Chapter 2
Temporal Context-Aware Recommendation

Abstract Users' behaviors in social media systems are generally influenced by intrinsic interest as well as the temporal context (e.g., the public's attention at that time). In this chapter, we focus on analyzing user behaviors in social media systems and designing a latent class statistical mixture model, named *temporal context-aware mixture model* (TCAM), to account for the intentions and preferences behind user behaviors. TCAM simultaneously models the topics related to users' intrinsic interests and the topics related to temporal context, and then combines the influences from the two factors to model user behaviors in a unified way. To further improve the performance of TCAM, an item-weighting scheme is proposed to enable TCAM to favor items that better represent topics related to user interests and topics related to temporal context, respectively. Extensive experiments have been conducted to evaluate the performance of TCAM on four real-world datasets crawled from different social media sites. The experimental results demonstrate the superiority of the TCAM models.

Keywords Dynamic user behavior · Temporal recommendation · Temporal context modeling · Entropy filtering

2.1 Introduction

With the rising popularity of social media, a better understanding of users' rating behaviors[1] is of great importance for the design of many applications, such as personalized recommendation, information filtering, behavioral targeting, and computational advertising. Research efforts [12, 15] have been undertaken to model users' interests to help them find interesting items by analyzing their historical behaviors. However, existing work [12, 15, 17] simply assumes that users prefer items based on their intrinsic interests, which may not be accurate in many social application scenarios. For example, when choosing a book to read or a movie to watch, the users are likely to prefer books/movies that interest them. In contrast, when selecting news

[1] We use the term "rating behavior" to denote general user actions on items in social media systems, such as rating and viewing.

© Springer Science+Business Media Singapore 2016

H. Yin and B. Cui, *Spatio-Temporal Recommendation in Social Media*,
SpringerBriefs in Computer Science, DOI 10.1007/978-981-10-0748-4_2

to read or users to follow in a social network (e.g., Twitter), it is most likely that users will be attracted, respectively, by breaking news or famous users who are followed by the general public [4, 10, 20]. Therefore, users' rating behaviors on items may not necessarily indicate users' intrinsic interests. New models are desired to better account for user behaviors in social medias to learn user preferences more precisely.

After investigating multiple social media systems, we observe that user rating behaviors are generally influenced by two factors: *the intrinsic interest of the user* and *the attention of the general public*. While the user's intrinsic interest is relatively stable, the attention of the general public changes from time to time; for example, the hot topics on a microblogging site evolve over time. Hence, in our work, we refer to the attention of the public during a particular time period as *temporal context*.

The two factors have different degrees of influence on user rating behaviors for different types of social media platforms as a result of the different characteristics (e.g., life cycles and updating rates) of various types of social media items. For instance, news is a type of time-sensitive item with a short life cycle—few people want to read outdated news; while the life cycle of movies is relatively longer, with many classic old movies being highly ranked in the popularity list. For time-sensitive social media items, users are more easily influenced by the temporal context, whereas they tend to make decisions based on their intrinsic interests when choosing less time-sensitive items such as books and movies.

To model user rating behaviors in social media systems, therefore, it is critical to identify users' intrinsic interests as well as the temporal context (i.e., the attention of the general public during a particular time period). Moreover, it is essential to model the influence degrees of the two factors in different social media systems.

To this end, we proposed TCAM to mimic user rating behaviors in a process of decision making in [21, 22]. As shown in Fig. 2.1, TCAM is a latent class statistical mixture model that simultaneously models the topics [1, 8] related to users' intrinsic interests and the topics related to temporal context, and then combines the influences from the user interest and the temporal context to model user behaviors in a unified manner. Specifically, the model discovers (1) users' personal interest distribution over a set of latent topics; (2) the temporal context distribution over a set of latent topics; (3) an item generative distribution for each latent topic; and (4) the mixing weights that represent the influence probabilities of users' personal interest and the temporal context. It is worth mentioning that the set of latent topics used to model user interest is different from the topics used to model the temporal context. The former are called *user-oriented topics* and the latter are referred to as *time-oriented topics*.

The generative process of user rating behaviors in TCAM is briefly illustrated as follows. Suppose a user u selects an item v in a time interval t. TCAM first tosses a coin, based on the influence probabilities of the two factors, to decide whether this behavior results from the influence of the user's personal interest or the influence of the temporal context. If it results from the influence of the user's personal interest, TCAM chooses a *user-oriented topic* for u based on the user's intrinsic interest (with a certain probability). The selected topic in turn generates an item v following on from the topic's item generative distribution. Otherwise, if the influence from the

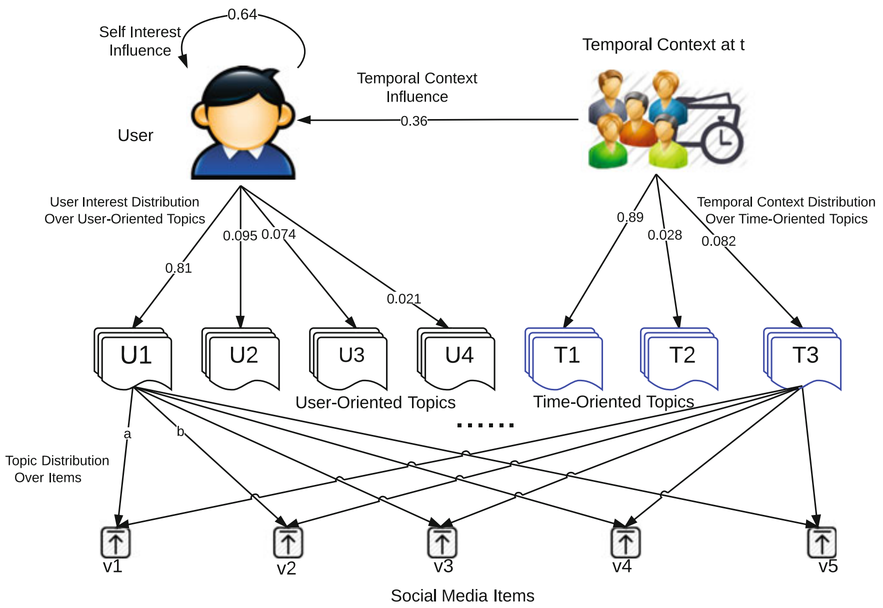

Fig. 2.1 An example of TCAM model [22] © 2015 Association for Computing Machinery, Inc. Reprinted by permission

temporal context is sampled, TCAM chooses a *time-oriented topic* according to the general public's interest during t, which in turn generates an item v.

Similar to traditional topic models where popular words in a document corpus are usually ranked high in each topic [3, 4], popular social media items tend to be estimated as having high generation probability by TCAM, which impairs the quality of the discovered user-oriented topics and time-oriented topics. User-oriented topics are supposed to capture user intrinsic interests, but a popular item favored by many users conveys less information about a user's intrinsic interest than an item favored by few users (i.e., a salient item) [23]. Similarly, a popular item constantly favored by users cannot well represent a time-oriented topic because the public's attentions change over time. Hence, to improve the performance of TCAM, we devise an item-weighting scheme to promote the importance of salient items and bursty items, which enhances the quality of the underlying topics detected by TCAM.

The remainder of this chapter is organized as follows. Section 2.2 details the TCAM. We deploy TCAM to temporal recommendation in Sect. 2.3. We carry out extensive experiments and report the experimental results in Sect. 2.4, and conclude the chapter in Sect. 2.5.

2.2 User Rating Behavior Modeling

In this section, we first introduce relevant definitions and notations used throughout this chapter. We then present the novel temporal context-aware mixture model for modeling user rating behaviors in social media systems.

2.2.1 Notations and Definitions

The notations used in this chapter are summarized in Table 2.1.

Definition 2.1 (*User Rating*) A user rating is a triple (u, t, v) that denotes a rating behavior (e.g., purchasing, clicking and tagging) made by user u on item v during time interval t.

Definition 2.2 (*User Document*) Given a user u, the user document, D_u, is a set of pairs $\{(v,\ t)\}$ representing the rating behaviors on items during different time intervals made by u.

Definition 2.3 (*Rating Cuboid*) A rating cuboid $\boldsymbol{C}$ is an $N \times T \times V$ cuboid, where N is the number of users, T is the number of time intervals and V is the number of items. A cell indexed by $(u,\ t,\ v)$ stores the rating score that user u assigned to item v during time interval t.

Table 2.1 Notations used in this model

Symbol	Description
u, t, v	User u, time interval t, item v
N, T, V	Number of users, time intervals, and items
M_u	Number of items rated by user u
λ_u	The mixing weight specific to user u
K_1	Number of user-oriented topics
$\theta_{u,z}$	Probability that user-oriented topic z is chosen by user u
θ_u	Intrinsic interest of user u denoted by $\theta_u = \{\theta_{u,z}\}_{z=1}^{K_1}$
ϕ_z	Item proportions of user-oriented topic z, denoted by $\phi_z = \{\phi_{z,v}\}_{v=1}^{V}$
$\phi_{z,v}$	Probability that item v is generated by user-oriented topic z
K_2	Number of time-oriented topics
θ'_t	The temporal context during time interval t denoted by $\theta'_t = \{\theta'_{t,x}\}_{x=1}^{K_2}$
$\theta'_{t,x}$	Probability that time-oriented topic x is generated by time interval t
ϕ'_x	Item proportions of time-oriented topic x denoted by $\phi'_x = \{\phi'_{x,v}\}_{v=1}^{V}$
$\phi'_{x,v}$	Probability that item v is generated by time-oriented topic x

User actions on items, such as tagging, downloading, purchasing, and clicking, can be represented as a user rating. Either explicit feedback or implicit feedback can be used to compute the value of rating score. For example, given a user u who frequently uses a tag v during time interval t, the usage frequency can be used as the rating score to reflect the user's preference on the tag during that time period.

Definition 2.4 (*Topic*) Given a collection of items $\boldsymbol{I} = \{v_i\}_{i=1}^{V}$, a topic z is represented by a topic model ϕ_z, which is a multinomial distribution over items $\phi_z = \{P(v_i|\phi_z)$ or $\phi_{z,v_i}\}_{i=1}^{V}$.

To illustrate the semantic meaning of a topic, we choose top-k items that have the highest probability under the topic, as shown in Fig. 2.2. In our work, we distinguish between **user-oriented topics** ϕ_z and **time-oriented topics** ϕ'_x although both of them are represented by a multinomial distribution over items. User-oriented topics are used to model user interest, which is assumed to be generally stable over time. In contrast, time-oriented topics are used to model the temporal context (i.e., the public's attention during a particular time), which has a clear temporal feature. For example, the popularity of the topics may increase or decrease over time and reach a peak during a certain period of time, as shown in Fig. 2.2.

Definition 2.5 (*User Interest*) Given a user u, her/his intrinsic interest, denoted as θ_u, is a multinomial distribution over user-oriented topics.

Definition 2.6 (*Temporal Context*) Given a time interval t, the temporal context during t, denoted as θ'_t, is a multinomial distribution over time-oriented topics or items.

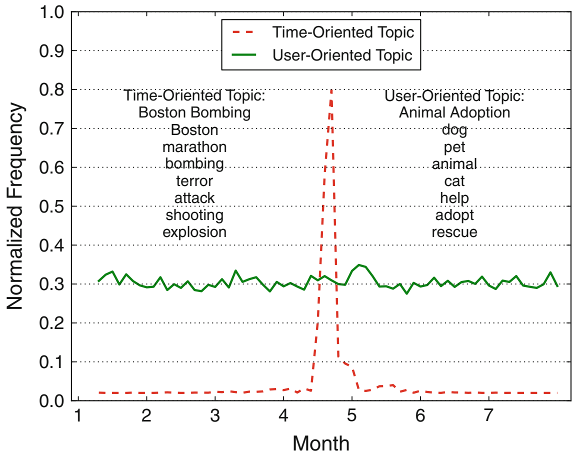

Fig. 2.2 An Example of two types of topics in delicious [22] © 2015 Association for Computing Machinery, Inc. reprinted by permission

2.2.2 Temporal Context-Aware Mixture Model

Given a rating cuboid $\boldsymbol{C}$ which stores users' rating histories, we aim to model user rating behaviors by exploiting the information captured in $\boldsymbol{C}$. Before presenting the devised model, we first describe an example to illustrate the motivation of our design.

As mentioned before, users' rating behaviors in social media systems are influenced by not only intrinsic interest but also the temporal context. It is crucial to distinguish between user-oriented topics and time-oriented topics, because the two have very different characteristics. For example, Fig. 2.2 shows an example of a user-oriented topic and a time-oriented topic detected by TCAM model from Delicious. For demonstration, we present only the top eight tags that have the highest probability under each topic. We can easily tell the difference between the two topics from both their temporal distributions and the content descriptions. For the time-oriented topic, the items (i.e., tags) are related to a certain event (e.g.,"Boston Marathon bombings"). The popularity of the topic experiences a sharp increase during a particular time interval (e.g., in April 2013). For the user-oriented topic, the items are about the user's regular interest (e.g., "Pet Adoption"). The temporal distribution of the topic does not show any spike-like fluctuation. Hence, our TCAM models the user-oriented topics and the time-oriented topics simultaneously.

To consider the influence of the user intrinsic interest and the temporal context in a unified manner, TCAM computes the likelihood that a user u will rate an item v during a time interval t as follows.

$$P(v|u, t, \Psi) = \lambda_u P(v|\theta_u) + (1 - \lambda_u)P(v|\theta_t') \tag{2.1}$$

where Ψ denotes the model parameter set, $P(v|\theta_u)$ is the probability that item v is generated from u's intrinsic interest, denoted as θ_u, and $P(v|\theta_t')$ denotes the probability that item v is generated from the temporal context during time interval t, i.e., θ_t'. The parameter λ_u is the mixing weight which represents the influence probability of the user interest. That is, user u is influenced by personal interest θ_u with probability λ_u, and is influenced by the temporal context θ_t' with probability $1 - \lambda_u$, for decision making. It is worth mentioning that TCAM holds personalized mixing weights for individual users, considering the differences between users in personalities (e.g., openness and agreeableness).

The user interest component θ_u is modeled by a multinomial distribution over user-oriented topics, and each item is generated from a user-oriented topic z. Thus, $P(v|\theta_u)$ is computed as follows.

$$P(v|\theta_u) = \sum_{z=1}^{K_1} P(v|\phi_z)P(z|\theta_u) \tag{2.2}$$

As for the temporal context component θ_t', it is modeled as a multinomial distribution over a set of latent time-oriented topics, and each item is generated from a time-oriented topic x. Then, $P(v|\theta_t')$ is formulated as follows:

$$P(v|\theta'_t) = \sum_{x=1}^{K_2} P(v|\phi'_x)P(x|\theta'_t). \tag{2.3}$$

As an illustrative example in Fig. 2.1, the user is influenced by personal interest and the temporal context with probabilities 0.64 and 0.36, respectively. Four user-oriented topics and three time-oriented topics are also shown, respectively, where the weights representing the user's interest distribution over the user-oriented topics as well as the temporal context distribution over the time-oriented topics are labeled in the corresponding edges. We can see that user-oriented topic U1 dominates the user's interest, and time-oriented topic T1 attracts most attentions from the general public at time t. The probabilities of topics' generating items are also labeled in the corresponding edges. For example, the weight b on the edge linking topic U1 and item v2 represents the probability of U1 generating item v2.

Figure 2.3 illustrates the generative process of TCAM with a graphical model. The structure of TCAM is similar to the PLSA model, but TCAM has additional machinery to handle the mixing weight λ_u. In particular, a latent random variable s, associated with each item, is adopted as a switch to determine whether the item is generated according to the temporal context θ'_t or the user's interest θ_u. s is sampled from a user-specific Bernoulli distribution with the mean λ_u. N indicates the number of users; K_1 is the number of user-oriented topics; K_2 is the number of time-oriented topics; T is the number of time slices and M_u is the number of items rated by u. The generative process of TCAM is summarized as follows.

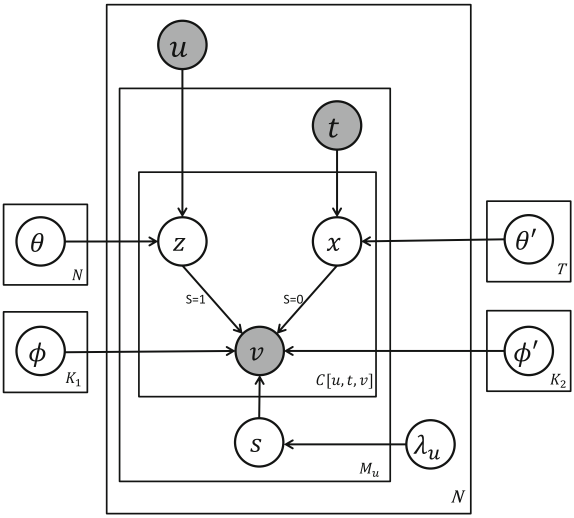

Fig. 2.3 The graphical representation of TCAM [22] ©2015 Association for Computing Machinery, Inc. reprinted by permission

For each item v rated by u at time slice t:

1. Sample s from $Bernoulli(\lambda_u)$
2. If $s = 1$
   a. Sample user-oriented topic z from $Multinomial(\theta_u)$
   b. Sample item v from $Multinomial(\phi_z)$
   c. Repeat the above two steps $C[u, t, v]$ times
3. Otherwise
   a. Sample time-oriented topic x from $Multinomial(\theta'_t)$
   b. Sample item v from $Multinomial(\phi'_x)$
   c. Repeat the above two steps $C[u, t, v]$ times

2.2.3 Model Inference

Given a rating cuboid $\boldsymbol{C}$, the learning procedure of our model is to estimate the unknown model parameter set $\Psi = \{\theta, \phi, \theta', \phi', \lambda\}$. The log-likelihood is derived as follows:

$$L(\Psi|\boldsymbol{C}) = \sum_{u=1}^{N}\sum_{t=1}^{T}\sum_{v=1}^{V} C[u, t, v] \log P(v|u, t, \Psi), \tag{2.4}$$

where $P(v|u, t, \Psi)$ is defined in Eq. (2.1).

The goal of parameter estimation is to maximize the log-likelihood in Eq. (2.4). As this equation cannot be solved directly by applying maximum likelihood estimation (MLE), we apply an EM approach instead. In the expectation (E) step of the EM approach, we introduce $P(s|u, t, v; \widehat{\Psi})$ which is the posterior probability of choosing personal interest θ_u (i.e., $s = 1$) or temporal context θ'_t (i.e., $s = 0$), respectively, given user rating behavior (u, t, v) and the current estimations of the parameters $\widehat{\Psi}$. In the maximization (M) step, parameters are updated by maximizing the expected complete data log-likelihood $Q(\Psi)$ based on the posterior probability computed in the E-step.

In the **E-step**, $P(s|u, t, v; \widehat{\Psi})$ is updated according to Bayes formulas as in Eq. (2.5).

$$P(s|u, t, v; \widehat{\Psi}) = \frac{s\lambda_u P(v|\theta_u) + (1-s)(1-\lambda_u)P(v|\theta'_t)}{\lambda_u P(v|\theta_u) + (1-\lambda_u)P(v|\theta'_t)}, \tag{2.5}$$

where $P(v|\theta_u)$ and $P(v|\theta'_t)$ are defined as in Eqs. (2.2) and (2.3), respectively. To obtain the updated parameters $P(z|\theta_u)$ and $P(v|\phi_z)$, the posterior probability $P(z|s = 1, u, t, v; \widehat{\Psi})$ is computed as:

$$P(z|s = 1, u, t, v; \widehat{\Psi}) = \frac{P(v|\phi_z)P(z|\theta_u)}{\sum_{z'=1}^{K_1} P(v|\phi_{z'})P(z'|\theta_u)}. \tag{2.6}$$

Based on $P(z|s=1,u,t,v;\widehat{\Psi})$ and $P(s=1|u,t,v;\widehat{\Psi})$, we introduce the notation $P(z|u,t,v;\widehat{\Psi})$ as follows:

$$P(z|u,t,v;\widehat{\Psi}) = P(z|s=1,u,t,v;\widehat{\Psi})P(s=1|u,t,v;\widehat{\Psi}). \tag{2.7}$$

To obtain the updated parameters $P(x|\theta'_t)$ and $P(v|\phi'_x)$, we update the posterior probability $P(x|s=0,u,t,v;\widehat{\Psi})$ as follows:

$$P(x|s=0,u,t,v;\widehat{\Psi}) = \frac{P(v|\phi'_x)P(x|\theta'_t)}{\sum_{x'=1}^{K_2} P(v|\phi'_{x'})P(x'|\theta'_t)}. \tag{2.8}$$

Based on $P(x|s=0,u,t,v;\widehat{\Psi})$ and $P(s=0|u,t,v;\widehat{\Psi})$, we introduce the notation $P(x|u,t,v;\widehat{\Psi})$ as follows:

$$P(x|u,t,v;\widehat{\Psi}) = P(x|s=0,u,t,v;\widehat{\Psi})P(s=0|u,t,v;\widehat{\Psi}). \tag{2.9}$$

With simple derivations [8], we obtain the expectation of complete data log-likelihood for TCAM:

$$\begin{aligned} Q(\Psi) = &\sum_{u=1}^{N}\sum_{v=1}^{V}\sum_{t=1}^{T} C[u,t,v]\{P(s=1|u,t,v;\widehat{\Psi})\sum_{z=1}^{K_1} P(z|s=1,u,t,v;\widehat{\Psi})\log[\lambda_u P(v|\phi_z)P(z|\theta_u)] \\ &+ P(s=0|u,t,v;\widehat{\Psi})\sum_{x=1}^{K_2} P(x|s=0,u,t,v;\widehat{\Psi})\log[(1-\lambda_u)P(v|\phi'_x)P(x|\theta'_t)]\}. \end{aligned} \tag{2.10}$$

In the **M-step**, we find the estimation Ψ that maximizes the expectation of the complete data log-likelihood $Q(\Psi)$ with the constraints $\sum_{v=1}^{V} P(v|\phi_z) = 1$, $\sum_{v=1}^{V} P(v|\phi'_x) = 1$, $\sum_{z=1}^{K_1} P(z|\theta_u) = 1$ and $\sum_{x=1}^{K_2} P(x|\theta'_t) = 1$, using the following updating formulas.

$$P(z|\theta_u) = \frac{\sum_{v=1}^{V}\sum_{t=1}^{T} C[u,t,v]P(z|u,t,v;\widehat{\Psi})}{\sum_{z'=1}^{K_1}\sum_{v=1}^{V}\sum_{t=1}^{T} C[u,t,v]P(z'|u,t,v;\widehat{\Psi})} \tag{2.11}$$

$$P(v|\phi_z) = \frac{\sum_{t=1}^{T}\sum_{u=1}^{N} C[u,t,v]P(z|u,t,v;\widehat{\Psi})}{\sum_{v'=1}^{V}\sum_{t=1}^{T}\sum_{u=1}^{N} C[u,t,v']P(z|u,t,v';\widehat{\Psi})} \tag{2.12}$$

$$P(x|\theta'_t) = \frac{\sum_{v=1}^{V}\sum_{u=1}^{N} C[u,t,v]P(x|u,t,v;\widehat{\Psi})}{\sum_{x'=1}^{K_2}\sum_{v=1}^{V}\sum_{u=1}^{N} C[u,t,v]P(x'|u,t,v;\widehat{\Psi})} \tag{2.13}$$

$$P(v|\phi'_x) = \frac{\sum_{t=1}^{T}\sum_{u=1}^{N} C[u,t,v]P(x|u,t,v;\widehat{\Psi})}{\sum_{v'=1}^{V}\sum_{t=1}^{T}\sum_{u=1}^{N} C[u,t,v']P(x|u,t,v';\widehat{\Psi})} \tag{2.14}$$

With an initial random guess of Ψ , we alternately apply the E-step and M-step until a termination condition is met. To adapt to different users, we estimate the parameter λ_u in M-step, instead of picking a fixed λ value for all users. This personalized treatment can automatically adapt the model parameter estimation for various users. Specifically, λ_u is estimated as follows.

$$\lambda_u = \frac{\sum_{t=1}^{T}\sum_{v=1}^{V} C[u,t,v]P(s=1|u,t,v;\widehat{\Psi})}{\sum_{t=1}^{T}\sum_{v=1}^{V}\sum_{s=0}^{1} C[u,t,v]P(s|u,t,v;\widehat{\Psi})} \tag{2.15}$$

2.2.4 *Discussion About TCAM*

A number of relevant issues of the proposed TCAM model are discussed in this subsection.

Hyperparameter setting. In our model, we still have four hyperparameters to tune manually, including the number of user-oriented topics K_1, the number of time-oriented topics K_2, the number of time slices T and the number of EM iterations. K_1 and K_2 are the desired numbers of user-oriented topics and time-oriented topics, respectively, which need to be tuned empirically. T is the number of time slices used in our model to generate time-oriented topics, which provides users with the flexibility to adjust the granularity/length of the time slice. The larger T is, the more fine-grained time slices are. Regarding the number of EM iterations, we observe that convergence can be achieved in a few iterations (e.g., 50) because the model inference procedure using the EM approach is fast. The time cost of each iteration is $\mathcal{O}(NK_1V + TK_2V)$, which is very similar to the time cost required for PLSA implementation [8]. It is worth mentioning that EM algorithms can be easily expressed in MapReduce [6, 18], so the inference procedure of TCAM can be naturally decomposed for parallel processing, which is scalable to large-scale datasets.

Guidance with Dirichlet Priors. Prior knowledge can be integrated into TCAM models to guide the topic discovery process. For example, in the MovieLens dataset, we can introduce prior knowledge and guide the user-oriented topics so that they are related to the genres of movies, such as action and comedy. Another example is the Digg dataset, where we can integrate prior knowledge to guide the time-oriented topics so that they are aligned with breaking events. Specifically, we define a conjugate prior (i.e., Dirichlet prior) on each multinomial topic distribution. Let us denote the Dirichlet prior β_z for user-oriented topic z and β'_x for time-oriented topic x. $\beta_z(v)$ and $\beta'_x(v)$ can be interpreted as the corresponding pseudo counts for item v when we estimate the topic distributions $P(v|\phi_z)$ and $P(v|\phi'_x)$, respectively. With these conjugate priors, we can use the maximum a posteriori (MAP) estimator for parameter estimation, which can be computed using the same EM algorithm except that we should replace the Eqs. (2.12) and (2.14) with the following formulas, respectively:

$$P(v|\phi_z) = \frac{\sum_t \sum_u C[u,t,v]P(z|u,t,v;\widehat{\Psi}) + \beta_z(v)}{\sum_{v'} \sum_t \sum_u (C[u,t,v']P(z|u,t,v';\widehat{\Psi}) + \beta_z(v'))},$$

$$P(v|\phi'_x) = \frac{\sum_t \sum_u C[u,t,v]P(x|u,t,v;\widehat{\Psi}) + \beta_x(v)}{\sum_{v'} \sum_t \sum_u (C[u,t,v']P(x|u,t,v';\widehat{\Psi}) + \beta_x(v'))}.$$

Advantages of TCAM. The advantages of the TCAM model are summarized as follows. (1) We unify the influences from the user interest and the temporal context to model user rating behaviors. (2) We distinguish between user-oriented topics and time-oriented topics. Two different types of latent topics are proposed to model user interest and the temporal context, respectively. By taking away the influence of the temporal context, *user-oriented topics* can capture user intrinsic interests more precisely. Likewise, without the influence of user interests, *time-oriented topics* can better reflect the temporal context because the noise induced by a wide variety of user interests could contaminate the time-oriented topics. (3) TCAM can generate interpretable individual user profiles that can be presented alongside item recommendations to allow users to understand the rationale behind the recommendations. We will show sample user profiles in Sect. 2.4.6.

2.2.5 *Item-Weighting for TCAM*

In this section, we propose an item-weighting scheme to improve TCAM's performance. Similarly to traditional topic models, TCAM assumes that all items are equally important in computing generation probabilities. As a result, popular items with more ratings tend to be estimated with high generation probability and ranked in top positions in each topic, which impairs the quality of both the user-oriented topics and the time-oriented topics.

For user-oriented topics, popular items are not good indicators of user intrinsic interests. A popular item rated by many users conveys less information about a user's interest than an item rated by few users. For time-oriented topics, it is expected that items representing the public's attention at a given time should be highly ranked, such as items with bursty temporal distributions, since bursts of items are generally triggered by breaking news or events that attract the public's attention. Unfortunately, bursty items are most likely to be overwhelmed by long-standing popular items. Figure 2.4 shows the temporal frequency of the top six tags of a sample time-oriented topic discovered from Delicious. It can be observed that the topic concerns swine flu. The temporal distributions of three bursty tags, "flu," "mexico," and "swineflu", undergo sharp spikes. Although the trends of the three tags do not always synchronize, they each go through a drastic increase and reach a peak in July 2009. The bursts in these curves are triggered by a real-world event, i.e., the swine flu outbreak in Mexico. The other three tags, "news," "health," and "death", maintain high frequency throughout the year. However, they convey little information about the event.

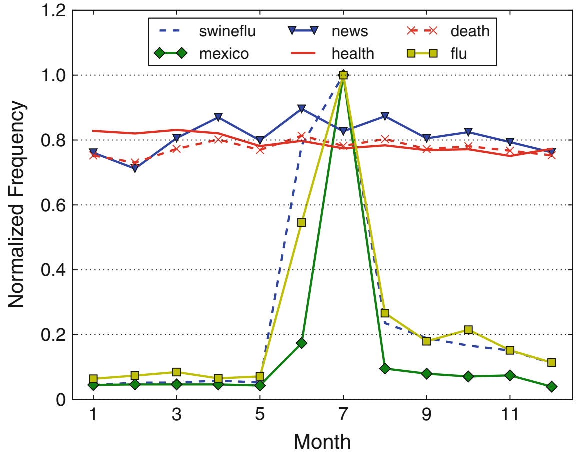

Fig. 2.4 An example of bursty tags and popular tags [22] ©2015 Association for Computing Machinery, Inc. reprinted by permission

Although they are relevant to the event, they are also related to many other topics. Hence, it is desirable to rank bursty items higher than popular item when representing time-oriented topics.

To address the challenge posed by the popular items, we propose an item-weighting scheme to reduce the importance of popular items while promoting weights for salient, but infrequent, and bursty items in computing generation probability. From the viewpoint of information theory [5], the entropy of an item v is defined as follows:

$$E(v) = -\sum_{u} P(u|v) \log P(u|v).$$

Suppose that the item v is preferred by users with equal probability $P(u|v) = \frac{1}{N(v)}$, the maximum entropy is,

$$E(v) = \log N(v).$$

Generally, the entropy of an item tends to be proportional to its frequency/ popularity $N(v)$. Hence, in the following analysis, we use the maximum entropy to approximate the exact entropy to simplify the calculation.

To allow salient items to be ranked higher in use-oriented topics, the weights of items should be inversely proportional to the entropy, as discussed above. Hence, we propose a concept called *inverse user frequency* to measure the ability of items to represent salient information. Let N be the total number of users in the entire dataset; the *inverse user frequency (IUF)* for the item v is defined as follows:

$$iuf(v) = \log N - \log N(v) = \log \frac{N}{N(v)}, \tag{2.16}$$

which is similar to the inverse document frequency for a term in text mining.

To take into account the bursty information of items, we propose to compute the *bursty degree* of an item v using the following equation:

$$B(v, t) = \frac{N_t(v)}{N_t} \frac{N}{N(v)}, \tag{2.17}$$

where $N_t(v)$ represents the popularity of item v at time slice t, i.e., the number of users who rate item v at time slice t, N_t is the number of active users at time slice t, $N(v)$ is the overall popularity of v across all time slices, and N is the total number of users in the dataset.

Combining the inverse user frequency and the bursty degree of items, we assign weight to the item v as follows.

$$w(v, t) = iuf(v) \times B(v, t) \tag{2.18}$$

Integrating the weights of items defined in Eq. (2.18), we obtain the weighted user-time item cuboid $\bar{\boldsymbol{C}}$ from the original $\boldsymbol{C}$ as follows:

$$\bar{C}[u, t, v] = C[u, t, v]w(v, t). \tag{2.19}$$

It should be noted that the item-weighting scheme makes TCAM no longer correspond to a well-defined probabilistic generative process since the values stored in the cuboid $\boldsymbol{C}$ are no longer integers, but it actually improves the empirical performance of TCAM in the tasks of both temporal recommendation and topic discovery by inserting IUF and bursty weights as heuristic factors into the model inference procedures, as shown in the experiment section. From the perspective of information theory, an item with lower entropy conveys more information about a user's intrinsic interests, and the one with higher bursty degree is more capable of representing the temporal context at some specific time. The TCAM model enhanced by IUF and bursty weights incorporate these observations and intuitions.

2.3 Temporal Recommendation

The conventional top-k recommendation task can be stated as follows: given a user, the recommender system should recommend a small number, say k, of items from all the available items. Note that the conventional top-k recommendation task does not consider the temporal information. However, in reality, user rating behaviors, influenced by both user interests and the temporal context, are dynamic. For example, user u rating item v in time interval t does not mean that u still favors v in time interval $t + 1$. Besides, each item has its own lifespan, especially for time-sensitive items such as news. It is undesirable to recommend outdated news. Hence, an ideal recommender system is expected to have the ability to recommend the right item v to user u in the right time interval t, rather than in other time intervals. In this chapter, we propose the task of temporal top-k recommendation as follows: given a query $q = (u, \ t)$, i.e., a querying user u with a time interval t, the recommender model recommends k items which match u's interests and the temporal context at t.

Below, we will present how to deploy TCAM to facilitate temporal recommendations. Once we have inferred model parameters of TCAM, such as user interest θ, temporal context θ', user-oriented topics ϕ, time-oriented topics ϕ' and mixing weights λ, given a query $q = (u, \ t)$, a ranking score $S(u, t, v)$ for each item v can be computed according to Eq. (2.21), and then the top-k items with highest ranking score will be returned. Specifically, when receiving a query $q = (u, \ t)$, a new multinomial distribution for the query, ϑ_q, is first constructed by combining θ_u and θ'_t. More specifically, we expand the user interest and temporal context spaces to be of the same dimension. For example, if there are K_1 user-oriented topics and K_2 time-oriented topics, the expanded topic space will have $K = K_1 + K_2$ topics. The expanded user interest distribution is defined as $\widetilde{\theta_u} =< \theta_u, 0, \cdots, 0 >$, where we set 0 on the time-oriented topics. Similarly, we define the expanded temporal context distribution to be $\widetilde{\theta'_t} =< 0, \ldots, 0, \theta'_t >$. The new distribution is defined as $\vartheta_q = \lambda_u \widetilde{\theta_u} + (1 - \lambda_u)\widetilde{\theta'_t}$. Correspondingly, we renumber the time-oriented topic x and change its range from $[1, \ldots, K_2]$ to $[K_1 + 1, \ldots, K]$. Then, we use $\varphi_{\widetilde{z},v}$ to denote the weight of item v on dimension $\widetilde{z}$ that corresponds to user-oriented topic z or time-oriented topic x, which depends on the value of $\widetilde{z}$, as shown in Eq. (2.20).

$$\varphi_{\widetilde{z},v} = \begin{cases} \phi_{z,v} & \widetilde{z} \leq K1 \\ \phi'_{x,v} & \widetilde{z} > K1 \end{cases} \tag{2.20}$$

$$S(u, t, v) = \sum_{\widetilde{z}=1}^{K} \vartheta_{q,\widetilde{z}} \varphi_{\widetilde{z},v} \tag{2.21}$$

2.4 Experiments

In this section, we experimentally evaluate the performance of our proposed models.

2.4.1 Datasets

Our experiments are conducted on four real datasets: Digg, MovieLens, Douban Movie, and Delicious. The basic statistics of the four datasets are shown in Table 2.2. Only the implicit feedback data can be available in Digg and Delicious datasets, so we compute the cell value $C[u, t, v]$ for these two datasets according to the frequency/number of the interaction between user u and item v at time t. For Douban

Table 2.2 Basic statistics of the four datasets

	Digg	MovieLens	Douban movie	Delicious
# of users	139,409	71,567	33,561	201,663
# of items	3,553	10,681	87,081	2,828,304
# of ratings	3,018,197	10,000,054	5,257,665	36,966,661
time span(year)	2009–2010	1998–2009	2005–2010	2008–2009

Movie and MovieLens datasets, the explicit feedback information is available. Hence, the user-time item rating cuboid $\boldsymbol{C}$ can be directly derived from users' star ratings.

- **Digg**. Digg is a popular social news aggregator, which allows users to vote news stories up or down, called *digging* or *burying*, respectively. The Digg dataset used in our experiment is Digg2009 [9], a publicly available dataset containing 3,018,197 votes on 3553 popular stories cast by 139,409 distinct users. This dataset also records the friendship network of these Digg users. Although this dataset contains only the IDs of news stories (the titles and the contents of stories are excluded), it is sufficient to evaluate the effectiveness of user behavior modeling in our work.
- **Douban Movie**. Douban[2] is the largest movie review website in China. In total, we crawled 33,561 unique users and 87,081 unique movies with 5,257,665 movie ratings.
- **MovieLens**. MovieLens is a publicly available movie dataset from the web-based recommender system MovieLens. The dataset contains $10M$ ratings on a scale from 1 to 5 made by 71567 users on 10681 movies. We selected users who had rated at least 20 movies.
- **Delicious**. Delicious is a collaborative tagging system where users can upload and tag web pages. We collected 201,663 users and their tagging behaviors during the period Feb. 2008–Dec. 2009. The dataset contains 2,828,304 tags. Topics on technology and electronics account for about half of all web pages. Most of the other web pages are about breaking news with strong temporal features.

Note that the Douban Movie and Delicious datasets are collected by ourselves, and we make them publicly available.[3]

2.4.2 *Comparisons*

The temporal context-aware mixture model (TCAM) was outlined in Sect. 2.2. TCAM can be enhanced by the item-weighting scheme, which leads to a weighted TCAM, called WTCAM. We compare them with four categories of competitor approaches.

[2] http://douban.com.

[3] http://net.pku.edu.cn/daim/hongzhi.yin/.

- **User-Topic Model (UT)**. We implemented a user-topic model following the previous works [12, 15]. This model is similar to the classic author-topic model (AT model) [14] which assumes that topics are generated according to user interests. A user document D_u is regarded as a sample of the following mixture model:

$$P(v|u;\Psi) = \lambda_B P(v|\theta_B) + (1-\lambda_B)\sum_z P(z|\theta_u)P(v|\phi_z),$$

where v is an item (or word) in user document D_u, $P(z|\theta_u)$ is the probability of user u choosing the zth topic ϕ_z. θ_B is a background model and λ_B is the mixing weight for it. The purpose of using a background model θ_B is to make the topics learned from the dataset more discriminative; since θ_B gives high probabilities to nondiscriminative and noninformative items or words, we expect such items or words to be accounted for by θ_B and thus the topic models to be more discriminative. In a nutshell, the background model θ_B is used to capture common items or words.
- **Time-Topic Model (TT)**. Following previous works [11, 16], we implemented a time-topic model. This model considers only the temporal information and ignores user interests. TT assumes that topics are generated by the temporal context, and that user behaviors are influenced by the temporal context. The probabilistic formula of the time-topic model is presented as follows:

$$P(v|t;\Psi) = \lambda_B P(v|\theta_B) + (1-\lambda_B)\sum_x P(x|\theta'_t)P(v|\phi'_x),$$

where $P(x|\theta'_t)$ is the probability of the general public choosing xth topic during time period t, and θ_B is a background model that plays the same role with the one in the above UT model.
- **BPRMF**. This is a state-of-the-art matrix factorization model for item ranking that is optimized using BPR [13]. This model outperforms most of the existing recommender models in the task of top-k item recommendation. We used the BPRMF implementation provided by MyMediaLite, a free software recommender system library [7].
- **BPTF**. This is a state-of-the-art recommender model for rating prediction that uses a probabilistic tensor factorization technique by introducing additional factors for time [19]. This model outperforms most of the existing recommender models that consider time information.

2.4.3 *Evaluation Methodology*

To make the evaluation process fair and reproducible, we adopt the *methodological description framework* proposed in [2] to describe our evaluation conditions. We will present our evaluation conditions by answering the following methodological questions:

1. What *base set* is used to perform the training-test building?
2. What *rating order* is used to assign ratings to the training and test sets?
3. How many *ratings* comprise the training and test sets?
4. Which items are considered as *target items*?
5. Which items are considered as *relevant items* for each user?

Base set condition. The base set conditions state whether the splitting procedure of training and test sets is based on the whole set of ratings $\boldsymbol{C}$, or on each of the sub-datasets of $\boldsymbol{C}$ independently. We adopt the *user-centered base set condition* where we perform the splitting independently on each user's ratings, ensuring that all users will have ratings in both the training and test sets.

Rating order and size conditions. We adopt the *time-dependent rating order condition*. Specifically, for each user u, his/her ratings $S(u)$ are ranked according to their rated timestamps. We use the 80-th percentile as the cut-off point so that ratings before this point will be used for training and the rest are for testing, i.e., $S(u)$ is divided into the training set $S^{train}(u)$ and test set $S^{test}(u)$.

Target item condition. To simulate a real-world setting, we require each tested recommender system to rank all the items except the target user's training items. In other words, given a target user u, each tested recommender system has to find top-k items from all available items except those in the set $S^{train}(u)$.

Relevant item condition. Relevant item conditions select the items to be interpreted as relevant for the target user. The notion of relevance is central for information retrieval metrics applied to evaluate top-k recommendations. We adopt the *test-based relevant items* condition in which the set of relevant items for target user u is formed by the items in u's test set $S^{test}(u)$.

The above condition combination is also called "uc_td_prop" for short. We repeat our experiments on four different social media datasets for increasing the generalization of the evaluation results, and we use a hold-out procedure on each dataset.

To make the experimental results comparable and reproducible, we use multiple well-known metrics to measure the ranked results. Similar to evaluations in information retrieval, we first use Precision@k to assess the quality of the top-k recommended items as follows:

$$Precision@k = \frac{\#relavances}{k},$$

where $\#relavances$ is the number of relevant items in the top-k recommended items. We also consider NDCG, a widely used metric for a ranked list. NDCG@k is defined as:

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

where r_i is 1 if the item at position i is a "relevant" item and 0 otherwise. IDCG is chosen for the purpose of normalization so that the perfect ranking has an NDCG value of 1. Considering that some users may have a large number of items in the test data while some have just a few, we also adopt the $F1$ score as our metric.

2.4.4 Recommendation Effectiveness

Tables 2.3, 2.4, 2.5 and 2.6 report the performance of the proposed models and other competitors in terms of Precision@k, NDCG@k and F1@k on Digg, MovieLens, Douban Movie and Delicious datasets, respectively. From the reported results, we observe that:

1. Our proposed models TCAM and W-TCAM consistently outperform other competitors such as UT, TT, BPRMF and BPTF on all four datasets. This observation shows that recommendation accuracy, especially temporal recommendation accu-

Table 2.3 Temporal recommendation accuracy on digg dataset

Methods	Precision			NDCG			F1 Score		
	P@1	P@5	P@10	N@1	N@5	N@10	F1@1	F1@5	F1@10
UT	0.091	0.086	0.084	0.093	0.091	0.088	0.007	0.028	0.044
TT	0.182	0.149	0.126	0.178	0.148	0.139	0.017	0.041	0.071
BPRMF	0.048	0.045	0.040	0.048	0.040	0.037	0.002	0.015	0.022
BPTF	0.194	0.166	0.152	0.195	0.176	0.165	0.017	0.050	0.080
TCAM	0.237	0.210	0.179	0.234	0.203	0.188	0.017	0.056	0.093
W-TCAM	**0.258**	**0.220**	**0.201**	**0.252**	**0.224**	**0.208**	**0.019**	**0.063**	**0.098**

Table 2.4 Temporal Recommendation Accuracy on MovieLens Dataset

Methods	Precision			NDCG			F1 Score		
	P@1	P@5	P@10	N@1	N@5	N@10	F1@1	F1@5	F1@10
UT	0.343	0.252	0.211	0.338	0.257	0.234	0.036	0.099	0.136
TT	0.260	0.193	0.168	0.248	0.198	0.186	0.024	0.070	0.093
BPRMF	0.342	0.239	0.192	0.303	0.229	0.195	0.034	0.084	0.119
BPTF	0.383	0.270	0.224	0.365	0.290	0.261	0.035	0.103	0.141
TCAM	0.385	0.304	0.259	0.406	0.325	0.286	0.037	0.115	0.155
W-TCAM	**0.401**	**0.324**	**0.264**	**0.427**	**0.350**	**0.313**	**0.040**	**0.123**	**0.165**

Table 2.5 Temporal Recommendation Accuracy on Douban Movie Dataset

Methods	Precision			NDCG			F1 Score		
	P@1	P@5	P@10	N@1	N@5	N@10	F1@1	F1@5	F1@10
UT	0.141	0.104	0.087	0.139	0.106	0.097	0.015	0.041	0.056
TT	0.105	0.078	0.068	0.101	0.080	0.075	0.010	0.028	0.038
BPRMF	0.138	0.101	0.085	0.136	0.103	0.094	0.015	0.040	0.055
BPTF	0.158	0.111	0.092	0.151	0.119	0.108	0.015	0.043	0.058
TCAM	0.168	0.133	0.113	0.177	0.142	0.125	0.016	0.050	0.068
W-TCAM	**0.175**	**0.141**	**0.115**	**0.186**	**0.153**	**0.137**	**0.018**	**0.054**	**0.072**

Table 2.6 Temporal Recommendation Accuracy on Delicious Dataset

Methods	Precision			NDCG			F1 Score		
	P@1	P@5	P@10	N@1	N@5	N@10	F1@1	F1@5	F1@10
UT	0.086	0.076	0.073	0.082	0.081	0.078	0.006	0.025	0.039
TT	0.104	0.085	0.072	0.102	0.085	0.080	0.010	0.024	0.041
BPRMF	0.065	0.059	0.051	0.064	0.062	0.060	0.005	0.019	0.030
BPTF	0.111	0.095	0.087	0.112	0.101	0.095	0.010	0.029	0.046
TCAM	0.136	0.120	0.103	0.134	0.116	0.108	0.010	0.032	0.053
W-TCAM	**0.148**	**0.126**	**0.113**	**0.144**	**0.128**	**0.119**	**0.011**	**0.036**	**0.056**

racy, can be improved by simultaneously considering both user intrinsic interests and the temporal context.

2. BPTF performs better than other competitor methods such as BPRMF, UT and TT because it also exploits the temporal context information when recommending items, but our proposed TCAM and W-TCAM consistently outperform BPTF. This may be because BPTF is designed for rating prediction rather than the top-k recommendation. It relies on high quality explicit feedback data (e.g., users' explicit star rating for items), however, which is not always available [13]. In contrast, our proposed TCAM and W-TCAM are suitable for both explicit and implicit user feedback data.
3. W-TCAM achieves higher temporal recommendation accuracy than TCAM, which demonstrates the benefits gained by the item-weighting scheme.
4. Comparing UT and TT, we find that UT performs better than TT on the MovieLens and Douban Movie datasets, while TT beats UT on the Digg dataset. This may be because news is a type of time-sensitive item while movies are not so time-sensitive and have longer life-span.

2.4.5 Temporal Context Influence Study

This section studies the influence degrees of users' personal interests and the temporal context on users' decision making. The user interest influence probability λ_u and the temporal context influence probability $1 - \lambda_u$ are learnt automatically in the TCAM model. We are interested in how significantly the temporal context influences the user's decisions on different social media platforms.

Since different people have different mixing weights, we plot the distributions of both the personal interest and the temporal context influence probabilities across all users. The results on the MovieLens data set are shown in Fig. 2.5, where Fig. 2.5a plots the cumulative distribution of personal interest influence probabilities, and Fig. 2.5b shows the temporal context influence probabilities. It is observed that, in general, people's personal interest influence is significantly higher than the influence

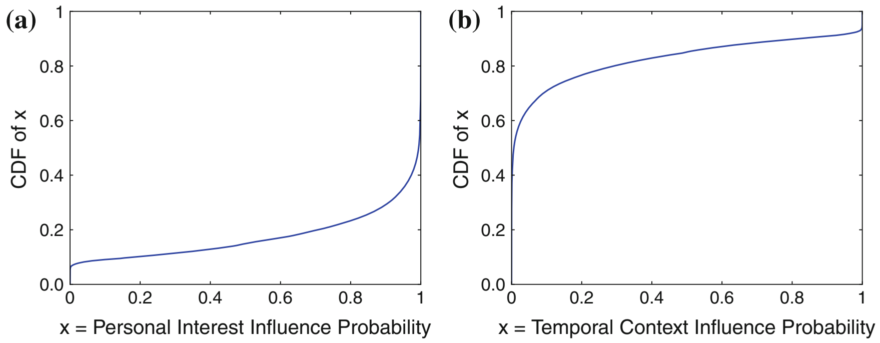

Fig. 2.5 Temporal context influence result (MovieLens). **a** Personal interest influence. **b** Temporal context influence

of the temporal context. For example, Fig. 2.5a shows that the personal interest influence probability of more than 76 % of users is higher than the 0.82. This observation indicates that most movies consumed by users are selected in accordance with their interests and tastes.

Figure 2.6a, b show, respectively, the personal interest influence probabilities and temporal context influence probabilities learnt from the Digg data. As shown in Fig. 2.6a, the personal interest influence probability is smaller than the temporal context influence probability. For example, the temporal context influence probability of more than 70 % of users is higher than 0.5. The implication of this finding is that people are mainly influenced by the temporal context when choosing news to read. By comparing the analysis results obtained from the two datasets, we observe that the temporal context influence on users' choice of news to read is much more significant than it is on the selection of movies to watch. This is probably because news is a time-sensitive item that is driven by offline social events, while movies are not so time-sensitive.

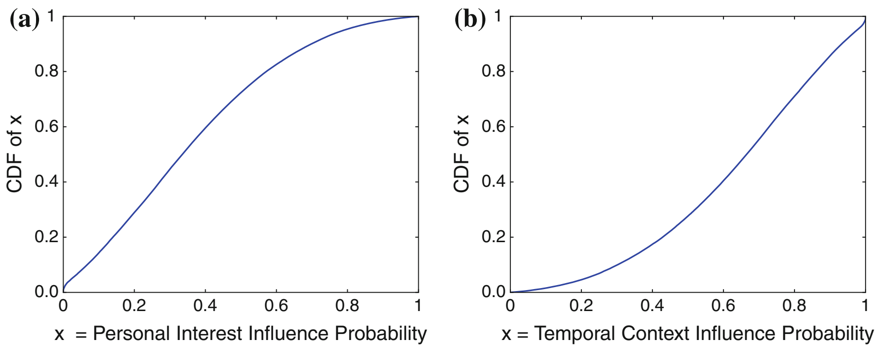

Fig. 2.6 Temporal context influence result (Digg). **a** Personal interest influence. **b** Temporal context influence

2.4.6 User Profile Analysis

Both the user interests and temporal context, as well as their influences on users' decision making, can be learnt by our TCAM model to build user profiles. This section first analyzes two sample user profiles to enable a better interpretation of user rating behaviors. Figure 2.7 shows the profiles of user 102 and user 384 learnt by TCAM from the Digg dataset. As shown in the figure, users 102 and 384 are influenced by the temporal context with influence probability values 0.88 and 0.76, respectively. We also show top-4 user-oriented topics with highest probabilities in θ_u. The weights on the edges indicate users' preferences for the topics. Note that we only choose top-4 user-oriented topics for demonstration, thus the sum of the weights on the edges is not equal to 1. There is only one overlapping user-oriented topic for users 102 and 384, and the dominating user-oriented topics for them are different (i.e., $U1$ vs. $U8$).

We also show two sample temporal context profiles for time slices 6 and 7 in Fig. 2.8. We present the top-4 time-oriented topics with highest probabilities in each θ'_t. The weights on the edges indicate the preference degrees of the general public for the chosen four time-oriented topics. We choose two temporal context profiles learnt by TCAM from Digg dataset for demonstration. By comparing the two adjacent temporal context profiles, we observe that the general public's preferences for time-oriented topics evolve over time. While the two adjacent time slices share the time-oriented topics $T1$ and $T16$, the general public focus more on topic $T1$ at time slice 6 and show more interest in topic $T6$ at time slice 7.

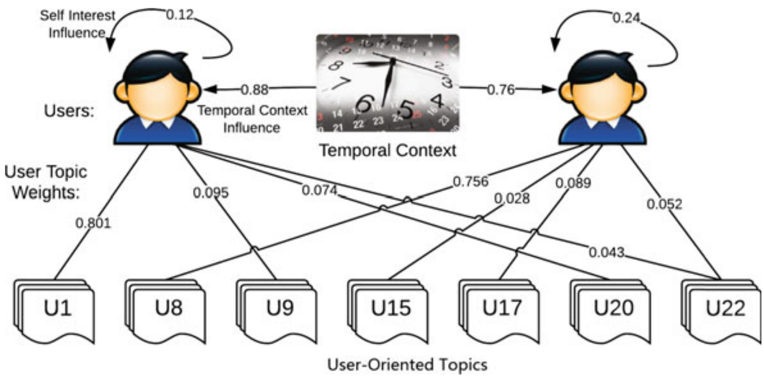

Fig. 2.7 Sample user profiles and temporal context influence learnt by TCAM

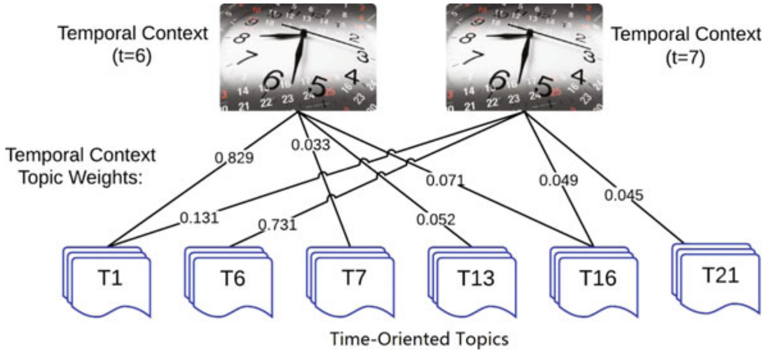

Fig. 2.8 Sample temporal context profiles

2.5 Summary

In this chapter, we focused on the problem of user behavior modeling in social media systems and its applications in temporal recommendation. Based on the intuition and observations that users' rating behaviors are influenced by two factors: user intrinsic interests as an internal factor, and the temporal context (i.e., the public's attention during a time period) as an external factor, we proposed a temporal context-aware mixture model (TCAM) that explicitly introduces two types of latent topics to model user interests and temporal context, respectively. An item-weighting scheme was developed to enhance the TCAM models by exploiting the frequency distribution and temporal distribution information of items. To demonstrate the applicability of TCAM, we deployed this model to facilitate temporal recommendation. We conducted extensive experiments on four large-scale real social media datasets, and the results and analysis demonstrated the superiority of our TCAM model over existing methods in the task of temporal recommendation, which verified our motivation. We also explored other applications for our TCAM model beyond time-aware recommendation, such as user profiling and temporal context modeling in an illustrative way in the experiment section.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. In: Journal of MachineLearning Research, pp. 993–1022 (2003)
2. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model. User-Adapt. Interact. **24**(1–2), 67–119 (2014)
3. Cha, Y., Bi, B., Hsieh, C.-C., Cho, J.: Incorporating popularity in topic models for social network analysis. In: SIGIR, pp. 223–232 (2013)
4. Cha, Y., Cho, J.: Social-network analysis using topic models. In: SIGIR, pp. 565–574 (2012)

5. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience, New Jersey (1991)
6. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: WWW, pp. 271–280 (2007)
7. Gantner, Z., Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Mymedialite: a free recommender system library. In: RecSys, pp. 305–308 (2011)
8. Hofmann, T.: Probabilistic latent semantic analysis. In: UAI (1999)
9. Lerman, K., Ghosh, R.: Information contagion: an empirical study of the spread of news on digg and twitter social networks. In: ICWSM (2010)
10. Liu, J., Dolan, P., Pedersen, E.R.: Personalized news recommendation based on click behavior. In: IUI, pp. 31–40 (2010)
11. Mei, Q., Zhai, C.: Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In: KDD, pp. 198–207 (2005)
12. Michelson, M., Macskassy, S.A.: Discovering users' topics of interest on twitter: a first look. In: AND, pp. 73–80 (2010)
13. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2009)
14. Rosen Zvi, M., Griffiths, T., Steyvers, M., Smyth P.: The author-topic model for authors and documents. In: UAI, pp. 487–494 (2004)
15. Stoyanovich, J., Amer Yahia, S., Marlow, C., Yu, C.: Leveraging tagging to model user interests in del.icio.us. In: AAAI (2008)
16. Wang, X., Zhai, C., Hu, X., Sproat, R.: Mining correlated bursty topic patterns from coordinated text streams. In: KDD, pp. 784–793 (2007)
17. Wen, Z., Lin, C.Y.: On the quality of inferring interests from social neighbors. In: KDD, pp. 373–382 (2010)
18. Wolfe, J., Haghighi, A., Klein, D.: Fully distributed em for very large datasets. In: ICML, pp. 1184–1191 (2008)
19. Xiong, L., Chen, X., Huang, T.-K., Schneider, J.G., Carbonell, J.G.: Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: SDM, pp. 211–222 (2010)
20. Xu, Z., Zhang, Y., Wu, Y., Yang, Q.: Modeling user posting behavior on social media. In: SIGIR, pp. 545–554 (2012)
21. Yin, H., Cui, B., Chen, L., Hu, Z., Huang, Z.:A temporal context-aware model for user behavior modeling in social media systems. In: SIGMOD, pp. 1543–1554 (2014)
22. Yin, H., Cui, B., Chen, L., Hu, Z., Zhou, X.: Dynamic user modeling in social media systems. ACM Trans. Inf. Syst. **33**(3), 10:1–10:44 (2015)
23. Yin, H., Cui, B., Li, J., Yao, J., Chen, C.: Challenging the long tail recommendation. Proc. VLDB Endow. **5**(9), 896–907 (2012)

http://www.springer.com/978-981-10-0747-7

Spatio-Temporal Recommendation in Social Media
Yin, H.; Cui, B.
2016, XIII, 114 p. 26 illus., 22 illus. in color., Softcover
ISBN: 978-981-10-0747-7