Proceedings of Machine Learning Research 101:1188–1203, 2019 ACML 2019

From Implicit to Explicit Feedback: A deep neural network for modeling the sequential behavior of online users

Anh Phan Tuan TUANANHLFC@GMAIL.COM
Hanoi University of Science and Technology, No. 1, Dai Co Viet road, Hanoi, Vietnam

Nhat Nguyen Trong TRONGNHAT2312@GMAIL.COM
Duong Bui Trong DUONGBUITRONG@ADMICRO.VN
VC Corporation

Linh Ngo Van* LINHNV@SOICT.HUST.EDU.VN
Hanoi University of Science and Technology, No. 1, Dai Co Viet road, Hanoi, Vietnam
Khoat Than KHOATTQ@SOICT.HUST.EDU.VN
Hanoi University of Science and Technology, No. 1, Dai Co Viet road, Hanoi, Vietnam
VinAI Research, Hanoi, Vietnam

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

We demonstrate the advantages of taking into account multiple types of behavior in recommendation systems. Intuitively, each user has to do some **implicit** actions (e.g., click) before making an **explicit** decision (e.g., purchase). Previous works showed that implicit and explicit feedback has distinct properties to make a useful recommendation. However, these works exploit implicit and explicit behavior separately and therefore ignore the semantic of interaction between users and items. In this paper, we propose a novel model namely *Implicit to Explicit (ITE)* which directly models the order of user actions. Furthermore, we present an extended version of ITE, namely *Implicit to Explicit with Side information (ITE-Si)*, which incorporates side information to enrich the representations of users and items. The experimental results show that both ITE and ITE-Si outperform existing recommendation systems and also demonstrate the effectiveness of side information in two large scale datasets.

Keywords: Recommendation systems, Implicit Feedback, Explicit Feedback, Deep Learning, Collaborative Filtering

1. Introduction

Most of the recommendation systems utilize data of user behavior to analyze the preferences of users and match them with suitable items. User behavior can be collected in various forms: like, view, click, purchase, rate, etc. Traditional collaborative filtering (CF) methods usually classify these behavior into two groups (implicit feedback (Hu et al., 2008; Koren et al., 2009) and explicit feedback (Mnih and Salakhutdinov, 2008; Zigoris and Zhang, 2006)) and manipulate them to find out the relationship between users and items. Previously, most existing works employ one type of behavior to suggest items and therefore ignore the relationship between multiple types of behavior.

* Corresponding Author.

Recently, several studies (Liu et al., 2010; Koren, 2011; Shi et al., 2017; Gao et al., 2019) found out that combining both explicit and implicit behavior improves recommendation effectiveness since these two types of data have different properties. Explicit behavior (e.g., rating, purchasing, liking) are credible (showing the level of interest) but scarce, while implicit behavior (e.g., view, click) contain a vast of data but do not show clearly the matching between users and items. However, the majority of recommendation systems can not comprehend the sequential relation from implicit to explicit behavior. Multi-task Matrix Factorization (MTMF) (Shi et al., 2017) explores the *view*, *want*, and *rate* behavior separately before composing to predict the rating score for a pair of (user, item). Neural Multi-Task Recommendation (NMTR) (Gao et al., 2019) models directly multiple types of behavior as multi-task learning, where each task learns a specific type of interaction. NTMR attempts to decompose the sequential relation by connecting all the predictive output of interaction as a cascading pattern. However, aggregating multi-behavior data via connecting predictive output (a scalar) is too simple, thus cannot capture the more complex semantic association between implicit and explicit interactions. Furthermore, the sequence of actions in real-world has a strong connection and follows the rule that user preferences influence implicit behavior, and implicit behavior affects the explicit. Therefore, the sequential nature from implicit and explicit behavior should be modeled in two consecutive phases in a single task instead of multi-tasks.

Side information is a valuable source to support in learning the semantic of latent user preferences and item features. Hoang et al. (2019); Zhang et al. (2010) made use of the category (tag) information of items to enhance the representation of each user and therefore improve predictive accuracy. Le et al. (2018); Wang and Blei (2011) proposed hybrid methods employing item description to handle the cold start problem. Lillegraven and Wolden (2010); Nadimi-Shahraki and Bahadorpour (2014) showed some popular items to new users and asked them to rate. An abundance of further researches showed that utilizing such side information not only improves performance but also helps the model avoid the cold start problem.

In this paper, we propose two novel models for modeling the sequential behavior of users, namely *Implicit to Explicit (ITE)* and *ITE with Side information (ITE-Si)*. The architectures of ITE and ITE-Si are very similar, except that ITE-Si incorporates side information encoded in the representations of users and items. In summary, the contributions of our work are as follows:

- We propose ITE which models the sequential behavior of users. Briefly, ITE is a neural network which can be decomposed as two feed-forward modules to learn implicit interactions between users and items, and the last hidden layer of the implicit module will be fed into the first layer of the explicit module. Therefore, this architecture helps discover the complex relationship between implicit and explicit behavior.

- ITE-Si can employ various side information to enrich the representations of users and items. Such extra information provides credible knowledge about users/items and thus helps avoid the cold start problem.

We conduct extensive experiments on two large scale datasets. The experiments show that our models outperform existing recommendation models even when ITE does not hold any additional data. Furthermore, the model attached with side information performs better than original ITE, which indicates that external data is a useful resource.

The rest of the paper is organized as follows. Section 2 briefly reviews some related work and background. Section 3 presents the details of ITE model and discusses several benefits. In Section 4, we conduct extensive experiments and describe the evaluation results of the proposed models. Finally, we provide the conclusion in Section 5.

2. Related work and background

2.1. Related work

ITE is a Deep Learning based model exploiting the sequence of user actions in online recommender systems. In what follows, we investigate several existing works related to our approach.

In recent years, most of the existing approaches for recommendation systems are utilizing deep neural networks to extract the latent features representing for users' interest and item properties. He et al. (2017) proposed Neural Matrix Factorization (NeuMF) which approximates each user-item interaction by combining the ideas of MultiLayer Perceptron (MLP) and Matrix Factorization (MF) model. With the same intuition, Cheng et al. (2016) presented Wide & Deep Learning model which combines the benefits of memorization and generalization via wide and deep neural networks. These Deep Learning based models gained lots of promising results over traditional collaborative filtering methods and provide a novel approach for recommendation systems.

Combining multiple types of behavior, i.e., implicit and explicit data, has been paid attention by several researches recently (Singh and Gordon, 2008; Krohn-Grimberghe et al., 2012; Zhao et al., 2015). However, these methods exploit implicit and explicit feedback as separate behavior and therefore ignore the ordinal relations between different types of behavior. On the other hand, (Liu et al., 2010; Gao et al., 2019; Shi et al., 2017) attempted to find the latent relationship between multiple behavior by modeling directly such interactions in the neural network architecture. However, the way that these models associate implicit and explicit behavior is too simple and therefore it has still open room for improving.

Finally, a large range of researches related to our work is to employ additional data besides the user-item interaction to avoid *cold start* problem. (Hoang et al., 2019; Zhang et al., 2010) utilized item category (tag) to help users find the new (or less popular) yet interesting objects. Zheng et al. (2017) proposed the Deep Cooperative Neural Networks (DeepCoNN) utilizing two separate neural networks to exploit reviews written by the user and the reviews written for the item respectively.

2.2. Background

In this section, we introduce some preliminaries serving as the base knowledge of our model.

2.2.1. Matrix Factorization

Matrix Factorization Koren et al. (2009) is one of the most popular methods in recommendation systems. This model aims to discover the latent factors representing for each user and each item. Let p_u and q_v denote the latent vector of user u and item v respectively, MF model predicts the rating of user u for item v as the inner product of p_u and q_v:

$$\hat{y}_{ij} = p_u^T q_v = \sum_{k=1}^{K} p_{uk} q_{ik} \tag{1}$$

The target of MF model is to estimate those latent vectors. Each latent vector can reveal some intuition of user preferences or item features.

2.2.2. Neural Matrix Factorization

Neural Matrix Factorization (NeuMF) (He et al., 2017) is an extended version of Matrix Factorization (Koren et al., 2009) by implementing deep neural network. Since the inner product of user preferences (p_u) and item features (q_i) can be considered as a linear function, MF may not find out complex relation between users and items. In contrast, NeuMF approximates the relation between user u and item v as a non-linear function by employing Deep Neural Networks.

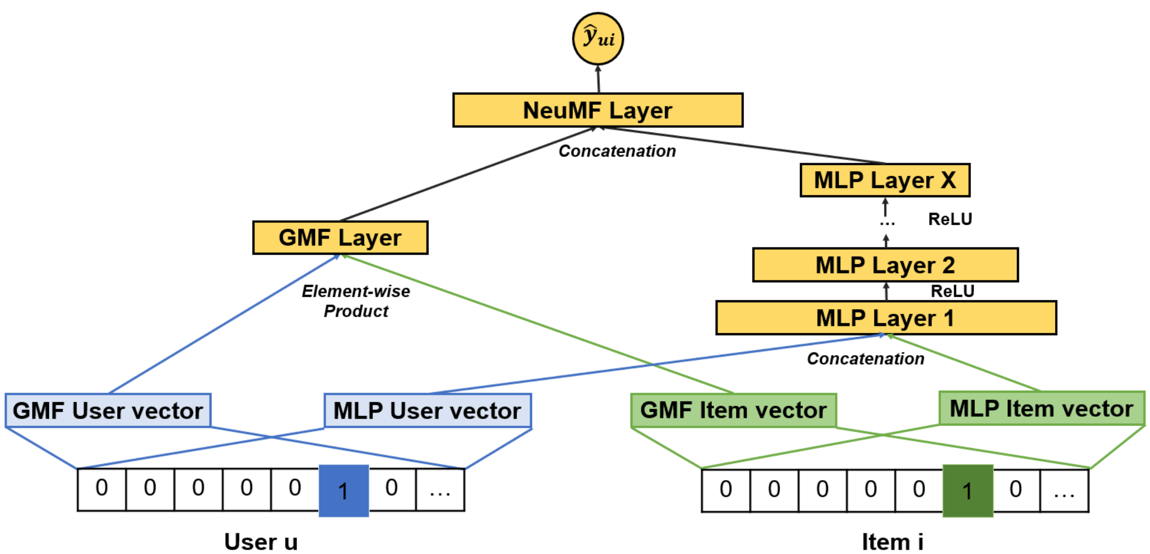

Figure 1: The architecture of NeuMF model

Fig. 1 shows the architecture of NeuMF. As can be seen from the Figure, NeuMF consists of two parts which are GMF (Generalized Matrix Factorization) and MLP (Multi-Layer Perceptron). GMF is an extended version of Matrix Factorization by implementing Neural Collaborative Filtering (NCF) framework (He et al., 2017). GMF achieved slightly better results than MF in several experiments (He et al., 2017). Furthermore, MLP can learn the relation of users and items as a non-linear function by concatenating and passing the latent vectors into several hidden layers. By letting GMF and MLP learn embedding layer separately and then concatenate the last hidden layers, NeuMF combines the strength of both models and therefore gains promise results over traditional collaborative filtering methods.

3. Proposed models

In this section, we propose the two novel models, namely Implicit to Explicit (ITE) and ITE with Side information (ITE-Si). We first present the details of ITE and ITE-Si and then discuss several advantages of our methods.

We use the following notations in this paper:

- M, N: the number of users and items respectively
- $\mathbf{X} = (x_{ui})_{M\times N}$: data of implicit behavior, where x_{ui} indicates that user u has interacted with item i
- $\mathbf{Y} = (y_{ui})_{M\times N}$: data of explicit behavior, where y_{ui} reveals that user u had an action showing interest in item i
- $\mathbf{u}, \mathbf{i}$: the representations vector of user u and item i respectively
- K: the dimension of the vector representing side information

3.1. Implicit to Explicit (ITE) model

The data used in ITE model includes implicit and explicit behavior. Both explicit and implicit behavior are in form of binary value. For general purpose, some non binary feedback can be transformed into binary scale. For example, the rating $r_{ui} \in \{0, ..., n\}$ is converted to explicit data: $y_{ui} = 1$ if $r_{ui} \geq k$ and $y_{ui} = 0$ if $r_{ui} < k$.

Furthermore, we also have to differentiate explicit behavior and implicit behavior regarding to our model. In this paper, we define that explicit behavior is the action showing that a user interests in an item such as **purchase**, **order**, **add to cart**. A user likes an item before deciding to purchase it. In contrast, implicit behavior is the action showing that a user wants to know more about an item such as **view**, **click**. Intuitively, a user might glance at lots of items to find the most suitable one. In formulation, the variable $y_{ui} \in \{0, 1\}$ represents explicit interaction between user u and item i, while $x_{ui} \in \{0, 1\}$ expresses the implicit one.

We present the architecture of **Implicit to Explicit (ITE) model** in Fig. 2. The input vector of user u and item i are denoted by $\mathbf{u} = (\alpha_1, ..., \alpha_S)$ and $\mathbf{i} = (\beta_1, ..., \beta_W)$ respectively. These representations can be modified to utilize a wide range of user and item representations depending on the available data, such as one-hot encoding (Koren et al., 2009; Gao et al., 2019), neighbor-based encoding (Rendle, 2010). In this work, we use one-hot encoding for representing the inputs.

The input vectors is then fed into a neural network which is represented for the **implicit module** of the model. We utilize the NeuMF model (He et al., 2017) for this module due to the good performance when dealing with implicit data. The last hidden layer of the **implicit module** is called **Implicit Layer**. Let $p_u^G, q_i^G, p_u^M, q_i^M$ denote the embedding vectors of user u and item i (G and M denotes for GMF and MLP Layer), ϕ^{GMF}, ϕ^{MLP} be the vectors of *GMF Layer* and *MLP Layer X* shown in Fig. 2. We will formulate the implicit module in what follows.

The hidden layer of GMF part can be expressed as follows:

$$\phi^{GMF} = p_u^G \odot q_i^G \tag{2}$$

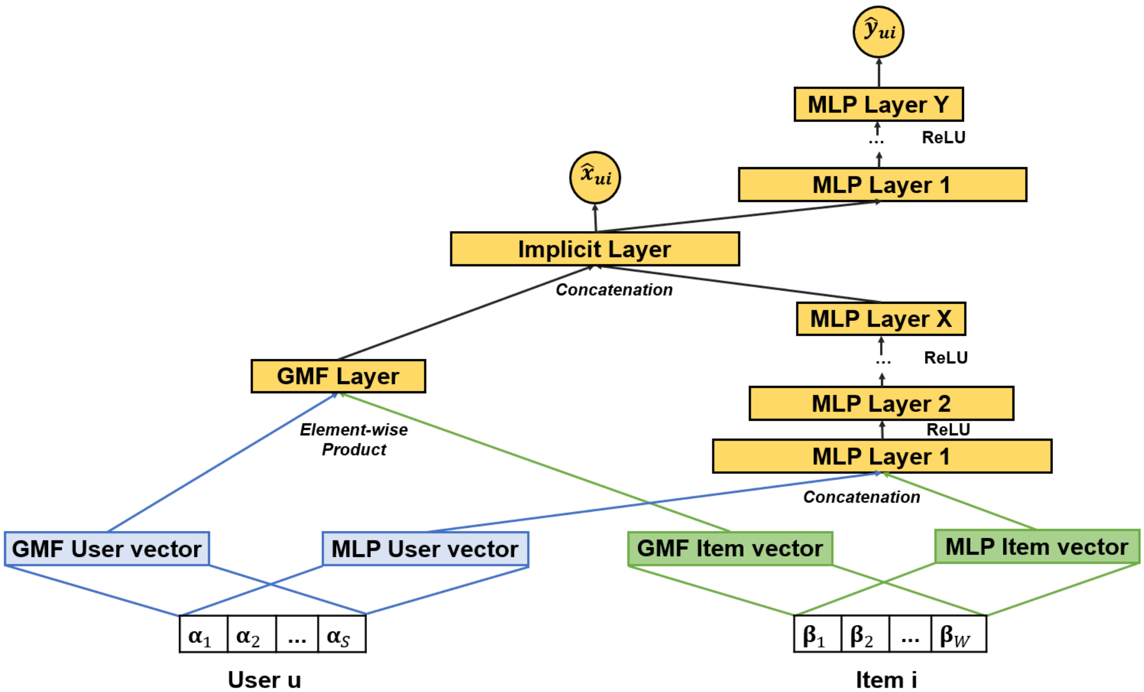

Figure 2: The architecture of ITE model

where $\odot$ denotes the pair-wise product. MLP user vector and MLP item vector are fed into a Multi-Layer Perceptron network including X hidden layers. Particularly, each layer performs the following computation:

$$\phi_{(l+1)}^{MLP} = f(W_{(l)}\phi_{(l)}^{MLP} + b_{(l)}) \tag{3}$$

where l is the layer number, f is the activation function (usually ReLU function). The GMF layer and MLP Layer X in Fig. 2 are then concatenated to get the Implicit Layer:

$$\phi^I = \begin{bmatrix} \phi^{GMF} \\ \phi_{(X)}^{MLP} \end{bmatrix} \tag{4}$$

where ϕ^I is the vector denoting *implicit layer*. From the implicit layer, we can output the likelihood that u will perform an implicit behavior with item i:

$$\hat{x}_{ui} = \sigma(\mathbf{h}_I^T \phi^I) \tag{5}$$

The sequence of behavior will be modeled by passing the Implicit Layer to a Multi-Layer Perceptron (MLP) network called **explicit module**. After several hidden layers of this MLP, we achieve the MLP Layer Y as shown in Fig. 2. Reminding that the vector of implicit layer is ϕ^I. The *MLP Layer 1* of explicit module is computed as follows:

$$\phi_1^E = f(V_{(0)}\phi^I + b_{(0)}) \tag{6}$$

Then, each hidden layer of explicit layer performs the following computation:

$$\phi_{l+1}^E = f(V_{(l)}\phi_{(l)} + b_{(l)}) \tag{7}$$

where $\mathbf{V}$ denotes the weight of hidden layer, $\mathbf{h}$ is the edge weight of output layer. The last layer of explicit module is used to compute the likelihood that u will perform a explicit behavior with item i:

$$\hat{y}_{ui} = \sigma(\mathbf{h}_E^T \phi_{(Y)}^E) \tag{8}$$

Note that the outputs of ITE are $\hat{x}_{ui}$ and $\hat{y}_{ui}$ representing the predicted outcome of implicit and explicit behavior respectively. In an interpretable way, $\hat{y}_{ui}$ denotes the probability of user u interacting explicitly with item i given that the implicit behavior happens. After learning the model, the probability that user u will interact item i is expressed by an additional variable $\hat{r}_{ui}$ which strengthens the association between $\hat{x}_{ui}$ and $\hat{y}_{ui}$:

$$\hat{r}_{ui} = \hat{x}_{ui}\hat{y}_{ui} \tag{9}$$

3.2. Implicit to Explicit model with Side information (ITE-Si)

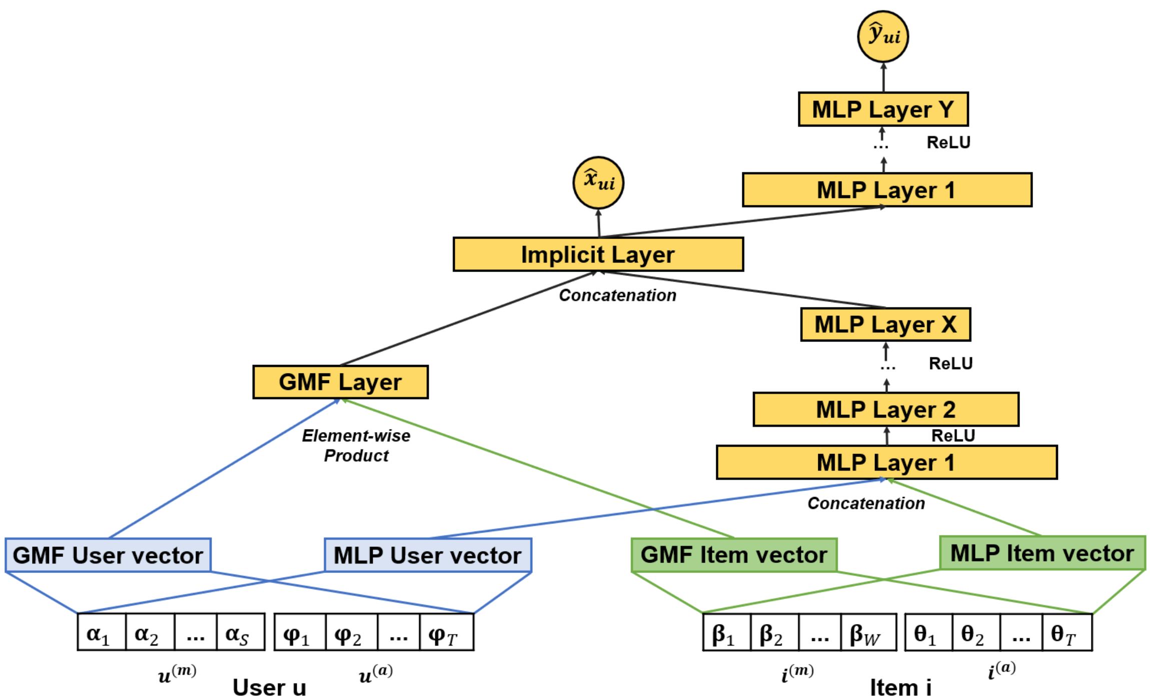

Figure 3: The architecture of ITE-Si model

We present the architecture of ITE-Si in Fig. 3. ITE-Si is very similar to ITE except that we use additionally *side information*. The data used in ITE-Si includes: implicit behavior, explicit behavior and side information. Side information often exists as item features, movie categories, music genres, etc. and can be encoded as a K-dimensional vector.

A worthy noting is that we directly encode this sort of information in the representations of users and items. In some situation, the one-hot encoding suffers from lacking of information and it is expected that side information can enrich their representations. Also, side information is expected to help the model avoid cold start as mentioned in a vast of previous research (Le et al., 2018; Wang and Blei, 2011).

Fig. 3 shows that ITE-Si differs from ITE in the bottom input layer. The input layer of ITE-Si consists of two representations vectors $\mathbf{u}$ and $\mathbf{i}$ describing user u and item i respectively, where $\mathbf{u} = concat(u^{(m)}, u^{(a)})$ and $\mathbf{i} = concat(i^{(m)}, i^{(a)})$ (*concat* means concatenating multiple vectors).

Here, $u^{(m)}, i^{(m)}$ are the same as the u, i (user, item) vector in ITE model, while $u^{(a)}, i^{(a)}$ are the vectors constructed from **item category** data and not required in ITE but can improve performance if they are available. More specifically, let T be the dimension of side information, $i^{(a)} = (\theta_1, \theta_2, ..., \theta_T)$ denotes the category information of item i. Then, $u^{(a)} = (\varphi_1, \varphi_2, ..., \varphi_T) = (\frac{\epsilon_1}{\epsilon}, \frac{\epsilon_2}{\epsilon}, ..., \frac{\epsilon_T}{\epsilon})$, ϵ_j denotes the number of items interacted by user u and belongs j^{th} category, $\epsilon = \epsilon_1 + \epsilon_2 + ... + \epsilon_T$ and $\varphi_j = \frac{\epsilon_j}{\epsilon}$.

The other components of ITE-Si are the same as ITE. In summary, we can formulate the implicit module and explicit module as follows:

$$
\begin{aligned}
\phi^{GMF} &= \mathbf{p}_u^G \odot \mathbf{q}_i^G, \\
\phi^{MLP} &= \mathbf{f}(\mathbf{W}_X^T(\mathbf{f}(...\mathbf{f}(\mathbf{W}_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{bmatrix} + \mathbf{b}_2)...)) + \mathbf{b}_X), \\
\phi^I &= \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix},
\end{aligned} \tag{10}
$$

where ϕ^I is the vector denoting *implicit layer*. The explicit module is computed as follows:

$$
\begin{aligned}
\hat{x}_{ui} &= \sigma(\mathbf{h}_I^T \phi^I), \\
\phi^E &= \mathbf{f}(\mathbf{V}_Y^T(\mathbf{f}(...\mathbf{f}(\mathbf{V}_2^T \phi^E + \mathbf{d}_2)...)) + \mathbf{d}_Y), \\
\hat{y}_{ui} &= \sigma(\mathbf{h}_E^T \phi^E)
\end{aligned} \tag{11}
$$

3.3. Learning the models

The objective function for both ITE and ITE-Si can be defined as follows:

$$
\mathcal{L} = \eta \mathcal{L}_I(\hat{x}, x) + \mathcal{L}_E(\hat{y}, y) + \lambda \mathcal{R}(u, i) \tag{12}
$$

where $\mathcal{L}_I$, $\mathcal{L}_E$ represent the objective function of implicit module and explicit module respectively, η is a hyperparameter to balance the effect of implicit behavior on explicit term. $\mathcal{R}(u, i)$ is the *regularization* to avoid overfitting.

We assume that each observable behavior would take 0 or 1 as their values. For explicit data such as the ratings in n-star scale, we convert to implicit data by marking entry as 1 or 0 indicating the user has rated the item. Define:

$$
\begin{aligned}
\mathcal{L}_I &= \sum_{(u,i) \in \mathcal{X}^+ \cup \mathcal{X}^-} x_{ui} \log \hat{x}_{ui} + (1 - x_{ui}) \log(1 - \hat{x}_{ui}) \\
\mathcal{L}_E &= \sum_{(u,i) \in \mathcal{Y}^+ \cup \mathcal{Y}^-} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui})
\end{aligned} \tag{13}
$$

where $\mathcal{X}^+$, $\mathcal{Y}^+$ denotes the set of observed interactions in behavior matrices X, Y respectively. Let $\mathcal{X}^-$ denote negative instances sampled from the unobserved implicit behaviors in X, and $\mathcal{Y}^-$ be negative instances from the implicit matrix Y. The regularization $\mathcal{R}(u,i)$ is computed as:

$$\mathcal{R}(u,i) = \sum_u ||\mathbf{p_u}||_2^2 + \sum_i ||\mathbf{q_i}||_2^2 \tag{14}$$

Both ITE and ITE-Si use Equation 12 to train the model. For optimizing this objective function, we adopt the Adam method (Kingma and Ba, 2014).

3.4. Benefits of ITE model

In this section, we introduce the advantages of ITE and ITE-Si models.

First, ITE utilizes the idea of combining multiple types of behavior as a sequence of actions and even improves the effectiveness of NMTR (Gao et al., 2019), the sole method attempting to model the order of user actions, by extracting a more complex relation between users and items. More specifically, NMTR aggregates explicit and implicit behavior by connecting multiple predictive outputs (each output represents an interaction) through a scalar which is too simple and thus can not exploit the complicated relation between multiple behavior. In fact, an implicit behavior can affect to the explicit one in many ways. For example, if a user *views* and then *clicks* on a banner, it is likely that the user interests in that product. However, if a user had seen an ad several times but still have not yet clicked then we can infer that the user does not like the product. From this point of view, ITE integrates the implicit behavior with the explicit one via a Deep Neural Network which can discover more complex relationship.

Second, ITE-Si improves the performance of ITE by employing side information. Particularly, additional data such as item description and item category employed in ITE model might provide an initial knowledge about users who have little interaction with items. ITE-Si includes historical activities, item category and item description and thus can have a little knowledge to make a recommendation for new users. Besides performance, ITE-Si is expected to avoid the cold start problem thanks to the enriched the representations of users and items. The benefits of employing additional data to avoid the cold start are reported in vast previous work (Zhang et al., 2010; Zheng et al., 2017; Vasile et al., 2016). We will analyze the advantages of ITE-Si in Section 5.

Finally, since the deep learning architecture of ITE or ITE-Si is flexible, we can easily extend our models to exploit more additional data such as contextual information. In some recommendation problems, users may see a list of suggested items in a target website and the information included in that web is called contextual information. This sort of information reveal partial user preferences and therefore is useful for recommendation systems. For example, a user might incline to visit the websites relevant to his/her hobbies such as sport, fashion, entertainment, etc. The contextual information (if available) can be encoded in a low-dimensional vector similar to side information and is passed directly into ITE or ITE-Si model.

4. Experiments

In this section, we conduct experiments to investigate the practical benefits of ITE and ITE-Si.

Datasets: We perform extensive experiments on the two datasets: Retailrocket and Recobell. Some statistics of Retailrocket and Recobell are shown in Table 1. The details of datasets are as follows:

Retailrocket: An e-commerce dataset taken from Kaggle[1]. This dataset contains the information about the behavioral history of users, the attributes of items and the item category collected in 4.5 months. Multiple types of behavior include *click*, *add to cart* and *transaction*. We determine that *click* is an implicit behavior, while *add to cart* and *transaction* are explicit. The users have less than 5 interactions are removed from the dataset.

Recobell: A dataset collected from an e-commerce website[2]. This data includes behavioral data and item label information collected over a period of 2 months from August to October of 2016. Behavioral data contains the *view* (implicit behavior) and *order* (explicit behavior). Similar to Retailrocket, the users have less than 5 interactions would be removed.

Dataset	RetailRocket	Recobell
Implicit #	396 923	2 285 261
Explicit #	18 450	52 786
users #	36 751	206 203
items #	83 274	118 293
labels #	1 699	1 939
Sparsity	99.987%	99.999%

Table 1: Statistics of the Retailrocket and Recobell datasets

Models in use: We evaluate the three models: ITE, ITE-Si and a variant of ITE-Si.

- **ITE (Implicit to Explicit):** the model described in Section 3.1 which represents users and items by one-hot encoding.
- **ITE-Si (ITE with Side information):** the model described in Section 3.2 which incorporates side information in both user and item representations.
- **ITE-OSSi (ITE with one-sided Side information):** A variant of ITE-Si which just incorporates side information in item representation. Particularly, $u^{(m)}$, $i^{(m)}$ are the one-hot encoding vectors, $u^{(a)}$ is none and $i^{(a)}$ is the same as ITE-Si which described in Section 3.2, i.e., the item category. ITE-OSSi is a simple version of ITE-Si when directly using the raw data of item category instead of combining side information with historical activities. We take ITE-OSSi into consideration to investigate further impact of side information.

Baselines: We take two state-of-the-art models into comparison:

- NMTR (Gao et al., 2019): A model which captures the cascading relation between implicit and explicit behavior as multi-task learning.

1. https://www.kaggle.com/retailrocket/ecommerce-dataset
2. http://www.recobell.co.kr/rb/main.php?menu=pakdd2017

- MTMF (Shi et al., 2017): A model which exploit several actions such as view, want and click separately.

Parameter Settings: The parameters are created randomly by the Gaussian Distribution. Several hyper-parameters having major impacts to the model are: lr (learning rate), $batch\_size$, η (to balance the influence of implicit module to explicit module), K (the number of factors in the last hidden layer of implicit module). We use the testing set as validation data and tune the hyper-parameters with $batch\_size$ in $\{512, 1024, 2048, 4096\}$, $lr \in \{0.0001, 0.0005, 0.001\}$, $\eta \in \{0.1, 0.5, 1.0, 2.0\}$. The tuning process give us $lr = 0.001$, $batch\_size = 2048$ for all the models; $\eta = 1.0$ for NMTR and $\eta = 0.5$ for all the other models. In the experiments, we analyze the impact of K which is called the **num factors** with $K \in \{8, 16, 32, 64\}$. Note that higher K might improve the performance, but if K is too large the model may suffer from over-fitting.

For all the models, we sample four negative instances per positive instance. All the ITE models follow the tower pattern, i.e., the number of hidden nodes in the next layer is equal to half of current layer. Additionally, for ITE, ITE-OSSi and ITE-Si, the representation of user u is a M-dimensional vector (the number of users), while item i is represented by a N-dimensional vector (the number of items).

Evaluated method: We use the leave-one-out (Rendle et al., 2009; He et al., 2016) method to evaluate the performance. For each user, we extract the most recent item that the user has interacted to the test set. The remain items would be used to train. In the evaluation phase, we rank all the items that each user have not interacted with. To decrease the computational time, we rank randomly 999 items instead of all the items.

The measure score we use are *Hit Ratio* (HR) (Deshpande and Karypis, 2004) and *Normalized Discounted Cumulative Gain* (NDCG) (He et al., 2015) to rank all the items in the test set. The goal is to compute HR@K and NDCG@K at $K = 10$.

Hit Ratio (HR): For each user, HR@K corresponds to whether the test item belongs to the top K items of that user. HR@K can be formulated as follows:

$$HR@K = \begin{cases} 1, & \text{if test item is in top K.} \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Normalized Discounted Cumulative Gain (NDCG): Instead of checking whether the test item is in top K as Hit Ratio, NDCG@K consider the ranking of the test item in top K items. The NDCG@K score is higher, the ranking is better. For each user, NDCG@K can be formulated as follows:

$$NDCG@K = \begin{cases} \frac{log(2)}{log(i+1)}, & \text{if test item is ranked at position i.} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The HR@K and NDCG@K for the entire system can be summarized using the average HR@K and NDCG@K from all users.

Performance comparison: Fig. 4 and Fig. 5 show the HR@10 and NDCG@10 with respect to the **num factors** K, $K \in \{8, 16, 32, 64\}$. In Fig 4, ITE achieves higher results of

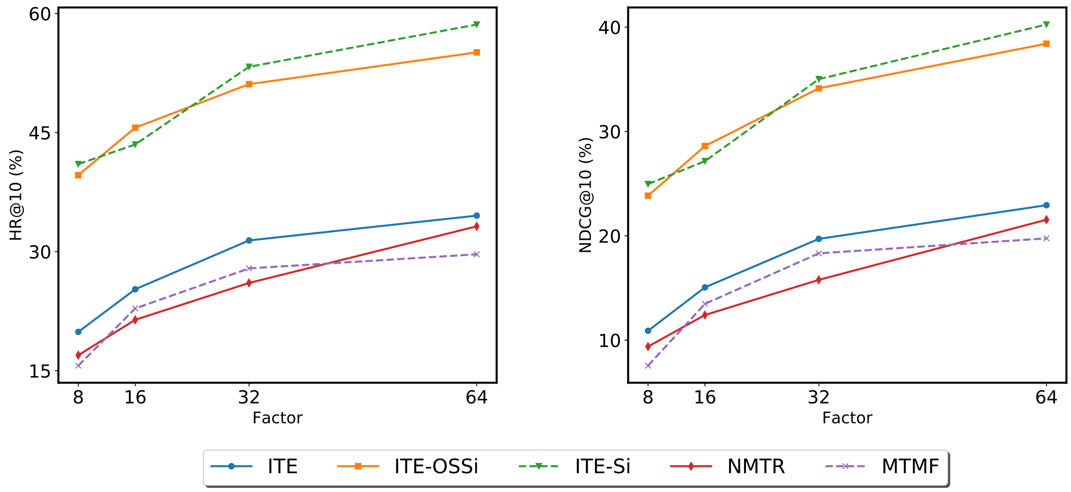

Figure 4: Comparison of the various models in **Retail Rocket** with K varied

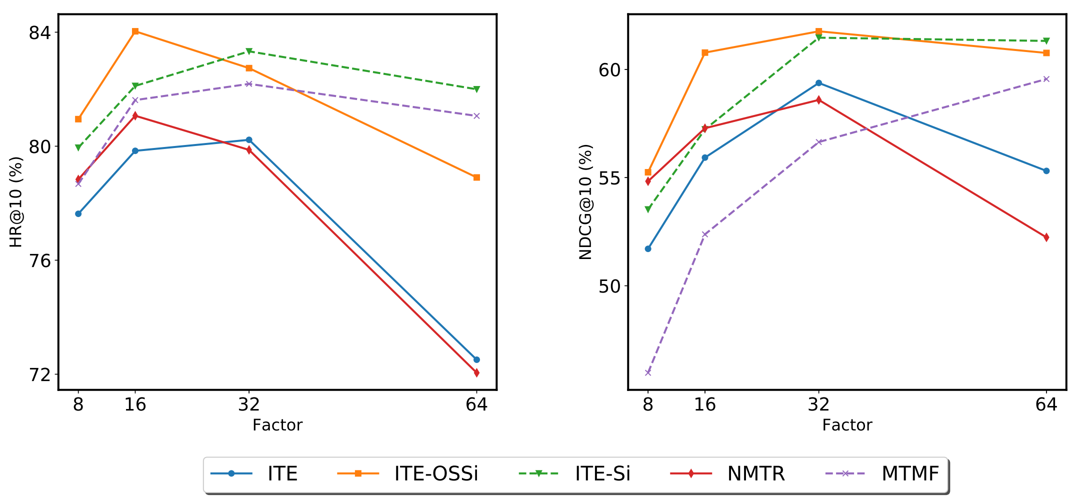

Figure 5: Comparison of the various models in **Recobell** with K varied

both *HR@10* and *NDCG@10* than MTMF and NMTR which explains the benefit of ITE over NMTR as discussed in Section 3.1. For the result of *Recobell* in Fig. 5, the figures for ITE, MTMF and NMTR are fluctuated. The best NDCG@10 of ITE is 0.59 (K=32), while the highest of MTMF and NMTR are 0.59 ($K = 64$) and 0.58 ($K = 32$) respectively. The efficiency of side information is shown in the performance of ITE-OSSi and ITE-Si, where these two models are the highest figures in most of the cases.

Fig. 6 and Fig. 7 reveal the performance of ITE, ITE-OSSi and ITE-Si on *Retail Rocket* and *Recobell* when the number of epochs increases gradually. An epoch is one learning

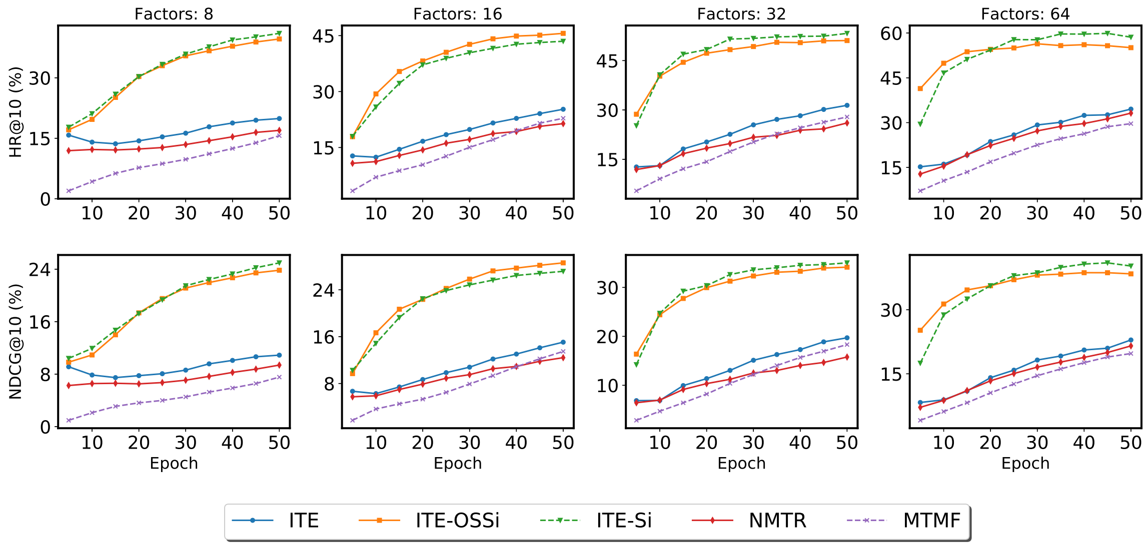

Figure 6: Comparison of the various models in **Retail Rocket** when increasing the number of epochs gradually. From left to right: $K = 8, 16, 32, 64$

cycle where the model learns through the whole training data. As we can see, ITE-OSSi and ITE-Si still achieve better results than the other models in most of the cases. When comparing ITE with MTMF and NMTR, ITE performs better than the others for Retail Rocket, while Recobell experiences the fluctuated results of the three models. Furthermore, the three models ITE, ITE-OSSi and ITE-Si gain high performance in several first epochs, which indicates that we can save training time for these models.

5. Conclusion

We have introduced ITE, a neural network for modeling the sequence of user behavior in real-world. ITE successfully learns the complex relation between implicit and explicit behavior thus makes the prediction more precisely. Additionally, we introduce an extended version of ITE, namely ITE with Side information (ITE-Si) to utilize the advantages of side information. Particularly, ITE-Si exploits a third-party resource besides the interaction between users and the items, i.e., information of items such as item features, music genres, movie categories and therefore facilitate significantly the predictive accuracy of the model versus ITE. Apart from improving the model performance, both ITE-Si is expected to avoid the cold start thanks to additional information attached in the input layer. We left further study on this benefit for future work. Extensive experiments show that ITE outperforms the state-of-the-art methods. Furthermore, ITE-Si gains higher experimental results than ITE in the two e-commerce dataset, which indicates the effectiveness of side information.

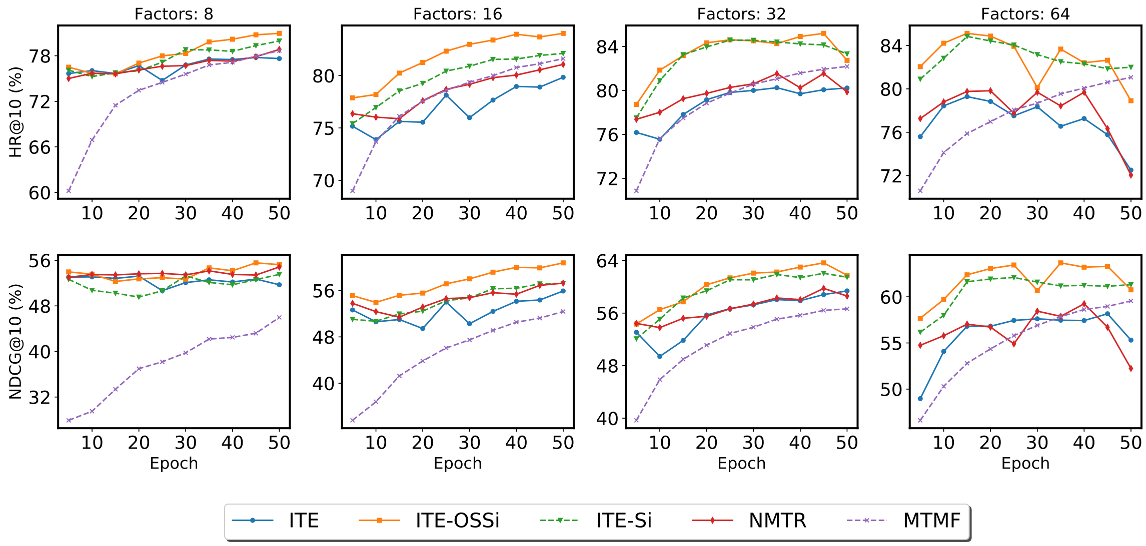

Figure 7: Comparison of the various models in **Recobell** when increasing the number of epochs gradually. From left to right: $K = 8, 16, 32, 64$

Acknowledgments

This research is supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA18, and by the Office of Naval Research Global (ONRG) under Award Number N62909-18-1-2072, and Air Force Office of Scientific Research (AFOSR), Asian Office of Aerospace Research & Development (AOARD) under Award Number 17IOA031.

References

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *DLRS*, pages 7–10. ACM, 2016.

Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1554–1557. IEEE, 2019.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1661–1670. ACM, 2015.

Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International*

ACM SIGIR conference on Research and Development in Information Retrieval, pages 549–558. ACM, 2016.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

Thanh Hai Hoang, Anh Phan Tuan, Linh Ngo Van, and Khoat Than. Enriching user representation in neural matrix factorization. In *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE, 2019.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008.

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

Yehuda Koren. Recommender system utilizing collaborative filtering combining explicit and implicit feedback with both neighborhood and latent factor models, October 11 2011. US Patent 8,037,080.

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

Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *WSDM*, pages 173–182. ACM, 2012.

Hoa M Le, Son Ta Cong, Quyen Pham The, Ngo Van Linh, and Khoat Than. Collaborative topic model for poisson distributed ratings. *International Journal of Approximate Reasoning*, 95:62–76, 2018.

Terje Nesbakken Lillegraven and Arnt Christian Wolden. Design of a bayesian recommender system for tourists presenting a solution to the cold-start user problem. Master's thesis, Institutt for datateknikk og informasjonsvitenskap, 2010.

Nathan N Liu, Evan W Xiang, Min Zhao, and Qiang Yang. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1445–1448. ACM, 2010.

Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

Mohammad-Hossein Nadimi-Shahraki and Mozhde Bahadorpour. Cold-start problem in collaborative recommender systems: efficient methods based on ask-to-rate technique. *Journal of computing and information technology*, 22(2):105–113, 2014.

Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

Wanlu Shi, Tun Lu, Dongsheng Li, Peng Zhang, and Ning Gu. Multi-task matrix factorization for collaborative filtering. In *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 343–348. IEEE, 2017.

Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008.

Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 225–232. ACM, 2016.

Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

Zi-Ke Zhang, Chuang Liu, Yi-Cheng Zhang, and Tao Zhou. Solving the cold-start problem in recommender systems with social tags. *EPL (Europhysics Letters)*, 92(2):28002, 2010.

Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H Chi. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1406–1416. International World Wide Web Conferences Steering Committee, 2015.

Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 425–434. ACM, 2017.

Philip Zigoris and Yi Zhang. Bayesian adaptive user profiling with explicit & implicit feedback. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 397–404. ACM, 2006.