end-to-end vectorization framework, which aims to realize the heuristic optimization of machine learning without manual feature extraction, so as to obtain the best vectorization factor of C loop code.
References

[1] Peng Gu, Chuanmin Zhu, Yiqing Yu, Dinghao Liu, Zhan Tao, Yinyue Wu. Evaluation and prediction of drilling wear based on machine vision [J]. The International Journal of Advanced Manufacturing Technology, 2021 (prepublish).

[2] David Jiří, Švec Pavel, Pusker Vít, Garzinová Romana. Usage of Real Time Machine Vision in Rolling Mill [J]. Sustainability, 2021, 13(7).

[3] Wang Jinya, Li Zhenye, Ni Chao. On-line color separation recognition algorithm for solid wood flooring based on machine vision [J/OL]. Journal of Forestry Engineering: 1-5 [2021-04-21]. https://doi.org/10.13360/j.issn.2096-1359.202012012.

[4] Shen Zhengfu, Wang Huifen, Yuan Tangxiao. Intelligent station design of size inspection based on machine vision [J/OL]. Computer measurement and control: 1-6 [2021-04-21]. Http://kns.cnki.net/kcms/detail/11.4762.TP.20210419.1520.016.html.

[5] Zhang Wuyi, Zhao Qiangsong, Wang Dongyun. The present situation and development trend of machine vision [J]. Journal of Central Plains Institute of Technology, 2008 (01): 9-1215.

[6] Yin Hongpeng. Research on moving object tracking algorithm based on computer Vision [D]. Chongqing University, 2009.