2018 IEEE International Conference on Big Data and Smart Computing

A Temporal and Topic-Aware Recommender Model

Dandan Song*, Lifei Qin*, Mingming Jiang†, Lejian Liao*§
*School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
†China merchants bank, Shenzhen, China
{sdd,liaolj,qinlifei}@bit.edu.cn, jiangmingming@cmbchina.com

***Abstract*—Individuals' interests and concerning topics are generally changing over time, with a strong impact on their behaviors in social media. Accordingly, designing an intelligent recommender system which can adapt with the temporal characters of both factors becomes a significant research task. In this paper, we suppose that users' current interests and topics are transferred from the previous time step with a Markov property. Based on this idea, we focus on designing a dynamic recommender model based on collective factorization, named *Temporal and Topic-Aware Recommender Model* (TTARM), which can express the transition process of both user interests and relevant topics in fine granularity. It is a hybrid recommender model which joint *Collaborative Filtering* (CF) and *Content-based* recommender method, thus can produce promising recommendations about both existing and newly published items. Experimental results on two real life data sets from CiteULike and MovieLens, demonstrate the effectiveness of our proposed model.**

***Keywords*-Recommender System, Collaborative filtering, Matrix Factorization**

I. Introduction

Nowadays recommender systems are playing an extremely important role for people to find attractive items more accurately and efficiently. Making good recommendations to users is crucial for achieving better use experience, promoting products, and enhancing business values.

Recently, approaches [1]–[4] based on *Collaborative Filtering* (CF) have achieved big success in practice. However, the fact is that user's interests and relevant topics are changing over time, and those static CF methods usually can not track these variations to propose appropriate suggestions. Considering this drawback, many research strategies such as [5], [6] have been undertaken to introduce time feature into their methods. These previous works infer user interests by decaying weights of instances according to time, or analyzing their historical behaviors throughout the life span. However, they do not describe the interests transition process in fine granularity.

User topic feature is another crucial factor that influences the performance of recommender systems. In the past decade, with the advent of user-driven social medias that allow users to store resources (contents, bookmarks, comments, and others) and associate them with personalized words, an army of topic-based models [7]–[9] have been proposed. They implement probability-based methods such as LDA [10] to extract latent topics from available contents in the user or object space and then produce recommendations [11]. These methods achieve successfully solve the cold-start problem generally suffered by CF-based methods and enhancing performance of recommender systems.

In order to take both above observational factors (*temporal user interests* and *user topics*) into account, we propose a *Temporal and Topic-Aware Recommender Model*, namely TTARM, to model the transition process of user interests and topics over time.

The main contributions of our work are summarized as follows.

- Supposing that an individual user's current interests and topics are shifted from the previous time step, we first propose a *Temporal Recommender Model*, namely TRM, based on *Joint Past-Present Decomposition* and *Collaborative Filtering* with Markov property to learn temporal trend of user interests.
- By defining dynamic topic similarity between users and items over time and incorporating it into TRM, we design a *Temporal and Topic-Aware Recommender Model* (TTARM) with both temporal interests and topic information.
- We systematically conduct extensive experiments on two large real datasets from CiteULike and MovieLens to evaluate the performance of our proposed model. The experimental results demonstrate that our model consistently outperforms other competitive methods.

The remaining of this paper is organized as follows. We review related methods in Section II. Then introduce the details of our proposed recommender model and its learning algorithms in Section III. We demonstrate the performance our model with a series of experiments and discuss the results in Section IV. Finally, we make a conclusion and look into the future work in Section V.

II. Related Work

In this section, we review relevant research works, which focus on temporal and topic-involved recommendation respectively.

§ Lejian Liao is the corresponding author.

2375-9356/18/$31.00 ©2018 IEEE
DOI 10.1109/BigComp.2018.00067

A. Temporal Recommendation

Methods [12], [13] based on tensor factorization are developed to take time information into account. In [12], a Bayesian probabilistic tensor factorization model (BPTF) was proposed, ratings are represented as triples – (user, item, time). Their experimental results demonstrated that BPTF benefits from the introduced time feature, outperforms other static recommender methods on both Netflix and MovieLens datasets. Xiang et al. [14] proposed a Session-based Temporal Graph (STG) which incorporates temporal information to model long-term and short-term preferences simultaneously. Another time-based model timeSVD++ [6] , incorporating latent temporal components into SVD++ [3], achieves the state-of-the-art temporal model on the Netflix. However, all the above methods do not depict the evolving process of user interests over time.

Based on this assumption that user's current preference depends on his preference at previous time step, Zhang et al. [15] proposed temporal probabilistic matrix factorization (TMF) and its fully Bayesian treatment model (BTMF), by incorporating a transition matrix into the conventional matrix factorization methods. With the analogous consideration, Li et al. [16] defined the transition of user interests in a way to let the user feature in previous time step be the Dirichlet prior of that in the current time step.

B. Topic-involved Recommendation

To improve the cold start and rating sparse problem which generally exists in CF-based recommender systems, some collaborative filtering approaches incorporating topic features of users and items have been proposed. The work [17] proposed an algorithm for recommending scientific articles by combining the merits of traditional collaborative filtering and probabilistic topic model. Moreover, [18] proposed a location-content-aware topic model called LCARS for recommendation by learning the interest of each user and the local preference of each city by capturing item co-occurrence patterns as well as exploiting item contents.

However, those static models above fail to take into account the fact that topics are generally evolving over time. To address this issue, [19] proposed a temporal context-aware mixture model (TCAM) to model users' rating behaviors by both user-oriented topics (intrinsic interests) and time-oriented topics (general public's interests). Although this model incorporates the public's temporal interests and topics, the historical behaviors are analyzed throughout the life span, whose intrinsic interests transition process is not described in fine granularity.

III. The TTARM Model

In this section, we first propose our *Temporal Recommender Model* (TRM) in Section III-A. In Section III-B, we employ Topic Model to learn dynamic topic similarity between users and items, then introduce this temporal topic feature into TRM and derived our proposed model – *Temporal and Topic-Aware Recommender Model* (TTARM). The learning algorithm and prediction are given in Section III-C.

A. Temporal Recommender Model

Inspired by *Joint Past-Present Decomposition Model* [20], we proposed the *Temporal Recommender Model*, which suppose that users' current interests are transferred from the previous time step with a Markov property, then express the transition process of user interests in fine granularity.

Assuming that a collection of user-item ratings arrives continuously in batches. Each batch is represented by a data matrix $R^{(t)}$ of size $N_u^{(t)} \times N_i^{(t)}$, where $N_u^{(t)}$ is the number of users and $N_i^{(t)}$ is the number of items at time step t.

Analogous to the *Joint Past-Present decomposition model*, we derive the present decomposition at time t:

$$R^{(t)} \approx P^{(t)}Q^{(t)} \tag{1}$$

where $P^{(t)}$ has a size of $N_u^{(t)} \times N_f$ and $Q^{(t)}$ has a size of $N_f \times N_i^{(t)}$, with N_f represents the number of latent factors. Obviously, $P^{(t)}$ measures the extent of interests that users have on the corresponding factors, while $Q^{(t)}$ measures the extent to which items possess those factors. Usually, N_f is much smaller than $N_i^{(t)}$.

Since user interests vary over time, we make the assumption that users' *Present* interests (i.e. $P^{(t)}$) transmits from the previous interests (i.e. $P^{(t-1)}$). Although the observation data is dynamic, we assume that user interests evolve smoothly during one time period, and the current interests depend on the interests that appear in the previous time-slot, rather than the sequence of interests that preceded it. Therefore, it has a Markov property, and we correspondingly derive the *Past* decomposition of $R^{(t)}$ at time t:

$$R^{(t)} \approx S^{(t)}P^{(t-1)}Q^{(t)} \tag{2}$$

Given $P^{(t-1)}$, $S^{(t)}$ is a interest-transition matrix to capture how much the current users' interests distribution ($P^{(t)}$) linearly transmits from the previous one ($P^{(t-1)}$).

Accordingly, for each time step t, given $R^{(t)}$ and $P^{(t-1)}$, joint the above two decompositions, we derive:

$$\begin{cases} R^{(t)} \approx P^{(t)}Q^{(t)} \\ R^{(t)} \approx S^{(t)}P^{(t-1)}Q^{(t)} \end{cases} \tag{3}$$

This model is a combination of *Collaborative Filtering* and *Joint Past-Present* decomposition model, and we call it *Temporal Recommender Model* (TRM).

B. Topic-Aware Enhancement

The dynamic topic similarity between users and items is another crucial factor for users' rating behaviors. By incorporating this property into the TRM (Eq. (3)), we obtain:

$$\begin{cases} R^{(t)} \approx (1-\eta)P^{(t)}Q^{(t)} + \eta C^{(t)} \\ R^{(t)} \approx (1-\eta)S^{(t)}P^{(t-1)}Q^{(t)} + \eta C^{(t)} \end{cases} \tag{4}$$

where parameter $\eta \in [0,1]$ is used to balance the basic rating score and topic similarity factor $C^{(t)}$. $C_{ui}^{(t)}$, an element of matrix $C^{(t)}$ denoting the topic similarity extent between user u and item i at time step t, is defined as:

$$\begin{aligned} C_{ui}^{(t)} &= \frac{W_u^{(t)} \cdot W_i^{(t)}}{|W_u^{(t)}| \cdot |W_i^{(t)}|} \\ &= \frac{(\frac{1}{\sum_{j\in D_u^{(t)}} R_{uj}^{(t)}} \sum_{j\in D_u^{(t)}} (R_{uj}^{(t)} W_j^{(t)})) \cdot W_i^{(t)}}{|\frac{1}{\sum_{j\in D_u^{(t)}} R_{uj}^{(t)}} \sum_{j\in D_u^{(t)}} (R_{uj}^{(t)} W_j^{(t)})| \cdot |W_i^{(t)}|} \end{aligned} \tag{5}$$

where $|.|$ denotes norm of vector. $D_u^{(t)}$ is the set of items that user u rated at time step t, and $R_{uj}^{(t)}$ is the rating user u gived to item j at time step t. Thus we can define the topic distribute of user u by calculating the average topic distribution of all items which user u rated at time step t, refers to $\frac{1}{\sum_{j\in D_u^{(t)}} R_{uj}^{(t)}} \sum_{j\in D_u^{(t)}} (R_{uj}^{(t)} W_j^{(t)})$ in Eq. (5), where $R_{uj}^{(t)}$ serves as the weight of items' topic influence on user u. $W_j^{(t)}$ is the topic distribution of item j at time step t, which can be obtained by applying Topic Model on items' content.

We call Eq. (4) *Temporal and Topic-Aware Recommender Model* (TTARM). Obviously, it leads to *Temporal Recommender Model* when $\eta = 0$ and pure topic-oriented recommender model when $\eta = 1$. The learning methods of TRM and TTARM are given in Section III-C.

C. Derived Algorithm

In order to learn *Temporal and Topic-Aware Recommender Model*, loss function $\mathcal{L}(R^{(t)}; P^{(t)}; Q^{(t)}; S^{(t)}; P^{(t-1)})$ for Eq. (4) needs to be specified. Consulting the work in [20], the following loss function is defined:

$$\begin{aligned} \mathcal{L} = \underset{S^{(t)},P^{(t)},Q^{(t)}}{\arg\min} & \|R^{(t)} - [(1-\eta)P^{(t)}Q^{(t)} + \eta C^{(t)}]\|_F^2 \\ & + \|R^{(t)} - [(1-\eta)S^{(t)}P^{(t-1)}Q^{(t)} + \eta C^{(t)}]\|_F^2 \\ & + \alpha\|P^{(t)}\|_1 + \beta\|Q^{(t)}\|_1 + \gamma\|S^{(t)}\|_1 \\ & + \lambda\|S^{(t)} - I\|_F^2 \\ & \text{subject to } P^{(t)} \geq 0, Q^{(t)} \geq 0, S^{(t)} \geq 0 \end{aligned} \tag{6}$$

where $\|.\|_F$ represents the Frobenius norm and $\|.\|_1$ stands for the L1 norm. The temporal regularization $\lambda\|S^{(t)} - I\|_F^2$ controls how much we want to bias the decomposition towards $P^{(t-1)}$. The λ parameter $\in (0, \infty)$ balances present and past information; it quantifies the extent to which the model is past (i.e. $\lambda \to \infty$) or present oriented (i.e. $\lambda \to 0$).

Our goal is to minimize the loss function in Eq. (6), but it is not convex for all parameters $P^{(t)}$, $Q^{(t)}$, $S^{(t)}$ simultaneously. Learning from [20], [21], a local minimum for the objective function could be reached using multiplicative updates.

First, considering the Karush-Kuhn-Tucker (KKT) first-order conditions applied to our problem, we derive:

$$\begin{cases} P^{(t)} \odot \nabla_{P^{(t)}}\mathcal{L} = 0, \quad P^{(t)} \geq 0, \quad \nabla_{P^{(t)}}\mathcal{L} \geq 0 \\ Q^{(t)} \odot \nabla_{Q^{(t)}}\mathcal{L} = 0, \quad Q^{(t)} \geq 0, \quad \nabla_{Q^{(t)}}\mathcal{L} \geq 0 \\ S^{(t)} \odot \nabla_{S^{(t)}}\mathcal{L} = 0, \quad S^{(t)} \geq 0, \quad \nabla_{S^{(t)}}\mathcal{L} \geq 0 \end{cases} \tag{7}$$

where $\odot$ is the element-wise product.

According to the loss function in Eq. (6), the gradients for each parameter are derived respectively:

$$\begin{aligned} \nabla_{P^{(t)}}\mathcal{L} = {} & 2P^{(t)}[(1-\eta)^2 Q^{(t)} {Q^{(t)}}^T + \alpha I] \\ & -2[(1-\eta)R^{(t)} - \eta(1-\eta)C^{(t)}]{Q^{(t)}}^T \end{aligned} \tag{8}$$

$$\begin{aligned} \nabla_{Q^{(t)}}\mathcal{L} = {} & 2(1-\eta){P^{(t)}}^T[\eta C^{(t)} + (1-\eta)P^{(t)}Q^{(t)}] \\ & + 2(1-\eta){P^{(t-1)}}^T {S^{(t)}}^T \\ & \cdot [\eta C^{(t)} + (1-\eta)S^{(t)}P^{(t-1)}Q^{(t)}] \\ & - 2(1-\eta)({P^{(t)}}^T + {P^{(t-1)}}^T {S^{(t)}}^T)R^{(t)} \\ & + 2\beta Q^{(t)} \end{aligned} \tag{9}$$

$$\begin{aligned} \nabla_{S^{(t)}}\mathcal{L} = {} & 2(1-\eta)[(1-\eta)S^{(t)}P^{(t-1)}Q^{(t)} + \eta C^{(t)}] \\ & \cdot {Q^{(t)}}^T {P^{(t-1)}}^T + 2(\lambda+\gamma)S^{(t)} \\ & - 2[(1-\eta)R^{(t)}{Q^{(t)}}^T {P^{(t-1)}}^T + \lambda I] \end{aligned} \tag{10}$$

By substituting the corresponding gradients in Eq. (7), the following update rules are obtained:

$$P^{(t)} = P^{(t)} \odot \frac{[(1-\eta)R^{(t)} - \eta(1-\eta)C^{(t)}]{Q^{(t)}}^T}{P^{(t)}[(1-\eta)^2 q^{(t)} {Q^{(t)}}^T + \alpha I]} \tag{11}$$

$$\begin{cases} Y = (1-\eta)\{{P^{(t)}}^T[\eta C^{(t)} + (1-\eta)P^{(t)}Q^{(t)}] \\ \quad + {P^{(t-1)}}^T {S^{(t)}}^T [\eta C^{(t)} \\ \quad + (1-\eta)S^{(t)}P^{(t-1)}Q^{(t)}]\} + \beta Q^{(t)} \\ Q^{(t)} = Q^{(t)} \odot \dfrac{(1-\eta)({P^{(t)}}^T + {P^{(t-1)}}^T {S^{(t)}}^T)R^{(t)}}{Y} \end{cases} \tag{12}$$

$$\begin{cases} Z = (1-\eta)[(1-\eta)S^{(t)}P^{(t-1)}Q^{(t)} + \eta C^{(t)}] \\ \quad \cdot {Q^{(t)}}^T {P^{(t-1)}}^T + (\lambda+\gamma)S^{(t)} \\ S^{(t)} = S^{(t)} \odot \dfrac{(1-\eta)R^{(t)}{Q^{(t)}}^T {P^{(t-1)}}^T + \lambda I}{Z} \end{cases} \tag{13}$$

The update Eqs. (11,12,13) lead to the algorithm which learns the *Temporal and Topic-Aware Recommender Model* (TTARM) as below:

IV. Experiments

In this section, to demonstrate the performance of our proposed recommender model, extensive experiments are conducted and results are analyzed. At first, we introduce the datasets, evaluation metrics and experimental settings, then present the main findings.

Algorithm 1 Learning algorithmns for TTARM

$P^{(t)}, Q^{(t)}, S^{(t)} \leftarrow$ random initialized non-negative
$\delta' \leftarrow maxInt, \delta \leftarrow \frac{\delta'}{2}$
while $abs(\delta' - \delta) \geq \epsilon$ **do**
  Update $P^{(t)}, Q^{(t)}, S^{(t)}$ by Eqs. (11,12,13) respectively.
  $\delta' \leftarrow \delta$
  $\delta \leftarrow \mathcal{L}(R^{(t)}, P^{(t)}, Q^{(t)}, S^{(t)}, P^{(t-1)})$
end while
return $P^{(t)}, Q^{(t)}, S^{(t)}$

A. Datasets

We evaluate our mathod and comparative methods on two large real life datasets — **CiteULike**[1] and **MovieLens**[2].

- **CiteULike**. This dataset is collected from CiteULike, which is a web service allows users to save and share citations to academic papers. The related articles' abstracts are provided by [17]. The time span is from November 4th in 2004 to April 16th in 2014. After merging duplicate articles and empty articles, we remove users with library's size less than 10 and articles with less than 10 "like".
- **MovieLens**. Ratings in this dataset are collected from MovieLens website, where users can give a rating to all movies. There are 2113 users, 9801 items and 824600 ratings(varying from 1 to 5) from Nov.1997 to Dec.2008. Similar to CiteULike dataset, we remove movies with less than 10 ratings and users who have rated less than 10 movies.

The statistics of the both preprocessed dataset are listed in Table I.

Table I
STATISTICS OF THE CITEULIKE DATASET

Dataset	CiteULike	MovieLens
#Rating	109,052	814,589
#User	3386	2113
#Item	7695	6624
#Avg.user-tag	32.20	385.5
#Avg.item-tagged	14.17	122.97
Sparse rate	99.58%	94.18%
Timespan	2004.11-2014.4	1997.11-2008.12

B. Evaluation Metrics

For each user u, we predict his ratings at items which he has not rated before the current time step t, then recommend the top-k items. If a recommended item is liked by the user u at time step t according to the test set, we call it a "hit" item, otherwise a "miss" item [19]. We use the following three

[1]http://www.citeulike.org/faq/data.adp
[2]http://www.grouplens.org/node/12

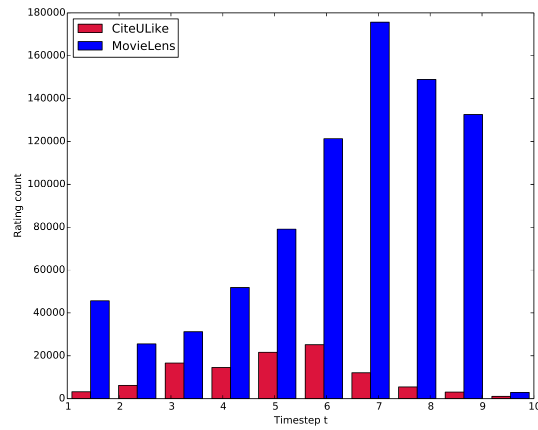


Figure 1. CiteULike and MovieLens datasets' rating distribution on 10 time steps

well-known metrics to evaluate the results. First, *Recall@k*:

$$Recall@k = \frac{N(hits)}{N(items)}$$

where $N(hits)$ is the number of "hit" items in the top-k recommended items, and $N(items)$ is the number of all items in the test set of user u. Obviously, a high recall with lower k indicates a better recommender system.

Concerning that Recall can not reflect the position importance of "hit" items in the ranked list, we also use *NDCG@k*:

$$NDCG@k = \frac{1}{IDCG} \times \sum_{i=1}^{k} \frac{2^{r_i} - 1}{log(i+1)}$$

where r_i is 1 if the item at position i is a "hit" item and 0 otherwise. IDCG is chosen for the purpose of normalization so that the perfect ranking has an NDCG value of 1 [19].

Summarizing the user-oriented metrics above, we have the the average metric value of all users:

$$Metric@k = \frac{\sum_{i=1}^{N} M_i@k}{N}$$

where N is the number of users, $M_i@k$ is the metric value for user i at position k, and the metric refers to *Recall*, *NDCG*.

C. Comparative Methods

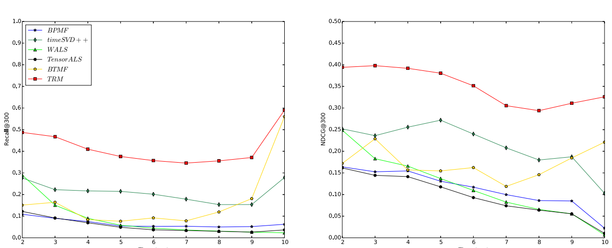
In order to analyze the performance of our proposed models, we design the comparison experiments between the following algorithms, including our **TRM** (in Section III-A) and **TTARM** (in Section III-B) models.

- **BPMF**. A fully Bayesian treatment of the Probabilistic Matrix Factorization (**PMF** [22]) model in which model capacity is controlled automatically byintegrating over all model parameters and hyperparameters [4].

- **timeSVD++**. A temporal recommender model which extends the **SVD++** [3] by introducing a time-variant bias for each user and item at every individual time step.
- **WALS**. A simple extension for Alternating Least Squares (ALS) where each user/item pair has an additional weight, which is a tensor algorithm since besides of the rating it also maintains a weight for each rating [23].
- **TensorALS**. A temporal recommender algorithm based on tensor factorization and alternating least squares, which considers time step as the third dimension [13].
- **BTMF**. A temporal bayesian probabilistic matrix factorization model (BTMF) [15], which incorporates a transition matrix into the conventional matrix factorization methods.
- **TRM**. The *Temporal Recommender Model* proposed in Section III-A of this paper without topic feature.
- **TTARM**. The *Temporal and Topic-Aware Recommender Model* proposed in Section III-B of this paper, where parameter $\eta \in (0, 1)$ implies a combination of temporal user interests and topics.

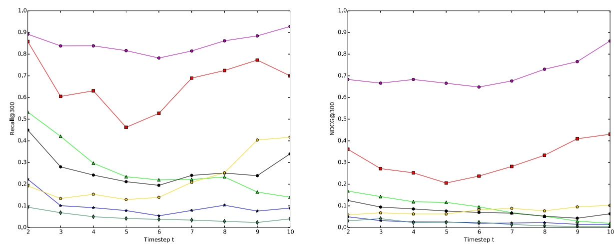
D. Experimental Setup

For each dataset, we create 10 time steps by splitting ratings yearly and merging extra rating to closest time step. TTARM can only run on CiteULike since the items' content information is unavailable on MovieLens. In addition, we set the number of latent factors D to be 20 for all latent factor models participated in competition. The parameter λ is set to be 10 for TRM and TTARM, the topic balance parameter η is set to be 0.3 and the topic number is set to be 50 for TTARM. Other parameters related to norm in loss function are set to be 0.05. According to Section III-A and III-B, both TRM and TTARM are launched by given the $P^{(t-1)}$ matrix. Therefore, in order to acquire a boot-able $P^{(t-1)}$, we apply NMF at the first time step and launch all comparative models at the second time step. That is why we only show the performance for time steps 2 to 10. Further more, 5-fold cross-validation is adopted to reduce the variance of all models' performance estimates and obtain more convincing experimental results.



(a) Recall@300 (b) NDCG@300

Figure 2. Metric@300 on MovieLens dataset



(a) Recall@300 (b) NDCG@300

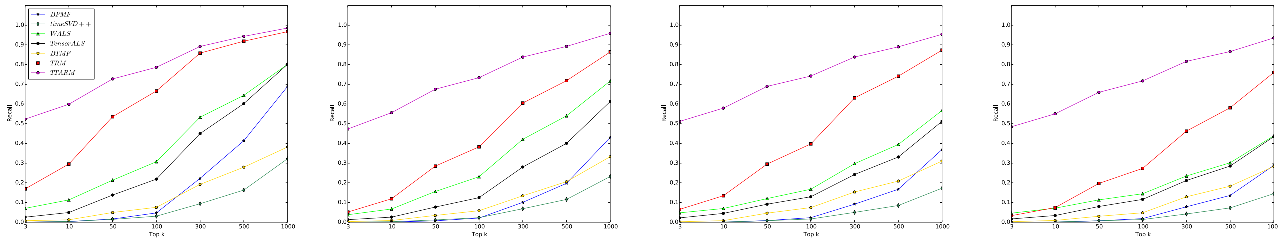
Figure 3. Metric@300 on CiteULike dataset

E. Experimental Results

The experimental results on both datasets are shown in Figure 2 and Figure 3 with Recall@300 and NDCG@300 from time step 2 to 10. It is obvious that both our TRM and TTARM outperform the other comparative recommender models consistently in these evaluation metrics, indicating how temporal feature of user interests can remarkably contribute to the performance of recommender system. The fact that TTARM outperforms TRM at all time steps implies that incorporating the temporal topic similarity between users and articles into our model achieves further improvement. Moreover, the introduction of similarity between topics benefits solving the cold start problem and the sparsity of dataset.

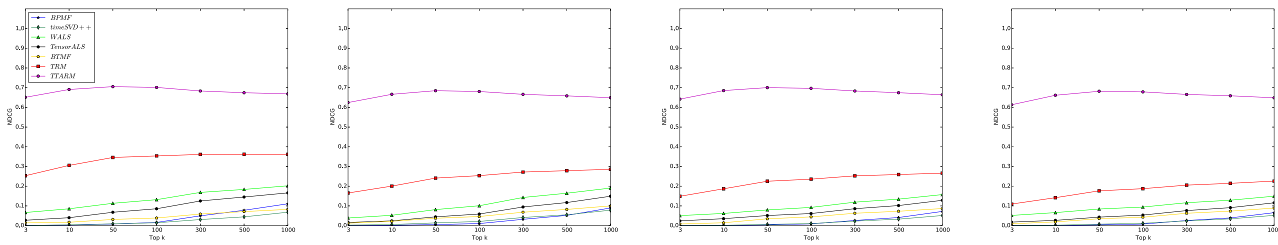
In Figure 3, TTARM and TRM's performance decreases at time steps 2-5 and rises after time step 6, while WALS, timeSVD++, BPMF's performance decreases almost across all time steps, BTMF and TensorALS's recall performance rises after time step 6. And in Figure 2, TRM decreases a little from time step 2 to 8 and rises afterward and BTMF has the similar trend across time steps. In addition, timeSVD++ beats other comparative methods at most time steps, and BTMF starts to show its performance after time step 7. Overall, our models (i.e., TRM and TTARM) show their steady and much better performance, which demonstrate the distinguishing ability of our methods in learning and predicting the variations of user interests and topics.

In details, Figure 4, 5, 6, 7 show results at 2-5 time steps, whose experimental results are similar. In each figure, there are four subfigures respectively demonstrate models' performance at time step 2-5, and the x axis is the k in *Recall*@k and *NDCG*@k. From the results, we can find that for each method, the trend of its performance over time is analogous under the two metrics, which contributes to the authorities of experimental results. It is obvious that both our TRM and TTARM outperform the other recommender models (i.e. BPMF, timeSVD++, TensorALS, BTMF and WALS) consistently in terms of the two evaluation metrics (i.e., *NDCG*@k and *Recall*@k). This comparison result is an indication of how temporal feature of user interests can remarkably contribute to the performance of recommender



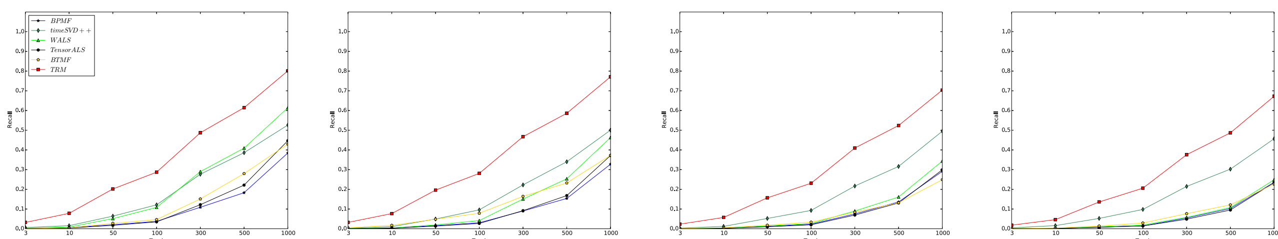
(a) Time step 2 (b) Time step 3 (c) Time step 4 (d) Time step 5

Figure 4. Recall@k for all comparative models on CiteULike dataset at time step 2-5 and vary k from 3 to 1000. Higher values are better.



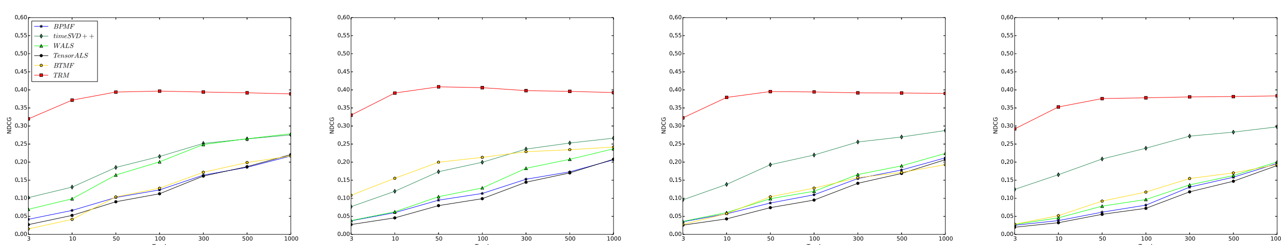
(a) Time step 2 (b) Time step 3 (c) Time step 4 (d) Time step 5

Figure 5. NDCG@k for all comparative models on CiteULike dataset at time step 2-5 and vary k from 3 to 1000. Higher values are better.



(a) Time step 2 (b) Time step 3 (c) Time step 4 (d) Time step 5

Figure 6. Recall@k for all comparative models on MovieLen dataset at time step 2-5 and vary k from 3 to 1000. Higher values are better.



(a) Time step 2 (b) Time step 3 (c) Time step 4 (d) Time step 5

Figure 7. NDCG@k for all comparative models on MovieLen dataset at time step 2-5 and vary k from 3 to 1000. Higher values are better.

Table II

NDCG@100 PERFORMANCE (MEAN ± STANDARD ERROR) COMPARISON OF TRM, TTARM AND COMPARATIVE METHODS. THE BEST PERFORMER IS IN **BOLDFACE** AND THE SECOND IS IN *italic*. THESE EXPERIMENTAL MEAN, STANDARD ERROR RESULTS ARE STATISTICS ON 5 TIMES EXPERIMENTS WITH SAME PARAMETERS.

	CiteULike				
Algorithm	Time step 2	Time step 3	Time step 4	Time step 5	Time step 6
BPMF	0.0161 ± 0.0040	0.0104 ± 0.0024	0.0089 ± 0.0040	0.0078 ± 0.0003	0.0075 ± 0.0004
timeSVD++	0.0144 ± 0.0018	0.0211 ± 0.0009	0.0104 ± 0.0008	0.0117 ± 0.0005	0.0117 ± 0.0015
WALS	0.1316 ± 0.0092	0.1008 ± 0.0070	0.0924 ± 0.0005	0.0942 ± 0.0102	0.0700 ± 0.0069
TensorALS	0.0856 ± 0.0029	0.0589 ± 0.0039	0.0613 ± 0.0043	0.0535 ± 0.0032	0.0454 ± 0.0074
BTMF	0.0386 ± 0.0000	0.0462 ± 0.0000	0.0442 ± 0.0000	0.0425 ± 0.0000	0.0637 ± 0.0000
TRM	*0.3535* ± 0.0017	*0.2537* ± 0.0019	*0.2358* ± 0.0044	*0.1874* ± 0.0023	*0.2212* ± 0.0020
TTARM	**0.7013** ± 0.0000	**0.6805** ± 0.0000	**0.6967** ± 0.0000	**0.6788** ± 0.0000	**0.6610** ± 0.0000
	MovieLens				
BPMF	0.1231 ± 0.0015	0.1133 ± 0.0023	0.1096 ± 0.0023	0.0810 ± 0.0018	0.0751 ± 0.0037
timeSVD++	*0.2156* ± 0.0000	0.1996 ± 0.0000	*0.2197* ± 0.0001	*0.2385* ± 0.0001	*0.2044* ± 0.0000
WALS	0.2008 ± 0.0022	0.1285 ± 0.0026	0.1193 ± 0.0010	0.0965 ± 0.0022	0.0732 ± 0.0011
TensorALS	0.1125 ± 0.0035	0.0988 ± 0.0016	0.0950 ± 0.0026	0.0721 ± 0.0038	0.0586 ± 0.0026
BTMF	0.1278 ± 0.0006	*0.2132* ± 0.0044	0.1281 ± 0.0196	0.1172 ± 0.0049	0.1341 ± 0.0027
TRM	**0.3966** ± 0.0032	**0.4061** ± 0.0024	**0.3943** ± 0.0032	**0.3781** ± 0.0016	**0.3469** ± 0.0020

system. The fact that TTARM outperforms TRM at all time steps implies that incorporating the temporal topic similarity between users and items into our model achieves great success. Moreover, the introduction of similarity between topics benefits solving the cold start problem and the sparsity of dataset.

TensorALS and WALS's performance is very close and better than timeSVD++, BPMF and BTMF on CiteULike dataset at time step 2-5 since they both are based on alternating least squares method (ALS). However, timeSVD++ beats TensorALS and WALS on MovieLens dataset. This phenomenon indicates that timeSVD++, TensorALS and WALS's performance depends on dataset more or less. Simultaneously, static methods, like BPMF, simply use all previous ratings will dismiss user interests' transmit over time step and performs not that well in our experiments.

TTARM beats all other methods both in Figure 4 and 5 on the CiteULike dataset. The items in this dataset are research articles, which have tight relation with topics, hence incorporating topic to TTARM contributes a lot. However, our proposed model TRM which doesn't introduce topic feature beats all other models under both metrics and datasets as well. These evidences suggest that considering the effect of temporal information and topic features contributes to the performance of recommendation.

We also investigate the trend of comparative models' performance under NDCG by adjusting the number of recommendations, i.e., the parameter k in *NDCG@k*. Figure 5 shows the influence of k on the seven comparative models by considering *NDCG@k* on CiteULike dataset. We can see that at time steps 2-5 when the parameter k grows, the performance of our TTARM model is stable, while TRM and all comparative methods increase slightly, illustrating that their recommended items are becoming accurate in ranks, as NDCG is related with the position of "hit" items in the ranked list. Figure 7 shows six models' NDCG@k performance on MovieLens dataset at time steps 2-5. We observe that with k growing, TRM's performance trends to a stable status, while other models increase all the time.

In Table II, we show comparative models' NDCG@100 performance on CiteULike and MovieLens datasets at time steps 2-6. The boldface and italic highlight the best and second best performers, respectively. Each value in this table is the mean value and standard error, which calculated by running 5 times experiments with same parameters. On CiteULike dataset, our TTARM model performs best since it introduces the topic feature of items and can track user interests over time. Moreover, TRM beats other comparative methods all the time. TTARM can not run on MovieLens dataset for it needs items' content information, which this dataset doesn't involve. However, our TRM model also achieve the best performance. In addition, timeSVD++ performs much better than BPMF, WALS, TensorALS and BTMF at time step 2, 4, 5, 6. However, BTMF beats timeSVD++ at time step 3.

In summary, the proposed TTARM and TRM outperform comparative models in most cases. Our models can learning user interests and topic feature very accurately and this fact contributes to the performance of recommender system.

V. CONCLUSION

In this paper, we propose a *Temporal and Topic-Aware Recommender Model* (TTARM), based on collective factorization to model temporal user interests and dynamic topic similarity over time for the purpose of making a better recommendation at current time. After incorporating topic similarity, the designed TTARM method is a hybrid recommender model which joints *Collaborative Filtering* (CF) and *Content-based* recommender methods and can form promising recommendations about both existing and newly published items. By applying on two large datasets (i.e. CiteULike and MovieLens), our proposed model outperforms competitive recommender algorithms, which demonstrates that temporal user interests and topic similarity features are crucial factors in recommender systems.

There are still several factors worthy of taking into account in the future. Referring to [19], user interests can be divided into user intrinsic and public interests in TTARM in the future. In addition, learning parameters automatically, incorporating the social information between users into TTARM, and visualization of user interests and dynamic topics are also very interesting works, we will study them in the future.

Acknowledgment

This work has been supported by the National Key Research and Development Program of China under grant (Grant No. 2016YFB1000902), and the National Natural Science Foundation of China (Grant No. 61472040).

References

[1] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.

[2] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 75–79, 2007.

[3] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[4] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 880–887.

[5] Y. Ding and X. Li, "Time weight collaborative filtering," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 485–492.

[6] Y. Koren, "Collaborative filtering with temporal dynamics," *Communications of the ACM*, vol. 53, no. 4, pp. 89–97, 2010.

[7] X. Si and M. Sun, "Tag-lda for scalable real-time tag recommendation," *Journal of Computational Information Systems*, vol. 6, no. 1, pp. 23–31, 2009.

[8] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 61–68.

[9] A. S. R. W. W. Umbrath and L. Hennig, "A hybrid plsa approach for warmer cold start in folksonomy recommendation," *Recommender Systems & the Social Web*, pp. 10–13, 2009.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[11] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.

[12] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization." in *Proceedings of the SIAM International Conference on Data Mining, SDM*, vol. 10. SIAM, 2010, pp. 211–222.

[13] P. Comon, X. Luciani, and A. L. De Almeida, "Tensor decompositions, alternating least squares and other tales," *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 393–405, 2009.

[14] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun, "Temporal recommendation on graphs via long- and short-term preference fusion," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 723–732.

[15] C. Zhang, K. Wang, H. Yu, J. Sun, and E.-P. Lim, "Latent factor transition for dynamic collaborative filtering," in *Proceedings of the SIAM International Conference on Data Mining, SDM*, vol. 14, 2014.

[16] B. Li, X. Zhu, R. Li, C. Zhang, X. Xue, and X. Wu, "Cross-domain collaborative filtering over time," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*. AAAI Press, 2011, pp. 2293–2298.

[17] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 448–456.

[18] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "Lcars: A location-content-aware recommender system," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 221–229.

[19] H. Yin, B. Cui, L. Chen, Z. Hu, and Z. Huang, "A temporal context-aware model for user behavior modeling in social media systems," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 1543–1554.

[20] C. K. Vaca, A. Mantrach, A. Jaimes, and M. Saerens, "A time-based collective factorization for topic discovery and monitoring in news," in *Proceedings of the 23rd international conference on World wide web*. International World Wide Web Conferences Steering Committee, 2014, pp. 527–538.

[21] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[22] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2007, pp. 1257–1264.

[23] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 263–272.