LARA: Attribute-to-feature Adversarial Learning for New-item Recommendation

Changfeng Sun[*,#], Han Liu[*], Meng Liu[*], Zhaochun Ren[*], Tian Gan[*], Liqiang Nie[*]
{hkldtgyf, hanliu.sdu, mengliu.sdu, nieliqiang}@gmail.com, {zhaochun.ren, gantian}@sdu.edu.cn
[*]School of Computer Science and Technology, Shandong University
[#]Pengcheng Laboratory, China

ABSTRACT

Recommending new items in real-world e-commerce portals is a challenging problem as the *cold start* phenomenon. To address this problem, we propose a novel recommendation model, *i.e.*, adversarial neural network with multiple generators, to generate users from multiple perspectives of items' attributes. Namely, the generated users are represented by attribute-level features. As both users and items are attribute-level representations, we can implicitly obtain user-item attribute-level interaction information. In light of this, the new item can be recommended to users based on attribute-level similarity. Extensive experimental results on two item cold-start scenarios, movie and goods recommendation, verify the effectiveness of our proposed model as compared to state-of-the-art baselines.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender System; Cold-start; Generative Adversarial Networks;

ACM Reference Format:
Changfeng Sun[*,#], Han Liu[*], Meng Liu[*], Zhaochun Ren[*], Tian Gan[*], Liqiang Nie[*]. 2020. LARA: Attribute-to-feature Adversarial Learning for New-item Recommendation. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371805

1 INTRODUCTION

Personalized recommendation has become an increasingly vital component in many online applications, such as e-commerce portals and content sharing applications [20, 28]. Aiming at providing a ranking list of items to a specific user, a personalized recommender calculates the predicted relevance between users and items [2, 21, 35]. Many techniques have been applied to improve the ranking quality in the personalized recommender systems. Traditionally, collaborative filtering approaches, *e.g.*, matrix factorization based methods [19, 31, 39] and k-nearest neighbor based methods [1, 26], have been proposed by computing the similarity

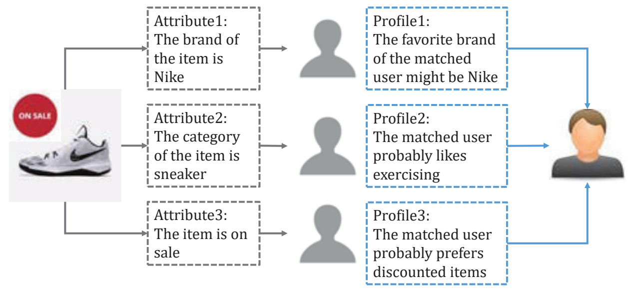

Figure 1: An example of the relationship between an item's attributes and user profiles. The left part lists the attributes of the shoes while the right part lists the possible user profiles that are inferred from the corresponding attributes.

between users and items through latent representations of them. Recently, several deep learning models [16, 18, 36] have been applied to improve the recommendation performance via computing scores of the pairs of users and items. However, the so-called cold-start problem [29] largely affects the recommendation performance, which makes the personalized recommendation challenging. Specifically, when a new item comes into the recommender, it is hard to make recommendations due to the absence of historical data about this user or item.

In this paper we focus on the item cold-start problem in personalized new-item recommendation scenario, *i.e.*, recommending a specific user new arrival items. Intuitively, we argue that one indispensable factor is the attributes of the new item [37]. Thus the key to tackle the cold-start problem in the new-item recommendation is to build a relationship between the new item and users. As each item has multiple attributes, each attribute often reflects a specific user profile in real-world personalized recommendation scenarios. An example is shown in Figure 1, based on attributes of the given on-sale Nike sneaker shoes, a potential user who is interested in this product may have various kinds of profiles, *e.g.*, Nike brand advocator, a sports enthusiast, or someone prefers discounted items. Inspired by this, we aim to optimize a matching problem between existing users and a virtual user profile generated from attributes of the targeting new-item. However, utilizing attribute information to reflect preference information of users is still a challenging issue: 1) To the best of our knowledge, the relationship between item attributes and user profiles has been untapped. 2) Given various attributes of an item, the generative module needs to simultaneously generate user profiles from various perspectives (*e.g.*, item brand, item category, and item price) of the new item. 3) The generated user profiles need to not only be close to the real user profiles but also be matched with the given items.

Zhaochun Ren and Meng Liu are corresponding authors.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02...$15.00
https://doi.org/10.1145/3336191.3371805

Generative adversarial networks (abbreviated as GAN) are well known for its powerful framework in learning generative models of arbitrarily complex data distributions [14]. A GAN involves a generator (abbreviated as G) and a discriminator (abbreviated as D). In the competition process, G attempts to generate samples which are similar to the existing data, whereas D aims at distinguishing between the real and generated samples. In recent years, GAN-based models have been successfully used to generate user-item interactions, *e.g.*, IRGAN [33], GraphGAN [32] and CFGAN [5]. In these approaches, GAN-based models learn from a user's latest behavior logs, and generate a user-item interaction representation (*i.e.*, a latent vector) for a given user. Each dimension of the vector reflects the probability that the user will buy the corresponding item. Thus, the generated interaction vectors could be regard as the representation of the corresponding users. Although these models have gained excellent performance, there is no attempt to use GANs for item cold-start recommendation, in which no prior events are known for certain items [12]. Moreover, all existing GAN-based approaches cannot be directly applied to the new-item recommendation: 1) These models aimed to generate interaction information from users' latest purchase history, however, the lack of historical information is the key problem in the item cold-start scenario. 2) Since modern online platforms have millions of items, the generated user-item interaction vector would be very large and sparse, however, it is hard for GAN to generate the large and sparse vectors.

Regarding to the aforementioned analysis, in this paper we build up an end-to-end adversariaL neurAl netwoRk with multi-generAtors (dubbed LARA) to map item attributes to user representation. Particularly, to generate the representation of the user who likes the new item, we design a generative model with multi-generators to generate user profiles from different perspectives of the new item, respectively. Based upon all the generated user profiles, a neural network is introduced to output the final user representation. At this step, the generated user is the most appropriate user for the given item. Afterwards, a discriminator is introduced to distinguish the real user profile and the generated profile. At this stage, we introduce three kinds of training samples, *i.e.*, the user who is interested in the item according to the ground-truth data, the user generated by the item's attributes, and the user who is not interested in the item according to the ground-truth data. The discriminator trained with these samples can guide the generator to generate user profile which is not only close to the real profile but also relevant to the given item. Finally, we can recommend the new item to users with similar attributes to the generated user.

To sum up, the key contributions of this work are three-fold:

- We design a novel item cold-start recommendation system by jointly modeling user profiles from different attributes of the item using an adversarial neural network with multi generators. To the best of our knowledge, this is the first work to learn attribute level item-user mapping by GAN.
- We introduce some tips to improve the performance of our system. To avoid the sparsity problem, we introduce a new method to represent users. Besides this, to push the generated users be close to the real users and be match with the given items, we introduced three samples to train the model.
- We construct a large brand-new dataset based on customer's purchase records from a retail supermarket. This dataset contains transaction records in three months, and each item has multiple attributes, such as category, brand, and price. We evaluate our proposed model on two real-world datasets to comparatively demonstrate the superiority of our model.[1].

2 RELATED WORK

In this section, we briefly review related work on cold-start recommendation and generative adversarial network respectively.

2.1 Cold Start Recommendation

One of the most challenging problem in existing recommender systems is the cold-start issue [42]. To solve this problem, additional information about entities (*e.g.*, age, gender, occupation of a user and category, and description of an item) must be considered during the recommendation process. Inspired by this, several models have been proposed. Gantner *et al.* [12] proposed a method that maps item attributes to the latent factors of a matrix factorization model. With such mapping, the factors trained by standard techniques can be applied to tackling the new-item problem. Park *et al.* [24] gave another solution to utilize items' attribute information. They constructed profiles for item pairs by outer product over their individual information, and built predictive model in regression framework on pairwise user preferences. Besides this, there are many works combine the additional information with other information (*e.g.*, collaborative information) to solve the cold-start issue. Saveski *et al.* [27] introduced a method that combines the content of the item and collaborative information into a unified matrix factorization framework. Specifically, given the description of a new item (*e.g.*, the content of a news article), this framework could project it into a low dimensional common space and infer the users who prefer it. By doing so, they overcame the item cold-start problem.

With the development of deep learning, the deep neural network (DNN) is applied to solving cold-start recommendation. Wei *et al.* [38] proposed a framework which tightly coupled CF approach and DNN. The DNN is utilized to extract features of items, while the CF approach is modified to predict ratings for cold start items via their extracted features. Shekasta *et al.* [30] also proposed a DNN based recommendation system, which designs a purchase intent session-based algorithm to predict the purchase intent for cold start session-based scenarios. Particularly, this approach uses word embeddings to model the content of items and then feeds these embeddings to a recurrent neural network. Beside this, in [11], to make accurate recommendations, the additional information is used to learn a robust feature representation by a DNN coupled with a latent factor model. And with this representation, the proposed model could generalize to new items.

2.2 Generative Adversarial Network

In order to solve the generative problem in the field of computer vision, Goodfellow *et al.* [14] proposed a model named Generative Adversarial Network, and it has attracted much attention for its powerful framework in learning generative models for arbitrarily

[1]The code and the dataset are available at https://anonymous274.wixsite.com/lara

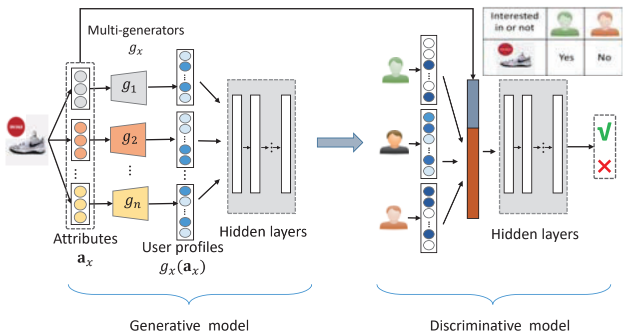

Figure 2: The architecture of our proposed LARA model. The generative model with multi-generators aiming to generate the user profiles corresponding to item attributes. The discriminative model is designed to distinguish the generated profiles and real profiles.

complex data distributions. Moreover, it has already gained huge success in many tasks, such as image generation [22, 40], natural language generation [15, 34, 41], and music generation [9, 10]. In order to control modes of the data being generated, Mirza *et al.* [23] proposed a model named conditional Generative Adversarial Network. They argued that generative adversarial net can be extended to a conditional model if both the generator and discriminator are conditioned on some extra information, such as class labels and data from other modalities.

Recently, there are many efforts to apply GAN to the field of recommender systems. IRGAN [33] and GraphGAN [32] are the pioneering methods that demonstrated the potential of GAN in collaborative filtering framework. Their main idea is that, given a user, let G generate the probability of items that the user might purchase, and then it samples the item which has the highest probability with the expectation to fool D, while let D discriminate ground truth item from the sampled one. After several this adversarial process, G will eventually capture the true distribution of users' preferences over items. It is worth mentioning that the policy-gradient based reinforcement learning is employed to guide G by using D's output as a reward signal [5]. By this way, IRGAN and GraphGAN have achieved a promising accuracy in recommendation on real-world data. However, as the training progresses, the G will sample more and more items that are exactly the same as those in the ground truth since the item indices are discrete and finite. In order to solve the problem arising in IRGAN and GraphGAN, Chae *et al.* [5] suggested a new direction of vector-wise adversarial training and proposed a framework called CFGAN. Different from IRGAN and GraphGAN, the G of CFGAN does not sample item indices, it just generates the probability vectors foritems to fool the D. In these works, the user presentations are constructed by their historical interactions with items. Consequently, they are not suitable for the cold start scenario which lacks of historical purchase records.

3 METHOD

In this section, we first formulate the task of item cold-start recommendation. And then we detail each part of our scheme. At last, we demonstrate how our framework can be utilized to solve the item cold-start problem in new-item recommendation.

3.1 Problem Formulation

Suppose that an online platform has a set of items $\mathcal{I} = \{I_1, I_2, ..., I_n\}$ and a set of users $\mathcal{U} = \{U_1, U_2, ..., U_m\}$. Each item $I_i \in \mathcal{I}$ is represented by its attributes $\mathcal{A}_i = \{\mathbf{a}_{i1}, \mathbf{a}_{i2}, \ldots, \mathbf{a}_{ik_i}\}$, where k_i denotes the number of attributes. Each attribute refers to a scalar (*e.g.*, the category ID or the brand ID of the item) or a vector (*e.g.*, the TF-IDF of item description). In the item cold-start scenario, given a new item I^t with k attributes, we aim to generate the representation of the user who likes the new item according to its attributes, *i.e.*, $\mathbf{u}^t \in \mathbb{R}^k$. This is a user-attribute interaction vector, where the i-th element reflects if the user has interaction with the i-th attribute of the item.

The scheme of our proposed approach is illustrated in Figure 2. It is made up of two parts: generative model and discriminative model. The generative model aims to generate the profiles of the user who may interested in the given new items, while the discriminative model aims to discriminate the generated user profiles and real user profiles.

3.2 Generative Model

The generative model aims to generate a vector denoting the user who most probably interested in the given conditional item I^c, and each dimension of the user vector is expected to correspond to one attribute in the attribute set collected from all items. Novelly, we bring a separately generating and sum-up structure to the generative model. In this way, the generative process is separated into two stages. Firstly, each attribute $\mathbf{a}_i^c$ of the conditional item is assigned to a special generator g_i, which accepts the mission to generate the profile of the latent user from a certain perspective. Afterwards, a neural network G is employed to incorporate all generated user profile vectors and output the completely generative user feature vector. The entire process in our generative model is formulated as follows,

$$\mathbf{u}^c = G\big(g_1(\mathbf{a}_1^c), g_2(\mathbf{a}_2^c), ..., g_k(\mathbf{a}_k^c)\big), \tag{1}$$

where $\{\mathbf{a}_1^c, \mathbf{a}_2^c, ..., \mathbf{a}_k^c\}$ denote attributes of the conditional item, user profiles generated in first stage are formulated as $g_i(\mathbf{a}_i^c)$, $\mathbf{u}^c$ refers to the user vector generated by our generative model from the conditional item attributes, and g_i is the generator of the i-th attribute. The structure of g_i should be adjust to the item attribute. In this paper, since the item attributes in dataset are discrete values, we set the g_i as trainable embedding matrices.

Specifically, if we have obtained the generated user profiles from different perspectives, *i.e.*, $g_i(\mathbf{a}_i^c)$. We hence can derive a holistic representation of the user by employing a neural network to the concatenation of all the generated user profile vectors, *i.e.*, $\mathbf{z}_1$, as follows,

$$\begin{cases} \mathbf{z}_2 = f_2(\mathbf{W}_2\mathbf{z}_1 + \mathbf{b}_2), \\ \cdots\cdots \\ \cdots\cdots \\ \mathbf{u}^c = f_L(\mathbf{W}_L\mathbf{z}_{L-1} + \mathbf{b}_L), \end{cases} \tag{2}$$

where $\mathbf{W}_j$, $\mathbf{b}_j$, and f_j denote the weight matrix, bias vector, and activation function of the j-th layer, respectively.

3.3 Discriminative Model

On the contrary, the discriminative model tries to discriminate the well-matched user-item pairs from ill-matched ones. In order to improve the discriminability of the discriminative model, we introduce three kinds of training pairs: the conditional item I^c and the generated user $\mathbf{u}^c$, the conditional item I^c and a true user $\mathbf{u}^+$, and the conditional item I^c and a false user $\mathbf{u}^-$. Here the user $\mathbf{u}^c$ is generated by generative model whose input is the conditional item I^c.

In the following, we will detailed the construction of the user $\mathbf{u}^+$ and $\mathbf{u}^-$. The true user $\mathbf{u}^+$ is the user who is interested in I^c according to the ground-truth data. Specifically, we traverse the dataset to get the user who has interaction with I^c and we regard this user as the true user $\mathbf{u}^+$. To represent the user, we firstly build a zero vector of which the dimension is the number of attributes in the dataset. Afterwards, we search the dataset to find items which have interactions with the user and fetch the attributes of these items to construct an attribute set. Finally, we traverse the attribute set and set the corresponding value in the zero vector as 1 to represent the user $\mathbf{u}^+$. As for the false user $\mathbf{u}^-$ who dislikes I^c according to the ground-truth data, we traverse the dataset to obtain the user who has no interaction with I^c and we regard it as the false user $\mathbf{u}^-$. Similar to the true user $\mathbf{u}^+$, we also represent the user $\mathbf{u}^-$ by the attributes of items which have interactions with it.

The goal of the discriminator is to distinguish the $(\mathbf{u}^+, I^c)$ pair from the other two kinds of pairs as much as possible. Therefore, the labels of the three kinds of training pairs are set as following,

$$y(\mathcal{T}) = \begin{cases} 1, & \mathcal{T} = (\mathbf{u}^+, I^c) \\ 0, & \mathcal{T} = (\mathbf{u}^c, I^c) \\ 0, & \mathcal{T} = (\mathbf{u}^-, I^c) \end{cases}, \tag{3}$$

where the label of the positive pair $(\mathbf{u}^+, I^c)$ is set to 1, otherwise, we set it 0.

The discriminative model D estimates the probability of user being relevant to conditional item, which is computed by the sigmoid function of the discriminator score, as,

$$D(\mathbf{u}|I_n) = \sigma(d_\phi(\mathbf{u}, I_n)) = \frac{\exp(d_\phi(\mathbf{u}, I_n))}{1 + \exp(d_\phi(\mathbf{u}, I_n))}, \tag{4}$$

where ϕ denotes the parameters of D. To be more specific, we employ a neural network with three layers as d_ϕ,

$$\begin{cases} \mathbf{z}_2^d & = f_2^d(\mathbf{W}_2^d \mathbf{z}_1^d + \mathbf{b}_2^d), \\ \mathbf{z}_3^d & = f_3^d(\mathbf{W}_3^d \mathbf{z}_2^d + \mathbf{b}_3^d), \\ d_\phi(\mathbf{u}, I_n) & = f_{out}^d(\mathbf{W}_{out}^d \mathbf{z}_3^d + \mathbf{b}_{out}^d), \end{cases} \tag{5}$$

where $\mathbf{z}_1^d$ is the concatenation of the user feature vector and all the attribute vectors of item I_n, and $\mathbf{W}_j^d$, $\mathbf{b}_j^d$, and f_j^d denote the weight matrix, bias vector, and activation function for the j-th layer, respectively.

3.4 Overall Objective

Inspired by the idea of conditional GAN, the objective function of our model is letting the two modules play a minimax game to unify themselves. The generative model makes effort to generate the relevant user that is seemed to be interested in the conditional item and simultaneously similar to the ground truth users, as a result it can cheat the discriminative model. While the discriminative model tries to make a distinction between the ground truth relevant users and the fake one generated from the opposed generative model. Compared with existing GAN-based recommendation methods, our method introduce third kind of training tuples, consisting of a conditional item and a false user who is uninterested in the conditional item. The reason of introducing this term is that the observed ground truth relevant user-item tuples can only make the generative user approximate the right users for conditional item, nevertheless, the observed ground truth ill-suited user-item tuples can turn the generative user away from the false users for conditional item. Formally, the objective function of our model is constructed as,

$$\begin{aligned} \mathcal{L}^{G^*, D^*} = \min_\theta \max_\phi \sum_{n=1}^{N} \big(& \mathbb{E}_{\mathbf{u}^+ \sim p_{true}(\mathbf{u}^+|I_n)} [\log(D(\mathbf{u}^+|I_n))] \\ & + \mathbb{E}_{\mathbf{u}^c \sim p_\theta(\mathbf{u}^c|I_n)} [\log(1 - D(\mathbf{u}^c|I_n))] \\ & + \mathbb{E}_{\mathbf{u}^- \sim p_{false}(\mathbf{u}^-|I_n)} [\log(1 - D(\mathbf{u}^-|I_n))] \big), \end{aligned} \tag{6}$$

where the generative model G is written as $p_\theta(\mathbf{u}^c|I_n)$, θ denotes the total parameters of G, and N denotes the number of conditional items in our training set.

3.5 Optimization

By iteratively minimizing and maximizing the same objective function in Eqn.(6), our method can learn the optimal parameters of the generative model and the discriminative model, respectively.

3.5.1 Optimizing Discriminative Model. The objective of discriminative model is to maximize the log-likelihood of correctly distinguishing the right users from ill-suited and generative users of the conditional item. With the observed relevant users, the observed ill-suited ones and the fake ones generated from the current optimal generative model $p_{\theta^*}(\mathbf{u}^c|I_n)$, the optimal parameters of the discriminative model can be obtained by,

$$\begin{aligned} \phi^* = \arg\max_\phi \sum_{n=1}^{N} \big(& \mathbb{E}_{\mathbf{u}^+ \sim p_{true}(\mathbf{u}^+|I_n)} [\log(\sigma(d_\phi(\mathbf{u}^+, I_n)))] + \\ & \mathbb{E}_{\mathbf{u}^c \sim p_{\theta^*}(\mathbf{u}^c|I_n)} [\log(1 - \sigma(d_\phi(\mathbf{u}^c, I_n)))] + \\ & \mathbb{E}_{\mathbf{u}^- \sim p_{false}(\mathbf{u}^-|I_n)} [\log(1 - \sigma(d_\phi(\mathbf{u}^-, I_n)))] \big). \end{aligned} \tag{7}$$

The above optimization can be solved by stochastic gradient descent.

3.5.2 Optimizing Generative Model. In contrast, the generative model $p_\theta(\mathbf{u}^c|I_n)$ aims to minimize the objective function, it learns the ability to conceive how a user vector looks like when only given the attributes of the item. Based on the observed feature vectors of the right users and the false ones for the conditional item, the generator forms the plausible user feature vector, both approximating the right users and away from the false users, to fool the discriminative model. It is worth noting that different from the existing GAN-based recommendation, our generative model directly generates the representations of users.

While optimizing the generative model, the discriminator D is supposed to be fixed. As a result, the optimization objective is just to minimize the term correlated with the generative model in Eqn.(6), as,

$$\begin{aligned}\theta^* &= \arg\min_\theta \sum_{n=1}^{N} \mathbb{E}_{\mathbf{u}^c \sim p_\theta(\mathbf{u}^c|I_n)}[\log(1-\sigma(d_\phi(\mathbf{u}^c, I_n)))] \\ &= \arg\max_\theta \sum_{n=1}^{N} \mathbb{E}_{\mathbf{u}^c \sim p_\theta(\mathbf{u}^c|I_n)}[\log(1+\exp(d_\phi(\mathbf{u}^c, I_n))].\end{aligned} \tag{8}$$

3.6 Recommendation via the Generative Model

After the alternating optimization is completed, given the attributes of a new item, the generative model can generate the attribute-level representation of the user who tends to purchase this new item. Each element of the vector denotes whether the user prefers this particular attribute, which also reflects the user's profile. Then we select the top-N users who are most similar to the generative user from the user set $\mathcal{U}$ as the objects of recommendation [25].

As users are represented by vectors, the similarity between two vectors is regarded as the similarity between the corresponding users [3]. We can use various vector similarity measurements as the metrics to select the most similar users, such as Euclidean distance, Mahalanobis distance, and cosine between two vectors. In this paper, the cosine value is selected as the similarity measurement.

4 EXPERIMENTS

To thoroughly justify the effectiveness of our proposed recommendation model, we carry out extensive experiments to answer the following research questions:

- **RQ1**: Can our LARA approach outperform the state-of-the-art baselines for item cold-start recommendation?
- **RQ2**: Is the attribute-level user representation helpful for boosting the recommendation performance?
- **RQ3**: Does eliminate the conditional item and false user pair affect the final results?

4.1 Datasets

To verify the effectiveness of our proposed method, we apply two real-world datasets in our experiments: the MovieLens 1M dataset[2] and the Inzone dataset[3].

MovieLens. As one of the most widely applied benchmark datasets on recommender systems, the MovieLens 1M dataset can be utilized to evaluate the effectiveness of LARA. In our experiments, to remove noise and sparseness, we retain the interactions that rating value is higher than 4 as positive feedbacks, and treat other entries as unknown feedbacks. Also, the genre information included in the MovieLens dataset is regarded as the item attribute information. In total, there are 18 genres (*e.g.*, action, fantasy, romance, and adventure, etc.) among movies, and each movie may have several genre tags. For example, *Marvel's The Avengers* has been involved three genres: action, fantasy, and adventure. But another famous movie, *i.e.*, *Titanic*, has two genres: romance and horror. In order to unify the number of genre tags for all the movie, we set all the movies have these 18 genre attributes. If one movie is in a specific genre, the value of this genre tag is one, otherwise, the value is zero.

Table 1: Statistics of the two datasets.

Datasets	# Users	# Items	# Interactions	#Attributes
MovieLens	6,039	3,883	1,000,209	18
Inzone	54,765	1,047	198,488	432

Inzone. This dataset describes retail transaction records and product attributes from the Inzone[4], a real-world offline retailer. It contains 198,488 transaction records and 432 attribute tags across 1,047 products, each of which has been purchased by at least 100 different users . These data were created by 54,765 users between March 01, 2017 and May 31, 2017. Users are selected at random for inclusion and no demographic information is included. Each user is represented by an id, and no other information is provided. Since the number of users in this dataset is much larger than those in existing datasets, it is more suitable for verifying the performance of item-based recommendation systems. Each product is represented by an id and has some attributes. The 432 product attributes of products are consist of two parts:178 brand tags (*e.g.*, Nike, Coca Cola, and Lining) and 254 category tags (*e.g.*, fruit, seafood, and T-shirt). Therefore, we set each product with two attribute tags, which is different from MovieLens. The first one is brand tag which has 178 values, while the second one is category tag which has 254 values.

Table 1 summarizes their detailed statistics. And the data of each dataset is stored in a matrix with three columns: userId, itemId, and item attributes. There are 1,000,209 rows in MovieLens and 198,488 rows in Inzone. Each row in the matrix means an interaction between one user and one item, and the item has its own attributes.

For each dataset, we randomly split its items into two parts: 80% for training items and the rest 20% for test items, and then we construct the training dataset containing the interactions related to the training items and the test dataset containing the interactions related to the test items.

4.2 Experimental Settings

4.2.1 Evaluation Metrics. To thoroughly measure our model and the baselines, we employ precision (P@K), mean average precision (M@K), and normalized discounted cumulative gain (NDCG@K) as the evaluation metrics [6] to measure the item cold-start recommendation performance from different angles. The metric P@K focuses on how many correct users are included in the recommendation list. The M@K and NDCG@K accounts for the ranked position of correct users in the recommendation list.

4.2.2 Implementation details. We adopt 18 generators on the MovieLens dataset and 2 generators on the Inzone dataset on account of the number of attributes tags in two datasets. Experimentally, we utilize the sigmoid function as the activation function for all neural networks; for initializing the weights and bias in our model, we utilize the popular Xavier's approach [13]. Specifically, we vary the number of the hidden layers of the generative and the discriminative model with $\{1, 2, 3, 4, 5\}$, the number of neurons in each hidden

[2]https://grouplens.org/
[3]the dataset is available at https://anonymous274.wixsite.com/lara.
[4]http://inzonegroup.cn

layer with {100, 200, 300, 400, 450}, the learning rate of the optimizer with {1E-05, 1E-04, 1E-03, 1E-02}, and the size of the training mini-batch with {128, 256, 512, 1024}.

4.3 RQ1: Performance Comparisons

To answer the first question, we compare the performance of our proposed method with the state-of-the-art cold-start recommendation methods. The baselines are listed as follows:

- **UserPop**: It is the simplest non-personalized recommendation algorithm that ranks users by the descending order of popularity (*i.e.*, the number of items purchased by the user).
- **BPR-kNN** [12]: This is a two-step method for cold-start recommendation. It uses Bayesian Personalized Ranking (BPR) to factorize the collaborative rating matrix, and then learns latent factor representations of users and items. Meanwhile, this method learns a mapping between item attributes and the corresponding latent factors by K-Nearest-Neighborhood. This mapping can be utilized to infer the factors of a new item from its attributes, and then the obtained factors are adopted to make predictions for cold-start recommendation.
- **LCE** [27]: This is a hybrid recommendation approach that combines both the content and the collaborative information. It decomposes the content and the collaborative matrix in a common low-dimensional space. In this way, it can project the content of a new item in this common low-dimensional space and infer which one the users are most likely to be interested in. We adopt the publicly available Matlab implementation[5] of the LCE algorithms.
- **NFM** [17]: This model is the state-of-art model for sparse data prediction. It enhances factorization machines by modelling higher-order and non-linear feature interactions. In order to adapt it to solve the item cold-start problem, we rank the prediction scores of all users for one new item and regard the users which have higher scores as the recommendation.
- **DNN**: This baseline makes some changes based on our proposed method. It adopts the same generative model to generate the users. The main difference is that the loss function is designed to minimize the Euclidean distance between the generated users and the real users, namely this approach regards the loss function as a linear regression problem rather than a logistic regression one.

For each baseline model, we test various values of its hyper-parameters (*e.g.*, the dimension of the latent factor vector in MF-based models) based on the values mentioned in the paper and select a set of values that provide the highest accuracy. We conduct an empirical study to investigate whether our proposed model can achieve better recommendation performance. To further demonstrate our improvement, we compare the second best results with our results to compute the improvement values. The results of all methods on two datasets are presented in Table 2. We have following several key observations.

First and foremost, **LARA** achieves the best performance, substantially surpassing all the baselines on both datasets. Particularly, **LARA** shows consistent improvements over **LCE**, **BPR-KNN**, and

[5]https://github.com/msaveski/LCE

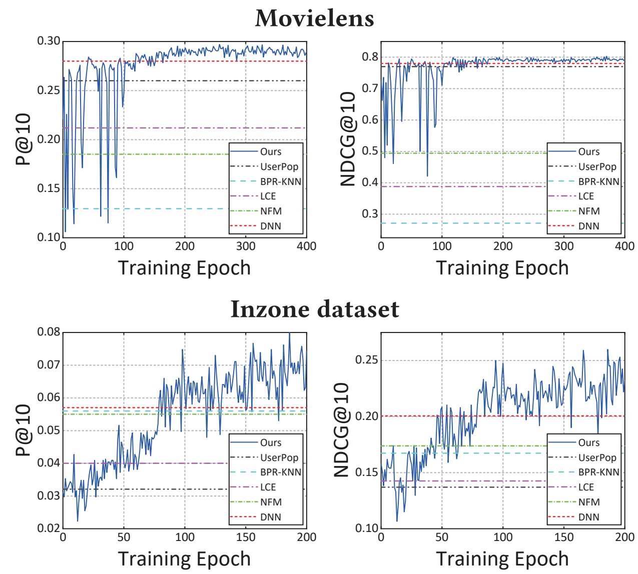

Figure 3: Learning curves of our method on two datasets.

NFM, verifying the importance of utilizing attribute-level user representation and employing the attribute-level item-user interaction to recommend item. In addition, the improvement over the **DNN** indicates the effectiveness of our discriminator and simultaneously demonstrates that our proposed discriminator can force the model generate well user representation.

Second, when performing item cold-start recommendation, **DNN** outperforms the **BPR-KNN**, **LCE**, and **NFM**, especially on the Movielens dataset. The observed results make sense since **DNN** is capable of exploiting the attribute-level user representation and directly establishes mappings between item attributes and users.

Third, we observe that **BPR-KNN**, **LCE**, and **NFM** have unstable performance on two datasets. The **BPR-KNN** and **LCE** outperform others on Inzone dataset and Movielens dataset respectively. These three method only use an id to present a user and then try to predict the interactions (*i.e.*, ratings) between users and new items, that means they ignore the relationships between item attributes and user preference. That is why they achieves the unstable performance on two datasets. It hence verifies the necessity of considering user preference information for item cold-start recommendation.

Lastly, the **UserPop** performs poorly than the other baselines since it makes recommendation according to the number of items purchased by the user, namely it ignores the information of the item and the preference of the user.

Moreover, since adversarial training is widely regarded as an effective but unstable technique, we further analysis the learning trend of our proposed method. Figure 3 shows the learning curves of our model on the two datasets. Here we only show the performance measured by P@10 and NDCG@10 as the other evaluation metrics reveal the similar trend. From Figure 3, we can observe that after about 120 epoches on Movielens dataset and 80 epoches on Inzone dataset of adversarial training, both P@10 and NDCG@10 converge and our method consistently outperforms other baseline methods.

Table 2: Performance comparison between our proposed model and several state-of-the-art baselines over two datasets. And values in bold black denote the best performance.

Datasets	Inzone						Movielens					
Metrics	P@10	P@20	M@10	M@20	NDCG@10	NDCG@20	P@10	P@20	M@10	M@20	NDCG@10	NDCG@20
UserPop	0.0321	0.033	0.0925	0.0938	0.1370	0.1772	0.2699	0.2400	0.4379	0.3989	0.7773	0.8130
BPR-KNN	0.0558	0.0523	0.1015	0.1106	0.1673	0.2054	0.1296	0.1250	0.1990	0.1851	0.2712	0.2960
LCE	0.0400	0.0340	0.0978	0.1014	0.1426	0.1917	0.2119	0.1975	0.3824	0.3378	0.3880	0.4233
NFM	0.0553	0.0491	0.1125	0.1200	0.1739	0.2180	0.1851	0.1408	0.3975	0.3633	0.4942	0.4910
DNN	0.0567	0.0542	0.1354	0.1401	0.2006	0.2451	0.2832	0.2427	0.4502	0.4130	0.7903	0.8050
Ours	**0.0781**	**0.0723**	**0.1684**	**0.1674**	**0.2541**	**0.2715**	**0.3006**	**0.2658**	**0.4776**	**0.4345**	**0.8092**	**0.8490**
Improvement	37.74%	25.03%	24.37%	19.48%	26.67%	10.77%	6.14%	9.51%	6.08%	5.20%	2.39%	5.46%

4.4 RQ2: Effectiveness of the Attribute-level User Representation

Another contribution in this paper is using the attribute information to represent users. In order to verify the effectiveness of the attribute-level user representation, we propose a new baseline model of which the input is the item attribute information and the output is the interaction-level user representation, *i.e.*, the dimension of the user representation equals to the number of items. We compare this baseline with our framework.

From Table 3, we find that utilizing interaction-level user representation degrades the recommendation results. We think the reasons are that: 1) As many items have similar attributes, the user-item interaction vector generated by baseline may not reflect the user's true preference, namely the items that the user dislikes may be selected as they have similar attributes with the given item. 2) In most situations, GANs have an advantage in generating dense feature vectors, just as the original GANs are applied to generating images [7, 8]. However, in recommender systems, the sparsity of the user-item interaction is inevitable since each user only has interaction with a small part of the whole items [17]. In the case of sparsity, while G tries to deceive D by generating user vectors resembling that of the ground truth, it may lead to a useless solution, namely simply predicts all the elements of the user vector to be 0. Therefore, the GAN model may fail to generate the sparse interaction-level user representation. Therefore, our user representation performs better as the user-attribute interaction vector shows significant increase on the density over the user-item interaction. This justifies the effectiveness of our proposed attribute-level user representation.

To gain the deep insights into our proposed attribute-level representation, We illustrate the learning curves of both methods on the two datasets, and the results are shown in Figure 4. It can be seen that the results are consistent with those in Table 3. Our model shows a significant improvement over the interaction-level representation. Besides this, we can observe that our model has more stable performance on both datasets. This phenomenon also justifies the advantage of GANs in generating dense feature vectors.

Table 3: Performance comparison between the attribute-level user representation and the interaction-level user representation on the Inzone dataset and the Movielens dataset.

Inzone dataset	P@10	M@10	NDCG@10
Attribute	**0.0781**	**0.1684**	**0.2541**
Interaction	0.0605	0.1631	0.2200
Improvement	28.92%	3.24%	15.50%
Movielens	P@10	M@10	NDCG@10
Attribute	**0.3006**	**0.4776**	**0.8092**
Interaction	0.2774	0.4506	0.7128
Improvement	8.36%	5.99%	13.52%

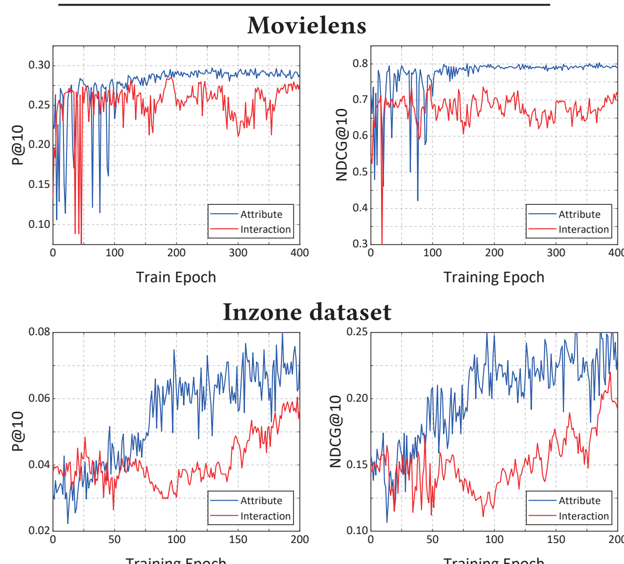

Figure 4: Learning curves of using different user represenations.

4.5 RQ3: Effect of Conditional Item and False User Pairs

Compared to the existing GAN-based recommendation methods, we present a new category of pair $(\mathbf{u}^-, I^c)$ to train our model. We construct several experiments to verify the impact of newly introduced pairs on the performance of our model. We regard the framework in which the discriminative model has two categories of training samples (*i.e.*, $(\mathbf{u}^+, I^c)$ and $(\mathbf{u}^c, I^c)$) as the baseline. This

Table 4: The performance w and w/o negative samples.

Inzone dataset	P@10	M@10	NDCG@10
W Negative Samples	**0.0781**	**0.1684**	**0.2541**
W/O Negative Samples	0.0614	0.1538	0.2150
Improvement	27.19%	9.49%	18.18%
Movielens	**P@10**	**M@10**	**NDCG@10**
W Negative Samples	**0.3006**	**0.4776**	**0.8092**
W/O Negative Samples	0.2717	0.4591	0.7673
Improvement	10.64%	4.03%	5.46%

baseline is compared with our LARA framework in which the discriminative model has three categories of training samples (*i.e.*, $(\mathbf{u}^+, I^c)$, $(\mathbf{u}^c, I^c)$, and $(\mathbf{u}^-, I^c)$). The detailed experimental settings are as follows: In the Inzone dataset, users provide implicit feedback by purchasing items. Each item contains the category and the brand attributes. Given a conditional item, we regard the users who never purchased the same category or the same brand items as the false users of the conditional item. For the Movielens dataset, we regard the exact users who rated low value (*i.e.*, lower than three) to the item as negative samples. We randomly select the false users for each conditional item to compose the training samples of the discriminative model.

The experimental results are shown in Table 4. It can be seen that our model achieves a significant improvement of performance on the two datasets after the negative samples are added in the training process of the discriminative model. The reason is that with the addition of new pairs, the discrminator has the ability of recognizing the true user from not only the generated user but also the false one when given a conditional item. Consequently, to fake the discriminator, the generator is supposed to make the generated user not only approximate to the true user, but also away from the false one.

Furthermore, we analyze learning trends of our proposed method and baselines on these two datasets. The results are shown in Figure 5. We observe that the learning curves start from different metric values and the curve, which represent the method with negative samples, consistently over the other. With the addition of new pairs, the distribution of train data becomes different from that of original data. Therefore, it is reasonable that learning curves start from different metric values. Also, because the addition of new pairs, the generator are pushed to make the generated user not only approximate to the true user, but also away from the false one. That is why our model with negative samples has a significant improvement of performance. Above all, we come to the conclusion that the negative samples consisting of conditional items and false users are absolutely necessary in the adversarial training of our model.

5 CONCLUSION

In this paper, we have proposed a novel network LARA to solve the item cold-start problem in recommender systems. The generative model is used to build the attribute-feature mapping. Specifically, we have assigned a generator to each attribute of the conditional item, the generator can generate a profile of the user who is interested in the conditional item according to the corresponding attribute. These user profiles are incorporated to construct the final user representation by a network. By this means, the generative model accomplishes the mapping between item attributes and user representation. The discriminative model attempts to distinguish three kinds of samples: the user vectors generated by the generative model, the true and the false ones from the ground truth according to the conditional items. The discriminative model guides the generative model to generate users well-matched with the conditional item. Extensive experiments on two datasets verify the effectiveness of our model. It is worth mentioning that we have constructed a new dataset, which is based on the transaction records of customers in a real-world retail supermarket. The dataset can be adapted to the experiments concerned with various problems in recommender systems.

As to future work, we will continue our research following two aspects: 1) As some state-of-the-art GAN models (*e.g.*, WGAN [4]) have made great achievements in acpects of the optimization stability and the generating effect. Therefore, we plan to apply their techniques to our model to improve the model stability and the recommendation performance. 2) We also plan to solve the user cold-start problem by using our framework, *i.e.*, generate item feature vectors from the profiles of conditional users. We hence require to find a method to represent the item using users' attributes.

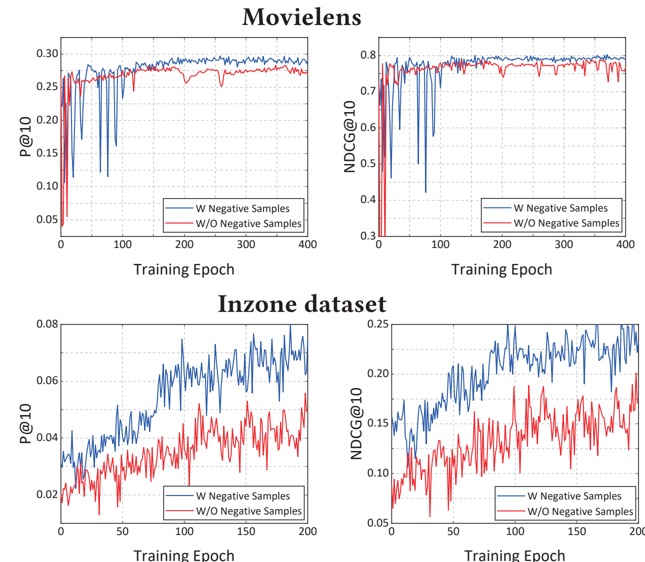

Figure 5: Learning curves w and w/o negative samples on two datasets.

6 ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China, No.: 61772310, No.: 61702300, No.: 61702302, No.: 61802231, No.: 61902219, and No.: U1836216; the Project of Thousand Youth Talents 2016; the Shandong Provincial Natural Science and Foundation, No.: ZR2019JQ23, No.: ZR2019QF001; the Future Talents Research Funds of Shandong University, No.: 2018WLJH63; the Tencent AI Lab Rhino-Bird Focused Research Program, No.: JR201932; the Fundamental Research Funds of Shandong University.

REFERENCES

[1] David Adedayo Adeniyi, Zhaoqiang Wei, and Y Yongquan. 2016. Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computing and Informatics* 12, 1 (2016), 90–108.

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering* 6 (2005), 734–749.

[3] Vinti Agarwal and KK Bharadwaj. 2013. A Collaborative Filtering Framework for Friends Recommendation in Social Networks Based on Interaction Intensity and Adaptive User Similarity. *Social Network Analysis and Mining* 3, 3 (2013), 359–379.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*. PMLR, 214–223.

[5] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. CFGAN: A Generic Collaborative Filtering Framework Based on Generative Adversarial Networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 137–146.

[6] Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhiming Ma, and Hang Li. 2009. Ranking Measures and Loss Functions in Learning to Rank. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 315–323.

[7] Emily L Denton, Soumith Chintala, Rob Fergus, et al. 2015. Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks. In *Advances in Neural Information Processing Systems*. 1486–1494.

[8] Ming Ding, Jie Tang, and Jie Zhang. 2018. Semi-supervised Learning on Graphs with Generative Adversarial Nets. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 913–922.

[9] Hao-Wen Dong and Yi-Hsuan Yang. 2018. Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR, 190–196.

[10] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. 2018. MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In *Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI, 34–41.

[11] Travis Ebesu and Yi Fang. 2017. Neural Semantic Personalized Ranking for Item Cold-start Recommendation. *Information Retrieval Journal* 20, 2 (2017), 109–131.

[12] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In *The 10th IEEE International Conference on Data Mining*. IEEE, 176–185.

[13] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR.org, 249–256.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*. MIT Press, 2672–2680.

[15] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long Text Generation via Adversarial Training with Leaked Information. In *Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI, 5141–5148.

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

[17] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 355–364.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*. ACM, 173–182.

[19] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[20] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the ACM on Conference on Information and Knowledge Management*. ACM, 1419–1428.

[21] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing Micro-videos via A Temporal Graph-guided Recommendation System. In *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, 1464–1472.

[22] Yifan Liu, Zengchang Qin, Tao Wan, and Zhenbo Luo. 2018. Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Wasserstein Generative Adversarial Networks. *Neurocomputing* 311 (2018), 78–87.

[23] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial nets. *arXiv preprint arXiv:1411.1784* (2014), 1–7.

[24] Seung-Taek Park and Wei Chu. 2009. Pairwise Preference Regression for Cold-start Recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*. ACM, 21–28.

[25] Dimitrios Rafailidis and Fabio Crestani. 2016. Joint Collaborative Ranking with Social Relationships in Top-N Recommendation. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. ACM, 1393–1402.

[26] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms. *Www* 1 (2001), 285–295.

[27] Martin Saveski and Amin Mantrach. 2014. Item Cold-start Recommendations: Learning Local Collective Embeddings. In *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, 89–96.

[28] J Ben Schafer, Joseph A Konstan, and John Riedl. 2001. E-commerce recommendation applications. *Data mining and knowledge discovery* 5, 1-2 (2001), 115–153.

[29] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and Metrics for Cold-start Recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 253–260.

[30] Michael Shekasta, Gilad Katz, Asnat Greenstein-Messica, Lior Rokach, and Bracha Shapira. 2019. New Item Consumption Prediction Using Deep Learning. *arXiv preprint arXiv:1905.01686* (2019), 1–9.

[31] Thanh Tran, Kyumin Lee, Yiming Liao, and Dongwon Lee. 2018. Regularizing Matrix Factorization with User and Item Embeddings for Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 687–696.

[32] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. Graphgan: Graph Representation Learning with Generative Adversarial Nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI, 2508–2515.

[33] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 515–524.

[34] Ke Wang and Xiaojun Wan. 2018. SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. ijcai.org, 4446–4452.

[35] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. *arXiv preprint arXiv:1905.07854* (2019).

[36] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 1543–1552.

[37] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item Silk Road: Recommending Items from Information Domains to Social Users. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 185–194.

[38] Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2017. Collaborative Filtering and Deep Learning Based Recommendation System for Cold Start Items. *Expert Systems with Applications* 69 (2017), 29–39.

[39] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*. 3203–3209.

[40] Liuchun Yuan, Congcong Ruan, Haifeng Hu, and Dihu Chen. 2019. Image Inpainting Based on Patch-GANs. *Institute of Electrical and Electronics Engineers Access* 7 (2019), 46411–46421.

[41] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial Feature Matching for Text Generation. In *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 4006–4015.

[42] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. 2011. Functional Matrix Factorizations for Cold-start Recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 315–324.