

# Your Tweets Reveal What You Like: Introducing Cross-media Content Information into Multi-domain Recommendation

Weizhi Ma, Min Zhang\*, Chenyang Wang, Cheng Luo, Yiqun Liu and Shaoping Ma

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China  
Beijing National Research Center for Information Science and Technology, Tsinghua University  
mawz14@mails.tsinghua.edu.cn, z-m@tsinghua.edu.cn

## Abstract

Cold start is a challenging problem in recommender systems. Many previous studies attempt to utilize extra information from other platforms to alleviate the problem. Most of the leveraged information is on-topic, directly related to users' preferences in the target domain. Thought to be unrelated, users' off-topic content information (such as user tweets) is usually omitted. However, the off-topic content information also helps to indicate the similarity of users on their tastes, interests, and opinions, which matches the underlying assumption of Collaborative Filtering (CF) algorithms. In this paper, we propose a framework to capture the features from user's off-topic content information in social media and introduce them into Matrix Factorization (MF) based algorithms. The framework is easy to understand and flexible in different embedding approaches and MF based algorithms. To the best of our knowledge, there is no previous study in which user's off-topic content in other platforms is taken into consideration. By capturing the cross-platform content including both on-topic and off-topic information, multiple algorithms with several embedding learning approaches have achieved significant improvements in rating prediction on three datasets. Especially in cold start scenarios, we observe greater enhancement. The results confirm our suggestion that off-topic cross-media information also contributes to the recommendation.

## 1 Introduction and Related Work

In the era of big data, recommender system plays an essential role in improving user experiences on the Internet, particularly in E-Commerce. A well-performing recommender system enables users to find their favorite products. Therefore, it is necessary to model how a user evaluates the products, namely rating prediction.

CF algorithms are widely used to tackle rating prediction problem with users' rating history in recommender sys-

tems. As a vital step in most CF algorithms to get the latent embeddings of users and items, MF achieves remarkable performance [Rennie and Srebro, 2005]. Later, discrete factorization is used to mend MF [Porteous *et al.*, 2008]. Ruslan and Andriy propose Probabilistic Matrix Factorization (PMF), where a probabilistic model is introduced [Mnih and Salakhutdinov, 2008]. Alex *et al.* describe a unified Bayesian approach, named as Collaborative Bayesian filtering (CoBaFi) [Beutel *et al.*, 2014]. However, these algorithms are challenged in the data sparsity problem when users or items rarely have rating history.

In another branch of research, content-based algorithms can overcome the obstacle of limited historical data, because they are driven by the attribute features of items and users, which are able to recommend items with little users' history information. According to this idea, several studies introduce content information, such as user reviews on products, into CF algorithms and achieve stronger results. Topic models, including Latent Dirichlet Allocation [Blei *et al.*, 2003], are applied to discover the aspects of different products from reviews. McAuley and Leskovec propose an algorithm named Hidden Factors and Hidden Topics (HFT) that unveils topics correlated with the 'hidden factors' of products and users [McAuley and Leskovec, 2013]. Besides, review sentiment is used in Diao *et al.*'s work [Diao *et al.*, 2014]. Reviews are helpful in tackling the problems in the recommendation.

Although additional information from recommender system enhances the prediction results of users with limited rating history, it cannot cope with the cold-start problem (new users without any rating history information). Hence some studies try to utilize cross-domain information (e.g. user's rating history in other domains) in transfer learning algorithms [Elkahky *et al.*, 2015; Zhang *et al.*, 2016b; Man *et al.*, 2017]. Some other studies try to introduce cross-media information, such as user relationships, user's meta-data and preferences in social media, into rating prediction. We focus on employing user's content information in social media in this study. Here are some related studies: based on the idea that users may have similar interests with their friends, Yang *et al.* propose Truster Matrix Factorization (TrustMF) [Yang *et al.*, 2013], where the integration of user relationships reinforces rating prediction. The compactness

\*Contact author

degree of user relationships is taken into consideration by [Wang *et al.*, 2017b]. Meanwhile, user’s preference in social media is mined for the recommendation. User’s ‘like’ history and other behavior are introduced into rating prediction algorithms [Yan *et al.*, 2015]. User metadata, such as user profile, is applied in [Deng *et al.*, 2013]. Recent deep learning solutions for cross-media information enhanced recommendation and achieve good performances [Zhang *et al.*, 2016a; Wang *et al.*, 2017a; He and Chua, 2017]. Encouraging results demonstrate that the cross data does facilitate user profiling and rating prediction. Not only new users, but also users with rich historical data in recommender system benefit from the introduction of the additional information.

It should be noticed that the cross-media features applied in previous studies are on-topic information, which directly related to users’ preferences in the target domain or characteristics of target items (e.g. user meta-data, ties, and behavior). Former studies have neglected the off-topic content information in social media since most of it is unrelated to target items or the target domain. For instance, most user tweets are off-topic and ignored in user preference mining. However, we consider that the tweets generated by users in social media are as valuable as other on-topic features, because they assist to learn the embedding of users for profiling. Therefore, user tweets are applied to rating prediction in this study. Our purpose is to confirm whether the omitted off-topic information is effective on improving the rating prediction results.

To shed light on this problem, we propose a framework to capture features from user’s off-topic content and include them in MF based algorithms. Within this framework, three feature extraction methods and four MF based algorithms are adopted. The experiments conducted on all yield achieve encouraging results, with further analysis enclosed. The main contributions are listed below:

- To the best of our knowledge, this is the first work that applies user tweets to rating prediction algorithms, most of the tweets are off-topic content information.
- We design a framework to introduce the content information from social media into MF based algorithms, in which user’s content features can be directly used to boost recommendation in multiple domains.
- The experimental results in multi-domain show that the off-topic content contributes to significant improvements on various state-of-the-art rating prediction algorithms with different embedding approaches. The proposed algorithms are practically advantageous when dealing with the cold start problem.

The remainder of this paper is organized as follows: A detailed description of our framework is stated in Section 2. Subsequently, in Section 3, we discuss experimental settings and the comparative results. Further analyses are shown in Section 4. The conclusions and the outline of future work are drawn in Section 5.

## 2 Social Media Content Enriched Framework

### 2.1 Framework

The task in this paper is rating prediction: to predict a user’s rating (an integer ranges from 1 to 5) on a product. The no-

Symbol	Definition
<b>R</b>	Rating matrix, $R_{u,i}$ is user $u$ ’s rating on item $i$
<b>U</b>	User matrix, $\vec{U}_u$ is user $u$ ’s latent vector
<b>I</b>	Item matrix, $\vec{I}_i$ is item $i$ ’s latent vector
$m, n$	The amount of users and items, respectively
$k$	The length of latent vector $\vec{U}_u$ and $\vec{I}_i$
$\mathbb{F}$	Function to learn the user embedding matrix <b>D</b>
$\mathbb{M}$	Function to train latent matrix <b>U</b> and <b>I</b>
<b>D</b>	User embedding matrix extracted from content.
<b>B</b>	User latent matrix which stores users’ domain specific features, $\vec{U}_u = [\vec{D}_u, \vec{B}_u]$

Table 1: Summary of Notations.

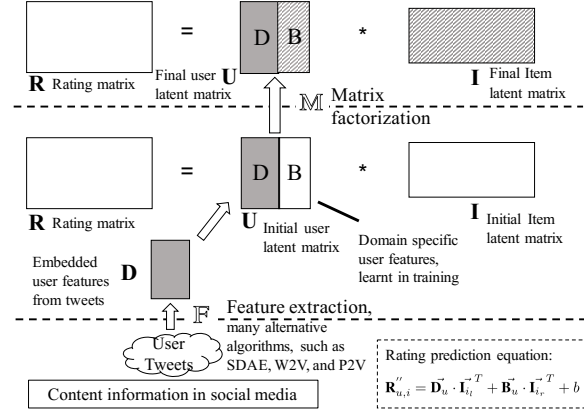


Figure 1: The framework of rating prediction with cross-media content information.

tations are summarized in Table 1. We aim to include user’s content information in social media into rating prediction. It then becomes a cross-media problem in two parts: 1) how to introduce the content-based features into rating prediction algorithms; 2) how to extract useful features from social media content. As shown in Figure 1, we propose a two-section framework accordingly: incorporated matrix factorization and feature extraction of content information.

For the former part of the problem, most rating prediction algorithms are based on MF, consisting of user latent matrix **U** and item latent matrix **I**. Inspired by previous studies, the user features  $\vec{D}_u$  learnt from social media content is introduced into user’s latent vector  $\vec{U}_u$  for MF training (the top part of Figure 1). The modified  $\vec{U}_u$  consists of two parts: user embeddings from the content information  $\vec{D}_u$  and domain-specific user features  $\vec{B}_u$ . **D** marked with grey means it is extracted from social media content. As **D** will affect the learning of **B** and **I**, the training results of them are marked with grey shadows to denote that they are affected by the cross-media information.

For the latter part of the problem, the length of  $\vec{U}_u$  in rating prediction algorithms ranges from 5 to 10 in most conditions. Thus,  $\vec{B}_u$  is supposed to be in low-dimension. The map function  $\mathbb{F}$  is used for user embedding learning, which can be implemented with various feature extraction methods. In our experiment, we examine multiple algorithms in this step.

Note that our main purpose is to investigate if the off-topic content information from social media can support stronger

performance of preceding algorithms in the rating prediction. We work with several embedding learning methods and apply them to different MF based algorithms. The two steps are elaborated in next subsections.

## 2.2 Incorporated Matrix Factorization

MF is widely used in rating prediction algorithms, in which rating matrix  $\mathbf{R}_{m \times n}$  is factorized as user matrix  $\mathbf{U}_{m \times k}$  and item matrix  $\mathbf{I}_{k \times n}$ . The predicted rating  $\mathbf{R}'_{u,i}$  that user  $u$  may give to item  $i$  is calculated by Equation 1.  $b$  is a bias factor. The objective function  $L_H$  is Equation 2, which aims to minimize the loss between the real rating  $\mathbf{R}_{u,i}$  and  $\mathbf{R}'_{u,i}$ .  $\|\cdot\|_F^2$  denotes the Frobenius norm for matrix and  $\lambda$  is the regularization parameter.

$$\mathbf{R}'_{u,i} = \vec{\mathbf{U}}_u \cdot \vec{\mathbf{I}}_i^T + b \quad (1)$$

$$L_{H1} = \sum_{(u,i) \in \text{Train}} (\mathbf{R}_{u,i} - \mathbf{R}'_{u,i})^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{I}\|_F^2) \quad (2)$$

As shown in Figure 1, user feature  $\mathbf{D}$  learnt from their content in social media, is introduced into MF based algorithms. We choose vector concatenation to fuse the embeddings here. The process to introduce  $\mathbf{D}$  into  $\mathbf{U}$  is as following:

- Initializing  $\mathbf{U}$ .  $\vec{\mathbf{B}}_u$  will be initialized with random normalization offsets before training and  $\vec{\mathbf{U}}_u = [\vec{\mathbf{D}}_u, \vec{\mathbf{B}}_u]$ .
- Using MF to train  $\mathbf{U}$  and  $\mathbf{I}$ .  $\vec{\mathbf{D}}_u$  will be kept while  $\vec{\mathbf{B}}_u$  is updated in training to get the domain specific features.

The new prediction result  $\mathbf{R}''_{u,i}$  is calculated by Equation 3, and the objective function is Equation 4.  $\vec{\mathbf{I}}_i$  is split into left part  $\vec{\mathbf{I}}_{i_l}$  and right part  $\vec{\mathbf{I}}_{i_r}$  in Equation 3.

$$\mathbf{R}''_{u,i} = \vec{\mathbf{D}}_u \cdot \vec{\mathbf{I}}_{i_l}^T + \vec{\mathbf{B}}_u \cdot \vec{\mathbf{I}}_{i_r}^T + b \quad (3)$$

$$L_{H2} = \sum_{(u,i) \in \text{Train}} (\mathbf{R}_{u,i} - \mathbf{R}''_{u,i})^2 + \lambda(\|\mathbf{D}, \mathbf{B}\|_F^2 + \|\mathbf{I}\|_F^2) \quad (4)$$

We do not replace  $\vec{\mathbf{U}}_u$  with  $\vec{\mathbf{D}}_u$  directly, because user's actions in recommender system can assist profiling too. Note that  $\vec{\mathbf{D}}_u$  is held in updating  $\mathbf{U}$ , so the content features are maintained and they will affect in training  $\mathbf{I}$  and  $\mathbf{B}$ . Meanwhile,  $\vec{\mathbf{B}}_u$  captures a user  $u$ 's features from his/her rating history. Consequently, the modified MF algorithm can profile a user's content features and rating features at the same time. If the off-topic content information contributes, results of rating prediction will get improved.

We name the as Social Media Content Enriched Matrix Factorization (MF\_S). Moreover, MF is common in various rating prediction algorithms, so it is convenient to apply the content information to other algorithms similarly. We will elaborate it in Section 3.2.

## 2.3 Feature Extraction of Content Information

Several embedding learning methods are adopted as  $\mathbb{F}$ , including Stacked Denoising Autoencoder (SDAE) [Vincent *et al.*, 2010], word2vector (W2V), and paragraph2vector (P2V).

The first method employs SDAE and bag-of-words (BOW), and it goes through two steps to learn  $\vec{\mathbf{D}}_u$ : 1) Learn

high-dimension embedding. The words in user content reveal his/her habits and contribute to profiling, so BOW is applied to learn a basic embedding. All of a user  $u$ 's tweets without stop words and signals are used in training. 2) Compress high-dimension representation to low-dimension. SDAE stands for its powerful ability to compress features, which consists of several Denoising Autoencoder (DAE). As shown in Equation 5 to 7, DAE is an unsupervised neural network which aims to minimize the MSE loss between the input data  $\vec{x}$  and the predicted  $\vec{x}'$ , and  $\vec{y}$  is the output. In this study, the input of SDAE is BOW embeddings and the output is  $\vec{\mathbf{D}}_u$ :

$$\vec{y} = \sigma(\mathbf{W}_{\text{encoder}} * \vec{x} + \vec{b}_{\text{encoder}}) \quad (5)$$

$$\vec{x}' = \sigma(\mathbf{W}_{\text{decoder}} * \vec{y} + \vec{b}_{\text{decoder}}) \quad (6)$$

$$L_e = \|\vec{x} - \vec{x}'\|^2 \quad (7)$$

The second method is based on W2V, a famous word embedding algorithm [Mikolov *et al.*, 2013]. It is handy to control the length of word embedding in training, so W2V is adopted to learn a low-dimension latent vector of each word.  $\vec{\mathbf{D}}$  is calculated by averaging the words in users' posts.

However, W2V only considers the embedding of each word rather than a sentence, the order of words is missed. Le *et al.* propose a P2V model that jointly learn word and document embeddings [Le and Mikolov, 2014]. So we take one's tweets as a document and use P2V to learn  $\vec{\mathbf{D}}_u$  from it. Similar to W2V, it is easy to adjust the embedding length.

The three methods are employed to get the user embedding matrix  $\mathbf{D}$ . In general condition, the other part of user embedding, matrix  $\mathbf{B}$ , is trained by MF introduced in Section 2.2. However, when coping with new users, due to the lack of rating history, MF cannot train the domain specific user matrix  $\mathbf{B}$ . In this scenario, we have  $\vec{\mathbf{U}}_u = [\vec{\mathbf{D}}_u, \vec{\mathbf{B}}_u]$  and  $\vec{\mathbf{B}}_u = \vec{\mathbf{0}}$ . Several methods are designed to complete matrix  $\mathbf{B}$ :

- $M_1$ : filling  $\vec{\mathbf{B}}_u$  with 0,  $\vec{\mathbf{B}}_u = \vec{\mathbf{0}}$ ;
- $M_2, M_3$ , and  $M_4$ : K-Nearest Neighbor based methods is applied.  $\vec{\mathbf{B}}_u$  is set as  $\vec{\mathbf{B}}_n$  if the similarity of  $\vec{\mathbf{D}}_u$  and  $\vec{\mathbf{D}}_n$  is the highest ( $n \in \text{training set}$ ). Eulerian distance, cosine similarity, and Manhattan distance are used to calculate the similarity and named as  $M_2, M_3$ , and  $M_4$ .
- $M_5$ :  $\vec{\mathbf{B}}_u$  is set as the averaged embedding of users in training set.

The cold start problem is a general obstacle in real scenarios. Facilitated by the additional information and user embedding learning methods, the modified algorithms can deal with new users.

## 2.4 Model Training

Our framework goes through two steps and there are two types of training methods, the loosely coupled and the tightly coupled. The former one conducts the two-step training one by one (to learn user embeddings and then apply them to MF), while the latter sets a global objective function to train the two parts together. Due to the limit of length, we only show the detailed tightly coupled training method of SDAE + MF here, which is named as MF\_S\_T.

As shown in Figure 2, SDAE and MF are trained together and the global loss function is Equation 8, which adds

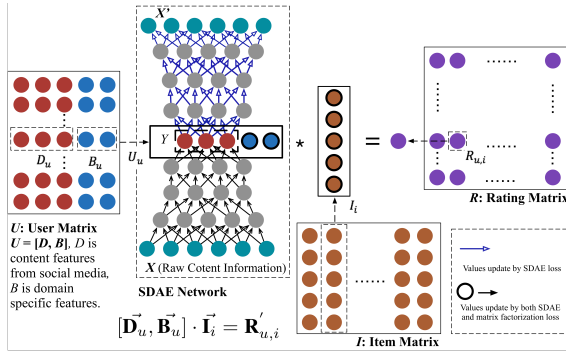


Figure 2: The tightly coupled training method of using SDAE and MF in our framework.

SDAE’s loss with weight  $\alpha$  into MF loss. The rating is predicted one by one and stochastic gradient descent (SGD) algorithm is used in model training.  $\vec{B}_u$  and  $\vec{I}_i$  are updated by the global loss. Because the neural network is back propagation, half weights are trained by the global loss and others are updated by SDAE loss (marked in Figure 2).  $\vec{D}_u$  will not be updated directly since it is learnt from SDAE.

$$L_{H_3} = \sum_{(u,i) \in \text{Train}} (\mathbf{R}_{u,i} - \mathbf{R}'_{u,i})^2 + \alpha \|\vec{x} - \vec{x}'\|^2 + \lambda (\|\mathbf{D}, \mathbf{B}\|_F^2 + \|\mathbf{I}\|_F^2) \quad (8)$$

Intuitively, tightly coupled method has stronger fitting ability so it is able to get better performance. While we prefer loosely coupled method because it is convenient to reuse the user embedding  $\vec{D}_u$  trained on user content into multi-domains without extra training. So the following experiments will learn  $\vec{D}_u$  in advance and then conduct MF. Tightly coupled based experiments will be reported in Section 4.2.

### 3 Experiments and Results

#### 3.1 Datasets

Two platform datasets of overlapped users, Douban<sup>1</sup> and Sina Weibo<sup>2</sup> (a social network like Twitter in China), are examined. These datasets are widely used in cross-media studies [Jiang *et al.*, 2016; Yuan *et al.*, 2013]. Users can rate three types of items in Douban: music, movie, and book. The ratings are integers ranging from 1 to 5 and some are accompanied by reviews. Tweets and user following relationships in Weibo are recorded. There are over 60,000 users in the dataset. Users who have less than 5 tweets in Weibo and 3 ratings in Douban are removed. 21,356 users remained and each user published 1,521 tweets on average. Table 2 lists the statistics of the three domains after filtering.

#### 3.2 Baseline Models and Evaluation Metric

MF, PMF, HFT, and TrustMF are four baseline algorithms. MF algorithm is illustrated in Section 2.2, and the introductions of other methods are as following:

- **PMF**: incorporates a probabilistic model into MF.

<sup>1</sup>www.douban.com

<sup>2</sup>www.weibo.com

	#User	#Item	#Rating
<b>Music</b>	11,216	76,307	1,144,193
<b>Movie</b>	14,298	30,807	2,541,243
<b>Book</b>	11,515	35,397	487,846

Table 2: The statistics of three domain datasets.

- **HFT**: discovers topics that are related to the ‘hidden factors’ of items and users from reviews.
- **TrustMF**: makes use of user ties in social media.

We modify above algorithms according to our framework that can cooperate with different MF based algorithms. Except for MF training part, we do not change the objective functions and prediction functions of them. The modified algorithms are named as PMF\_S, HFT\_S, and TrustMF\_S respectively. If the off-topic content in social media is indeed informative, the modified algorithms should get better performance than the originals. Similar to previous studies, we use Root Mean Square Error (RMSE) as the evaluation metric. Higher RMSE indicates worse performance.

#### 3.3 Setting of User Embedding Learning

The content information used here is user tweets in Weibo, most of which are off-topic. As introduced before,  $k$  is an essential parameter constraining the length of  $\vec{U}_u$ . We conduct some pilot experiments and find the best parameter setting is  $|\vec{U}_u| = 5$  and  $|\vec{D}_u| = 3$  in these datasets. The settings of user embedding learning methods are listed below:

1) BOW is calculated based on high-frequency words (top 4,000) as the input of SDAE. The output of SDAE is set as 3 according to  $|\vec{D}_u|$ . We use four DAE here (the widths are 800, 100, 20, and 3 respectively) and the 4th DAE’s output is  $\vec{D}_u$ . The batch size is 64 and the epoch is 25 in training; 2) The setting of W2V is the same as Mikolov’s work [Mikolov *et al.*, 2013]. Words with low frequency are removed (less than 3). To ensure the embedding short enough to be introduced into  $\vec{U}_u$ , the length of the word vector is  $|\vec{D}_u|$ ; 3) We apply Ai’s PV-DBOW model [Ai *et al.*, 2016] in P2V and the length of the embedding vector is the same as W2V.

#### 3.4 Rating Prediction Results

The modified algorithms incorporate users’ content features into MF, and reminders are the same as the originals. We tune the parameters (such as  $\lambda$ ) to achieve the best performance of algorithms. Prediction experiments are conducted with 5-fold cross-validation. The RMSE results are shown in Table 3.

Basic results show that TrustMF achieves the best performance among the original algorithms on Music domain and Book domain, indicating that user relationships do help in rating prediction. The performance of HFT algorithm is not as good as expected. It applies users’ reviews to construct the user and item matrices, while only 48.0% ratings have reviews and the average review length is only 5.76 words. Short and low quality reviews weaken the performance.

Apparently, with any one of the three embedding learning methods, social media content enriched rating prediction algorithms outperform the originals in three domains. Most of the improvements are statistically significant, expect for

Algorithm		Music	Movie	Book
MF		0.6862	0.7127	0.7698
MF_S	SDAE	<b>0.6784**</b>	<b>0.7125</b>	<b>0.7595**</b>
	W2V	0.6833**	0.7126	0.7643**
	P2V	0.6789**	0.7125	0.7626**
Best Improvement		1.15%	0.03%	1.36%
PMF		0.6881	0.7133	0.7480
PMF_S	SDAE	<b>0.6766**</b>	0.7102**	0.7391**
	W2V	0.6776**	<b>0.7101**</b>	<b>0.7390**</b>
	P2V	0.6776**	0.7102**	<b>0.7390**</b>
Best Improvement		1.70%	0.45%	1.20%
HFT		0.7842	0.7357	0.9632
HFT_S	SDAE	0.7635**	0.7244**	0.9472**
	W2V	0.7639**	0.7257**	0.9341**
	P2V	<b>0.7567**</b>	<b>0.7234**</b>	<b>0.9221**</b>
Best Improvement		3.51%	1.67%	4.27%
TrustMF		0.6751	0.7236	0.7438
TrustMF_S	SDAE	<b>0.6712**</b>	<b>0.7154**</b>	<b>0.7372**</b>
	W2V	0.6715**	0.7155**	0.7377**
	P2V	0.6732**	0.7165**	<b>0.7372**</b>
Best Improvement		0.58%	1.15%	0.90%

Table 3: The rating prediction results. We conduct z-test to evaluate the differences comparing with the original algorithms (E.g.: MF\_S vs. MF). \*\* means  $p < 0.01$ .

MF\_S in Movie domain (The reason is elaborated in Section 4.1). The degrees of improvements are distinct across the algorithms and the most remarkable improvements are ranging from 0.03% to 4.27%. TrustMF\_S algorithm yields the best performances on Music domain and Book domain, which integrates both social relationships and content features in prediction. PMF\_S outperforms in Movie domain, with constantly better performances compared to MF\_S. In these experiments, HFT\_S achieves the best enhancement.

Comparing the performance of different feature extraction methods, the efficiency of the methods varies in the prediction algorithms. In most conditions, W2V performs worse than P2V. It is due to that the order of words is considered in P2V. SDAE performs the best in most tasks for its two-step fine-grained embedding learning. Meanwhile, whatever embedding approaches are employed, these prediction algorithms achieve obviously better performances.

All of users' tweets are used in these algorithms, most of which are not directly related to the three domains (off-topic), but they are effective in prediction. We believe the reason is that user's tweets reveal his/her traits and hobbies. Similar to users' social ties, the embedding learnt from user tweets facilitate to identify users' preferences by other users with similar interests. Consequently the prediction results are improved. More detailed comparative analyses and case study are shown in Section 4.1 and 4.3.

To summarize, the integration of the social media content assists multiple rating prediction algorithms achieve good performances. The results also indicate that both on-topic and off-topic content (the latter has been omitted in previous studies) are valuable and useful for the recommendation.

Algorithm		Music	Movie	Book
MF_S + On-topic		0.6810	0.7131	0.7669
MF_S + All tweets		<b>0.6784**</b>	<b>0.7125</b>	<b>0.7595**</b>
PMF_S + On-topic		0.6776	0.7105	<b>0.7390</b>
PMF_S + All tweets		<b>0.6766*</b>	<b>0.7102</b>	0.7391
HFT_S + On-topic		0.7646	0.7243	0.9568
HFT_S + All tweets		<b>0.7635*</b>	<b>0.7244</b>	<b>0.9472**</b>
TrustMF_S + On-topic		0.6723	0.7159	0.7391
TrustMF_S + All tweets		<b>0.6712*</b>	<b>0.7154</b>	<b>0.7372*</b>

Table 4: The rating prediction results with on-topic tweets and all tweets. Significant differences in z-test are noted, \* means  $p < 0.05$  and \*\* means  $p < 0.01$ .

Algorithm	No tweets	On-topic	All tweets
MF_S	0.7328	0.7271 *	<b>0.7254</b> * †
PMF_S	0.7416	0.7303 *	<b>0.7265</b> * †
HFT_S	0.7818	0.7764 *	<b>0.7750</b> * †
TrustMF_S	0.7367	0.7314 *	<b>0.7291</b> * †

Table 5: The prediction results in the sparsified Movie dataset (Half ratings are removed). \*: comparing with no tweets ( $p < 0.01$ ); †: comparing with using only on-topic tweets ( $p < 0.01$ ). 'No tweets' records the performance of the original algorithms.

## 4 Analyses and Discussions

### 4.1 On-topic Tweets Vs. All tweets

Experiments have verified that user tweets are constructive in rating prediction, but we are not sure whether the improvements are solely contributed by a few on-topic tweets or not. A tweet is on-topic if it contains at least a music/movie/book title in datasets (some ambiguous titles are removed, such as *Love*, *Friend*). The percentages of on-topic tweets in music, movie, and book domains are 1.88%, 0.38%, and 2.63%. Since there are still some ambiguous titles, part of off-topic tweets are wrongly recognized as on-topic, so the ratios are higher than expected. Additional experiments are conducted to examine the performance when only on-topic tweets are used. SDAE is applied to extract user features for its good performance with the loosely coupled method. The results are listed in Table 4.

From the table, algorithms with all tweets outperform with only on-topic tweets, most of which are statistically significant in Music domain and Book domain. However, the improvements in Movie domain are limited. The sparsity of Movie domain (99.42%) is lower than others (99.86% in Music and 99.88% in Book), which means Movie dataset contains more user logs. A suggested reason is that there is already enough user logs in Movie dataset, so the addition of tweets is fruitless. To verify this assumption, half user-item ratings in Movie dataset are randomly removed (its sparsity is increased to 99.68%) and more experiments are conducted.

As shown in Table 5, the tweets boosted algorithms perform significantly better than the originals in the sparsified Movie dataset. The performances of different algorithms with all tweets are significantly better than with only on-topic tweets too. To conclude, both on-topic & off-topic tweets are effective and off-topic content should not be ignored in rating prediction. These results also indicate that the content infor-



Algorithm	Music	Movie_S	Book
MF_S	0.6784	0.7254	0.7595
MF_S_T	<b>0.6740**</b>	<b>0.7239**</b>	<b>0.7486**</b>
PMF_S	0.6766	0.7300	0.7391
PMF_S_T	<b>0.6648**</b>	<b>0.7186**</b>	<b>0.7283**</b>
TrustMF_S	0.6712	0.7291	0.7372
TrustMF_S_T	<b>0.6668**</b>	<b>0.7201**</b>	<b>0.7254**</b>

Table 6: The rating prediction results of tightly coupled methods. Significant differences in z-test are noted (E.g.: MF\_S vs. MF\_S\_T), \*\* means  $p < 0.01$ . Movie\_S means we use the sparsified Movie dataset.

mation in social media is more helpful in coping with sparse data, likewise in the cold start scenario.

## 4.2 Tightly Coupled Training Results

The experimental results of applying tightly coupled method to conduct framework training are introduced in this section. Similar to Section 2.4, we modify PMF\_S and TrustMF\_S based on this idea too, and the new algorithms are named as PMF\_S\_T and TrustMF\_S\_T. The best experimental results and are shown in Table 6, which are achieved with distinct values of  $\alpha$  ranges from 0.5 to 5.

As expected, the tightly coupled trained algorithms significantly outperform the loosely coupled in all conditions and beat the original algorithms easily for their stronger fitting ability with the global loss function. The results show that our framework delivers more accurate results with this training method. While these algorithms spend a longer time in training and user embedding  $\mathbf{D}_k$  is domain specific and cannot be applied to other domains. Nonetheless, we can still choose to use the loosely coupled method or the tightly coupled method according to specific scenarios.

## 4.3 Tweets Analysis: a Case Study

The experimental results show the effectiveness of the off-topic content information in promoting rating prediction. How do the tweets affect the prediction results? We look into the prediction results in Movie domain, and there are two tweets posted by  $U_1$  (Uid 1405524) and  $U_2$  (Uid 1305835):

$U_1$ : *I really like funny stories, which always makes me happy...*  $U_2$ : *Looking back into what I have seen in recent years, most of them are funny cartoons...*

The tweets indicate that  $U_1$  and  $U_2$  are likely to have a similar preference. It is easy to capture this preference by bag-of-words or other methods for collaborative filtering. Therefore, the off-topic tweets assist to enhance prediction results. In a word, user profiling based on the content features is performed efficiently in a hidden way for the recommendation.

## 4.4 Prediction Performance for New Users

An obvious advantage of employing cross-media information for rating prediction is that it will be effective on dealing with the cold start problem, especially for new users. The performance of MF\_S in handling new users is examined here. To get new users, we randomly sample 90% users as training set and the rest 10% as the test set in each dataset. The logs of

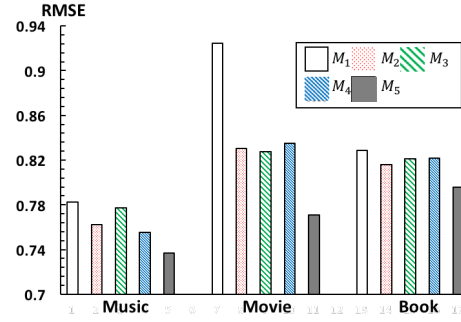


Figure 3: The prediction results in new users. The experimental results indicate that  $M_5$  performs significant better than others in each domain ( $p < 0.01$ ).

users in the test set are removed in training set. Most traditional algorithms cannot conduct prediction or recommendation for these users. However, with the help of cross-media information, MF\_S algorithm can learn user embedding from their tweets. We follow the setting of SDAE in Section 3.3. The methods  $M_1 - M_5$  to complete matrix  $\mathbf{B}$ , introduced in Section 2.3, are applied here.

Figure 3 shows the prediction results.  $M_5$  performs significantly better than other methods in multiple domains.  $M_1$  with the worst RMSE shows that using  $\bar{\mathbf{0}}$  is a bad idea. The three-similarity based methods perform worse than  $M_5$ . We attribute it to the sparsity of users: the scale of the training set is not large enough to find a user with same interests to the user  $u$ . Comparing the RMSE results of  $M_5$  with the results in Table 3, rating prediction results in new users are worse than in normal condition due to the lacking of user logs. Generally speaking, the performance of MF\_S in new users is not bad. RMSE value is only 0.051 higher than the results in predicting non-cold-start users on average. The incorporation of cross-media information enables the algorithms handle new users, which is essential for recommender systems.

## 5 Conclusions and Future Work

In this paper, we propose a novel and effective framework to leverage full content information in social media, including both on-topic and off-topic content, for rating prediction. The framework is easy to understand and implement, meanwhile shows great flexibility with different embedding approaches and different algorithms. It also achieves remarkable results in multiple domains with the modified algorithms. To the best of our knowledge, it is the first work that the off-topic content information is utilized and observed positive effects in the recommendation for rating prediction. In the new-user scenario, equipped with the cross-media content information, greater improvements have been achieved. Based on this study, we suggest that further studies should utilize full content information in cross-platform to the recommendation.

## Acknowledgements

This work is supported by Natural Science Foundation of China (Grant No. 61672311, 61532011). And this project is partially funded by the MOE-Microsoft Key Laboratory of Web and Multimedia, Tsinghua University.

## References

- [Ai *et al.*, 2016] Qingyao Ai, Liu Yang, Jiafeng Guo, and W. Bruce Croft. Improving language estimation with the paragraph vector model for ad-hoc retrieval. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 869–872, 2016.
- [Beutel *et al.*, 2014] Alex Beutel, Kenton Murray, Christos Faloutsos, and Alexander J Smola. Cobafi: collaborative bayesian filtering. In *Proceedings of the 23rd international conference on World wide web*, pages 97–108. ACM, 2014.
- [Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [Deng *et al.*, 2013] Zhengyu Deng, Jitao Sang, and Changsheng Xu. Personalized video recommendation based on cross-platform user modeling. In *Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2013.
- [Diao *et al.*, 2014] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM, 2014.
- [Elkahky *et al.*, 2015] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288. International World Wide Web Conferences Steering Committee, 2015.
- [He and Chua, 2017] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. 2017.
- [Jiang *et al.*, 2016] Meng Jiang, Peng Cui, Nicholas Jing Yuan, Xing Xie, and Shiqiang Yang. Little is much: Bridging cross-platform behaviors through overlapped crowds. In *AAAI*, pages 13–19, 2016.
- [Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.
- [Man *et al.*, 2017] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: an embedding and mapping approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2464–2470. AAAI Press, 2017.
- [McAuley and Leskovec, 2013] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Computer Science*, 2013.
- [Mnih and Salakhutdinov, 2008] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.
- [Porteous *et al.*, 2008] Ian Porteous, Evgeniy Bart, and Max Welling. Multi-hdp: A non parametric bayesian model for tensor factorization. In *Aaai*, volume 8, pages 1487–1490, 2008.
- [Rennie and Srebro, 2005] Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005.
- [Vincent *et al.*, 2010] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.
- [Wang *et al.*, 2017a] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Item silk road: Recommending items from information domains to social users. *arXiv preprint arXiv:1706.03205*, 2017.
- [Wang *et al.*, 2017b] Xin Wang, Steven CH Hoi, Martin Ester, Jiajun Bu, and Chun Chen. Learning personalized preference of strong and weak ties for social recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1601–1610. International World Wide Web Conferences Steering Committee, 2017.
- [Yan *et al.*, 2015] Ming Yan, Jitao Sang, and Changsheng Xu. Unified youtube video recommendation via cross-network collaboration. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 19–26. ACM, 2015.
- [Yang *et al.*, 2013] Bo Yang, Yu Lei, Dayou Liu, and Jiming Liu. Social collaborative filtering by trust. In *International Joint Conference on Artificial Intelligence*, pages 2747–2753, 2013.
- [Yuan *et al.*, 2013] Nicholas Jing Yuan, Fuzheng Zhang, Defu Lian, Kai Zheng, Siyu Yu, and Xing Xie. We know how you live: exploring the spectrum of urban lifestyles. In *Proceedings of the first ACM conference on Online social networks*, pages 3–14. ACM, 2013.
- [Zhang *et al.*, 2016a] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362. ACM, 2016.
- [Zhang *et al.*, 2016b] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. Multi-domain active learning for recommendation. In *AAAI*, pages 2358–2364, 2016.