Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention

Jingyuan Chen
National University of Singapore
jingyuanchen91@gmail.com

Hanwang Zhang
Columbia University
hanwangzhang@gmail.com

Xiangnan He*
National University of Singapore
xiangnanhe@gmail.com

Liqiang Nie
ShanDong University
nieliqiang@gmail.com

Wei Liu
Tencent AI Lab
wliu@ee.columbia.edu

Tat-Seng Chua
National University of Singapore
dcscts@nus.edu.sg

ABSTRACT

Multimedia content is dominating today's Web information. The nature of multimedia user-item interactions is 1/0 binary implicit feedback (*e.g.*, photo likes, video views, song downloads, *etc.*), which can be collected at a larger scale with a much lower cost than explicit feedback (*e.g.*, product ratings). However, the majority of existing collaborative filtering (CF) systems are not well-designed for multimedia recommendation, since they ignore the implicitness in users' interactions with multimedia content. We argue that, in multimedia recommendation, there exists *item-* and *component-level* implicitness which blurs the underlying users' preferences. The item-level implicitness means that users' preferences on items (*e.g.*, photos, videos, songs, *etc.*) are unknown, while the component-level implicitness means that inside each item users' preferences on different components (*e.g.*, regions in an image, frames of a video, *etc.*) are unknown. For example, a "view" on a video does not provide any specific information about how the user likes the video (*i.e.*, item-level) and which parts of the video the user is interested in (*i.e.*, component-level). In this paper, we introduce a novel *attention* mechanism in CF to address the challenging item- and component-level implicit feedback in multimedia recommendation, dubbed Attentive Collaborative Filtering (ACF). Specifically, our attention model is a neural network that consists of two attention modules: the component-level attention module, starting from any content feature extraction network (*e.g.*, CNN for images/videos), which learns to select informative components of multimedia items, and the item-level attention module, which learns to score the item preferences. ACF can be seamlessly incorporated into classic CF models with implicit feedback, such as BPR and SVD++, and efficiently trained using SGD. Through extensive experiments on two real-world multimedia Web services: Vine and Pinterest, we show that ACF significantly outperforms state-of-the-art CF methods.

CCS CONCEPTS

•Information systems →Multimedia information systems;

KEYWORDS

Collaborative Filtering, Implicit Feedback, Attention, Multimedia Recommendation

1 INTRODUCTION

As we log into a multimedia Web service, *e.g.*, Youtube, just like other billions of users, we have billions of contents online ready to view and share. Meanwhile, due to the advance of mobile devices, millions of new images and videos are streaming into these websites. Take Snapchat, one of the most popular video-based social App, as an example. During the time of reading this paragraph, around 50 thousand video snippets are shared and 2.4 million videos are viewed. Without a doubt, the dominating Web multimedia content requires modern recommender systems, in particular those based on Collaborative Filtering (CF), to sift through massive multimedia contents for users in a highly dynamic environment.

CF analyzes relationships between users and interdependencies among items, in order to identify new user-*item* associations [21, 23, 37]. In the context of multimedia recommendation, *item* refers to different kinds of multimedia contents consumed by users, such as a video, a photo or a song. Most CF systems rely on explicit user interests as input, *e.g.*, star ratings of products, which provide *explicit feedback* [19, 25, 38]. However, explicit ratings are not always available in many applications. Due to the large-scale and extreme diversity of multimedia contents [15], inherent user-item interactions in multimedia recommendation systems are mostly based on *implicit feedback*, such as "view" of a video, "like" of a photo, "play" of a song, *etc.* As implicit feedback lacks substantial evidence on which items user dislike (*i.e.*, negative feedback), existing CF methods [20, 21, 30] with implicit feedback generally focus on how to tap the missing user-item interactions into preference modeling. However, few methods deeply explore the implicitness of users' preferences. In particular, we argue that there are two levels of implicit feedback in multimedia recommendation, which have been neglected by most existing CF methods.

Item-Level Implicit Feedback. Each user is associated with a set of items (*i.e.*, positive feedback) via tracking their consumption habits. However, a positive set of user feedback does not necessarily indicate equal item preferences. This phenomenon is extremely

[*] Xiangnan He is the corresponding author.

SIGIR'17, August 7–11, 2017, Shinjuku, Tokyo, Japan.
© 2017 ACM. 978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080797

prevailing in multimedia services as most of them are social-oriented. For example, some images clicked as "like" may be only due to the fact that they are taken by friends but are not of users' real interests. Even though for images consistent with users' real interests, users' preferences on them are not the same. Such cases that the preference information on each item is not provided are named as *item-level implicitness.* To better characterize users' preference profile, the implicit feedback in the item-level requires different attentions on the set of items. However, to the best of our knowledge, existing CF models generally resort to either a constant weight [23] or pre-defined heuristic weights [21], and thus the conventional neighborhood context obtained by such a weighted sum fails to model the item-level implicit feedback.

Component-Level Implicit Feedback. Feedback on multimedia content is typically at the whole item level. However, multimedia content usually contains diverse semantics and multiple components. We use *component-level implicitness* to denote cases that feedback for each component is not available. Take a video about a basketball match as an example, the whole video contains multiple players and abundant actions. A "play" feedback from a user on this video does not necessarily indicate that the user likes the whole content of the video, while it may be triggered by his interest in the last part of the video which is about the final scores in the match. Therefore, unlike traditional content-based CF methods that only consider multimedia content as a whole [4, 12], we should model user preferences with lower-level content components, *e.g.*, image features in different locations [39] and video features of various frames [6, 10, 41].

However, directly modeling the item-level and component-level implicit feedback to facilitate recommendation is non-trivial since the ground-truth for the implicitness in each level is not available. To address this problem, we propose a novel CF framework dubbed Attentive Collaborative Filtering (ACF) for multimedia recommendation, which can automatically assign weights to the two levels of feedback in a distant supervised manner. ACF draws on the latent factor model, by transforming both items and users to the same latent factor space to make them directly comparable. To incorporate the two levels of implicit feedback, a neighborhood-based model is integrated to characterize users' interest profile through their historical behavior which is a weighted sum of items. The influence of two levels feedbacks is reflected by the weights of items in the neighborhood model. Specifically, in order to model the item-level feedback, we propose a weighting function which is a multi-layer neural network and takes the characteristics of both user and item, as well as the multimedia content feature as input (cf. Section 4.2). The multimedia content feature in the item-level is actually generated by assembling multiple components of the item with attentive weights. In particular, the component-level attention is also a multi-layer neural network that takes user and component features as input. Then, all the attentive components together compose a content feature vector, which is one of the input of the item-level attention (cf. Section 4.3). ACF can be efficiently trained using Stochastic Gradient Decent (SGD) on large user-item interactions of both images and videos (cf. Section 4.4). We evaluate ACF extensively on two real-world datasets that represent a spectrum of different media: Pinterest (images) and Vine (videos). Experimental results show that ACF consistently outperforms competing methods ranging from CF-based methods, content-based methods [28, 35] and hybrid methods [9, 33] (cf. Section 5).

Our contributions are summarized as follows:

- We propose a novel CF framework named Attentive Collaborative Filtering (ACF) to employ attention modeling in CF with implicit feedback. To the best of our knowledge, this is the first framework that is designed to tackle the implicit feedback in multimedia recommendation.
- To address two levels of implicit feedback, we introduce two attention modules, each is a neural network that can be seamlessly incorporated into any neighborhood models with efficient end-to-end SGD training.
- Through extensive experiments conducted on two real-world datasets, we show that ACF consistently outperforms several state-of-the-art CF methods with implicit feedback.

2 RELATED WORK

2.1 Implicit Feedback

Recommendation with implicit feedback is also called the one-class problem [27] because of the lack of negative feedback, where only positive feedback (*e.g.*, click, view) is available. Apart from the positive feedback, the remaining data is a mixture of real negative feedback and missing values. Therefore, it is hard to reliably infer which item a user did not like from implicit feedback.

To cope with the problem of missing negative samples, several approaches have been proposed which can be roughly classified into two categories: sample based learning [20, 27, 30] and whole-data based learning [21, 23]. The former samples negative feedback from the missing data, while the latter treats all the missing data as negative. Therefore, sample-based approaches are more effective while whole-data based approaches provide higher coverage.

Traditional whole-data based methods assume that all unobserved events are negative samples and are equally weighted [23]. However, this may not be realistic, due to the fact that the unobserved data may contain missing values which are false negative. Towards this end, several recent efforts [21, 27] focus on the weighting scheme, taking the confidence whether the unobserved samples are indeed negative ones into consideration. For example, certain nonuniform weighting schemes on the negative samples, such as user-oriented [27] and item popularity-oriented [21], have been proposed and proven to be more effective than the uniform weighting scheme. However, one major limitation of the non-uniform weighting method is that the weighting schemes are defined based on assumptions proposed by the authors, which may not be correct in the real data.

As can be seen, most of the existing efforts till now focused on the negative feedback sampling or weighting schemes to tackle the problem of no negative feedback, while no much attention has been paid on the two levels of implicit feedback—item-level attention and component-level attention—which can be seen as the weighting strategy on positive samples. To fill up the empty in positive sample weighting, we propose a novel attention mechanism to weight positive implicit signal automatically based on the user-item interaction matrix and the content of the item.

2.2 Multimedia Recommendation

The significance of multimedia recommendation has led to the great attention from both the industry and academia [8, 16, 32, 36]. Most of the current state-of-the-art multimedia recommendation techniques are based on the CF analysis [2, 4]. Although these approaches work well for popular and frequently watched contents, they are less applicable to fresh contents or tail contents with few views, due to the sparsity of the data. Therefore, for these items, CF analysis based solely on user-item interaction matrix or co-views information may yield either low-quality suggestions or no suggestions at all. To address the problem of tail contents, researchers have developed hybrid approaches [33] that incorporate the context and content of multimedia items with the CF model for recommendation. For example, several efforts have been dedicated to conduct the video recommendation utilizing different context information, such as the multi-modal relevance [26, 34], cross-domain knowledge [11, 14] and latent attributes feature [12, 44]. Moreover, [3, 4] have proposed hybrid approaches to video recommendation, which combines the video content (topics mined from video metadata, related queries, *etc.*) with the co-view information. Another widely used strategy is using a latent factor model for recommendation, and further predicting the latent factors from multimedia contents to handle the cold start scenario [15, 33]. However, most of the exisitng methods failed to pay attention to the two levels of implicitness in the multimedia recommendation, which is the major concern of our work.

2.3 Attention Mechanism

Attention mechanism has been shown effective in various machine learning tasks such as image/video captioning [7, 39, 40] and machine translation [1]. Its success is mainly due to the reasonable assumption that human recognition does not tend to process a whole signal in its entirety at once; instead, one only focuses on selective parts of the whole perception space when and where as needed. Our component-level attention adopts the soft spatial attention model in [39] for images and the soft temporal attention model in [40] for videos. The key idea of soft attention is to learn to assign attentive weights (normalized by sum to 1) for a set of features: higher (lower) weights indicate that the corresponding features are informative (less informative) for the end task.

In fact, the attention assumption is reasonable in many real-world situations, not just in the domain of computer vision and natural language processing. To the best of our knowledge, ACF is the first attention-based CF model in the area of recommender systems.

3 PRELIMINARIES

We begin with some notations. We denote a user-item interaction matrix as $\mathbf{R} \in \mathbb{R}^{M \times N}$, where M and N denote the number of users and items, respectively. Specifically, we use R_{ij} to represent the (i, j)-th entry of $\mathbf{R}$. As for implicit feedback, $R_{ij} = 1$ indicates that the i-th user has interacted with the j-th item and $R_{ij} = 0$ indicates that there is no interaction between user i and item j in the observed data. We use $\mathcal{R} = \{(i, j) | R_{ij} = 1\}$ to denote the set of user-item pairs where there exist implicit interactions. The goal of a CF model with implicit feedback is to exploit the entire $\mathbf{R}$ to estimate $\hat{R}_{ij}$ for the unobserved interactions.

3.1 Latent Factor Models

Latent factor models map both users and items to a joint low-dimensional latent space where the user-item preference score is estimated by vector inner product. We will focus on models that are induced by Singular Value Decomposition (SVD) on the user-item ratings matrix. We denote user latent vectors as $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_M] \in \mathbb{R}^{D \times M}$ and item latent vectors as $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_N] \in \mathbb{R}^{D \times N}$, where $D \ll min(M, N)$ is the latent feature dimension. The preference score R_{ij} is estimated as:

$$\hat{R}_{ij} =< \mathbf{u}_i, \mathbf{v}_j >= \mathbf{u}_i^T \mathbf{v}_j. \tag{1}$$

The objective is to minimize the following regularized squared loss on observed ratings:

$$\arg\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j) \in \mathcal{R}} (R_{ij} - \hat{R}_{ij})^2 + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2), \tag{2}$$

where λ controls the strength of regularization, which is usually an L_2 norm to prevent overfitting. After we obtain the optimized user and item vectors, recommendation is then reduced to a ranking problem according to the estimated scores $\hat{R}_{ij}$.

However, applying SVD in implicit feedback domain raises difficulties due to the high portion of unobservable data. Carelessly treating the unobserved entries as negative samples in SVD may introduce false negative samples in the training data.

3.2 Bayesian Personalized Ranking (BPR)

BPR is a well-known framework for addressing the implicitness in CF [30]. Instead of point-wise learning as in SVD, BPR models a triplet of one user and two items, where one of the items is observed and the other one is not. Specifically, from the user-item matrix $\mathbf{R}$, if an item j has been viewed by user i, then it is assumed that the user prefers this item over all the other unobserved items.

The optimization objective for BPR is based on the maximum posterior estimator. In particular, by applying the above latent factor models, a widely used BPR model is given as:

$$\arg\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j,k) \in \mathcal{R}_B} - \ln \sigma(\hat{R}_{ij} - \hat{R}_{ik}) + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2), \tag{3}$$

where σ is the logistic sigmoid function and λ is regularization parameter. The training data $\mathcal{R}_B$ is generated as:

$$\mathcal{R}_B = \{(i, j, k) | j \in \mathcal{R}(i) \wedge k \in \mathcal{I} \backslash \mathcal{R}(i)\}, \tag{4}$$

where $\mathcal{I}$ denotes the set of all items in the dataset and $\mathcal{R}(i)$ represents the set of items that are interacted by the i-th user. The semantics of $(i, j, k) \in \mathcal{R}_B$ is that user i is assumed to prefer item j over k.

In this work, we use BPR as our basic learning model because of its effectiveness in exploiting the unobserved user-item feedback.

4 ATTENTIVE COLLABORATIVE FILTERING

In this section, we will introduce our Attentive Collaborative Filtering (ACF) model in detail. First, we present the general ACF framework, elaborating the motivation of the model. We then show the detailed formulations of the proposed item-level and component-level attentions. Note that in the following sections, "item-" means video or image, and "component-" means the frame in video or space region in images. Lastly we will go through the optimization details of ACF.

4.1 General Framework

ACF is a hierarchical neural network that models user's preference score with respect to the item in *item*-level and content in *component*-level. Given a user i, an item l and the m-th component in item l, we use $\alpha(i,l)$ to denote user i's preference degree in item l and further $\beta(i,l,m)$ to denote user i's preference degree in the m-th component of item l. We use two attention sub-networks to learn these two preference scores jointly. Specifically, we employ component-level module to generate content representations for each item and item-level module to obtain user representation.

Objective Function. In addition to explicitly parameterizing each user i with $\mathbf{u}_i$, ACF also models users based on the set of items $\mathcal{R}(i)$ that they interacted with. Therefore, each item l is associated with two factor vectors. One is denoted by $\mathbf{v}_l$, which is the basic item vector in latent factor model. The other one, denoted by $\mathbf{p}_l$, is the auxiliary item vector which is used to characterize users based on the set of items they interacted with. The representation of a user i is through the sum: $\mathbf{u}_i + \sum_{l \in \mathcal{R}(i)} \alpha(i,l)\mathbf{p}_l$.

ACF is optimized in the BPR pairwise learning objective [30]: optimizing the pairwise ranking between the positive and non-observable items:

$$\arg\min_{\mathbf{U},\mathbf{V},\mathbf{P},\mathbf{\Theta}} \sum_{(i,j,k)\in\mathcal{R}_B} -\ln\sigma\left\{\left(\mathbf{u}_i + \sum_{l\in\mathcal{R}(i)}\alpha(i,l)\mathbf{p}_l\right)^T\mathbf{v}_j - \left(\mathbf{u}_i + \sum_{l\in\mathcal{R}(i)}\alpha(i,l)\mathbf{p}_l\right)^T\mathbf{v}_k\right\} + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{P}||^2), \tag{5}$$

where set $\mathcal{R}(i)$ denotes the set of items that are interacted by the i-th user and $\mathbf{\Theta}$ is the parameters in attention network. $\alpha(i,l)$ is the item-level attention module, which measures the preference degree of user i to item l. Note that the component-level attention module is also integrated into $\alpha(i,l)$.

Inference. After we obtain the optimized user, item and auxiliary item vectors, *i.e.*, $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{P}$, as well as the parameters of the attention networks, recommendation is then reduced to a ranking problem among all the items in the dataset based on estimated score $\hat{R}_{ij}$:

$$\hat{R}_{ij} = \left(\mathbf{u}_i + \sum_{l\in\mathcal{R}(i)}\alpha(i,l)\mathbf{p}_l\right)^T\mathbf{v}_j. \tag{6}$$

Relations to Neighborhood Models. Note that if we rewrite Eqn. (6) as:

$$\hat{R}_{ij} = \overbrace{\mathbf{u}_i^T\mathbf{v}_j}^{latent\ factor\ model} + \underbrace{\sum_{l\in\mathcal{R}(i)}\alpha(i,l)\mathbf{p}_l^T\mathbf{v}_j}_{neighborhood\ model}, \tag{7}$$

where $\mathbf{p}_l^T\mathbf{v}_j$ can be viewed as the similarity measure function between items in the neighborhood-based collaborative filtering [24]. The first part of Eqn. (7) corresponds to the latent factor model and the second part corresponds to the neighborhood model. Specifically, if we replace the attention weight $\alpha(i,l)$ with a normalized weight $\frac{1}{|\mathcal{R}(i)|}$, our ACF model will degenerate into SVD++ [25]; or, the weight is a heuristic function, ACF is similar to FISM [24]. However, they failed to consider the two levels of implicit feedback in recommendation, where a fixed weight assumes that all the items contribute equally to the prediction. In fact, the weights should be highly dependent to the user and the item content as we will introduce in Section 4.2 and Section 4.3.

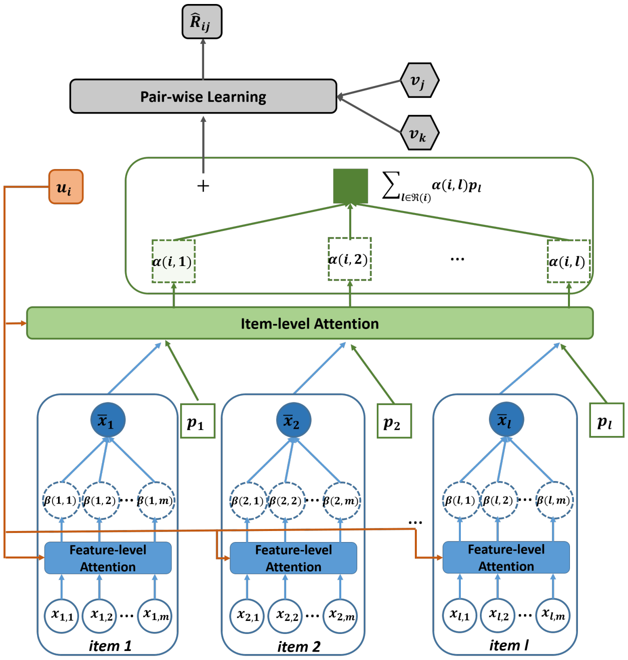

$\hat{R}_{ij}$
Pair-wise Learning
v_j
v_k
u_i
+
$\sum_{l\in\Re(i)} \alpha(i,l)p_l$
$\alpha(i,1)$
$\alpha(i,2)$
...
$\alpha(i,l)$
Item-level Attention
$\bar{x}_1$
p_1
$\bar{x}_2$
p_2
$\bar{x}_l$
p_l
$\beta(1,1)$ $\beta(1,2)$ ... $\beta(1,m)$
$\beta(2,1)$ $\beta(2,2)$ ... $\beta(2,m)$
$\beta(l,1)$ $\beta(l,2)$ ... $\beta(l,m)$
...
Feature-level Attention
Feature-level Attention
Feature-level Attention
$x_{1,1}$ $x_{1,2}$ ... $x_{1,m}$
$x_{2,1}$ $x_{2,2}$ ... $x_{2,m}$
$x_{l,1}$ $x_{l,2}$ ... $x_{l,m}$
item 1
item 2
item l

Figure 1: The architecture of our proposed Attentive Collaborative Filtering framework. Our attention model contains two level modules: component-level attention and item-level attention (cf. Section 4.1).

Figure 1 illustrates the workflow of ACF. We start from the set of items that are liked by the i-th user. First, for each item l, we access the set of component features $\{\mathbf{x}_{lm}\}$ (blue solid circles), where $\mathbf{x}_{lm}$ could be the image region feature at the m-th spatial location [39] or the frame feature of the m-th frame in a video [41]. Then, the component-level attention module, which is a sub-network, takes the user latent vector $\mathbf{u}_i$ and the feature $\mathbf{x}_{lm}$ as input and output the component-level attentive weight $\beta(l,m)$ for the m-th component (dashed blue circles). Thus, the final representation of the l-th item content $\bar{\mathbf{x}}_l$ is calculated by the weighted sum $\sum\beta(l,m)\mathbf{x}_{lm}$ (filled blue circles). After we have obtained $\bar{\mathbf{x}}_l$, we can use the item-level attention module by taking user latent vector $\mathbf{u}_i$, item latent vector $\mathbf{v}_l$, auxiliary item latent vector $\mathbf{p}_l$, and the content feature $\bar{\mathbf{x}}_l$ to calculate the item-level attentive weight $\alpha(i,l)$ for each neighborhood item (dashed green squares). Then, similar to the component-level attentions, we obtain the final neighborhood vectors for user i by the weighted sum $\sum\alpha(i,l)\mathbf{p}_l$ (filled green

Algorithm 1: Attentive Collaborative Filtering

```
Input: User-item interaction matrix $\mathbf{R}$. Each item $l$ is
    represented by a set of component features $\{x_{l*}\}$.
Output: Latent feature matrix $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$ and parameters in
    attention model $\Theta$
1:  Initialize $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{P}$ with Gaussian distribution. Initialize $\Theta$
    with xavier [17].
2:  repeat
3:    draw $(i, j, k)$ from $\mathcal{R}_B$
4:    For each item $l$ in $\mathcal{R}(i)$:
5:      For each component $m$ in $\{\mathbf{x}_{l*}\}$:
6:        Compute $\beta(i, l, m)$ according to Eqns. (10) and (11)
7:      Compute $\overline{\mathbf{x}}_l$ according to Eqn. (12)
8:    Compute $\alpha(i, l)$ according to Eqns. (8) and (9)
9:    $\mathbf{u}'_i \leftarrow \mathbf{u}_i + \sum_{l \in \mathcal{R}(i)} \alpha(i, l)\mathbf{p}_l$
10:   $\hat{R}_{ijk} \leftarrow \mathbf{u}'_i\mathbf{v}_j - \mathbf{u}'_i\mathbf{v}_k$
11:   For each parameter $\theta$ in $\{\mathbf{U}, \mathbf{V}, \mathbf{P}, \Theta\}$:
12:     Update $\theta \leftarrow \theta + \eta \cdot (\frac{\exp^{-\hat{R}_{ijk}}}{1+\exp^{-\hat{R}_{ijk}}} \cdot \frac{\partial \hat{R}_{ijk}}{\partial \theta} + \lambda \cdot \theta)$.
13: until convergence
14: return $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$ and $\Theta$.
```

squares). Lastly, combined with the basic user latent vector, we can use stochastic gradient descent to optimize the BPR pairwise learning objective (cf Eqn. (5)).

4.2 Item-Level Attention

The goal of the item-level attention is to select items that are representative to users' preferences and then aggregate the representation of informative items to characterize users. Given the basic user latent representation $\mathbf{u}_i$, the neighborhood item latent vector $\mathbf{v}_l$, the neighborhood auxiliary item vector $\mathbf{p}_l$, and the item content feature $\overline{\mathbf{x}}_l$ (detailed in the next section), we use a two-layer network to compute the attention score $a(i, l)$ as,

$$a(i, l) = \mathbf{w}_1^T \phi(\mathbf{W}_{1u}\mathbf{u}_i + \mathbf{W}_{1v}\mathbf{v}_l + \mathbf{W}_{1p}\mathbf{p}_l + \mathbf{W}_{1x}\overline{\mathbf{x}}_l + \mathbf{b}_1) + \mathbf{c}_1. \quad (8)$$

where the matrices $\mathbf{W}_{1*}$ and bias $\mathbf{b}_1$ are the first layer parameters, and the vector $\mathbf{w}_1$ and bias $\mathbf{c}_1$ are the second layer parameters; $\phi(x) = max(0, x)$ is the ReLU function, which was found better than a single layer perceptron with a hyperbolic tangent nonlinearity.

The final item-level weights are obtained by normalizing the above attentive scores using Softmax, which can be interpreted as the contribution of the item l to the preference profile of user i:

$$\alpha(i, l) = \frac{exp\,(a(i, l))}{\sum_{n \in \mathcal{R}(i)} exp\,(a(i, n))}. \quad (9)$$

4.3 Component-Level Attention

Multimedia items contain complex information while different users may like different parts of contents in the same multimedia item. Each multimedia item l may be encoded into a variable-sized set of component features $\{\mathbf{x}_{l*}\}$. We use $|\{\mathbf{x}_{l*}\}|$ to denote the size of the set and $\mathbf{x}_{lm}$ to denote the feature of the m-th component in the set. Unlike conventional content-based CF models that generally adopt average pooling [6, 33] for extracting a unified representation, the goal of component-level attention is to assign components attentive weights that are consistent with user preference, and then apply the weighted sum to construct the content representation.

Similar to the item-level attention, the component-level attention score for the m-th component $\mathbf{x}_{lm}$ of item l from user i is also a two-layer network:

$$b(i, l, m) = \mathbf{w}_2^T \phi(\mathbf{W}_{2u}\mathbf{u}_i + \mathbf{W}_{2x}\mathbf{x}_{lm} + \mathbf{b}_2) + \mathbf{c}_2, \quad (10)$$

where the matrices $\mathbf{W}_{2*}$ and bias $\mathbf{b}_2$ are the first layer parameters, and the vector $\mathbf{w}_2$ and bias $\mathbf{c}_2$ are the second layer parameters; $\phi(x) = max(0, x)$ is the ReLU function. Then, the final component-level attention is normalized as:

$$\beta(i, l, m) = \frac{exp\,(b(i, l, m))}{\sum_{n=1}^{|\{\mathbf{x}_{l*}\}|} exp\,(b(i, l, n))}. \quad (11)$$

After we obtain the component-level attention $\beta(i, l, m)$, the content representation of item l with the encoded preference of user i is calculated as the following weighted sum:

$$\overline{\mathbf{x}}_l = \sum_{m=1}^{|\{\mathbf{x}_{l*}\}|} \beta(i, l, m) \cdot \mathbf{x}_{lm}, \quad (12)$$

4.4 Algorithm

A stochastic gradient descent algorithm based on bootstrap sampling of training triples is proposed to solve the network. The steps for training the model are summarized in Algorithm 1.

For notational simplicity, we divide ACF into three steps: 1) subroutine ACFcomp runs from Line 5 to Line 7. Note that the component can be image region features for image or video frame feature for video; 2) subroutine ACFitem runs from Line 4 to Line 9; and 3) subroutine BPR-OPT runs back propagation with respect to Eqn. (5). Due to space limit, we use Θ to denote the set of parameters in item-level attention and component-level attention, and $\hat{R}_{ijk}$ to denote $\hat{R}_{ij} - \hat{R}_{ik}$. Note that Line 12 are the gradients of the model parameters updated using chain rules. To optimize the objective function, we employ stochastic gradient descent (SGD) — a universal solver for optimizing neural network models. At each time, it randomly selects a training instance and updates each model parameter towards the direction of its negative gradient.

Note that if more computational resources are available, we can also achieve end-to-end CNN module fine-tuning. We will investigate whether we can train more powerful visual features using user-item implicit feedback in future.

5 EXPERIMENTS

In this section, we will conduct experiments to answer the following research questions:

- **RQ1** Does ACF outperform state-of-the-art recommendation methods?
- **RQ2** How do the proposed item-level and component-level attentions perform?

We will first present the experimental settings, follow by answering the above two research questions, and end with some illustrative examples.

5.1 Experimental Settings

Datasets. We experimented with two publicly accessible datasets: Pinterest[1] and Vine [5]. The characteristics of the two datasets are summarized in Table 1.

Table 1: Statistics of the evaluation datasets.

Dataset	Interaction#	Item#	User#	Sparsity
Pinterest	1,091,733	14,965	50,000	99.85%
Vine	125,089	16,243	18,017	99.96%

1. Pinterest. This implicit feedback dataset is constructed by [15] for evaluating image recommendation. Due to the large volume and high sparsity of this dataset, for instance, over 20% of users have only one pin, we filter the dataset by retaining the top 15, 000 popular images and sampling 50, 000 users who have interactions on the 15, 000 images. This results in a subset of data that contains 50, 000 users, 14, 965 images and 1, 091, 733 interactions. Each interaction denotes whether the user has pinned the image to his/her own board.

2. Vine. This video dataset [6, 45] is crawled from Vine, a micro-video sharing social network. The crawling starts with a set of active users. Then the breadth-first crawling strategy is adopted to expand the seed users by crawling their followers. Totally, the dataset contains 98, 166 users and their interactions on 1, 303, 242 micro-videos. An interaction denotes whether the user has posted or re-posted the video. To evaluate the recommendation task, we filtered the dataset by retaining users with at least 4 interactions. This results in a subset of data that contains 18, 017 users, 16, 243 videos and 125, 089 interactions.

Evaluation Protocols. To evaluate the performance of item recommendation, we adopted the leave-one-out evaluation, which has been widely used in literature [21, 30]. For each user, we held-out his/her latest interaction as the test set and utilized the remaining data for training. As we mentioned in Section 3.1, the recommendation task is reduced to a ranking problem based on the estimated score. To assess the ranked list with the ground-truth item that user has actually consumed, we adopt Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [13], where HR measures whether the ground truth item is present on the ranked list and NDCG accounts for the position of hit [19]. We report the average score for all test users. If not specifically specified, we truncate the ranked list at 100 among all the items for both metrics.

Baselines. We compared ACF with the following methods. Note that all model-based CF models are learned by optimizing the same pairwise ranking loss of BPR (cf Eqn. (3)) for a fair comparison.

CF-based Methods:

- **UCF [46].** User-based collaborative filtering analyzes the user-item matrix to compute the similarities between users and then recommends items to people with similar tastes and preference.
- **ItemKNN [31].** This is a standard item-based CF for which we adopted Cosine similarity to measure the similarity among items and followed the setting of [23] to adapt it for implicit data.
- **BPR [30].** This method optimizes the latent factor model with a pairwise ranking loss, which is tailored to learn from implicit feedback. It is a highly competitive baseline for item recommendation, which is also the basic learning scheme of our model (cf. Section 3.2).
- **SVD++ [25].** SVD++ is a merged model of latent factor and neighborhood models, in which a second set of item factors is added, to model the item-item similarity, which is also a special case of our model when the item-level attention scheme is replaced by the average pooling.

Content-based Methods:

- **CBF [28].** Content-based filtering generates a user feature vector by averaging all the item features interacted with the user and then recommend items based on the similarity between the item features and the user features.

Hybrid Methods:

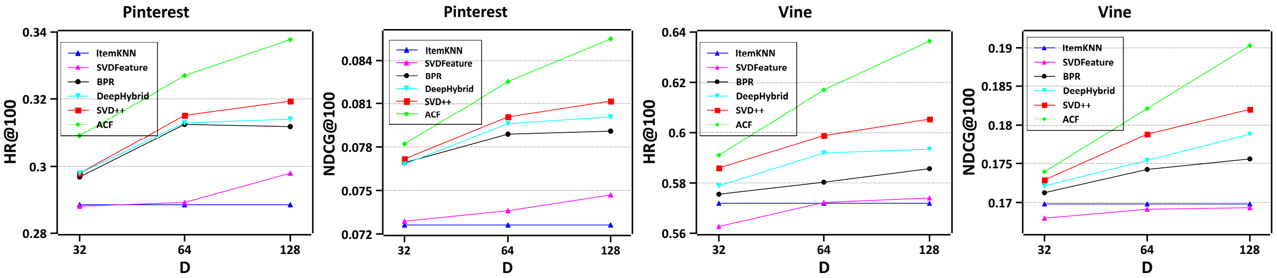
- **SVDFeature [9].** SVDFeature is a generic model for feature-based collaborative filtering, which incorporates different features that directly affect users' preferences over items with CF. In this paper we use the item visual feature as raw features to feed into SVDFeature.
- **Deep Hybrid [33].** Deep content-based method decomposes the user-item matrix into latent user and item vectors by matrix factorization (MF) and uses convolution neural network to regress multimedia content to the item latent vectors. In this paper, we use the SVD++ framework to learn the latent vectors for a fair comparison and use CNN [18] to regress the visual representations of multimedia items to the latent vectors.

Feature Extraction. We adopted the widely-used architectures ResNet-152 [18] to extract visual features for both images and frames of videos.

- **Image.** As we mentioned before, different users may be interested in different parts/components of the same image; in the context of image recommendation, the "components" of an image is considered in spatial level as the "regions" of the image. We use the *res5c* layer feature map in the ResNet-152 architecture to construct the component-level features. Specifically, for each image, the $7 \times 7 \times 2048$ feature map can be seen as 49 feature vectors of 2048-D for the 49 different regions in the image.
- **Video.** For each video, the component-level visual features are decomposed into the frame level. The frame feature can be obtained through the same way as that for the image feature based on the feature map. To simplify the process, we use the output of *pool5* layer in ResNet-152, which is actually the mean pooling of the feature maps, as the feature vector for each frame.

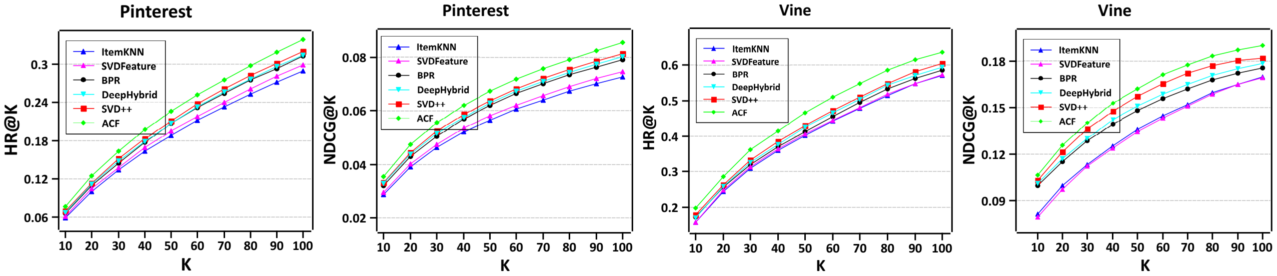
Parameter Settings. For models that are based on MF, we randomly initialized model parameters with a Gaussian distribution (with a mean of 0 and standard deviation of 0.01), optimizing the model with stochastic gradient descent (SGD). We tested the batch size of [256, 512], the latent feature dimension of [32,64,128], the learning rate of [0.001, 0.005, 0.01, 0.05, 0.1] and the regularizer of [0.00001, 0.0001, 0.001, 0.01, 0.1, 0]. As the findings are consistent across the dimension of latent vectors, if not specified, we only show the results of $D = 128$, a relatively large number that returns good accuracy.

[1] https://goo.gl/LjMoYa



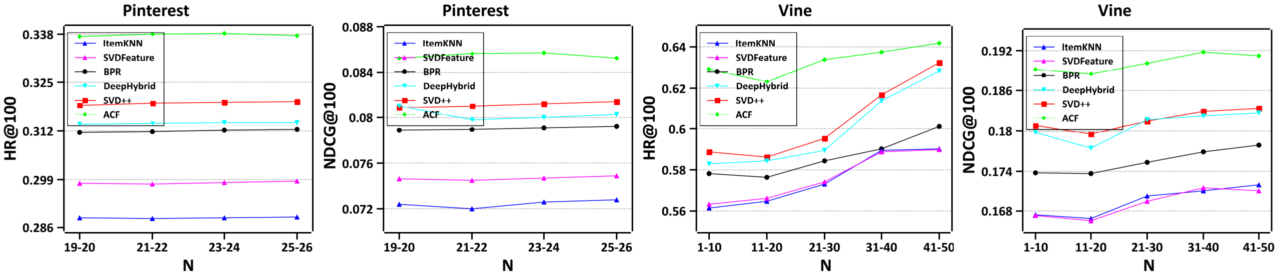
(a) Pinterest-HR@100 (b) Pinterest-NDCG@100 (c) Vine-HR@100 (d) Vine-NDCG@100

Figure 2: Performance of HR@100 and NDCG@100 w.r.t. the number of predictive factors on two datasets.



(a) Pinterest-HR@K (b) Pinterest-NDCG@K (c) Vine-HR@K (d) Vine-NDCG@K

Figure 3: Performance of Top-K item recommendation where K ranges from 10 to 100 on two datasets.



(a) Pinterest-HR@100 (b) Pinterest-NDCG@100 (c) Vine-HR@100 (d) Vine-NDCG@100

Figure 4: Performance of HR@100 and NDCG@100 w.r.t. the number of items per user on two datasets.

5.2 Model Comparison (RQ1)

Figure 2 shows the performance of HR@100 and NDCG@100 with respect to the number of latent factors. Due to the poor performance of UCF and CBF, they are omitted in Figure 2 to better highlight the performance differences among the rest of methods. From the figure, we can observe that:

(1) Our proposed method achieves the best performance on both datasets, significantly outperforming the state-of-the-art MF and Hybrid methods (on average, the relative improvement over the best baseline SVD++ is 5.19%).

(2) Although the Vine dataset is more sparse than Pinterest, the performance is much better. The reason may be that the set of videos and users in Vine is constructed from the set of densely connected users, in which the user-item pattern is more strong. This is also the reason why the performance of ItemKNN on Vine is closer to that of the other MF methods, since the neighborhood CF method, such as ItemKNN, could achieve acceptable performance based on the strong pattern.

(3) With the increase of the number of latent factors, the performance improvement of ACF compared with other baseline methods also increases. The reason may be that the visual features are more informative, which require relatively larger hidden dimension to incorporate the visual information.

Figure 3 shows the performance of Top-K recommended lists where the ranking position K ranges from 10 to 100. As can be seen, ACF demonstrates consistent improvements over other methods across all positions, and we further conducted the one-sample paired t-tests, to verify that all improvements are statistically significant for $p < 0.05$.

Table 2: Effect of attention mechanism on item and component (comp) level. AVG represents the average pooling strategy and ATT represents the attention mechanism. * denotes the statistical significance for $p < 0.05$.

Model	Level		Pinterest		Vine	
	Item	Comp	HR	NDCG	HR	NDCG
ACF	AVG	–	31.95%	8.12%	60.54%	18.20%
	ATT	AVG	33.21%	8.42%	62.81%	18.75%
	ATT	ATT	**33.78%***	**8.55%***	**63.65%***	**19.03%***

5.3 Model Analysis: Performance over Users of Different Sparsity Levels (RQ1)

Recall that our model characterizes each user based on the set of items the user has interacted with. To investigate the performance of our model over users of different sparsity levels, we show the performance with respect to the number of items a user has in Figure 4. Note that we did not re-train the model with different sets of users, instead we divide the test set into different groups by the number of items per user. From Figure 4, we observe that:

(1) Our model ACF with attention mechanisms of item- and component-level information consistently outperforms other baseline methods for all the number of item settings. It demonstrates the robustness and flexibility of ACF on different datasets.

(2) We also found that when the number of items per user is relatively small, ACF performs much better than the other methods, which indicates that the attention mechanism could improve the recommendation quality when there is insufficient training data for each user.

5.4 Model Ablation: Effect of Attention Mechanisms at Item- and Component-Level (RQ2)

To get a better understanding of the proposed ACF model, we further evaluate the key components of ACF — attention mechanism at item and component level. Table 2 shows the effect of attention mechanism at item- or component-level respectively. Note that (1) when we do not consider both the two levels of attentions, which means a normalized constant weight is used for neighborhood nodes in Eqn. (7), our model degenerates to SVD++ and (2) when we consider only the item-level attention, the item content feature at the item-level attention is the whole image/video feature which is the average of component features in the item. From the table, we can observe that:

(1) When the attention mechanism is applied at both the item- and the component-level, the performance for multimedia recommendation is improved as compared with utilizing average pooling in each level. The good performance of attention mechanism shows that the characteristics of users, items and visual contents are reflected at both levels. The contribution of collaborative information of users and items and the visual content will be evaluated in the next section.

Table 3: Effect of user, item and content attention mechanisms. U, V and P represents the user, item, and the auxiliary item information in Eqn. (5) respectively, and X indicates the content information of the item in Eqn. (8). * denotes the statistical significance for $p < 0.05$.

Model	Attention Type	Pinterest		Vine	
		HR	NDCG	HR	NDCG
ACF	None	31.95%	8.12%	60.54%	18.20%
	U+V	32.17%	8.31%	61.68%	18.36%
	U+P	32.69%	8.34%	62.37%	18.65%
	U+V+P	32.96%	8.32%	62.60%	18.71%
	U+V+P+X	**33.78%***	**8.55%***	**63.65%***	**19.03%***

(2) The attention mechanism at item-level contributes more for our model as compared to that at component-level. This may be due to the fact that the item-level attention mechanism can capture the representative items among all user's interactions, while the component-level attention mechanism may only work in complex items with rich contents. For example, as for some micro-videos on Vine, the visual content is highly related to a single theme and the difference among frames is not significant. In such situation, the component-level attentive network could give similar weights to frames, which may weaken the effect of component-level attention.

5.5 Model Ablation: Effect of User, Item and Content Information (RQ2)

Recall that to generate the attention weights α and β in Eqn. (8) and Eqn. (10), we incorporate different information sources, such as collaborative information of users (**U**) and items (**V** and **P**), and the visual content (**X**). To evaluate the contribution of each information source to the attention mechanism, we conducted experiments based on different combination of these sources as shown in Table 3. Note that since the number of all combinations is too large, we omit the ones without user information, which perform the worst among all combinations. From the table, we can observe that:

(1) The information of both user and item contributes to our model as compared to a constant weight model. It demonstrates that our attention mechanism can utilize the characteristics of each user and item to improve the performance of the recommendation task.

(2) The information of users is more effective than the items to improve recommendation performance. Hence, the discrimination of user preference is more discriminative than item characteristics, which is consistent with the previous finding that item-level attention is more important than component-level attention. Another interesting finding is that the auxiliary item latent vector **P** is more effective than **V**. This may be due to the different functions of the two vectors, that **V** is used to represent the item itself while **P** is proposed to characterize the user from the perspective of items. Therefore, when combining with user vector **U** to calculate the attention weight, **P** performs better since they are in the same domain.

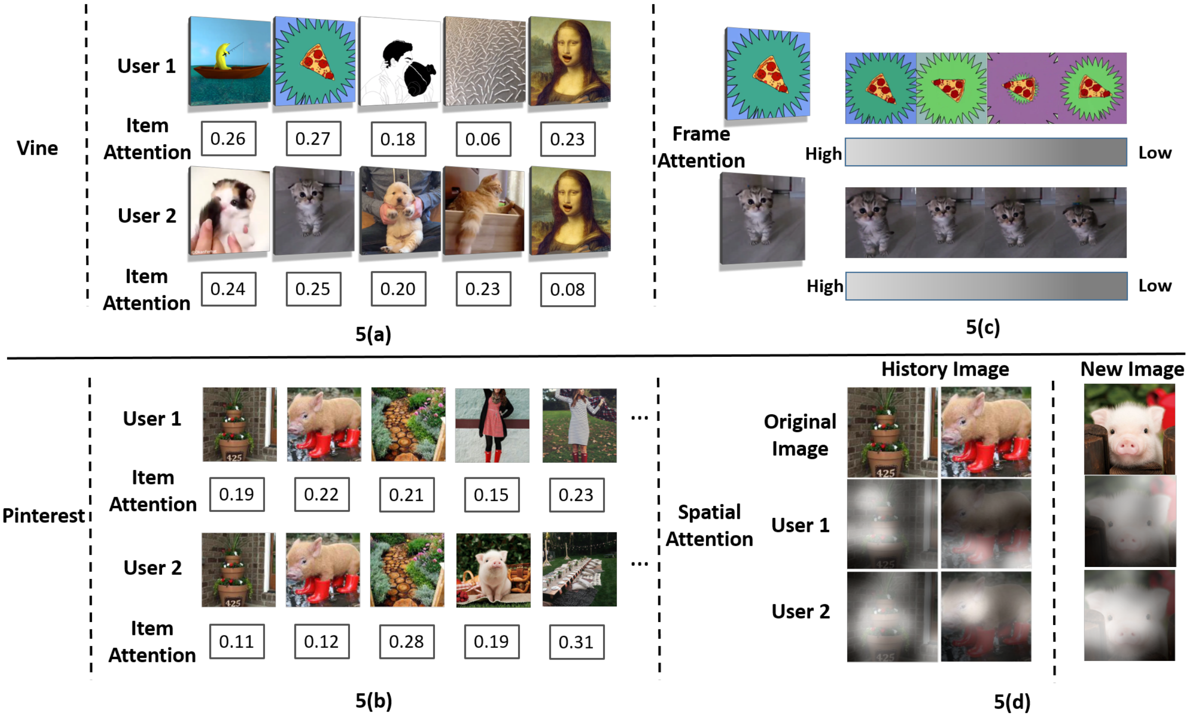


Figure 5: Visualization results on item-level attention and component-level attention from each dataset. For the item-level, the value under each item represents the attention weight of the item. While for the component-level, we use a heat map to represent the attention value, in which the darker the color is, the lower its represented attention value is (cf. Section 5.6).

5.6 Attention Visualization

We provide qualitative examples in Figure 5 for the better understanding of our attention model. In particular, Figure 5(a) and 5(b) show the item-level attention weights with respect to the items users liked in Vine and Pinsterest datasets, respectively. As can be seen from Figure 5(a), user 1 seems to have a preference for "cartoon" videos as he/she liked four related videos (except the 4-th one), while user 2 tends to prefer videos about "animals". Accordingly, we observe that the corresponding item-level attention weights depict these facts well. Specifically, the item-level attention weights of user 1 of the "cartoon" videos are much higher compared to that of the 4-th video. In addition, we find that for the same user, similar videos would share close attention weights, as the first 4 videos liked by user 2 are all about "animals" and hence share similar weights. These observations suggest that our model is able to capture user preferences via the item-level attention.

In addition, we also investigate the component-level attention visualization on both datasets. Figure 5(c) shows users' frame attention over videos from Vine and Figure 5(d) presents users' spatial attention over images from Pinterest. As shown in Figure 5(c), users can show different interest in different frames of the same videos. This may be affected by the content of the frame and the taste of the user. However, for each video example in Figure 5(c), the attention weights of frames are not that different since there is no much visual difference among them. For the spatial attention shown in Figure 5(d), we observe that users 1 and 2 share much similar attention pattern for the first image, which only contains a simple object (i.e., the "potted landscape"). From the second image that generally consists of two objects—a "pig" and a pair of red "boots"—we can see that users 1 and 2 are then attracted by different parts, *i.e.*, the "boots" and the "pig" parts. This implies that users can be interested in different regions of images with rich semantics. User 1 gives high attention weight to the "boots" part may due to the fact that she has liked many fashion-related images that contain red "boots". To further validate this point, we took a new image as an testing sample and visualized users 1 and 2's attentions. As can be seen, user 1 focuses more on the red flowers while user 2 is more interested in the "pig".

6 CONCLUSIONS

In this paper, we have proposed an Attentive Collaborative Filtering (ACF) model to address the implicit feedback in multimedia recommendation. We argue that there are two types of implicit feedback: item-level and component-level, which are usually neglected in conventional methods. To this end, we introduced the item- and component-level attention model to assign attentive weights for inferring the underlying users' preferences encoded in the implicit user feedback. ACF can be efficiently trained by employing SGD. To the best of our knowledge, ACF is the first model that exploits an attention mechanism in CF with implicit feedback. We conducted the extensive experiments on two real-world multimedia social networks: Vine and Pinterest, and demonstrated that ACF can consistently outperform the state-of-the-art CF models in multimedia recommendation. Since ACF is a generic attention-based CF framework, we plan to extend ACF in various CF models such as factorization machines [29], and the recently proposed Neural CF [20] and Discrete CF [43]. Moreover, we would explore higher-order component-level attentions such as relationships between objects [22, 42].

7 ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments. NExT research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2014.

[2] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *WWW*, pages 895–904. ACM, 2008.

[3] M. Bendersky, L. G. Pueyo, J. J. Harmsen, V. Josifovski, and D. Lepikhin. Up next: retrieval methods for large scale related video suggestion. In *KDD*, pages 1769–1778. ACM, 2014.

[4] B. Chen, J. Wang, Q. Huang, and T. Mei. Personalized video recommendation through tripartite graph propagation. In *Proceedings of the International Conference on Multimedia*, pages 1133–1136. ACM, 2012.

[5] J. Chen. Multi-modal learning: Study on A large-scale micro-video data collection. In *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*, pages 1454–1458. ACM, 2016.

[6] J. Chen, X. Song, L. Nie, X. Wang, H. Zhang, and T. Chua. Micro tells macro: Predicting the popularity of micro-videos via a transductive model. In *MM*, pages 898–907. ACM, 2016.

[7] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*. IEEE, 2017.

[8] T. Chen, X. He, and M.-Y. Kan. Context-aware image tweet modelling and recommendation. In *MM*, pages 1018–1027. ACM, 2016.

[9] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu. Svdfeature: a toolkit for feature-based collaborative filtering. *JMLR*, 13:3619–3622, 2012.

[10] X. Chen, Y. Zhang, H. X. Qingyao Ai, J. Yan, and Z. Qin. Personalized key frame recommendation. In *SIGIR*. ACM, 2017.

[11] Z. Cheng and J. Shen. On effective location-aware music recommendation. *TOIS*, 34(2):13:1–13:32, 2016.

[12] P. Cui, Z. Wang, and Z. Su. What videos are similar with you?: Learning a common attributed representation for video recommendation. In *MM*, pages 597–606. ACM, 2014.

[13] A. Farseev, I. Samborskii, A. Filchenkov, and T.-S. Chua. Cross-domain recommendation via clustering on multi-layer graphs. In *SIGIR*. ACM, 2017.

[14] F. Feng, L. Nie, X. Wang, R. Hong, and C. Tat-Seng. Computational social indicators: a case study of chinese university ranking. In *SIGIR*. ACM, 2017.

[15] X. Geng, H. Zhang, J. Bian, and T. Chua. Learning image and user features for recommendation in social networks. In *ICCV*, pages 4274–4282. IEEE, 2015.

[16] X. Geng, H. Zhang, Z. Song, Y. Yang, H. Luan, and T. Chua. One of a kind: User profiling by social curation. In *MM*, pages 567–576. ACM, 2014.

[17] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *JMLR*, pages 249–256. JMLR.org, 2010.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE, 2016.

[19] X. He, M. Gao, M.-Y. Kan, and D. Wang. Birank: Towards ranking on bipartite graphs. *TKDE*, 29(1):57–71, 2017.

[20] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *WWW*, pages 173–182. ACM, 2017.

[21] X. He, H. Zhang, M. Kan, and T. Chua. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*, pages 549–558. ACM, 2016.

[22] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 2016.

[23] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. IEEE, 2008.

[24] S. Kabbur, X. Ning, and G. Karypis. FISM: factored item similarity models for top-n recommender systems. In *KDD*, pages 659–667. ACM, 2013.

[25] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434. ACM, 2008.

[26] T. Mei, B. Yang, X. Hua, L. Yang, S. Yang, and S. Li. Videoreach: an online video recommendation system. In *SIGIR*, pages 767–768. ACM, 2007.

[27] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, pages 502–511. IEEE, 2008.

[28] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *Proceedings of the Adaptive Web, Methods and Strategies of Web Personalization*, pages 325–341. Springer, 2007.

[29] S. Rendle. Factorization machines. In *ICDM*, pages 995–1000. IEEE, 2010.

[30] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. IEEE, 2009.

[31] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM, 2001.

[32] J. Shen, M. Wang, S. Yan, and P. Cui. Multimedia recommendation: technology and techniques. In *SIGIR*, page 1131. ACM, 2013.

[33] A. van den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *NIPS*, pages 2643–2651. NIPS Foundation, 2013.

[34] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu. Multimodal graph-based reranking for web image search. *TIP*, 21(11):4649–4661, 2012.

[35] M. Wang, X. Liu, and X. Wu. Visual classification by l1-hypergraph modeling. *TKDE*, 27(9):2564–2574, 2015.

[36] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In *WWW*, pages 391–400. ACM, 2017.

[37] X. Wang, X. He, L. Nie, and T.-S. Chua. Item silk road: Recommending items from information domains to social users. In *SIGIR*. ACM, 2017.

[38] X. Wang, L. Nie, X. Song, D. Zhang, and T.-S. Chua. Unifying virtual and physical worlds: Learning toward local and global consistency. *TOIS*, 36(1):4, 2017.

[39] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057. JMLR.org, 2015.

[40] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, pages 4651–4659. IEEE, 2016.

[41] M. Zanfir, E. Marinoiu, and C. Sminchisescu. Spatio-temporal attention models for grounded video captioning. In *ACCV*, pages 104–119. Springer, 2016.

[42] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017.

[43] H. Zhang, F. Shen, W. Liu, X. He, H. Luan, and T. Chua. Discrete collaborative filtering. In *SIGIR*, pages 325–334. ACM, 2016.

[44] H. Zhang, Z. Zha, Y. Yang, S. Yan, Y. Gao, and T. Chua. Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval. In *MM*, pages 33–42. ACM, 2013.

[45] J. Zhang, L. Nie, X. Wang, X. He, X. Huang, and T. Chua. Shorter-is-better: Venue category estimation from micro-video. In *MM*, pages 1415–1424. ACM, 2016.

[46] Z. Zhao and M. Shang. User-based collaborative-filtering recommendation algorithms on hadoop. In *KDD*, pages 478–481. ACM, 2010.