Aspect-Level Deep Collaborative Filtering via Heterogeneous Information Networks

Xiaotian Han[1], **Chuan Shi**[1*], **Senzhang Wang**[2], **Philip S. Yu**[3,4], **Li Song**[1]
[1] Beijing University of Posts and Telecommunications
[2] Nanjing University of Aeronautics and Astronautics
[3] University of Illinois at Chicago
[4] Institute for Data Science, Tsinghua University
{hanxiaotian.h,song200626}@gmail.com, shichuan@bupt.edu.cn,
szwang@nuaa.edu.cn, psyu@cs.uic.edu

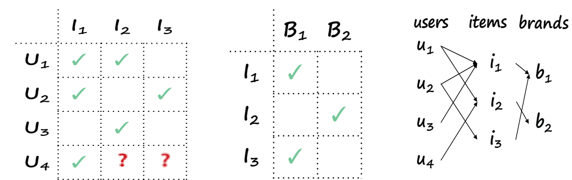
Abstract

Latent factor models have been widely used for recommendation. Most existing latent factor models mainly utilize the rating information between users and items, although some recently extended models add some auxiliary information to learn a unified latent factor between users and items. The unified latent factor only represents the latent features of users and items from the aspect of purchase history. However, the latent features of users and items may stem from different aspects, e.g., the brand-aspect and category-aspect of items. In this paper, we propose a Neural network based Aspect-level Collaborative Filtering model (NeuACF) to exploit different aspect latent factors. Through modelling rich objects and relations in recommender system as a heterogeneous information network, NeuACF first extracts different aspect-level similarity matrices of users and items through different meta-paths and then feeds an elaborately designed deep neural network with these matrices to learn aspect-level latent factors. Finally, the aspect-level latent factors are effectively fused with an attention mechanism for the top-N recommendation. Extensive experiments on three real datasets show that NeuACF significantly outperforms both existing latent factor models and recent neural network models.

1 Introduction

Currently the overloaded online information overwhelms users. In order to tackle the information overload problem, Recommender Systems (RS) are widely employed to guide users in a personalized way of discovering products or services they might be interested in from a large number of possible alternatives. Due to its importance in practice, recommender systems have been attracting remarkable attention in both industry and academic research community.

*Corresponding author.



(a) User-Item purchase relation (b) Item-Brand relation (c) heterogeneous network

Figure 1: A toy example of aspect-level interactions between users and items.

Collaborative Filtering (CF) [Hu *et al.*, 2008] is the most popular method for recommendation, whose basic assumption is that people who share similar purchase in the past tend to have similar choices in the future. In order to exploit users' similar purchase preference, latent factor models (e.g., matrix factorization) [Koren *et al.*, 2009; Koren, 2008] have been proposed, which usually factorize the user-item interaction matrix (e.g., rating matrix) into two low-rank user-specific and item-specific factors, and then use the low-rank factors to make predictions. Since latent factor models may suffer from data sparsity, many extended latent factor models integrate auxiliary information under the matrix factorization framework, such as social recommendation [Ma *et al.*, 2008] and heterogeneous network based recommendation [Shi *et al.*, 2016]. Recently, with the surge of deep learning, deep neural networks are also employed to deeply capture the latent features of users and items for recommendation. NeuMF [He *et al.*, 2017] replaces the inner product operations in matrix factorization with a multi-layer feed-forward neural network to capture the nonlinear relationship between users and items. DMF [Xue *et al.*, 2017] uses the rating matrix directly as the input and maps user and items into common low-dimensional space via a deep neural network.

Although these latent factor models achieve good performance, we find that they usually only capture the information of purchase history. Existing models usually focus on exploiting latent factors of users and items through their interaction information (especially rating information), which only reflects user preferences and item characteristics from one aspect, i.e., purchase history. However, the latent factors of users and items usually stem from different aspects in real applications. These different aspect-level features can more comprehensively reflect user preferences and item characteristics. Thus the latent factor models should exploit latent features of users and items from different aspects. Figure 1 shows a toy example of our idea. If we only exploit the interaction matrix (illustrating purchase history) in Figure 1a, we may infer that user U_4 will purchase item I_2 and I_3. However, when considering the item brand information shown in Figure 1b, we may find item I_3 is a better recommendation to U_4 because items I_1 and I_3 belong to the same brand B_1.

Although it is promising to comprehensively utilize multiple aspect-level latent features of users and items, it still faces the following two challenges. (1) How to extract different aspect-level features. We need to effectively organize the different types of objects and interactions in RS. The extracted aspect-level features should reflect different aspects of users preferences and embody rich semantics. (2) How to learn and fuse latent factors from different aspects. Even if we can extract different aspect-level features, it is still not easy to learn their latent factors and effectively fuse them. Although matrix factorization is an option, it only learns the "shallow" factors. Deep neural network is a promising method, while we still need to design proper network structure and fusing mechanism for our problem setting.

In this paper, to address the challenges above, we propose a novel Neural network based Aspect-level Collaborative Filtering model (NeuACF). NeuACF can effectively model and fuse different aspect-level latent factors which represent the user preferences and item characteristics from different aspects. Particularly, the objects and interactions of different types in RS are organized as a Heterogeneous Information Network (HIN) [Shi *et al.*, 2017b]. Meta-paths [Sun *et al.*, 2011], relation sequences connecting objects, are employed to extract features of users and items in different aspects. As an example shown in Figure 1c, we can extract the latent factors of users from the aspect of purchase history with the $User$-$Item$-$User$ path, which is usually analyzed by existing latent factor models. Furthermore, we design a delicate deep neural network to learn different aspect-level latent factors for users and items and utilize an attention mechanism to effectively fuse them for the top-N recommendation. Note that, different from focusing on the rating information with the auxiliary information in those hybrid recommendation models [Wang *et al.*, 2015], NeuACF treats different aspect-level latent factors extracted with meta-paths equally, and automatically determines the importance of these aspects. NeuACF is also different from those HIN based methods [Yu *et al.*, 2014; Shi *et al.*, 2017a] in its deep model and fusing mechanism. Extensive experiments illustrate the effectiveness of NeuACF, as well as the traits of aspect-level latent factors.

2 Preliminaries

2.1 Latent Factor Model

The latent factor model has been widely studied in recommender system. Its basic idea is to map users and items to latent factors and use these factors for recommendation. The representative works are Matrix Factorization (MF) [Koren *et al.*, 2009], PMF [Mnih and Salakhutdinov, 2008] and SVD++ [Koren, 2008]. Taking the MF for example, the objective function of MF in Equation 1 aims to minimize the regularized squared loss on the observed ratings:

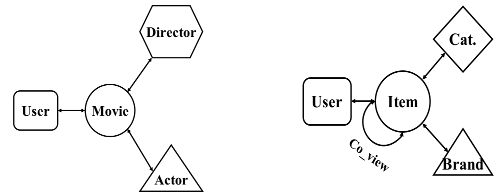
$$\arg\min_{\boldsymbol{u},\boldsymbol{v}} \sum_i \sum_j (\boldsymbol{R}_{i,j} - \boldsymbol{u}_i^T \boldsymbol{v}_j)^2 + \lambda \left(\sum_i ||\boldsymbol{u}_i||_2^2 + \sum_j ||\boldsymbol{v}_j||_2^2 \right) \tag{1}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ denote the latent factors of user U_i and item I_j. λ controls the strength of regularization, which is usually a L-2 norm aiming to prevent overfitting.

Based on this basic MF framework, many extended latent factor models have been proposed through adding some auxiliary information, such as social recommendation [Ma *et al.*, 2008] and heterogeneous network based recommendation [Shi *et al.*, 2015]. The limitation of existing latent factor models is that the latent factors are mainly extracted from one aspect, i.e., the rating matrix. However, some other more fine-grained aspect-level user-item interaction information is largely ignored, although such information is also useful.

2.2 Heterogeneous Information Network

The recently emerging HIN [Shi *et al.*, 2017b] is a good way to model complex objects and relations in RS. Particularly, HIN is a special kind of information network, which either contains multiple types of objects or multiple types of links. The network schema of a HIN specifies the type constraints on the sets of objects and relations among the objects. Two examples used in experiments are shown in Figure 2. In addition, meta-path [Sun *et al.*, 2011], a relation sequence connecting objects, can effectively extract features of objects and embody rich semantics. For example, in Figure 2b, the meta-path $User$-$Item$-$User$ (UIU) extracts the features of users in the purchase history aspect, which means users having the same purchase records. A HIN-based recommendation model was first proposed by [Yu *et al.*, 2014]. After that, several HIN based recommendations [Shi *et al.*, 2017a; Zhao *et al.*, 2017] have been proposed to utilize rich heterogeneous information in RS, while they usually focus on rating prediction with the "shallow" model.



(a) MovieLens (b) Amazon

Figure 2: Network schema of HINs for the experimental datasets.

Datasets	Aspect	Meta-Paths	
		User	Movie/Item
MovieLens	History	UMU	MUM
	Director	$UMDMU$	MDM
	Actor	$UMAMU$	MAM
Amazon	History	UIU	IUI
	Brand	$UIBIU$	IBI
	Category	$UICIU$	ICI
	Co_view	$UIVIU$	IVI

Table 1: Meta-paths used in experiments and the corresponding aspects.

3 The NeuACF Model

3.1 Model Framework

As we have discussed, existing latent factor models generally focus on learning one aspect of latent factors (e.g., rating interaction), but ignore other aspects. In this work, we propose a Neural network based Aspect-level Collaborative Filtering (NeuACF) model for the top-N recommendation. The basic idea of NeuACF is to extract different aspect-level latent features for users and items, and then learn and fuse these latent factors with deep neural network. The model contains three major steps. First, we construct a HIN based on the rich user-item interaction information in RS, and compute the aspect-level similarity matrices under different meta-paths of HIN which reflects different aspect-level features of users and items. Next, a deep neural network is designed to learn the aspect-level latent factors separately by taking these similarity matrices as inputs. Finally, the aspect-level latent factors are combined with an attention component to obtain the overall latent factors for users and items. Next we will elaborate the three steps in the following subsections.

3.2 Aspect-level Similarity Matrix Extraction

We employ HIN to organize objects and relations in RS, due to its power of information fusion and semantics representation [Shi *et al.*, 2015]. Furthermore, we utilize metapath to extract different-aspect features of users and items. Taking Figure 2b as an example, we can use UIU and IUI paths to extract features of users and items on the purchase history aspect, which is extensively exploited by existing latent factor models. In addition, we can also extract features from other aspects. Table 1 shows more aspect examples in our experimental datasets.

Given a specific meta-path, there are several alternatives to extract the aspect-level features: commuting matrix or similarity matrix. In this paper, we employ the similarity matrix based on the following reasons. (1) Similarity measure can alleviate noisy information; (2) Similarity values within the [0,1] range are more suitable for learning latent factors. We employ the PathSim [Sun *et al.*, 2011] to calculate aspect-level similarity matrices under different meta-paths in our experiments. For example, we compute the similarity matrices of user-user and item-item based on the meta-paths $UIBIU$ and IBI for the brand-aspect features respectively.

3.3 Learning Aspect-level Latent Factors

With the computed user-user and item-item similarity matrices of different aspects, we next learn their latent factors. Different from previous HIN based recommendation models,

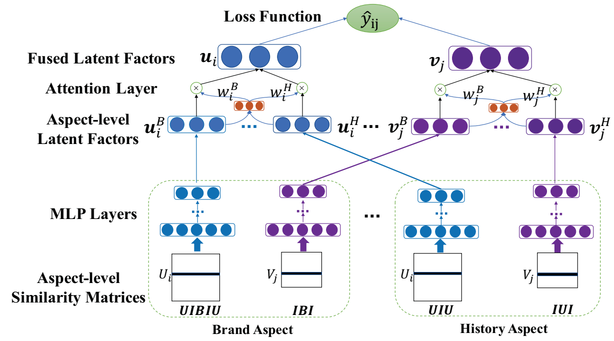


Figure 3: Deep neural network in the NeuACF model.

we design a deep neural network to learn their corresponding aspect-level latent factors separately, and the model architecture is shown in Figure 3. Concretely, for each user in each aspect, we extract the user's similarity vector from the aspect-specific similarity matrix. Then we take the similarity matrix as the input of the Multi-Layer Perceptron (MLP) and MLP learns the aspect-level latent factor as the output. The item latent factors of each aspect can be learned in a similar way.

Take the similarity matrix $S^B \in \mathbb{R}^{N \times N}$ of users under the meta-path $UIBIU$ as an example. User U_i is represented as an N-dimensional vector of S^B_{i*}, which means U_i's similarity to all the other users. Here the N means the total number of users in the datasets. The MLP projects the initial similarity vector S^B_{i*} of user U_i to a low-dimensional aspect-level latent factor. In each layer of MLP, the input vector is mapped into another vector in a new space. Formally, given the initial input vector S^B_{i*}, and the hidden layer $\boldsymbol{H}_l$ where the l is the l-th layer, the final aspect-level latent factor $\boldsymbol{u}^B_i$ can be learned through the following multi-layer mapping functions.

$$\begin{aligned}
\boldsymbol{H_0} &= \boldsymbol{S}^B_{i*} \\
\boldsymbol{H_1} &= f(\boldsymbol{W}_1^T * \boldsymbol{H}_0 + \boldsymbol{b}_1) \\
&\dots \\
\boldsymbol{H_l} &= f(\boldsymbol{W}_l^T * \boldsymbol{H}_{l-1} + \boldsymbol{b}_l) \\
&\dots \\
\boldsymbol{u}^B_i &= f(\boldsymbol{W}_n^T * \boldsymbol{H}_{n-1} + \boldsymbol{b}_n)
\end{aligned} \tag{2}$$

where $\boldsymbol{W}_i$, and $\boldsymbol{b}_i$ are the weight matrix and bias for the i-th layer, respectively, and we use the $ReLU$, i.e., $f(x) = max(0, x)$ as the activation function in the hidden layers. From the learning framework in Figure 3, one can see that for each aspect-level similarity matrix of both users and items there is a corresponding MLP learning component described above to learn the aspect-level latent factors.

3.4 Attention based Aspect-level Latent Factors Fusing

After the aspect-level latent factors are learned separately for users and items, next we need to integrate them together to obtain aggregated latent factors. A straightforward way is to concatenate all the aspect-level latent factors to form a higher-dimensional vector. Another intuitive way is to average all the latent factors. The issue is that both methods do not distinguish their different importance because not all the aspects contribute to the recommendation equally (we will show that in the experiment part). Therefore, we choose the attention mechanism to fuse these aspect-level latent factors. Attention mechanism has been shown effective in various machine learning tasks such as image captioning and machine translation [You *et al.*, 2016; Bahdanau *et al.*, 2014]. The advantage of attention mechanism is that it can learn to assign attentive weights (normalized by sum to 1) for all the aspect-level latent factors: higher (lower) weights indicate that the corresponding features are informative (less informative) for recommendation. Specifically, given the user's brand-aspect latent factor $\boldsymbol{u}_i^B$, we use a two-layer network to compute the attention scores $\boldsymbol{s}_i^B$ by the following Equation 3.

$$\boldsymbol{s}_i^B = \boldsymbol{W}_2^T f\left(\boldsymbol{W}_1^T * \boldsymbol{u}_i^B + \boldsymbol{b_1}\right) + \boldsymbol{b_2} \tag{3}$$

where $\boldsymbol{W}$ is the weight matrix and $\boldsymbol{b}$ is the bias.

The final attention weights for the aspect-level latent factors are obtained by normalizing the above attentive scores with the Softmax function given in Equation 4, which can be interpreted as the contributions of different aspects a to the aggregated latent factor of user U_i.

$$\boldsymbol{w}_i^a = \frac{exp(\boldsymbol{s}_i^a)}{\sum_{n\in\mathbb{A}} exp(\boldsymbol{s}_i^n)} \tag{4}$$

Here, $\mathbb{A}$ is the set of all the aspects.

After obtaining all the attention weights $\boldsymbol{w}_i^a$ of all the aspect-level latent factors for user U_i, the aggregated latent factor $\boldsymbol{u}_i$ can be calculated by the Equation 5.

$$\boldsymbol{u}_i = \sum_{a\in\mathbb{A}} \boldsymbol{w}_i^a \cdot \boldsymbol{u}_i^a \tag{5}$$

3.5 Model Optimization

We model the top-N recommendation as a classification problem which predicts the probability of interaction between users and items in the future. In order to ensure that the output value is a probability, we need to constrain the output $\hat{y}_{ij}$ in the range of [0,1], where we use a Logistic function as the activation function for the output layer. The probability of the interaction between the user U_i and item I_j is calculated according to Equation 6.

$$\hat{y}_{ij} = sigmod(\boldsymbol{u}_i * \boldsymbol{v}_j) = \frac{1}{1+e^{-\boldsymbol{u}_i * \boldsymbol{v}_j}} \tag{6}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ are the aggregated latent factors of user U_i and item I_j respectively.

Over all the training set, according to the above settings, the likelihood function is

$$p(\mathcal{Y}, \mathcal{Y}^-|\Theta) = \prod_{i,j\in\mathcal{Y}} \hat{y}_{ij} \prod_{i,k\in\mathcal{Y}^-} (1-\hat{y}_{ik}) \tag{7}$$

where the $\mathcal{Y}$ and the $\mathcal{Y}^-$ are the positive and negative instances set, respectively. The Θ is the parameters set.

Then we take the negative logarithm of the likelihood function Equation 7 to get the point-wise loss function in Equation 8.

$$Loss = \sum_{i,j\in\mathcal{Y}\cup\mathcal{Y}^-} (y_{ij} log\hat{y}_{ij} + (1-y_{ij}) log(1-\hat{y}_{ij})) \tag{8}$$

Dataset	#users	#items	#ratings	#density
ML100K	943	1682	100,000	6.304%
ML1M	6040	3706	1,000,209	4.468%
Amazon	3532	3105	57,104	0.521%

Table 2: The statistics of the datasets.

where y_{ij} is the ground truth of the instance and $\hat{y}_{ij}$ is predicted score. This is the objective function to minimize in our model, and we can optimize it by stochastic gradient descent or its variants.

4 Experiments

4.1 Experimental Settings

Datasets

We evaluate the proposed model over the publicly available MovieLens dataset [Harper and Konstan, 2016] and Amazon dataset [He and McAuley, 2016; McAuley *et al.*, 2015]. The network schema is shown in Figure 2, and the statistics of the datasets are summarized in Table 2.

- MovieLens-100K (ML100k)/MovieLens-1M (ML1M) [1]: MovieLens datasets have been widely used for movie recommendation. We used the version ML100K and ML1M. For each movie, we crawl the director, actor of the movie from IMDb.
- Amazon[2]: This dataset contains users' rating data in Amazon. In our experiment, we select the items of Electronics categories for evaluation.

Evaluation Metric

We adopt the leave-one-out method [He *et al.*, 2017; Xue *et al.*, 2017] for evaluation. The latest rated item of each user is held out for testing, and the remaining data for training. Following previous works [He *et al.*, 2017; Xue *et al.*, 2017], we randomly select 99 items that are not rated by the user as negative samples and rank the 100 sampled items for the user. For a fair comparison with the baseline methods, we use the same negative sample set for each (*user, item*) pair in the test set for all the methods. We evaluate the model performance through the Hit Ratio (HR) and the Normalized Discounted Cumulative Gain (NDCG) defined in Equation 9.

$$HR = \frac{\#hits}{\#users}, NDCG = \frac{1}{\#users} \sum_{i=1}^{\#users} \frac{1}{log_2(p_i+1)} \tag{9}$$

where $\#hits$ is the number of users whose test item appears in the recommended list and p_i is the position of the test item in the list for the i-th hit. In our experiments, we truncated the ranked list at $K \in [5, 10, 15, 20]$ for both metrics.

Baselines

Besides two basic methods (i.e., ItemPop and ItemKNN), the baselines include two MF methods (MF and eALS), one pairwise ranking method (BPR), and two neural network based

[1] https://grouplens.org/datasets/movielens/

[2] http://jmcauley.ucsd.edu/data/amazon/

methods(DMF and NeuMF). In addition, we also adopt a recent HIN based method (FMG) as baseline, since most HIN based methods are designed for rating prediction.

- ItemPop. Items are simply ranked by their interaction popularity.
- ItemKNN. It is a standard item-based collaborative filtering method.
- MF [Koren *et al.*, 2009]. Matrix factorization is a representative latent factor model.
- eALS [He *et al.*, 2016]. It is a state-of-the-art MF method for recommendation with the square loss.
- BPR [Rendle *et al.*, 2009]. The Bayesian Personalized Ranking approach optimizes the MF model with a pairwise ranking loss.
- DMF [Xue *et al.*, 2017]. DMF uses the interaction matrix as the input and maps users and items into a common low-dimensional space using a deep neural network.
- NeuMF [He *et al.*, 2017]. It combines the linearity of MF and non-linearity of DNNs for modelling user–item latent structures.
- FMG [Zhao *et al.*, 2017]. It proposes "MF+FM" framework for the HIN-based rating prediction. We modify its optimization object as point-wise ranking loss for the top-N recommendation.

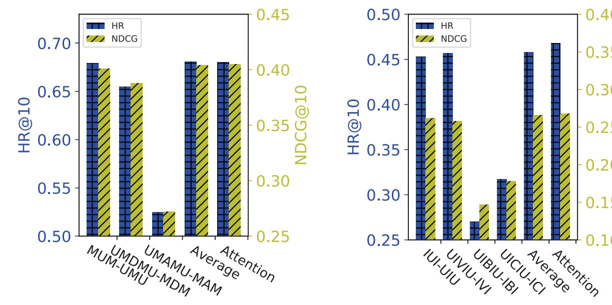
Implementation

We implement the proposed NeuACF based on Tensorflow [Abadi *et al.*, 2016]. We use the same hyper-parameters for all the datasets. For the neural network, we use a two-layer MLP with each hidden layer having 600 hidden units. The dimension of latent factors is 64. We randomly initialized the model parameters with a xavier initializer [Glorot and Bengio, 2010], and used the Adam [Kingma and Ba, 2014] as the optimizer. We set the batch size to 1024 and set the learning rate to 0.0005. When training our model, 10 negative instances are sampled for each positive instance. Table 1 illustrates the extracted aspects and corresponding meta-paths. The optimal parameters for baselines are set according to literatures. All the experiments are conducted on a machine with two GPUs (NVIDIA GTX-1080 *2) and two CPUs (Intel Xeon E5-2690 * 2).

4.2 Experiment Results

Performance Analysis

Table 3 shows the experiment results of different methods. One can observe that, NeuACF almost achieves all the best performance over all the datasets and criteria. As the newest model with neural network, NeuMF also performs well on most conditions, while NeuACF consistently outperforms NeuMF in almost all the cases with only one exception. We think the reasons lie in that multiple aspects of latent factors learned by NeuACF provide more overall features of users and items. Although FMG also utilizes the same features with NeuACF, the better performance of NeuACF implies that the deep neural network in NeuACF may have the better ability to learn latent factors of users and items than the "shadow" model in FMG.



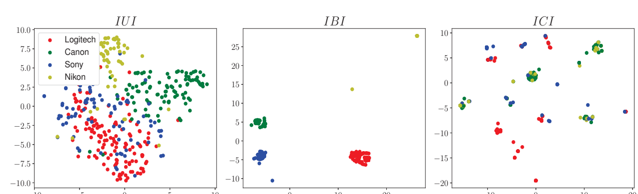
(a) ML100K (b) Amazon

Figure 4: The impact of different aspect-level latent factors.

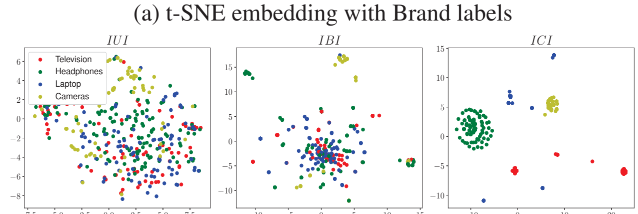
Impact of Different Aspect-level Latent Factors

To analyze the impact of different aspect-level latent factors on the algorithm performance, we run NeuACF with individual aspect-level latent factor through setting meta-path. In Figure 4, for example, $UIBIU$-IBI means that we only learn the brand-aspect latent factor for users and items. In addition, we also run NeuACF with the "Average" and "Attention" fusion mechanism, where "Average" means averaging all the aspect-level latent factors and "Attention" means fusing latent factors with the proposed attention mechanism. From the results shown in Figure 4, one can observe that the purchase-history aspect factors (e.g., UIU-IUI and $UMU-MUM$) usually get the best performance in all the individual aspects because this aspect usually contains the most important information which indicates the purchase history of users and items. One can also see that "Average" and "Attention" always perform better than individual meta-path, demonstrating fusing all the aspect-level latent factors can improve the performance. In addition, the better performance of "Attention" than "Average" also shows the benefit of the attention mechanism in NeuACF.

Visualization of Different Aspect-level Latent Factors



(a) t-SNE embedding with Brand labels



(b) t-SNE embedding with Category labels

Figure 5: t-SNE embedding with different labels of the learned latent factors of items for Amazon.

Datasets	Metrics	ItemPop	ItemKNN	MF	eALS	BPR	DMF	NeuMF	FMG	NeuACF
ML100K	HR@5	0.2831	0.4072	0.4634	0.4698	0.4984	0.3483	0.4942	0.4602	**0.5097**
	NDCG@5	0.1892	0.2667	0.3021	0.3201	0.3315	0.2287	0.3357	0.3014	**0.3505**
	HR@10	0.3998	0.5891	0.6437	0.6638	**0.6914**	0.4994	0.6766	0.6373	0.6846
	NDCG@10	0.2264	0.3283	0.3605	0.3819	0.3933	0.2769	0.3945	0.3588	**0.4068**
	HR@15	0.5366	0.7094	0.7338	0.7529	0.7741	0.5873	0.7635	0.7338	**0.7813**
	NDCG@15	0.2624	0.3576	0.3843	0.4056	0.4149	0.3002	0.4175	0.3844	**0.4318**
	HR@20	0.6225	0.7656	0.8144	0.8155	0.8388	0.6519	0.8324	0.8006	**0.8464**
	NDCG@20	0.2826	0.3708	0.4034	0.4204	0.4302	0.3151	0.4338	0.4002	**0.4469**
ML1M	HR@5	0.3088	0.4437	0.5111	0.5353	0.5414	0.4892	0.5485	0.4732	**0.5630**
	NDCG@5	0.2033	0.3012	0.3463	0.3670	0.3756	0.3314	0.3865	0.3183	**0.3944**
	HR@10	0.4553	0.6171	0.6896	0.7055	0.7161	0.6652	0.7177	0.6528	**0.7202**
	NDCG@10	0.2505	0.3572	0.4040	0.4220	0.4321	0.3877	0.4415	0.3767	**0.4453**
	HR@15	0.5568	0.7118	0.7783	0.7914	0.7988	0.7649	0.7982	0.7536	**0.8018**
	NDCG@15	0.2773	0.3822	0.4275	0.4448	0.4541	0.4143	0.4628	0.4034	**0.4667**
	HR@20	0.6409	0.7773	0.8425	0.8409	0.8545	0.8305	**0.8586**	0.8169	0.8540
	NDCG@20	0.2971	0.3977	0.4427	0.4565	0.4673	0.4296	0.4771	0.4184	**0.4789**
Amazon	HR@5	0.2412	0.1897	0.3027	0.3063	**0.3296**	0.2693	0.3117	0.3216	0.3268
	NDCG@5	0.1642	0.1279	0.2068	0.2049	**0.2254**	0.1848	0.2141	0.2168	0.2232
	HR@10	0.3576	0.3126	0.4278	0.4287	0.4657	0.3715	0.4309	0.4539	**0.4686**
	NDCG@10	0.2016	0.1672	0.2471	0.2441	**0.2693**	0.2179	0.2524	0.2595	0.2683
	HR@15	0.4408	0.3901	0.5054	0.5065	0.5467	0.4328	0.5258	0.5430	**0.5591**
	NDCG@15	0.2236	0.1877	0.2676	0.2647	0.2908	0.2332	0.2774	0.2831	**0.2924**
	HR@20	0.4997	0.4431	0.5680	0.5702	0.6141	0.4850	0.5897	0.6076	**0.6257**
	NDCG@20	0.2375	0.2002	0.2824	0.2797	0.3067	0.2458	0.2925	0.2983	**0.3080**

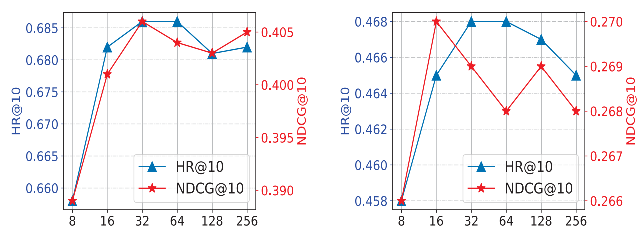
Table 3: HR@K and NDCG@K comparisons of different methods.

In our model, we aim to learn the aspect-level latent factors from different meta-paths. For example, we expect that the brand-aspect latent factor $\boldsymbol{v}_j^B$ for item I_j can be learned from the meta-path IBI, and the category-aspect latent factor $\boldsymbol{v}_j^C$ from the meta-path ICI. To intuitively show whether NeuACF performs well on this task, we visualize the learned aspect-level latent factors on the Amazon dataset. We apply t-SNE [Maaten and Hinton, 2008] to embed the high-dimensional aspect-level latent factors into a 2-dimensional space, and then visualize each item as a point.

Figure 5a shows the embedding result for four famous electronics Brand: *Logitech*, *Canon*, *Sony*, and *Nikon*. One can observe that the brand-aspect latent factors can clearly separate the four brands, while the history-aspect and category-aspect latent factors are mixed with each other. It demonstrates the meta-path IBI can learn a good brand-aspect latent factors. Similarly, in Figure 5b, only the category-aspect latent factors learned from the meta-path ICI clearly separate the items of different categories including *Television*, *Headphones*, *Laptop* and *Cameras*. The results demonstrate that the aspect-level latent factors of items learned by NeuACF can indeed capture the aspect characteristics of items.

Effect of the Latent Factor Dimension

In the latent factor models, the dimension of the latent factors may have a vital impact on the performance of recommendation. Thus we study the effect of dimension of the latent factor learned from the last MLP layer in our proposed model. We conduct the experiments on a 2-layer model, and set the dimensions of the latent factors increasing from 8 to 256. The results on the ML100k and Amazon datasets are shown in Figure 6. One can see that on both cases the performance first increases with the increase of the dimension, and the best performance is achieved at around 16-32. Then the performance drops if the dimension further increases.



(a) ML100k (b) Amazon

Figure 6: Performance with different dimension of latent factors.

5 Conclusion and Future Work

In this work, we explore aspect-level information for collaborative filtering and propose a Neural network based Aspect-level Collaborative Filtering model (NeuACF). Based on different-aspect features extracted from heterogeneous network with meta-paths, the NeuACF learns aspect-level latent factors with a well-designed deep neural network and then fuses them with an attention mechanism for the top-N recommendation. Extensive evaluations demonstrate the superior performance of NeuACF.

As future work, we would like to utilize better attention mechanisms to fuse aspect-level latent factors. In addition, we can explore the strategy of automatic selection of meta-paths in different datasets.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (No. 61772082, 61602237, 61672313, 61320106006), the National Key Research and Development Program of China (2017YFB0803304), and the Beijing Municipal Natural Science Foundation (4182043). This work is also supported in part by NSF through grants IIS-1526499, IIS-1763325, and CNS-1626432.

References

[Abadi *et al.*, 2016] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[Harper and Konstan, 2016] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

[He and McAuley, 2016] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517, 2016.

[He *et al.*, 2016] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*, pages 549–558. ACM, 2016.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.

[Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. IEEE, 2008.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, pages 426–434. ACM, 2008.

[Ma *et al.*, 2008] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940. ACM, 2008.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[McAuley *et al.*, 2015] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52. ACM, 2015.

[Mnih and Salakhutdinov, 2008] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2008.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[Shi *et al.*, 2015] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, and Bin Wu. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *CIKM*, pages 453–462. ACM, 2015.

[Shi *et al.*, 2016] Chuan Shi, Jian Liu, Fuzhen Zhuang, S Yu Philip, and Bin Wu. Integrating heterogeneous information via flexible regularization framework for recommendation. *Knowledge and Information Systems*, 49(3):835–859, 2016.

[Shi *et al.*, 2017a] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S Yu. Heterogeneous information network embedding for recommendation. *arXiv preprint arXiv:1711.10730*, 2017.

[Shi *et al.*, 2017b] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. A survey of heterogeneous information network analysis. *TKDE*, 29(1):17–37, 2017.

[Sun *et al.*, 2011] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB*, 4(11):992–1003, 2011.

[Wang *et al.*, 2015] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *SIGKDD*, pages 1235–1244. ACM, 2015.

[Xue *et al.*, 2017] Hong-Jian Xue, Xin-Yu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. *IJCAI*, 2017.

[You *et al.*, 2016] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, pages 4651–4659, 2016.

[Yu *et al.*, 2014] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*, pages 283–292. ACM, 2014.

[Zhao *et al.*, 2017] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. In *SIGKDD*, pages 635–644. ACM, 2017.