Enhanced Multi-Relationships Integration Graph Convolutional Network for Inferring Substitutable and Complementary Items

Huajie Chen[1*], Jiyuan He[1*], Weisheng Xu[1,2], Tao Feng[1], Ming Liu[1], Tianyu Song[2], Runfeng Yao[2], Yuanyuan Qiao[2†]

[1]Meituan Group, Beijing, China
[2] Beijing University of Posts and Telecommunications, Beijing, China
{chenhuajie02, hejiyuan, fengtao02, liuming04}@meituan.com
{xuweisheng, songtianyu, yaorunfeng, yyqiao}@bupt.edu.cn

Abstract

Understanding the relationships between items can improve the accuracy and interpretability of recommender systems. Among these relationships, the substitute and complement relationships attract the most attention in e-commerce platforms. The substitutable items are interchangeable and might be compared with each other before purchasing, while the complementary items are used in conjunction and are usually bought together with the query item. In this paper, we focus on two issues of inferring the substitutable and complementary items: 1) how to model their mutual influence to improve the performance of downstream tasks, 2) how to further discriminate them by considering the strength of relationship for different item pairs. We propose a novel multi-task learning framework named Enhanced Multi-Relationships Integration Graph Convolutional Network (EMRIGCN). We regard the relationship inference task as a link prediction task in heterogeneous graph with different types of edges between nodes (items). To model the mutual influence between substitute and complement, EMRIGCN adopts a two-level integration module, i.e., feature and structure integration, based on experts sharing mechanism during message passing. To obtain the strength of relationship for item pairs, we build an auxiliary loss function to further increase or decrease the distances between embeddings of items with weak or strong relation in latent space. Extensive experiments on both public and industrial datasets prove that EMRIGCN significantly outperforms the state-of-the-art solutions. We also conducted A/B tests on real world recommender systems of Meituan Maicai, an online supermarket platform in China, and obtained 15.3% improvement on VBR and 15.34% improvement on RPM.

Introduction

Recommender systems play an increasingly crucial role in modern online applications. It helps users to find the items that they may be interested in from a huge collection set. Besides modeling the interaction between users and items, understanding the relationships between items also boosts the development of recommender systems for it can bring better user experience and trigger more revenue (Yang et al. 2022; Hao et al. 2020). There are various relationships between items, among them, substitute and complement are most popular and valuable, especially in e-commerce platforms (e.g. Amazon, Taobao) (McAuley, Pandey, and Leskovec 2015). Substitutable items have similar attributes which are often viewed and compared by users. Complementary items are used in conjunction which are usually bought together with the query item. These different relationships should be considered while generating candidates for recommender system. Figure 1 illustrates a toy example on our online supermarket platform. When a user is browsing bread, in order to offer more choices, it is reasonable to recommend other breads belonged to different brands or flavors (*substitutable items*). After the user has added bread to shopping cart, in order to explore new needs, it is more appropriate to recommend milk or cheese (*complementary items*). Understanding the relationships between items can improve the accuracy and interpretability of recommender systems.

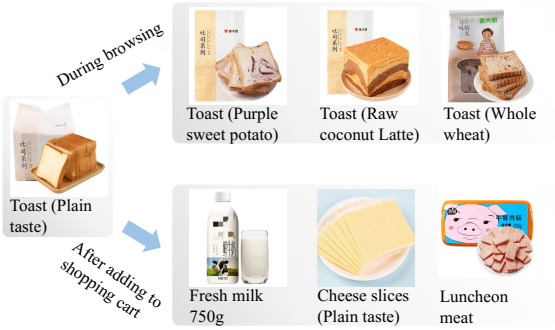

Figure 1: An example of item recommendation on Meituan Maicai. Given a query item, plain taste toast for example, its substitutable items are recommended during browsing, and its complementary items are recommended after adding it to shopping cart.

In early stage, most studies extract features of items from reviews and follow logistic regression (McAuley, Pandey, and Leskovec 2015) or full Bayesian (Rakesh et al. 2019) to predict substitute and complement relationships. Recently, Graph Neural Network (GNN) has shown superior performance on graph analysis and is widely applied in many ap-

*These authors contributed equally.
†Corresponding author is Yuanyuan Qiao.

plications (Zhou et al. 2020). Items and their relationships can be modeled by graph naturally and the multi-hop information is helpful for inferring relationships between items. IRGNN (Liu et al. 2021) designs an edge relational predict network and incorporates multi-hop relationships between items by recursively updating node embeddings using the messages from their neighbors. SPEM (Zhang et al. 2019b) uses complement relationship graph to model the substitutes between items by preserving the second-order proximity, negative first-order proximity and semantic similarity. Similarly, substitute information can also be used to predict complement relationship. DecGCN (Liu et al. 2020) proposes a knowledge integration scheme which considers the mutual influence between different graph structures and different item semantics. It uses a co-attention mechanism to attentively integrate both substitute and complement neighbors to formulate item embeddings, then adopts transfer mechanism to combine two types of semantics embeddings in the final stage. DecGCN works effectively when items have both the substitute and complement neighbors. If an item only has one type of neighbors (substitute or complement), which is common in practice, the integration scheme of DecGCN will be ineffective.

Besides, most existing literature regards relationship prediction as a classification task and only outputs the multi-classification probability distribution by adopting max-margin loss (Hao et al. 2020; Zheng et al. 2021; Yan et al. 2022) or cross-entropy loss(Liu et al. 2020; Zhang et al. 2019a; Liu et al. 2021; Yang et al. 2022). However, they ignored that the strength of relationships are diverse between different related items. For example, the complement relationship between bread and milk is stronger than that between bread and luncheon meat according to the users' order behavior (The number of people buying bread and milk together is 10.5 times than the number of people buying bread and luncheon meat). To bring better shopping experience, it's reasonable to give priority to recommend items that have strong relationship with the query item. Therefore, the strength of relationship is important for recommender systems.

Based on these observations, we raise two research issues:

- How to efficiently build model to learn the mutual influence between different types of relationships, especially for the item that only has one type of neighbors?
- In addition to predicting the relationship between items, how to further study the strength of relationship for different item pairs?

In this paper, we propose a novel framework named Enhanced Multi-Relationships Integration Graph Convolutional Network (EMRIGCN). We build a heterogeneous graph where the items are represented as nodes and the relationships between items are represented as edges. We formulate multiple relationships prediction task as a link prediction task on the graph. In order to address the first issue, we design a two-level relationship integration module based on a Graph Convolutional Network (GCN). In low-level, we build a feature integration module to discovery the association between different relationships at the initial stage. In high-level, we consider the contribution of different types of neighbors and build a structure integration module during convolutional process. By this, we integrate the information of both relationships during the whole message propagation process in GNN, including feature generation and message aggregation. These integration modules employ Multi-gate Mixture-of-Experts (MMoE) (Ma et al. 2018) mechanism which can automatically capture relationship shared information and relationship specific information. Our model can generate a robust item embedding representation by transferring knowledge between different relationships even though an item only has one type of neighbors. In order to address the second issue, we build an enhanced auxiliary loss function to further distinguish the distances between embeddings of items with weak or strong relation in latent space.

The major contributions are summarized as follows:

- We focus on the mutual influence between substitute and complement relationships, and propose a two-level integration module based on MMoE to automatically capture relationship shared information and relationship specific information on a heterogeneous graph.
- In order to further distinguish the strength of relationship, we build an auxiliary loss function based on triplet loss focusing on the diversity of different item pairs.
- Extensive experiments on five datasets show that EMRIGCN outperforms the state-of-the-art baseline methods, especially for cold-start items. We also applied EMRIGCN on our online recommender systems of Meituan Maicai. The A/B test results demonstrate the effectiveness of our method.

Related Work

GNN in Recommender System

The user-user, user-item and item-item relationships of recommender system can be naturally represented in the form of graphs (McAuley, Pandey, and Leskovec 2015; Rakesh et al. 2019; Liu et al. 2020; Schlichtkrull et al. 2018; Ying et al. 2018). GNN has superiority in graph representation learning (Wu et al. 2020c). In the real word recommender system scenario, users and items usually have multiple types of relationships (Wu et al. 2020b). These multiple relationships can be modeled as a heterogeneous graph with GNN (Liu et al. 2020; Zheng et al. 2021; Fan et al. 2019; Wu et al. 2019a,b; Yu et al. 2021; Wu et al. 2020a; Liu et al. 2021).

Some of these work try to better learn the heterogeneous relationships to mitigate data sparsity issue and cold start problem (Wu et al. 2019a,b; Yu et al. 2021; Wu et al. 2020a; Liu et al. 2021) since different types of relations can leverage information from each other. Two strategies are considered to represent and fuse different information sources in heterogeneous graph (Gao et al. 2022; Wu et al. 2020b). One is to integrate the representation learnt from distinct networks. DANSER (Wu et al. 2019b) used two dual the Graph Attention Networks (GATs) to capture user dynamic/static preference and item dynamic/static attribute respectively. To capture the change of user's interests in the recursive process, DiffNet (Wu et al. 2019a) leveraged the graph convolution

operation for the recursive social diffusion in the social networks, which works even the user (item) attributes or the social network structure is not available. Another strategy is to use one unified network to learn representations from different relationships. DiffNet++ (Wu et al. 2020a) modelled both the higher-order social and interest network structure in a unified model to alleviate the data sparsity issue. R-GCN (Schlichtkrull et al. 2018) used multi-relational GCN model as entity encoder and factorization mechanism as decoder for link prediction tasks.

Inferring Substitutable and Complementary Items

Item relationships play a significant role in helping users make wise purchase decisions of e-commerce platforms (Liu et al. 2021; Yang et al. 2022). The existing researches in this area can be generally divided into three stages. Firstly, researchers conduct probability statistics through user purchase and browsing data (Linden, Smith, and York 2003; Zheng et al. 2009). The probability that a pair of items will be purchased and viewed together is calculated to predict the item relationships. In the second stage, the approaches characterize the semantics of complement and substitute relationships by learning item content features in different methods, such as Latent Dirichlet Allocation (LDA) (McAuley, Pandey, and Leskovec 2015) and variational autoencoders (VAE) (Rakesh et al. 2019). Then the researches focus on learning item representation through the relationship constraints between items, such as PMSC (Wang et al. 2018) and SPEM (Zhang et al. 2019b).

Recently, GNNs shows promising results on item relationship inferring (Hao et al. 2020; Zheng et al. 2021). P-companion (Hao et al. 2020) built a homogeneous graph and used GAT to aggregate the latent representations for diversified complementary product recommendation. HetSAGE (Zheng et al. 2021) focused on substitute product recommendation and proposed neighbor-Sim attention mechanism to make the model suitable for large-scale heterogeneous graphs. IRGNN (Liu et al. 2021) designed an edge relational network and aggregated information over the multi-hop neighbor of nodes with a GNN-based framework to incorporate edge features for predicting item relationships. DecGCN (Liu et al. 2020) used structural integration and semantic integration to learn the mutual influence between different relationships.

Method

In this section, we introduce the Enhanced Multi-Relationships Integration Graph Convolutional Network (EMRIGCN) in detail. Firstly, we review the basic GCN for single relationship prediction. Then we propose a multi-relationships integration framework to build connection between substitute and complement relationships. Finally, we design an auxiliary loss to learn the strength of relationship.

Problem Definition

Following previous studies (Liu et al. 2020), we regard the substitute and complement relationship inference task as a link prediction task on graph. The multi-relationship graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, v_2, \cdots, v_{|\mathcal{V}|}\}$ is the node(items) set with size $|\mathcal{V}|$ and $\mathcal{E} = \mathcal{E}^s \cup \mathcal{E}^c$ is the edge set containing both substitute and complement relationships. $\mathcal{X} = \{x_1, x_2, \cdots, x_{|\mathcal{V}|} | x_i \in \mathbb{R}^{1\times nd}\}$ is the node feature which is generated by concatenating d-dimensional dense representation in each feature field (e.g., category, brand. n is the number of fields) (Guo et al. 2017; Song et al. 2019). Our goal is to use node feature $\mathcal{X}$ and the link information $\mathcal{E}$ to generate the complement embeddings $Z^c = \{z_1^c, z_2^c, \cdots, z_{|\mathcal{V}|}^c | z_i^c \in \mathbb{R}^{1\times d}\}$ and the substitute embeddings $Z^s = \{z_1^s, z_2^s, \cdots, z_{|\mathcal{V}|}^s | z_i^s \in \mathbb{R}^{1\times d}\}$ for each item simultaneously. The embeddings can be applied in relationship prediction and other downstream tasks (e.g., retrieval in recommender system).

Review of BaseModel

Our framework is an embedding-based method which can be combined with any GCN approaches. We choose PinSage (Ying et al. 2018) as base model, since it is a data-efficient GCN algorithm and has been widely used in recommendation applications (Wu et al. 2020b). For simplify, we only introduce how to learn the node embeddings Z^c with single relationship graph $\mathcal{G}^c = (\mathcal{V}^c, \mathcal{E}^c)$. The node embeddings Z^s can be learned by the same way.

Firstly, the base node representation $h_i^{c:0}$ at the initial stage is obtained from the input feature.

$$h_i^{c:0} = f^c(x_i), \tag{1}$$

where f^c is a multilayer perceptron.

Then, a multi-layer localized convolution operation is adopted to aggregate message from neighbors. The hidden state of node v_i in l-layer is defined as:

$$\hat{h}_i^{c:l} = \mathrm{agg}(\{h_j^{c:l-1} | j \in \mathcal{N}_i^c\}), \tag{2}$$

$$h_i^{c:l} = \sigma(W^{c:l}[h_i^{c:l-1} : \hat{h}_i^{c:l}]), \tag{3}$$

where agg is aggregation function (e.g., max-pooling or avg-pooling operations), $\mathcal{N}_i^c$ are the complement neighbors of v_i which are sampled by a random walk strategy with multi-hop connections, σ is an activation function (e.g., ReLU), $W^{c:l}$ represents the weight matrix in l-layer for the complement and ":" is concatenate operator. Finally, the hidden state in the last layer is regarded as node embedding z_i^c.

Pinsage adopts max-margin ranking loss during training stage. The loss function is defined as:

$$\mathcal{L}^c(z_i^c) = max\{0, s(z_i^c, z_{neg}^c) - s(z_i^c, z_{pos}^c) + \Delta\}, \tag{4}$$

where Δ is the margin hyper parameter and s is the scoring function to evaluate the relationship between different nodes by using specific embedding. The basic idea is that we want to maximize the score of positive examples and make the score of negative examples smaller than that of positive examples by Δ.

Multi-Relationships Integration

In practice, there is association between complement and substitute. As show in Figure 3, two substitutable items shared a same complementary item. Conversely, there may

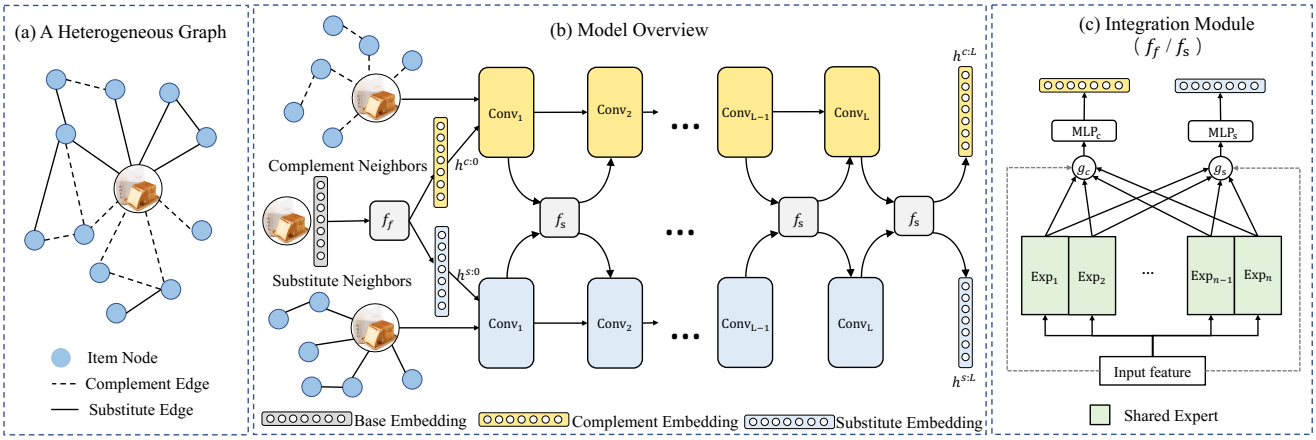

Figure 2: Overview of EMRIGCN. The left part shows a heterogeneous graph where the items are regarded as nodes and the complement and substitute relationships between items are regarded as edges. The middle part shows the skeleton of EMRIGCN which includes feature integration module and structure integration module. The right part shows the implementation detail of the feature/structure integration module.

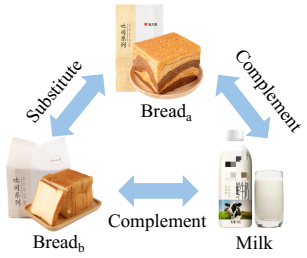

Figure 3: An example of association between the complement and substitute. $\mathrm{Bread_b}$ is a substitutable item of $\mathrm{Bread_a}$. Both of them are complementary with the Milk.

exists substitute relationship between different complements of a same item. The knowledge learnt from one relationship could provide information to infer the other potential relationships missing in data. Based on this intuition, we model multi-relationship prediction by multi-task learning and propose a two-level integration module on a heterogeneous graph. The heterogeneous graph regards items as nodes and regards the complement and substitute relationships as two type of edges respectively. The overall model framework is shown in Figure 2.

Feature Integration In low-level, we propose feature integration module to generate initial node representations for different relationships automatically. The node features are the basic characteristics of items. Instead of using separate feature encoding function in a single relationship prediction (as shown in Equation 1), we build a shared feature encoder module for different relationships in order to model the various mixture patterns of item features at the initial stage. The base node representation is redefined as follows:

$$h_i^{c:0}, h_i^{s:0} = f_f(x_i), \tag{5}$$

where f_f is a share feature encoding function. It takes the node features as input and outputs the initial representations for complement and substitute simultaneously. The implementation detail is listed in "Integration Function" section.

Structure Integration In high-level, we propose structure integration module to explore the mutual influence between different types of neighbors. GNN is good at handling the structure data and exploring high-order information (Wu et al. 2020b). In homogeneous (a single relationship) graph, each item accesses high-order neighbors' information by aggregating the same type of neighbors' message iteratively (see Equation 2). However, besides the high-order information from the same type neighbors, the different type of neighbors can also provide valuable information for their correlation. Inspired by Cross-stitch (Misra et al. 2016), we build a multi-type neighbor aggregate operator in each convolution layer to adaptively integrate high-order information from different relationships. The message aggregation processing is redefined as follows:

$$\hat{h}_i^l = [\mathrm{agg}(\{h_j^{c:l-1} | j \in \mathcal{N}_i^c\}) : \mathrm{agg}(\{h_j^{s:l-1} | j \in \mathcal{N}_i^s\})], \tag{6}$$

$$\hat{h}_i^{c:l}, \hat{h}_i^{s:l} = f_s(\hat{h}_i^l), \tag{7}$$

where f_s is a structure integration function. It integrates the aggregated messages from different type of neighbors and outputs the augmented aggregated result for complement and substitute simultaneously. Since $\hat{h}_i^{c:l}$ and $\hat{h}_i^{s:l}$ have already migrated information to each other, they can be used to update the hidden state of v_i in l-layer by Equation 3 respectively.

Integration Function We hope f_f and f_s can automatically learn the representations of complement and substitute by their mutual influence. Inspired by multi-task learning, we employ MMoE (Ma et al. 2018) as the base integration function which consists of several expert networks and two gate networks (complement gate and substitute gate). MMoE can explicitly model multi-type relationships

and adaptively adjust network parameters between modeling shared information and relationship specific information.

The share feature encoding function f_f is defined as:

$$f_f(x_i) = (\sum_{j=1}^{n_f} g^c_{f:j}(x_i) f_{f:j}(x_i), \sum_{j=1}^{n_f} g^s_{f:j}(x_i) f_{f:j}(x_i)) \quad (8)$$

where $f_{f:j}$ is jth expert for feature integration using multi-layer perceptron. n_f is the number of experts, $g^c_{f:j}$ and $g^s_{f:j}$ are the gating scalar values for jth expert, which are calculated as follows:

$$g^c_f(x_i) = \text{softmax}(W^c_f x_i), \quad (9)$$

$$g^s_f(x_i) = \text{softmax}(W^s_f x_i), \quad (10)$$

where W^c_f and W^s_f are two linear transformation matrixes for complement and substitute in feature integration module.

The structure integration function f_s is defined in a similar way:

$$f_s(\hat{h}^l_i) = (\sum_{j=1}^{n_s} g^c_{s:j}(\hat{h}^l_i) f_{s:j}(\hat{h}^l_i), \sum_{j=1}^{n_s} g^s_{s:j}(\hat{h}^l_i) f_{s:j}(\hat{h}^l_i)), \quad (11)$$

where $g^c_{s:j}$ and $g^s_{s:j}$ are the gating scalar values calculated as:

$$g^c_s(\hat{h}^l_i) = \text{softmax}(W^c_s \hat{h}^l_i), \quad (12)$$

$$g^s_s(\hat{h}^l_i) = \text{softmax}(W^s_s \hat{h}^l_i), \quad (13)$$

The expert networks are shared and co-trained in both type of relationships prediction tasks. Multi-relationships integration can improve the accuracy of the model by transferring the knowledge between different relationships. Even when an item lacks of one type of neighbors, it can still generate a robust representation by reasoning from the information of the other type relationship.

Relationship Enhance

Most existing literature regards item relationship prediction as a classification task then adopts max-margin loss function or cross-entropy loss function to classify positive and negative examples. The strength of substitute and complement connections has never been concerned before. However, whether the connection between items is strong or weak is valuable for downstream applications, such as retrieval and ranking. In order to further infer the strength information, we build an auxiliary loss function based on triplet loss between two related items to distinguish which one is more associated with the query item.

$$\mathcal{L}'(z^c_i) = \text{I} * max\{0, s(z^c_i, z^c_{pos_1}) - s(z^c_i, z^c_{pos_2}) + \Delta'\}, \quad (14)$$

$$\text{I} = \begin{cases} 1 & \text{if } r^c(v_i, v_{pos_1}) < r^c(v_i, v_{pos_2}) \\ -1 & \text{else ,} \end{cases} \quad (15)$$

$$\Delta' = \gamma * (r^c(v_i, v_{pos_2}) - r^c(v_i, v_{pos_1})), \quad (16)$$

where v_{pos_1} and v_{pos_2} are sampled positive examples according to the query item v_i, r^c is a piecewise function for complementary relationship which divides positive examples into different grades by the ranking of a predefined strength value. I is an indicator function. If the strength of (v_i, v_{pos_1}) is stronger than that of (v_i, v_{pos_2}), the score of $(z^c_i, z^c_{pos_2})$ should be smaller than that of $(z^c_i, z^c_{pos_1})$ by Δ'. Δ' is the gap of grades with scale parameter γ.

The same as Equation 4, we regard the margin between positive and negative examples as main loss $\mathcal{L}_{main}$. The loss for z^c_i can be defined as:

$$\mathcal{L}(z^c_i) = (1-\alpha)\mathcal{L}_{main}(z^c_i) + \alpha\mathcal{L}'(z^c_i). \quad (17)$$

The loss for z^s_i ($\mathcal{L}(z^s_i)$) is generated by the same way. The final loss function can be formulated as a multi-task loss:

$$\mathcal{L} = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} (\mathcal{L}(z^c_i) + \mathcal{L}(z^s_i)). \quad (18)$$

	Beauty	Electronics	Cellphones	Clothing	Supermarket
Total item	109.0K	108.4K	51.6K	45.3K	16.5K
Total edges	3.13M	1.93M	822.7K	435.0K	465.4K
Com. item	85.5k	67.2K	38.6K	23.5K	9.50K
Com. edges	1.33M	1.24M	692.2K	134.2K	161.9K
Sub. item	106.9K	94.7K	34.2K	43.8K	16.2K
Sub. edges	1.81M	690.6K	130.5K	300.8K	303.5K

Table 1: Statistics of the datasets

Experiments

In this section, we first introduce the used datasets including four public datasets and an industrial dataset. We then compare the proposed EMRIGCN with a series of state-of-the-art baseline methods. We further conduct cold-start study and ablation study to give deep analysis. At last, we share how to apply EMRIGCN in downstream recommendation tasks and show the performance on online A/B testing.

Datasets

Public Dataset We conduct experiments on the Amazon datasets released by(McAuley, Pandey, and Leskovec 2015), whereas several types of item pairs are crawled. We select four subsets from the dataset, including Beauty, Electronics, Cellphones and Clothing. Following the previous works (McAuley, Pandey, and Leskovec 2015), we regard "also-viewed" and "also-bought" as substitute and complement relationships respectively. We randomly choose 85% of the labelled pairs as training set, and use the rest as testing set.

Industrial Dataset Industrial dataset is constructed by user behavior logs from Meituan Maicai, an online supermarket platform in China. For the training set, we regard the items in the same search result as substitutable items and regard the items in the same shopping order as complementary items over the last 6 months. The testing set is built by the same way over the following one month. Compared with the public dataset, the industrial dataset is a comprehensive dataset which contains more categories and cross-categories related pairs.

The detailed statistics of datasets are shown in Table 1.

Compared Methods & Metrics

We compare EMRIGCN with following methods:

- LVA (Rakesh et al. 2019). This model is a context-based model. LVA links two variational autoencoders to learn item representation from the context features for relationship inference.
- PinSage (Ying et al. 2018). This is a widely-used GCN-based model (Wu et al. 2020b). We apply it on a single relationship prediction task and train two separated PinSage model for complement and substitute respectively.
- R-GCN (Schlichtkrull et al. 2018). R-GCN learns a single shared representation with edge relational information for each item and uses DistMult factorization (Yang et al. 2014) for multi-relationship prediction.
- DecGCN (Liu et al. 2020). DecGCN models item complement representation and substitute representation in separated embedding spaces. It adopts attention mechanism and knowledge transferring to explicitly learn the mutual influence between different relationships.

For each node in testing set, we fuse its' related items with 3000 sampled negative items and rank them by different methods. We use Recall@k and NDCG@k to evaluate the performance of top-k result. We abbreviate Recall@k and NDCG@k as R@k and N@k in following tables.

Implement Details

We implement our method by DGL[1]. We adopt a 2-layers convolutional networks and sample 10 neighbors in each layer for all GCN-based methods. In public dataset, the predefined strength value is obtained by the co-occurrence probability of categories corresponding to the items. In industrial dataset, the predefined strength value is obtained by the co-occurrence probability of items in search result or shopping order. To calculate r, we divided the strength into 10 grades with equal frequency. The hyper parameters are $d = 64$, $\Delta = 1$, $\gamma = 0.02$, and $\alpha = 0.2$. All the models are trained for 10 epochs using Adam optimizer on NVIDIA Tesla V100 GPU.

Overall Performance

Table 2 shows the experiment result on five datasets. We omit the comparison with LVA in supermarket dataset since the reviews of items are too short and contain default copywriting. We mark the best and the second best performance among the methods with boldfaced and underlined. From the result, we can make the following observations:

- Our proposed model consistently performs best among all method over all datasets on all evaluation metrics. For example, EMRIGCN outperforms best baseline by 0.0090 and 0.0590 w.r.t Recall@30, NDCG@30 on the task of inferring complementary items of Amazon Beauty. In the supermarket dataset, the task is more difficult due to there are more categories and cross-categories related pairs. EMRIGCN also achieves best performance.

[1]https://www.dgl.ai/

Dataset	Method	Complement		Substitute	
		R@30	N@30	R@30	N@30
Beauty	LVA	0.1967	0.1395	0.2082	0.1754
	PinSage	0.9310	0.7648	0.9621	0.7713
	R-GCN	0.9472	0.7781	0.9520	0.7922
	DecGCN	0.8422	0.4705	0.8508	0.6530
	EMRIGCN	**0.9562**	**0.8371**	**0.9781**	**0.8652**
Electronics	LVA	0.3077	0.2319	0.3163	0.2206
	PinSage	0.8436	0.6368	0.9443	0.7324
	R-GCN	0.8774	0.6844	0.8798	0.6282
	DecGCN	0.7895	0.5223	0.8122	0.5673
	EMRIGCN	**0.9144**	**0.7425**	**0.9745**	**0.8043**
Cellphone	LVA	0.1195	0.0722	0.1402	0.1078
	PinSage	0.8922	0.6926	0.8643	0.6127
	R-GCN	0.9193	0.7255	0.7023	0.4460
	DecGCN	0.8452	0.7175	0.5622	0.4275
	EMRIGCN	**0.9528**	**0.7989**	**0.9298**	**0.7219**
Clothing	LVA	0.2201	0.1395	0.2782	0.1754
	PinSage	0.8660	0.6181	0.9514	0.7472
	R-GCN	0.9338	0.7079	0.9563	0.7778
	DecGCN	0.6162	0.5095	0.8505	0.6894
	EMRIGCN	**0.9643**	**0.7866**	**0.9850**	**0.8653**
SuperMarket	LVA	-	-	-	-
	PinSage	0.6575	0.4484	0.9231	0.8234
	R-GCN	0.7144	0.4813	0.9422	0.8563
	DecGCN	0.4323	0.2355	0.6855	0.4872
	EMRIGCN	**0.7391**	**0.5021**	**0.9673**	**0.8646**

Table 2: Performance comparison on five datasets

- By and large, the multi-relationship model (R-GCN, EMRIGCN) achieves better result than single relationship model (PinSage), except for DecGCN (More detail analysis will be conducted in next section).
- R-GCN performs poor on the substitute task of Electronics and Cellphone. Because the number of neighbors for complement and substitute are imbalanced.[2] R-GCN relies on a single shared representation for each item, it's hard to deal with imbalanced data.
- The text-based models learn relation among one-hop related items which is inefficient. It's hard for LVA to distinguish related items from thousands of unrelated items.

Cold-Start Study

To evaluate the robustness of EMRIGCN, we conduct analysis on cold-start items. Following previous work (Hao et al. 2020), we define an item as a cold-start item if it has less than four complement or substitute neighbors. Due to the space limit, only the result on Amazon Beauty is reported in Table 3 (other datasets show similar results). The number of labelled items in the cold-start testing data is less than that of in the full testing data, so we focus on the performance of top-1 and top-10 result instead of top-30 result.

EMRIGCN achieves more significant improvement. It outperforms PinSage by 0.1548, 0.0641 and 0.1180 w.r.t Recall@1, Recall@10, NDCG@10 on the complement task

[2]Specifically, the average number of neighbors for complement and substitute is 31 and 12 on the Electronics dataset.

Method	Complement			Substitute		
	R@1	R@10	N@10	R@1	R@10	N@10
PinSage	0.5344	0.9090	0.7226	0.5487	0.9073	0.7295
R-GCN	0.5250	0.9020	0.7119	0.4785	0.8300	0.6496
DecGCN	0.0120	0.0682	0.0326	0.0277	0.1009	0.0482
EMRIGCN	**0.6892**	**0.9731**	**0.8406**	**0.7068**	**0.9630**	**0.8422**

Table 3: Performance comparison on cold-start dataset of the Amazon Beauty

Method	Complement		Substitute	
	R@30	N@30	R@30	N@30
PinSage	0.9310	0.7648	0.9621	0.7713
EMRIGCN-FI	0.9406	0.8034	0.9632	0.8182
EMRIGCN-SI	0.9475	0.8078	0.9648	0.8221
EMRIGCN-RE	0.9496	0.8053	0.9654	0.8245
EMRIGCN	0.9562	0.8371	0.9781	0.8652

Table 4: Ablation studies on Amazon Beauty dataset

and 0.1581, 0.0557 and 0.1127 w.r.t Recall@1, Recall@10, NDCG@10 on the substitute task. Apparently, R-GCN and DecGCN perform worse than PinSage. DecGCN works effectively for the node which has both type of neighbors simultaneously. However, if a node only has one type of neighbors, the attention and knowledge transfer mechanism between different type of relationships will be invalid (the missing relationship may become a random noise during the attention and transfer processing). With the increasing of neighbors for each item, DecGCN achieves much better performance. Experiment result demonstrates that our relationship integration mechanism is more efficient and robust.

Ablation Study

To further investigate the effect of each component, we design three variants of EMRIGCN as follows:

- EMRIGCN-FI: EMRIGCN-FI removes the feature integration module from the full EMRIGCN.
- EMRIGCN-SI: EMRIGCN-SI removes the structure integration module from the full EMRIGCN.
- EMRIGCN-RE: EMRIGCN-RE removes the relationship enhance module from the full EMRIGCN.

We compare EMRIGCN with these variants on Amazon Beauty. As shown in Table 4, EMRIGCN significantly outperforms the variants on both tasks which indicates the benefits brought by combining all components. All the variants can beat PinSage. Therefore, we can conclude that the mutual influence between different types of relationship and the strength of relationship are benefit to understand relationship between items. More specifically, experiment results show that the feature integration module influences EMRIGCN the most comparing with other components. This is because the feature encoder is the base of the representation for items which will affect structure integration and relationship enhance in following steps.

To measure the role of strength information, we evaluated the ndcg@k on related items by regarding the grades of strength as label instead of 0/1 in industrial dataset. EMRIGCN and EMRIGCN-RE achieve 0.7577 and 0.7496 w.r.t NDCG@30 on the task of inferring complementary items (the conclusion is consistent on the substitute task). EMRIGCN is more helpful to make the items with strong relationship rank before the weak which verifies the model has learned the strength of relationship among related items.

Experiment	VBR	RPM
Retrieval Experiment	5.62%	5.78%
Ranking Experiment	9.68%	9.56%

Table 5: Online A/B test

Online A/B Test

We deploy EMRIGCN on various downstream recommendation tasks of our online supermarket platform, namely Meituan Maicai, to evaluate the online effectiveness. We design two methods applied in retrieval and ranking stage, respectively. In the retrieval stage, the substitute and complement item embeddings are published in an ANN (approximate near neighbor) indexing system. We use the embeddings and ANN to generate new candidates based on user historical behaviors and then merge them with other candidate items from different sources. In the ranking stage, we choose DIN (Zhou et al. 2018) as the base method to model user interaction sequences using target attention module. Instead of using random matrix to initialize the sequences, we apply the substitute and complement embeddings which are flexible during the training process.

We choose online VBR (Visit Buy Rate) and RPM (Revenue Per Thousand Impressions) as the observed metrics. Table 5 shows the performance improvements over retrieval and ranking experiments. The new candidates generated by EMRIGCN brings 5.62% improvements on VBR and 5.78% improvements on RPM. In the ranking experiment, compared with base DIN model, EMRIGCN achieves 9.68% improvements on VBR and 9.56% on RPM, respectively. EMRIGCN has been deployed online, which contributes a significant business revenue growth.

Conclusion

In this paper, we propose a novel framework, namely EMRIGCN, to infer substitutable and complementary items. The EMRIGCN improves the performance of inferring result by learning the mutual influence between different types of relationships and the strength of related pairs. Specifically, we design a two-level integration module to integrate the information of both relationship during the whole message propagation process in GNN. Then, we use an auxiliary loss to further distinguish the strength of related pairs with more supervision. Our solution achieved outstanding performance on four public datasets and an industrial dataset. We also deploy the EMRIGCN on our online recommender system for retrieval task and ranking task. The A/B testing demonstrates the effectiveness of EMRIGCN in real world application.

References

Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; and Yin, D. 2019. Graph neural networks for social recommendation. In *The world wide web conference*, 417–426.

Gao, C.; Wang, X.; He, X.; and Li, Y. 2022. Graph neural networks for recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 1623–1625.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247*.

Hao, J.; Zhao, T.; Li, J.; Dong, X. L.; Faloutsos, C.; Sun, Y.; and Wang, W. 2020. P-companion: A principled framework for diversified complementary product recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2517–2524.

Linden, G.; Smith, B.; and York, J. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1): 76–80.

Liu, W.; Zhang, Y.; Wang, J.; He, Y.; Caverlee, J.; Chan, P. P.; Yeung, D. S.; and Heng, P.-A. 2021. Item relationship graph neural networks for e-commerce. *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, Y.; Gu, Y.; Ding, Z.; Gao, J.; Guo, Z.; Bao, Y.; and Yan, W. 2020. Decoupled graph convolution network for inferring substitutable and complementary items. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2621–2628.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1930–1939.

McAuley, J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 785–794.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3994–4003.

Rakesh, V.; Wang, S.; Shu, K.; and Liu, H. 2019. Linked variational autoencoders for inferring substitutable and supplementary items. In *Proceedings of the twelfth acm international conference on web search and data mining*, 438–446.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. v. d.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, 593–607. Springer.

Song, W.; Shi, C.; Xiao, Z.; Duan, Z.; Xu, Y.; Zhang, M.; and Tang, J. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1161–1170.

Wang, Z.; Jiang, Z.; Ren, Z.; Tang, J.; and Yin, D. 2018. A path-constrained framework for discriminating substitutable and complementary products in e-commerce. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 619–627.

Wu, L.; Li, J.; Sun, P.; Hong, R.; Ge, Y.; and Wang, M. 2020a. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Wu, L.; Sun, P.; Fu, Y.; Hong, R.; Wang, X.; and Wang, M. 2019a. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 235–244.

Wu, Q.; Zhang, H.; Gao, X.; He, P.; Weng, P.; Gao, H.; and Chen, G. 2019b. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*, 2091–2102.

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

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020c. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Yan, A.; Dong, C.; Gao, Y.; Fu, J.; Zhao, T.; Sun, Y.; and McAuley, J. 2022. Personalized complementary product recommendation. In *Companion Proceedings of the Web Conference 2022*, 146–151.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Yang, Z.; Ye, J.; Wang, L.; Lin, X.; and He, L. 2022. Inferring substitutable and complementary products with Knowledge-Aware Path Reasoning based on dynamic policy network. *Knowledge-Based Systems*, 235: 107579.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 974–983.

Yu, J.; Yin, H.; Li, J.; Wang, Q.; Hung, N. Q. V.; and Zhang, X. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*, 413–424.

Zhang, M.; Wei, X.; Guo, X.; Chen, G.; and Wei, Q. 2019a. Identifying complements and substitutes of products: A neural network framework based on product embedding. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(3): 1–29.

Zhang, S.; Yin, H.; Wang, Q.; Chen, T.; Chen, H.; and Nguyen, Q. V. H. 2019b. Inferring Substitutable Products with Deep Network Embedding. In *IJCAI*, 4306–4312.

Zheng, J.; Wu, X.; Niu, J.; and Bolivar, A. 2009. Substitutes or complements: another step forward in recommendations. In *Proceedings of the 10th ACM conference on Electronic commerce*, 139–146.

Zheng, L.; Zuo, Z.; Wang, W.; Dong, C.; Momma, M.; and Sun, Y. 2021. Heterogeneous Graph Neural Networks with Neighbor-SIM Attention Mechanism for Substitute Product Recommendation. In *AAAI Deep Learning on Graphs Workshop*.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and Gai, K. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1059–1068.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81.