

# RETHINKING GENERALIZED MATRIX FACTORIZATION FOR RECOMMENDATION: THE IMPORTANCE OF MULTI-HOT ENCODING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Learning good representations of users and items is crucially important to recommendation with implicit feedback. Matrix factorization is the basic idea to derive the representations of users and items by decomposing the given interaction matrix. However, existing matrix factorization based approaches share the limitation in that the interaction between user embedding and item embedding is only weakly enforced by fitting the given individual rating value, which may lose potentially useful information. In this paper, we propose a novel *Augmented Generalized Matrix Factorization* (AGMF) approach that is able to incorporate the historical interaction information of users and items for learning effective representations of users and items. Despite the simplicity of our proposed approach, extensive experiments on four public implicit feedback datasets demonstrate that our approach outperforms state-of-the-art counterparts. Furthermore, the ablation study demonstrates that by using multi-hot encoding to enrich user embedding and item embedding for *Generalized Matrix Factorization*, better performance, faster convergence, and lower training loss can be achieved.

## 1 INTRODUCTION

In the era of big data, we are seriously confronted with the problem of information overload. Recommender systems play an important role in dealing with such issue, thereby having been widely deployed by social media, E-commerce platforms, and so on. Among the techniques used in recommender systems, collaborative filtering (Sarwar et al., 2001; Hu et al., 2008; Su & Khoshgoftaar, 2009; He et al., 2017; Wang et al., 2019) is the dominant one that leverages user-item interaction data to predict user preference. Among various collaborative filtering methods, *Matrix Factorization* (MF) is the most popular approach that has inspired a large number of variations (Koren, 2008; Rendle et al., 2009; He et al., 2017; Xue et al., 2017). MF aims to project users and items into a shared latent space, and each user or item could be represented by a vector composed by latent features. In this way, the user-item interaction score could be recovered by the inner product of the two latent vectors. Most of the existing extensions of MF normally focus on the modeling perspective (Wang et al., 2015; 2018) and the learning perspective (Xue et al., 2017; He et al., 2017). For example, BPR-MF (Rendle et al., 2009) learns user embedding and item embedding from implicit feedback by optimizing a Bayesian pairwise ranking objective function. NeuMF (He et al., 2017) learns compact embeddings by fusing the outputs from different models. DeepMF (Xue et al., 2017) employs deep neural networks to learn nonlinear interactions of users and items.

Although these approaches have achieved great success, they still cannot resolve the inherent limitation of MF. Specifically, apart from the interaction by inner product, there are no explicit relationships between user embedding and item embedding. In other words, the connection between user embedding and item embedding is only weakly enforced by fitting the given individual rating value. However, in real-world scenarios, user embedding and item embedding may be interpreted as some high-level descriptions or properties of user and item, which are supposed to have some explicit connections. For example, a user likes some item, probably because the user and the item share some similar high-level descriptions or properties. Which means, the latent features of a user could be potentially enriched by taking into account the latent features of the user’s interacted items, since these interacted items could expose the latent features of the user to some degree. Similarly,

the latent features of an item may also be enriched by the latent features of the item’s interacted users. However, most of the existing approaches regrettably ignore such useful information. An exception is the SVD++ model (Koren, 2008), which provides each user embedding with additional latent features of items that the user has interacted with. Despite the effectiveness of the SVD++ model, it suffers from two major problems. First, it only enriches user embedding, and ignores the fact that item embedding could also be enriched by the latent features of users that the item has interacted with. Second, the latent features of the interacted items are averagely integrated without discrimination, while each user normally has different preferences on different items. Note that there are also other approaches that regularize or enrich user embedding and item embedding by exploiting supplementary information, such as social relations (Ma et al., 2011; Guo et al., 2017) and text reviews (Chen et al., 2018). However, in this paper, we do not assume there is any supplementary information, and only focus on the data with implicit feedback.

Motivated by the above observations, this paper proposes a novel *Augmented Generalized Matrix Factorization* (AGMF) approach for learning from implicit feedback. Different from existing approaches, AGMF aims to enrich both user embedding and item embedding by applying multi-hot encoding with the attention mechanism on historical items and users. In this way, user embedding and item embedding are explicitly related to each other, which could further improve the performance of the *Generalized Matrix Factorization* (GMF) model. Extensive experimental results clearly demonstrate our contributions.

## 2 PRELIMINARIES

In this section, we first present the problem statement and then briefly introduce the basic MF model, the GMF model, and the SVD++ model.

### 2.1 PROBLEM STATEMENT

Let  $\mathcal{U} = \{1, 2, \dots, U\}$  be the set of  $U$  users, and  $\mathcal{I} = \{1, 2, \dots, I\}$  be the set of  $I$  items. We define the given user-item interaction matrix  $\mathbf{Y} = [y_{ui}]_{U \times I}$  from implicit feedback data as:  $y_{ui} = 1$  if the interaction  $(u, i)$  is observed, otherwise  $y_{ui} = 0$ . It is worth noting that  $y_{ui} = 1$  indicates that there is an observed interaction between user  $u$  and item  $i$ , while it does not necessarily mean that  $u$  likes  $i$ . In addition,  $y_{ui} = 0$  does not necessarily mean that  $u$  does not like  $i$ , and it is possible that  $u$  is not aware of  $i$ . Such setting could inevitably bring additional challenges for learning from implicit feedback, since it may provide misleading information about user’s preference. The goal of recommendation with implicit feedback is to predict the values of the unobserved entries in  $\mathbf{Y}$ , which can be further used to rank the items.

### 2.2 MATRIX FACTORIZATION

MF is a basic latent factor model, which aims to characterize each user and item by a real-valued vector of latent features (Koren et al., 2009). Let  $\mathbf{p}_u$  and  $\mathbf{q}_i$  be the latent vectors for user  $u$  and item  $i$ , respectively. MF tries to give an estimation  $\hat{y}_{ui}$  of  $y_{ui}$  by the inner product of  $\mathbf{p}_u$  and  $\mathbf{q}_i$ :

$$\hat{y}_{ui} = \mathbf{p}_u^\top \mathbf{q}_i = \sum_{k=1}^K p_{uk} q_{ik} \quad (1)$$

where  $K$  is the dimension of the latent vectors. As can be seen, the latent features could be considered as linearly combined in MF. Hence MF can be regarded as a linear model with respect to latent features. This linear property of MF restricts its performance to some degree. As a result, there are an increasing number of approaches (Xue et al., 2017; He et al., 2017; Zhang et al., 2018) proposed to alleviate this problem, by learning a nonlinear interaction function using deep neural networks.

### 2.3 GENERALIZED MATRIX FACTORIZATION

*Generalized Matrix Factorization* (GMF) (He et al., 2017) is a simple nonlinear generalization of MF, which makes a prediction  $\hat{y}_{ui}$  of  $y_{ui}$  as follows:

$$\hat{y}_{ui} = \sigma(\mathbf{h}^\top (\mathbf{p}_u \odot \mathbf{q}_i)) \quad (2)$$

where  $\odot$  denotes the element-wise product of vectors,  $\mathbf{h}$  is a weight vector, and  $\sigma(\cdot)$  is an activation function. To show that MF is a special case of GMF, we can simply set  $\mathbf{h} = \mathbf{1}$  where  $\mathbf{1}$  is the vector with all elements equal to 1. In this way, apart from the activation function, the MF model is exactly recovered by GMF, since  $\mathbf{p}_u^\top \mathbf{q}_i = \mathbf{1}^\top (\mathbf{p}_u \odot \mathbf{q}_i)$ .

## 2.4 SVD++

SVD++ extends MF by leveraging both explicit ratings and implicit feedback to make prediction:

$$\hat{y}_{ui} = \mathbf{q}_i^\top (\mathbf{p}_u + |\mathcal{N}(u)|^{-\frac{1}{2}} \sum_{j \in \mathcal{N}(u)} \mathbf{c}_j) \quad (3)$$

where  $\mathcal{N}(u)$  denotes the set that stores all the items for which  $u$  has provided implicit feedback, and  $\mathbf{c}_j$  is a latent vector of item  $j$  for implicit feedback, while  $\mathbf{p}_u$  and  $\mathbf{q}_i$  are free user-specific and item-specific latent vectors specially learned for explicit ratings. The only difference between SVD++ and MF lies in that  $\mathbf{p}_u$  is enriched by  $|\mathcal{N}(u)|^{-\frac{1}{2}} \sum_{j \in \mathcal{N}(u)} \mathbf{c}_j$ .

## 3 THE PROPOSED APPROACH

Figure 1 illustrates the framework of our AGMF model. It is worthy noting that for the input layer, unlike most of the existing approaches (Koren et al., 2009; He et al., 2018) that only employ one-hot encoding on the target user’s ID (denoted by  $u$ ) and the target item’s ID (denoted by  $i$ ), we additionally apply multi-hot encoding on user  $u$ ’s interacted items, and item  $i$ ’s interacted users. In this way, potentially useful information is incorporated, which could enrich the embedding of  $u$  and  $i$ . Note that this part is the core design of our proposed AGMF model. By enriching the one-hot encoding with multi-hot encoding, the historical interactions between users and items are exploited, therefore our AGMF model with multi-hot encoding achieves superior performance to the GMF model with only one-hot encoding. We argue that this design is simple but more advantageous in threefold. First, it is an augmented version of the GMF model that only takes into account the one-hot embedding of the target user and the target item. Second, it encodes more information than the GMF model, by considering numerous historical interactions of users and items. Such natural information is valuable and should not be ignored for recommendation with only implicit feedback. Third, different from most matrix factorization based recommendation models that only relates user embedding and item embedding by fitting the given individual value, our AGMF model joint learns user embedding and item embedding in a more explicit way.

In order to avoid the conflict that user  $u$  may overly concentrate on item  $i$  if the target item is  $i$ , we will exclude  $i$  from  $\mathcal{N}(u)$  (denoted by  $\mathcal{N}(u) \setminus \{i\}$ ) when predicting  $\hat{y}_{ui}$ . Similarly, we will also exclude  $u$  from  $\mathcal{N}(i)$  (denoted by  $\mathcal{N}(i) \setminus \{u\}$ ) when predicting  $\hat{y}_{ui}$ . In what follows, we detail elaborate the design of our AGMF model layer by layer.

### 3.1 INPUT AND EMBEDDING LAYER

Given the target user  $u$ , the target item  $i$ , user  $u$ ’s interacted items  $\mathcal{N}(u)$ , and item  $i$ ’s interacted users  $\mathcal{N}(i)$ , we not only apply one-hot encoding on  $u$  and  $i$ , but also apply multi-hot encoding on  $\mathcal{N}(u)$  and  $\mathcal{N}(i)$ . In this way,  $u$  and  $i$  are projected to latent feature vectors  $\mathbf{p}_u \in \mathbb{R}^K$  and  $\mathbf{q}_i \in \mathbb{R}^K$ . Similarly, for each historical item  $j \in \mathcal{N}(u) \setminus \{i\}$  and each historical user  $k \in \mathcal{N}(i) \setminus \{u\}$ , we can obtain  $\{\mathbf{q}_j \in \mathbb{R}^K | j \in \mathcal{N}(u) \setminus \{i\}\}$  and  $\{\mathbf{p}_k \in \mathbb{R}^K | k \in \mathcal{N}(i) \setminus \{u\}\}$ .

### 3.2 PAIRWISE INTERACTION LAYER

Following the interaction way used in GMF, we also apply the widely-used element-wise product (He et al., 2017; Cheng et al., 2018; Xue et al., 2018) to model the interactions of  $u$  and  $\mathcal{N}(u)$  as well as  $i$  and  $\mathcal{N}(i)$ . Generally, interaction ways such as  $\mathbf{p}_u + \mathbf{q}_j$ ,  $\mathbf{p}_u - \mathbf{q}_j$ , or any other function that integrates two vectors into a single vector, can also be used. Here, we choose element-wise product because it generalizes inner product to vector space, which could retrain the signal of inner product to a great extent.

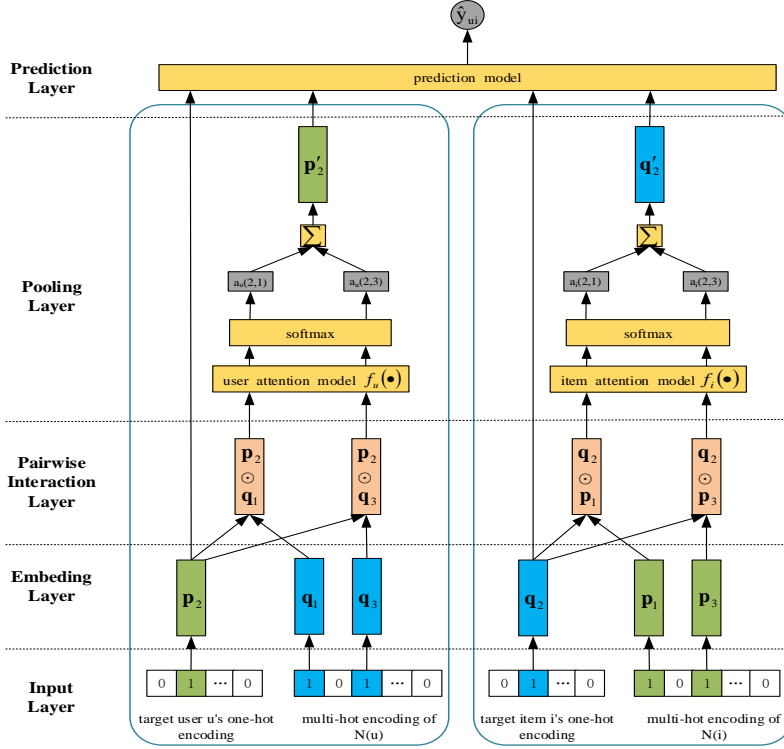


Figure 1: The framework of our AGMF model.

### 3.3 POOLING LAYER

Since there are multiple historical items of user  $u$  and multiple historical users of item  $i$ , how to extract useful information from these generated latent vectors is crucially important. In reality, the historical items of the target user  $u$  normally make different contributions to  $u$  on decision of the target item  $i$ . The same situation holds for the target item  $i$  while interacting with the target user  $u$ . Therefore, we perform a weighted sum on the latent vectors obtained from the pairwise interaction layer, i.e.,

$$\mathbf{p}_{u'} = \sum_{j \in N_u \setminus \{i\}} a_u(u, j) \mathbf{q}_j, \quad \mathbf{q}_{i'} = \sum_{k \in N_i \setminus \{u\}} a_i(i, k) \mathbf{p}_k \quad (4)$$

where  $a_u(u, j)$  denotes the attention weight that the target user  $u$  on its interacted item  $j$ , and  $a_i(i, k)$  is the attention weight that the target item  $i$  on its interacted user  $k$ . Note that  $a_u(u, i) = a_i(i, u)$  does not necessarily hold, as you are my best friend, while I may not be your best friend.  $\mathbf{p}_{u'}$  and  $\mathbf{q}_{i'}$  are the supplementary latent vectors generated by the pooling layer, which will be used to enrich  $\mathbf{p}_u$  and  $\mathbf{q}_i$ . In this paper, we define  $a_u(u, j)$  ( $\forall j \in N_u \setminus \{i\}$ ) and  $a_i(i, k)$  ( $\forall k \in N_i \setminus \{u\}$ ) as the softmax normalization of the interaction scores between users and items:

$$a_u(u, j) = \frac{\exp(f_u(\mathbf{p}_u \odot \mathbf{q}_j))}{\sum_j \exp(f_u(\mathbf{p}_u \odot \mathbf{q}_j))}, \quad a_i(i, k) = \frac{\exp(f_i(\mathbf{q}_i \odot \mathbf{p}_k))}{\sum_k \exp(f_i(\mathbf{q}_i \odot \mathbf{p}_k))} \quad (5)$$

where  $f_u(\cdot)$  ( $f_i(\cdot)$ ) is the user (item) attention model that takes the user-item interaction vector as an input, and outputs the corresponding interaction score. In this paper, we define  $f_u(\cdot)$  and  $f_i(\cdot)$  as:

$$f_u(\mathbf{p}_u \odot \mathbf{q}_j) = \sigma(\mathbf{h}_u^\top (\mathbf{p}_u \odot \mathbf{q}_j)), \quad f_i(\mathbf{q}_i \odot \mathbf{p}_k) = \sigma(\mathbf{h}_i^\top (\mathbf{q}_i \odot \mathbf{p}_k)) \quad (6)$$

where  $\mathbf{h}_u$  and  $\mathbf{h}_i$  are the weight vectors of the user attention model and the item attention model, respectively. Note that unlike existing approaches that normally take multi-layer neural networks as the attention model, we only use a single-layer perceptron. In this way, our proposed attention model is exactly a standard GMF model. Our experimental results show that such simple GMF model can achieve satisfactory performance, with keeping simple and efficient. While deeper structures could potentially achieve better performance, we leave the exploration of deeper structures for attention modeling in future work.

### 3.4 PREDICTION LAYER

With the supplementary latent vectors  $\mathbf{p}_{u'}$  and  $\mathbf{q}_{i'}$  for  $\mathbf{p}_u$  and  $\mathbf{q}_i$ , inspired by SVD++, we represent the latent vector of user  $u$  by  $\mathbf{p}_u + \mathbf{p}_{u'}$ , and represent the latent vector of item  $i$  by  $\mathbf{q}_i + \mathbf{q}_{i'}$ . Then we reuse the GMF model as the prediction model, and the predicted interaction score  $\hat{y}_{ui}$  is given by:

$$\hat{y}_{ui} = \sigma(\mathbf{h}^\top((\mathbf{p}_u + \mathbf{p}_{u'}) \odot (\mathbf{q}_i + \mathbf{q}_{i'}))) \quad (7)$$

where  $\mathbf{h}$  is the weight vector of the prediction model. We empirically use the sigmoid function  $\sigma(\mathbf{x}) = 1/(1 + \exp(-\mathbf{x}))$  as the activation function throughout this paper.

### 3.5 LOSS FUNCTION

Since this paper focuses on learning from implicit feedback data, the output  $\hat{y}_{ui}$  of our AGMF model is constrained in the range of  $[0, 1]$ , which could provide a probability explanation. In such setting, the commonly used *Binary Cross Entropy* (BCE) loss could be employed:

$$\mathcal{L} = - \sum_{(u,i) \in \mathcal{O}^+} \log \hat{y}_{ui} - \sum_{(u,j) \in \mathcal{O}^-} \log(1 - \hat{y}_{uj}) \quad (8)$$

where  $\mathcal{O}^+$  denotes the observed interactions and  $\mathcal{O}^-$  denotes the set of negative instances that could be sampled from unobserved interactions. In this paper, for each training epoch, we randomly sample four negative instances per positive instance.

## 4 EXPERIMENTS

In this section, we conduct extensive experiments to demonstrate that our AGMF model outperforms state-of-the-art counterparts. In addition, we provide ablation study to clearly demonstrate the importance of multi-hot encoding for generalized matrix factorization.

### 4.1 EXPERIMENTAL SETTINGS

#### 4.1.1 DATASETS

We conduct experiments on four publicly available datasets: MovieLens 1M (ML-1M)<sup>1</sup>, Yelp<sup>2</sup>, Amazon Movies and Tv (Movies&Tv)<sup>3</sup>, and Amazon CDs and Vinyl (CDs&Vinyl). For ML-1M, we directly use the original dataset downloaded from the MovieLens website. Since the high sparsity of the original dataset makes it much difficult to evaluate recommendation approaches, we follow the common practice (Rendle et al., 2009; He et al., 2016) to process the other three datasets. For the Yelp dataset, we filter out the users and items with less than 10 interactions (He et al., 2016). For Movies&Tv and CDs&Vinyl, we filter out the users that have less than 10 interactions. As this paper focuses on the data with implicit feedback, we mask all the data with explicit feedback to have only implicit feedback by marking each entry 0 or 1, which indicates whether the user has interacted the item. The main characteristics of these datasets are provided in Table 1.

Table 1: Characteristics of the used datasets.

Dataset	ML-1M	Yelp	Movies&Tv	CDs&Vinyl
Number of users	6040	25,677	40,928	26,876
Number of Items	3706	25,815	51,509	66,820
Number of interactions	1,000,209	698,506	1,163,413	770,188
Rating density	0.04468	0.00105	0.00055	0.00043

<sup>1</sup><https://grouplens.org/datasets/movielens/>

<sup>2</sup><https://www.yelp.com/dataset/challenge>

<sup>3</sup><https://www.amazon.com/>

#### 4.1.2 COMPARING ALGORITHMS

We compare AGMF with the following state-of-the-art approaches:

- **SVD++** (Koren, 2008) It merges the latent factor model and the neighborhood model by enriching the user latent feature with the interacted items’ latent features.
- **BPR-MF** (Rendle et al., 2009) It trains the basic MF model by optimizing the Bayesian personalized ranking loss.
- **FISM** (Kabbur et al., 2013) It is an item-based approach, which factorizes the similarity matrix into two low-rank matrices.
- **MLP** (He et al., 2017) It learns the interactions between users and items by multi-layer perceptron.
- **GMF** (He et al., 2017) It generalizes the basic MF model to a non-linear setting.
- **NeuMF-p** (He et al., 2017) NeuMF is a combination of MLP and GMF, and its pretraining version is called NeuMF-p. In this paper, we compare with NeuMF-p, as NeuMF-p provides better performance than NeuMF without pretraining (He et al., 2017).
- **ConvNCF** (He et al., 2018) It employs a convolutional neural network to learn high-order interactions based on the interaction map generated by the outer product of user embedding and item embedding.

#### 4.1.3 TRAINING DETAILS

We randomly holdout 1 training interaction for each user as the development set to tune hyperparameters suggested by respective literatures. Unless otherwise specified, for all the algorithms, the learning rate is chosen from  $[5e^{-5}, 1e^{-4}, 5e^{-4}, 1e^{-3}, 5e^{-3}]$ , the embedding size  $K$  is chosen from  $[16, 32, 64, 128]$ , the regularization parameter (that controls the model complexity) is chosen from  $[1e^{-5}, 5e^{-6}, 1e^{-5}, 5e^{-5}]$ , and the batch size is set to 256. For MLP and NeuMF-p that have multiple fully connected layers, we follow the tower structure of neural networks (He et al., 2017), and tune the number of hidden layers from 1 to 3. For ConvNCF<sup>4</sup>, we follow the configuration and architectures proposed in (He et al., 2018). All the models are trained until convergence or the default maximum number of epochs (by respective literature) is reached.

For our proposed AGMF model, no neural network is adopted, hence we do not need to tune the network structure. We initialize the weight vectors by the Xavier initialization (Jia et al., 2014), and initialize the embedding vectors using a uniform distribution from 0 to 1.

For training AGMF, we employ the Adaptive Moment Estimation (Adam) (Kingma & Ba, 2014), which adapts the learning rate for each parameter by performing small updates for frequent parameters and large updates for infrequent parameters. We implement AGMF using PyTorch<sup>5</sup>, and the source code as well as the used datasets are released. We fix the embedding size at 128, since we found that a larger embedding size always performs better. Note that the number of interacted users or items may be very large, to mitigate this issue, we truncate the list of interacted users and items such that the latent representation of each user/item is enriched by the latent vectors of at most 50 latest interacted items/users.

## 4.2 EXPERIMENTAL RESULTS

In this paper, we adopt the widely used *leave-one-out* evaluation method (Rendle et al., 2009; He et al., 2016; Bayer et al., 2017) to compare AGMF with other approaches. Specifically, for each dataset, we holdout the latest interaction of each user as the test positive examples, and randomly select 99 items that the user has not interacted with as the test negative examples. In this way, all the algorithms make ranking predictions for each user based on these 100 user-item interactions.

To evaluate the ranking performance, we adopt two widely used evaluation criteria, including *Hit Ratio* (HR) and *Normalized Discount Cumulative Gain* (NDCG).  $HR@k$  is a recall-based metric

<sup>4</sup><https://github.com/duxy-me/ConvNCF>

<sup>5</sup><https://pytorch.org/>

Table 2: HR@5 and NDCG@5 comparisons of different approaches. The best results are highlighted.

	ML-1M		Yelp		Movie&Tv		CDs&Vinyl	
	HR@5	NDCG@5	HR@5	NDCG@5	HR@5	NDCG@5	HR@5	NDCG@5
BPR-MF	0.496	0.344	0.700	0.526	0.633	0.479	0.671	0.523
SVD++	0.541	0.376	0.703	0.528	0.631	0.480	0.666	0.520
FISM	0.528	0.372	0.691	0.511	0.583	0.452	0.592	0.457
MLP	0.526	0.362	0.671	0.498	0.570	0.425	0.588	0.445
GMF	0.540	0.372	0.676	0.507	0.569	0.427	0.620	0.481
NeuMF	0.548	0.381	0.695	0.521	0.596	0.453	0.629	0.491
ConvNCF	0.549	0.391	0.708	0.532	0.634	0.484	0.673	0.525
AGMF	<b>0.561</b>	<b>0.393</b>	<b>0.716</b>	<b>0.547</b>	<b>0.644</b>	<b>0.497</b>	<b>0.685</b>	<b>0.542</b>

Table 3: HR@10 and NDCG@10 comparisons of different approaches. The best results are highlighted.

	ML-1M		Yelp		Movie&Tv		CDs&Vinyl	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
BPR-MF	0.675	0.401	0.833	0.569	0.765	0.527	0.784	0.560
SVD++	0.716	0.437	0.834	0.570	0.762	0.524	0.780	0.556
FISM	0.699	0.433	0.824	0.567	0.708	0.471	0.729	0.512
MLP	0.703	0.421	0.805	0.537	0.703	0.471	0.712	0.485
GMF	0.711	0.429	0.809	0.552	0.712	0.479	0.729	0.515
NeuMF	0.727	0.443	0.824	0.560	0.721	0.493	0.750	0.529
ConvNCF	0.713	0.445	0.836	0.572	0.762	0.525	0.785	0.562
AGMF	<b>0.731</b>	<b>0.449</b>	<b>0.838</b>	<b>0.585</b>	<b>0.766</b>	<b>0.535</b>	<b>0.793</b>	<b>0.576</b>

that measures whether the testing item is on the top- $k$  list, and  $NDCG@k$  assigns higher scores to the items with higher positions within the top- $k$  list (He et al., 2018).

#### 4.2.1 PERFORMANCE COMPARISON

Table 2 and Table 3 show the top- $k$  performance of all the algorithms when  $k = 5$  and  $k = 10$ , respectively. From the two tables, we can observe that:

- AGMF achieves the best performance (the highest HR and NDCG scores) on the four datasets.
- Although AGMF is a simple extension of GMF, it still outperforms the complex state-of-the-art approaches NeuMF-p and ConvNCF.
- Compared with GMF, AGMF achieves significantly better performance. Such success owes to multi-hot encoding with the attention mechanism, which provides enriched information for user embedding and item embedding.

#### 4.2.2 ABLATION STUDY

As aforementioned, the GMF model makes prediction by  $\hat{y}_{ui} = \sigma(\mathbf{h}^\top(\mathbf{p}_u \odot \mathbf{q}_i))$ , while our AGMF model makes prediction by  $\hat{y}_{ui} = \sigma(\mathbf{h}^\top((\mathbf{p}_u + \mathbf{p}'_u) \odot (\mathbf{q}_i + \mathbf{q}'_i)))$ . Clearly, without the supplementary latent vectors  $\mathbf{p}'_u$  and  $\mathbf{q}'_i$ , AGMF reduces to GMF.

Table 2 and Table 3 have clearly showed that AGMF significantly outperforms GMF. While the used GMF model for performance evaluation is provided by (He et al., 2017), which is implemented by Keras, and uses a different initialization strategy. Hence it may be slightly different from AGMF without  $\mathbf{p}'_u$  and  $\mathbf{q}'_i$ . For fair comparison and pure ablation study, we conduct experiments using the codes of AGMF, to compare the performance of AGMF and AGMF *without* the supplementary latent vectors  $\mathbf{p}'_u$  and  $\mathbf{q}'_i$ . While with a slight abuse of naming, in this ablation study, we still name AGMF *without*  $\mathbf{p}'_u$  and  $\mathbf{q}'_i$  as GMF.

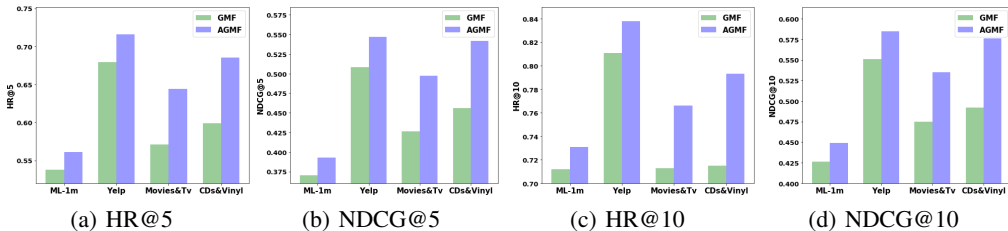


Figure 2: Comparison results of AGMF and GMF on all the used datasets.

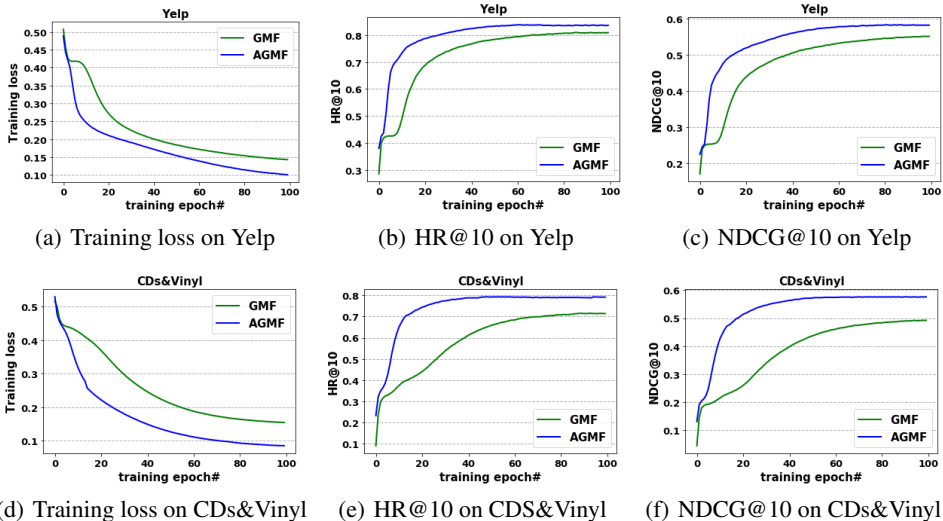


Figure 3: Comparison results of AGMF and GMF in each training epoch.

Figure 2 reports the comparison results of AGMF and GMF on all the datasets. It can be seen that AGMF always achieves better performance than GMF, in terms of both evaluation metrics. Furthermore, we also report the comparison results of AGMF and GMF in each training epoch in Figure 3. We can observe that:

- AGMF consistently outperforms GMF in each training epoch.
- AGMF converges faster than GMF.
- AGMF achieves lower training loss than GMF.

By integrating historical interactions into user embedding and item embedding, the above observations are revealed by this paper for the first time. Therefore, the importance of multi-hot encoding for generalized matrix factorization is clearly demonstrated. Moreover, these observations may bring new inspirations about how to properly integrate the one-hot encoding and multi-hot encoding for effectively improving the recommendation performance.

## 5 CONCLUSION

Learning good representations of users and items is crucially important to recommendation with implicit feedback. In this paper, we propose a novel *Augmented Generalized Matrix Factorization* (AGMF) model for learning from implicit feedback data. Extensive experimental results demonstrate that our proposed approach outperforms state-of-the-art counterparts. Besides, our ablation study clearly demonstrates the importance of multi-hot encoding for *Generalized Matrix Factorization*. As user-item interaction relationships are vitally important for learning effective user embedding and item embedding, hence in future work, we will investigate if there exist better user-item interaction relationships that can be exploited to improve the recommendation performance.



## REFERENCES

- Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. A generic coordinate descent framework for learning from implicit feedback. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 1341–1350, 2017.
- Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the International Conference on World Wide Web*, pp. 1583–1592, 2018.
- Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S Kankanhalli. A<sup>3</sup>ncf: An adaptive aspect attention model for rating prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3748–3754, 2018.
- Guibing Guo, Jie Zhang, Feida Zhu, and Xingwei Wang. Factored similarity models with social trust for top-n item recommendation. *Knowledge-Based Systems*, 122:17–25, 2017.
- Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 549–558, 2016.
- Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, 2017.
- Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. Outer product-based neural collaborative filtering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3669–3675, 2018.
- Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Eighth IEEE International Conference on Data Mining*, pp. 263–272, 2008.
- Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 675–678, 2014.
- Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 659–667, 2013.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426–434, 2008.
- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pp. 287–296, 2011.
- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, 2009.
- Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, 2001.

- Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.
- Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. Exploring implicit hierarchical structures for recommender systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1813–1819, 2015.
- Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the International Conference on World Wide Web*, pp. 1543–1552, 2018.
- Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.
- Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. Deep item-based collaborative filtering for top-n recommendation. *ACM Transactions on Information Systems*, 37(3):17–25, 2018.
- Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3203–3209, 2017.
- Shuai Zhang, Lina Yao, Aixin Sun, Sen Wang, Guodong Long, and Manqing Dong. Neurec: On nonlinear transformation for personalized ranking. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3669–3675, 2018.