# Meta-Graph based Attention-aware Recommendation over Heterogeneous Information Networks

Feifei Dai[1,2], Xiaoyan Gu[1(✉)], Bo Li[1], Jinchao Zhang[1], Mingda Qian[1], and Weiping Wang[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{daifeifei,guxiaoyan,libo,zhangjinchao,qianmingda,wangweiping}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

**Abstract.** Heterogeneous information network (HIN), which involves diverse types of data, has been widely used in recommender systems. However, most existing HINs based recommendation methods equally treat different latent features and simply model various feature interactions in the same way so that the rich semantic information cannot be fully utilized. To comprehensively exploit the heterogeneous information for recommendation, in this paper, we propose a Meta-Graph based Attention-aware Recommendation (MGAR) over HINs. First of all, MGAR utilizes rich meta-graph based latent features to guide the heterogeneous information fusion recommendation. Specifically, in order to discriminate the importance of latent features generated by different meta-graphs, we propose an attention-based feature enhancement model. The model enables useful features and useless features contribute differently to the prediction, thus improves the performance of the recommendation. Furthermore, to holistically exploit the different interrelation of features, we propose a hierarchical feature interaction method which consists three layers of second-order interaction to mine the underlying correlations between users and items. Extensive experiments show that MGAR outperforms the state-of-the-art recommendation methods in terms of RMSE on Yelp and Amazon Electronics.

**Keywords:** Recommendation · Meta-graph · Heterogeneous Information Networks · Attention Model · Feature Interaction.

## 1 Introduction

In the era of information explosion, recommender systems, which help users discover items by interests, play a pivotal role in various online services [1,9,10,16]. Traditional recommendation methods, based on matrix factorization, mainly focus on characterizing the user-item interaction while neglecting abundant auxiliary data. However, as auxiliary data is likely to contain useful information, neglecting it may degrade the performance of recommendation. To solve this drawback, heterogeneous information network (HIN), consisting of multiple types

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_41

of entities and relations, has been adopted in recommender systems to flexibly model the heterogeneity of rich auxiliary data. As an emerging topic, HINs based recommendation methods have attracted intensive research interests.

Existing HINs based recommendation approaches have evolved mainly through two dimensions: meta-path based recommendation models [5,13] and meta-graph based recommendation methods [18]. The former predefine a set of meta-paths for generating the semantic relatedness constrained by entity types. Then they treat the semantic relatedness as latent features for recommendation. This category of models, however, can only express simple relationship between two different types of entities that hardly capturing complicated semantics. Meta-graph based recommendation methods, in contrast, leverage meta-graph (or meta-structure) [6, 17] to generate the complex relevance of two entities, and thus can capture more complex semantics that meta-path cannot. Although these methods have achieved performance improvement to some extent, they still face two major challenges. First, most existing methods simply treat all latent features indiscriminately for recommendation. However, as not all latent features are equally useful, the useless features may even introduce noises and degrade the performance. Furthermore, as the way of assembling features has significant influence on the accuracy of recommendation, factorization machine (FM) [4, 10, 11], which leverages the second-order feature interaction, is widely adopted in the recommender systems. Nevertheless, most existing FM based recommendations are hindered by modelling all feature interactions roughly in the same way, neglecting the fact that not all interactions are equally predictive.

To address the above challenges, we propose a Meta-Graph based Attention-aware Recommendation (MGAR) method. Fig. 2 illustrates the overall view of our proposed method which utilizes rich meta-graph based latent features to make heterogeneous information fusion recommendation. Firstly, we propose a feature extracting module to generate different features. Furthermore, we propose an attention-based feature enhancement module to learn the importance of the latent features generated by different meta-graphs. The useless features are learned to set a lower weight as they may introduce noises and degrade the performance, while the useful features are enhanced. Finally, a hierarchical feature interaction method is introduced for recommendation, which considers three layers of second-order feature interaction. The method can mine the underlying correlations between users and items so that the features could be efficiently assembled and thoroughly used to predict the rating for recommendation. Our contributions can be summarized as follows:

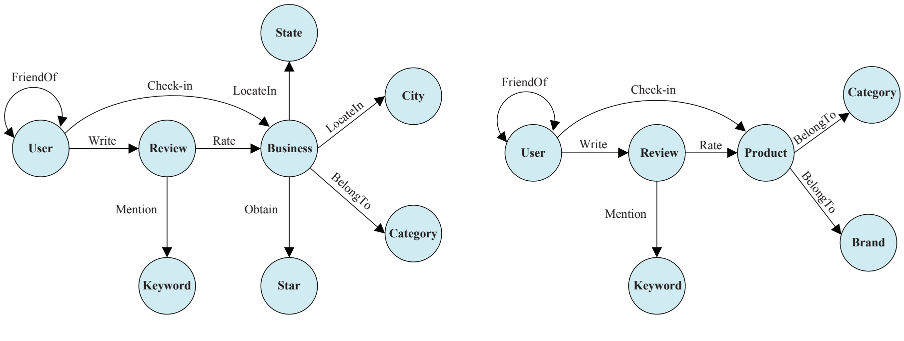
- We propose a novel attention-based feature enhancement module to learn the importance of different latent features for enabling feature interactions contribute differently to the prediction.
- We design a hierarchical feature interaction method to make full use of the generated features so that the performance of recommendation could be improved.
- Extensive experiments conducted on two widely used datasets demonstrate that MGAR outperforms several state-of-the-art recommendation methods.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_41

## 2 Preliminary

In this section, we introduce the definitions of heterogeneous information network [14] and meta-graph [18] which are frequently used in this paper.

**Definition 1.** ***Heterogeneous Information Network.*** *An* ***information network*** *is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node type mapping $\varphi : \mathcal{V} \to \mathcal{A}$ and an edge type mapping $\phi : \mathcal{E} \to \mathcal{R}$. Particularly, when the number of node types $|\mathcal{A}| > 1$ or the number of edge types $|\mathcal{R}| > 1$, the network is called a* ***heterogeneous information network*** *(HIN).*



(a) Yelp

(b) Amazon Electronics

**Fig. 1.** Examples of heterogeneous information network schema.

*Example 1.* HIN contains diverse node types and edge types. As shown in Fig. 1, the dataset of Yelp consists of 8 types of nodes (User, Review, Business, City, State, Star, Category, Keyword) and 8 types of edges (FriendOf, Write, Rate, Check-in, Mention, LocateIn, Obtain, BelongTo). Similarly, the dataset of Amazon Electronics consists of 6 types of nodes and 6 types of edges as well.

**Definition 2.** ***Meta-graph.*** *A meta-graph $M$ is a directed acyclic graph (DAG) with a single source node $n_s$ and a single target node $n_t$, defined on an HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with schema $\mathcal{T}_{\mathcal{G}} = (\mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ is the node set, $\mathcal{E}$ is the edge set, $\mathcal{A}$ is the node type set, and $\mathcal{R}$ is the edge type set. Then we define a meta-graph $M = (\mathcal{V}_M, \mathcal{E}_M, \mathcal{A}_M, \mathcal{R}_M, n_s, n_t)$, where $\mathcal{V}_M \subseteq \mathcal{V}$, $\mathcal{E}_M \subseteq \mathcal{E}$, constrained by $\mathcal{A}_M \subseteq \mathcal{A}$ and $\mathcal{R}_M \subseteq \mathcal{R}$, respectively.*

*Example 2.* The example of meta-graph is shown in Fig. 3. It is a schema both in Yelp and Amazon Electronics datasets. We can see that the meta-graph is a DAG with U (User) as the source node and I (Business in Yelp and Product in Amazon Electronics) as the target node.

## 3 The proposed model

Given HIN containing $m$ users and $n$ items, the goal of the proposed method is to predict the user-item rating matrix, denoted as $\hat{R} \in \mathbb{R}^{m \times n}$. As shown in Fig. 2, our framework consists of three main phases. 1) Feature extractor: in order to generate different features, we propose a feature extractor which not only learns the user and item embeddings but also generates the latent features based on different meta-graphs. 2) Attention-based feature enhancement: to learn the importance of the generated latent features, we design a two-layer attention network to set a lower weight on the useless features, while the useful features are enhanced. 3) Hierarchical feature interaction for rating: to efficiently assemble and thoroughly use the features for rating, it considers three layers of second-order feature interaction which could mine the underlying correlations between users and items.

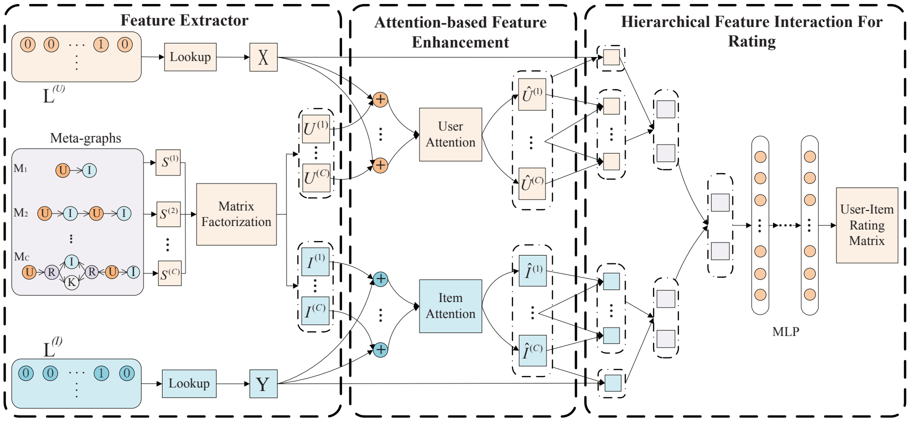


**Fig. 2.** The overall architecture of the proposed model.

### 3.1 Feature Extractor

**User and Item Embedding.** To learn the representation of users and items in HIN, we leverage the lookup-based embedding method following [5]. Firstly, we represent the users and items by one-hot encodings, denoted as $L^{(U)} \in \mathbb{R}^{m\times m}$ and $L^{(I)} \in \mathbb{R}^{n\times n}$ individually. As the one-hot encodings may result in corse of dimensionality and enormous sparsity, we then adopt the lookup operation in projecting them into dense embeddings. We use $X = \{x\}^m$ and $Y = \{y\}^n$ denote the embeddings of users and items, where $x, y \in \mathbb{R}^d$ denotes the embedding of user $u$ and item $i$ with $d$ denotes the embedding dimensionality ($d \ll m$, $d \ll n$).

**Meta-graph based Latent Features Generating.** In order to thoroughly explore the semantic information between users and items, we learn their latent features based on different meta-graphs. Following [18], we first compute the user-item similarity matrices and then generate the latent features of users and items. More specifically, we compute the similarity matrix of the meta-graph by adjacency matrix. Take Fig. 3 as an example, the user-item similarity matrix is calculated by:

$$S = W_{UR} \cdot ((W_{RI} \cdot W_{RI}^T) \odot (W_{RK} \cdot W_{RK}^T)) \cdot W_{UR}^T \cdot W_{UI}, \tag{1}$$

where $\odot$ is the Hadamard product and $W_*$ is the adjacency matrix between two different node types. Following similarity steps, by designing C different meta-graphs, we can get C different user-item similarity matrices, denoted by $\{S^{(i)}\}^C$. In addition, we use matrix factorization (MF) to generate the latent features of users and items from the user-item similarity matrix $S^{(i)}$. First, we let $U^{(i)}$ and $I^{(i)}$ denote the $i$-th user-specific matrix and item-specific matrix, respectively. Then, MF factorizes $S^{(i)}$ into two low-rank user-specific and item-specific matrices ($U^{(i)} \in \mathbb{R}^{m \times d}$ and $I^{(i)} \in \mathbb{R}^{n \times d}$, $i \in \{1, 2, ..., C\}$). MF model maps both user and item to a joint latent factor space of dimensionality $d$, such that user-item interactions are modeled as inner products in that space and the interaction can be captured by $S^{(i)} = U^{(i)} I^{(i)T}$. As the sparsity of $S^{(i)}$, we only model the observed interaction while avoiding overfitting through a regularized model. Therefore, we can learn $U^{(i)}$ and $I^{(i)}$ by:

$$\min_{U^{(i)}, I^{(i)}} \frac{1}{2} ||S^{(i)} - U^{(i)} I^{(i)T}||_d^2 + \frac{\lambda_{U^{(i)}}}{2} ||U^{(i)}||_d^2 + \frac{\lambda_{I^{(i)}}}{2} ||I^{(i)}||_d^2, \tag{2}$$

where $\lambda_{U^{(i)}}$ and $\lambda_{I^{(i)}}$ are hyper-parameters that used to avoid overfitting. As obtaining C different user-item similarity matrices before, we got C user-specific matrices (latent feature matrices of users) and C item-specific matrices (latent feature matrices of items), denoted as $\{U^{(i)}\}^C$, $\{I^{(i)}\}^C$.

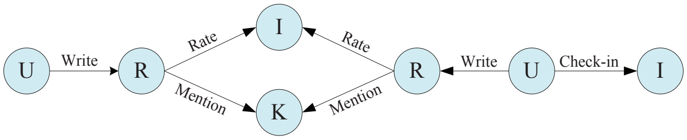


**Fig. 3.** An example of meta-graph. (U: user, R: review, I: item, K: keyword)

### 3.2 Attention-based Feature Enhancement

As different latent features of user and item obtained from different meta-graphs contribute differently to recommendation, lacks of differentiating the importance of latent features may result in suboptimal prediction. To solve this problem, we propose a novel attention-based feature enhancement module to discriminate the importance of different latent features. The flowchart of the module is shown in Fig. 4. Specifically, let $E^{(i)} \in \mathbb{R}^{m \times d}$ and $F^{(i)} \in \mathbb{R}^{n \times d}$ denote the attention weight matrices of user and item latent features generated by the $i$-th meta-graph, respectively. The corresponding weighted user latent features $\hat{U}^{(i)}$ and weighted item latent features $\hat{I}^{(i)}$ can be given as follows:

$$\hat{U}^{(i)} = E^{(i)} \odot U^{(i)}, \tag{3}$$

$$\hat{I}^{(i)} = F^{(i)} \odot I^{(i)}, \tag{4}$$

where $i \in \{1, 2, ..., C\}$ and the attention weight matrix $E^{(i)}$ is learned by a two-layer attention network:

$$E_1{}^{(i)} = ReLU(W_1{}^{(i)}(X \oplus U^{(i)}) + b_1{}^{(i)}), \tag{5}$$

$$E_2{}^{(i)} = ReLU(W_2{}^{(i)}E_1{}^{(i)} + b_2{}^{(i)}), \tag{6}$$

where $W_*{}^{(i)}$ and $b_*{}^{(i)}$, $* \in \{1, 2\}$ denote the weight matrix and the bias vector for the $*$-th fully connected layer, $\oplus$ denotes the feature concatenation operation. Since the range of the attention weights is uncertain, we restrict the distance to (0,1) by normalizing the above attention weight matrix with the softmax function:

$$E^{(i)} = \frac{\exp(E_2^{(i)})}{\sum\limits_{c=1}^{C} \exp(E_2{}^{(c)})}. \tag{7}$$

We learn the attention weight matrix $F^{(i)}$ with the same way of $E^{(i)}$, so that we do not repeat here.

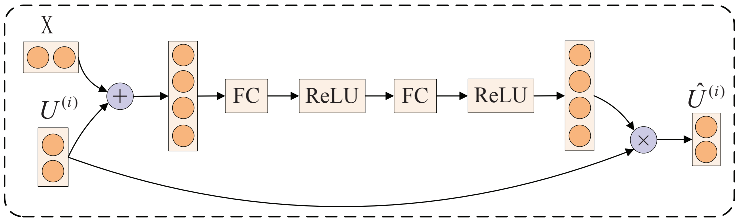


**Fig. 4.** Demonstration of the attention-based feature enhancement module.

### 3.3 Hierarchical Feature Interaction For Rating

To efficiently assemble and thoroughly use the features for rating, we design a hierarchical feature interaction method with factorization machine to model complicated interactions between user $u$, item $i$ and their weighted latent features. The predicted rating of user $u$ on item $i$ is defined as:

$$\hat{R}_{u,i} = w_0 + \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d} \hat{w}_j \hat{U}_{u,j} + \sum_{j=1}^{d} w'_j y_j + \sum_{j=1}^{d} \hat{w}'_j \hat{I}_{i,j} + H_{u,i}, \tag{8}$$

where $w_0$ is the global bias. $w_j$, $\hat{w}_j$, $w'_j$ and $\hat{w}'_j$ represent the weight for $x_j$, $\hat{U}_{u,j}$, $y_j$ and $\hat{I}_{i,j}$, respectively. In which, $x_j$ denotes the $j$-th feature of user $u$, $\hat{U}_{u,j}$ denotes the $j$-th weighted latent feature of user $u$, $y_j$ denotes the $j$-th feature of item $i$ and $\hat{I}_{i,j}$ denotes the $j$-th weighted latent feature of item $i$. Furthermore, $\hat{U}_{u,j} \in \hat{U}$ and $\hat{U} = \sum\limits_{i=1}^{C} \hat{U}^{(i)}$, $\hat{I}_{i,j} \in \hat{I}$ and $\hat{I} = \sum\limits_{j=1}^{C} \hat{I}^{(j)}$. $H_{u,i}$ is the

DOI: 10.1007/978-3-030-22741-8_41

result of interactions between user $u$, item $i$ and their weighted latent features. $H_{u,i} \in H$ and $H$ denotes the result of interactions between users, items and their weighted latent features which is the core component for modelling hierarchical feature interactions. In order to obtain $H$, we first consider to model the pairwise correlation between users and their weighted latent features, and all nested correlations among weighted latent features. Therefore, the user representations $\hat{X}$ can be defined as follows:

$$\hat{X} = \sum_{i=1}^{C} X \odot \hat{U}^{(i)} + \sum_{i=1}^{C} \sum_{j=i+1}^{C} \hat{U}^{(i)} \odot \hat{U}^{(j)}. \tag{9}$$

By employing the same operation on items and their weighted latent features, we can similarly define the item representations $\hat{Y}$ as follows:

$$\hat{Y} = \sum_{i=1}^{C} Y \odot \hat{I}^{(i)} + \sum_{i=1}^{C} \sum_{j=i+1}^{C} \hat{I}^{(i)} \odot \hat{I}^{(j)}. \tag{10}$$

Then, we employ a stack of fully connected layers to capture the correlation among users and items. We first merge user representations $\hat{X}$ and item representations $\hat{Y}$ with the Hadamard product, which models the two-way interaction between users and items. We then place a multi-layer perception (MLP) above the Hadamard product. By specifying non-linearity activation function (ReLU) which is more biologically plausible and proven to be non-saturated [3, 12], we allow the model to learn higher-order feature interactions in a non-linearity way. The definition of fully connected layers is as follows:

$$\begin{cases} Z_1 = ReLU(W_1(\hat{X} \odot \hat{Y}) + b_1), \\ Z_2 = ReLU(W_2 Z_1 + b_2), \\ \quad\quad ...... \\ Z_t = ReLU(W_t Z_{t-1} + b_t), \end{cases} \tag{11}$$

where $\mathrm{W}_t$ and $b_t$ denote the weight matrix and bias vector of the $t$-th layer, respectively. At last, the final result of interactions between users, items and their weighted latent features is defined as:

$$H = W_Z Z_t, \tag{12}$$

where $W_Z$ denote the neuron weights.

### 3.4 Optimization

To estimate the parameters of MGAR, we specify an objective function $L$. The overall objective to be optimized is defined as:

$$L = \sum\nolimits_{(u,i)\in\tau} \left(R_{u,i} - \hat{R}_{u,i}\right)^2 + \lambda ||W||^2, \tag{13}$$

where $\tau$ denotes the set of training instances, the target $R_{u,i}$ is a real value in our datasets, $\lambda$ controls the strength of regularization and W is the set of weights in our implementation. Moreover, we use Adam as the optimizer so that the learning rate can be self adapted during the training phase.

DOI: 10.1007/978-3-030-22741-8_41

**Overfitting Prevention.** Besides the regularization, we also use dropout technique to prevent overfitting as neural network models are easy to overfit on training data. We employ dropout in attention networks and randomly drop $\rho$ percent of latent factors, where $\rho$ is termed as the dropout ratio.

**Batch Normalization.** One difficulty of training multi-layered neural networks is caused by the fact of covariance shift [7]. If the parameters of the previous layer change, the distribution of the later layer's inputs needs to change accordingly. To address this problem, we employ batch normalization (BN), which normalizes the input to a zero-mean unit-variance Gaussian distribution for each training mini-batch. It can lead to faster convergence and better performance in MGAR.

## 4 Experiments

In order to evaluate the effectiveness of the proposed method, MGAR, we conduct extensive experiments comparing with several state-of-the-art recommendation methods on two widely used datasets. The experiments are implemented with TensorFlow and run in a Linux server with NVIDIA Tesla M40 GPU.

### 4.1 Datasets and Evaluation Metric

**Datasets.** Experiments are conducted on two datasets, Yelp and Amazon Electronics, which are provided in [18] and have rich heterogeneous information. The detailed description of our datasets is shown in Table 1. For Yelp[1] which has 191,506 user-item interaction records, it contains 8 entity types and 9 relations. There are many domains in Amazon[2] and we use the electronics domain for our experiments. For Amazon Electronics which has 183,807 user-item interaction records, it contains 6 entity types and 6 relations. Moreover, ratings in our two datasets are from 1 to 5.

**Evaluation Metric.** To evaluate the performance of the recommendation, we adopt Root Mean Square Error (RMSE) as evaluation metric where a lower RMSE score indicates a better performance. It is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_{test}} (R_{u,i} - \hat{R}_{u,i})^2}{|R_{test}|}}, \tag{14}$$

where $R_{test}$ is the set of all user-item pairs $(u, i)$ in the test set, and $R_{u,i}$ is the observed rating in the test set.

[1] https://www.yelp.com/dataset challenge
[2] http://jmcauley.ucsd.edu/data/amazon/

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_41

**Table 1.** The detailed description of Yelp and Amazon Electronics datasets.

| Datasets | Relations (A-B) | Number of A | Number of B | Number of (A-B) |
|---|---|---|---|---|
| Yelp | User-Business | 36,105 | 22,496 | 191,506 |
| | User-Review | 36,105 | 191,506 | 191,506 |
| | Review-Business | 191,506 | 22,496 | 191,506 |
| | Review-Keyword | 191,506 | 10 | 955,041 |
| | User-User | 17,065 | 17,065 | 140,344 |
| | Business-Category | 22,496 | 869 | 67,940 |
| | Business-City | 22,496 | 215 | 22,496 |
| | Business-State | 22,496 | 18 | 22,496 |
| | Business-Star | 22,496 | 9 | 22,496 |
| Amazon Electronics | User-Product | 59,297 | 20,216 | 183,807 |
| | User-Review | 59,297 | 183,807 | 183,807 |
| | Review-Product | 183,807 | 20,216 | 183,807 |
| | Review-Keyword | 183,807 | 10 | 796,392 |
| | Product-Category | 20,216 | 682 | 87,587 |
| | Product-Brand | 9,533 | 2,015 | 9,533 |

### 4.2 Experimental settings

To evaluate the effectiveness of our method, we compare it with several traditional recommendation methods and deep learning models. They are introduced as follows:

- **MF** [8]: Matrix factorization is a traditional model for recommendation which only uses the user-item rating matrix.
- **LibFM** [11]: LibFM has shown its strong performance for personalized recommendation and rating prediction task which is released by Rendle[3]. In this paper, we only use the user-item rating matrix for comparison.
- **FMG** [18]: FMG is designed for rating prediction which gets latent features of users and items by factorizing the similarity matrices of different meta-graphs. It simply concatenates all the latent features and embeddings as the input features.
- **AFM** [2]: This is an attention-driven factor model which integrates item features driven by user's attention.
- **NFM** [4]: NFM combines neural network with factorization machine for rating prediction. It converts each log (i.e., user ID, item ID and all context variables) to a feature vector by using one-hot encoding.

To demonstrate the capability of our model, we randomly split the datasets into training set (80%) and testing set (20%). For fairly comparing the performance of our model, we use the meta-graphs designed in [18] that the number of meta-graph types in Yelp and Amazon Electronics is 9 and 6, respectively. We randomly initialize the parameters with a Gaussian distribution, where the mean and the standard deviation is 0 and 0.01, respectively. We trained the proposed

[3] http://www.libfm.org/

DOI: 10.1007/978-3-030-22741-8_41

model for 6000 batches in total and the batch size was set to 512. The learning rate was searched in [0.001, 0.003, 0.005, 0.1].

### 4.3 Comparison with existing models

The RMSE results on Yelp and Amazon Electronics datasets are shown in Table 2 which presents the best effect achieved by MGAR and comparative methods. From Table 2, we have the following observations:

- MF and LibFM have worse results than FMG. The reason is that FMG employs HINs based meta-graphs to characterize the rich semantic information while MF and LibFM only use the user-item rating matrix which have insufficient information.
- The performance of AFM is better than FMG. FMG simply concatenates all the corresponding features as the input of rating prediction without differentiating the importance of the latent features generated by different meta-graphs. Compared with FMG, AFM learns what feature of an item a user cares most about which demonstrates the design of attention-based feature enhancement module in our work is significant.
- NFM obtains a better performance than other baselines. It seamlessly combines the linearity of factorization machine in modelling second-order feature interactions and the non-linearity of neural network in modelling higher-order feature interactions which improves the result effectively.
- MGAR observably outperforms all baselines in all cases and well demonstrates its effectiveness. The reason is that MGAR proposes not only a novel attention-based feature enhancement module to discriminate the importance of different latent features but also a hierarchical feature interaction method with factorization machine to efficiently assemble and thoroughly use the features for rating prediction.

**Table 2.** The RMSE results on Yelp and Amazon Electronics of MGAR and all baseline methods. The best results are in boldface. Percentages in the brackets are the reduction of RMSE comparing MGAR with the corresponding approaches in the table header.

| Methods | Yelp | Amazon Electronics |
|---|---|---|
| MF [8] | 2.5141 (+53.7%) | 2.9656 (+61.8%) |
| LibFM [11] | 1.7637 (+34.0%) | 1.3462 (+15.9%) |
| FMG [18] | 1.2456 (+6.6%) | 1.1864 (+4.6%) |
| AFM [2] | 1.2193 (+4.6%) | 1.1681 (+3.1%) |
| NFM [4] | 1.1988 (+2.9%) | 1.1627 (+2.6%) |
| **MGAR** | **1.1637** | **1.1322** |

### 4.4 Further Analysis on MGAR

**Ablation Study.** To verify the impact of different modules on the performance of MGAR, we design three diverse baselines for experiments: a) MGAR-att is built by removing the attention-based feature enhancement module of MGAR; b) MGAR-inter is built by removing the hierarchical feature interaction module; c) MGAR-fm is built by removing the factorization machine. The results are shown in Table 3. It can be observed that MGAR is most affected by the attention-based feature enhancement module. It proves that latent features of different meta-graphs make different contributions to prediction. Furthermore, the hierarchical feature interaction module also plays a key role as it could holistically exploit the different interrelations of features. In addition, the factorization machine improves the performance of MGAR, to some extent, as the way of assembling features has significant influence on the accuracy of recommendation.

**Table 3.** The RMSE results on Yelp and Amazon Electronics for ablation analysis.

| Method | Yelp | Amazon Electronics |
|---|---|---|
| MGAR-att | 1.4974 | 1.2142 |
| MGAR-inter | 1.3041 | 1.1816 |
| MGAR-fm | 1.2532 | 1.1934 |
| **MGAR** | **1.1637** | **1.1322** |

**Impact of MLP with different layers.** We also verify the impact of MLP with different fully connected layers. MLP-0, MLP-1, MLP-2, MLP-3 and MLP-4 denote MLP with $0 \sim 4$ layers, respectively. In which, MLP-0 denotes we only merge user representations and item representations with the Hadamard product which has no fully connected layer. The results are shown in Table 4. We can observe that the performance of MLP-0 is worse than MLP-1 as MLP-0 only learns the linearity interaction of second-order features and neglects non-linearity interaction of higher-order features which is important in capturing useful information for rating prediction. Furthermore, we can find that more layers do not improve the performance and the best one is only one layer. The reason is that the hierarchical feature interaction module has encoded informative second-order feature interactions, and based on which, a simple non-linearity function is sufficient to capture higher-order feature interactions.

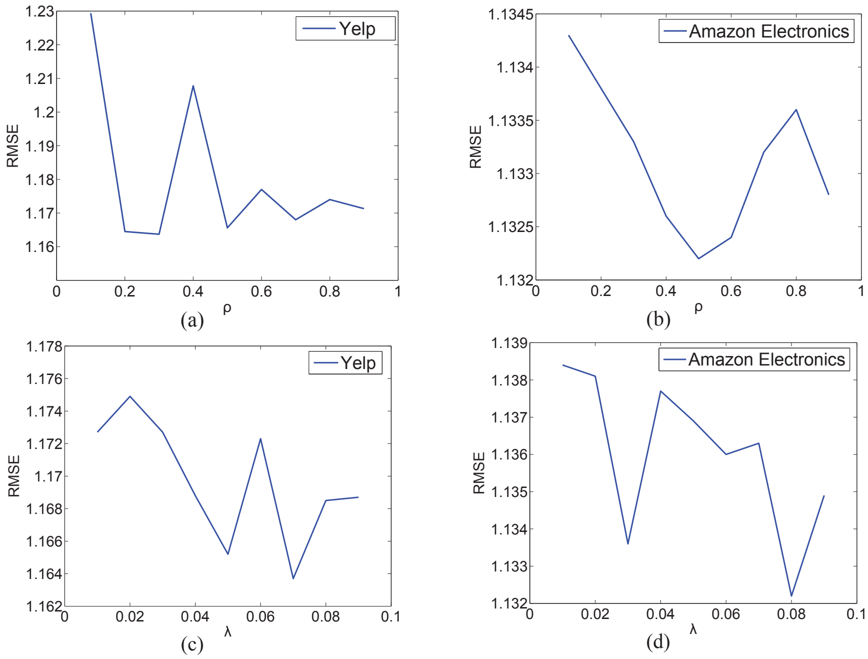
**Effect of model Parameters.** We conducted experiments to further analyse the influence of parameters $\rho$ and $\lambda$ on performance. The dropout ratio $\rho$ is used to prevent MGAR from overfitting. The trade-off parameter $\lambda$ controls the regularization strength as Eq.(13) shows. The RMSE results on Yelp and Amazon Electronics are shown in Fig. 5. We can observe that MGAR is sensitive to $\rho$ and $\lambda$. A too small or too large value may lead to a bad learning of the model.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22741-8_41

**Table 4.** Impact of MLP with different layers.

| Method | Yelp | Amazon Electronics |
|---|---|---|
| MLP-0 | 1.1678 | 1.1336 |
| **MLP-1** | **1.1637** | **1.1322** |
| MLP-2 | 1.1655 | 1.1331 |
| MLP-3 | 1.1673 | 1.1337 |
| MLP-4 | 1.1709 | 1.1345 |

The best result is obtained with $\rho = 0.3$ and $\lambda = 0.07$ on Yelp, $\rho = 0.5$ and $\lambda = 0.08$ on Amazon Electronics.



(a)

(b)

(c)

(d)

**Fig. 5.** Parameter sensitivity analysis of $\rho$ and $\lambda$ on performance.

## 5 Related work

Over the last few years, various recommendation methods have been proposed, which can be roughly divided into traditional methods [1, 10] and HINs based methods [13, 16, 18]. In traditional methods, they only use historical interactions for recommendation. For example, MF [8] is a standard collaborative filtering method which only uses the user-item rating matrix with $L_2$ regularization. LibFM [11] only models the latent feature interactions of users and items for personalized recommendation and rating prediction.

DOI: 10.1007/978-3-030-22741-8_41

Since traditional methods can only use specific information and their performances are not well, many works attempt to leverage additional information for recommendation, such as heterogeneous information [6, 13]. As a newly emerging direction, heterogeneous information network can naturally model complex entities and rich relations in recommender system. HeteRec [16] measures similarity between users and items by meta-paths and the weighted ensemble model is learned from the latent features of users and items. FMG [18] employs metagraphs to characterize the rich semantic information so that it has strong expressive ability. However, their performance is unsatisfactory as they cannot fully utilize the captured features by a simple feature interaction.

To fully use the captured features, some works [3,12,15] start to explore deep learning techniques for recommendation which have achieved immense success and been widely used on speech recognition, computer vision and natural language processing. NFM [4] combines neural network with factorization machine to learn linearity interaction of second-order features and non-linearity interaction of higher-order features. AFM [2] is used to integrate item features driven by user's attention. Compared with them, our approach uses attention networks to distinguish the importance of different latent features and adopts a novel hierarchical feature interaction method to model complicated feature interactions so that the performance of MGAR has greatly improvement.

## 6 Conclusion

We have presented a novel recommendation method over heterogeneous information networks, named Meta-Graph based Attention-aware Recommendation (MGAR). It enhances the useful features by discriminating the importance of different latent features with the attention-based feature enhancement model. Furthermore, we predict the rating by modelling the complicated feature interactions with a hierarchical feature interaction method. Experiments on Yelp and Amazon Electronics show that MGAR outperforms several state-of-the-art recommendation algorithms.

**Acknowledgement.** This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDC02050200) and Beijing Municipal Science and Technology Project (Z181100002718004).

## References

1. Anyosa, S.C., Vinagre, J., Jorge, A.M.: Incremental matrix co-factorization for recommender systems with implicit feedback. In: Proceedings of the 27th International Conference on World Wide Web. pp. 1413–1418 (2018)
2. Chen, J., Zhuang, F., Hong, X., Ao, X., Xie, X., He, Q.: Attention-driven factor model for explainable personalized recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 909–912 (2018)

3. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 191–198 (2016)
4. He, X., Chua, T.S.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 355–364 (2017)
5. Hu, B., Shi, C., Zhao, W.X., Yu, P.S.: Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1531–1540 (2018)
6. Huang, Z., Zheng, Y., Cheng, R., Sun, Y., Mamoulis, N., Li, X.: Meta structure: Computing relevance in large heterogeneous information networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1595–1604 (2016)
7. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 1–9 (2015)
8. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer (8), 30–37 (2009)
9. Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., Sun, G.: xdeepfm: Combining explicit and implicit feature interactions for recommender systems. arXiv preprint arXiv:1803.05170 (2018)
10. Liu, H., He, X., Feng, F., Nie, L., Zhang, H.: Discrete factorization machines for fast feature-based recommendation. arXiv preprint arXiv:1805.02232 (2018)
11. Rendle, S.: Factorization machines with libfm. ACM Transactions on Intelligent Systems and Technology (TIST) **3**(3), 57 (2012)
12. Shan, Y., Hoens, T.R., Jiao, J., Wang, H., Yu, D., Mao, J.: Deep crossing: Web-scale modeling without manually crafted combinatorial features. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 255–262 (2016)
13. Shi, C., Zhang, Z., Luo, P., Yu, P.S., Yue, Y., Wu, B.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 453–462 (2015)
14. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment **4**(11), 992–1003 (2011)
15. Xu, J., He, X., Li, H.: Deep learning for matching in search and recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1365–1368 (2018)
16. Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., Han, J.: Personalized entity recommendation: A heterogeneous information network approach. In: Proceedings of the 7th ACM international conference on Web Search and Data Mining. pp. 283–292 (2014)
17. Zhang, D., Yin, J., Zhu, X., Zhang, C.: Metagraph2vec: Complex semantic path augmented heterogeneous network embedding. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 196–208 (2018)
18. Zhao, H., Yao, Q., Li, J., Song, Y., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 635–644 (2017)

DOI: 10.1007/978-3-030-22741-8_41