Embedding-Augmented Generalized Matrix Factorization for Recommendation with Implicit Feedback

Lei Feng, Hongxin Wei, Qingyu Guo, Zhuoyi Lin, Bo An

***Abstract*—Learning effective 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 the historical interactions to enrich user embedding and item embedding for *Generalized Matrix Factorization*, better performance, faster convergence, and lower training loss can be achieved.**

***Index Terms*—Recommender systems, matrix factorization, multi-hot encoding, representation learning**

I. 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 [1, 2, 3, 4, 5] 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 [6, 7, 3, 8]. 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 [9] and the learning perspective [8, 3]. For example, BPR-MF [7] learns user embedding and item embedding from implicit feedback by optimizing a Bayesian pairwise ranking objective function. NeuMF [3] learns compact embeddings by fusing the outputs from different models. DeepMF [8] employs deep neural networks to learn nonlinear interactions of users and items.

Although the above 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 (latent features) 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 (latent features). Which means, the latent features of a user could be explicitly connected to 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. To properly incorporate such useful information, the SVD++ model [6] proposes to enrich each user embedding by additional latent features of items that the user has interacted with. Despite the effectiveness of the SVD++ model, it suffers from three 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. Last but most important, the model capacity of the SVD++ model is quite limited, which only considers the linear combination of the latent features of uses and items and cannot deal with complex nonlinear case. However, the relationships between users and items are generally complex, and may not be linear, which inherently limits the performance of the SVD++ model.

Motivated by the above observations, this paper makes the following contributions:

- We propose a novel latent factor model named AGMF for recommendation with implicit feedback, which explicitly encodes the interaction information of both user's side and item's side. To differentiate the importance of interaction relations, the attention models are seamlessly incorporated to learn better representations of users and items.

Lei Feng, Hongxin Wei, Qingyu Guo, Zhuoyi Lin, and Bo An are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (E-mail: feng0093@e.ntu.edu.sg, hongxin001@e.ntu.edu.sg, qguo005@e.ntu.edu.sg, zhuoyi001@e.ntu.edu.sg, boan@ntu.edu.sg).

Corresponding author: Hongxin Wei.

- By enriching original embedding with historical interactions, better performance, faster convergence, and lower training loss can be achieved. These observations may motivate us to rethink the relationships or influences between one-hot encoding and multi-hot encoding.
- We conduct extensive experiments on four real-world datasets with implicit feedback, and the experimental results clearly demonstrate that our proposed AGMF model outperforms state-of-the-art counterparts.

II. Preliminaries and Related Works

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

A. Problem Statement

Let $\mathcal{U} = \{1, 2, \cdots, U\}$ be the set of U users, and $\mathcal{I} = \{1, 2, \cdots, I\}$ 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.

B. 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 [10]. 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 $\mathbf{q}_i$:

$$\hat{y}_{ui} = \mathbf{p}_u^\top \mathbf{q}_i = \sum\nolimits_{k=1}^{K} p_{uk} q_{ik} \tag{1}$$

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 [8, 3] proposed to alleviate this problem, by learning a nonlinear interaction function using deep neural networks.

C. Generalized Matrix Factorization

Generalized Matrix Factorization (GMF) [3] is a simple nonlinear generalization of MF, which makes a prediction $\hat{y}_{ui}$ of y_{ui} as follows:

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

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 $\mathbf{p}_u^\top \mathbf{q}_i = \mathbf{1}^\top (\mathbf{p}_u \odot \mathbf{q}_i)$.

D. SVD++

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

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

where $\mathrm{N}(u)$ 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 $|\mathrm{N}(u)|^{-\frac{1}{2}} \sum_{j \in \mathrm{N}_u} \mathbf{c}_j$. It is worth noting that SVD++ can only model linear relationships between users and items. There is another related work [11] that learns non-linear representations of items using historical interactions.

Note that there are also other approaches that regularize or enrich user embedding and item embedding by exploiting supplementary information, such as social relations [12] and text reviews [13]. However, in this paper, we do not assume there is any supplementary information, and only focus on the data with implicit feedback.

III. The Proposed Approach

Fig. 1 illustrates the framework of our AGMF model. It is worth noting that for the input layer, unlike most of the existing approaches [10, 14] 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.

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 $\mathrm{N}(u)$ (denoted by $\mathrm{N}(u) \backslash \{i\}$) when predicting $\hat{y}_{ui}$. Similarly, we will also exclude u from $\mathrm{N}(i)$ (denoted by $\mathrm{N}(i) \backslash \{u\}$) when predicting $\hat{y}_{ui}$. In what follows, we detail elaborate the design of our AGMF model layer by layer.

A. Input and Embedding Layer

Given the target user u, the target item i, user u's interacted items $\mathrm{N}(u)$, and item i's interacted users $\mathrm{N}(i)$, we not only apply one-hot encoding on u and i, but also apply multi-hot encoding on $\mathrm{N}(u)$ and $\mathrm{N}(i)$. 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 $j \in \mathrm{N}(u) \backslash \{i\}$ and each historical user $k \in \mathrm{N}(i) \backslash \{u\}$, we can obtain $\{\mathbf{q}_j \in \mathbb{R}^K | j \in \mathrm{N}(u) \backslash \{i\}\}$ and $\{\mathbf{p}_k \in \mathbb{R}^K | k \in \mathrm{N}(i) \backslash \{u\}\}$.

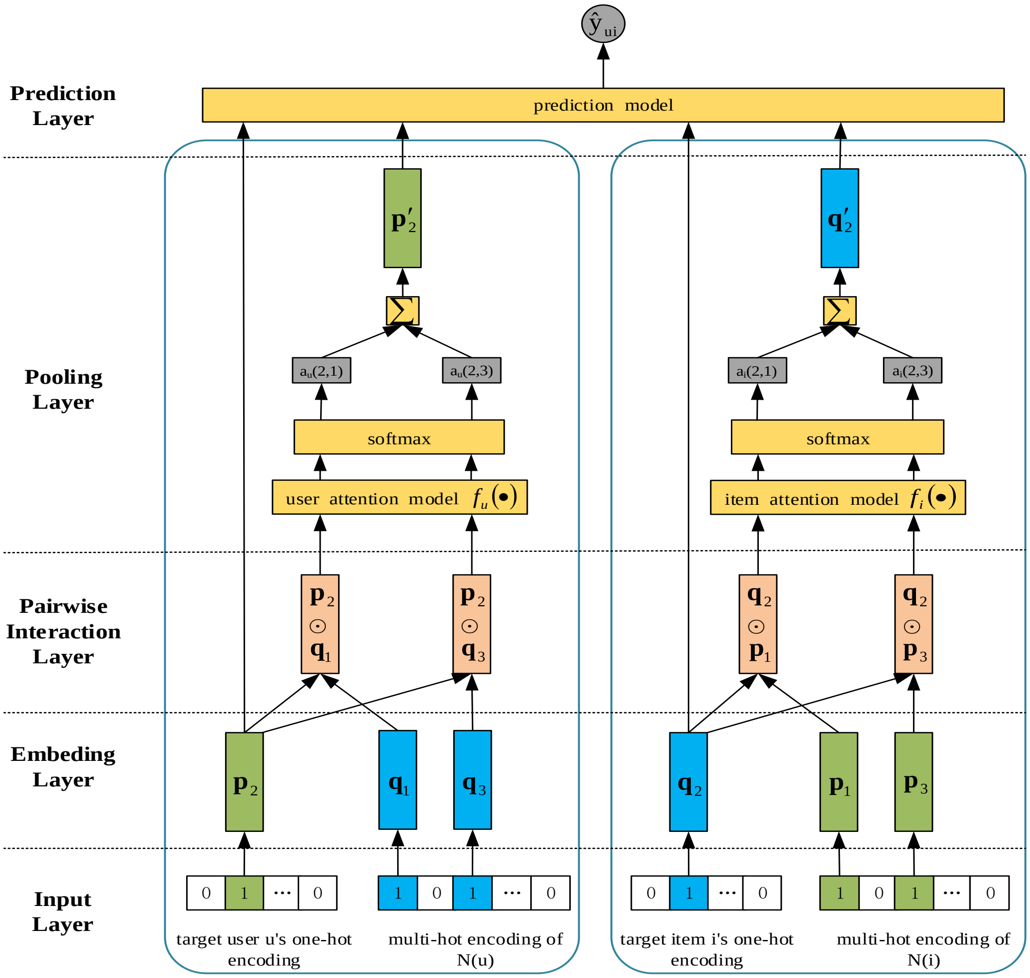

$\hat{y}_{ui}$
Prediction Layer
prediction model
Pooling Layer
$\mathbf{p}'_2$
$\sum$
$a_u(2,1)$
$a_u(2,3)$
softmax
user attention model $f_u(\bullet)$
$\mathbf{q}'_2$
$a_i(2,1)$
$a_i(2,3)$
item attention model $f_i(\bullet)$
Pairwise Interaction Layer
$\mathbf{p}_2 \odot \mathbf{q}_1$
$\mathbf{p}_2 \odot \mathbf{q}_3$
$\mathbf{q}_2 \odot \mathbf{p}_1$
$\mathbf{q}_2 \odot \mathbf{p}_3$
Embeding Layer
$\mathbf{p}_2$
$\mathbf{q}_1$
$\mathbf{q}_3$
$\mathbf{q}_2$
$\mathbf{p}_1$
$\mathbf{p}_3$
Input Layer
0 1 ··· 0
1 0 1 ··· 0
target user u's one-hot encoding
multi-hot encoding of N(u)
target item i's one-hot encoding
multi-hot encoding of N(i)

Fig. 1. The framework of our proposed AGMF model.

B. Pairwise Interaction Layer

Following the interaction way used in GMF, we also apply the widely-used element-wise product [3, 15, 16] to model the interactions of u and $\mathrm{N}(u)$ as well as i and $\mathrm{N}(i)$, 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.

C. 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.,

$$\mathbf{p}_u\prime = \sum_{j \in \mathrm{N}_u \backslash \{i\}} a_u(u,j)\mathbf{q}_j, \quad \mathbf{q}_i\prime = \sum_{k \in \mathrm{N}_i \backslash \{u\}} a_i(i,k)\mathbf{p}_k \quad (4)$$

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. $\mathbf{p}_u\prime$ and $\mathbf{q}_i\prime$ are the augmented latent vectors generated by the pooling layer, which will be used to enrich $\mathbf{p}_u$ and $\mathbf{q}_i$. It is also worth noting in Fig. 1 that the attention weights are not associated with the pairwise interacted embeddings. Instead, the attention weights are actually associated with the original item embeddings and user embeddings.

In this paper, we define $a_u(u,j)$ $(\forall j \in \mathrm{N}_u \backslash \{i\})$ and $a_i(i,k)$ $(\forall k \in \mathrm{N}_i \backslash \{u\})$ as the softmax normalization of the interaction scores between users and items:

$$a_u(u,j) = \frac{\exp(f_u(\mathbf{p}_u \odot \mathbf{q}_j))}{\sum_j \exp(f_u(\mathbf{p}_u \odot \mathbf{q}_j))} \quad (5)$$

$$a_i(i,k) = \frac{\exp(f_i(\mathbf{q}_i \odot \mathbf{p}_k))}{\sum_k \exp(f_i(\mathbf{q}_i \odot \mathbf{p}_k))} \quad (6)$$

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:

$$f_u(\mathbf{p}_u \odot \mathbf{q}_j) = \sigma(\mathbf{h}_u^\top(\mathbf{p}_u \odot \mathbf{q}_j)) \tag{7}$$

$$f_i(\mathbf{q}_i \odot \mathbf{p}_k) = \sigma(\mathbf{h}_i^\top(\mathbf{q}_i \odot \mathbf{p}_k)) \tag{8}$$

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.

D. Prediction Layer

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

$$\hat{y}_{ui} = \sigma(\mathbf{h}^\top((\mathbf{p}_u + \mathbf{p}_u\prime) \odot (\mathbf{q}_i + \mathbf{q}_i\prime))) \tag{9}$$

where $\mathbf{h}$ is the weight vector of the prediction model. Throughout this paper, we empirically use the sigmoid function as the activation function:

$$\sigma(\mathbf{z}) = \frac{1}{1 + \exp(-\mathbf{z})} \tag{10}$$

E. 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:

$$\mathcal{L} = -\sum_{(u,i)\in\mathcal{O}^+} \log \hat{y}_{ui} - \sum_{(u,j)\in\mathcal{O}^-} \log(1 - \hat{y}_{uj}) \tag{11}$$

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 to contruct the training set.

IV. 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.

Specifically, our experiments aim to answer the following questions:

- **RQ1** Does our proposed AGMF model outperform the state-of-the-art approaches?
- **RQ2** Does AGMF really benefit from the latent representations enriched by the augmented latent vectors?
- **RQ3** How do the key hyperparameters affect AGMF's performance?

A. Experimental Settings

1) Datasets: We conduct experiments on four publicly available datasets: MovieLens 1M (ML-1M)[1], Yelp[2], Amazon Movies and Tv (Movies&Tv)[3], 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 [7, 17] to process the other three datasets. For the Yelp dataset, we filter out the users and items with less than 10 interactions [17]. 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 I.

2) Comparing Algorithms: We compare AGMF with the following state-of-the-art counterparts:

- **SVD++** [6] It merges the latent factor model and the neighborhood model by enriching the user latent feature with the interacted items' latent features.
- **BPR-MF** [7] It trains the basic MF model by optimizing the Bayesian personalized ranking loss.
- **FISM** [18] It is an item-based approach, which factorizes the similarity matrix into two low-rank matrices.
- **MLP** [3] It learns the interactions between users and items by multi-layer perceptron.
- **GMF** [3] It generalizes the basic MF model to a non-linear setting.
- **NeuMF** [3] NeuMF is a combination of MLP and GMF. In this paper, we compare with the pre-training version of NeuMF, as this version provides better performance than NeuMF without pre-training [3].
- **ConvNCF** [14] 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.
- **DeepMF** [8] It employs deep neural networks to learn non-linear interactions of users and items (in the form of multi-hot representation).
- **NSNMF** [11] It learns the non-linear item representations for non-negative matrix factorization.

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 $[5\text{e}^{-5}, 1\text{e}^{-4}, 5\text{e}^{-4}, 1\text{e}^{-3}, 5\text{e}^{-3}]$, the embedding size K is chosen from $[16, 32, 64, 128]$, the regularization

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

[2] https://www.yelp.com/dataset/challenge

[3] https://www.amazon.com/

TABLE I
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

TABLE II
HR@5 AND NDCG@5 COMPARISONS OF DIFFERENT APPROACHES. THE BEST RESULTS ARE HIGHLIGHTED. IN ADDITION, ●/○ INDICATES WHETHER THE PERFORMANCE OF OUR AGMF METHOD IS STATISTICALLY SUPERIOR/INFERIOR TO THE COMPARED METHOD ON EACH DATASET (PAIRED t-TEST AT 0.05 SIGNIFICANCE LEVEL).

	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±0.001●	0.344±0.002●	0.700±0.001●	0.526±0.001●	0.633±0.002●	0.479±0.001●	0.671±0.001●	0.523±0.001●
SVD++	0.557±0.002	0.388±0.001●	0.664±0.001●	0.500±0.000●	0.606±0.001●	0.462±0.001●	0.607±0.002●	0.466±0.001●
FISM	0.528±0.002●	0.372±0.002●	0.691±0.002●	0.511±0.001●	0.583±0.003●	0.452±0.002●	0.592±0.001●	0.457±0.002●
MLP	0.526±0.003●	0.362±0.003●	0.671±0.002●	0.498±0.002●	0.570±0.002●	0.425±0.001●	0.588±0.003●	0.445±0.001●
GMF	0.540±0.001●	0.372±0.001●	0.676±0.002●	0.507±0.001●	0.569±0.004●	0.427±0.003●	0.620±0.004●	0.481±0.002●
NeuMF	0.548±0.003●	0.381±0.002●	0.695±0.002●	0.521±0.002●	0.596±0.001●	0.453±0.001●	0.629±0.001●	0.491±0.001●
ConvNCF	0.549±0.001●	0.391±0.001	0.708±0.002●	0.532±0.001●	0.634±0.002●	0.484±0.001●	0.673±0.002●	0.525±0.001●
DeepMF	0.469±0.005●	0.314±0.004●	0.692±0.006●	0.462±0.003●	0.582±0.004●	0.429±0.002●	0.574±0.006●	0.425±0.005●
NSNMF	0.524±0.002●	0.358±0.001●	0.671±0.004●	0.498±0.003●	0.566±0.001●	0.422±0.001●	0.602±0.004●	0.456±0.003●
AGMF	**0.561±0.002**	**0.393±0.002**	**0.714±0.001**	**0.545±0.001**	**0.649±0.002**	**0.499±0.001**	**0.690±0.002**	**0.549±0.001**

TABLE III
HR@10 AND NDCG@10 COMPARISONS OF DIFFERENT APPROACHES. THE BEST RESULTS ARE HIGHLIGHTED. IN ADDITION, ●/○ INDICATES WHETHER THE PERFORMANCE OF OUR AGMF METHOD IS STATISTICALLY SUPERIOR/INFERIOR TO THE COMPARED METHOD ON EACH DATASET (PAIRED t-TEST AT 0.05 SIGNIFICANCE LEVEL).

	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±0.001●	0.401±0.002●	0.833±0.002●	0.569±0.001●	0.765±0.001	0.527±0.001●	0.784±0.001●	0.560±0.001●
SVD++	0.713±0.003●	0.438±0.002●	0.785±0.001●	0.539±0.001●	0.728±0.001●	0.502±0.001●	0.719±0.001●	0.502±0.001●
FISM	0.699±0.003●	0.433±0.001●	0.824±0.002●	0.567±0.002●	0.708±0.002●	0.471±0.001●	0.729±0.002●	0.512±0.001●
MLP	0.703±0.003●	0.421±0.004●	0.805±0.003●	0.537±0.002●	0.703±0.002●	0.471±0.001●	0.712±0.003●	0.485±0.004●
GMF	0.711±0.001●	0.429±0.002●	0.809±0.002●	0.552±0.004●	0.712±0.006●	0.479±0.005●	0.729±0.006●	0.515±0.002●
NeuMF	0.727±0.004	0.443±0.002●	0.824±0.001●	0.560±0.003●	0.721±0.001●	0.493±0.002●	0.750±0.001●	0.529±0.001●
ConvNCF	0.713±0.002●	0.445±0.001●	0.836±0.001	0.572±0.002●	0.762±0.001●	0.525±0.001●	0.785±0.001●	0.562±0.001●
DeepMF	0.649±0.006●	0.374±0.003●	0.816±0.004●	0.543±0.003●	0.721±0.003●	0.479±0.002●	0.719±0.004●	0.474±0.003●
NSNMF	0.700±0.003●	0.421±0.002●	0.809±0.002●	0.542±0.002●	0.700±0.002●	0.465±0.001●	0.728±0.003●	0.499±0.003●
AGMF	**0.731±0.002**	**0.449±0.001**	**0.837±0.001**	**0.584±0.001**	**0.767±0.001**	**0.535±0.001**	**0.795±0.001**	**0.583±0.001**

parameter (that controls the model complexity) is chosen from $[1\mathrm{e}^{-5}, 5\mathrm{e}^{-6}, 1\mathrm{e}^{-5}, 5\mathrm{e}^{-5}]$, and the batch size is set to 256. For MLP and NeuMF that have multiple fully connected layers, we follow the tower structure of neural networks [3], and tune the number of hidden layers from 1 to 3. For ConvNCF[4], we follow the configuration and architectures proposed in [14]. All the models are trained until convergence or the default maximum number of epochs (according to the corresponding literature) is reached. Note that for the compared methods that are designed for explicit ratings, we provide 0/1 ratings in the implicit setting.

For our proposed AGMF model, no deep neural network is adopted, hence we do not need to tune the network structure. We initialize the weight vectors by the Xavier initialization [19], and initialize the embedding vectors using a uniform distribution from 0 to 1. The source codes of AGMF are provided in https://lfeng1995.github.io/Codes/AGMF.rar.

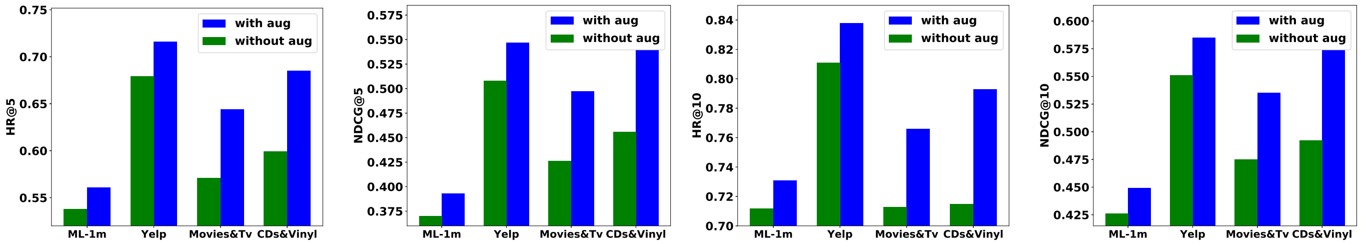
For training AGMF, we employ the Adaptive Moment Estimation (Adam), which adapts the learning rate for each parameter by performing small updates for frequent parameters and large updates for infrequent parameters. We implement AGMF using PyTorch[5], and the source code as well as the used datasets are 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.

B. Experimental Results

In this paper, we adopt the widely used *leave-one-out* evaluation method [7, 17] to compare AGMF with other approaches. Specifically, for each dataset, we holdout the latest interaction of each user as the test positive examples, and

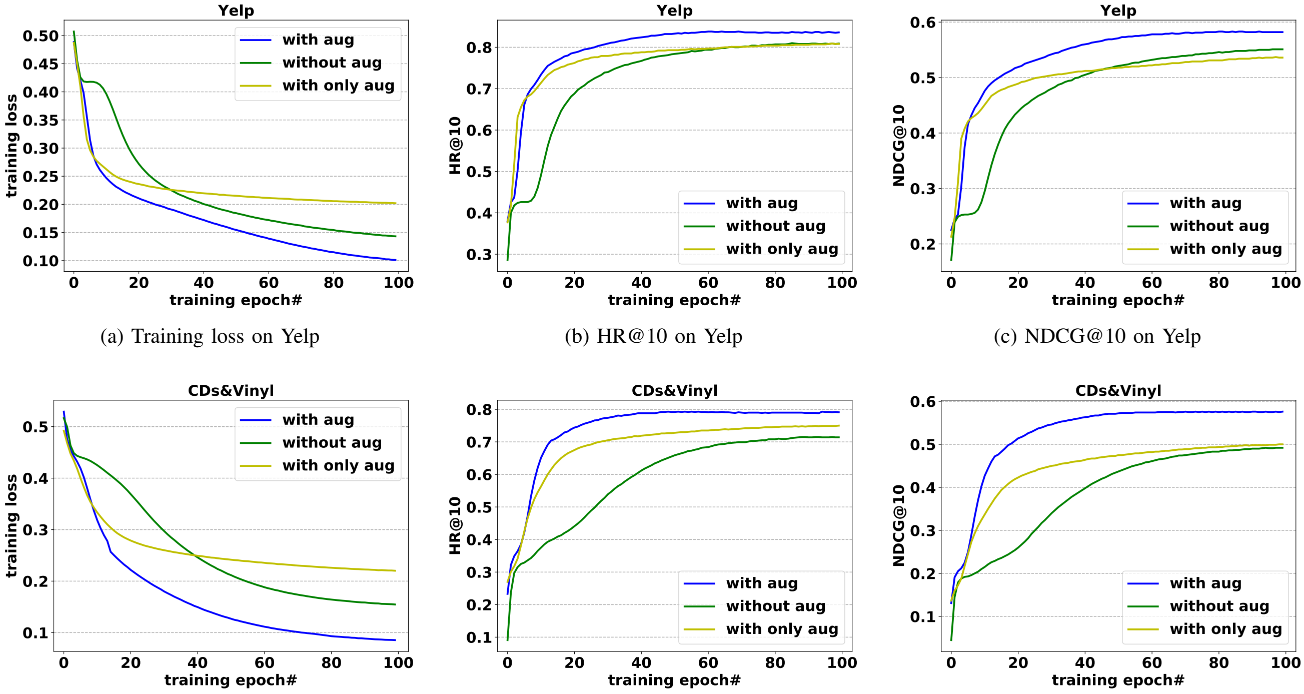
[4] https://github.com/duxy-me/ConvNCF

[5] https://pytorch.org/



(a) HR@5 (b) NDCG@5 (c) HR@10 (d) NDCG@10

Fig. 2. Comparison results of “with aug” and “without aug” on all the used datasets.



(a) Training loss on Yelp (b) HR@10 on Yelp (c) NDCG@10 on Yelp

(d) Training loss on CDs&Vinyl (e) HR@10 on CDS&Vinyl (f) NDCG@10 on CDs&Vinyl

Fig. 3. Detailed experimental results of ablation study in each training epoch.

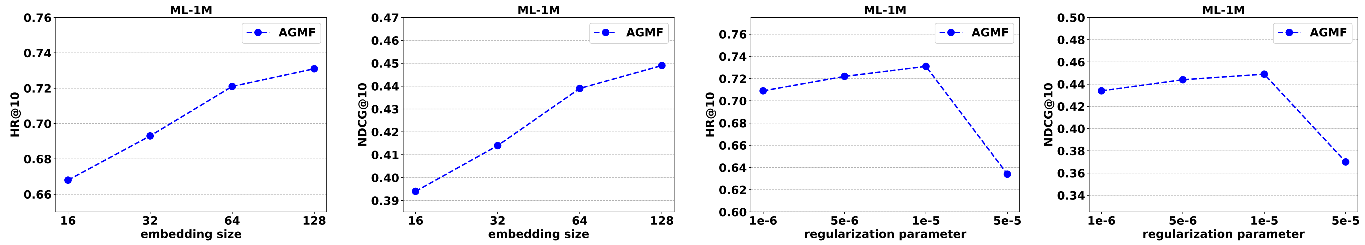
randomly select 99 items that the user has not interacted with as the test negative examples. In this way, all the algorithms make ranking predictions for each user based on these 100 user-item interactions.

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 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 [14].

1) Performance Comparison (RQ1): Table II and Table III show the top-k performance of all the algorithms when $k = 5$ and $k = 10$, respectively. We randomly initialize the model parameters and run the experiments for five times (different initializations for different trials). We report the mean performance with standard deviation for all methods in Table II and Table III. In addition, we adopt the paired t-test at 0.05 significance level to investigate whether our proposed AGMF method is statistically superior to the compared methods on each dataset. 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.
- AGMF also significantly outperforms SVD++, because SVD++ can only model linear interaction relationships between users and items while AGMF can capture the non-linear relationships. In addition, SVD++ only simply averages multiple embeddings of items, while AGMF applies the attention mechanism to automatically learn different weights of users and items.

2) Ablation Study (RQ2): As aforementioned, the GMF model makes prediction by $\hat{y}_{ui} = \sigma(\mathbf{h}^\top(\mathbf{p}_u \odot \mathbf{q}_i))$, while our



(a) HR@10 (b) NDCG@10 (c) HR@10 (d) NDCG@10

Fig. 4. Parameter sensitivity for AGMF on the ML-1M dataset.

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

Table II and Table III have clearly showed that AGMF significantly outperforms GMF. While the used GMF model for performance evaluation is provided by [3], which is implemented by Keras with a different initialization strategy. Therefore it may be slightly different from AGMF without $\mathbf{p}_{u\prime}$ and $\mathbf{q}_{i\prime}$. For completely fair comparison and pure ablation study, we conduct experiments using the codes of AGMF, to compare the performance of AGMF and AGMF *without* the augmented latent vectors $\mathbf{p}_{u\prime}$ and $\mathbf{q}_{i\prime}$. While with a slight abuse of naming, in this ablation study, we name our original AGMF model as "with aug", and name the AGMF model *without* the augmented latent vectors $\mathbf{p}_{u\prime}$ and $\mathbf{q}_{i\prime}$ as "without aug". In addition, we also compare with the AGMF model *with only* the augmented latent vectors $\mathbf{p}'_u$ and $\mathbf{q}'_i$, and we name the model $\hat{y}_{ui} = \sigma(\mathbf{h}^\top(\mathbf{p}'_u \odot \mathbf{q}'_i))$ as "with only aug".

Fig. 2 reports the comparison results of "with aug" and "without aug" on all the datasets. It can be seen that "with aug" always achieves better performance than "without aug", in terms of both evaluation metrics HR and NDCG. This observation clearly demonstrate that the augmented latent vectors $\mathbf{p}'_u$ and $\mathbf{q}'_i$ play an important role in generalized matrix factorization. Therefore, although our proposed AGMF model is a simple extension of the GMF model, it achieves signicantly better performance than GMF.

Furthermore, to thoroughly conduct ablation study, we also report the experimental results of "with aug", "without aug", and "with only aug" in each training epoch in Fig. 3. From Fig. 3, we can observe that:

- In terms of the evaluation metrics HR and NDCG, "with aug" consistently outperforms "without aug" and "with only aug" in each training epoch.
- In terms of the fitting ability, "with aug" achieves lower training loss than "without aug" and "with only aug".
- In terms of the convergence rate, "with aug" converges faster than "without aug".

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 further demonstrated. Moreover, these observations may bring new inspirations about how to properly integrate one-hot encoding and multi-hot encoding for effectively improving the recommendation performance.

C. Sensitivity Analysis (RQ3)

Here we investigate the influence of the embedding size K and the regularization parameter on AGMF. Note that we conduct sensitivity analysis by varying one parameter while keeping other parameters fixed at the best setting. The experimental results of parameter sensitivity analysis are reported in Fig. 4.

As can be seen from Fig. 4, AGMF achieves better performance as the embedding size increases. This is intuitive since a larger embedding size could potentially provide richer and more accurate representations of users and items. This observation also indicates that the bigger the model capacity is, the more complex the representations or relationships of users and items can be captured.

Besides, AGMF achieves slightly better performance as the regularization parameter increases at the beginning, while if the regularization parameter becomes overly large, AGMF will result in poor performance. Because if the regularization parameter is very small, the model may suffer from the overfitting issue, while if the regularization parameter is overly large, the model may suffer from the underfitting issue. Such observation agrees with the intuition that it is important to balance between overfitting and underfitting.

V. 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.

Acknowledgments

References

[1] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl *et al.*, "Item-based collaborative filtering recommendation algorithms." in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 285–295.

[2] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, 2009.

[3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.

[4] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[5] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 335–344.

[6] Y. 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*, 2008, pp. 426–434.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.

[8] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems." in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2017, pp. 3203–3209.

[9] S. Wang, J. Tang, Y. Wang, and H. Liu, "Exploring implicit hierarchical structures for recommender systems." in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2015, pp. 1813–1819.

[10] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[11] V. Krishna, T. Guo, and N. Antulov-Fantulin, "Is simple better? revisiting non-linear matrix factorization for learning incomplete ratings," in *IEEE International Conference on Data Mining Workshops*, 2018, pp. 1289–1293.

[12] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, 2011, pp. 287–296.

[13] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the International Conference on World Wide Web*, 2018, pp. 1583–1592.

[14] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua, "Outer product-based neural collaborative filtering," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3669–3675.

[15] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. S. Kankanhalli, "A^{3}ncf: An adaptive aspect attention model for rating prediction." in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3748–3754.

[16] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 3, pp. 17–25, 2018.

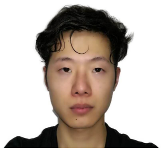
[17] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, pp. 549–558.

[18] S. Kabbur, X. Ning, and G. 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*, 2013, pp. 659–667.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 675–678.

Lei Feng received his B.E. degree in Computer Science from Southwest University, Chongqing, China, in 2017. He is currently pursuing his Ph.D. degree in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His main research interests include weakly supervised learning, statistical learning theory, domain adaptation, and data mining. He has published more than 10 papers on top conferences such as ICML, NeurIPS, AAAI, IJCAI, CVPR, and ICDM. He also serves as a program committee member for NeurIPS, ICLR, AAAI, and IJCAI.

Hongxin Wei received the B.E. degree in Software Engineering from Huazhong University of Science and Technology in 2016. He is currently pursuing his Ph.D. degree in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His main research interest is robustness in deep learning, including deep learning with noisy labels (weakly supervised learning) and adversarial examples.



Qingyu Guo received his Ph.D. degree from School of Computer Science and Engineering, Nanyang Technological University, Singapore in 2017. His main research interests include multi-agent system, computational game theory, adversarial machine learning, and data mining. He has published various papers on top conferences such as NeurIPS, AAAI, IJCAI, WWW, and AAMAS. He also serves as a program committee member for AAMAS.

Zhuoyi Lin received the B.S. degree in Electronic Engineering from Ming Chuan University, Taiwan, in 2017. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include recommender system and representation learning.

Bo An is a President's Council Chair Associate Professor in Computer Science and Engineering, Nanyang Technological University, Singapore. He received the Ph.D. degree in Computer Science from the University of Massachusetts, Amherst, in 2010. His current research interests include artificial intelligence, multiagent systems, computational game theory, reinforcement learning, and optimization. He has published over 100 referred papers at AAMAS, IJCAI, AAAI, ICAPS, KDD, UAI, EC, WWW, ICLR, NeurIPS, ICML, JAAMAS, AIJ and ACM/IEEE Transactions. Dr. An was the recipient of the 2010 IFAAMAS Victor Lesser Distinguished Dissertation Award, an Operational Excellence Award from the Commander, First Coast Guard District of the United States, the 2012 INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice, and 2018 Nanyang Research Award (Young Investigator). His publications won the Best Innovative Application Paper Award at AAMAS'12 and the Innovative Application Award at IAAI'16. He was invited to give Early Career Spotlight talk at IJCAI'17. He led the team HogRider which won the 2017 Microsoft Collaborative AI Challenge. He was named to IEEE Intelligent Systems' "AI's 10 to Watch" list for 2018. He is PC Co-Chair of AAMAS'20. He is a member of the editorial board of JAIR and the Associate Editor of JAAMAS, IEEE Intelligent Systems, and ACM TIST. He was elected to the board of directors of IFAAMAS and senior member of AAAI.