Journal of Computer Science & Technology

Indexed in: SCIE, EI, INSPEC, CBST, DBLP, etc.

Sponsored by: ICT, CAS & CCF

JCST Papers

Only for academic and non-commercial use

Thanks for reading!

JCST WeChat Subscription Account

JCST URL: https://jcst.ict.ac.cn

SPRINGER URL: https://www.springer.com/journal/11390

E-mail: jcst@ict.ac.cn

Online Submission: https://mc03.manuscriptcentral.com/jcst

Twitter: JCST_Journal

LinkedIn: Journal of Computer Science and Technology

Hao PY, Liu SH, Bai C. Intent-aware graph-level embedding learning based recommendation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(5): 1138-1152 Sept. 2024. DOI: 10.1007/s11390-024-3522-9

Intent-Aware Graph-Level Embedding Learning Based Recommendation

Peng-Yi Hao (郝鹏翼), *Member, CCF*, Si-Hao Liu (刘思浩), and Cong Bai* (白　琮), *Member, CCF*

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310012, China

E-mail: haopy@zjut.edu.cn; 202003340313@zjut.edu.cn; congbai@zjut.edu.cn

Received June 18, 2023; accepted February 7, 2024.

Abstract Recommendation has been widely used in business scenarios to provide users with personalized and accurate item lists by efficiently analyzing complex user-item interactions. However, existing recommendation methods have significant shortcomings in capturing the dynamic preference changes of users and discovering their true potential intents. To address these problems, a novel framework named Intent-Aware Graph-Level Embedding Learning (IaGEL) is proposed for recommendation. In this framework, the potential user interest is explored by capturing the co-occurrence of items in different periods, and then user interest is further improved based on an adaptive aggregation algorithm, forming generic intents and specific intents. In addition, for better representing the intents, graph-level embedding learning is designed based on the mutual information comparison among positive intents and negative intents. Finally, an intent-based recommendation strategy is designed to further mine the dynamic changes in user preferences. Experiments on three public and industrial datasets demonstrate the effectiveness of the proposed IaGEL in the task of recommendation.

Keywords recommendation system, graph embedding learning, graph neural network, intent-aware

1 Introduction

Recommendation methods, which can provide personalized content and products to users based on their past behavior, interest, social network relationships, and other information, have become increasingly important in business platforms. In recent years, there has been a growing number of recommendation approaches. They can be divided into two categories. 1) One is collaborative filtering based recommendation[1, 2], which analyzes users' historical behavior data (e.g., ratings, clicks) to infer users' interest, and then makes similarity matches based on users' interest and the characteristics of items to make recommendations. 2) The other is graph-based recommendation[3–5], which constructs graphs according to the interactions between users and items, and then graph neural networks are explored to learn the representations of user nodes and item nodes. Based on the representations of users and items, matrix factorization is always used to get the recommendation lists. The graph-based recommendations can better handle complex user-item interactions and can easily add user attributes, item attributes, and other auxiliary information to the graph structure to improve the accuracy of recommendation.

However, there are still some problems in recommendations. 1) Users' interest and preferences are complex and varied. Even the same user may purchase the same item for different purposes. Therefore, a user's potential intents is often not reflected by a single item. For example, when coke and popcorn are purchased together, the user may be planning to watch a movie. Existing methods such as collaborative filtering tend to recommend food-related items based on item-similarity rather than movie-related items. 2) User preferences can change over time. For example, a user initially buys a few pairs of sneakers

Regular Paper

Recommended by ChinaMM 2023

This work was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant No. LR21F020002 and the National Natural Science Foundation of China under Grant No. 61976192.

*Corresponding Author

because he/she likes to run. However, over time, the user may become less interested in running and more interested in tennis. Conventional recommendation methods such as matrix decomposition often assume that user preferences are static, but in real-world scenarios, recommendation systems need to capture long-term dynamic trends and ignore some temporarily exhibited noise.

To solve the above problems, we propose to learn user intents. User intents usually has two kinds of characteristics: generality and personality. Generality represents the aggregation of a large number of similar behaviors among users, which is applicable to a wide range of scenarios and user groups, and not dependent on specific contexts and scenarios. Personality is based on a specific scenario or context to meet more specific and personalized user needs. Therefore, in this paper, two kinds of intents are generated and explored. One is user generic intents that means the common interest and interrelated items among users, which is explored by discovering the stable structures among items. The other one is user specific intents that means the special interested items of users, which is explored based on users' behavior over a period of time.

According to the above ideas, a novel framework named Intent-Aware Graph-Level Embedding Learning (IaGEL[①]) is proposed for recommendation in this paper. This framework mainly includes capturing intents, learning intent-oriented graph embedding, and intent-based recommendation. To mine the potential intents and the changes of preferences for users, the co-occurrence graph is constructed based on the frequent co-occurrence relationships among items generated by users (e.g., click, purchase). Then, basic generic intents and specific intents of users are generated from the co-occurrence graph. For deeply exploring the real motivations and potential preferences of users, adaptive aggregation is designed to generate diverse user generic intents and user specific intents. After that, contrastive sampling is designed to capture positive samples and negative samples of specific intents. Specifically, positive samples are randomly selected from generic intents, and negative samples are constructed based on the graph corrosion of generic intents. Then, graph-level embedding learning is designed according to the mutual information loss between specific intents with positive generic intent samples and negative generic intent samples, resulting in the representations of user generic intents and specific intents. Finally, for further mining users' preference changes, an intent-based recommendation is designed based on an exploration and exploitation strategy[6], to capture dynamic trends in user interest by dynamically updating the weights of intents during the recommendation process. Experiments conducted on three public datasets show that the proposed IaGEL significantly outperforms the advanced recommended algorithms. The main contributions of this paper can be summarized as follows.

1) A novel framework is proposed for recommendation, which captures the diverse user intents and learns their embedding, and makes a recommendation from the aspect of communities instead of from the aspect of nodes themselves.

2) A community-oriented adaptive aggregation algorithm is designed, which discovers not only the differences but also their relationships among users' intents.

3) An intent-based recommendation is designed based on mutual information calculation among intents, which further explores the users' interest and preferences, resulting in accurate recommendations.

The following content is organized as follows. Section 2 summarizes the related work on recommendation methods and graph embedding learning. Section 3 shows the details of the proposed IaGEL. Section 4 analyzes the extensive experiments, and Section 5 concludes this work.

2 Related Work

2.1 Recommendation Methods

In recent years, recommendation has attracted increasing attention in both academia and industry. Collaborative filtering (CF)[1, 2, 7], as one of the most traditional approaches, has been widely studied and applied in various domains. Its basic idea is to analyze the similarity between users to predict the rating or preference of users for items. In the recent years, the combination of traditional collaborative filtering and neural network have presented the mainstream of the innovation in collaborative filtering approaches. For example, He *et al.*[8] proposed a general framework for collaborative filtering based on neural networks. Wang *et al.*[7] proposed a vector quantization auto-encoder for collaborative filtering. Wang *et*

[①]https://github.com/illyasvia/IaGEL, Sept. 2024.

al.[9] improved the performance of collaborative filtering by systematically profiling the design space of graph neural networks.

On the other hand, with the rapid development of graph neural networks, graph-based recommendations are also popular. Meta-path[3, 4, 10–13] is commonly used for mining potential association patterns in heterogeneous networks. For example, Dong *et al.*[3] proposed a meta-path based random walk to construct heterogeneous neighborhoods of nodes. Yang *et al.*[11] proposed an architecture of hierarchical aggregation to provide useful meta-paths for each object at different levels. Some scholars proposed the idea of converting heterogeneous networks into homogeneous networks by using the co-occurrence graph[5, 14, 15]. For example, Li *et al.*[14] proposed a multi-layer diffusion method on a co-occurrence graph to enrich user behavior. Jiang *et al.*[15] used an attention-based mechanism in the co-occurrence graph. However, these methods are still based on the matrix factorization framework, which can only solve specific applications and cannot provide a generalized learning framework for complex scenarios.

Additionally, most graph-based recommendation approaches fail to take into account the dynamic interactions between users and items. Recently, some approaches have been proposed to capture the intentions of users. For example, Chen *et al.*[16] modeled user intentions as a tuple of action type and product category. Yang *et al.*[17] encoded sequential patterns and modeled user behavior by unifying the latent item-wise relatedness and item-specific behavior correlations. Wang *et al.*[18] captured user intents based on the category information of the items. Guo *et al.*[19] extracted consecutive intent units and fused the representations of all intent units to get the recommendation results. In [20], intents were presented as a heterogeneous network for recommendation. In [21], user intents were treated as purposes for which a user purchases the product, and a ranking model was adopted to aggregate ranking lists for recommendation. Although such researches have the concept of user intents, they always consider the intents from the categories of items, and a lot of work focuses on how to fuse the lists based on the ranking results. In comparison, our work tends to analyze the generality of different people and the personality of the specific person, which captures general intents and specific intents of users through adaptive aggregation, and adopts an intent-based method for recommendation.

2.2 Graph Embedding Learning

Graph embedding learning focuses on learning low-dimensional vector representations of graph entities, such as nodes, edges. Graph embedding is commonly used in a lot of graph-related tasks, such as node classification, link prediction, and recommendation. One of the main approaches is random walk[22–24], where each node is mapped into a low-dimensional vector by using the path of random walk as a corpus. Auto-encoder algorithms are also widely used in graph embedding learning. Kipf *et al.*[25] proposed the variation graph auto-encoders to learn the embedding of nodes in a graph, in which the encoder maps the node feature vectors to a low-dimensional space and the decoder reverts the embedding to the original space. Petar *et al.*[26] proposed the deep graph infomax to learn the embedding of nodes by maximizing the information entropy of local neighbors.

In recent years, graph embedding learning via graph neural networks has also emerged. Welling *et al.*[27] proposed Graph Convolutional Network (GCN) to update the features of nodes by convolving the features of neighboring nodes with those of their own nodes. Velickovic *et al.*[28] proposed Graph Attention Network (GAT), in which weights are calculated by adding the features of a center node's neighboring nodes the weighted features of the center node when updating the features of the nodes. Hamilton *et al.*[29] proposed GraphSAGE, in which the features of neighboring nodes are aggregated based on sampling, and the features of the nodes are updated through a fully connected network. Xu *et al.*[30] proposed Graph Isomorphism Network (GIN), in which the feature representations of the substructures in a graph are summed based on the graph isomorphism property, and then the features of the nodes are updated through a multi-layer fully connected network. In our work, the purpose is to learn the representations of intents instead of the nodes themselves, therefore the subgraph level embedding should be learnt rather than the node level embedding.

3 Proposed Method

3.1 Problem Description

Suppose the original data consists of m users and n items. Each user u_t has a sequence $S_t = c_1^t \to c_2^t \to \ldots \to c_{|S_t|}^t$ including the user's historical behaviors in chronological order, where c_i^t $(1 \leqslant i \leqslant |S_t|)$ denotes the i-th item that u_t has interacted with. Each item

contains different attributes such as review, price, and brand, which have rich semantic information. Using word2vec[31] and one-hot encoding, the attributes of all items are projected into a d-dimensional feature matrix $\boldsymbol{H} \in \mathbb{R}^{n \times d}$, where each row represents the initial feature of one item. The goal is to explore the user's potential interest and preference changes, and then recommend items that the user is most likely to interact with in the future. This paper gives a novel framework for implementing this goal, as shown in Fig.1. The explanations of main notations are given in Table 1.

3.2 Co-Occurrence Graph Generation

To model time-related relationships for items, we construct a co-occurrence graph $\mathcal{G} = (V, \boldsymbol{E}, \boldsymbol{W})$. V denotes the set of nodes containing all items in the dataset. $\boldsymbol{E}$ is described as the adjacency matrix of nodes, which is given as follows:

$$\boldsymbol{E} = \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & e_{ij} & \vdots \\ e_{n1} & e_{n2} & \cdots & e_{nn} \end{pmatrix},$$

where e_{ij} denotes the number of adjacent relations between v_i and v_j in S_t, like $v_i \to v_j$ or $v_j \to v_i$. $\boldsymbol{W} = (w_1, w_2, \ldots, w_n)$ denotes the weights of the nodes in $\mathcal{G}$. Here PageRank[32] is used to calculate the weights of nodes.

3.3 User Generic Intents Generation

To capture the universal interest across users, the

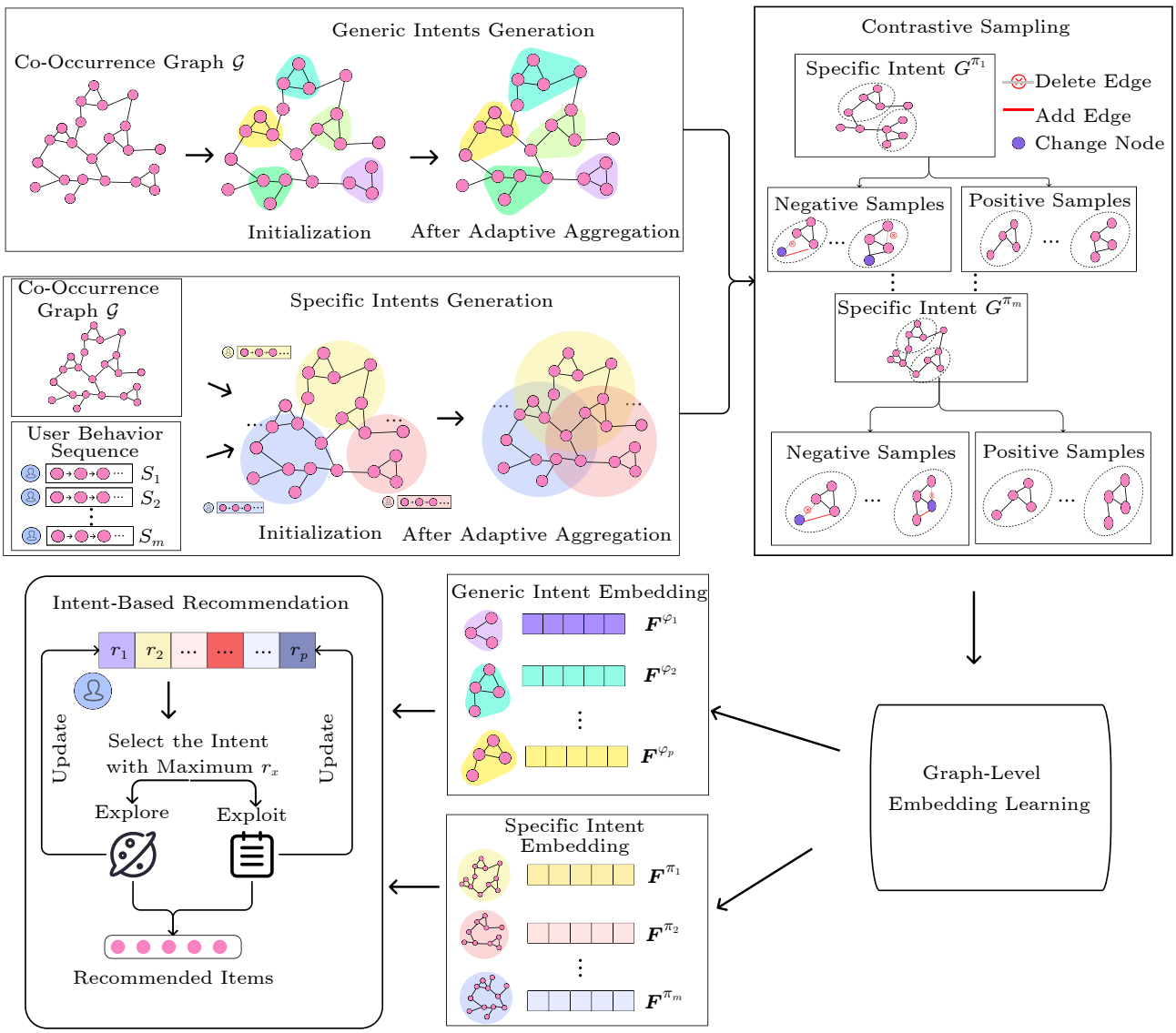

Fig.1. Framework of IaGEL for recommendation.

Table 1. Notations and Explanations

Notation	Explanation
S_t	Historical behavior sequence of u_t
$\boldsymbol{H}$	Feature of items
$\mathcal{G}$	Co-occurrence graph
G^{φ_x}	User generic intent
G^{Φ}	User generic intents set
$\mathcal{H}(G^{\varphi_x})$	Homogeneity of G^{φ_x}
$\mathcal{Q}(G^{\Phi})$	Modularity of G^{Φ}
G^{π_t}	User specific intent
G^{Π}	User specific intents set
$\boldsymbol{F}^{\pi_t}$	Embedding of G^{π_t}
$\boldsymbol{F}^{\varphi_x}$	Embedding of G^{φ_x}
$I(\boldsymbol{F}^{\pi_t}, \boldsymbol{F}^{\varphi_x})$	Mutual information between G^{π_t} and G^{φ_x}
r_x	Recommend rating for G^{φ_x}

high-frequency items are mined from the co-occurrence graph $\mathcal{G}$, resulting in a set of generic intents $G^{\Phi} = \{G^{\varphi_1}, G^{\varphi_2}, \ldots, G^{\varphi_p}\}$, p is the total number of generic intents generated from $\mathcal{G}$.

3.3.1 Triangular Structure Capture

The triangle is an important structure in complex graph analysis. Existing study[15] has shown that the triangular structure has the property of internal homogeneity that can effectively reduce the noise information contained in large-scale commodity data, and the property of external heterogeneity that reflects different aspects of user interest. Based on such two properties, the triangle structures are captured from the co-occurrence graph $\mathcal{G}$ as the most basic generic intents. To clearly explain the generation of user generic intents, we use G^{φ_x} to denote a generic intent from G^{Φ}. G^{φ_x} is initialized as $(V^{\varphi_x}, \boldsymbol{E}^{\varphi_x}, \boldsymbol{W}^{\varphi_x})$. $V^{\varphi_x} = \{v_i, v_j, v_k\}$ is a node set with a triangular structure, $\boldsymbol{E}^{\varphi_x}$ is an adjacency matrix of this triangular structure, and $\boldsymbol{W}^{\varphi_x}$ includes the weights of the corresponding nodes.

3.3.2 Adaptive Aggregation

Discovering the basic generic intents based on the triangular structure is a crucial step in exploring the potential intents of a user. However, the triangular structure does not adequately reflect the user's intents and often neglects isolated nodes, leading to disruptions. Inspired by community diffusion methods[33–35], an adaptive aggregation approach is designed here to improve the acquisition process of intents.

To ensure that the aggregated results retain the high degree of relevance that comes from the triangular structure, homogeneity, an important property of intents, is defined as:

$$\mathcal{H}(G^{\varphi_x}) = \sum_{v_i \in V^{\varphi_x}} \sum_{v_j \in V^{\varphi_x}} \frac{w_i w_j sim(v_i, v_j)}{|V^{\varphi_x}|^2}, \tag{1}$$

where w_i and w_j denote the weights of nodes v_i and v_j in $\mathcal{G}$, respectively, $sim(v_i, v_j)$ denotes the similarity between attributes $\boldsymbol{h}_i$ of node v_i and $\boldsymbol{h}_j$ of node v_j. Here the cosine similarity is used, which is calculated as:

$$sim(v_i, v_j) = \frac{\boldsymbol{h}_i \boldsymbol{h}_j}{\|\boldsymbol{h}_i\| \|\boldsymbol{h}_j\|}.$$

Homogeneity measures the degree of similarity among item features within a community. Higher values indicate greater internal consistency and homogeneity. For each basic generic intent $G^{\varphi_x} \in G^{\Phi}$, we choose the node with the maximum homogeneity gain from the node's γ-order neighbors $N_\gamma(G^{\varphi_x})$ for aggregation, which is defined as follows:

$$\arg\max_{v_i \in N_\gamma(G^{\varphi_x})} \{\mathcal{H}(G^{\varphi_x} + \{v_i\}) - \mathcal{H}(G^{\varphi_x})\}. \tag{2}$$

At the same time, the homogeneity metric may lead to popularity bias, where the model may focus on more popular products and lack the exposure of long-tail products. Therefore, we need a metric to assess the overall quality of the user's generic intents. Modularity[35] regularizes long-tailed and popular products by adding popularity penalty scores of product nodes to encourage the model to actively give exposure to cold products, which is defined as follows:

$$\mathcal{Q}(G^{\Phi}) = \frac{1}{\tau} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(e_{ij} - \frac{g_i g_j}{\tau}\right) \delta(i, j), \tag{3}$$

where e_{ij} represents the weight of the edge between nodes v_i and v_j in $\mathcal{G}$, g_i and g_j represent the degrees of nodes v_i and v_j, respectively. τ is the sum of the weights of all the edges in $\boldsymbol{E}$, $\delta(i, j)$ is a function, which equals 1 if v_i and v_j belong to a same intent G^{φ_x}, otherwise 0.

After each aggregation, the difference between the new aggregation scheme $G^{\Phi'}$ and the previous aggregation scheme G^{Φ} is calculated as follows:

$$\Delta\mathcal{Q}(G^{\Phi'}, G^{\Phi}) = \mathcal{Q}(G^{\Phi'}) - \mathcal{Q}(G^{\Phi}). \tag{4}$$

When $\Delta\mathcal{Q}(G^{\Phi'}, G^{\Phi}) > 0$, the aggregation scheme G^{Φ} is updated to be $G^{\Phi'}$, and a new iteration is started until $\Delta\mathcal{Q}(G^{\Phi'}, G^{\Phi}) < 0$. Finally the best aggregation scheme G^{Φ} is obtained. Algorithm 1 shows the whole process of adaptive aggregation for generating

user generic intents G^{Φ}, which inputs triangular structured basic intents and outputs the intents with an arbitrary number of nodes based on the measurement of homogeneity and modularity among nodes.

Algorithm 1. Adaptive Aggregation

Input:
initial features of items $\boldsymbol{H}$;
basic generic intents $G^{\Phi} = \{G^{\varphi_1}, G^{\varphi_2}, \ldots, G^{\varphi_p}\}$;
co-occurrence graph $\mathcal{G}$;
Output:
user generic intents G^{Φ} after adaptive aggregation

Initialize $flag = 1$;
while $flag == 1$ **do**
  for $G^{\varphi_x} \in G^{\Phi}$ **do**
    Calculate $\mathcal{H}(G^{\varphi_x})$ by (1);
    for $v_i \in V^{\varphi_x}$ **do**
      Calculate $\mathcal{H}(G^{\varphi_x} + \{v_i\})$ by (1);
      Calculate $\mathcal{H}(G^{\varphi_x} + \{v_i\}) - \mathcal{H}(G^{\varphi_x})$;
    end for
    Find node with the maximum homogeneity gain by (2) and add it to G^{φ_x};
  end for
  Calculate $\mathcal{Q}(G^{\Phi'})$ by (3);
  Calculate $\Delta\mathcal{Q}(G^{\Phi'}, G^{\Phi})$ by (4);
  if $\Delta\mathcal{Q}(G^{\Phi'}, G^{\Phi}) < 0$ **then**
    $flag$=0;
  else
    $G^{\Phi} \Leftarrow G^{\Phi'}$;
  end if
end while
return G^{Φ}

3.4 User Specific Intents Generation

To capture the complete interest profiles and preference trends of individual users, the items in the user behavior sequence S_t are taken as the initial set of nodes V^{π_t} for each user, and the corresponding subgraph from $\mathcal{G}$ is extracted as the initial user specific intent graph. All initial user specific intents are denoted as $G^{\Pi} = \{G^{\pi_1}, G^{\pi_2}, \ldots, G^{\pi_m}\}$. To clearly explain the generation of user specific intents, we use $G^{\pi_t} = \{V^{\pi_t}, \boldsymbol{E}^{\pi_t}, \boldsymbol{W}^{\pi_t}\}$ to denote an initial user specific intent in G^{Π}. V^{π_t} is the set of all nodes contained in the user sequence S_t. $\boldsymbol{E}^{\pi_t}$ denotes the corresponding connection relations among nodes, and $\boldsymbol{W}^{\pi_t}$ contains the weights of nodes.

Similar to the aggregation for generic intents generation, we adaptively aggregate γ-order neighbors of G^{π_t} to mine neighboring users' preferences and improve the user profile based on social semantic information. By traversing $N_{\gamma}(G^{\pi_t})$, the node with the maximum homogeneity $\mathcal{H}(G^{\pi_t})$ is found as the aggregation node. After each round of aggregation, the modularity improvement $\Delta Q(G^{\Pi'}, G^{\Pi})$ is calculated to evaluate the new aggregation scheme. Finally, the best aggregation scheme G^{Π} is obtained.

3.5 Graph-Level Embedding Learning

Since recommendation algorithms are often applied to different fields such as music, business, and social media, the adoption of comparative learning can help the network adapt to learn the representation related to the specific task. Here, for learning the representation of intents, a set of positive samples and a set of negative samples for each user specific intent G^{π_t} are obtained from the set of generic intents G^{Φ}. Positive samples are randomly selected from G^{Φ} where $V^{\varphi_x} \subset V^{\pi_t}$, resulting in α generic intents. The same number of negative samples are constructed by randomly changing nodes, adding extra edges, and deleting existing edges to destroy the graph structure of the positive samples.

Now the goal is to learn the mutual information relationship between user specific intents and user generic intents, and generate the corresponding embedding for user specific intents and user generic intents, therefore that the mutual information between the positive user generic intents embedding and the corresponding user specific intents embedding is maximized, and the mutual information between the negative user generic intents embedding and the corresponding user specific intents embedding is minimized. Fig.2 clearly shows the framework of this graph-level embedding learning. We use the standard binary cross entropy loss function[26] to train the network, which is defined as follows:

$$\mathcal{L}_{MI} = \sum_{t=1}^{m}\sum_{x=1}^{\alpha} \log(I(\boldsymbol{F}^{\pi_t}, \boldsymbol{F}^{\varphi_x})) + \sum_{t=1}^{m}\sum_{x=1}^{\alpha} \log\left(1 - I(\boldsymbol{F}^{\pi_t}, \boldsymbol{F}^{\hat{\varphi_x}})\right),$$

where $\boldsymbol{F}^{\varphi_x}$ represents the embedding representation corresponding to positive generic intent samples, $\boldsymbol{F}^{\hat{\varphi_x}}$ represents the embedding representation corresponding to negative generic intent samples. I denotes the function for calculating mutual information between user specific intents representation and user generic intents representation as follows:

$$I(\boldsymbol{F}^{\pi_t}, \boldsymbol{F}^{\varphi_x}) = \Theta\left((\boldsymbol{F}^{\pi_t})^{\mathrm{T}} \boldsymbol{B} \boldsymbol{F}^{\varphi_x}\right),$$

$$I(\boldsymbol{F}^{\pi_t}, \boldsymbol{F}^{\hat{\varphi_x}}) = \Theta\left((\boldsymbol{F}^{\pi_t})^{\mathrm{T}} \boldsymbol{B} \boldsymbol{F}^{\hat{\varphi_x}}\right),$$

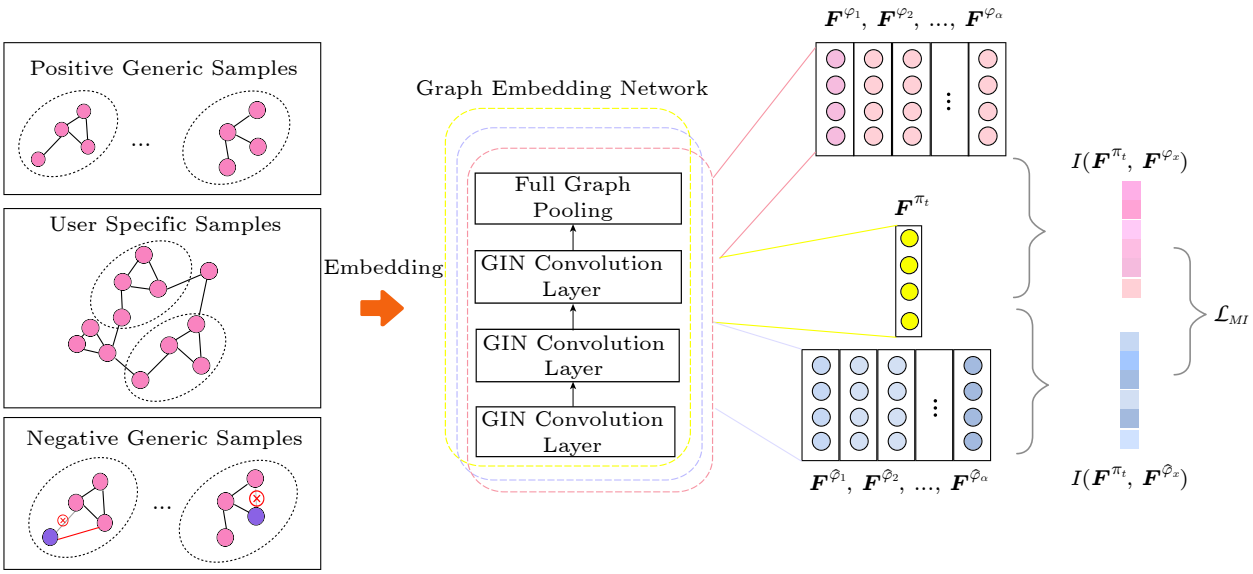

Fig.2. Framework of graph-level embedding learning.

where $\boldsymbol{B}$ is a matrix of learnable rating weight parameters. Θ denotes the sigmoid activation function.

A graph embedding network is constructed to generate user specific intents embedding $\boldsymbol{F}^{\pi_t} \in \mathbb{R}^{1\times z}$, where z is the dimension of embedding. The network consists of graph isomorphism convolutional layers and an average graph pooling function. The network are calculated as follows:

$$\boldsymbol{F}^1 = \sigma(MLP((\boldsymbol{E}^{\pi_t} + (1+\epsilon^1)\cdot \boldsymbol{I})\cdot \boldsymbol{F}^0)), \quad (5)$$

$$\boldsymbol{F}^2 = \sigma(MLP((\boldsymbol{E}^{\pi_t} + (1+\epsilon^2)\cdot \boldsymbol{I})\cdot \boldsymbol{F}^1)), \quad (6)$$

$$\boldsymbol{F}^3 = \sigma(MLP((\boldsymbol{E}^{\pi_t} + (1+\epsilon^3)\cdot \boldsymbol{I})\cdot \boldsymbol{F}^2)), \quad (7)$$

$$\boldsymbol{F}^{\pi_t} = \frac{1}{|V^{\pi_t}|}\boldsymbol{I}^{\mathrm{T}}_{|V^{\pi_t}|}\boldsymbol{F}^3, \quad (8)$$

where $\boldsymbol{F}^0$ is just the features H of attributes of nodes, $\boldsymbol{F}^1$ represents the embedding representation of the graph G^{π_t} at the first layer, and $\boldsymbol{F}^2, \boldsymbol{F}^3$ have the similar meaning. $\boldsymbol{I}$ represents the identity matrix. ϵ^1, ϵ^2, and ϵ^3 are trainable parameters. $\sigma(\cdot)$ denotes the ReLU activation function. MLP is a multilayer perceptron.

Using the same way, we can obtain the embedding $\boldsymbol{F}^{\varphi_x}$ corresponding to the positive generic intent samples and the embedding $\boldsymbol{F}^{\bar{\varphi}_x}$ corresponding to the negative generic intent samples.

3.6 Intent-Based Recommendation

Commonly, the matrix decomposition[7] is applied for recommendation, which decomposes the user-item interaction matrix into multiple low-rank matrices, and scores the item through the decomposition matrix to generate a recommendation list. However, matrix decomposition cannot dynamically capture the user's interest. Based on the concept of intents given in Subsection 3.5, we design an intent-based recommendation strategy here, which calculates the recommendation scores in terms of intents and returns the recommendation list based on our exploration and exploitation strategy.

To be specific, for the i-th recommendation of u_t, the recommendation score r_x is calculated for each alternative user generic intent G^{φ_x}, which is defined as:

$$r_x = I(\boldsymbol{F}^{\boldsymbol{\pi}_t}, \boldsymbol{F}^{\varphi_x}) + \frac{\theta\eta_x}{\beta_x} + \sqrt{\frac{2\ln i}{\beta_x}}, \quad (9)$$

where β_x represents the number of times G^{φ_x} has been selected in the historical recommendations for user u_t. η_x represents the sum of previous rewards from the choice of G^{φ_x} in the historical recommendations, with an initial value of 0. θ is a parameter with a range from 0 to 1, which is used to control the bias of recommendation. When θ decreases, it tends to explore new items different from the previous recommendation to discover potential user preferences, which means the recommendation is more diverse for users. When θ increases, it tends to recommend expected items by exploiting the existing knowledge, which means the recommendation is more stable for the user.

For the top-K recommendation task, the above strategy is employed K rounds. After extracting initial features $\boldsymbol{H}$ for items, and generating user generic intents and the specific intents of u_t, each round involves sorting user generic intents based on r_x and selecting the user generic intent with the maximum r_x. Then an item from this intent is randomly selected as the recommended item y_k, $k \in [1, K]$. η_x and β_x are updated based on user feedback. After completing K rounds, a recommendation list $\boldsymbol{Y}$ with length K is given for u_t. Algorithm 2 shows the whole process of getting a recommendation list for user u_t, which inputs the generic intents and specific intents of users, generates the graph embedding of each intent, calculates the recommendation scores in terms of intents, and finally returns the recommendation list based on an exploration and exploitation strategy.

Algorithm 2. Intent-Based Recommendation

Input:
the features of items $\boldsymbol{H}$;
user generic intents G^{Φ};
user specific intent G^{π_t};
Output:
recommendation list of items $\boldsymbol{Y} = (y_1, y_2, \ldots, y_K)$;

Initialize $\boldsymbol{Y} = ()$;
for $i = 1, 2, 3, \ldots, K$ **do**
  Initialize $\boldsymbol{F}_0$ by the feature matrix of the nodes in G^{π_t};
  Calculate $\boldsymbol{F}^1$ by (5);
  Calculate $\boldsymbol{F}^2$ by (6);
  Calculate $\boldsymbol{F}^3$ by (7);
  Calculate $\boldsymbol{F}^{\pi_t}$ by (8);
  for $G^{\varphi_x} \in G^{\Phi}$ **do**
    Initialize $\mathbf{F}_0$ by the feature matrix of the nodes in G^{φ_x};
    Calculate $\boldsymbol{F}^1$ by (5);
    Calculate $\boldsymbol{F}^2$ by (6);
    Calculate $\boldsymbol{F}^3$ by (7);
    Calculate $\boldsymbol{F}^{\varphi_x}$ by (8);
    Calculate r_x by (9);
  end for
  Sort G^{Φ} based on r_x;
  Select G^{φ_x} with maximum r_x;
  Randomly select an item from G^{φ_x} as y_k;
  if user is interested in y_k **then**
    $\eta_x = \eta_x + 1$;
  end if
  Update $\beta_x = \beta_x + 1$;
  Add y_k into $\boldsymbol{Y}$;
end for
return $\boldsymbol{Y}$;

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

We adopt three public datasets Movielen[②], Yelp2018[③], Amazon2018[④] to verify the proposed recommendation framework IaGEL.

The MovieLens dataset includes 100 835 ratings of 9 741 movies given by 1 010 users. The ratings are on a scale of one to five stars and each user has rated at least 20 movies. In addition to the rating data, the dataset also provides basic attributes of both movies and users. For movies, the attributes include genre, director, actors, release year, and internet movie database (IMDB) link. For users, the attributes include age, gender, and occupation. MovieLen's large size and high-quality rating data, along with rich movie and user attributes, make it a popular dataset for studying recommendation systems.

Yelp is a widely-used business dataset that contains 185 484 review records, 30 099 business records, and 301 757 user records. The review data includes user-generated text, ratings, and timestamps of reviews, with ratings ranging from one to five stars. The business data includes information such as business names, addresses, geolocation, and review counts for multiple cities around the world. The user data includes user ID, username, and the number of reviews.

The Amazon-Book dataset contains 24 531 ratings and information about 29 978 books. The rating data contains a large number of user ratings from Amazon.com, including user ID, rating, rating time, and other information. Each user's rating of a book is an integer value from one to five. The book data contains the basic attributes of the book, including the title, author, publisher, ISBN, publication date, and other information.

The numbers of users, items and records of these three datasets are summarized in Table 2. The density of each dataset is also listed in Table 2. Density is an important evaluation parameter of a business recommendation dataset, reflecting the ratio of the number of interactions already in the dataset relative to the maximum number of possible interactions. A higher density indicates that the dataset is richer in

[②] https://grouplens.org/datasets/movielens/1m/, Sept. 2024.
[③] https://www.yelp.com/dataset, Sept. 2024.
[④] https://nijianmo.github.io/amazon/index.html, Sept. 2024.

Table 2. Statistics of Datasets

Dataset	#Users	#Items	#Records	Density
MovieLens 1M	1 010	9 741	100 835	0.010 20
Yelp2018	301 757	30 099	185 484	0.000 02
Amazon-Book	24 531	29 978	299 258	0.000 27

existing interactions and the interactions between users and items are more intensive. Conversely, a lower density means that the dataset is sparser and the interaction between users and items is more limited.

4.1.2 *Comparison Methods and Evaluation Metrics*

To evaluate the performance of the proposed method, we consider both the state-of-the-art methods of recommendation and graph neural networks.

NGCF[36]: a collaborative filtering approach that captures collaborative signals in higher-order connections by stacking multiple embedded propagation layers.

GraphDA[37]: a collaborative filtering approach that enhances the user-item interaction matrix via top-K sampling.

MCLSR[38]: a graph-based recommendation which learns the representations of users and items through a cross-view contrastive learning paradigm.

Multi-GCCF[39]: a graph-based approach that explicitly incorporates multiple graphs in the embedding learning process.

GIM[14]: a graph-based intention mining method with multi-layered intention diffusion on the co-occurrence relationship graph.

TGIN[15]: a graph interest network that introduces triangle structures in item-item co-occurrence graphs as the basic unit of user interest.

The metrics widely used for recommendation tasks are used to evaluate the performance of all the approaches, including recall of top-K items (Recall@K) and normalized discounted cumulative gain of top-K items (NDCG@K). Recall@K is an accuracy-based metric that reflects how accurately the model predicts positive cases. NDCG@K is a precision-based metric that evaluates the difference between the ranked list and the real interaction list of the user. In the experiments, we set k=10 and 20 to make the generated recommendation list the same length as the regular recommendation list. In the following tables, Recall@K and NDCG@K are shorted as R@K and N@K, respectively.

The recommendation algorithms based on matrix decomposition like NGCF, GraphDA and Multi-GCCF use user-item interaction data to select the K items with the largest rating function to form a recommendation list. The sequence-based recommendation algorithms like IaGEL, MCLSR, GIM and TGIN use a sequence of the user's historical behaviors to make K consecutive recommendations to form a recommendation list. We evaluate the recommendation performance using Recall@K and NDCG@K based on the number of products contained in the final recommendation list.

4.1.3 *Implementation Details*

In the proposed IaGEL, the dimension of embedding representation is set to 64 and the learning rate is set to 1×10^{-4}. To ensure fairness, we follow the officially reported hyper-parameter settings with the best model performance for these compared approaches. For NGCF, the number of layers in the graph encoder is set to three and the embedding size is fixed to 64. For GraphDA, the number of neighbors from different semantics is set to 5. For MCLSR, the depth of GNN layers is set to 2. For Multi-GCCF, two BiparGCN layers are used and the neighborhood sampling size is set to 15. For GIM, five-layer full-connection perceptron is adopted as the forward network. For TGIN, the window size is set to 3 and the embedding size is set to 18.

4.2 Comparison with State-of-the-Art Methods

We randomly choose 60% of the data as the training set, 20% as the test set and the remaining data as the validation set for the experiments. The comparison of the proposed IaGEL and the state-of-the-art methods is given in Table 3 (bold represents the best performing method). It can be seen that IaGEL outperforms these advanced approaches on these three different datasets. For the Moivelen dataset, the recall of IaGEL is 0.67% to 2.40% higher than that of the other methods, and the NDCG of IaGEL is 1.01% to 3.00% higher than those of the other methods. For the Yelp2018 dataset, the recall of IaGEL is 0.49% to 2.21% higher than those of the other methods, and the NDCG of IaGEL is 1.14% to 3.50% higher than that of the other methods. For the Amazon dataset, the recall of IaGEL is 0.32% to 2.61% higher than that of the others, and the NDCG of IaGEL is 0.39%

Table 3. Results Obtained with Different Methods in MovieLens, Yelp, and Amazon Datasets

Model	MovieLens (%)				Yelp2018 (%)				Amazon (%)			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
NGCF[36]	20.12	39.53	23.83	43.32	4.62	8.53	5.13	9.25	5.12	5.68	5.69	7.13
GraphDA[37]	20.92	40.62	24.33	43.97	5.84	9.12	6.49	9.93	6.34	6.77	7.26	7.56
MCLSR[38]	21.53	40.93	24.31	44.67	6.22	9.91	6.36	9.62	6.68	6.93	7.46	8.43
Multi-GCCF[39]	20.35	39.87	23.72	43.34	6.04	9.55	6.31	9.74	6.29	6.21	7.34	7.65
GIM[14]	20.84	40.35	23.67	43.17	5.95	9.38	6.22	9.92	6.53	6.84	7.26	8.13
TGIN[15]	21.95	41.52	24.12	43.64	6.34	10.07	6.78	10.14	6.87	7.05	7.81	8.42
IaGEL	**22.62**	**42.53**	**25.74**	**45.93**	**6.83**	**11.21**	**7.32**	**12.75**	**7.73**	**8.24**	**8.13**	**8.82**

to 2.56% higher than that of the others. An interesting thing should be pointed out: IaGEL has significant advantages for sparse datasets, such as Yelp2018 whose density is only 0.000 02. The reason is that the IaGEL mines the potential semantic information in the neighborhood space to complement the missing contextual information of the sparse graph.

IaGEL has obvious performance improvement compared with inductive collaborative filtering like NGCF, GraphDA, and graph-based recommendation like MCLSR and Multi-GCCF. This is because IaGEL not only captures the relationship between users and items through structural and temporal correlations, but also mines users' real purposes and higher-order semantic features through intents, which has more powerful generalization ability and interpretability. Due to the dynamic recommendation weighting strategy and adaptive aggregation strategy, IaGEL has a significant performance improvement compared with intent-based recommendation like GIM and TGIN. In IaGEL, intents generation is offline operations, which does not require iterative training. Therefore the complexity of IaGEL depends on the embedding learning which needs to specify the number of sampled positive and negative general intents for each user specific intent. Compared with another graph-based recommendations algorithm[9] which depends on the number of nodes, intents tend to be more granular and less numerous. Using intents as recommendation units has better time complexity.

4.3 Evaluation of the Capture of Preference Changes

Dynamic user preference changes are a frequent problem for recommendation systems. Traditional recommendation algorithms are often modeled based on user-item interaction matrices, which often fail to capture dynamic user preferences over time. To demonstrate that IaGEL has the ability of capturing dynamic user preferences over time, we do experiments on the Movielen dataset whose density is larger than those of the other two datasets. Yelp and Amazon datasets are relatively sparse and lack a sufficient number of long user behavior sequences.

The first 25% of the data is used as the initial data to train IaGEL, and its performance metrics are recorded. Then, the data was incrementally increased by subsequent 25%, and each time when new data is added, IaGEL will be retrained. By this way, the user's dynamic preferences are simulated and evaluated. IaGEL is compared with MCLSR and TGIN which also extract intents or use contrastive learning. From Fig.3, we can see that when the user behavior sequence is short, the performance improvement of IaGEL is not obvious compared with these two methods, only 0.34% and 0.94% higher at Recall@20, only 0.14% and 0.54% higher at NDCG@20. However, as the user behavior sequence becomes longer, other models gradually fail to capture the long-term trend of user preferences, while IaGEL captures the evolution of user preferences through intentions. For the complete user sequence, the Recall@20 of IaGEL is 1.78% and 2.94% higher than that of MCLSR and TGIN, respectively. The NDCG@20 of IaGEL is 2.78% and 2.91% higher than that of MCLSR and TGIN, respectively.

4.4 Parameter Analysis

4.4.1 *Evaluation of the Order of Adaptive Aggregation*

The order γ tells the search scope of adaptive aggregation for basic intents. When $\gamma = 0$, the intents are just the pure triangular structure. When γ is larger than 0, the intents are generated from the basic triangular structure and basic triangular structure's γ order neighbors. Intuitively, the order of neighbors affects the performance of the relevant intents embedding. As shown in Table 4, we can see that the intents generated based on γ order neighbors can improve Recall@20 and NDCG@20 compared with the

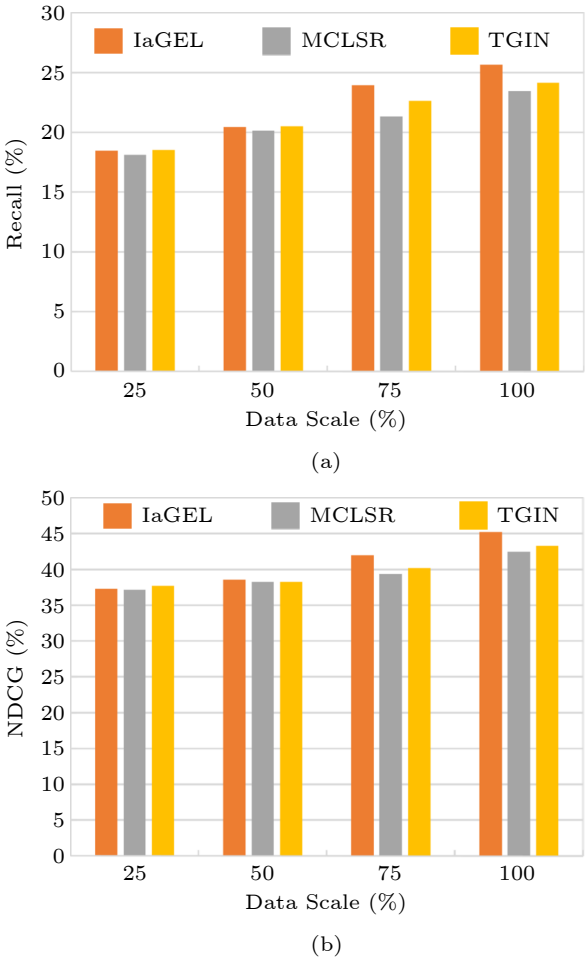

Fig.3. Performance of user dynamic preference capture in the Movielen dataset. (a) Recall. (b) NDCG.

Table 4. Comparison of Different Orders of Intents' Neighbor

γ	MovieLens (%)		Yelp2018 (%)		Amazon (%)	
	R@20	N@20	R@20	N@20	R@20	N@20
0	19.32	36.43	6.42	10.32	7.79	8.12
1	24.46	43.27	7.12	12.51	8.04	8.62
2	**25.74**	**45.93**	**7.32**	**12.75**	**8.13**	**8.82**
3	23.32	41.45	6.72	11.73	7.63	7.83

pure triangular structural intents. When the order is 2, Recall@20 and NDCG@20 are much higher: Recall@20 and NDCG@20 are 6.42% and 9.5% higher than the basic intents in the MovieLens dataset. On the two sparse datasets Yelp2018 and Amazon, Recall@20 and NDCG@20 of $\gamma = 2$ are also higher than those of $\gamma = 0$. But when the order is larger than 2, Recall@20 and NDCG@20 are decreased significantly. The reason may be that when the search scope is too large, the integration will bring too much irrelevant nodes in the intents, resulting in semantic confusion and information redundancy. Therefore we fix the order of aggregation to 2 in the experiments.

4.4.2 *Evaluation of the Number of Intent Samples*

The complexity and accuracy of IaGEL are also influenced by the number of sampled intents. As shown in Fig.4, we can see that it needs more epochs to get network convergence when the number of samples is smaller than 20 or larger than 50. As α increases from 10 to 30, the final performance improves by 4.84% in NDCG@20 and 3.36% in Recall@20 for the Movielen dataset, while $\alpha = 30$ gives the highest NDCG and recall values at 175 epochs. However, when α exceeds 30, the network takes up more space, and the final performance decreases in terms of NDCG and recall. This may be due to the increase of α leading to the overfitting of the network. Therefore in the experiments, we select 30 positive samples and 30 negative samples for each specific intent. For the Yelp2018 and Amazon datasets, it has the similar phenomena, therefore $\alpha = 30$ is also used in these two datasets.

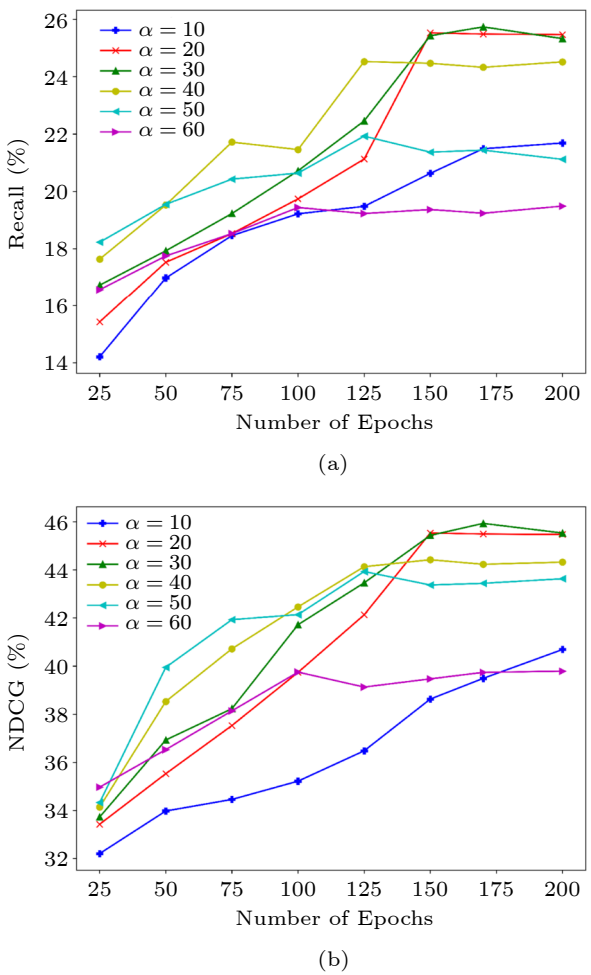

Fig.4. Performance of different numbers of samples in the Movielen dataset. (a) Recall. (b) NDCG.

4.4.3 Evaluation of Different Graph Embedding Operators

For graph-level embedding learning in IaGEL, there are a lot of choices for graph embedding. Here, we try to compare several commonly used graph embedding networks to discuss which one is better for this task. Table 5 shows the performance of these graph embedding networks for these three dataset. We can see that Graph Convolutional Network (GCN)[27] suffers from the complexity of commodity data, which makes it difficult to fully explore the relationships and features on co-occurrence graphs. GraphSAGE[29] uses a fixed approach to aggregate feature information from neighboring nodes and cannot fully identify complex associations between different levels of intents. Graph Attention Network (GAT)[28] has good performance on recommendation tasks, but the attention mechanism is computationally intensive and has a much higher memory and time overhead than other graph embedding models when dealing with large datasets. Graph Isomorphism Network (GIN)[30] learns global features by recursively learning local subgraphs, which is suitable for handling graph data with different structures, and is excellent at the task of learning mutual information relationships between different levels of intents.

Table 5. Comparison of Different Graph Embedding Operators

Operator	MovieLens (%)		Yelp2018 (%)		Amazon (%)	
	R@20	N@20	R@20	N@20	R@20	N@20
GraphSAGE[29]	20.87	41.34	6.14	9.87	6.84	7.53
GCN[27]	21.35	40.57	6.22	10.02	7.35	8.29
GAT[28]	25.42	42.21	6.24	10.17	8.03	8.65
GIN[30]	**25.74**	**45.93**	**7.32**	**12.75**	**8.13**	**8.82**

4.4.4 Evaluation of Intent-Based Recommendation Strategy

In the proposed intent-based recommendation strategy, there is a factor θ to control recommendation bias. It is a trade-off between the stability and the diversity. If the recommendation is more stable, the recommended items are more uniform, which means that the changes of users' preferences are less reflected. Therefore, we quantify the stability and diversity to show that the recommendation bias can be changed by adjusting θ. The stability and diversity are quantified as follows:

$$Stability = \frac{N-1}{\sum_{t=1}^{N}(o_t - \bar{o})^2},$$

$$Diversity = \frac{1}{K}\sum_{i=1}^{K}\sum_{j=1}^{i-1}\frac{(1 - cos(c_i, c_j))}{i-1},$$

where the stability is obtained by calculating the sample variance of the accuracy of the N recommendations and taking the reciprocal as the stability value. o_t is the accuracy of the t-th recommendation, and $\bar{o}$ is the average accuracy of N recommendations and a smaller value represents more stable. The diversity is calculated by averaging the similarity between the recommended items and all historical recommended items, and $cos(c_i, c_j)$ calculates the cosine similarity between the i-th recommended item and the j-th recommended item by using their attributes' features.

As shown in Fig.5, we can see that the stability of

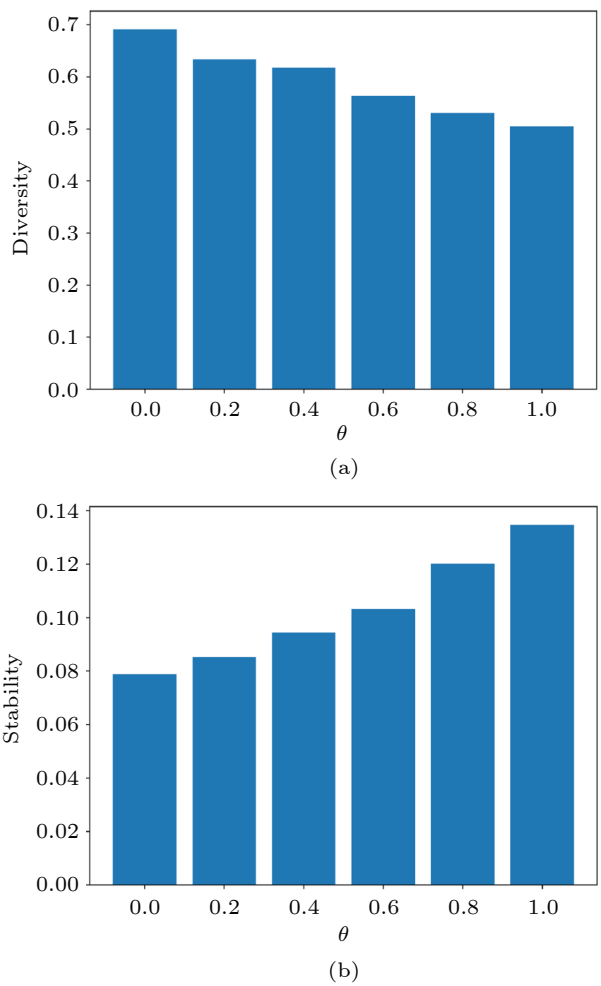

Fig.5. Performance of different values of trade-off factor θ in the Movielen dataset. (a) Diversity. (b) Stability.

recommendation is increased and the diversity of recommendation is decreased as θ increases. For the common recommendation, θ can be selected as 0.5. For the situation of cold start, θ can be selected as 0.2.

5 Conclusions

In this paper, a novel recommendation framework named IaGEL was proposed for solving the problem of dynamic user preference changes and potential intents recognition. Its accuracy exceeds the existing advanced recommendation algorithms. The framework explores user generic intents and specific intents to capture users' potential interest. Furthermore, an intent-based recommendation strategy was designed to accommodate the changes in users' preferences. Experiments not only tested the important parameters of the proposed framework but also compared its performance with several advanced recommendation approaches. What is more, how to make a trade-off between the stability and the diversity of recommendations was also analyzed. It is also a future research direction for solving the issue of cold start in recommendation.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Yang M, Li Z, Zhou M, Liu J, Irwin K. HICf: Hyperbolic informative collaborative filtering. In *Proc. the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2022, pp.2212–2221. DOI: 10.1145/3534678.3539475.

[2] Xia L, Huang C, Zhang C. Self-supervised hypergraph transformer for recommender systems. In *Proc. the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2022, pp.2100–2109. DOI: 10.1145/3534678.3539473.

[3] Dong Y, Chawla N V, Swami A. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proc. the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2017, pp.135–144. DOI: 10.1145/3097983.3098036.

[4] Fu T Y, Lee W C, Lei Z. HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proc. the 2017 ACM Conference on Information and Knowledge Management*, Nov. 2017, pp.1797–1806. DOI: 10.1145/3132847.3132953.

[5] Feng Y, Lv F, Hu B, Sun F, Kuang K, Liu Y, Liu Q, Ou W. MTBRN: Multiplex target-behavior relation enhanced network for click-through rate prediction. In *Proc. the 29th ACM International Conference on Information & Knowledge Management*, Oct. 2020, pp.2421–2428. DOI: 10.1145/3340531.3412729.

[6] Wu K, Bian W, Chan Z, Ren L, Xiang S, Han S G, Deng H, Zheng B. Adversarial gradient driven exploration for deep click-through rate prediction. In *Proc. the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2022, pp.2050–2058. DOI: 10.1145/3534678.3539461.

[7] Wang G, Zhong T, Xu X, Zhang K, Zhou F, Wang Y. Vector-quantized autoencoder with copula for collaborative filtering. In *Proc. the 30th ACM International Conference on Information & Knowledge Management*, Nov. 2021, pp.3458–3462. DOI: 10.1145/3459637.3482216.

[8] He X, Liao L, Zhang H, Nie L, Hu X, Chua T S. Neural collaborative filtering. In *Proc. the 26th International Conference on World Wide Web*, Apr. 2017, pp.173–182. DOI: 10.1145/3038912.3052569.

[9] Wang Z, Zhao H, Shi C. Profiling the design space for graph neural networks based collaborative filtering. In *Proc. the 15th ACM International Conference on Web Search and Data Mining*, Feb. 2022, pp.1109–1119. DOI: 10.1145/3488560.3498520.

[10] Gong J, Wang S, Wang J, Feng W, Hao P, Tang J, Yu P S. Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view. In *Proc. the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2020, pp.79–88. DOI: 10.1145/3397271.3401057.

[11] Yang Y, Guan Z, Li J, Zhao W, Cui J, Wang Q. Interpretable and efficient heterogeneous graph convolutional network. *IEEE Trans. Knowledge and Data Engineering*, 2023, 35(2): 1637–1650. DOI: 10.1109/TKDE.2021.3101356.

[12] Fu X, Zhang J, Meng Z, King I. MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proc. the 29th International Conference on World Wide Web*, Apr. 2020, pp.2331–2341. DOI: 10.1145/3366423.3380297.

[13] Hao P, Qian Z, Wang S, Bai C. Community aware graph embedding learning for item recommendation. *World Wide Web*, 2023, 26(6): 4093–4108. DOI: 10.1007/s11280-023-01224-5.

[14] Li F, Chen Z, Wang P, Ren Y, Zhang D, Zhu X. Graph intention network for click-through rate prediction in sponsored search. In *Proc. the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2019, pp.961–964. DOI: 10.1145/3331184.3331283.

[15] Jiang W, Jiao Y, Wang Q, Liang C, Guo L, Zhang Y, Sun Z, Xiong Y, Zhu Y. Triangle graph interest network for click-through rate prediction. In *Proc. the 15th ACM International Conference on Web Search and Data Mining*, Feb. 2022, pp.401–409. DOI: 10.1145/3488560.

3498458.

[16] Chen T, Yin H, Chen H, Yan R, Nguyen Q V H, Li X. AIR: Attentional intention-aware recommender systems. In *Proc. the 35th IEEE International Conference on Data Engineering*, Apr. 2019, pp.304–315. DOI: 10.1109/ICDE.2019.00035.

[17] Yang Y, Huang C, Xia L, Liang Y, Yu Y, Li C. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proc. the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2022, pp.2263–2274. DOI: 10.1145/3534678.3539342.

[18] Wang R, Yang N, Yu P S. Learning aspect-level complementarity for intent-aware complementary recommendation. *Knowledge-Based Systems*, 2022, 258:109936. DOI: 10.1016/j.knosys.2022.109936.

[19] Guo J, Yang Y, Song X, Zhang Y, Wang Y, Bai J, Zhang Y. Learning multi-granularity consecutive user intent unit for session-based recommendation. In *Proc. the 15th ACM International Conference on Web Search and Data Mining*, Feb. 2022, pp.343–352. DOI: 10.1145/3488560.3498524.

[20] Fan S, Zhu J, Han X, Shi C, Hu L, Ma B, Li Y. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proc. the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug. 2019, pp.2478–2486. DOI: 10.1145/3292500.3330673.

[21] Li J, Sun P, Wang Z, Ma W, Li Y, Zhang M, Feng Z. Intent-aware ranking ensemble for personalized recommendation. In *Proc. the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2023, pp.1004–1013. DOI: 10.1145/3539618.3591702.

[22] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations. In *Proc. the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2014, pp.701–710. DOI: 10.1145/2623330.2623732.

[23] Grover A, Leskovec J. Node2vec: Scalable feature learning for networks. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp.855–864. DOI: 10.1145/2939672.2939754.

[24] Wang J, Huang P, Zhao H, Zhang Z, Zhao B, Lee D L. Billion-scale commodity embedding for E-commerce recommendation in Alibaba. In *Proc. the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul. 2018, pp.839–848. DOI: 10.1145/3219819.3219869.

[25] Kipf T N, Welling M. Variational graph auto-encoders. arXiv: 1611.07308, 2016. https://arxiv.org/abs/1611.07308, Sept. 2024.

[26] Velickovic P, Fedus W, Hamilton W L, Liò P, Bengio Y, Hjelm R D. Deep graph infomax. In *Proc. the 7th International Conference on Learning Representations*, May 2019.

[27] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In *Proc. the 5th International Conference on Learning Representations*, Apr. 2017.

[28] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. arXiv: 1710.10903, 2017. https://arxiv.org/abs/1710.10903, Sept. 2024.

[29] Hamilton W L, Ying Z, Leskovec J. Inductive representation learning on large graphs. In *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017. DOI: 10.5555/3294771.3294869.

[30] Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? In *Proc. the 7th International Conference on Learning Representations*, May 2019.

[31] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In *Proc. the 1st International Conference on Learning Representations*, May 2013.

[32] Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: Bringing order to the web. Technical Report, Stanford InfoLab, 1999. http://ilpubs.stanford.edu:8090/422/, Sept. 2024.

[33] Park N, Kan A, Dong X L, Zhao T, Faloutsos C. Estimating node importance in knowledge graphs using graph neural networks. In *Proc. the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug. 2019, pp.596–606. DOI: 10.1145/3292500.3330855.

[34] Liu Y, Wang Q, Wang X, Zhang F, Geng L, Wu J, Xiao Z. Community enhanced graph convolutional networks. *Pattern Recognition Letters*, 2020, 138:462–468. DOI: 10.1016/j.patrec.2020.08.015.

[35] Newman M E J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 2006, 103(23): 8577–8582. DOI: 10.1073/pnas.0601602103.

[36] Wang X, He X, Wang M, Feng F, Chua T S. Neural graph collaborative filtering. In *Proc. the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2019, pp.165–174. DOI: 10.1145/3331184.3331267.

[37] Fan Z, Xu K, Dong Z, Peng H, Zhang J, Yu P S. Graph collaborative signals denoising and augmentation for recommendation. In *Proc. the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2023, pp.2037–2041. DOI: 10.1145/3539618.3591994.

[38] Wang Z, Liu H, Wei W, Hu Y, Mao X L, He S, Fang R, Chen D. Multi-level contrastive learning framework for sequential recommendation. In *Proc. the 31st ACM International Conference on Information & Knowledge Management*, Oct. 2022, pp.2098–2107. DOI: 10.1145/3511808.3557404.

[39] Sun J, Zhang Y, Ma C, Coates M, Guo H, Tang R, He X. Multi-graph convolution collaborative filtering. In *Proc. the 2019 IEEE International Conference on Data Mining*, Nov. 2019, pp.1306–1311. DOI: 10.1109/ICDM.2019.00165.

Peng-Yi Hao received her Ph.D. degree in computer science from Graduate School of Information, Production and Systems, Waseda University, Tokyo, in 2013. She is currently an associate professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. Her current research interests include multimedia analysis and graph neural networks.

Si-Hao Liu is currently an undergraduate student of the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. His research interests include data mining and graph neural networks.

Cong Bai received his B.E. degree from Shandong University, Jinan, in 2003, his M.E. degree from Shanghai University, Shanghai, in 2009, and his Ph.D. degree from the National Institute of Applied Sciences, Rennes, in 2013. He is currently a professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. His research interests include computer vision and multimedia processing.