國立臺灣大學管理學院資訊管理研究所

碩士論文

Department of Information Management
College of Management

Master Thesis

以領域轉移學習來解決新進使用者推薦問題

A Domain Transfer Learning For New User Cold-Start Recommendation Problem

賴柏霖

Po-Lin Lai

指導教授：陳建錦 博士

Advisor: Chien-Chin Chen, Ph.D.

中華民國 110 年 7 月

July 2021

doi:10.6342/NTU202101275

國立臺灣大學碩士學位論文
口試委員會審定書

以領域轉移學習來解決新進使用者推薦問題

A Domain Transfer Learning For New User Cold-Start Recommendation Problem

本論文係 賴柏霖 君（學號 R08725022）在國立臺灣大學資訊管理學系、所完成之碩士學位論文，於民國 110 年 7 月 5 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

陳建錦

陳孟彰

所　　長：　陳建錦

doi:10.6342/NTU202101275

謝辭

在這邊首先要感謝我的爸媽在我念書的時期一直給予我幫助，讓我可以無後顧之憂的學習，沒有你們的幫助我沒辦法像現在這樣心無旁騖的做研究、寫論文。也感謝憨古、鼎元等系上好友可以一起打球耍寶，讓研究生的生活過得有趣。謝謝 LAB 好戰友良瑋、傑尼斯兩年來讓實驗室充滿快活的氣息，不論是互相討論模型，或是面試刷題等等，都讓我獲益良多，進步許多。在這邊尤其要特別感謝芝妘(傑尼斯)的幫助，除了一直以來互相討論各種模型包括 related work 的實作，更重要的是在 paper 中有許多論述的架構語法都給了我很大的幫忙。最後特別感謝建錦老師的指點以及包容，每週的 meeting 總能確保事情都在軌道以及進度上，老師的指點以及提醒也讓整篇 paper 變得更加完整。從開始研究以來，受到了許多人的幫助，也度過了許多開心的時光，實在非常感謝。

doi:10.6342/NTU202101275

ABSTRACT

Mitigating the new user cold-start problem has been critical in the recommendation system for online service providers. Several methods in cross-domain transfer learning use additional information from other domains to improve the recommendations for cold-start users. However, these studies only focus on transferring information in two separate domains. In this paper, we present a VAE-GAN-based model with the idea of transfer learning to resolve the cold-start problem. The main idea of the proposed model is to exploit experienced users as the source domain and use the knowledge to address the new user cold-start problem as the target domain. In addition, we design a rejuvenation function to restore the user to cold-start states and form a proper representation for specific users by leveraging side information. With extensive experiments on the real-world dataset, the results show that our proposed method significantly improves recommendation performance for cold-start new users compare with the state-of-the-art recommendation methods.

doi:10.6342/NTU202101275

摘要

解決新進使用者冷啟動問題在網路平台的推薦系統中是一個很重要的議題，許多在跨領域學習的方法中都會利用額外領域的資訊來解決目標領域上資料不足的問題。然而，這些研究只針對在不同的領域上互相轉換。在這篇論文中，我們用了一個以生成對抗式網路為基底的領域轉移學習模型，來解決新進使用者冷啟動問題。我們將該使用者剛進入平台時的冷啟動狀態，以及之後擁有豐富經驗的狀態視為兩個不同的領域，希望能透過利用使用者額外的資訊例如性別、年齡、職業等其他特徵，成功的將冷啟動狀態的使用者轉移成有豐富經驗的狀態，再以此進行推薦。透過實驗證明，我們的方法成功的超過許多現有知名的推薦系統算法，成為目前最好的冷啟動推薦方法之一。

doi:10.6342/NTU202101275

TABLE OF CONTENTS

doi:10.6342/NTU202101275

LIST OF FIGURES

doi:10.6342/NTU202101275

Chapter 1. INTRODUCTION

With data on the Internet growing exponentially, a variety of internet services such as e-commerce, streaming platforms, and social networking platforms are rapidly growing as well. This has led to the need for companies to develop their recommendation systems to mitigate the difficulties of information overload for their customers. As personalized recommendations can not only improve user experience but also generate customer loyalty, it has the potential to become a significant source of revenue and is thus an area of research with great practical importance [7, 12, 17].

Currently, most recommendation systems provide recommendations based on collecting user feedback. The most well-known example of this is the method of collaborative filtering (CF), which hopes to train representative features based on collecting user-item feedback. For example, if one user A likes items *b, c, d,* and another user B likes items *b* and *c,* then collaborative filtering uses this feedback to train features that map these user-item interactions into similar feature spaces. As such, the system might recommend item *d* to user B in accordance with the preferences of user A. Traditionally, the most widely used CF-based method is Matrix Factorization [19]. This method takes the matrix of user-item ratings, splits this matrix into two separate matrices representing user-feature and item-feature interactions separately, then attempts to train both. As such, the

doi:10.6342/NTU202101275

predicted ratings of user i for item j is the inner product of row i in the user-feature matrix and row j in the item-feature matrix. By using the original user-item matrix, we can sample many user-item pairs to train the user-feature and item-feature matrices [34]. In recent years, with the proliferation of deep learning and its nonlinear transformative power, the quality of recommendation systems has increased significantly, overtaking the traditional linear feature-generating methods using the MF approach [13, 53, 56]. With the emergence of Generative Adversarial Networks(GAN) [10], an advanced deep learning approach, its remarkable success in video generation has prompted a few studies to apply it in recommendation systems as well. The concept of GAN is to play a min-max game between two neural networks: generator and discriminator. The goal of the generator is to fool the discriminator by generating plausible samples approaching real data [1, 10]. The very first of these examples is IRGAN [48], and its success has led to more and more people attempting to use GAN in the field of recommendation systems. We detail those main GAN-based recommendation methods in the related works section. In connection with the recommendation system, a typical GAN-based model uses its ability to generate data, performing data-augmentation using the user-item matrix and attempting to imitate real data for the missing user-item interactions. An example of this is RAGAN [2], which uses GAN to generate an augmented

doi:10.6342/NTU202101275

user-item matrix, which it then inputs to the matrix factorization method to later make item recommendations.

Nevertheless, despite the continuous improvements using MF and deep learning approaches, recommendation systems still face the new user cold-start challenge [26]. As mentioned above, with the rapid development of our Internet technology, the number of available items is also increasing exponentially. For a single user, the items that she has previously interacted with or used only represent an extremely small proportion of the total items available. In general, when facing a new user, the recommendation suffers from the circumstances of little to no user-item interaction data, which may lead to unsatisfied recommendations for the new user. The cold-start phenomenon prevents recommendation methods from learning the preferences of new users, which further diminishes lifetime value for users. Thus, how to deal with the mass number of missing interaction feedback of new users is a significant challenge that all recommendation systems face.

Previous collaborative filtering–based recommendation systems are unable to make accurate recommendations for cold-start users due to data sparsity from new users. Most researchers attempt to use various side information to solve this problem, such as leveraging user demographic information, item content (e.g., text description, audio, or image), and exploiting information from social network platforms (e.g., Facebook,

doi:10.6342/NTU202101275

Twitter) [5, 24, 40, 46, 55]. For example, Yin et al. [55] provide recommendations to a user based on the purchasing information of their family and friends. Besides, the idea of information fusion also motivates the use of transfer learning from different domains. Several methods in cross-domain transfer learning for recommendation have been proposed to tackle the data sparsity problem by including more information from other domains to improve the recommendations for cold-start users [20, 21, 29, 31]. The core task of cross-domain transfer learning recommendation is user preference mapping between the two or multiple relevant domains. For instance, if a user likes a certain topic of books, we can recommend movies that share a similar topic with the movie, further forming a cycle for recommendation in both domains [20]. With the revival of deep learning techniques, many deep learning-based models are proposed to enhance knowledge transfer. Li and Tuzhilin [21] propose a DDTCDR framework to transfer a latent representation of user preference and item features between source and target domain for capturing complex relations.

These methods have achieved success in addressing the cold-start problem and prove that side information is useful to discover user preference for cold-start users. However, previous studies only focus on transferring information in two separate domains, without considering exploiting experienced users as the *source* and use the knowledge to address the new

doi:10.6342/NTU202101275

user cold-start problem as the *target*. Thus, we propose a VAE-GAN-based recommendation system based on the aforementioned idea to tackle the cold start problem in particular. Our model is extended from BicycleGAN [59]. In the proposed model, a user is represented in two respective ways: the *user preference embedding* and the *user style embedding*, the former was extracted from user rating history and the latter from side information. The proposed model mainly consists of two training parts: 1) cVAE-GAN extract the information from user rating history to obtain user preference embeddings and then train a GAN model in which the generative network mimics the rating distributions of warm users given their cold-start states and user preference embeddings, and the discriminative network acts as a detector to distinguish the generated ratings from the real ratings. 2) cLR-GAN attempts to make the generator utilize user style embeddings to produce plausible but specific warm states because the generator is trained without seeing ground truth input pairs (i.e., *user style embeddings*-cold-start states pairs). The learned generative network functions as a recommender to suggest items useful to new users. A user style generation was designed to form a proper representation for specific users by leveraging user side information and the generation is incorporated into the GAN model for effective learning.

doi:10.6342/NTU202101275

Chapter 2. RELATED WORKS

In this section, we aim at reviewing approaches that are closely related to our research. We first review the Model-based collaborative filtering in existing methods and discuss some existing GAN methods and cross-domain transfer learning methods in the recommendation system. Finally, we will detail several methods that have been proposed to address the cold-start problem.

2.1 Model-based Collaborative filtering Recommendation

We can split Recommendation systems into three main categories: item-based, user-based, and model-based. Item-based methods calculate the correlation between different items and recommend users similar items based on what they bought previously [6, 38], whereas user-based methods calculate user-user similarity to provide new recommendations [57]. Model-based methods take the user-item interactions into account and use a model to find representative features of these interactions. It aims to capture the relationships between users and items with learned embedding vectors from historical information to satisfy the user-item interaction data. The similarity between these embedding vectors can then be calculated and be used to provide recommendations. Matrix Factorization (MF) [19] is the most common in the model-based methods. The main idea behind the MF

doi:10.6342/NTU202101275

is mapping users and items into low-dimensional feature space respectively, and then conduct the dot product of them to reconstruct the user-item interaction matrix from well-trained feature space. Finally, it can successfully predict item preferences for users. Korean [19] maps user and item ID as vectors in the latent space and conducts inner product of user latent and item latent to predict user-item interaction (i.e., rating). Paterek [33] employed a gradient descent algorithm to optimize the mean square error between true and predicted ratings for computing the latent factor. With the success of the deep learning technique, various methods have adopted neural networks to achieve leading accuracy without the hard work of human feature engineering. Deep neural network (DNN) has shown its ability to extract features and modeling additional information [23, 47]. While MF utilizes linear transformation such as dot product to predict missing feedback, deep CF models can model a more complex representation of latent factors via non-linear transformation. AutoRec [39] and CDAE [51] are representative examples that apply deep learning by integrating Autoencoder(AE) and Denoising Autoencoder (DAE) [45] to reconstruct initial feedback in the output layer. Each user is represented as a vector that contains the user preference (e.g., rating score) over items, where the entry value could be binary for implicit feedback or numerical for explicit feedback. The design is to pass preference vectors of users

doi:10.6342/NTU202101275

(e.g., rating vector) through an encoder to construct robust embeddings. Then, feed the embeddings in the hidden layer into the decoder to reconstruct the true rating vector. In addition, to enhance the power of reconstruction, CDAE corrupts the input rating vector by randomly dropping out units at a ratio. Furthermore, model-based CF methods enable the system to deal with a deluge of data efficiently. They perform well when lots of preference information is available but start to degenerate in highly sparse situations [43]

2.2 GAN-based Recommendation

Recently, GAN has attracted attention in recommendation systems to enhance item recommendation performances. IRGAN [49] is the first GAN-based method that unifies the idea of generative retrieval model and discriminative models in information retrieval. The generative retrieval model act as a generator to generate(select) the relevant item while the discriminative models act as a discriminator to tell these items are relevant to the user preference or not. By repeating this training procedure until converge, the generative model will learn the relevance distribution over items and recommend the most relevant item for users. CFGAN [4] was the first vector-wise GAN method that utilizes the generator for effective item recommendations. It collects the purchase records made by users and represents each user as an item purchase vector that indicates whether the

doi:10.6342/NTU202101275

user has bought an item or not. The binary vectors are fed into the generator that learns to deceive the discriminator by constructing correct purchase vectors. Note that CFGAN adopts a negative sampling technique to help the generator distinguish items the users do not like for enriching vectors with non-purchase information. Last, the purchase vector of a target user is applied to the learned generator. Items that the user might further purchase can be suggested by investigating the reconstructed purchase vector. Chae et al. [3] designed RAGAN, a GAN-based model, which utilizes the observed item ratings made by users to train a generator for augmenting the user-item rating matrix. Besides the GAN part, it also considers the negative sampling to make the generator more robust. Finally, the augmented user-item rating matrix containing the observed ratings and the ratings generated by the learned generator is used to train collaborative filtering models.

2.3 Cross-domain transfer learning recommendation

Transfer learning [32] has been a widely studied technique in cross-domain recommendation. It aims to transfer the user preference from a source domain that is denser to a target domain or specific sources such as side information (e.g., user/item features, and social networks) with an implicit assumption that information overlaps between users across different domains. So when having insufficient data in the domain of

doi:10.6342/NTU202101275

interest, we can exploit data from mature domains which already have enough data. For instance, if a user provides positive feedback on a certain movie, the model can recommend games that share the same topic with the movie. Singh and Gordon [41] proposed a unified view of matrix factorization by providing collective matrix factorization with side information to enhance the prediction. An example of this schema is to utilize three entities: user, movie, genres. A user-movie matrix containing observed movie ratings from users can be employed to build user-feature latent and item-feature latent. A movie-genres matrix indicates which genres a movie belongs to and can be factorized to item-feature latent and genre-type latent. Since the item-feature latent are related to both matrices, the paper shares its weight to enhance the approximation of matrix factorization. After simultaneously training two factorization matrices, we can predict preference for users of certain items by taking the dot product between trained user-feature latent and item-feature latent. Li and Tuzhilin [22] present a Deep Dual Transfer Cross-Domain Recommendation (DDTCDR) framework to enable the bidirectional transfer of user preference. This method collects user and item features in two different domains and then uses two autoencoders to transform these features into embeddings. The embeddings then are fed into two separate multi-layer perceptrons to obtain within-domain user preference. For transferring user

doi:10.6342/NTU202101275

preference from one domain to the other, they use a trainable latent orthogonal mapping. With trained autoencoders and mapping matrix, DDTCDR can obtain predicted ratings for items by taking a linear combination of cross-domain and within-domain estimated user preference. Domain-to-Domain Translation Model for Recommender System (D2D-TM)[30] is the first VAE-GAN based multi-domain recommendation model extended from which extends the Unsupervised image-to-image Translation Network(UNIT) [27] in the Computer Vision domain. The proposed D2D-TM can capture both similarities and differences among features of domains by only leveraging implicit feedback (e.g., click vector) as model input. The click vectors from two domains are fed into the model and are mapped to embeddings in shared latent space by two autoencoders. Then, use two respective generators to construct the embeddings from the other domain back to click vectors. Finally, two separate discriminators are employed to clarify whether the reconstructed vectors are real. Thus, D2D-TM can transfer the knowledge in multi-domains simultaneously by building the shared latent space between domains.

2.4 Cold-Start Recommendation

As traditional CF models are not able to make proper recommendations for new users, several methods have been developed to

doi:10.6342/NTU202101275

address the cold-start new user problem. Xu et al [52] presented RAPARE, a two-staged matrix factorization that only utilizing explicit feedback from users. It first trains a matrix factorization by ratings made by warm users, then in the second stage calibrates (updates) the latent vectors of cold-start users by comparing the item ratings of the warm users. During the calibration, the item's latent vectors are fixed.

Most studies examine side information to estimate the preferences of cold-start users [23, 47], such as user content data, item features (e.g., text description, audio, or image). Volkovs et al. [47] designed DropoutNet that leverages two individual deep neural networks (DNNs) to learn respective embeddings from user rating and user demographic data. The two respective embeddings are concatenated and then are fed into another network to obtain the embeddings of the user. During the model training, the inner product of the user and item embeddings is conducted to estimate the rating the user has made on an item. A dropout mechanism is employed to remove a portion of item ratings to simulate the cold-start situation of the user. With content data from users, the model can recommend items for real cold-start users having little rating information. Jointed Training Capsule Network (JTCN) [23] was proposed to resolve the complete cold-start recommendation problem. A user is completely cold-start if the user has no rating (or purchase) data. The idea of JTCN is similar to the model

doi:10.6342/NTU202101275

of Volkovs et al., which generates item rating embeddings and user content embedding. Nevertheless, JTCN incorporates another network that mimics (reproduces) the rating embedding by using the user content data only. A mimic loss is defined between the rating embeddings of the two networks; hence, the later network is guided to learn the rating preference of a complete cold-start user. Last, the predicted rating preference concatenated with the content embedding is used to predict item ratings for a complete cold-start user. Conversely, Sedhain et al. [19] exploited social networks to recommend items to cold-start users. The method assumed that friends generally influence user's decisions so that items preferred by friends on social networks are likely to be favored by a cold-start user.

To sum up, our proposed model differs from most existing GAN-based recommendation methods as well as transfer learning-based works applied to transfer information in two separate domains [8, 31]. Moreover, we focus on extracting preference from warm users as the source and using the knowledge to address new user cold-start problems as the target. We will illustrate the details of our method in the next section

doi:10.6342/NTU202101275

Chapter 3. Proposed method

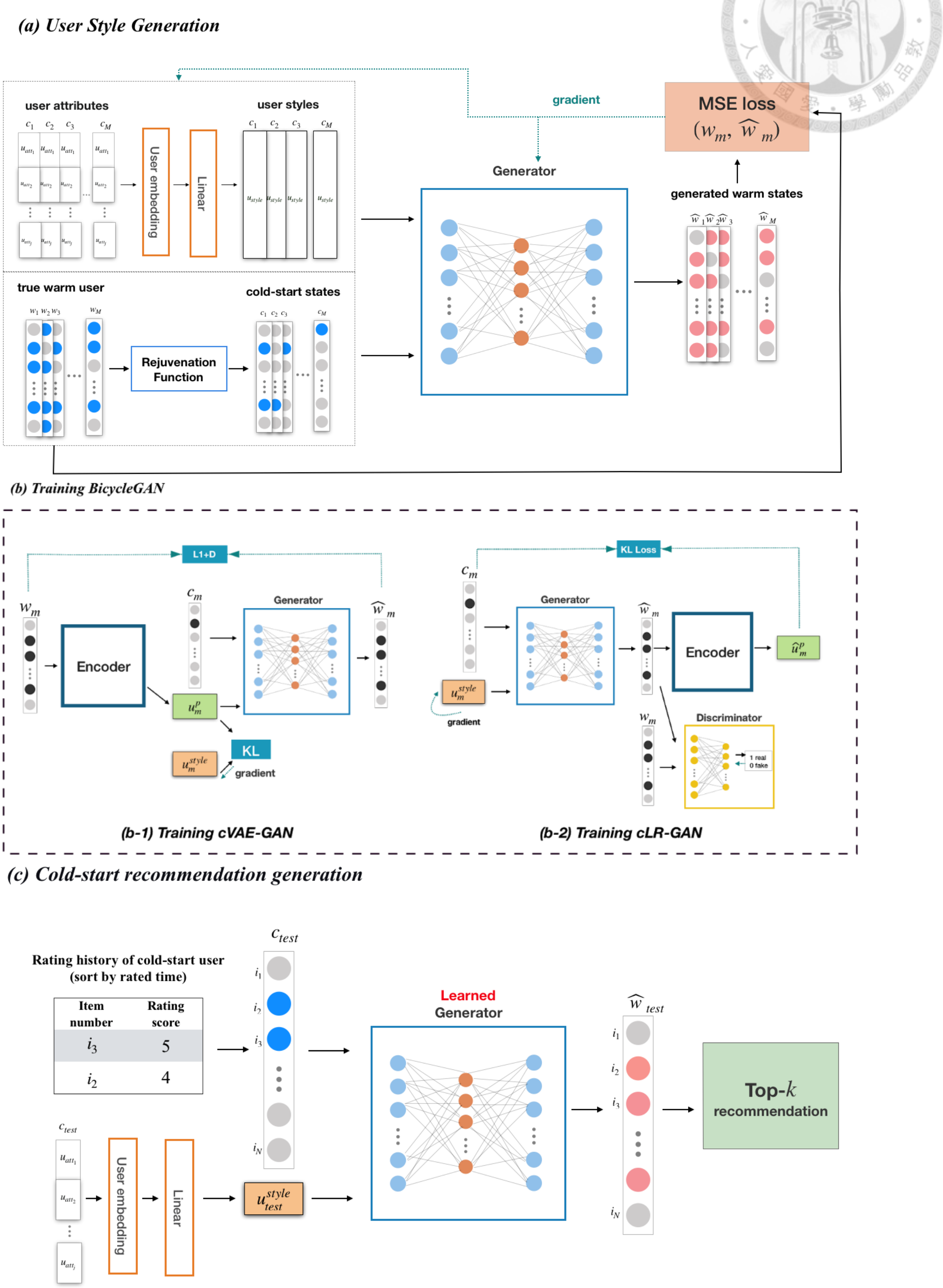

Fig 1. Overview of the proposed model.

doi:10.6342/NTU202101275

Fig 1 shows the system architecture of the proposed model, which consists of three parts: the user style generation, the BicycleGAN training (cVAE-GAN and cLR-GAN), and cold-start recommendation. As these works [25, 52], we consider users as cold-start users if they have few item ratings, and warm users if they have many item ratings. Warm users were cold-start when they just entered recommendation systems, and then they got experienced as their item ratings accumulated. To restore warm users back to their cold-start state, we design a rejuvenation function that drops some item ratings of warm users by taking item popularity and rated timestamp. The main idea is to train a GAN model with constraints in both directions (cVAE-GAN and cLR-GAN) for generating more plausible and diverse warm states for cold-start users. As a first step, we need to obtain a robust representation of user style by using user attributes, which will be used during cold-start recommendation generation. We detail in section 3.1. Next, in the cVAE-GAN module, the rating vectors of warm users are used to obtain user preference embeddings, giving the generator user information of desired output. The user preference embeddings along with the rejuvenated cold-start states are fed into a GAN model to learn a generator G and discriminator D; specifically, the network G strives to produce plausible warm states given the cold-start states and user preference information while the discriminative network discriminates the

doi:10.6342/NTU202101275

real warm states from the plausible states. At the same time, to ensure representation of user style can be used during cold-start recommendation, we use KL-divergence to regularize latent distribution to get close with user preference. However, since cVAE-GAN has different inputs as cold-start recommendation generation, we include cLR-GAN into BicycleGAN training to address this problem. The objective of the cLR-GAN module is to provide the generator a clear picture of cold-start recommendation where inputs are user style embedding and rating vector of cold-start users. After the generator is learned, it is used by the cold-start recommendation generation to suggest items relevant to the preferences of a new user.

3.1 User Style Generation

It's a challenge to recommend items that are both diverse and satisfying to users in recommendation systems, which is a similar challenge in computer vision to produce diverse and realistic images. A common approach in computer vision is to learn a low-dimensional latent code, which encapsulates the aspects of the possible output image which are not present in the input image[16, 44]. For instance, a pair of gloves could map to various colors and textures, which could get compressed in this latent code. Generally, a latent code is randomly sampled from a standard normal distribution. However, it is not appropriate for users in the recommendation system. Since many studies indicate that combining side information with

doi:10.6342/NTU202101275

user preference can effectively improve the performance of new user cold-start recommendation[23, 47], we design a simple mapping function to generate representations of user style for the user based on side information. As shown in figure 1(a), we first fed the user attributes of a user into the user embedding layer that embeds the features of user attributes into low-dimensional vectors. Next, concatenate embeddings of collected user attributes and then conduct a linear transformation to obtain the representation of user style. Instead of directly incorporated into the training process of the proposed model, we first pre-train the user style generation to ensure stability in the early stage of training cVAE-GAN.

3.1.1 The Rejuvenation

In this section, we will detail our rejuvenation function, which is novel transform function restores a warm state back to the cold-start state. The main idea behind the function composed by two important factors: the time of the item be rated and the popularity of the rated item. It's straight forward to consider the time of the item be rated since each warm-state user

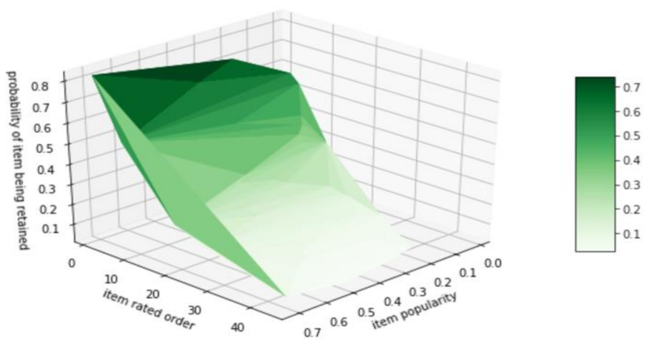

Fig 2. The distribution of our rejuvenation function.

doi:10.6342/NTU202101275

is come from their cold-state, and accumulate the rated item as time accumulated. However, several studies (e.g., [15]) show herding often influences Internet users to start using an e-service or purchasing a product. As a result, we corporate the items' popularity into our rejuvenation function. We measured item's rating frequency to decide its popularity. The more the item be rated, the more popular it is. The rejuvenation function is defined as follows:

$$p_m(t) \; = p_{min} + (p_{max} \; - \; p_{min}) * e^{\left(-\alpha * \frac{t - pop(i_t)}{count(w_m)}\right)} \qquad (1)$$

where $p_m(t)$ indicates the probability that the *t*-th item (denoted by i_t) rated by u_m will be retained in c_m. We define a function $count$ which returns the number of rated items in w_m. We define a function *pop* to measures the popularity of i_t and is defined as $pop(i_t) = freq(i_t)/max_{1 \le n \le N} freq(i_n)$ and $freq(i_n)$ counts the number of users rated i_n. The range of *pop* is [0,1] and the value is close to 1 if i_n is a popular item Symbols p_{min}, p_{max}, and α are real numbers and $0 \le p_{min} < p_{max} \le 1$. Fig 2 shows the distribution of our rejuvenation function. We can see that the earlier rated items and the more popular item, the more likely the rating will be retained in c_m, and vice versa.

3.2 BicycleGAN

doi:10.6342/NTU202101275

3.2.1 Generative Adversarial Network

Given imperfect data (e.g., images corrupted by salt-and-pepper noise), the generative network learns to re-produce perfect data (e.g., original images) from the imperfect data; the generated data are then evaluated by the discriminative network trained for detecting data that are not perfect enough. The benefits of perfect data generation also motivated us to use GAN to resolve the new user cold-start recommendation problem, which is why we take the states (i.e., the item rating vectors) of warm users as perfect data. Our proposed GAN model contains a generator G and a discriminator D, and training can be divided into two training modules: cVAE-GAN and cLR-GAN, which we detail in the subsections.

Specifically, let $U = \{u_1, \ldots, u_M\}$ be a set of warm users and $I = \{i_1, \ldots, i_N\}$

be the recommendable items of the system. We represent the warm state of a user u_m as a *rating vector* $w_m \in \mathbb{R}^N$ that records the ratings of items made by u_m [39, 42]. Next, we rejuvenate the warm state w_m to a cold-start state $c_m \in \mathbb{R}^N$ according to the designed rejuvenation function. The warm states of the users in U and the corresponding cold-start states are then fed into the GAN model. u_m^p represents user preference embedding which are extracted from w_m through an Encoder in the cVAE-GAN

doi:10.6342/NTU202101275

module, while u_m^{style} represents user style embedding transformed from u_m 's side information.

The generator *G* learns to generate plausible warm states, where input could be pair (c_m, u_m^p) or pair (c_m, u_m^{style}). On the other hand, the discriminator *D* learns to distinguish generated warm states from real warm states. After generator is learned, it has the ability to infer the preference of the cold-start user. The objective of proposed model is stated as following:

$$G^*, E^* = \arg\min_{G,E}\max_{D} \quad \mathcal{L}_{GAN}^{VAE} \ (G, D, E) + \ \mathcal{L}_1^{VAE}(G, E) + \ \mathcal{L}_{GAN} \ (G, D) + \mathcal{L}_1^{style} \ (G, E) + \quad \mathcal{L}_{KL}^{style}(E) \quad (2)$$

where

$$\mathcal{L}_{KL}(E) \ = \ \mathbb{E}_{w_m}\left[D_{KL}(E(w_m)||u_m^{style})\right] \quad (3)$$

$$\mathcal{L}_{KL}^{style} = \ \mathbb{E}_{c_m, u_{style}}[D_{KL}(u_m^{style} - E\left(G\left(c_m, u_m^p\right)\right)] \quad (4)$$

$$\mathcal{L}_{GAN}^{VAE} \ = \ \mathbb{E}_{c_m, w_m}\left[\log\left(D\left(w_m, G(c_m, u_m^p)\right)\right] + \mathbb{E}_{c_m, w_m, u_p \sim E(w_m)}\left[\log\left(1 - D\left(w_m, G(c_m, u_m^p)\right)\right] \quad (5)$$

doi:10.6342/NTU202101275

$$\mathcal{L}_{GAN}\ (G,D) = \ \mathbb{E}_{c_m,w_m,}\left[\log\left(D\left(w_m, G(c_m, u_m^{style})\right)\right)\right] +$$

$$\mathbb{E}_{c_m,w_m}\left[\log\left(1 - D\left(w_m, G(c_m, u_m^{style}\ )\right)\right)\right] \quad (6)$$

3.2.2 Conditional Variational Autoencoder GAN (cVAE-GAN)

As shown in figure 1(b-1), we present a Conditional Variational Autoencoder GAN (cVAE-GAN) to generate warm states for cold-start users. We design an encoder E to extract the user preference embedding from warm users and combine it with cold-start states as inputs of generator G. In this module, the generator G strives to produce plausible warm states given the user preference embedding u_m^p and cold-start states while the discriminator distinguishes the real from the generated. Because we cannot obtain the user preference embedding u_m^p during the cold-start scenario, the user style embedding u_m^{style} should have the ability to represent the user preferences. Therefore, we calculate the KL divergence loss between user preference embedding u_m^p that is the output of encoder *E* given warm states and user style embedding u_m^{style} based on side information.

doi:10.6342/NTU202101275

3.2.3 cLR-GAN

Since the inputs for our generator in our cold-start scenario are cold-start state c_m and user style embedding u_m^{style} obtained from side information, we also introduce a conditional latent regressor GAN (cLR-GAN) whose idea is similar to [58] to make our generator *G* more powerful and "familiar" to the user style embedding. Although we use the KL divergence in cVAE-GAN module to shorten the distance between u_m^p and u_m^{style}, the generator was still never seen about the true cold-start scenario input pair (c_m, u_m^{style}), it might hurt the inferenced recommendation. Hence, in this module, we input (c_m, u_m^{style}) to generator *G*. Again, *G* strives to produce plausible warm states users given (c_m, u_m^{style}) while the discriminator discriminates the real warm states user from generated states. Furthermore, the generated state coming from generator *G* will be an input to Encoder *E* to extract the fake preference embedding, then we calculate the KL divergence loss between fake preference embedding $\widehat{u_m^p}$, and user style embedding u_m^{style} to make them closer. To sum up, cLR-GAN module not only helps the generator G see the cold-start scenario but also shortens the distance between user style embedding and user preference embedding.

doi:10.6342/NTU202101275

3.3 Cold-Start User Recommendation

As mentioned before, we can't get the warm states in the cold-start scenario, hence, we use cold-start users' side information through a linear transformation to get user style embedding u_{test}^{style} and put it along with the rating vector of cold-start user c_{test} as input to generator *G*. After the bicycleGAN training, the generator will output the plausible warm states $\hat{w}_{test}$ which contains ratings that cold-start users may rate in the future. Finally, except the items that have been rated in their cold-start state vector, we rank all the items according to their ratings in plausible warm-state $\hat{w}_{test}$ and recommend top *k* items to the user.

doi:10.6342/NTU202101275

4. Experiments

In this section, we first introduce the evaluation dataset and metrics. Next, we conduct several experiments to validate the effectiveness and improvement of proposed model and compare our proposed model with state-of-the-art recommendation methods.

4.1 Experimental Settings

Table 1. Statistics of datasets

Datasets	#user	#item	#rating	scale	sparsity
ML1M	6,040	3,706	1,000,209	[1,2,3,4,5]	95.5%

We conduct experiments on a real-world dataset: MovieLens 1M, a well-known public dataset, contains abundant rating information. Table 1 details the statistics of the evaluation dataset. We randomly split 80% as the training set and the remaining 20% as the testing set. The users in testing set and their corresponding raring were used to examine our performance on recommending relevant item to cold-start new users. Besides, 10% of users in the training set were selected for hyper-parameter validation. For each user in the testing set, we retain the top 10 earliest item ratings to represent their cold-start state while the rest item ratings serve as ground truth. An item is relevant to the preference of a testing user if its rating score is greater than the average rating of the user. At Movielens 1M, in addition to rating

doi:10.6342/NTU202101275

information, we also utilize user attributes such as gender, age, and occupation to form the representation of a user.

To evaluate the performance of our proposed model, we adopt evaluation metrics: precision at *k*(P@*k*)[36], recall at *k* (R@*k*)[36], f1 at *k* (f1@*k*) and normalized discounted cumulative gain at *k* (nDCG@k)[18], and we set k as 5,10 for the experiments. The precision at *k* calculates the fraction of the top-*k* recommended items that are relevant to the preference of a cold-start user, and is defined as follows:

$$\mathrm{P@}k = \frac{1}{|U_{test}|}\sum_{u_{test}\in U_{test}} \frac{|relevant\ items\ in\ the\ top\text{-}k\ recommend}{k} \tag{7}$$

where U_{test} represents the set of testing users. The recall at *k* measures the fraction of relevant items suggested in the top-*k* recommendation list, and is defined as follows:

$$\mathrm{R@}k = \frac{1}{|U_{test}|}\sum_{u_{test}\in U_{test}} \frac{|relevant\ items\ in\ the\ top\text{-}k\ recommend}{number\ of\ relevant\ items\ of\ u} \tag{8}$$

The F1@k is the harmonic mean of the precision and recall scores, and is frequently used to judge the superiority of recommendation systems.

$$\mathrm{F1@}k = \frac{2 * P@k * R@k}{P@k + R@k}. \tag{9}$$

doi:10.6342/NTU202101275

The nDCG at k measures the ranking quality of the top-k recommended items, and is defined as follows:

$$\mathrm{nDCG}@k = \frac{1}{|U_{test}|} \sum_{u_{test} \in U_{test}} \frac{DCG_k}{IDCG_k}, \quad (10)$$

where DCG_k returns the discounted cumulative gain for a testing recommendation list, and is defined as follows:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)}, \quad (11)$$

where i denotes the i-th item in the top-k recommendation list and rel_i indicates the relevance score of the item. The denominator of Equation 8 $IDCG_k$ is the discounted cumulative gain of the ideal top-k item ranking. Briefly, nDCG@k not only measures the number of relevant items are suggested in k recommend item but also how early the relevant items are suggested.

Above all the metrics, the ranges of them are all between 0 and 1, with a larger value denoting a better cold-start recommendation performance.

We implement our model using a popular deep learning library PyTorch[1], and we set the RMSprop optimizer[37] for generator and discriminator. The rejuvenation parameters p_{max}, p_{min} and α are 0.01 and 0.85, respectively.

---

[1] https://pytorch.org/

doi:10.6342/NTU202101275

4.2 Examination of different type User Style

Table 2.
Comparison Results of purposed method and its variants on the MovieLens dataset

	P@5	R@5	F1@5	nDCG@5	P@10	R@10	F1@10	nDCG@10
$\text{Style}_{\text{normal}}$	0.3216	0.0300	0.0549	0.3500	0.3120	0.0589	0.0991	0.3507
$\text{Style}_{\text{user attributes}}$	0.3410	0.0340	0.0618	0.3770	0.3142	0.0605	0.1015	0.3576
$\text{Style}_{\text{pretrained}}$	0.3755	0.0400	0.0723	0.4130	0.3434	0.0700	0.1163	0.3901

As illustrated in section 3.1, we form user style embedding by leveraging side information from the user, instead of randomly sampling noise vector from a standard normal distribution. In this section, we conduct an ablation study on different types of conditional information in GAN training and also examine the effectiveness of pre-training user style embedding. Table 2 presents the cold-start recommendation performance of variants and the proposed model. In $\text{Style}_{\text{normal}}$, instead of using any side information to generate user style embedding, we randomly sample a latent vector *z* from a normal distribution. Therefore, in the bicycleGAN training, a cold-start rating vector and the latent vector z are combined and fed into the generator. In $\text{Style}_{\text{user attributes}}$, we replace the latent vector from normal distribution to linear transformed side information (user attributes). We concatenate user attributes embedding and pass through an MLP layer to get the representation. From table 2, we can notice that the result of $\text{Style}_{\text{normal}}$ is inferior to those of other variants, which indicate the value of side information. This is because the latent sampled from the normal distribution

doi:10.6342/NTU202101275

could represent any user, which discards the warm state of each specific user so that the generalized recommendation results cannot satisfy the individual need of diverse cold-start new users. The performance improvement of $\text{Style}_{\text{user attributes}}$, and $\text{Style}_{\text{pretrained}}$ validates the use of side information is more reasonable and effective. The results are consistent with previous findings in [9, 23] that side information is useful to discover user preference for cold-start users. However, we observed from the experiments that user style embeddings keep change during training and the user preference embedding extracted by Encoder E. This implicates the difficultly to have a stable KL divergence loss. To prevent this problem, we pre-train our user style embedding constructed as method 2.

All the above experiments demonstrate the pre-training and usage of side information enhance the cold-start new user recommendations effectively and items suggested by the proposed model are relevant to cold-start user preference.

doi:10.6342/NTU202101275

4.3 Comparison with Other methods

Table 3.

Comparison Results of Purposed method and other methods on the MovieLens dataset

	P@5	R@5	F1@5	nDCG@5	P@10	R@10	F1@10	nDCG@10
MF	0.0522	0.0038	0.0071	0.0633	0.0223	0.0029	0.0052	0.0286
RAPARE	0.0679	0.0040	0.0076	0.0810	0.0688	0.0099	0.0173	0.0868
CDAE	0.3265	0.0350	0.0632	0.3657	0.3057	0.0590	0.0989	0.3496
IRGAN	0.2356	0.0188	0.0349	0.2776	0.2847	0.0503	0.0855	0.3310
CFGAN	0.1543	0.0090	0.0169	0.1567	0.1832	0.0149	0.0275	0.1741
RAGAN MF	0.1427	0.0113	0.0209	0.0449	0.1306	0.0202	0.0350	0.1091
RAGAN AutoRec	0.2793	0.0241	0.0443	0.2728	0.2858	0.0499	0.0850	0.3242
DropoutNet	0.3166	0.0291	0.0532	0.3392	0.2802	0.0498	0.0846	0.3195
Our Method	0.3755	0.0400	0.0723	0.4130	0.3434	0.0700	0.1163	0.3901

In this section, nine representative recommendation methods introduced in the related work section are selected for comparisons. Including CF-based methods such as Matrix Factorization, RAPARE, and two deep learning methods CDAE and DropoutNet, we also compared with other GAN-based methods: the simple GAN, IRGAN, CFGAN, and RAGAN.

Note that RAGAN can adopt any CF-based approach to conduct item recommendations. Here, we measure RAGAN with the matrix factorization

doi:10.6342/NTU202101275

(MF)[3] and AutoRec [39], denoted as RAGAN_{MF} and $\text{RAGAN}_{\text{AutoRec}}$, respectively; these two methods were also evaluated in their respective papers mentioned above. We further evaluate the two-classic matrix factorization-based recommendation methods of MF and RAPARE-MF [32] in order to investigate the benefit of using deep-learning techniques in cold-start recommendations. Among the nine compared methods, RAPARE and DropoutNet are aimed to solve the cold-start problem. In this experiment, to ensure fair comparability, All the six compared methods were implemented as described from the relevant papers using public packages. Also, the same experimental settings were adopted to obtain the experiment results.

As illustrated result in Table 3, our proposed model reaches the best performance among all methods. We can notice that all methods adopted deep-learning techniques (i.e., our method, CDAE, DropoutNet, IRGAN, CFGAN, RAGAN_{MF} and $\text{RAGAN}_{\text{AutoRec}}$) outperformed MF and RAPARE. Likewise, $\text{RAGAN}_{\text{AutoRec}}$ achieved better performance than RAGAN_{MF}. These validate the findings in [14, 54, 56], the ability of neural network to extract better patterns compared with linear methods. With limited observed user-item interaction, especially in the cold-start scenario, the MF-based methods cannot provide satisfying recommendations. RAPARE performs slightly better than MF because it mitigates the problem by updating the embeddings of cold-start users using rating information from warm users.

doi:10.6342/NTU202101275

Compared with other GAN-based methods, according to the experimental results, IRGAN gains a pretty good result. Surprisingly, although CFGAN is an enhancement on IRGAN as stated in the respective paper, it performs slightly worse than IRGAN on the cold-start scenario. The main idea behind RAGAN is to leverage GAN's ability on data generation to augment the user-item interaction matrix and using the augmented matrix to train the MF approach. This leads to the result that $RAGAN_{MF}$ is inferior to those end-to-end deep learning methods.

To verify the GAN's effect, we compare the GAN-based model with deep-learning-based CDAE and dropoutNet. CDAE adopts a denoising autoencoder(DAE) [51] to reconstruct the rating vector of users given their cold-start states. After learned, the autoencoder can be used to recommend appropriate items for users. DropoutNet also employs dropout mechanisms to simulate cold-start scenarios by randomly dropping item ratings of the user. Hence, with the idea of using dropout mechanisms, CDAE and DropoutNet similar to our proposed model. According to the comparison results, it suggests GAN is a more effective technique than DAE and MLP. Nevertheless, out of our expectations, CDAE and DropoutNet perform better than other GAN-based methods. This may because other compared GAN-based methods are designed for non-cold-start scenarios, which makes them inferior to CDAE and DropoutNet. Note that our

doi:10.6342/NTU202101275

proposed model and DropoutNet both utilize dropout mechanisms and side information to deal with new user cold-start recommendation, both achieve good results demonstrate that dropout mechanism and side information are crucial to discover user preference for cold-start users.

The outperformance of our model not only demonstrates the success of leveraging side information to represent specific user style in cold-start scenarios but also shows that we can view the cold start problem as a domain transfer task.

doi:10.6342/NTU202101275

5 Conclusion

With the rapid growth of the e-service industry, the e-service industry has become more and more competitive, and various strategies are deployed to help service providers attract and retain new customers. Providing appropriate personalized recommendations has been verified as an effective approach. However, e-service providers usually suffer from data sparsity in the real world, which makes new user cold-start recommendations remain a challenge. In this paper, we have developed a GAN-based recommendation model to address new user cold-start recommendations. The main idea behind our model is domain transferring, we view the cold-start user and warm user as different domains. Moreover, we integrate side information from users as a supplementary to guide the transformation of the cold-start state. The experiments demonstrate that our method is robust and significantly outperforms many well-known recommendation methods in precision, recall, F1, and nDCG.

doi:10.6342/NTU202101275

6 Future Work

As the rise of graph embedding[11, 50] , more and more recommendation systems corporate the graph embedding to enhance the model performance. Since Graph embedding not only provides better quantitative understanding, but also can handle heterogeneous data, some previous method has leverage graph embedding and side information to deal with cold start problem[28, 35], and get pretty good result and interpretability. Since our preliminary experiments on user style embedding were transformed by nonlinear neural networks, we plan to investigate different types of graph embedding methods. The idea of adapting graph embedding may enhance the performance and make the results more explainable.

doi:10.6342/NTU202101275

Reference

[1] A. Antoniou, A. Storkey, H. Edwards, Data augmentation generative adversarial networks, arXiv preprint arXiv:1711.04340, (2017).

[2] D.-K. Chae, J.-S. Kang, S.-W. Kim, J. Choi, Rating Augmentation with Generative Adversarial Networks towards Accurate Collaborative Filtering, in: The World Wide Web Conference, Association for Computing Machinery, San Francisco, CA, USA, 2019, pp. 2616–2622.

[3] D.-K. Chae, J.-S. Kang, S.-W. Kim, J. Choi, Rating augmentation with generative adversarial networks towards accurate collaborative filtering, in: The World Wide Web Conference, 2019, pp. 2616-2622.

[4] D.-K. Chae, J.-S. Kang, S.-W. Kim, J.-T. Lee, Cfgan: A generic collaborative filtering framework based on generative adversarial networks, in: Proceedings of the 27th ACM international conference on information and knowledge management, 2018, pp. 137-146.

[5] C.C. Chen, Y.-H. Wan, M.-C. Chung, Y.-C. Sun, An effective recommendation method for cold start new users using trust and distrust networks, Information Sciences, 224 (2013) 19-36.

[6] M. Deshpande, G. Karypis, Item-based top-N recommendation algorithms, ACM Trans. Inf. Syst., 22 (2004) 143–177.

[7] M.J. Eppler, J. Mengis, The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines, The Information Society, 20 (2004) 325-344.

[8] I. Fernández-Tobías, I. Cantador, M. Kaminskas, F. Ricci, Cross-domain recommender systems: A survey of the state of the art, in: Spanish conference on information retrieval, sn, 2012, pp. 1-12.

doi:10.6342/NTU202101275

[9] R. Forsati, M. Mahdavi, M. Shamsfard, M. Sarwat, Matrix factorization with explicit trust and distrust side information for improved social recommendation, ACM Transactions on Information Systems (TOIS), 32 (2014) 1-38.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, in: International conference on neural information processing systems, 2014, pp. 2672–2680.

[11] P. Goyal, E. Ferrara, Graph embedding techniques, applications, and performance: A survey, Knowledge-Based Systems, 151 (2018) 78-94.

[12] S. Gupta, D. Hanssens, B. Hardie, W. Kahn, V. Kumar, N. Lin, N. Ravishanker, S. Sriram, Modeling customer lifetime value, Journal of service research, 9 (2006) 139-155.

[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: 26th international conference on world wide web, 2017, pp. 173-182.

[14] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 173-182.

[15] J.H. Huang, Y.F. Chen, Herding in online product choice, Psychology & Marketing, 23 (2006) 413-428.

[16] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125-1134.

[17] Y. Jiang, J. Shang, Y. Liu, Maximizing customer satisfaction through an online recommendation system: A novel associative classification model, Decision Support Systems, 48 (2010) 470-479.

[18] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques,

doi:10.6342/NTU202101275

ACM Transactions on Information Systems (TOIS), 20 (2002) 422-446.

[19] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer, 42 (2009) 30-37.

[20] B. Li, Q. Yang, X. Xue, Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction, in: Proceedings of the 21st international jont conference on Artifical intelligence, Morgan Kaufmann Publishers Inc., Pasadena, California, USA, 2009, pp. 2052–2057.

[21] P. Li, A. Tuzhilin, DDTCDR: Deep Dual Transfer Cross Domain Recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, Association for Computing Machinery, Houston, TX, USA, 2020, pp. 331–339.

[22] P. Li, A. Tuzhilin, DDTCDR: Deep dual transfer cross domain recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 331-339.

[23] T. Liang, C. Xia, Y. Yin, P.S. Yu, Joint Training Capsule Network for Cold Start Recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1769-1772.

[24] T. Liang, C. Xia, Y. Yin, P.S. Yu, Joint Training Capsule Network for Cold Start Recommendation, in: 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1769-1772.

[25] J. Lin, K. Sugiyama, M.-Y. Kan, T.-S. Chua, Addressing cold-start in app recommendation: latent user models constructed from twitter followers, in: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, 2013, pp. 283-292.

[26] J. Lin, K. Sugiyama, M.-Y. Kan, T.-S. Chua, Addressing cold-start in app recommendation: latent user models constructed from twitter followers, in: 36th

doi:10.6342/NTU202101275

international ACM SIGIR conference on Research and development in information retrieval, 2013, pp. 283-292.

[27] M.-Y. Liu, T. Breuel, J. Kautz, Unsupervised image-to-image translation networks, arXiv preprint arXiv:1703.00848, (2017).

[28] S. Liu, I. Ounis, C. Macdonald, Z. Meng, A Heterogeneous Graph Neural Model for Cold-Start Recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 2029-2032.

[29] N. Mirbakhsh, C.X. Ling, Improving Top-N Recommendation for Cold-Start Users via Cross-Domain Information, ACM Trans. Knowl. Discov. Data, 9 (2015) Article 33.

[30] L. Nguyen, T. Ishigaki, Domain-to-Domain Translation Model for Recommender System, arXiv preprint arXiv:1812.06229, (2018).

[31] L. Nguyen, T. Ishigaki, D2D-TM: A Cycle VAE-GAN for Multi-Domain Collaborative Filtering, in: 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 1175-1180.

[32] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering, 22 (2009) 1345-1359.

[33] A. Paterek, Improving regularized singular value decomposition for collaborative filtering, in: Proceedings of KDD cup and workshop, 2007, pp. 5-8.

[34] A. Paterek, Improving regularized singular value decomposition for collaborative filtering, in: Proceedings of KDD cup and workshop,13th ACM Int. Conf. on Knowledge Discovery and Data Mining, San Jose, CA, USA, 2007, pp. 39-42.

[35] T. Qian, Y. Liang, Q. Li, Solving cold start problem in recommendation with attribute graph neural networks, arXiv preprint arXiv:1912.12398, (2019).

[36] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook,

doi:10.6342/NTU202101275

in: Recommender systems handbook, Springer, 2011, pp. 1-35.

[37] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747, (2016).

[38] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th international conference on World Wide Web, Association for Computing Machinery, Hong Kong, Hong Kong, 2001, pp. 285–295.

[39] S. Sedhain, A.K. Menon, S. Sanner, L. Xie, Autorec: Autoencoders meet collaborative filtering, in: Proceedings of the 24th international conference on World Wide Web, 2015, pp. 111-112.

[40] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, J. Christensen, Social collaborative filtering for cold-start recommendations, in: 8th ACM Conference on Recommender systems, 2014, pp. 345-348.

[41] A.P. Singh, G.J. Gordon, Relational learning via collective matrix factorization, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 650-658.

[42] F. Strub, J. Mary, Collaborative filtering with stacked denoising autoencoders and sparse inputs, in: NIPS workshop on machine learning for eCommerce, 2015.

[43] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in artificial intelligence, 2009 (2009).

[44] Y. Taigman, A. Polyak, L. Wolf, Unsupervised cross-domain image generation, arXiv preprint arXiv:1611.02200, (2016).

[45] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096-1103.

doi:10.6342/NTU202101275

[46] M. Volkovs, G.W. Yu, T. Poutanen, DropoutNet: Addressing Cold Start in Recommender Systems, in: 31st International Conference on Neural Information Processing Systems, 2017, pp. 4964–4973.

[47] M. Volkovs, G.W. Yu, T. Poutanen, DropoutNet: Addressing Cold Start in Recommender Systems, in: NIPS, 2017, pp. 4957-4966.

[48] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, D. Zhang, 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, Association for Computing Machinery, Shinjuku, Tokyo, Japan, 2017, pp. 515–524.

[49] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, D. Zhang, 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, 2017, pp. 515-524.

[50] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Transactions on Knowledge and Data Engineering, 29 (2017) 2724-2743.

[51] Y. Wu, C. DuBois, A.X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: Proceedings of the ninth ACM international conference on web search and data mining, 2016, pp. 153-162.

[52] J. Xu, Y. Yao, H. Tong, X. Tao, J. Lu, R a P are: A generic strategy for cold-start rating prediction problem, IEEE Transactions on Knowledge and Data Engineering, 29 (2016) 1296-1309.

[53] H.-J. Xue, X. Dai, J. Zhang, S. Huang, J. Chen, Deep Matrix Factorization Models for Recommender Systems, in: 26th International Joint Conference on Artificial

doi:10.6342/NTU202101275

Intelligence, Melbourne, Australia, 2017, pp. 3203-3209.

[54] H.-J. Xue, X. Dai, J. Zhang, S. Huang, J. Chen, Deep Matrix Factorization Models for Recommender Systems, in: IJCAI, Melbourne, Australia, 2017, pp. 3203-3209.

[55] H. Yin, L. Zou, Q.V.H. Nguyen, Z. Huang, X. Zhou, Joint Event-Partner Recommendation in Event-Based Social Networks, in: 2018 IEEE 34th International Conference on Data Engineering (ICDE), 2018, pp. 929-940.

[56] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys (CSUR), 52 (2019) 1-38.

[57] Z.-D. Zhao, M.-S. Shang, User-based collaborative-filtering recommendation algorithms on hadoop, in: 2010 third international conference on knowledge discovery and data mining, IEEE, 2010, pp. 478-481.

[58] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A.A. Efros, O. Wang, E. Shechtman, Toward multimodal image-to-image translation, arXiv preprint arXiv:1711.11586, (2017).

[59] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A.A. Efros, O. Wang, E. Shechtman, Toward multimodal image-to-image translation, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Long Beach, California, USA, 2017, pp. 465–476.

doi:10.6342/NTU202101275