Crowdsourced Time-Sync Video Recommendation via Semantic-Aware Neural Collaborative Filtering

Zhanpeng Wu[1,2], Yan Zhou[1,2], Di Wu[1,2](✉), Yipeng Zhou[3], and Jing Qin[4]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
{wuzhp5,zhouy347}@mail2.sysu.edu.cn, wudi27@mail.sysu.edu.cn

[2] Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China

[3] Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2109, Australia
yipeng.job@gmail.com

[4] School of Nursing, The Hong Kong Polytechnic University, Hong Kong, China
harry.qin@polyu.edu.hk

Abstract. As an emerging type of video comments, time-sync comments (TSCs) enable viewers to make comments on video shots in a real-time manner. Such comments well reflect user interests in the frame level, which can be utilized to further improve the accuracy of video recommendation. In this paper, we make the first attempt in this direction and propose a new video recommendation algorithm called SACF by exploiting temporal relationship between time-sync comments and video frames. Our algorithm can extract a rich set of semantic features from crowdsourced time-sync comments, and combine latent semantic representations of users and videos by neural collaborative filtering. We conduct extensive experiments using real TSC datasets, and our results show that our proposed algorithm can improve the recommendation performance by 9.73% in HR@10 and 5.72% in NDCG@10 compared with other baseline solutions.

Keywords: Recommender system · Time-synchronized comment · Collaborative filtering

1 Introduction

The past decade has witnessed a dramatic increase of online videos, including online TV episodes, online movies, user-generated contents, livecast, and etc. The traffic generated by world-leading online video websites (e.g., Youtube, Netflix, Tencent Video, Hulu) has dominated the whole Internet backbone. In our daily life, users watch online videos for learning, news, and funny stuff. Normally, users tend to make comments on video after watching. In recent years, there emerges a new type of video comments, called Time-Sync Comments (or Danmu, bullet-screen comments), which allow a user to make comments on video shots in a real-time manner. Time-sync comments are flying across the screen and people who are watching the same video also see the flying comments. To date, the service of time-sync comments has been provided by quite a few online video websites, such as YouTube[1], Twitch[2], AcFun[3], BiliBili[4], NicoNico[5], and so on. In Fig. 1, we show an example of a video clip[6] with time-sync comments.

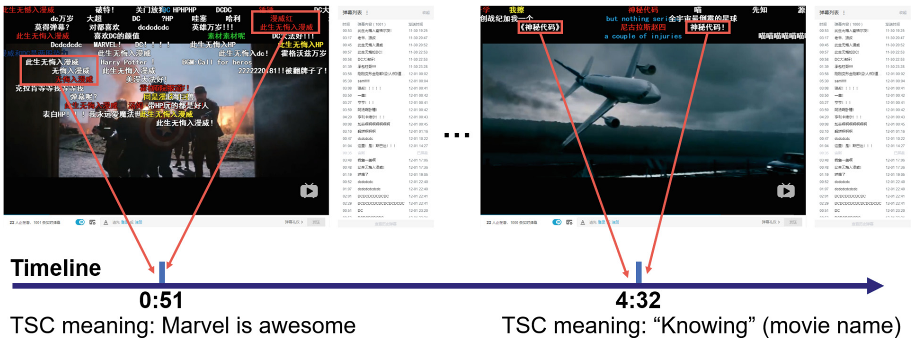

Timeline
0:51
TSC meaning: Marvel is awesome
4:32
TSC meaning: "Knowing" (movie name)

Fig. 1. Example of a crowdsourced time-sync video.

Different from conventional video comments, time-sync comments are synchronized with a video's playback time. It provides a possibility for viewers who are watching the same video to share their watching experience and interact with each other. Latent features can be extracted from time-sync comments to provide more detailed information on user interests. For instance, viewers who wrote comments in nearby playback time positions are likely to have some kinds of similarity or association (e.g., like or dislike a specific video shot). Intuitively, those viewers can be categorized into the same group with implicit similar preferences. Moreover, a continuous bundle of time-sync comments can describe video contents to some extent. Such kind of information is useful for video recommendation to further improve user experience.

In this paper, we propose a new video recommendation algorithm for crowdsourced time-sync videos, which is called *SACF (Semantic-Aware Collaborative Filtering)*. The basic idea of SACF is to exploit the temporal relationship between time-sync comments and video frames, and extract latent semantic

1 https://www.youtube.com/.
2 https://www.twitch.tv/.
3 http://www.acfun.cn.
4 https://www.bilibili.com.
5 https://www.nicovideo.jp.
6 Available at https://www.bilibili.com/video/av22135056.

representations of time-sync comments to provide more accurate video recommendation. Our proposed algorithm can model user preferences in the frame level. In summary, our main contributions in this paper can be listed as below:

- We propose a novel video recommendation algorithm called SACF to improve the performance of crowdsourced time-sync videos. Our algorithm extends traditional video recommendation algorithms by embedding latent semantic representations extracted from TSCs.
- To better utilize interaction patterns, we integrate all the representations with a multi-layer perceptron (MLP) model. By embedding extra semantic-aware information, our approach can easily achieve similar user and item interest filtering, and mitigate the cold-start problem.
- We also validate our proposed algorithm using a real TSC dataset obtained from the BiliBili video website. The experiment results show that our algorithm significantly outperforms other baselines by up to 9.73% in HR@10 and 5.72% in NDCG@10.

The rest of our paper is organized as follows: we first introduce the relevant work in Sect. 2. We describe the details of our algorithm in Sect. 3. The dataset and experiments are introduced in Sect. 4. Finally we conclude the whole paper and discuss our future work in Sect. 5.

2 Related Work

The topic of recommender systems has been extensively studied in the past years. He *et al.* [8], Koren *et al.* [13], Mnih and Salakhutdinov [19] have shown the excellent ability of matrix factorization model in the field of rating prediction problem. In addition to such classic research question, Top-K recommendation using implicit feedback are also worthy of attention. Hu *et al.* [9] and Rendle *et al.* [21] are the masterpieces of them. Moreover, in recent years, neural network models [29] have gained widespread attention because of their ability to easily fit multi-dimensional features and learn nonlinear relationships, which are concerned by Covington *et al.* [4] and He *et al.* [7].

With the development of this field, in order to improve the effectiveness in user preference modeling, the incorporation of contextual information has attracted major research interests, such as the work of Adomavicius and Tuzhilin [1] and Verbert *et al.* [25]. User review is one of the most effective contextual information to model user preference and these algorithms are receiving more and more research attention. Tang *et al.* [23] aims to incorporate user- and product-level information for document sentiment classification. Tang *et al.* [24] leverages the reviews for user modeling and predicts the rating of user review. In addition, lots of the review-based models focus on enhancing the effectiveness of rating prediction, like Ganu *et al.* [5], McAuley and Leskovec [17]. To achieve better overall recommendation performance, Liu *et al.* [15], Tan *et al.* [22], Wu and Ester [27] extract the user opinions from review text and combine these information into the conventional models for higher recommendation accuracy. Besides, Zhang *et al.* [30] integrates traditional matrix factorization technology with word2vec model proposed by Mikolov *et al.* [18] for precise modeling. Notably, these proposed models mainly focus on rating prediction problem, and learn knowledge from traditional reviews which are longer than time-sync comments and contain richer content and semantics.

Time-synchronized comment is first introduced by Wu *et al.* [26] for video shot tagging. Recently, as an emerging type of user-generated comment, TSC has many practical properties to describe a video in frame-level. As mentioned by He *et al.* [6], herding effect and multiple-burst phenomena make TSC significant different from the traditional reviews, and it also proves that TSCs have a great correlation with video frame content and user reactions. Thus this emerging comment type is worthy of being used for video highlight shot extraction and content annotation, presented by Xian *et al.* [28] and Ikeda *et al.* [10]. Besides, an increasing number of models are trying to label videos based on TSCs, such representative work as Lv *et al.* [16]. Chen *et al.* [3] and Ping [20] further leverages the TSCs to extract the features of video frames and applies it to key frame recommendation. However, these methods are clinging to the video shot itself and do not consider the co-occurrence among TSCs.

Compared with the aforementioned approaches, we conduct a semantic-aware collaborative filtering algorithm, which can efficiently extract latent representations from TSCs within the videos and achieve significantly performance improvement in Top-K recommendation task.

3 Design of Semantic-Aware Video Recommendation Algorithm

By fusing latent semantic representation from video TSCs and traditional interaction paradigm as a composite entity, we propose our semantic-aware collaborative filtering (SACF) video recommendation algorithm, which has flexibility and non-linearity profited by exploiting multi-layer perceptron as a fundamental framework. We will discuss the details of our algorithm in the following subsections.

3.1 Problem Definition

We first define the problem formally. Suppose there are N users $\boldsymbol{u} = \{u_1, u_2, \ldots, u_N\}$, M videos $\boldsymbol{i} = \{i_1, i_2, \ldots, i_M\}$ and T TSCs $\boldsymbol{c} = \{c_1, c_2, \ldots, c_T\}$. The TSC written by user u leaving in video i at video time t is defined as a 2-tuple $<u_i^t, c_i^t>$. Thereby, for each video $i \in \boldsymbol{i}$, we can obtain two different sequences, that is, TSC writer sequence $\boldsymbol{s}_i^{(u)} = (u_i^1, u_i^2, \ldots, u_i^{T_i})$ and TSC content sequence $\boldsymbol{s}_i^{(c)} = (c_i^1, c_i^2, \ldots, c_i^{T_i})$, where T_i is the amount of TSCs in video i and each element within the sequence is ranked by its video time.

Suppose the representation of user u is defined as $\boldsymbol{w}_u$, and the representation of video i is defined as $\boldsymbol{d}_i$. The user semantic representations $\boldsymbol{W} = \{\boldsymbol{w}_u | u \in \boldsymbol{u}\}$

Table 1. Notations used in SACF algorithm

Notation	Description
T_i	Total number of TSCs in video i
$\boldsymbol{u}$	The set of N users $\{u_1, u_2, \ldots, u_N\}$
$\boldsymbol{i}$	The set of M videos $\{i_i, i_2, \ldots, i_M\}$
$\boldsymbol{c}$	The set of T TSCs $\{c_i, c_2, \ldots, c_T\}$
u_i^t	User u writes a TSC in video i at time t
c_i^t	The TSC content c written in video i at time t
$\boldsymbol{s}_i^{(u)}$	The sequence of TSC users $(u_i^t)_{t=1}^{T_i}$ in video i
$\boldsymbol{s}_i^{(c)}$	The sequence of TSC content $(c_i^t)_{t=1}^{T_i}$ in video i
$\boldsymbol{w}_u$	Semantic representation of user u
$\boldsymbol{W}$	Semantic representations of $\boldsymbol{u}$
$\boldsymbol{d}_i$	Semantic representation of video i
$\boldsymbol{D}$	Semantic representations of $\boldsymbol{i}$
$\boldsymbol{P}$	Latent representations of $\boldsymbol{u}$
$\boldsymbol{Q}$	Latent representations of $\boldsymbol{i}$
$\boldsymbol{v}_u$	Index of user u
$\boldsymbol{v}_i$	Index of video i
f	Interaction function
Θ	Parameters of neural network
Φ_L	The activation function of L^{th} layer
$\boldsymbol{x}_L$	The hidden state of L^{th} layer
$\mathcal{O}$	The set of positive
$\mathcal{O}^-$	The set of sampled negative
$\mathcal{D}$	The set of training data

and video semantic representations $\boldsymbol{D} = \{\boldsymbol{d}_i | i \in \boldsymbol{i}\}$ are learned from the sequencing data $\boldsymbol{s}_i^{(u)}$ and $\boldsymbol{s}_i^{(c)}$, which exploit the improved word embedding technology.

Given all the historical user-video interaction data $\mathcal{D} = \{\mathcal{O}, \mathcal{O}^-\}$, where $\mathcal{O}$ and $\mathcal{O}^-$ denote the positive and negative instances, respectively. For a user u and a set of corresponding unseen videos, our goal is to find a interaction function $f(\cdot)$ to rank all the unseen videos based on how much s/he like the video. The top K most likely to watch videos are the final results recommended to user u. For reference, we list the notations used throughout the algorithm in Table 1.

3.2 Latent Semantic Representation

In this section, we explain the methods to capture latent semantic representation, which aim to extract the similarity lurked in users and videos. Before digging into the details, to better model the information of Internet slangs, we need to conduct a few data preprocessing work:

- Different character components within a TSC may indicate different meanings. Thus, we will split a complete TSC into multiple substrings, where each substring represents a series of consecutive characters of the same type and these substrings will be treated as part of the TSC set $\boldsymbol{c}$ as well. These character types can be English letter, pure number and Chinese character. For instance, TSC "awesome!!2333" will be treated as two TSCs, i.e. "awesome" and "2333" (laughter).
- Moreover, the excessive TSCs will obviously impair the performance of the algorithm. Thereby, those Chinese substrings that are too long will also be segmented (the threshold of segmentation in our experiments is more than five consecutive Chinese characters).

Inspired by [14,18], we propose the improved word embedding methods to learn representations. In this schema, each user is mapped to a unique latent vector $\boldsymbol{w}_u$ and each vector is represented as one column of the matrix $\boldsymbol{W}$ where $\boldsymbol{W}$ indicates the user semantic representation matrix. Given the sequence of user $\boldsymbol{s}_i^{(u)}$ from a finite user set $\boldsymbol{u}$, the objective function aiming at maximizing is formulated as follows,

$$\frac{1}{T_i}\sum_{t=k}^{T_i-k}\sum_{-k\leq j\leq k, j\neq 0}\log p(u_i^t|u_i^{t+j}) \tag{1}$$

where k is the context window size and $p(\cdot)$ is the softmax function,

$$p(u_i^t|u_i^{t+j}) = \frac{\exp(\boldsymbol{w}_{u_i^t}{}^T\cdot\boldsymbol{w}_{u_i^{t+j}})}{\sum_{u'\in\boldsymbol{u}}\exp(\boldsymbol{w}_{u_i^{t+j}}{}^T\cdot\boldsymbol{w}_{u'})} \tag{2}$$

After the training converges, users who have the similar watching patterns will be projected to a similar position in the vector space. We leave these user semantic representations $\boldsymbol{W}=\{\boldsymbol{w}_u|u\in\boldsymbol{u}\}$ for later use. Likewise, TSC and video are mapped to a unique latent vector $\boldsymbol{w}_c$ and $\boldsymbol{d}_i$, respectively. Each vector is a column of their respective matrix, $\boldsymbol{W}_c$ and $\boldsymbol{D}$ where $\boldsymbol{D}$ indicates the video semantic representation matrix. We will further leverage the information within $\boldsymbol{W}$ and $\boldsymbol{D}$ in the next subsection. Given the sequence of $\boldsymbol{s}_i^{(c)}$ from a finite TSC set $\boldsymbol{c}$, the objective function is defined as,

$$\frac{1}{T_i}\sum_{t=k}^{T_i-k}\sum_{-k\leq j\leq k, j\neq 0}\log p(c_i^t|c_i^{t+j}) \tag{3}$$

Similarly, the softmax function can be formulated as,

$$p(c_i^t|c_i^{t+j}) = \frac{\exp(\boldsymbol{z}_{c_i^t,i}{}^T\cdot\boldsymbol{z}_{c_i^{t+j}})}{\sum_{c'\in\boldsymbol{c}\ and\ i'\in\boldsymbol{i}}\exp(\boldsymbol{z}_{c_i^{t+j},i}{}^T\cdot\boldsymbol{z}_{c',i'})} \tag{4}$$

where the latent vector $\boldsymbol{z}$ is constructed from $\boldsymbol{w}_c$ and $\boldsymbol{d}_i$. In particular, the vector $\boldsymbol{z}$ is the sum of $\boldsymbol{w}_c$ and $\boldsymbol{d}_i$. Our ultimate goal is to obtain the video semantic representation set $\boldsymbol{D} = \{\boldsymbol{d}_i | i \in \boldsymbol{i}\}$ for the next step.

3.3 Algorithm Design

We now elaborate the SACF algorithm in details. Our approach employs a deep neural network, which is empowered the capability to learn the non-linear interactions from input data and can be easily embedded with extra features. Therefore, in addition to treat the identify of a user and a video in pure collaborative filtering as basic input feature, SACF also transforms the video TSC corpus to two kinds of representations, which are generated from user and video latent semantic information. As shown in Fig. 2, on the top of the input layer, each user and video is mapped to two corresponding vectors, i.e. identification dense vector and latent semantic representation vector. The generation methods of latent semantic representation have been fully explained in Sect. 3.2. And then such a 4-tuple of embedding vectors is fed into a multi-layer neural network, where each layer is combined with fully connection. The final output layer represents a predictive probability $\hat{r}_{ui}$, which is trained by minimizing the binary cross-entropy loss between $\hat{r}_{ui}$ and its target value r_{ui}.

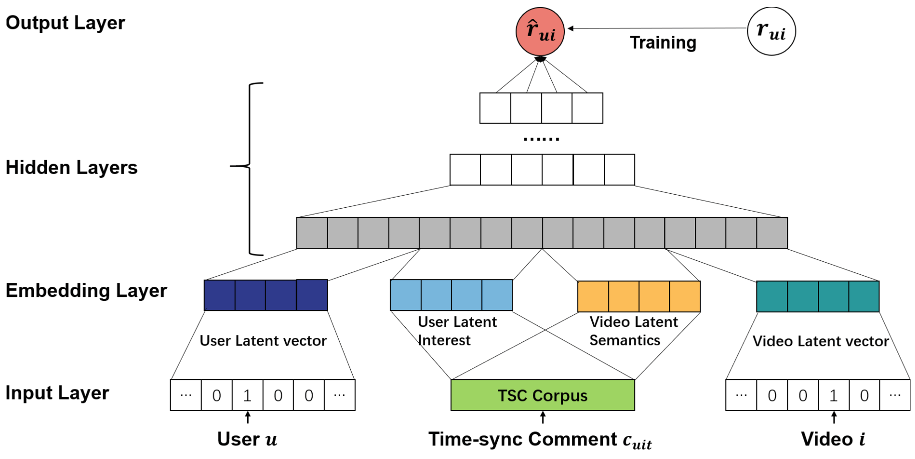


Fig. 2. SACF model structure for time-sync video recommendation.

Consequently, we can further formulate the SACF algorithm as,

$$\hat{r}_{ui} = f(\boldsymbol{P}^T \boldsymbol{v}_u, \boldsymbol{Q}^T \boldsymbol{v}_i, \boldsymbol{W}^T \boldsymbol{v}_u, \boldsymbol{D}^T \boldsymbol{v}_i, |\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{W}, \boldsymbol{D}, \Theta) \quad (5)$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ denote the latent factor matrix for users and videos respectively. $\boldsymbol{W}$ is the user semantic representation matrix, and analogously, $\boldsymbol{D}$ is the video semantic representation matrix which extracts the latent features from the TSC corpus. $\boldsymbol{v}_u$ and $\boldsymbol{v}_i$ separately denote the index vector of user u and video i. Θ represents the parameters of the interaction function $f(\cdot)$.

As mentioned above, the function $f(\cdot)$ can be further defined as a multi-layer neural network,

$$\begin{aligned} & f(\boldsymbol{P}^T\boldsymbol{v}_u, \boldsymbol{Q}^T\boldsymbol{v}_i, \boldsymbol{W}^T\boldsymbol{v}_u, \boldsymbol{D}^T\boldsymbol{v}_i) \\ & \quad = \sigma(\Phi_L(\Phi_{L-1}(\ldots\Phi_1(\boldsymbol{P}^T\boldsymbol{v}_u, \boldsymbol{Q}^T\boldsymbol{v}_i, \boldsymbol{W}^T\boldsymbol{v}_u, \boldsymbol{D}^T\boldsymbol{v}_i)\ldots))) \end{aligned} \tag{6}$$

where σ is the mapping function of output layer and Φ_L denotes the L^{th} hidden layer in neural network. More specifically, we formulate each layer as follows,

$$\begin{aligned} \boldsymbol{x}_1 &= \Phi_1(\boldsymbol{p}_u, \boldsymbol{q}_i, \boldsymbol{w}_u, \boldsymbol{d}_i) = \left[\boldsymbol{p}_u \ \boldsymbol{q}_i \ \boldsymbol{w}_u \ \boldsymbol{d}_i\right]^T \\ \boldsymbol{x}_2 &= \Phi_2(\boldsymbol{x}_1) = g_2(\boldsymbol{A}_2^T\boldsymbol{x}_1 + \boldsymbol{b}_1) \\ &\ldots \\ \boldsymbol{x}_L &= \Phi_L(\boldsymbol{x}_{L-1}) = g_L(\boldsymbol{A}_L^T\boldsymbol{x}_{L-1} + \boldsymbol{b}_L) \\ \hat{r}_{ui} &= \sigma(\boldsymbol{h}^T\boldsymbol{x}_L) \end{aligned} \tag{7}$$

where $\boldsymbol{A}_L$, $\boldsymbol{b}_L$ and g_L respectively denote the weight matrix, bias vector and activation function of the L^{th} layer. $\boldsymbol{p}_u$ and $\boldsymbol{q}_i$ are the dense vector embedded with the one-hot encoding of user u and video i. $\boldsymbol{w}_u$ and $\boldsymbol{d}_i$ are latent semantic representation generated from crowdsourced TSC data. $\boldsymbol{h}$ represents the weight vector of the output layer.

To endow a probabilistic explanation for SACF, we need to constraint $\hat{r}_{ui}$ in range of $[0, 1]$, which can be achieved by adopting *Logit* or *Probit* as an activation function in output layer. We finally optimize SACF by minimizing the binary cross-entropy loss, and the objective function of SACF can be formulated as,

$$\mathcal{L} = -\sum_{(u,i)\in\mathcal{O}\cup\mathcal{O}^-} r_{ui}\log\hat{r}_{ui} + (1-r_{ui})\log(1-\hat{r}_{ui}) \tag{8}$$

where $\mathcal{O}$ denotes the set of observed interactions, and $\mathcal{O}^-$ denotes the set of unobserved interactions. To improve the training efficiency, $\mathcal{O}^-$ can be regarded as the negative instances sampled from all the unobserved interactions.

The semantic-aware neural collaborative filtering algorithm is illustrated in Algorithm 1. The algorithm can be considered as a two-stage process. Precisely, we pretrain the semantic representation of users and items in the first stage and embed it with the user-video interaction data in the second stage to maintain a holistic recommendation task. The algorithm works when the video has a series of TSC data. These data contain meta information about users and videos, and thus even if the user rarely sends any TSC, the algorithm can still leverage the implicit information to get the probability of watching a video.

Algorithm 1. Semantic-Aware Neural Collaborative Filtering Algorithm

```
 1: procedure SEMANTIC-AWARE NEURAL COLLABORATIVE FILTERING
    Input: User-item interaction set $\mathcal{D}$, TSC writer sequence set $\{\boldsymbol{s}_1^{(u)}, \boldsymbol{s}_2^{(u)}, \ldots, \boldsymbol{s}_M^{(u)}\}$
    and TSC content sequence $\{\boldsymbol{s}_1^{(c)}, \boldsymbol{s}_2^{(c)}, \ldots, \boldsymbol{s}_M^{(c)}\}$ for $M$ videos
    Output: Representations of $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{W}$, $\boldsymbol{D}$ and network parameters $\Theta$
 2:     initialize matrix $\boldsymbol{W}$ and $\boldsymbol{D}$
 3:     for $i = 1 \rightarrow M$ do
 4:         $\boldsymbol{s}_i^{(c)} \leftarrow SameTypeCharacterSegmentation(\boldsymbol{s}_i^{(c)})$
 5:         $\boldsymbol{s}_i^{(c)} \leftarrow ExtraLongCharacterSegmentation(\boldsymbol{s}_i^{(c)})$
 6:         for each $u_i \in \boldsymbol{s}_i^{(u)}$ and its contextual users $Context(u_i)$ do
 7:             for each $u_i^{'} \in Context(u_i)$ do
 8:                 update $\boldsymbol{W}$ by optimizing $softmax(u_i|u_i^{'})$
 9:             end for
10:         end for
11:         for each $c_i \in \boldsymbol{s}_i^{(c)}$ and its contextual TSCs $Context(c_i)$ do
12:             for each $c_i^{'} \in Context(c_i)$ do
13:                 update $\boldsymbol{D}$ by optimizing $softmax(c_i|c_i^{'})$
14:             end for
15:         end for
16:     end for
17:     initialize matrix $\boldsymbol{P}$ and $\boldsymbol{Q}$
18:     for each training epoch do
19:         for each paired index $(\boldsymbol{v}_u, \boldsymbol{v}_i) \in Loader(\mathcal{D})$ do   ▷ Loader will provide
                                                                        multiple iterators over
                                                                        the dataset
20:             compute:
21:                 $\hat{r}_{ui} \leftarrow f(\boldsymbol{P}^T\boldsymbol{v}_u, \boldsymbol{Q}^T\boldsymbol{v}_i, \boldsymbol{W}^T\boldsymbol{v}_u, \boldsymbol{D}^T\boldsymbol{v}_i)$
22:                 $loss \leftarrow criterion(\hat{r}_{ui}, r_{ui})$
23:             update $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\Theta$ by minimizing the $loss$
24:         end for
25:     end for
26:     return $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{W}$, $\boldsymbol{D}$ and $\Theta$
27: end procedure
```

4 Performance Evaluation

To demonstrate the superiority of our method, a time-sync video dataset crawled from Bilibili website is utilized for performance evaluation. We will discuss an overview of the dataset and experiment settings before presenting the experimental results.

4.1 Dataset Overview

Bilibili is one of the most popular TSC video sharing websites in China, and leads a trend of video interaction via TSC. We collected the video meta information and its corresponding TSC data from Bilibili website till December 15th, 2018. Note that the video TSC data we collected is only a part of fully historical data, because the platform will periodically remove the stale TSC data from the TSC pool and remain the latest TSC data. For the sake of reflecting the experimental results more significantly, we mainly focus on the data of gaming category, which attracts the most traffic on Bilibili. Based on these premises, our collected dataset totally contains 57,294 users and 2,637 videos. All these videos include 836,806 TSCs and 3,483 user-generated tags. More detailed statistic analyses are presented in Table 2.

Table 2. Overall statistics of our time-sync comment dataset

# of videos	2,637
# of users	57,294
# of TSCs	836,806
# of user-generated tags	3,483
Avg # of TSCs per user	14.61
Avg # of TSCs per video	317.33
Avg # of user-generated tags per video	7.01
Max/Min # of TSCs for a user	731/5
Max/Min # of TSCs for a video	4393/1
Max/Min # of user-generated tags for a video	14/1
Max/Min # of TSCs for a user leaving in a video	299/1

4.2 Experiment Settings

Evaluation Metrics. To evaluate the performance of TSC video recommendation, we employ two widespread adopted metrics, that is, *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain* (NDCG) [11]. These two metrics can respectively measure the classification and ranking performance in recommendation problem.

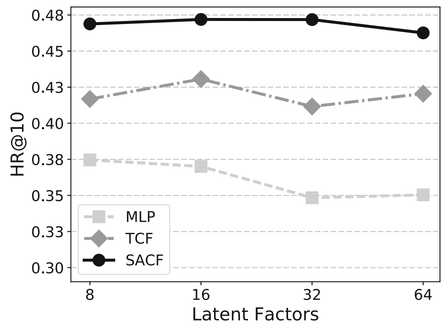
Baselines. Besides SACF method, we also implement two other algorithms for comparison, which are described as below,

- **MLP** is proposed by [7]. It is a pure collaborative filtering method, which uses only the identifies of user and item as embedding features. Previous work has shown its strong generalization ability benefited from DNN model.
- **TCF** is a variant of SACF. In contrast to SACF, TCF embeds the user-generated video tag information instead of the TSC information in SACF. It is a highly competitive baseline for video recommendation that fuses with conventional content features.

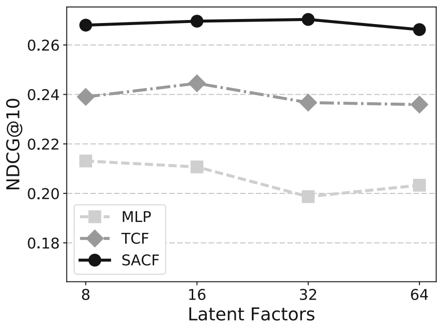
Parameter Settings. For better generality and comparison, we chose the widely used *leave-one-out* [2,7,8] evaluation schema, which holds the latest interaction as a test case and use the rest as training set. All the algorithms are optimized by the cross-entropy function defined in Eq. (8), where we randomly sample 4 negative instances for each positive instance. We use *Adam* [12] as optimizer to train algorithms, and fix the batch-size and learning rate at 256 and 0.001. Besides, in fairness, the number of hidden layers is set to 4 in all the experiments.

4.3 Experiment Results

Experiment 1: Performance Comparison. Since the size of last hidden layer implies the learning capability in DNN models, we can evaluate the performance in different factor size to achieve comprehensive comparison. Figure 3 illustrates that SACF outperforms other two baselines with the factors of 8, 16, 32 and 64 in both metrics when embedding size (ES) is set to 64. And the overall performance trend is presented as MLP<TCF<SACF. Meanwhile, it is worth mentioning that all the curves have different degrees of decline when factors become larger, indicating that a large factor size may probably cause overfitting and degrade the overall performance.



(a) **HR@10, ES=64**



(b) **NDCG@10, ES=64**

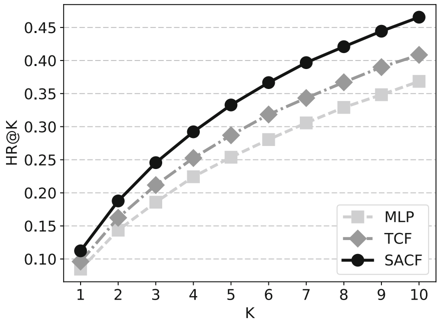
Fig. 3. Performance comparison in different size of latent factors.

Table 3 shows the precise results in different factor size. We notice that our proposed SACF algorithm exceeds MLP with a maximum performance improvement of 9.73% in HR@10 and 5.72% in NDCG@10. To the contrast of TCF, SACF also outperforms with 4.13% and 2.58% respectively.

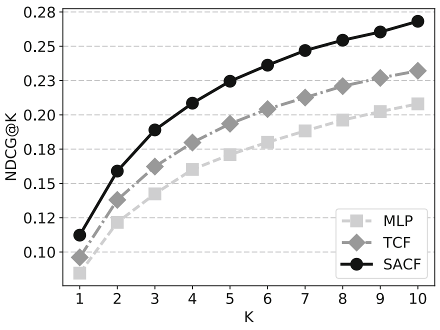
We also evaluate the performance in Top-K recommendation. We fix embedding size (ES) to 64 and latent factor (LF) size to 8. The results presented in Fig. 4 show that, both HR@K and NDCG@K follow the same trend, i.e. MLP<TCF<SACF, where K ranges from 1 to 10. These findings further reveal that SACF has significantly performance advantages over the baselines.

Table 3. Performance of HR@10 and NDCG@10 in different size of latent factors.

Algorithm	LF = 8		LF = 16		LF = 32		LF = 64	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
MLP	0.3746	0.2131	0.3701	0.2107	0.3484	0.1987	0.3504	0.2033
TCF	0.4168	0.2390	0.4306	0.2445	0.4116	0.2367	0.4205	0.2359
SACF	0.4688	0.2680	0.4719	0.2696	0.4718	0.2703	0.4626	0.2662



(a) **HR@K, ES=64, LF=8**



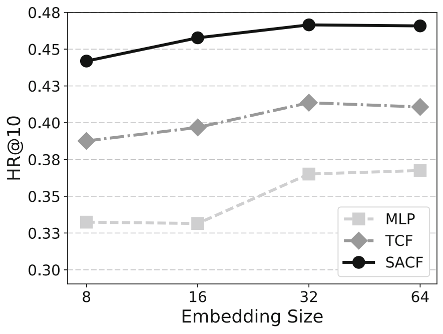
(b) **NDCG@K, ES=64, LF=8**

Fig. 4. Performance of Top-K recommendation where ranges K from 1 to 10.

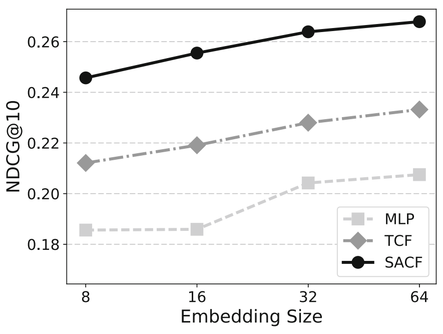
Table 4. Performance of HR@10 and NDCG@10 in Top-K recommendation.

Top-K	MLP		TCF		SACF	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
K = 1	0.0846	0.0846	0.0961	0.0961	0.1123	0.1123
K = 2	0.1435	0.1216	0.1623	0.1380	0.1877	0.1590
K = 3	0.1858	0.1424	0.2117	0.1623	0.2456	0.1890
K = 4	0.2243	0.1602	0.2526	0.1798	0.2922	0.2085
K = 5	0.2538	0.1710	0.2870	0.1935	0.3328	0.2245
K = 6	0.2805	0.1800	0.3184	0.2042	0.3666	0.2362
K = 7	0.3057	0.1883	0.3433	0.2126	0.3968	0.2469
K = 8	0.3291	0.1962	0.3671	0.2209	0.4209	0.2544
K = 9	0.3483	0.2024	0.3900	0.2269	0.4443	0.2604
K = 10	0.3746	0.2131	0.4168	0.2390	0.4688	0.2680

The detailed performance data of Top-K recommendation is presented in Table 4. Normally, as the value of K increases, all the performance indicators are gradually improving. At the same time, the performance gap between SACF and the other two baselines is gradually expanding.

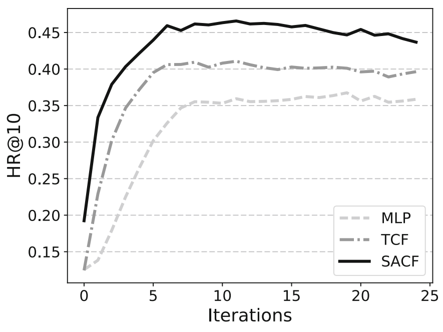


(a) **HR@10, LF=8**

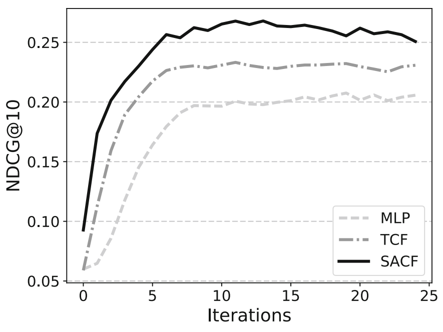


(b) **NDCG@10, LF=8**

Fig. 5. Performance comparison in different size of embedding vector.



(a) **HR@10, ES=64, LF=8**



(b) **NDCG@10, ES=64, LF=8**

Fig. 6. Recommendation performance in different number of iterations.

Experiment 2: Impact of Embedding Size. The size of embedding vector determines the feature description ability especially when we embed various kinds of input data. Towards this end, we further investigate the impact of embedding size, which is summarized in Fig. 5. The latent factor (LF) size is set to 8. The empirical evidence shows that the performance curves rise first and then stabilize. We speculate that increasing the embedding size can partly alleviate recommendation efficiency, but as the dimension continues to rise, it also brings the risk of overfitting and harms the recommendation performance.

Experiment 3: Performance Changes with Iterations. As the number of iterations increases, the parameters in the neural network will be updated numerous times, and the fitting effect goes from underfitting to overfitting. Figure 6 shows the recommendation performance of the algorithms of each iteration on our dataset. We can see that with more iterations, the performance first rises rapidly and then decreases gradually in both two metrics, which indicates that the more iterations may lead to overfitting.

5 Conclusion

In this work, we proposed an efficient recommendation algorithm called SACF for crowdsourced time-sync videos, which exploits the characteristics of the TSC data. By integrating the semantic embedding with collaborative filtering paradigm, SACF achieves much better performance compared to other algorithms on real datasets. In the future, we will continue to delve into this field. On the one hand, we can better model the user's interest based on the rich emoji data in TSCs. On the other hand, we can also infer the user's mood in real time according to the TSC that the user just sent. Such mood-aware data can definitely improve the performance in real-time recommender systems.

Acknowledgement. This work was supported by the National Key R&D Program of China under Grant 2018YFB0204100, the National Natural Science Foundation of China under Grant 61572538, Guangdong Special Support Program under Grant 2017TX04X148, Hong Kong Innovation and Technology Commission under Grant ITS/319/17, Australia Research Council under Grant DE180100950.

References

1. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 217–253. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_7
2. Bayer, I., He, X., Kanagal, B., Rendle, S.: A generic coordinate descent framework for learning from implicit feedback. In: Proceedings of the 26th International Conference on World Wide Web, pp. 1341–1350. International World Wide Web Conferences Steering Committee (2017)
3. Chen, X., Zhang, Y., Ai, Q., Xu, H., Yan, J., Qin, Z.: Personalized key frame recommendation. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 315–324. ACM (2017)
4. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 191–198. ACM (2016)
5. Ganu, G., Elhadad, N., Marian, A.: Beyond the stars: improving rating predictions using review text content. In: WebDB, vol. 9, pp. 1–6. Citeseer (2009)
6. He, M., Ge, Y., Chen, E., Liu, Q., Wang, X.: Exploring the emerging type of comment for online videos: Danmu. ACM Trans. Web (TWEB) **12**(1), 1 (2018)
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182. International World Wide Web Conferences Steering Committee (2017)
8. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 549–558. ACM (2016)
9. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 263–272. IEEE (2008)

10. Ikeda, A., Kobayashi, A., Sakaji, H., Masuyama, S.: Classification of comments on Nico Nico Douga for annotation based on referred contents. In: 2015 18th International Conference on Network-Based Information Systems (NBiS), pp. 673–678. IEEE (2015)
11. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. (TOIS) **20**(4), 422–446 (2002)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **8**, 30–37 (2009)
14. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
15. Liu, H., He, J., Wang, T., Song, W., Du, X.: Combining user preferences and user opinions for accurate recommendation. Electron. Commer. Res. Appl. **12**(1), 14–23 (2013)
16. Lv, G., Xu, T., Chen, E., Liu, Q., Zheng, Y.: Reading the videos: temporal labeling for crowdsourced time-sync videos based on semantic embedding. In: AAAI, pp. 3000–3006 (2016)
17. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: understanding rating dimensions with review text. In: Proceedings of the 7th ACM Conference on Recommender Systems, pp. 165–172. ACM (2013)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
19. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2008)
20. Ping, Q.: Video recommendation using crowdsourced time-sync comments. In: Proceedings of the 12th ACM Conference on Recommender Systems, pp. 568–572. ACM (2018)
21. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
22. Tan, Y., Zhang, M., Liu, Y., Ma, S.: Rating-boosted latent topics: understanding users and items with ratings and reviews. In: IJCAI, pp. 2640–2646 (2016)
23. Tang, D., Qin, B., Liu, T.: Learning semantic representations of users and products for document level sentiment classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1014–1023 (2015)
24. Tang, D., Qin, B., Liu, T., Yang, Y.: User modeling with neural network for review rating prediction. In: IJCAI, pp. 1340–1346 (2015)
25. Verbert, K., et al.: Context-aware recommender systems for learning: a survey and future challenges. IEEE Trans. Learn. Technol. **5**(4), 318–335 (2012)
26. Wu, B., Zhong, E., Tan, B., Horner, A., Yang, Q.: Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 721–730. ACM (2014)
27. Wu, Y., Ester, M.: Flame: a probabilistic model combining aspect based opinion mining and collaborative filtering. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 199–208. ACM (2015)

28. Xian, Y., Li, J., Zhang, C., Liao, Z.: Video highlight shot extraction with time-sync comment. In: Proceedings of the 7th International Workshop on Hot Topics in Planet-Scale mObile Computing and Online Social neTworking, pp. 31–36. ACM (2015)
29. Zhang, S., Yao, L., Sun, A.: Deep learning based recommender system: a survey and new perspectives. arXiv preprint arXiv:1707.07435 (2017)
30. Zhang, W., Yuan, Q., Han, J., Wang, J.: Collaborative multi-level embedding learning from reviews for rating prediction. In: IJCAI, pp. 2986–2992 (2016)