Metapath- and Entity-aware Graph Neural Network for Recommendation

Zhiwei Han* [1,3], **Muhammad Umer Anwaar*** [2,3], **Shyam Arumugaswamy*** [2,3]
Thomas Weber [4], **Tianming Qiu** [1], **Hao Shen** [1], **Yuanting Liu** [1], **Martin Kleinsteuber** [2,3]
[1]fortiss GmbH, [2]Mercateo Services GmbH,
[3]Technical University of Munich, [4]Ludwig-Maximilians-University

Abstract

Due to the shallow structure, classic graph neural networks (GNNs) failed in modelling high-order graph structures that deliver critical insights of task relevant relations. The negligence of those insights lead to insufficient distillation of collaborative signals in recommender systems. In this paper, we propose PEAGNN, a unified GNN framework tailored for recommendation tasks, which is capable of exploiting the rich semantics in metapaths. PEAGNN trains multilayer GNNs to perform metapath-aware information aggregation on collaborative subgraphs, h-hop subgraphs around the target user-item pairs. After the attentive fusion of aggregated information from different metapaths, a graph-level representation is then extracted for matching score prediction. To leverage the local structure of collaborative subgraphs, we present entity-awareness that regularizes node embedding with the presence of features in a contrastive manner. Moreover, PEAGNN is compatible with the mainstream GNN structures such as GCN, GAT and GraphSage. The empirical analysis on three public datasets demonstrate that our model outperforms or is at least on par with other competitive baselines. Further analysis indicates that trained PEAGNN automatically derives meaningful metapath combinations from the given metapaths.

*The first three authors have equal contribution.

Preliminary work. Do not distribute.

1 Introduction

Integrating content information for user preference prediction remains a challenging task in the development of recommender systems (RSs). In spite of the effectiveness, most collaborative filtering (CF) [20, 15, 25, 28] methods still suffer from the incapability of modeling content information such as user profiles and item features [17, 22]. This leads to the poor performance of recommendation models in a cold-start setting.

Recent efforts addressing this problem can be mainly categorized into two classes. Factorization models such as factorization machine (FM) [23], neural factorization machine (NFM) [13] and Wide&Deep models [5] fuse numerical features for each individual training sample and have showed competitive results on various datasets, though the dependencies between content information is still neglected. Graph-based methods such as NGCF [34], KGAT [33] and KGCN [32] represent RSs with graph structured data [2, 43, 34, 39, 27] and exploit the graph structure to enhance the node-level representations for better recommendation performance. However, learning with node-level representations loses the correspondences and interactions between the content information of users and items, since their node embeddings are learned independently as indicated by Zhang et al. [41]. Another disadvantage of previous GNN-based methods is that the relations of long-term correspondences are either ignored (KG-based methods) or mixed up (GNN-based methods) without the explicit modelling of high-order structure.

A natural solution of capturing the inter- and intra-relations between content features and user-item pairs is to explore the high-order information encoded by metapaths [6, 40], a set of composite relations designed for representing multi-hop structure and sequential semantics. To our best knowledge, only a limited number of research efforts have attempted to enhance GNNs with metapaths. MAGNN [9] aggregates

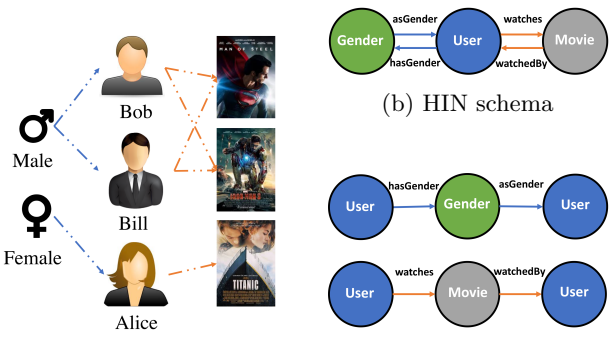
intra-metapath information for each path instance, but it can lead to high memory consumption. Fan et al. [7] utilizes the structural information in metapaths to improve the node-level representation for recommendation, but they have a different focus of intent recommendation. Nevertheless, we propose MetaPath- and Entity-Aware Graph Neural Network (PEAGNN), a unified GNN framework, which aggregates information over multiple metapath-aware subgraphs and fuse the aggregated information to obtain node representation using attention mechanism. As a first step, we extract an enclosing collaborative subgraph for each training user-item pair within its h-hop neighbours. Secondly, the collaborative subgraph is decomposed into m metapath-aware subgraphs based on the schema of the m selected metapaths. After PEAGNN has updated the node representation and outputs a graph-level representation of the given collaborative subgraph, an MLP is trained for matching score prediction. To further exploit the local structure of collaborative subgraphs, we introduce entity-awareness, a contrastive regularizer which pushes the user nodes and item nodes closer to their connected feature entity nodes, while pushing them apart from unconnected ones. We train PEAGNN by jointly minimizing entity-aware regulation term and Bayesian Personalized Rank (BPR) loss, a widely used contrastive loss in recommender system. Furthermore, our PEAGNN can be combined with most mainstream graph convolution layers such as GAT, GCN and GraphSage. The contributions of our work are summarized as follows:

1. We decouple the sequential semantics conveyed by metapaths into different metapath-aware subgraphs and propose PEAGNN, which explicitly propagates and aggregates multi-hop semantics on metapath-aware subgraphs.
2. We fuse the aggregated information from metapath-aware subgraphs using soft attention and extract graph-level representation for matching score prediction of target user-item pairs.
3. The empirical analysis on three public datasets demonstrate that PEAGNN outperforms other competitive baselines and is capable of choosing meaningful metapath combinations.

2 Preliminaries

In this section, we start with the definitions of basic concepts used in our models. We then discuss the recommendation task description for using collaborative subgraphs as input.

Heterogeneous Information Network and Metapath In recommendation tasks, we typically have user-item interaction history e.g., purchases and clicks as input. Those interactions as well as their features can be naturally represented in the setting of a **Heterogeneous Information Network** (HIN) [29]. A HIN is defined as a graph $G = (V, E)$ with a network schema. The network schema reveals the meta structure of a HIN such as node types T_v and edge types T_e, where $|T_v| + |T_e| > 2$. A **Metapath** [30], a relation sequence composed with node types and edge types, is proposed to represent the structural and semantic relation between connected objects. Figures 1(a)(b) shows a toy example of a HIN for a recommendation task with its network schema. As shown in the left subfigure, the HIN contains multiple node types (e.g., *user*, *gender* and *item* etc.) and interaction types (e.g., *watches* and *asGender*). Each metapath in Figure 1(c) encodes unique semantics e.g., the *user-gender-user* (U-G-U) metapath reveals the shared interest of the same gender, while the *user-movie-user* (U-M-U) metapath indicates the co-watch relations. In this paper, we focus on metapaths ending up with either a user or item node for better distillation of collaborative signals.



(a) HIN toy example (b) HIN schema (c) Metapaths

Figure 1: An example of heterogeneous information network, its schema and two given metapaths.

Collaborative Subgraph A collaborative subgraph is an h-hop undirected enclosing subgraph around one user-item pair. Given the interaction history and content information of a recommendation task, the collaborative subgraph of a user-item pair (u, i) is induced from the HIN of the recommendation task by nodes u, i and their feature entity nodes within h hops. Such local subgraphs contain rich semantic and collective pattern information of user-item interactions. A major difference between our collaborative subgraphs and the subgraphs proposed by Zhang et al. [41] is that their subgraphs neglect side information by excluding all feature entity nodes.

Task description HINs provide a unified framework of representing the relations between user-item interactions and side information with heterogeneous

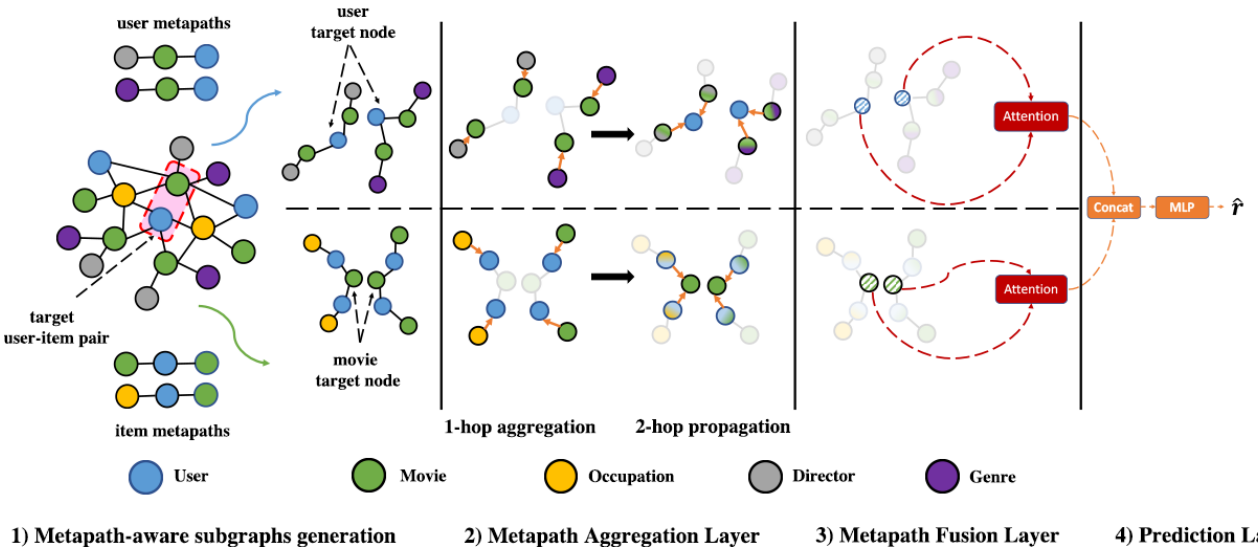


Figure 2: Illustration of the proposed PEAGNN model on the MovieLens dataset. Subfigure 1) shows the metapath-aware subgraphs generated from a collaborative subgraph with the given user- and item metapaths. Subfigures 2), 3) and 4) illustrate the metapath-aware information aggregation and fusion workflow of our PEAGNN model. For simplicity, we have only adopted 2-hop metapaths.

graph structures. Hence, we formulate the recommendation task to be tackled in this paper on top of HINs as follows:

- **Input** A HIN that includes all user/item historical interactions as well as all their feature entity nodes and the initial node embeddings $\mathbf{E}$.
- **Output** Learned heterogeneous node embedding $\mathbf{E}'$, a graph encoder $f_{PEAGNN}(\cdot)$ which extracts the graph-level representations from given collaborative subgraphs and a prediction model $f_{\boldsymbol{\theta}}(\cdot)$, where $\boldsymbol{\theta}$ is a trainable parameter. The prediction model maps the collaborative subgraph, whose center is a user-item pair (u, i), to a scalar matching score that the user u would interact with the item i.

3 Approach

We propose PEAGNN, a unified GNN framework, which exploits and fuses rich sequential semantics in selected metapaths. To leverage the underlying local structure for recommendation, we introduce an entity-aware regularizer that distinguishes users and items from their unrelated features in a contrastive fashion. Figure 2 illustrates the PEAGNN framework, which consists of three components:

1. A **Metapath Aggregation Layer**, which explicitly aggregates information on metapath-aware subgraphs.
2. A **Metapath Fusion Layer**, which fuses the aggregated node representations from multiple metapath-aware subgraphs using attention mechanism.
3. A **Prediction Layer**, which readouts the graph-level representations of collaborative subgraphs and estimate the likelihood of potential user-item interactions.

3.1 Metapath Aggregation Layer

Sequential semantics encoded by metapaths reveal different aspects towards the connected objects. Appropriate modelling of metapaths can improve the expressiveness power of node representations. Our aim is to learn node representations that preserve the sequential semantics in metapaths. Rather than consider each individual path as input [7], which is memory and time intensive, PEAGNN performs a step-wise information propagation over metapath-aware subgraphs.

Metapath-aware Subgraph A metapath-aware subgraph is a directed graph induced from the corresponding collaborative subgraph by following one specific metapath. Since the goal is to learn metapath-aware user/item representation for recommendation, we choose those metapaths ending up with either a user or an item node, such that the information aggregation on metapath-aware subgraphs always terminate on user(item) nodes. In the following, we denote the terminal of metapath information propagation on a metapath-aware subgraph as its target node.

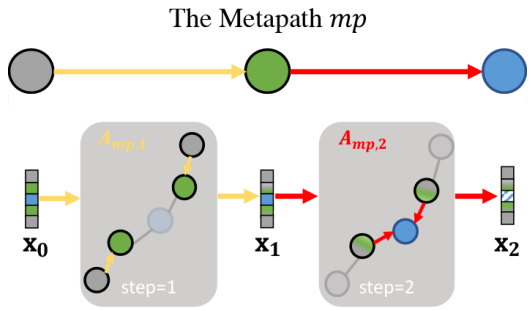


Figure 3: Information propagation on a metapath-aware subgraph. For clarity, we have omitted the symbol of the given metapath-aware subgraph in the node representation.

Information Propagation on Metapath-aware Subgraphs PEAGNN trains a GNN model to perform stepwise information aggregation on metapath-aware subgraphs. By stacking multiple GNN layers, PEAGNN is capable of explicitly exploring the high-order connectivity in a metapath, so as to capture collaborative signal effectively. Figure 3 illustrates the flow of information propagation on a given metapath-aware subgraph generated from the metapath mp. In Figure 3, $\boldsymbol{H}_k$ is the node representations of all nodes in the metapath-aware subgraph after kth propagation, $A_{mp,k}$ is the adjacency matrix of the propagation at the propagation step k and we use orange and red color to highlight the edges being propagated on certain aggregation steps. Considering the high-order semantic revealed by multi-hop metapaths, we stack multiple GNN layers and recurrently aggregate the representation on the metapaths to target node, so the high-order semantic is injected into target node representation. The metapath-aware information aggregation is shown as follows

$$\boldsymbol{h}_1 = \sigma(GNN_1(\boldsymbol{h}_0, A_{mp,1})),$$
$$\boldsymbol{h}_2 = \sigma(GNN_2(\boldsymbol{h}_1, A_{mp,2})).$$

Without loss of generality, the final representation of a target node on a metapath-aware subgraph with N-hop neighbours can be derived by stacking N GNN layers. Output of the nth layer can be written as follows,

$$\boldsymbol{h}_n = \sigma(GNN_n(\boldsymbol{h}_{n-1}, A_{mp,n})),$$

where $n \in \mathbb{N}$ and $1 < n < N$.

3.2 Metapath Fusion Layer

After information aggregation within each metapath-aware subgraphs, the metapath fusion layer combines and fuses the semantic information revealed by all metapaths. Assume for a node v, we have a set of its node representations $\mathbf{H}_v = \{\mathbf{h}_v^{mp_1}, \mathbf{h}_v^{mp_2}, ..., \mathbf{h}_v^{mp_k}\}$ aggregated from multiple metapaths, where k is the number of available metapaths. Semantics disclosed by metapaths are not of equal importance to node representations and the contribution of every metapath should also be adjusted accordingly. Therefore, we leverage soft attention to learn the importance of each metapath, instead of adopting element-wise *mean*, *max* and *add* operators. Note, PEAGNN applies a node-wise attentive fusion of metapath aggregated node representation, rather than use fixed attention factor for all nodes as in previous works, which failed to capture the node-specific metapath preference. We utilize the attention mechanism to fuse the metapath aggregated node representation for each target node. For a given target node v, the metapath fusion is as following:

$$\mathbf{c}_v = trace(\mathbf{W}_a^T \mathbf{H}_v), \tag{1}$$

where $\mathbf{c}_v$ is a vector of metapath importance and $\mathbf{W}_a^T$ is a learnable parameter. We then normalize the metapath importance score using softmax function and get the attention factor for each metapath:

$$\text{att}_v^{mp_i} = \frac{\exp(c_v^{mp_i})}{\sum_{j=1}^{K} \exp(c_v^{mp_j})}, \tag{2}$$

where $\text{att}_v^{mp_i}$ denotes the normalized attention factor of metapath mp_i on the node v. With the learned attention factors, we can fuse all metapath aggregated node representations to the final metapath-aware node representation as:

$$\mathbf{h}_v = \sum_{i=1}^{K} c_v^{mp_i} \mathbf{h}_v^{mp_i}, \tag{3}$$

3.3 Prediction Layer

Next, we readout the node representations of collaborative subgraphs into a graph-level feature vector. In previous work, many pooling methods were investigated such as SumPool, MeanPooling, SortPooling [42] and DiffPooling [37]. However, we adopt a different pooling strategy which concatenates the aggregated representations of the center user and item nodes in the collaborative subgraphs.

$$\mathbf{h_g} = \text{concat}(\mathbf{h}_u, \mathbf{h}_i) \tag{4}$$

After obtaining the graph-level representation of a collaborative subgraph, we utilize a 2-layer MLP to compute the matching score of a user-item pair. Given a collaborative subgraph $\mathcal{G}_{u,i}$, the matching score of user u and item i is:

$$\tilde{r}(\mathcal{G}_{u,i}) = \mathbf{w}_2^T \sigma(\mathbf{w}_1^T \mathbf{h}_g + \mathbf{b}_1) + \mathbf{b}_2 \tag{5}$$

where $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{b}_1$ and $\mathbf{b}_2$ are the trainable parameters of the MLP which map the graph-level representation $\mathbf{h}_g$ to a scalar matching score $\tilde{r}$, and σ is the non-linear ReLU activation function.

3.4 Graph-level representation for recommendation

Compared to the previous GNN-based methods such as NGCF, KGAT and KGCN that use node-level representations for recommendation, PEAGNN predicts the matching score of a user-item pair by mapping its corresponding metapath-aware subgraph to a scalar as shown in Figure 2. As manifested by [41], methods using node-level representation suffer from the over-smoothness problem [21][19] and fail to model the correspondences of the structural proximity of a node pair, as their node-level representations are learned independently, while a graph-level GNN with sufficient rounds of message passing can better capture the interactions between the local structures of two nodes [35].

3.5 Model Training

After applying components introduced in the previous subsections, we obtain the predicted matching scores of arbitrary user-item pairs. To train model parameters in an end-to-end manner, we minimize the pairwise Bayesian Personalized Rank (BPR) loss [24], which has been widely used in RSs. It leverages the presence of user-item interactions in a contrastive way. To be more specific, BPR assumes that the observed interactions, which can better represent users' preferences, should be assigned higher matching score compared to those unobserved ones. The BPR loss used in this paper is formed as following,

$$\mathcal{L}_{CF} = \sum_{(u,i_+,i_-)\in\mathcal{O}} -\ln\Big(\tilde{r}(u,i_+) - \tilde{r}(u,i_-)\Big), \qquad (6)$$

where $\mathcal{O} = \{(u,i_+,i_-)|(u,i_+) \in R_+, (u,i_-) \in R_-\}$ is the training set, R_+ is the observed user-item interactions (positive samples) while R_- is the unobserved user-item interactions (negative samples), and σ is the non-linear sigmoid function.

Item Entity-awareness Although user and item representations can be derived by information aggregation and fusion on metapath-aware subgraphs, the local structural proximity of user(item) nodes are still missing. Towards this end, we propose *entity-awareness* to regularize the local structural of user(item) nodes. The idea of *Entity-awareness* is to distinguish items or users with their unrelated feature entities in the embedding space. Specifically, *Entity-awareness* is a distance-based contrastive regularization term that pulls the related feature entity nodes closer to the corresponding user(item) nodes, while push the unrelated ones apart. The regularization term is defined as following:

$$\mathcal{L}_{Entity} = \sum_{(u,i_+,i_-)\in\mathcal{O}} -\Bigg[d(\mathbf{e}_u, \mathbf{e}_{e-,u}) - ln\Big(d(\mathbf{e}_u, \mathbf{e}_{e+,u})\Big) + \Big(d(\mathbf{e}_{i_+}, \mathbf{e}_{e-,i_+}) - d(\mathbf{e}_{i_+}, \mathbf{e}_{e+,i_+})\Big)\Bigg], \qquad (7)$$

where $\mathbf{e}_{e+,u}, \mathbf{e}_{e-,u}$ denote the observed and unobserved feature entity embeddings of user u, $\mathbf{e}_{e+,i_+}, \mathbf{e}_{e-,i_-}$ denote the observed and unobserved feature entity embeddings of positive item i and $d(\cdot,\cdot)$ is a distance measure on the embedding space. In this paper, we use euclidean norm $d(\mathbf{x},\mathbf{y}) = ||\mathbf{x}-\mathbf{y}||_2$ as our distance measure.

Finally, we optimize our model by jointly minimizing the Equation 6 and Equation 7 as follows:

$$\mathcal{L} = \mathcal{L}_{CF} + \mathcal{L}_{Entity} + \lambda\mathbf{\Theta}, \qquad (8)$$

where $\mathbf{\Theta}$ denotes all trainable model parameters, and λ is the regularizer of $l2$ norm to avoid overfitting. We adopt mini-batch Adam [18] optimizer to optimize the model and update the model parameters. For a batch randomly sampled from training set $\mathcal{O}$, we establish their representation by performing information aggregation and fusion on their embeddings, and then update model parameters via back propagation accordingly.

4 Experiments

We conduct experiments in three public datasets to evaluate the effectiveness of proposed PEAGNN. The experiments aim to address the following research questions:

- **RQ1**: How does PEAGNN perform compared to other baseline methods?
- **RQ2**: How does the entity-awareness affect the performance of PEAGNN?
- **RQ3**: What is the impact of different metapaths in recommendation tasks?

4.1 Experimental Settings

4.1.1 Datasets

To verify PEAGNN's validity, we evaluate our model on three public datasets of different sizes, MovieLens-small (small), Yelp (medium) and MovieLens-25M (large).

MovieLens This is widely used benchmark dataset

https://grouplens.org/datasets/movielens/

	MovieLens-small	MovieLens-25M	Yelp
#Nodes	2933	33249	89252
#Users	608	14982	60808
#Items	2121	11560	28237
#Interactions	79619	1270237	754425

Table 1: Statistics of datasets

for movie recommendation. We use small (with approximately 0.1 million ratings) and 25M (with 25 million ratings) versions of the dataset. Here we consider movies as items and ratings as interactions. For ML-small, we use 10-core setting i.e. each user and item will have at least ten interactions. Since the ML-25M is sparse, we include last three years (from 2018) rated movies (items) for computational efficiency. We select item which has at least 10 interactions and user with 10 to 300 interactions to ensure dataset quality.

Yelp[*] This dataset is used for business recommendations and has around 10 million interactions. Here we consider businesses as items; reviews and tips as interactions. The original dataset is highly sparse. So, to ensure dataset quality, we select item which has at least 50 interactions and user with 10 to 20 interactions.

Along with user-item interactions, we use their features as entities to build HIN graph. For both versions of MovieLens dataset, we use user feature of tag and item features like year, genre, actor, director, writer, tag, genometag (only for ML-25M). For Yelp, we extract user features like counts of reviews, friends, fans, stars and item features like attributes, categories and counts of stars, reviews and check-ins. We initially preprocess both the datasets and filter out the infrequent user-item features. The statistics of the three datasets are summarized in Table 1.

4.1.2 Evaluation Metrics

To evaluate our user-item recommendation, we use leave-one-out evaluation. For each user, in case of both MovieLens datasets, we filter the latest interacted item and for Yelp, the most interacted item as the test set. The remaining items are labelled as training set for both datasets. For each user-item positive interaction in training set, we employ negative sampling strategy to generate four negative items for that user. For Yelp and ML-25M, we sample randomly while for ML-small, we sample from the unseen items for each user to generate negative interaction pairs. During evaluation, we randomly sample 99 items for each user and then rank the test item among the 100 items. The ranked list is then evaluated by two metrics: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). We consider only top-10 ranking from the list for both metrics. HR@10 indicates if the test item is present in top-10 recommendations and NDCG@10 assigns higher scores for items at top of the ranked list [14]. We compute the metrics for each test user and report the average score.

[*]https://www.yelp.com/dataset

4.1.3 Baselines

We compare our PEAGNN models with multiple types of competitive baselines, factorization-based (NFM), regularization-based (CFKG), path-based (HeRec, Metapath2Vec) and GNN-based (NGCF, KGCN, KGAT) models. Following the previous works [15, 33], we utilized a look-up table for all node embeddings and initialized the node embedding using Glorot initialization [11].

NFM [13] utilizes neural networks to enhance high-order feature interactions with non-linearity. As suggested by He et al. [13], we apply one hidden layer neural network on input features.

CFKG [1] applies TransE [3] to learn heterogeneous node embedding and converts recommendation to a link prediction problem.

HeRec [6] extends the matrix factorization model with the joint learning of a set of embedding fusion functions.

Metapath2Vec [6] utilizes a skip-gram model to update the node embeddings generataed by metapath-guided random walks. We then use a MLP to predict the matching score with the learned embeddings.

NGCF [34] integrates the user-item bipartite graph structure into the embedding process for collaborative filtering.

KGCN [32] exploits multi-hop proximity information with a receptive field to learn user preference.

KGAT [33] incorporates high-order information by performing attentive embedding propagation with the learned entity attention on knowledge graph.

4.1.4 Hyper Parameter Settings

We implement our PEAGNN models and baselines in Pytorch Geometric 1.5.0 [8]. We fix embedding size to 64 and batch size to 1024 across all models and datasets except for ML-25M where we have batch size of 4096. The representation size is set as 16 for GNN-based models and hidden layer size as 64 for factorization and GNN-based models. We optimize all models with Adam optimizer. For path-based models HeRec and Metapath2Vec, we manually define the random paths. Due to sparsity in paths, we use SparseAdam optimizer and set the random walks per node as 1000 with walk length of 100, context size of 7 and random walk batch size of 2. The dropout ratio is tuned to 0.1 for KGAT/KGCN, 0.2 for NGCF, 0.3 for NFM.

model	MovieLens- small		MovieLens- 25M		Yelp	
	HR@10	**NDCG@10**	**HR@10**	**NDCG@10**	**HR@10**	**NDCG@10**
NFM	0.477	0.2668	0.8132	0.5347	0.8595	0.6062
CFKG	0.4378	0.2381	0.8152	0.5196	0.8729	0.5826
HeRec	0.2668	0.1449	0.607	0.3291	0.5533	0.3302
Metapath2Vec	0.3063	0.1614	0.7956	0.5051	0.6307	0.402
NGCF	0.5016	0.2755	0.7807	0.4866	0.8068	0.481
KGCN	0.5132	0.2788	0.7771	0.4699	0.8125	0.4668
KGAT	0.5214	0.2846	0.8147	0.5236	0.8762	0.6136
PEAGCN	0.5382	0.2951	0.8185	0.5344	0.9041	0.6379
PEAGCN*	0.5576	0.3036	0.8187	0.5361	0.9125	0.6443
(entity-awareness % improv.)	3.60%	2.88%	0.02%	0.32%	0.93%	1.00%
(% improv. w.r.t. baseline)	6.94%	6.68%	0.43%	0.26%	4.14%	5.00%
PEAGAT	0.5375	0.2983	0.8249	0.5414	0.9057	0.6382
PEAGAT*	0.5477	0.3045	**0.8284**	**0.5475**	**0.9128**	**0.6641**
(entity-awareness % improv.)	1.90%	2.08%	0.42%	1.13%	0.78%	4.06%
(% improv. w.r.t. baseline)	5.04%	6.99%	**1.62%**	**2.39%**	**4.18%**	**8.23%**
PEASage	0.5444	0.3003	0.8176	0.5383	0.8772	0.6247
PEASage*	**0.5609**	**0.307**	0.8273	0.5462	0.8837	0.6308
(entity-awareness % improv.)	3.03%	2.23%	1.19%	1.47%	0.74%	0.98%
(% improv. w.r.t. baseline)	**7.58%**	**7.87%**	1.48%	2.15%	0.86%	2.80%

* with entity-awareness

Table 2: Overall Performance Comparison

The learning rate and weight decay rate is 0.001 for all models except for NFM which has learning rate of 0.0001. We use 2-step metapaths and attention channel aggregation across all PEAGNN models. The metapaths for ML-small, ML-25M and Yelp are 9, 13 and 11 respectively. We use L2 regularizer with coefficient of 0.1 for entity-awareness in PEAGNN models. We conduct experiments for 5 runs (30 epochs/run) for MovieLens and 3 runs (20 epochs/run) for Yelp.

4.2 Overall Performance Comparison (RQ1)

The performances of PEAGNN variants and the baselines are presented in Table 2. The major observation from the experimental results are summarized as follows:

1. All variations of PEAGNN consistently outperform the strongest baseline on all three datasets. In particular, the performance gains achieved by PEAGNN are 7.87%, 2.39%, and 8.23% w.r.t. NDCG@10 on ML-small, ML-25m and Yelp datasets, respectively. This verifies the significance of explicit modelling and fusing sequential semantics in metapath, rather than leaving all node messages mixed up during message passing. Moreover, the superior performance also reveals the effectiveness of modelling the local graph structure, while other GNN-based methods simply pay no attention to their structure proximity.

2. GNN-based methods achieve better performance than the CFKG and NFM on sparse dataset, demonstrating the knowledge enriched representations contribute to a performance boost when only few interactions are available. However, CFKG and NFM can better capture collaborative signals than other GNN-based baselines when enough interactions are given.

3. Path-based methods underperform GNN-based models with a large margin on three datasets. Both HeRec and Metapath2Vec+MLP have incorporated the static node embedding using Metapath2vec [6]. The poor performance indicates that unsupervised learned static node embeddings have limited the power of recommender to capture the complex collaborative signals and complex content relations.

4.3 Effect of Entity-awareness (RQ2)

The goal of introducing entity-awareness is to take the advantage of the first-order structure of collaborative subgraphs, which is not well exploited by pure message passing [10] in graph-based RSs. We study the effect of entity-awareness by comparing the performances of our models with and without entity-awareness. The improvements w.r.t. entity-awareness in Table 2 summarizes the effect of entity-awareness for different base models on three datasets. Generally, entity-awareness delivers consistently better performance than PEAGNN without entity-awareness. In particular, a more significant performance gain has been observed in the smaller dataset ML-small with a minimum improvement of 1.9% on HR@10, while models with entity-awareness slightly outperform base models on the larger and denser datasets. It indicates the benefit of leveraging local structure

#run	U-M-U	M-U-M	Y-M-U	A-M-U	W-M-U	D-M-U	G-M-U	T-M-U	T-U-M
1	**-27.45%**	-4.41%	**-24.3%**	-2.21%	-1.9%	-0.96%	-0.96%	-2.53%	-6%
2	**-30.31%**	**-46.98%**	**-11.51%**	-6.37%	-7.28%	-8.18%	-5.16%	-9.71%	-5.45%
3	-1.88%	-5.3%	**-40.62%**	-4.37%	-3.12%	-0.3%	-0.63%	-4.69%	**-45.32%**
4	+3.38%	**-48.77%**	**-23.63%**	+1.53%	-1.85%	-0.62%	+0.3%	0%	-7.67%
5	-4.97%	**-42.24%**	**-49.38%**	-4.34%	-3.1%	-1.55%	-5.29%	-1.55%	-4.04%

Table 3: Effect of metapaths on PEASage w.r.t. HR@10 on ML-small. We highlight the performance drops larger than 10% in the table. (Abbreviation for nodes: U-User, M-Movie, Y-Year, A-Actor, W-Writer, D-Director, G-Genre and T-Tag)

on sparse datasets. Nonetheless, NDCG@10 benefits more from the entity-awareness in the comparison with HR@10 on both MovieLens and Yelp datasets, where PEAGAT has achieved an improvement of 4.06% on NDCG@10. The results verifies the importance of explicit modelling first-order structure in RSs.

4.4 Effect of Metapaths (RQ3)

We adopted the leave-one-out principle to study the impact of each metapath. To be more precise, we drop one specific metapath in the metapath fusion and compare the model performance with the original one without metapath dropout. Table 3 summarizes the performance difference of all selected metapaths of our best Model PEASage with entity-awareness on the ML-small dataset. We consider the metapaths that bring more than 10% performance gain as the key metapaths. The comparison has been done in 5 runs with different random seeds for evaluation. We have the following findings:

1. PEASage learned 4 different key metapaths combinations. It indicates that there could be multiple meaningful metapath combinations and PEASage may not learn the optimal one.

2. Metapaths capture collaborative effects such as U-M-U and M-U-M which are still of great importance among all metapaths.The published year of movies has a significant impact on users' choice since the metapath Y-M-U is selected in all 5 runs. What's more, the tag given by users can be a complementary relation of user-item interactions as shown in the run 3.

5 Related Work

GNN is designed for learning on graph structured data [26, 4, 19]. GNNs apply message passing to iteratively pass messages between each node to update node representation with the underlying graph structure. Besides, GNNs followed by an additional pooling layer are typically used to extract graph presentation for graph-level tasks, e.g., graph classification/clustering. Due to its superior learnability on graphs, GNNs have achieved state-of-the-art performance on node classification [19], graph representation learning [12] and RSs [36]. In the task of RSs, relations such as user-item interactions and feature properties can be presented as multi-typed edges in the graphs. Therefore, GNNs have been proposed to solve the recommendation problem [34, 33, 2]. NGCF [34] embed bipartite graphs of users and items into node representation to capture collaborative signals. GCMC [2] proposed a graph auto-encoder framework, which produced latent features of users and items through a form of differentiable message passing on the user-item graph. KGAT [33] proposed a knowledge graph based attentive propagation, which enhances the node feature by modeling high-order connectivities.

Prior to GNNs, several efforts have been established to explicitly guide the recommender learning with metapaths [16, 31, 38]. Heitmann et al. [16] utilized linked data from heterogeneous data source to enhance CF for the cold-start problem. Sun et al. [31] converted recommendation tasks to relation prediction problems and tackled it with metapath-based relation reasoning. Yu et al. [38] conducted MF framework over metapath similarity matrices to perform recommendation. However, only a limited number of works attempted to combine GNNs with metapaths. MAGNN [9] aggregated intra-metapath information for each path instance that leads to unaffordable memory consumption. MEIRec [7] devised a GNN to perform metapath-guided propagation though, it failed to characterize the importance of each metapath.

6 Conclusion and Future Work

In this work, we study the necessity of incorporating high-order semantics in metapath for recommendation. Instead of mixing up multi-hop messages in graphs, we devised a unified GNN framework PEAGNN, which explicitly performs independent information aggregation on generated metapath-aware subgraphs. Specifically, a metapath fusion layer is trained to learn the metapath importance and adaptively fuse the aggregated metapath semantics in an end-to-end fashion. For first-order local structure exploitation, the entity-awareness, a contrastive connectivity regularizer, is employed on user nodes and item nodes to regularize the learning process. Positive re-

sults of conducted experiments demonstrate the effectiveness and rationality of the proposed PEAGNN.

This work explore the potential of explicitly injecting semantics in metapath into GNNs. However, the advance performance achieved by our model relies on meaningful metapaths. In practice, selection of representative metapaths is a challenging task and it, thus, also opens the door of automatic sequential semantics discovery for future research.

References

[1] Q. Ai, V. Azizi, X. Chen, and Y. Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9): 137, 2018.

[2] R. v. d. Berg, T. N. Kipf, and M. Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[4] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[5] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[6] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[7] S. Fan, J. Zhu, X. Han, C. Shi, L. Hu, B. Ma, and Y. Li. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2478–2486, 2019.

[8] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[9] X. Fu, J. Zhang, Z. Meng, and I. King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, pages 2331–2341, 2020.

[10] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[11] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[12] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[13] X. He and T.-S. Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 355–364, 2017.

[14] X. He, T. Chen, M.-Y. Kan, and X. Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670, 2015.

[15] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[16] B. Heitmann and C. Hayes. Using linked data to build open, collaborative recommender systems. In *AAAI spring symposium: linked data meets artificial intelligence*, volume 2010, 2010.

[17] Z. Huang, W. Chung, T.-H. Ong, and H. Chen. A graph-based recommender system for digital library. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 65–73, 2002.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[20] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[21] Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *arXiv preprint arXiv:1801.07606*, 2018.

[22] P. Lops, M. De Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[23] S. Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[24] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[25] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[26] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[27] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[28] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[29] Y. Sun and J. Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[30] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[31] Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla. When will it happen? relationship prediction in heterogeneous information networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 663–672, 2012.

[32] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*, pages 3307–3313, 2019.

[33] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958, 2019.

[34] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

[35] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[36] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.

[37] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810, 2018.

[38] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han. Collaborative filtering with entity similarity regularization in heterogeneous information networks. *IJCAI HINA*, 27, 2013.

[39] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han. Recommendation in heterogeneous information networks with implicit user feedback. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 347–350, 2013.

[40] J. Zhang, P. S. Yu, and Z.-H. Zhou. Meta-path based multi-network collective link prediction. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1286–1295, 2014.

[41] M. Zhang and Y. Chen. Inductive matrix completion based on graph neural networks. *arXiv preprint arXiv:1904.12058*, 2019.

[42] M. Zhang, Z. Cui, M. Neumann, and Y. Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[43] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang. Bipartite network projection and personal recommendation. *Physical review E*, 76(4):046115, 2007.