An Efficient Adaptive Transfer Neural Network for Social-aware Recommendation

Chong Chen[1], Min Zhang[1*], Chenyang Wang[1], Weizhi Ma[1],

Minming Li[2], Yiqun Liu[1] and Shaoping Ma[1]

[1]Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology, Tsinghua University
[2]Department of Computer Science, City University of Hong Kong
cc17@mails.tsinghua.edu.cn,z-m@tsinghua.edu.cn

ABSTRACT

Many previous studies attempt to utilize information from other domains to achieve better performance of recommendation. Recently, social information has been shown effective in improving recommendation results with transfer learning frameworks, and the transfer part helps to learn users' preferences from both item domain and social domain. However, two vital issues have not been well-considered in existing methods: 1) Usually, a static transfer scheme is adopted to share a user's common preference between item and social domains, which is not robust in real life where the degrees of sharing and information richness are varied for different users. Hence a non-personalized transfer scheme may be insufficient and unsuccessful. 2) Most previous neural recommendation methods rely on negative sampling in training to increase computational efficiency, which makes them highly sensitive to sampling strategies and hence difficult to achieve optimal results in practical applications.

To address the above problems, we propose an Efficient Adaptive Transfer Neural Network (EATNN). By introducing attention mechanisms, the proposed model automatically assign a personalized transfer scheme for each user. Moreover, we devise an efficient optimization method to learn from the whole training set without negative sampling, and further extend it to support multi-task learning. Extensive experiments on three real-world public datasets indicate that our EATNN method consistently outperforms the state-of-the-art methods on Top-K recommendation task, especially for cold-start users who have few item interactions. Remarkably, EATNN shows significant advantages in training efficiency, which makes it more practical to be applied in real E-commerce scenarios. The code is available at (https://github.com/chenchongthu/EATNN).

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**;

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331192

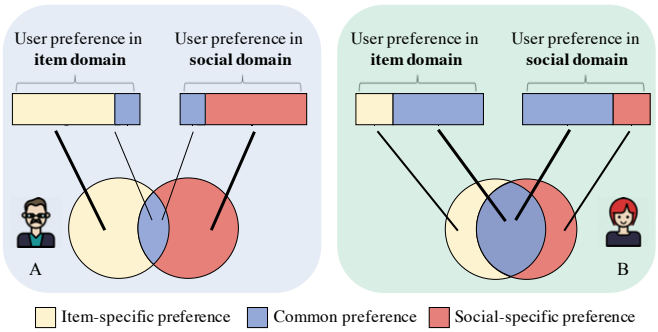

Figure 1: Social connections have been shown to be helpful for user preference modeling, but the preference sharing between item domain and social domain are varied for different users in real life.

KEYWORDS

Recommender Systems, Adaptive Transfer Learning, Whole-data based Learning, Social Connections, Implicit Feedback

ACM Reference Format:
Chong Chen[1], Min Zhang[1], Chenyang Wang[1], Weizhi Ma[1], Minming Li[2], Yiqun Liu[1] and Shaoping Ma[1]. 2019. An Efficient Adaptive Transfer Neural Network for Social-aware Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3331184.3331192

1 INTRODUCTION

Recommender systems provide essential web services on the Internet to alleviate the information overload problem. Recently, many E-commerce sites, such as Ciao, Epinion, and Flixster, have become popular social platforms in which users can follow other users, discuss, and select items. The social connections in these applications reflect users' interests or profiles, which are helpful for user modeling and personalized recommendation.

Traditional Collaborative Filtering (CF) methods [12–14, 28] mainly make use of users' historical records such as ratings, clicks, and purchases. Although they have shown good results, the performance will degrade significantly when the records matrix is very sparse. To address the lack of data, there has been a trend to augment user-item interactions with users' social connections for recommendation [4, 18, 20, 27, 43]. Generally, a user's preferences

can not only be inferred from the items he/she bought and clicked, but also be affected by his/her social connections. As a result, social-aware methods can utilize a much larger volume of data to tackle the data sparsity issue, and further improve the performance of recommender systems [4].

Many social-aware recommendation methods are based on transfer learning [15, 18, 25], as it is a suitable choice for the coordination of user-item interactions and user-user connections. The key concept behind transfer learning is to transfer the shared knowledge from one domain to other domains. However, most existing methods simply transfer a fixed proportion of common knowledge between item domain and social domain for each user [15, 18, 31], which is not robust in real life due to: 1) the information richness of the two domains usually varies for different users; 2) the degrees of the preference sharing between the two domains are varied for different users. As shown in Figure 1, user B has similar preferences in item domain and social domain, while A only shares very few preferences between the two domains. Therefore, a non-personalized transfer is insufficient and unsuccessful. To better characterize users' preferences, recommender systems require adaptive transfer schemes for different users.

In addition, since implicit data is often a natural byproduct of users' behavior (e.g., browsing histories, click logs), user interactions that can be observed in both item and social domains are rather limited, and non-observed instances, which is taken as negative examples in model learning, are with much larger scale. To increase computational efficiency, existing neural recommendation methods [4, 10, 33, 34, 40] mainly rely on negative sampling for optimization, which is, however, highly sensitive to the sampling distribution and the number of negative samples [10]. Moreover, social-aware recommendation usually needs to optimize the loss function in both item and social domains, which is a multi-task problem. Hence for social-aware problem, it is more difficult for sampling-based strategy to converge to the optimum performance. By contrast, whole-data based strategy computes the gradient on all training data. Thus it can easily converge to a better optimum [12, 41]. Unfortunately, the difficulty in applying whole-data based strategy lies in the expensive computational cost for large-scale data, which makes it less applicable to neural models.

Motivated by the above observations, in this paper, we propose an Efficient Adaptive Transfer Neural Network (EATNN) for social-aware recommendation. To adaptively capture the interplay between item and social domain for each user, we introduce attention mechanisms [1, 3] to automatically estimate the difference of mutual influence between item domain and social domain. The key idea is to learn two attention-based kernels to model the weights of the outputs come from different domains. Besides, we propose an efficient optimization method to learn from the whole training set without negative sampling, and extend it to support multi-task learning. To ensure training efficiency, we accelerate the optimization method by reformulating a commonly used square loss function with rigorous mathematical reasoning. By leveraging sparsity in implicit data, we succeed to update each parameter in a manageable time complexity without sampling.

To evaluate the recommendation performance and training efficiency of our model, we apply EATNN on three real-world datasets with extensive experiments. The results indicate that our model consistently outperforms the state-of-the-art methods on Top-K personalized recommendation task, especially for cold-start users who have few item interactions. Furthermore, EATNN also shows significant advantages in training efficiency, which makes it more practical in real E-commerce scenarios. The main contributions of this work are as follows:

(1) We propose a novel Efficient Adaptive Transfer Neural Network for social-aware recommendation. By introducing attention mechanisms, the proposed model can adaptively capture the interplay between item domain and social domain for each user.
(2) We devise an efficient optimization method to avoid negative sampling and achieve more accurate performance. The proposed method is not only suitable for learning from implicit data that only contains positive examples, but also capable of jointly learning multi-task problems.
(3) Extensive experiments are conducted on three benchmark datasets. The results show that EATNN consistently and significantly outperforms the state-of-the-art models in terms of both recommendation performance and training efficiency.

2 RELATED WORK

2.1 Social-aware Recommendation

Social-aware recommendation aims at leveraging users' social connections to improve the performance of recommender systems. It works based on the assumption that users tend to share similar preferences with their friends. In previous work, Krohn et al. [15] proposed a Multi-Relational Bayesian Personalized Ranking (MR-BPR) model based on Collective Matrix Factorization (CMF) [31], which predicts both user feedback on items and on social connections. Zhao et al. [43] assumed that users are more likely to have seen items consumed by their friends, and extended BPR [28] by changing the negative sampling strategy (SBPR). Recently, the authors in [18] proposed to consider the visibility of both items and social relationships, and utilized transfer learning to combine the item and social domains for recommendation (TranSIV).

Since deep learning has yielded great success in many fields, some researchers also tried to explore different neural network structures for social-aware recommendation task. For instance, Sun et al. [33] presented an attentive recurrent network for temporal social-aware recommendation (ARSE). Wang et al. [37] enhanced NCF method [10] by combining with the graph regularization technique to model the cross-domain social relations. Recently, Chen et al. [4] proposed a Social Attentional Memory Network (SAMN), which considered to model both aspect- and friend-level differences in social-aware recommendation. However, existing neural methods [4, 10, 33, 37] mainly rely on negative sampling for model optimization, which may limit the performance of recommender systems. Efficient optimization from all training data without sampling is one of the main concerns of this paper.

2.2 Transfer Learning

Transfer learning has been adopted in various systems for cross-domain tasks [2, 22, 29, 45]. The key idea of transfer learning is to transfer the common knowledge from the source domain to the target domain. As previously noted [44], social media contains multi-domain information, which provides a bridge for transfer learning. In previous studies, Roy et al. [29] utilized transfer learning to deal with multi-relational data representation in social networks, but it

did not specifically focus on recommendation tasks. Eaton et al. [9] pointed out that parts of the source domain data are inconsistent with the target domain observations, which may affect the construction of the model in the target domain. Based on that, some researchers [18, 19] designed selective latent factor transfer models to better capture the consistency and heterogeneity across domains. However, in these work, the transfer ratio needs to be properly selected through human effort and can not change dynamically in different scenarios.

There are also some studies considering the adaption issue in transfer learning. However, existing methods mainly focus on task adaptation or domain adaption. E.g., based on Gaussian Processes, Cao et al. [2] proposed to adapt the transfer-all and transfer-none schemes by estimating the similarity between a source and a target task. Zhang et al. [42] studied domain adaptive transfer learning which assumed that the pre-training and test sets have different distributions. The authors in [22] designed a method for completely heterogeneous transfer learning to determine different transferability of source knowledge. Our work differs from the above studies as the designed model is not limited to task adaptation or domain adaption. Instead, we propose to adapt each user's two kinds of information (item interactions and social connections) with a finer granularity, which allows the shared knowledge of each user to be transferred in a personalized manner.

2.3 Model Learning in Recommendation

There are two strategies to optimize a recommendation model with implicit feedback: 1) negative sampling strategy [4, 10, 28] that samples negative instances from missing data; 2) whole-data based strategy [7, 12, 17, 18] that sees all the missing data as negative. As shown in previous studies [11, 41], both strategies have pros and cons: negative sampling strategy is more efficient by reducing negative examples in training, but may decrease the model's performance; whole-data based strategy leverages the full data with a potentially better coverage, but inefficiency can be an issue. Existing neural recommendation methods [4, 10, 33, 40] mainly rely on negative sampling for efficient optimization. To retain the model's fidelity, we persist in whole-data based learning in this paper, and develop a fast optimization method to address the inefficiency issue.

Some efforts have been devoted to resolving the inefficiency issue of whole-data based strategy. Most of them are based on Alternating Least Squares (ALS) [12]. E.g., Pilaszy et al. [26] described an approximate solution of ALS. He et al. [11] proposed an efficient element-wise ALS with non-uniform missing data. Unfortunately, ALS based methods are not applicable to neural models which use Gradient Descent (GD) for optimization. Recently, some researchers [39, 41] studied fast Batch Gradient Descent (BGD) methods to learn from all training examples. However, they only focus on optimizing traditional non-neural models. Distinct from previous studies, we derive a new whole-data based loss function, which is, to the best of our knowledge, the first efficient whole-data based learning strategy tailored for neural recommendation models. The loss function is further extended to jointly learn both item and social domains in our model.

3 PRELIMINARY

We first introduce the key notations used in this work and the whole-data based MF method for learning from implicit data.

Table 1: A summary of key notations in this work.

Symbol	Description
$\mathbf{U}$	set of users
$\mathbf{B}$	batch of users
$\mathbf{V}$	set of items
$\mathbf{R}$	user-item interactions
$\mathcal{R}$	the set of user-item pairs whose values are non-zero
$\mathbf{X}$	user-user social connections
$\mathcal{X}$	the set of user-user social pairs whose values are non-zero
$\mathbf{u}^I$	item-specific latent factor vector of user u
$\mathbf{u}^S$	social-specific latent factor vector of user u
$\mathbf{u}^C$	common latent factor vector of user u
$\mathbf{p}_u^I$	latent vector of user u for item domain after transferring
$\mathbf{p}_u^S$	latent vector of user u for social domain after transferring
$\mathbf{q}_v$	latent factor vector of item v
$\mathbf{g}_t$	latent factor vector of user t as a friend
c_{uv}^I	the weight of entry R_{uv}
c_{ut}^S	the weight of entry X_{ut}
$\alpha_{(I,u)}$	the weight of item-specific vector $\mathbf{u}^I$ for item domain
$\alpha_{(C,u)}$	the weight of common vector $\mathbf{u}^C$ for item domain
$\beta_{(S,u)}$	the weight of social-specific vector $\mathbf{u}^S$ for social domain
$\beta_{(C,u)}$	the weight of common vector $\mathbf{u}^C$ for social domain
d	latent factor number
Θ	set of neural parameters

3.1 Notations

Table 1 depicts the notations and key concepts. Suppose we have M users and N items in the dataset, and we use the index u to denote a user, t to denote another user, and v to denote an item. There are two kinds of observed interactions: user-item interactions $\mathbf{R} = [R_{uv}]_{M\times N} \in \{0,1\}$ indicates whether u has purchased or clicked on item v, and user-user social interactions $\mathbf{X} = [X_{ut}]_{M\times M} \in \{0,1\}$ indicates whether u trusts (or is a friend of) t in the social network. $\mathcal{R}$ and $\mathcal{X}$ denote the sets of interactions whose values are non-zero for the item domain and the social domain, respectively.

For user u, $\mathbf{u}^C$ represents the latent factors shared between the item and social domains; $\mathbf{u}^I$ and $\mathbf{u}^S$ represent user latent factors corresponding to the item and social domains, respectively. Vector $\mathbf{q}_v$ denotes the latent vector of v, and $\mathbf{g}_t$ denotes the latent vector of t as a friend. α and β are the parameters for adaptive transfer learning. Vector $\mathbf{p}_u^I$ and $\mathbf{p}_u^S$ are the representations of user u for item domain and social domain after transferring, respectively. More details are introduced in Section 4.

3.2 MF Method for Implicit Feedback

Matrix Factorization (MF) maps both users and items into a joint latent feature space of d dimension such that interactions are modeled as inner products in that space. Mathematically, each entry R_{uv} of $\mathbf{R}$ is estimated as:

$$\hat{R}_{uv} =< \mathbf{p}_u, \mathbf{q}_v >= \mathbf{p}_u^T \mathbf{q}_v \tag{1}$$

The item recommendation problem is formulated as estimating the scoring function R_{uv}, which is used to rank items.

For implicit data, the observed interactions are rather limited, and non-observed examples are of a much larger scale. To learn model parameters, Hu et al. [12] introduced a weighted regression function, which associates a confidence to each prediction in the

implicit feedback matrix $\mathbf{R}$:

$$\mathcal{L}(\Theta) = \sum_{u \in \mathbf{U}} \sum_{v \in \mathbf{V}} c_{uv}(R_{uv} - \hat{R}_{uv})^2 \tag{2}$$

where c_{uv} denotes the weight of entry R_{uv}. Note that in implicit feedback learning, missing entries are usually assigned a zero R_{uv} value but non-zero c_{uv} weight.

As can be seen, the time complexity of computing the loss in Eq.(2) is $O(|\mathbf{U}||\mathbf{V}|d)$. Clearly, the straightforward way to calculate gradients is generally infeasible, because $|\mathbf{U}||\mathbf{V}|$ can easily reach billion level or even higher in real life.

4 EFFICIENT ADAPTIVE TRANSFER NEURAL NETWORK (EATNN)

In this section, we first present a general overview of the EATNN framework, then introduce the two key ingredients of our proposed model in detail, which are: 1) attention-based adaptive transfer learning and 2) efficient whole-data based optimization.

4.1 Model Overview

The goal of our model is to make item recommendations based on implicit feedback and social data. The overall model architecture is described in Figure 2. From the figure, we can present a simple high-level overview of our model:

(1) Users and items are converted to dense vector representations through embeddings. Specifically, as the bridge of transfer learning, each user u has three latent vectors. $\mathbf{u}^C$ represents the knowledge shared between the item and social domains. $\mathbf{u}^I$ and $\mathbf{u}^S$ represent user u's specific preference corresponding to item domain and social domain, respectively.
(2) An attention-based adaptive transfer layer is designed to automatically model the domain relationships and learn domain-specific functionalities to leverage shared representations. It allows parameters to be automatically allocated to capture both shared knowledge and domain-specific knowledge, while avoiding the need of adding many new parameters.
(3) The model is jointly optimized by a newly derived efficient whole-data based training strategy.

4.2 Attention-based Adaptive Transfer

Attention mechanism has been widely utilized in many fields, such as computer vision [5], machine translation [1], and recommendation [3, 4, 38]. Since attention mechanism has superior ability to assign non-uniform weights according to input instances, it is adopted in our model to achieve personalized adaptive transfer learning. Specifically, we apply attention networks for item domain and social domain respectively. Each is a two-layer network with user representations ($\mathbf{u}^C$, $\mathbf{u}^I$, $\mathbf{u}^S$) as the inputs. For a user, if the two domains are less related, then the shared knowledge ($\mathbf{u}^C$) will be penalized and the attention network will learn to utilize more domain-specific information ($\mathbf{u}^I$ or $\mathbf{u}^S$) instead. Formally, the item domain attention and the social domain attention are defined as:

$$\begin{aligned} \alpha^*_{(C,u)} &= \mathbf{h}_\alpha^T \sigma(\mathbf{W}_\alpha \mathbf{u}^C + \mathbf{b}_\alpha); \quad \alpha^*_{(I,u)} = \mathbf{h}_\alpha^T \sigma(\mathbf{W}_\alpha \mathbf{u}^I + \mathbf{b}_\alpha) \\ \beta^*_{(C,u)} &= \mathbf{h}_\beta^T \sigma(\mathbf{W}_\beta \mathbf{u}^C + \mathbf{b}_\beta); \quad \beta^*_{(S,u)} = \mathbf{h}_\beta^T \sigma(\mathbf{W}_\beta \mathbf{u}^S + \mathbf{b}_\beta) \end{aligned} \tag{3}$$

where $\mathbf{W}_\alpha \in \mathbb{R}^{k \times d}$, $\mathbf{b}_\alpha \in \mathbb{R}^k$, $\mathbf{h}_\alpha \in \mathbb{R}^k$ are parameters of the item domain attention, $\mathbf{W}_\beta \in \mathbb{R}^{k \times d}$, $\mathbf{b}_\beta \in \mathbb{R}^k$, $\mathbf{h}_\beta \in \mathbb{R}^k$ are parameters of the social domain attention. d is the dimension of embedding vector, k is the dimension of the attention network, and σ is the nonlinear activation function *ReLU* [23].

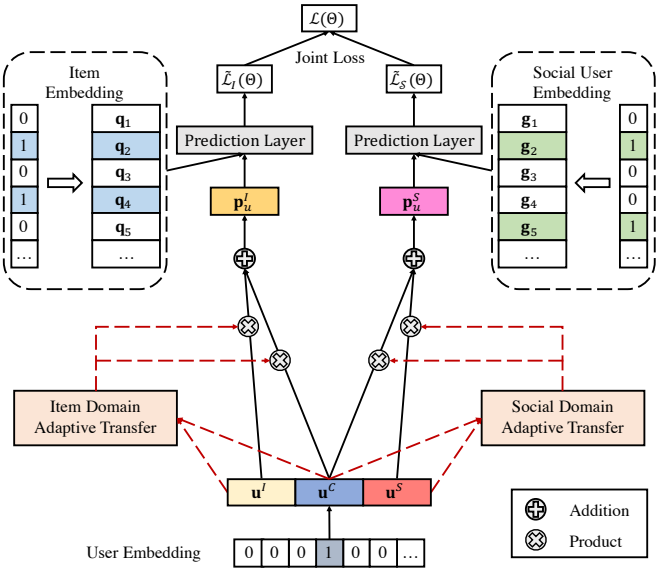

Figure 2: Illustration of our Efficient Adaptive Transfer Neural Network (EATNN).

Then, the final attention scores are normalized with a softmax function:

$$\begin{aligned} \alpha_{(C,u)} &= \frac{\exp(\alpha^*_{(C,u)})}{\exp(\alpha^*_{(C,u)}) + \exp(\alpha^*_{(I,u)})} = 1 - \alpha_{(I,u)} \\ \beta_{(C,u)} &= \frac{\exp(\beta^*_{(C,u)})}{\exp(\beta^*_{(C,u)}) + \exp(\beta^*_{(I,u)})} = 1 - \beta_{(S,u)} \end{aligned} \tag{4}$$

$\alpha_{(C,u)}$ and $\beta_{(C,u)}$ are the weights of shared knowledge ($\mathbf{u}^C$) for item domain and social domain respectively, which determine how much to transfer in each domain. After obtaining the above attention weights, the representations of user u for the two domains are calculated as follows:

$$\mathbf{p}_u^I = \alpha_{(I,u)}\mathbf{u}^I + \alpha_{(C,u)}\mathbf{u}^C; \quad \mathbf{p}_u^S = \beta_{(S,u)}\mathbf{u}^S + \beta_{(C,u)}\mathbf{u}^C \tag{5}$$

$\mathbf{p}_u^I$ and $\mathbf{p}_u^S$ are two feature vectors, which represent the user's preferences for items and other users after transferring the common knowledge between the two domains.

Based on the learnt feature vectors, the prediction part aims to generate a score that indicates a user's preferences for an item or a friend. The prediction part is built on a neural form of MF [10]. For each domain task, a specific output layer is employed. The scores of user u for item v and another user t are calculated as follows:

$$\hat{R}_{uv} = \mathbf{h}_I^T(\mathbf{p}_u^I \odot \mathbf{q}_v); \quad \hat{X}_{ut} = \mathbf{h}_S^T(\mathbf{p}_u^S \odot \mathbf{g}_t) \tag{6}$$

where $\mathbf{q}_v \in \mathbb{R}^d$ and $\mathbf{g}_t \in \mathbb{R}^d$ are latent vectors of item v and user t as a friend, $\odot$ denotes the element-wise product of vectors, and $\mathbf{h}_I \in \mathbb{R}^d$ and $\mathbf{h}_S \in \mathbb{R}^d$ denote the output layer for item domain and social domain, respectively. Then for our target task – recommendation, the candidate items will be ranked in descending order of $\hat{R}_{uv}$ to provide Top-K item recommendation list.

4.3 Efficient Whole-data based Learning

To improve the speed of whole-data based optimization, we derive an efficient loss function for learning from implicit feedback.

4.3.1 ***Weighting Strategy.*** We first present the weighting strategy for each entry in matrices $\mathbf{R}$ and $\mathbf{X}$. There have been many studies on how to assign proper weights for implicit interactions, such as uniform weighting strategy [12, 26, 36] and frequency-based weighting strategy [11, 17]. Since this is not the main concern of our work, we follow the settings of previous work [11]: 1) the weight of each positive entry (c_{uv}^{I+} and c_{ut}^{S+}) is set to 1; 2) the weights of negative instances are calculated as follows, which assign the larger weights to the items and friends with higher frequencies:

$$\begin{aligned} c_{uv}^{I-} &= c_v^{I-} = c_0^I \frac{m_v^{\rho_1}}{\sum_{j=1}^{V} m_j^{\rho_1}};\ m_v = \frac{|\mathcal{R}_v|}{\sum_{j=1}^{V} |\mathcal{R}_j|} \\ c_{ut}^{S-} &= c_t^{S-} = c_0^S \frac{n_t^{\rho_2}}{\sum_{j=1}^{U} n_j^{\rho_2}};\ n_t = \frac{|\mathcal{X}_t|}{\sum_{j=1}^{U} |\mathcal{X}_j|} \end{aligned} \tag{7}$$

where m_v and n_t denote the frequency of item v and friend t in $\mathbf{R}$ and $\mathbf{X}$, respectively; $\mathcal{R}_v$ and $\mathcal{X}_t$ denote the positive interactions of v and t; c_0^I and c_0^S determine the overall weight of missing data, and ρ_1 and ρ_2 control the significance level of popular items over unpopular ones.

4.3.2 ***Loss Inference.*** In our method, the loss functions of item domain and social domain only differ in their inputs, thus we focus on illustrating the inference of the item domain in detail.

According Eq.(2), for a batch of users, the loss of item domain is:

$$\begin{aligned} \mathcal{L}_I(\Theta) &= \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}} c_{uv}^I (R_{uv} - \hat{R}_{uv})^2 \\ &= \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}} c_{uv}^I (R_{uv}^2 - 2R_{uv}\hat{R}_{uv} + \hat{R}_{uv}^2) \end{aligned} \tag{8}$$

In implicit data, since $R_{uv} \in \{0, 1\}$ indicates whether u has purchased or clicked on item v, it can be replaced by a constant to simplify the equation. Also, the loss of missing data can be expressed by the residual between the loss of all data and that of positive data:

$$\begin{aligned} \mathcal{L}_I(\Theta) &= const - 2\sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}^+} c_{uv}^{I+} \hat{R}_{uv} + \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}} c_{uv} \hat{R}_{uv}^2 \\ &= const - 2\sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}^+} c_{uv}^{I+} \hat{R}_{uv} + \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}^+} c_{uv}^{I+} \hat{R}_{uv}^2 \\ &\quad + \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}^-} c_{uv}^{I-} \hat{R}_{uv}^2 \\ &= const - 2\sum_{u\in B} \sum_{v\in\mathbf{V}^+} c_{uv}^{I+} \hat{R}_{uv} + \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}^+} c_{uv}^{I+} \hat{R}_{uv}^2 \\ &\quad + \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}} c_{uv}^{I-} \hat{R}_{uv}^2 - \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}^+} c_{uv}^{I-} \hat{R}_{uv}^2 \\ &= const + \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}^+} \left((c_{uv}^{I+} - c_{uv}^{I-}) \hat{R}_{uv}^2 - 2c_{uv}^{I+} \hat{R}_{uv} \right) \\ &\quad + \underbrace{\sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}} c_{uv}^{I-} \hat{R}_{uv}^2}_{\mathcal{L}_I^{\mathcal{A}}(\Theta)} \end{aligned} \tag{9}$$

where *const* denotes a Θ-invariant constant value, and $\mathcal{L}_I^{\mathcal{A}}(\Theta)$ denotes the loss for all data. Thus, $\mathcal{L}_I(\Theta)$ can be seen as a combination of the loss of positive data and the loss of all data. And the loss of missing data has been eliminated. The new computational bottleneck lies in $\mathcal{L}_I^{\mathcal{A}}(\Theta)$ now.

Recall the prediction of $\hat{R}_{uv}$, we have:

$$\hat{R}_{uv} = \mathbf{h}_I^T (\mathbf{p}_u^I \odot \mathbf{q}_v) = \sum_{i=1}^{d} h_{I,i} p_{u,i}^I q_{v,i} \tag{10}$$

Based on a decouple manipulation for the inner product operation, the summation operator and elements in $\mathbf{p}_u$ and $\mathbf{q}_v$ can be rearranged.

$$\begin{aligned} \hat{R}_{uv}^2 &= \sum_{i=1}^{d} h_{I,i} p_{u,i}^I q_{v,i} \sum_{j=1}^{d} h_{I,j} p_{u,j}^I q_{v,j} \\ &= \sum_{i=1}^{d} \sum_{j=1}^{d} (h_{I,i} h_{I,j}) \left(p_{u,i}^I p_{u,j}^I \right) (q_{v,i} q_{v,j}) \end{aligned} \tag{11}$$

By substituting Eq.(11) in $\mathcal{L}_I^{\mathcal{A}}(\Theta)$, there emerges a nice structure: if we set c_{uv}^- to c_v^- (Eq.(7)), the interaction between $p_{u,i}^I$ and $q_{v,i}$ can be properly separated. Thus, the optimization of $\sum_{v\in\mathbf{V}} c_v^{I-} q_{v,i} q_{v,j}$ and $\sum_{u\in\mathbf{B}} p_{u,i}^I p_{u,j}^I$ are independent of each other, which means we could achieve a significant speed-up by precomputing the two terms:

$$\mathcal{L}_I^{\mathcal{A}}(\Theta) = \sum_{i=1}^{d} \sum_{j=1}^{d} \left((h_{I,i} h_{I,j}) \left(\sum_{u\in\mathbf{B}} p_{u,i}^I p_{u,j}^I \right) \left(\sum_{v\in\mathbf{V}} c_v^{I-} q_{v,i} q_{v,j} \right) \right) \tag{12}$$

The rearrangement of nested sums in Eq.(12) is the key transformation that allows the fast optimization. The computing complexity of $\mathcal{L}_I^{\mathcal{A}}(\Theta)$ has been reduced from $O(|\mathbf{B}||\mathbf{V}|d)$ to $O((|\mathbf{B}| + |\mathbf{V}|)d^2)$.

By substituting Eq.(12) in Eq.(9) and removing the *const* part, we get the final efficient whole-data based loss of the item domain as follows:

$$\begin{aligned} \tilde{\mathcal{L}}_I(\Theta) = &\sum_{i=1}^{d} \sum_{j=1}^{d} \left((h_{I,i} h_{I,j}) \left(\sum_{u\in\mathbf{B}} p_{u,i}^I p_{u,j}^I \right) \left(\sum_{v\in\mathbf{V}} c_v^{I-} q_{v,i} q_{v,j} \right) \right) \\ &+ \sum_{u\in\mathbf{B}} \sum_{v\in\mathbf{V}^+} \left((1 - c_v^{I-}) \hat{R}_{uv}^2 - 2\hat{R}_{uv} \right) \end{aligned} \tag{13}$$

where c_{uv}^{I+} is set to 1 and c_{uv}^{I-} is simplified to c_v^{I-} as discussed before.

4.3.3 ***Joint Learning.*** Similarly, we can derive the loss function of social domain:

$$\begin{aligned} \tilde{\mathcal{L}}_S(\Theta) = &\sum_{i=1}^{d} \sum_{j=1}^{d} \left((h_{S,i} h_{S,j}) \left(\sum_{u\in\mathbf{B}} p_{u,i}^S p_{u,j}^S \right) \left(\sum_{t\in\mathbf{U}} c_t^{S-} g_{t,i} g_{t,j} \right) \right) \\ &+ \sum_{u\in\mathbf{B}} \sum_{t\in\mathbf{U}^+} \left((1 - c_t^{S-}) \hat{X}_{ut}^2 - 2\hat{X}_{ut} \right) \end{aligned} \tag{14}$$

After that, we integrate both the subtasks of item domain and social domain into a unified multi-task learning framework whose objective function is:

$$\mathcal{L}(\Theta) = \tilde{\mathcal{L}}_I(\Theta) + \mu \tilde{\mathcal{L}}_S(\Theta) \tag{15}$$

where $\tilde{\mathcal{L}}_{\mathcal{I}}(\Theta)$ is the item domain loss from Eq.(13), $\tilde{\mathcal{L}}_{\mathcal{S}}(\Theta)$ is the social domain loss from Eq.(14), and μ is the parameter to adjust the weight proportion of each term. The whole framework can be efficiently trained using existing optimizers in an end-to-end manner.

4.3.4 ***Discussion.*** So far we have derived the efficient whole-data based learning method. Note that the method is not limited to optimize recommendation models. It has the potential to benefit many other tasks where only positive data is observed, such as word embedding [21] and multi-label classification [35].

To analyze the time complexity of our optimization method, we exclude the time overhead of adaptive transfer learning in the model. In Eq.(15), updating a batch of users in item domain takes $O((|\mathbf{B}| + |\mathbf{V}|)d^2 + |\mathcal{R}_{\mathbf{B}}|d)$ time, where $\mathcal{R}_{\mathbf{B}}$ denotes positive item interactions of this batch of users. Similarly, in social domain it takes $O((|\mathbf{B}| + |\mathbf{U}|)d^2 + |\mathcal{X}_{\mathbf{B}}|d)$ time. Thus, one batch takes total $O((2|\mathbf{B}| + |\mathbf{U}| + |\mathbf{V}|)d^2 + (|\mathcal{R}_{\mathbf{B}}| + |\mathcal{X}_{\mathbf{B}}|)d)$ time. For the original regression loss, it takes $O((|\mathbf{B}||\mathbf{V}| + |\mathbf{B}||\mathbf{U}|)d)$ time. Since $|\mathcal{R}_{\mathbf{B}}| \ll |\mathbf{B}||\mathbf{V}|$, $|\mathcal{X}_{\mathbf{B}}| \ll |\mathbf{B}||\mathbf{U}|$, and $d \ll |\mathbf{B}|$ in practice, the computional complexity of our optimization method is reduced by several magnitudes. This makes it possible to apply whole-data based optimization strategy for neural models. Moreover, since no approximation is introduced during the derivation process, the optimization results are exactly the same with the original whole-data based regression loss.

As fast whole-data based learning is a challenging problem, our current efficient optimization method is still preliminary and has a limitation. It is not suitable for models with non-linear prediction layers, because the rearrange operation in Eq.(11) requires the prediction of $\hat{R}_{uv}$ to be linear. We leave the extension of the method as future work.

4.4 Model Training

To optimize the objective function, we adopt mini-batch Adagrad [8] as the optimizer. Its main advantage is that the learning rate can be self-adapted during the training phase, which eases the pain of choosing a proper learning rate. Specifically, users are first divided into multiple batches. Then, for each batch of users, all positive interactions in both item domain and social domain are utilized to form the training instances.

Dropout is an effective solution to prevent deep neural networks from overfitting [32], which randomly drops part of neurons during training. In this work, we employ dropout to improve our model's generalization ability. Specifically, after transferring, we randomly drop ρ percent of $\mathbf{p}_u^I$ and $\mathbf{p}_u^S$,where ρ is the dropout ratio.

5 EXPERIMENTS

5.1 Experimental Settings

5.1.1 ***Datasets.*** We experimented with three public accessible datasets: ***Ciao***[1], ***Epinion***[2] and ***Flixster***[3]. The three datasets are widely used in previous studies [4, 18, 33]. Each dataset contains users' ratings to the items they have purchased and the social connections between users. Among all the benchmark datasets, Flixster is the largest one and contains more than seven million item interactions from about seventy thousand users.

[1]http://www.jiliang.xyz/trust.html
[2]http://alchemy.cs.washington.edu/data/epinions/
[3]http://www.cs.ubc.ca/jamalim/datasets/

Table 2: Statistical details of the evaluation datasets. "Item Interaction" means user-item historical records, and "Social Connection" denotes user relationships in social networks.

	Ciao	*Epinion*	*Flixster*
#User	7,267	20,608	69,251
#Item	11,211	23,585	17,318
#Item Interaction	157,995	454,002	7,940,096
#Social Connection	111,781	351,486	967,195

Table 3: Comparison of the methods

Characteristics	BPR	ExpoMF	NCF	SBPR	TranSIV	SAMN	EATNN
Item domain	√	√	√	√	√	√	√
Social domain	\	\	\	√	√	√	√
Neural model	\	\	√	\	\	√	√
Adaptive transfer	\	\	\	\	\	\	√
Whole-data	\	√	\	\	√	\	√

All the datasets are preprocessed to make sure that all items have at least five interactions. As long as there exist some user–user or user–item interactions, the corresponding rating is assigned a value of 1 as implicit feedback. The statistical details of these datasets are summarized in Table 2.

5.1.2 ***Baselines.*** To evaluate the performance of Top-K recommendation, we compare our EATNN with the following methods.

- **Bayesian Personalized Ranking (BPR)** [28]: This method optimizes MF with the Bayesian Personalized Ranking objective function.
- **Exposure MF (ExpoMF)** [17]: This is a whole-data based method for item recommendation. It treats all missing interactions as negative and weighs them by item popularity.
- **Neural Collaborative Filtering (NCF)** [10]: This is the state-of-the-art deep learning method which uses users' historical feedback for item ranking. It combines MF with a multilayer perceptron (MLP) model .
- **Social Bayesian Personalized Ranking (SBPR)** [43]: This is a ranking model which assumes that users tend to assign higher scores to items that their friends prefer.
- **Transfer Model with Social and Item Visibilities (TranSIV)** [18]: This is a state-of-the-art social-aware recommendation method. It considers the visibility of both items and friend relationships, and utilizes transfer learning to combine the item domain and social domain for recommendation.
- **Social Attentional Memory Network (SAMN)** [4]: SAMN is a state-of-the-art deep learning method, which leverages attention mechanisms to model both aspect- and friend-level differences for social-aware recommendation.

The comparison of EATNN and the baseline methods are listed in Table 3.

5.1.3 ***Evaluation Metrics.*** We adopt *Recall@K* and *NDCG@K* to evaluate the performance of all methods. The two metrics have been widely used in previous recommendation studies [4, 18, 40]. *Recall@K* considers whether the ground truth is ranked among the top K items, while *NDCG@K* is a position-aware ranking metric.

5.1.4 ***Experiments Details.*** We randomly split each dataset into training (80%), validation (10%), and test (10%) sets. The parameters for all baseline methods were initialized as in the corresponding papers, and were then carefully tuned to achieve optimal performances. The learning rate for all models were tuned amongst [0.005,

Table 4: Comparisons of different methods on Three datasets. Best baselines are underlined. The proposed method achieves best performances on all metrics which are in boldface. ** denotes the statistical significance for $p < 0.01$, compared to the best baseline. The last column "RI" indicates the relative improvement of EATNN over the corresponding baseline on average.

Ciao	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
BPR	0.0591	0.1600	0.2135	0.0409	0.0688	0.0805	+20.08%
ExpoMF	0.0642	0.1556	0.2050	0.0445	0.0706	0.0816	+17.03%
NCF	0.0667	0.1584	0.2141	0.0456	0.0718	0.0837	+13.84%
SBPR	0.0623	0.1631	0.2146	0.0436	0.0695	0.0832	+16.30%
TranSIV	0.0678	0.1651	0.2184	0.0473	0.0753	0.0865	+10.20%
SAMN	0.0719	0.1671	0.2233	0.0495	0.0768	0.0883	+6.97%
EATNN	**0.0778****	**0.1764****	**0.2305****	**0.0547****	**0.0824****	**0.0943****	–
Epinion	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
BPR	0.0528	0.1477	0.2115	0.0353	0.0613	0.0751	+21.49%
ExpoMF	0.0611	0.1508	0.2077	0.0422	0.0673	0.0798	+11.82%
NCF	0.0535	0.1489	0.2144	0.0367	0.0624	0.0772	+19.06%
SBPR	0.0547	0.1511	0.2142	0.0387	0.0665	0.0783	+15.71%
TranSIV	0.0631	0.1552	0.2227	0.0423	0.0681	0.0829	+8.49%
SAMN	0.0621	0.1583	0.2274	0.0417	0.0698	0.0842	+7.62%
EATNN	**0.0696****	**0.1675****	**0.2309****	**0.0474****	**0.0749****	**0.0887****	–
Flixster	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
BPR	0.1733	0.3945	0.5272	0.1612	0.2193	0.2568	+35.88%
ExpoMF	0.2596	0.4488	0.5659	0.2012	0.2633	0.3002	+10.94%
NCF	0.2613	0.4564	0.5632	0.2112	0.2687	0.3075	+8.81%
SBPR	0.2314	0.4517	0.5697	0.1989	0.2514	0.3016	+14.05%
TranSIV	0.2748	0.4633	0.5749	0.2277	0.2804	0.3224	+4.35%
SAMN	0.2767	0.4661	0.5746	0.2316	0.2833	0.3251	+3.51%
EATNN	**0.2948****	**0.4736****	**0.5896****	**0.2401****	**0.2962****	**0.3319****	–

0.01, 0.02, 0.05]. To prevent overfitting, we tuned the dropout ratio in [0.1, 0.3, 0.5, 0.7, 0.9]. The batch size was tested in [128, 256, 512,1024], the dimension of attention network k and the latent factor number d were tested in [32, 64, 128]. After the tuning process, the batch size was set to 512, the size of the latent factor dimension d was set to 64. For our EATNN model, the attention size k was set to 32, the learning rate was set to 0.05, and the dropout ratio ρ was set to 0.3 for Ciao and Epinion, and 0.7 for Flixster. For the optimization objective, we set the weight parameter μ=0.1.

5.2 Comparative Analyses on Overall Performances

The results of the comparison of different methods on three datasets are shown in Table 4. To evaluate on different recommendation lengths, we set the length K = 10, 50, and 100 in our experiments. From the results, the following observations can be made:

First, methods incorporating social information generally perform better than non-social methods. For example, in Table 4, the performance of SBPR is better than BPR, and TranSIV, SAMN, and EATNN outperform BPR, ExpoMF, and NCF. This is consistent with previous work [4, 18, 43], which indicates that social information reflects users' interests, and hence is helpful in the recommendation.

Second, our method EATNN achieves the best performance on the three datasets, and significantly outperforms all baseline methods (including neural models NCF and SAMN) with p-values smaller than 0.01. Specifically, compared to SAMN – a recently proposed and very expressive deep learning model, EATNN exhibits average improvements of 6.97%, 7.62% and 3.51% on the three datasets. The substantial improvement of our model over the baselines could be attributed to two reasons: (1) our model uses attention mechanisms to adaptively transfer the common knowledge between item domain and social domain, which allows the social information to be modeled with a finer granularity; (2) the parameters in our model are jointly optimized on the whole data, while sample-based methods (BPR, NCF, SAMN) only use a fraction of sampled data and may ignore important negative examples.

Third, considering the performance on each dataset, we find the improvements of EATNN depend on the sparsity of the item domain data. The Flixster dataset is relatively dense in terms of user–item interactions (averaging 114.66 interactions per user, compared with 21.74 and 22.03 for Ciao and Epinions, respectively). User preferences are more difficult to learn from sparse user–item interactions (cold-start data), but can be enriched by the knowledge learnt from social domain. Thus, the results show that our transfer learning based model is more useful on sparse datasets. To make further verifications, we conduct experiments on less training data and the results are shown in Section 5.4.

5.3 Efficiency Analyses

In this section, we conducted experiments to explore the training efficiencies of our EATNN and two state-of-the-art social-aware recommendation methods: TranSIV and SAMN.

We first compared the overall runtime of the three methods. The results of EATNN-E is also added to show the efficiency of our proposed optimization method, where EATNN-E represents the variant model of EATNN using the original regression loss (Eq.2). In our experiments, the traditional method TranSIV was trained with 8 threads on the Intel Xeon 8-Core CPU of 2.4 GHz, while the neural models SAMN, EATNN-E and EATNN were trained on a

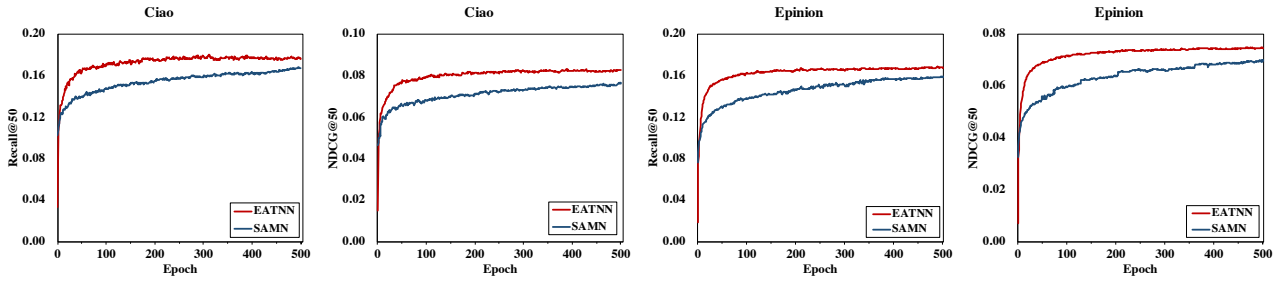

Figure 3: Performance curves of SAMN and EATNN on Ciao and Epinion datasets.

Table 5: Comparisons of runtime (second/minute/hour/day [s/m/h/d]). "S", "I", and "T" represents the training time for a single iteration, the number of iterations to converge, and the total training time, respectively.

Model	*Ciao*			*Epinion*			*Flixster*		
	S	I	T	S	I	T	S	I	T
TranSIV	55s	50	46m	410s	50	342m	37m	50	31h
SAMN	31s	500	258m	92s	500	767m	56m	200	8d
EATNN-E	13s	200	43m	97s	200	324m	32m	200	5d
EATNN	**1.8s**	200	**6m**	**11s**	200	**37m**	**8m**	200	**27h**

single NVIDIA GeForce GTX TITAN X GPU. The runtime results are shown in Table 5. We can first observe that the training time cost of EATNN is much less than EATNN-E, which certifies that our derived loss can be learned more efficiently compared to the original regression loss. Second, generally the training of EATNN is much faster than TranSIV, SAMN and EATNN-E. In particular, for the biggest dataset Flixster, EATNN only needs 27 hours to achieve the optimal performance, while SAMN and EATNN-E need about 8 and 5 days, respectively. For other datasets, the results of EATNN are also remarkable. In real E-commerce scenarios, the cost of training time is also an important factor to be considered. Our EATNN model shows significant advantages in training efficiency, which makes it more practical in real life.

We also investigated the training process of the neural models SAMN and our EATNN (The results of EATNN-E and EATNN are exactly the same). Figure 3 shows the prediction accuracy of the two models with respect to different training epochs. Due to the space limitation, we only show the results of Ciao and Epinion datasets on Recall@50 and NDCG@50 metrics. For Flixster dataset and other metrics, the observations are similar. From the figure, we can see that EATNN converges much faster than SAMN and consistently achieves better performance. The reason is that EATNN is optimized with a newly derived whole-data based method, while SAMN is based on negative sampling, which can be sub-optimal.

5.4 Handling Cold-Start Issue

We validated the ability of our model in handling the cold-start problem, where users have few interactions in item domain. Specifically, the experiments were conducted by using different proportions of the training data, including: 1) 25% for training, 75% for testing and 2) 50% for training, 50% for testing. All of the social information is used in the social-aware algorithms (SBPR, TranSIV, SAMN, and EATNN). The results are similar for all the three datasets. Due to the space limitation, we show the results of Epinion dataset in Table 6. Note that a larger test set contains more positive examples, which may lead to bigger values of Recall and NDCG compared to Table 4 where only 10% data is used for testing [18]. From Table 6, we have the following observations:

Firstly, compared with non-social methods, social-aware methods show much better performances, and more improvements are achieved when fewer data are used for training. Considering that social domain and item domain are correlated, the knowledge learnt from social behavior can compensate for the shortage of user feedback on items. As a result, the use of social information produces great improvement when the training data are scarce. Secondly, our EATNN demonstrates significant improvements over other baselines including social-aware methods SBPR, TranSIV, and SAMN. Specifically, the improvements over the best baseline are 8.68% for 25% training and 7.81% for 50% training. This indicates the effectiveness of EATNN in addressing the cold-start issue by leveraging adaptive transfer learning and users' social information. Thirdly, TranSIV, and EATNN generally achieve greater improvements when the training data are scarce. This observation coincides with previous work [16, 18], which states that transfer learning methods contribute even more when data in the target domain is sparse.

5.5 Ablation Study

To further understand the effectiveness of social information and the designed attention-based adaptive transfer learning framework, we conducted experiments with the following variants of EATNN:

- **EATNN-S:** A variant model of EATNN without using social information.
- **EATNN-A:** A variant model of EATNN in which the transfer framework is not adaptive. A constant weight (0.5 in our experiments) is assigned to the shared knowledge between item and social domains for every user.

Figure 4 shows the performance of different variants. The results of the state-of-art methods NCF (non-social) and SAMN (social-aware) are shown as baselines. Due to the space limitation, we also only show the results of Ciao and Epinion datasets on Recall@50 and NDCG@50 metrics. From Figure 4, two observations are made:

1) When using social information, EATNN and the variant EATNN-A both perform better than SAMN (p<0.01). And when the attention-based adaptive transfer framework is applied, the performances are further improved significantly compared with the constant weight method EATNN-A (p<0.05). It indicates that the shared knowledge between the item and social domains are varied and should be adaptively transferred for different users. The better results of EATNN also show that our attention-based adaptive transfer framework can effectively learn the weight of the shared knowledge.

Table 6: Performance comparisons on *Epinion* in cold scenarios (training : test = 25% : 75%, and 50% : 50%). ** : $p < 0.01$ compared to the best baseline. "RI" (last column) : the relative improvement of EATNN over the corresponding baseline on average.

25%	Recall@10	Recall@50	Recall@100	NDCG@10	NDCG@50	NDCG@100	RI
BPR	0.0211	0.0656	0.1009	0.0394	0.0501	0.0625	+61.84%
ExpoMF	0.0466	0.0752	0.1068	0.0497	0.0584	0.0699	+22.13%
NCF	0.0440	0.0781	0.1162	0.0449	0.0570	0.0709	+23.22%
SBPR	0.0387	0.0784	0.1157	0.0423	0.0543	0.0691	+29.07%
TranSIV	0.0519	0.0859	0.1211	0.0549	0.0657	0.0785	+8.68%
SAMN	0.0494	0.0832	0.1197	0.0517	0.0604	0.0731	+14.43%
EATNN	**0.0567****	**0.0934****	**0.1328****	**0.0593****	**0.0708****	**0.0853****	–
50%	**Recall@10**	**Recall@50**	**Recall@100**	**NDCG@10**	**NDCG@50**	**NDCG@100**	**RI**
BPR	0.0408	0.1143	0.1680	0.0525	0.0743	0.0912	+31.03%
ExpoMF	0.0611	0.1223	0.1701	0.0606	0.0827	0.0982	+13.30%
NCF	0.0576	0.1216	0.1745	0.0559	0.0784	0.0955	+17.30%
SBPR	0.0433	0.1257	0.1742	0.0554	0.0798	0.0974	+23.00%
TranSIV	0.0648	0.1284	0.1797	0.0633	0.0856	0.1010	+8.34%
SAMN	0.0642	0.1301	0.1805	0.0630	0.0867	0.1024	+7.81%
EATNN	**0.0712****	**0.1352****	**0.1909****	**0.0709****	**0.0921****	**0.1101****	–

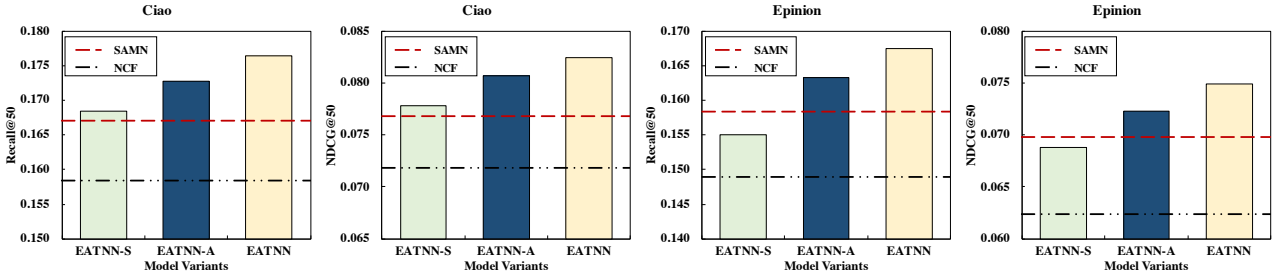

Figure 4: Performance of variants of EATNN on Ciao and Epinion datasets. EATNN-S is EATNN without social information, EATNN-A is EATNN without adaptive transfer (using constant weight for transfer). The two dotted lines represent the results of SAMN (social-aware) and NCF (non-social) respectively, which are added as baselines.

2) EATNN-S performs the worst among the variant models since no social interactions are utilized. Nevertheless, when using only item interactions, EATNN-S still significantly outperforms NCF (p<0.01), indicating the effectiveness of whole-data based learning.

5.6 Case Study on Adaptive Transfer

The attention weights reflect how the model learns and recommends. We provide some examples to show the adaptive transfer learning process when making recommendations. Table 7 shows some samples in different scenarios from Epinion dataset. EATNN learns the difference between these two domains and automatically balances the shared and non-shared parameters. The first user in EX1 has rich interactions in both item and social domains, and attention weights reflect how the shared knowledge is transferred. 66.3% of $\mathbf{u}^C$ is used to predict user preferences to items, while 77.4% of $\mathbf{u}^C$ is to predict user preferences to other users in social networks. EX2 is an example that the user has no item interactions in training set, whose attention weight of item-specific vector is around 0. Note that in EX2, both $\mathbf{u}^C$ and $\mathbf{u}^S$ are trained only on social data after random initialization, thus $\beta_{(C,u)}$ and $\beta_{(S,u)}$ exhibit similar weights. It is shown that attention weights also reflect the richness of feedback information. Opposite case is shown in EX3 which has no social interactions in training set, where $\alpha_{(C,u)}$ and $\alpha_{(I,u)}$ are similar. EX4 shows that the item feedback information is not rich enough to dominate the recommendation.

Table 7: Examples of attention weights on Epinion dataset. Each example is a user with the number of interactions he/she has in item domain and social domain.

	#Interaction in Training		Item Domain Attention		Social Domain Attention	
	Item	Social	$\alpha_{(C,u)}$	$\alpha_{(I,u)}$	$\beta_{(C,u)}$	$\beta_{(S,u)}$
EX1	72	85	0.663	0.337	0.774	0.226
EX2	0	29	0.914	0.086	0.479	0.521
EX3	34	0	0.422	0.578	0.882	0.118
EX4	1	15	0.823	0.177	0.511	0.489

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel Efficient Adaptive Transfer Neural Network (EATNN) for social-aware recommendation. Specifically, by introducing attention mechanisms, EATNN is able to adaptively assign a personalized scheme to transfer the shared knowledge between item domain and social domain. We also derive an efficient whole-data based optimization method, whose complexity is reduced significantly. Extensive experiments have been made on three real-life datasets. The proposed EATNN consistently and significantly outperforms the state-of-the-art recommendation models on different evaluation metrics, especially for cold-start users that

have few item interactions. Moreover, EATNN shows significant advantages in training efficiency, which makes it more practical to be applied in real E-commerce scenarios.

Our efficient whole-data based strategy has the potential to benefit many other tasks where only positive data is observed. The EATNN model is also not limited to the task in this paper. In the future, we are interested in exploring EATNN and our efficient whole-data based strategy in other related tasks like content recommendation [6], network embedding [21, 30], and multi-domain classification [24]. Also, we will try to extend our optimization method to make it suitable for learning deep non-linear models.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by Natural Science Foundation of China (Grant No. 61672311, 61532011) and the National Key Research and Development Program of China (2018YFC0831900).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[2] Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. 2010. Adaptive Transfer Learning.. In *Proceedings of AAAI*, Vol. 2. 7.
[3] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *Proceedings of WWW*. 1583–1592.
[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social Attentional Memory Network: Modeling Aspect-and Friend-level Differences in Recommendation. In *Proceedings of WSDM*.
[5] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. 2016. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. *arXiv preprint arXiv:1611.05594* (2016).
[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of Recsys*. 191–198.
[7] Robin Devooght, Nicolas Kourtellis, and Amin Mantrach. 2015. Dynamic matrix factorization with priors on unknown values. In *Proceedings of SIGKDD*. 189–198.
[8] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.
[9] Eric Eaton and Marie desJardins. 2011. Selective Transfer Between Learning Tasks Using Task-Based Boosting.. In *Proceedings of AAAI*.
[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of WWW*. 173–182.
[11] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of SIGIR*. 549–558.
[12] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of ICDM*. 263–272.
[13] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of SIGKDD*. 426–434.
[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.
[15] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of WSDM*. 173–182.
[16] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th annual international conference on machine learning*. 617–624.
[17] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of WWW*. 951–961.
[18] Xiao Lin, Zhang Min, Zhang Yongfeng, Liu Yiqun, and Ma Shaoping. 2017. Learning and transferring social and item visibilities for personalized recommendation. In *Proceedings of CIKM*. 337–346.
[19] Zhongqi Lu, Erheng Zhong, Lili Zhao, Evan Wei Xiang, Weike Pan, and Qiang Yang. 2013. Selective transfer learning for cross domain recommendation. In *Proceedings of ICDM*. 641–649.
[20] Hao Ma. 2014. On measuring social friend interest similarities in recommender systems. In *Proceedings of SIGIR*. 465–474.
[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*. 3111–3119.
[22] Seungwhan Moon and Jaime G Carbonell. 2017. Completely Heterogeneous Transfer Learning with Attention-What And What Not To Transfer.. In *Proceedings of IJCAI*.
[23] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of ICML*. 807–814.
[24] Hyeonseob Nam and Bohyung Han. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of CVPR*. 4293–4302.
[25] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction.. In *AAAI*, Vol. 10. 230–235.
[26] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. 2010. Fast als-based matrix factorization for explicit and implicit feedback datasets. In *Proceedings of Recsys*. 71–78.
[27] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of WSDM*. 485–494.
[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI*. 452–461.
[29] Suman Deb Roy, Tao Mei, Wenjun Zeng, and Shipeng Li. 2013. Towards cross-domain learning for social video popularity prediction. *IEEE Transactions on multimedia* 15, 6 (2013), 1255–1267.
[30] Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. 2018. Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. In *Proceedings of SIGKDD*. 2190–2199.
[31] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of SIGKDD*. 650–658.
[32] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
[33] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive Recurrent Social Recommendation. In *Proceedings of SIGIR*. 185–194.
[34] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In *Proceedings of WWW*. 729–739.
[35] Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining* 3, 3 (2007), 1–13.
[36] Maksims Volkovs and Guang Wei Yu. 2015. Effective latent models for binary feedback in recommender systems. In *Proceedings of SIGIR*. ACM, 313–322.
[37] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road: Recommending items from information domains to social users. In *Proceedings of SIGIR*. 185–194.
[38] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).
[39] Xin Xin, Fajie Yuan, Xiangnan He, and Joemon M Jose. 2018. Batch IS NOT Heavy: LearningWord Representations From All Samples. (2018).
[40] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based clothing recommendation. In *Proceedings of WWW*. 649–658.
[41] Fajie Yuan, Xin Xin, Xiangnan He, Guibing Guo, Weinan Zhang, Chua Tat-Seng, and Joemon M Jose. 2018. fbgd: Learning embeddings from positive unlabeled data with bgd. (2018).
[42] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. 2013. Domain adaptation under target and conditional shift. In *Proceedings of ICML*. 819–827.
[43] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of CIKM*. 261–270.
[44] Erheng Zhong, Wei Fan, and Qiang Yang. 2014. User behavior learning and transfer in composite social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8, 1 (2014), 6.
[45] Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. 2011. Heterogeneous Transfer Learning for Image Classification.. In *Proceedings of AAAI*.