

Inferring Continuous Dynamic Social Influence and Personal Preference for Temporal Behavior Prediction*

Jun Zhang^{1,2,3,4} Chaokun Wang^{2,3,4} Jianmin Wang^{2,3,4} Jeffrey Xu Yu⁵

¹ Department of Computer Science and Technology, Tsinghua University

² School of Software, Tsinghua University

³ Tsinghua National Laboratory for Information Science and Technology

⁴ Key Laboratory for Information System Security, Ministry of Education, P. R. China

⁵ The Chinese University of Hong Kong

zhang-jun10@mails.thu.edu.cn, {chaokun, jimwang}@tsinghua.edu.cn, yu@se.cuhk.edu.hk

ABSTRACT

It is always attractive and challenging to explore the intricate behavior data and uncover people's motivations, preference and habits, which can greatly benefit many tasks including link prediction, item recommendation, etc. Traditional work usually studies people's behaviors without time information in a static or discrete manner, assuming the underlying factors stay invariant in a long period. However, we believe people's behaviors are dynamic, and the contributing factors including the social influence and personal preference for behaviors are *varying continuously over time*. Such continuous dynamics convey important knowledge about people's behavior patterns; ignoring them would lead to inaccurate models.

In this work, we address the continuous dynamic modeling of temporal behaviors. To model the fully continuous temporal dynamics of behaviors and the underlying factors, we propose the DP-Space, a dynamic preference probability space, which can capture their smooth variation in various shapes over time with flexible basis functions. Upon that we propose a generative dynamic behavior model, ConTyor, which considers the temporal item-adoption behaviors as joint effect of dynamic social influence and varying personal preference over continuous time. We also develop effective inference methods for ConTyor and present its applications.

We conduct a comprehensive experimental study using real-world datasets to evaluate the effectiveness of our model and the temporal modeling. Results verify that ConTyor outperforms existing state-of-the-art static and temporal models in behavior predictions. Moreover, in our detailed study on temporal modeling, we show that temporal modeling is superior to static approaches and modeling over continuous time is further better than that over discrete time. We also demonstrate that the ancient behavior data can still become important and beneficial if modeled well.

1. INTRODUCTION

The massive growing behavioral data have attracted increasingly interests from both the academia and industry. These data are usually valuable for many stakeholders, including the advertisers, op-

*Corresponding authors: Chaokun Wang and Jianmin Wang.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/>. Obtain permission prior to any use beyond those covered by the license. Contact copyright holder by emailing info@vldb.org. Articles from this volume were invited to present their results at the 41st International Conference on Very Large Data Bases, August 31st - September 4th 2015, Kohala Coast, Hawaii.

Proceedings of the VLDB Endowment, Vol. 8, No. 3
Copyright 2014 VLDB Endowment 2150-8097/14/11.

erators and merchants, because they convey important knowledge about what one prefers, how one behaves, and why one appears as what we observe. It motivates the study on behavior modeling and prediction, which focuses on *modeling how these behaviors happen and predicting what behaviors will happen*, respectively. In the literature, the personal preference and social influence are widely recognized as two major contributing factors for individual behaviors, and utilized to model, explain and predict the massive complicated behavior data [6, 25, 5, 29, 24, 19].

Existing work, however, usually ignored the dynamic nature of behavior data and the underlying factors. For behavior modeling, they usually assume the behaviors are independent with each other and do not consider the temporal information of behaviors. For behavior prediction, they usually predict whether specific behaviors will happen in the future, without indicating when they are expected to happen. The ignorance of temporality simplifies the problem, meanwhile limits the application of the proposed approaches.

As Heraclitus ever opined, “*No man ever steps in the same river twice, for it is not the same river and he is not the same man*”. People's friends, interests and attitudes are always changing over time, and thus their behaviors at different time should also be studied from the dynamic perspective, rather than the static perspective. As we know, the behavior data are usually time-stamped, and the temporality conveys important knowledge about why and how each behavior took place at specific time. In this paper, we address the problem of *temporal behavior modeling and prediction* (TBM & TBP) over continuous time domain. TBM and TBP are two coupled problems: Given the time-stamped behavior data, the TBM problem seeks for a model which explains how the data were produced in the past time, and based on that model the TBP makes predictions for unseen behaviors at future specific time.

1.1 Challenges

Unlike traditional work, in this study, we argue the world is continuously dynamic, and the continuous dynamics of behaviors and the underlying factors should be carefully modeled. It should be noted that when we say “*continuous dynamics*”, we don't mean the social influence or personal preference has to keep changing rapidly at any time. Indeed one's preference may stay relatively stable for some periods of time, but we believe *the change happens gradually, or continuously, not suddenly*, despite the change speed.

The TBM & TBP problem faces several unique challenges. Among them is the inference of the underlying dynamic factors, including the social influence and personal preference, which are usually invisible to us. Thus the first challenge is: *How can we infer the underlying social influence and personal preference from past behavioral data?*

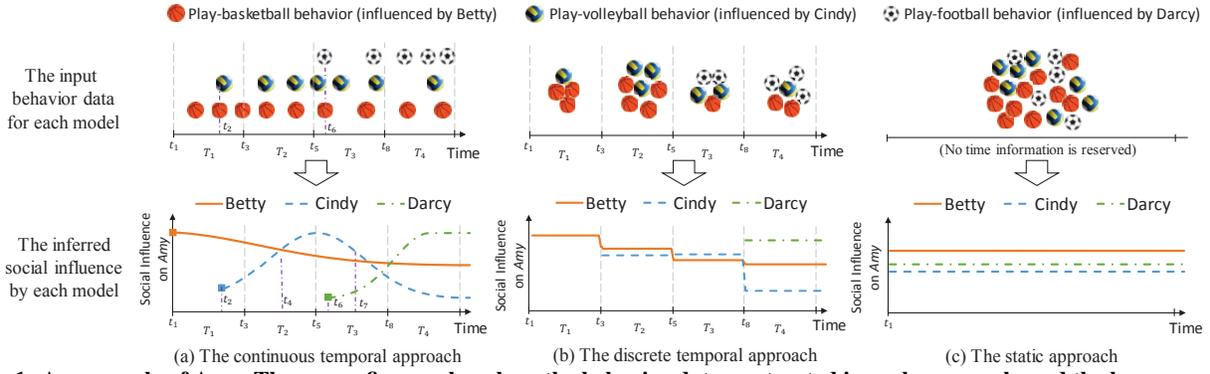


Figure 1: An example of Amy. The upper figures show how the behavior data are treated in each approach, and the lower ones show the inferred social influence curves for Amy with different approaches.

The second challenge lies in the modeling over the continuous dynamics of behaviors. Most of existing models are static or discrete, assuming the underlying factors are invariant in a long period. We argue that both personal preference and social influence are *varying with time*, and more importantly, the dynamics of them should be considered as *continuous over time*. Instead of modeling discrete behaviors, we believe continuous temporal modeling should have better capability for depicting past data and predicting future behaviors. Now we meet the second challenge: *How can we design and infer a fully-continuous temporal behavior model to explain the generation of past temporal behavior data?*

Except explaining the past data, an important application of the learned temporal behavior model is behavior prediction. Unlike traditional static prediction tasks which predict whether specific behaviors will happen in the future regardless of the time, in this work we focus on the temporal behavior prediction which makes predictions for specific time. Here we get our third challenge: *How can we make precise predictions for behaviors at specific time?*

1.2 Continuous vs. Static / Discrete Approach

In this subsection, we elaborate why traditional static or discrete models are insufficient for the TBM & TBP problem, and why continuous models are necessary. To make it concise, we take a simplified real-life story of Amy as example.

Amy made three friends (influencers), including Betty at t_1 , Cindy at t_3 and Darcy at t_6 , respectively. Betty likes basketball, Cindy likes volleyball and Darcy likes football. Amy likes all of the three sports (with equal personal preference) and her sport behaviors are mostly influenced by her friends. Thus, observing Amy played basketball at some specific time, we can infer the playing-basketball behavior was influenced by Betty. In Fig. 1, we illustrate the *input* of each approach, i.e. the preprocessed behavior data that can be treated in each approach, and the corresponding *output*, i.e. the inferred social influence curves for Amy with different approaches. As shown in Fig. 1(a), our continuous approach can capture and infer the detailed variance of social influence by processing the behaviors successively in chronological order.

The static approach. The static approach studies the behaviors after removing all the timestamps from the data, as shown in Fig. 1(c). In consequence, the temporal dynamics of social influence cannot be reflected.

The discrete vs. continuous approach. Discrete approaches usually address the temporality of behaviors via splitting the continuous log into discrete intervals w.r.t. time [28, 29, 5], and applying static models in each interval. By processing in batch, we get the step changes of social influence as Fig. 1(b). Though simple, the discrete approach suffers from several critical drawbacks.

First of all, *discretizing time leads to loss of many dynamic details, such as the temporal distribution of behaviors*. As shown in Fig. 1(a), Betty contributed to most of Amy’s behaviors at the beginning, but her influence on Amy decreased gradually. The frequency of Cindy-influenced behaviors increased first and afterwards dropped when Amy met Darcy, who had increasingly important influence on Amy. These detailed dynamics can be entirely captured by our continuous approach. However, since the detailed time information is erased in the discrete approach, we can only see the average influence in each interval (as in Fig. 1(b)) and the mutations of social influence between intervals, but cannot tell why.

Furthermore, *discrete approaches usually erase the temporal orders of intra-intervals behaviors, and only reserve the inter-interval orders*. The orders of behaviors are critical because one’s behaviors are not independent with each other. Specifically, the friend-making behaviors are usually preconditions of other subsequent behaviors. In our example, Cindy became Amy’s friend at $t_2 \in T_1$, and influenced the playing-volleyball behaviors after t_2 . However, the discrete approach erases the detailed time of the behaviors and only reserves the coarse-grained interval information, as shown in Fig. 1(b). In this way, Amy’s behaviors of making-friend-with-Cindy and playing-volleyball in the same interval T_1 are assumed to have happened meanwhile and independently, and thus Cindy cannot be regarded as influential for Amy until T_2 . In contrast, the continuous approach considers the *total orders* of all behavior data, and can capture the influence of new friends once they appear.

Moreover, *the performance of discrete approaches relies heavily on the quality of discretization, but choosing the optimal granularity is not easy*. We face such a paradox: We expect to improve the performance and make the discrete approach “less discrete” by shortening each interval, but it comes at the expense of reducing the training data in each interval inevitably, and results in immature or over-fitted models, which perform even worse. Besides, decreasing the granularity also leads to the increase of both the interval amount and computation complexity. Conversely, the continuous approach bypasses such paradox by looking at the whole picture directly.

Final remarks. By modeling the detailed dynamics of social influence and personal preference over continuous time, our continuous approach not only overcomes the drawbacks of other approaches, but also enables us to *analyze the dynamic changes more finely and precisely*. For example, with the discrete approach we only know Betty was slightly important than Cindy in T_2 in Fig. 1(b). However, from Fig. 1(a) we can see more: Before t_4 Betty was the most important influencer and afterwards Cindy caught up.

Besides, continuous temporal modeling is also expected for *better predictive capability for future behaviors*. Discrete models can only capture the influence drifts between intervals in the past, while continuous models are capable of seeing the long-range dependen-

cy of social influence and personal preference over time and capturing the change tendency for future. Assume that we are given the training data by t_7 , when we see Cindy acted as the most important influencer, Betty came next and Darcy was the last. Beyond that we also observe the influence of Darcy was growing while those of others were weakening. Thus in the near future, we can predict Darcy may become more influential, as we observe in T_4 .

1.3 Contributions

The main contributions of this paper are summarized as follows.

Firstly, we present approaches to depicting the full dynamics of social influence and personal preference over continuous time with **DP-Space**, a temporal dynamic preference space, which can capture the smooth change tendency based on flexible mixtures of basis functions.

Secondly, we propose a generative temporal behavior model, i.e. **ConTyor**, in which the generation of behavior data is modeled as the joint effect of the dynamic social influence and varying personal preference over continuous time. We also present methods for model inference using **EMO** (expectation-maximization-optimization) and for the TBP task based on ConTyor.

Thirdly, we conduct comprehensive experiments on both synthetic and real-world datasets to evaluate the performance of our proposed approach. Results show that the ConTyor model outperforms both the state-of-the-art static and dynamic models, and can improve the performance of behavior prediction significantly.

Finally, we study the effect of temporal modeling from multiple aspects through experiments. The results lead us to draw the conclusion that temporal modeling is greatly superior to static approaches and modeling over continuous time is further better than that over discrete time. Unlike the observations in our previous work [29], the ancient behavior data can still become important and beneficial for model learning if modeled well.

1.4 Roadmap

We will describe our problem firstly in Sec. 2. In Sec. 3 we present the ConTyor model, our approach to modeling and predicting continuous dynamic behaviors. Afterwards we illustrate some applications in Sec. 4. The effectiveness of the proposed model is evaluated and demonstrated in Sec. 5. In Sec. 6 we survey the related work and then conclude this study in Sec. 7.

2. PROBLEM DESCRIPTION

In this study, we focus on the temporal item-adoption behaviors. Let $U = \{u_1, u_2, \dots, u_N\}$ and $V = \{v_1, v_2, \dots, v_M\}$ be the user set and item set, respectively. Each temporal item-adoption behavior is an event that a user u adopts an item v at time t , denoted by a tuple (u, v, t) . The item here is a general term which can be anything in various scenarios. For example, one may play sports (where a sport is an item), buy a product in an online shopping site (where a product is an item), or watch a movie in a cinema (where a movie is an item). The time $t \in \mathbb{R}^+$ is a continuous real number indicating the timestamp. As the social influence is usually taken into consideration, we use $F_u^{(t)}$ to denote the friends of u at time t , and let \mathbb{G} record all the friendships of all users at all times.

Without loss of generality, we can record the behavior data of u as a sequence $\mathbb{B}_u = \{\langle v_1, t_1 \rangle, \langle v_2, t_2 \rangle, \dots, \langle v_{m(u)}, t_{m(u)} \rangle\}$, and the whole behavior data of all users as the set $\mathbb{B} = \bigcup_{u \in U} \mathbb{B}_u$.

In the TBM problem, we would like to seek a model \mathcal{M} which explains the generation process of the data \mathbb{B} best. We exploit the generative behavior models, which describe the probabilistic generative process of the observed behavior data and target at max-

imizing the joint probability of all the observed data and hidden variables. Formally, the TBM problem can be described as follows:

The Temporal Behavior Modeling (TBM) Problem. Given the observed data $\mathbb{D} = \{U, V, \mathbb{G}, \mathbb{B}\}$, the TBM problem seeks for the optimal \mathcal{M} which maximizes the posterior probability $P_{\mathcal{M}}(\mathbb{B}|U, V, \mathbb{G})$ given by the model \mathcal{M} , i.e. :

$$\arg_{\mathcal{M}} \max P_{\mathcal{M}}(\mathbb{B}|U, V, \mathbb{G}). \quad (1)$$

In the second problem, TBP, we consider to what extent we can make predictions for future behaviors based on the model given by the TBM solution. Usually, for any specific user at any specific time, TBP accepts a set of candidate items and then outputs a ranked list, ordered by the probability that the user adopts each item at that time. We define the TBP problem formally as follows:

The Temporal Behavior Prediction (TBP) Problem. Given the observed data $\mathbb{D} = \{U, V, \mathbb{G}, \mathbb{B}\}$, for any specific user u at time t , and a candidate item set V' , the TBP problem estimates the adoption probability of $P(v|u; t)$ that u would adopt each item $v \in V'$ at t , and outputs a ranked list V^L of the candidate items, ordered by the adoption probability.

Generally speaking, the TBM problem is the basis for the TBP problem, while the latter is an important application of the former.

3. TEMPORAL BEHAVIOR MODELING & PREDICTION

In this section, we present the **ConTyor** (short for **Continuous Temporal Dynamic Behavior**), a generative temporal model for item-adoption behaviors driven by dynamic social influence and varying personal preference over continuous time. In ConTyor, everything is dynamic: the friends, interests, items, and the relations among them. By modeling a fully dynamic system, we try to understand how people behave in the changing world, with evolving social network and varying external environment.

We will first introduce how we depict the continuous dynamics of social influence and personal preference, and then introduce the proposed ConTyor model. Then the implementation details and the inference algorithm are given. At last we present how to make temporal behavior predictions based on ConTyor.

3.1 Modeling of Temporal Dynamics

Personal preference and social influence have been recognized as two important factors for item-adoption behaviors [25, 29, 5]. However, the dynamics of them haven't been fully studied yet. They are usually either assumed static [25], or treated in a discrete way [29, 5] in existing work. In this study, we believe both the personal preference and social influence are continuously varying with time, no matter slowly or quickly.

Firstly, we define the preference space as follows.

Definition 1. The **preference space** for a subject s , denoted by $\Lambda(s) = (O_s, p_s)$, is a probability space which describes the preference probability of the subject towards specific objects. O_s is the *object set* (a.k.a. the sample space) corresponding to s , and p_s is the *preference probability function* which gives the probability that s prefers each object $o \in O_s$.

Traditional static approaches usually model one's behaviors using a single static preference space. In order to incorporate the temporal information, discrete approaches usually split the time into intervals, and for each interval define a static preference space. However, the preferences of subjects towards other objects are usually gradually changing all the time. Here, we propose to model such continuous dynamics with time using *DPS functions*.

Definition 2. The **dynamic preference strength (DPS)** function for a pair of subject s and object o , denoted by $q_s(t; o)$, is a

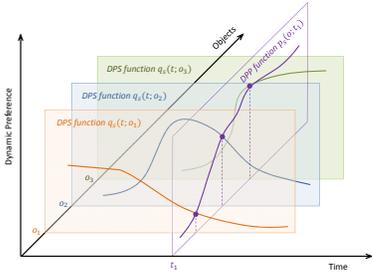


Figure 2: An illustration of the DP-Space $\Omega(s)$.

real-valued continuous function over time, measuring the strength of the preference of s towards o at any time t .

The DPS function can be implemented in any appropriate functions. We will discuss our implementations later in Sec. 3.3. It should be noted that the above definition doesn't conform to the probability definition, i.e. $q_s(t; o)$ gives a real-valued measure of preference, instead of a probability. To measure the preference probability of a subject towards all objects at any specific time, we introduce the following definition.

Definition 3. The **dynamic preference probability (DPP)** function for a pair of subject s and object o , denoted by $p_s(o; t)$, measures the probability that s prefers o at any time t .

What are assumed dynamic here include not only the preference, but also the subjects and objects. At any time t , the dynamic preference probability $p_s(o; t)$ can be calculated as follows:

$$p_s(o; t) = \begin{cases} \frac{q_s(t; o)}{\sum_{o' \in O_s(t)} q_s(t; o')} & \text{if } o \in O_s(t) \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

where $O_s(t)$ is the corresponding object set of s at time t .

Now we can define the *dynamic preference space* in a continuous dynamic world. It is also illustrated in Fig. 2.

Definition 4. The **dynamic preference space (DP-Space)** for a subject s , denoted by $\Omega(s) = (O_s, p_s, q_s, \tau_s)$, is a probability space which describes the temporal varying preference probability of the subject s towards specific objects in O_s over continuous time. O_s is the *object set* corresponding to s , p_s is the *dynamic preference probability* function, q_s is the *dynamic preference strength* function, and $\tau_s(o, t)$ is an indicator function checking whether an object o appears in O_s at time t . We define $O_s^{(t)} = \{o \in O_s | \tau_s(o, t) = 1\}$.

3.2 Generation of Temporal Behaviors

Now we study the generative process of temporal behavior data. This process can be regarded as the interplay of multiple entities, including the individuals/users, their friends and interest areas, and the items. In the following, we discuss our generative model in which each temporal behavior denoted by the triple (*user, item, time*), e.g. (u, v, t) , is generated by three steps, including the temporal influencer generation, temporal interest generation and the final temporal item generation.

3.2.1 Temporal Influencer Generation

In the first step, we generate the influencers for this behavior (u, v, t) from u 's *DP-Space of Influencers*, which is defined as:

Definition 5. The **DP-Space of Influencers** is a DP-Space defined for users and measures one's temporal dynamic preference for influencers towards her friends.

Specifically, for user u , her DP-Space of Influencers is denoted by $\Omega^{\alpha}(u) = (O_u^{\alpha}, p_u^{\alpha}, q_u^{\alpha}, \tau_u^{\alpha})$, where $O_u^{\alpha} = F_u$, i.e. the set of u 's friends, p_u^{α} gives the temporal probability that u selects a friend as

influencer at any time, q_u^{α} measures the temporal strength of u 's influencer preference towards her friends, and the indicator function $\tau_u^{\alpha}(f, t)$ checks whether user f is a friend (candidate influencer) of u at t . We define $F_u^{(t)} = \{f \in F_u | \tau_u^{\alpha}(f, t) = 1\}$.

In this model, we consider a behavior as the result of the co-effect of temporal personal preference and social influence. To this end, we treat each user u as a special "friend" of herself, i.e. $F_u^{(t=0)} = \{u\}$, as one may behave independently without influence from others.

Thus at time t , the probability that u selects $f \in F_u$ from her DP-Space of Influencers as her influencer is:

$$P(f|u; t) = p_u^{\alpha}(f; t) = \begin{cases} \frac{q_u^{\alpha}(t; f)}{\sum_{f' \in F_u^{(t)}} q_u^{\alpha}(t; f')} & \text{if } f \in F_u^{(t)} \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

3.2.2 Temporal Interest Generation

Given the influencers, the next step is to generate the interest areas from the corresponding DP-Space of Interests, defined as:

Definition 6. The **DP-Space of Interests** is a DP-Space which measures one's temporal dynamic preference for interest areas.

Specifically, for user u , her DP-Space of Interests is denoted by $\Omega^{\beta}(u) = (O_u^{\beta}, p_u^{\beta}, q_u^{\beta}, \tau_u^{\beta})$, where $O_u^{\beta} = Z$ and the indicator function $\tau_u^{\beta}(z, t)$ checks whether the interest area z is available in Z at t . We define $Z^{(t)} = \{z \in Z | \tau_u^{\beta}(z, t) = 1\}$.

Thus at time t , given the influencer f , the probability that the interest area z is sampled from $\Omega^{\beta}(f)$ is:

$$P(z|f; t) = p_f^{\beta}(z; t) = \begin{cases} \frac{q_f^{\beta}(t; z)}{\sum_{z' \in Z^{(t)}} q_f^{\beta}(t; z')} & \text{if } z \in Z^{(t)} \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

3.2.3 Temporal Item Generation

The last step is to generate the item v from the DP-Space of Items w.r.t. the generated interest areas, defined as:

Definition 7. The **DP-Space of Items** is a DP-Space defined for interest areas and measures the temporal dynamic popularity of items in each interest area.

Specifically, for interest area v , its DP-Space of Items is denoted by $\Omega^{\gamma}(z) = (O_z^{\gamma}, p_z^{\gamma}, q_z^{\gamma}, \tau_z^{\gamma})$, where $O_z^{\gamma} = V$ and the indicator function $\tau_z^{\gamma}(v, t)$ checks whether the item v is available in V at t . We define $V^{(t)} = \{v \in V | \tau_z^{\gamma}(v, t) = 1\}$.

Thus at time t , given the interest area z , the probability that the item v is sampled from $\Omega^{\gamma}(z)$ is:

$$P(v|z; t) = p_z^{\gamma}(v; t) = \begin{cases} \frac{q_z^{\gamma}(t; v)}{\sum_{v' \in V^{(t)}} q_z^{\gamma}(t; v')} & \text{if } v \in V^{(t)} \\ 0 & \text{otherwise} \end{cases}. \quad (5)$$

3.2.4 The Overall ConTyor Model

As illustrated in Fig. 3, the above three generative steps constitutes the overall generative process of our ConTyor model. In ConTyor, the three steps are assumed independent with each other, and the objects of each DP-Space are also independent with each other. Each behavior is considered as the joint effect of all objects in each DP-Space. Let $\Phi = \{\Phi^{\alpha}, \Phi^{\beta}, \Phi^{\gamma}\}$ be all the parameter configurations for the three DP-Spaces and the objective ConTyor model \mathcal{M} . The overall generative probability of the behavior that u adopts an item v at time t in ConTyor is given by summing over all latent factors (f and z):

$$\begin{aligned} P(v|u; t, \Phi) &= \sum_{f \in F_u^{(t)}} P(v, f|u; t, \Phi) \\ &= \sum_{f \in F_u^{(t)}} P(f|u; t, \Phi) \sum_{z \in Z^{(t)}} P(z|f; t, \Phi) P(v|z; t, \Phi). \end{aligned} \quad (6)$$

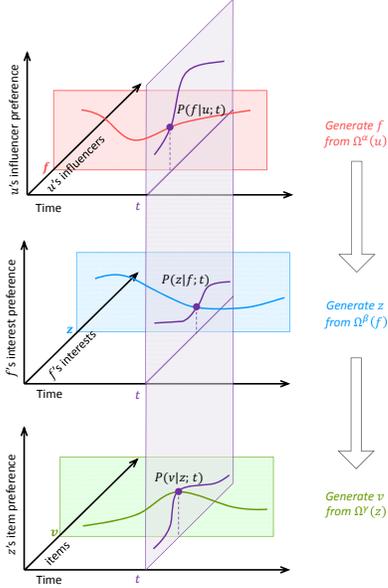


Figure 3: The generative process of behaviors in ConTyor.

Let \mathbb{F} and \mathbb{Z} be the hidden influencers and interest areas for the generation of observable behavior data \mathbb{B} . Given the parameters Φ , the log-likelihood of the given observed data can be written as:

$$\begin{aligned} \log L(\Phi; \mathbb{B}) &= \log P(\mathbb{B}|U; \Phi) = \log \left(\prod_{u \in U} \prod_{\langle v, t \rangle \in \mathbb{B}_u} P(v|u; t, \Phi) \right) \\ &= \sum_{u \in U} \sum_{\langle v, t \rangle \in \mathbb{B}_u} \log \left(\sum_{f \in F_u^{(t)}} P(f|u; t, \Phi) \sum_{z \in Z^{(t)}} P(z|f; t, \Phi) P(v|z; t, \Phi) \right). \end{aligned} \quad (7)$$

Now we calculate the log-likelihood of the complete data including both the unobservable \mathbb{F} and \mathbb{Z} and the observable \mathbb{B} :

$$\begin{aligned} \log L(\Phi|\mathbb{F}, \mathbb{Z}, \mathbb{B}|U, \Phi) &= \log P(\mathbb{F}, \mathbb{Z}, \mathbb{B}|U, \Phi) \\ &= \sum_{u \in U} \sum_{\langle v, t \rangle \in \mathbb{B}_u} (\log P(f|u; t, \Phi) + \log P(z|f; t, \Phi) + \log P(v|z; t, \Phi)). \end{aligned} \quad (8)$$

In the above formula, the log-likelihood of both the observed and unobserved data depends on the parameter configuration Φ , which needs to be estimated from the observed data.

3.3 Implementation of DPS Functions

As defined earlier, the DPS functions are the key components to capture the temporal dynamics. They can be implemented using various ways in different applications. Without loss of generality, we exploit the linear basis function model [2] for DPS functions in this study. It involves a linear combination of basis functions of the input variable t , of the form

$$q(t; \Phi) = w_0 + \sum_{i=1}^C w_i \varphi_i(t; \phi_i), \quad (9)$$

where each $\varphi_i(t; \phi_i)$ is a basis function, and $\Phi = \{\{w_0, \dots, w_C\}, \{\phi_1, \dots, \phi_C\}\}$ is the parameter set. The first term can be regarded as the static time-invariant factor while the subsequent items as time-variant factors. It is often convenient to define an additional dummy basis function $\varphi_0(t; \phi) = 1$ so that

$$q(t; \Phi) = \sum_{i=0}^C w_i \varphi_i(t; \phi_0) = \mathbf{w}^T \varphi(t; \Phi). \quad (10)$$

The parameter C controls the complexity of DPS functions. Generally speaking, increasing C exploits more basis functions to explain the preference variance and is expected for better fitting on the past observed data; however, it also comes with the risk of over-fitting and worse predictive power for the future. Thus the appropriate complexity should be decided according to specific applications.

The basis functions decide the shape of the DPS functions. In different applications the preference dynamics may appear in different shapes and thus we should select appropriate basis functions accordingly. Due to the limit of space, we list some popular ones in Appendix A. We will analyze and compare the capability of these basis functions for modeling social influence and personal preference dynamics through experimental study later in Sec. 5.6.

3.4 Inference Algorithm of ConTyor

The generation of temporal behavior data is explained using the ConTyor model \mathcal{M} parameterized by the configuration Φ . These parameters can be obtained by maximizing the likelihood defined in Eq. 8. However, the existence of the hidden variables makes it intractable to learn the parameters directly. Even the expectation-maximization (EM) algorithm cannot be directly applied due to the introduction of temporal DPS functions. Thus here we extend the EM approach to expectation-maximization-optimization (EMO) with an additional optimization step (O-step) except the traditional E-step and M-step. EMO calculates the expectation of the log-likelihood with current estimates of parameters in the E-step, and then maximizes this expectation to get new estimates of DPS function values, and finally looks for the optimal DPS functions by optimizing the parameter configurations. We explain this in detail as follows.

3.4.1 E-Step

Assume that we are given the estimates for the parameters as Φ^g . In the E-step, for each observable behavior (u, v, t) , we compute the posterior distribution of hidden variables f and z firstly:

$$P(f, z|u, v, t, \Phi^g) = \frac{P(f, z|u; t, \Phi^g) \cdot P(v|u, f, z; t, \Phi^g)}{P(v|u; t, \Phi^g)} \quad (11)$$

and then compute the expectation of the log-likelihood of all observed and hidden data as:

$$\begin{aligned} Q(\Phi, \Phi^g) &= \mathbb{E}(\log L(\Phi|\mathbb{F}, \mathbb{Z}, \mathbb{B})|\Phi^g) \\ &= \sum_{u \in U} \sum_{\langle v, t \rangle \in \mathbb{B}_u} \sum_{f \in F_u^{(t)}} \sum_{z \in Z^{(t)}} \left(\log p_u^\alpha(f; t, \Phi^g) \right. \\ &\quad \left. + \log p_f^\beta(z; t, \Phi^g) + \log p_z^\gamma(v; t, \Phi^g) \right) \cdot P(f, z|u, v, t, \Phi^g). \end{aligned} \quad (12)$$

3.4.2 M-Step

Instead of maximizing the log-likelihood defined in Eq. 8, the EMO algorithm looks for the optimal parameter configuration which maximizes the above expectation of the log-likelihood. However, the maximization is neither straightforward nor easy because it involves evaluations of the summations in the denominators of the DPP functions, as defined in Eq. 3–5. We avoid that by regularizing the values of DPS functions using probability constraints, and thus we can rewrite the Q -function as:

$$\begin{aligned} Q(\Phi, \Phi^g) &= \sum_{u \in U} \sum_{\langle v, t \rangle \in \mathbb{B}_u} \sum_{f \in F_u^{(t)}} \sum_{z \in Z^{(t)}} \left(\log q_u^\alpha(t; f, \Phi^\alpha) \right. \\ &\quad \left. + \log q_f^\beta(t; z, \Phi^\beta) + \log q_z^\gamma(t; v, \Phi^\gamma) \right) \cdot P(f, z|u, v, t, \Phi^g) \\ &\quad + \lambda_1 \sum_{u \in U} \sum_t \left(1 - \sum_{f' \in F_u^{(t)}} q_{u'}^\alpha(t; f', \Phi^\alpha) \right) \\ &\quad + \lambda_2 \sum_{f \in F} \sum_t \left(1 - \sum_{z' \in Z^{(t)}} q_f^\beta(t; z', \Phi^\beta) \right) \\ &\quad + \lambda_3 \sum_{z \in Z} \sum_t \left(1 - \sum_{v' \in V^{(t)}} q_z^\gamma(t; v', \Phi^\gamma) \right), \end{aligned} \quad (13)$$

where λ_1 , λ_2 and λ_3 are the Lagrange multipliers.

In the above equation, each DPS function of q^α , q^β and q^γ is a combination of basis functions. Since it is complex to obtain the parameters for them by maximizing the expectation directly, we divide the maximization into M-step and O-step. In this M-step, we only learn the optimal function values at each time t , which maximize the expectation.

As the three DPS functions are independent with each other, their optimal values can be obtained respectively via a similar procedure. Here we only present the process for q^α as an example for the limit of space. For each user u and one of her friend f , we take the derivative of the Q -function at any time t w.r.t. $q_u^\alpha(t; f, \Phi^\alpha)$:

$$\frac{\partial Q(\Phi, \Phi^g)}{\partial q_u^\alpha(t; f, \Phi^\alpha)} = \sum_{(v,t) \in \mathbb{B}_u} \sum_{z \in Z^{(t)}} \frac{P(f, z|u, v; t, \Phi)}{q_u^\alpha(t; f, \Phi^\alpha)} - \lambda_1 = 0, \quad (14)$$

and we get

$$\lambda_1 q_u^\alpha(t; f, \Phi^\alpha) = \sum_{(v,t) \in \mathbb{B}_u} \sum_{z \in Z^{(t)}} P(f, z|u, v; t, \Phi). \quad (15)$$

Summing both sides over f , we get:

$$\lambda_1 = \sum_{(v,t) \in \mathbb{B}_u} \sum_{z \in Z^{(t)}} P(z|u, v; t, \Phi). \quad (16)$$

Thus the optimal value of $q_u^\alpha(t; f, \Phi^\alpha)$ is given as :

$$\tilde{q}_{u,f}^\alpha[t] = \frac{\sum_{v \in \mathbb{B}_u^{(t)}} \sum_{z \in Z^{(t)}} P(f, z|u, v; t, \Phi^g)}{\sum_{v \in \mathbb{B}_u^{(t)}} \sum_{z \in Z^{(t)}} P(z|u, v; t, \Phi^g)}. \quad (17)$$

Similarly, we get:

$$\tilde{q}_{f,z}^\beta[t] = \frac{\sum_{u \in U} \sum_{v \in \mathbb{B}_u^{(t)}} P(f, z|u, v; t, \Phi^g)}{\sum_{u \in U} \sum_{v \in \mathbb{B}_u^{(t)}} P(f|u, v; t, \Phi^g)}, \quad (18)$$

$$\tilde{q}_{z,v}^\gamma[t] = \frac{\sum_{u \in U} \sum_{f \in Y_t(u)} P(f, z|u, v; t, \Phi^g)}{\sum_{u \in U} \sum_{f \in F_u^{(t)}} \sum_{v' \in \mathbb{B}_u^{(t)}} P(f, z|u, v'; t, \Phi^g)}. \quad (19)$$

3.4.3 O-Step

In the O-step we seek for the optimal parameters for the optimal DPS function values. For each function q , assuming \mathbf{t} records the time sequence of all behaviors and $\tilde{\mathbf{q}}$ stores the corresponding optimal values obtained in the M-step, the optimal parameters Φ^* for q can be obtained by solving the following optimization problem:

$$\Phi^* = \arg \min_{\Phi} \sum_{t \in \mathbf{t}} \left(q(t; \Phi) - \tilde{\mathbf{q}}[t] \right)^2 + \lambda(\Phi), \quad (20)$$

where λ is a regularization function.

It can be solved easily using the gradient descent method. The details are omitted here for the space limitation.

The EMO algorithm starts with randomly-set parameters Φ^g , and then repeats the E-step, M-step and O-step iteratively to improve the estimates of model parameters until convergence.

3.5 Temporal Behavior Prediction

Providing the fully-continuous temporal modeling on dynamic behaviors, ConTyor enables us to make temporal behavior predictions for any specific time or interval.

In the TBP problem, we're concerned about that at some specific time what behaviors may happen. For our concerned user u at specific time t , we generate a candidate set V' of behavioral items, which are usually a subset of V , and then for each item $v \in V'$, we calculate the temporal adoption probability as follows:

$$P(v|u; t, \Phi) = \sum_{f \in F_u^{(t)}} P(f|u; t, \Phi) \sum_{z \in Z^{(t)}} P(z|f; t, \Phi) P(v|z; t, \Phi). \quad (21)$$

Then we sort all the candidate items by the probability and produce the ranked list.

Instead of predicting behaviors at specific absolute time, sometimes we may be more interested in predicting behaviors in some interval denoted by $[t_s, t_e]$, where t_s and t_e indicate the start and

the end of the interval respectively. For specific behavior v in specific interval $[t_s, t_e]$, its probability can be estimated as:

$$P(v|u; [t_s, t_e], \Phi) = \int_{t \in [t_s, t_e]} P(v|u; t, \Phi). \quad (22)$$

It is simple since $P(v|u, t, \Theta)$ is a continuous function over t .

4. APPLICATIONS

ConTyor is a general model for temporal behavior modeling and prediction and can be applied into various areas. Without loss of generality, we present three interesting applications in the scenario of academic research and collaboration, where researchers usually collaborate to conduct research and publish papers, and one's research interests and writing habits are usually influenced by her current or previous coauthors. In this scenario, one's influencers (friends) are all researchers who have ever coauthored with her, and the interests are usually the natural research areas (e.g. database, data mining, machine learning, etc.).

Paper Content Prediction. In this task, we predict the content of one's paper at specific time. We can regard the paper writing as the process consisting of a sequence of *word-adoption behaviors*. Each temporal word-adoption behavior refers to an event that the author employs a word in her paper at specific time, and the item of each behavior is a word in the paper. By predicting the most possible word-adoption behaviors for specific time, we can predict the content of the paper written at that time.

Citation Prediction. An important part of a research paper is the references since one needs to review and cite the important related work. Also, the references of each paper may be different according to different topics and contents. We can also view the references as the result of a sequence of *citation-adoption behaviors*, which refer to the events that the author cites a previous paper in her own work. Here each behavior item is a paper to be cited. By predicting the most likely citation-adoption behaviors, we may predict the references of a paper.

Collaborator Prediction. As the collaboration is quite popular in academic research, we are also interested in that who an author may collaborate with for each new study. In this task, we focus on the *coauthor-adoption behaviors*, which refer to the events that the author invites other researchers as her coauthors of her paper. In this task the coauthors are not only items, but also regarded as friends (influencers) who may influence her subsequent behaviors. By predicting the coauthor-adoption behaviors with high probability, we can predict the co-authors of a paper.

Beyond the aforementioned ones, ConTyor can also be applied for many other tasks. Due to the limit of space, we would like to leave them for future work. In the next section, we will evaluate our proposed model and methods with the above applications.

5. EXPERIMENTAL EVALUATION

In this section, we evaluate our proposed model for behavior modeling and prediction on both synthetic and real-world datasets with the aforementioned applications. We introduce our experimental settings in the first three subsections, and then report the details of the experimental results. Finally we present a case study on the dynamic analysis of social influence for a real-world dataset.

5.1 Baselines

We evaluate our model by examining its performance in temporal behavior modeling and prediction. In this study, we consider two kinds of baseline methods. The first group consists of the state-of-the-art static models, including:

- **pLSA** [8], a classic latent factor model which models behaviors based on personal interests.

Table 1: The statistics of our experimental datasets (where the item density is defined as the ratio of #items / #users).

| dataset | # users | # friendships | # publications | # words | # word -adoptions | # citation -adoptions | # coauthor -adoptions | word density | publication density | friendship density |
|-----------|---------|---------------|----------------|---------|-------------------|-----------------------|-----------------------|--------------|---------------------|--------------------|
| sci-comp | 181,035 | 580,551 | 178,363 | 69,803 | 2,165,143 | 3,602,373 | 665,512 | 0.99 | 0.39 | 3.68 |
| comp-edu | 62,071 | 193,475 | 47,308 | 24,451 | 640,094 | 561,754 | 214,594 | 0.76 | 0.39 | 3.46 |
| simu | 38,832 | 118,753 | 29,200 | 15,941 | 398,147 | 587,567 | 132,718 | 0.75 | 0.41 | 3.42 |
| sec-priv | 65,884 | 232,873 | 63,122 | 29,493 | 827,426 | 1,600,772 | 279,488 | 0.96 | 0.45 | 4.24 |
| comp-ling | 15,835 | 98,307 | 19,424 | 10,709 | 309,637 | 243,102 | 126,508 | 1.23 | 0.68 | 7.99 |

- **LFPM** [29], a latest static generative model which was originally designed for modeling friend-making behaviors based on the *friendship propagations*. Here we extend it for general behavior prediction based on *behavior propagations*.
- **SIS** [25], a state-of-the-art static behavior model which incorporates the social influence and personal interests jointly for behavior prediction.

Another group of latest methods consider the temporal information in behavior prediction:

- **TimeSVD++** [12], a time-changing factor model which exploits the varying personal interests.
- **CasLFPM** [29], the cascade LFPM model, which considers the temporal variance of social influence in discrete manner.
- **RTM** [5], a state-of-the-art discrete temporal model with joint modeling of user interest and social relationships based on the Markov assumption.

5.2 The Data

We employed both real-world and synthetic data for evaluation.

5.2.1 Real-world Data

Although many datasets for behavior prediction are available, hardly any of them contain the temporal information of social ties. Thus we construct datasets from real-world academic collaborative networks as it has been argued extensively that these networks capture many key features of social networks more generally [17].

Specifically, we have constructed 5 real-world datasets from different research areas. The first four, including **sci-comp** (Science Computation), **comp-edu** (Computer Education), **simu** (Simulation) and **sec-priv** (Security & Privacy), are from Microsoft Academic Search¹. Another one, **comp-ling** (Computational Linguistics), is from the University of Michigan CLAIR Group’s ACL Anthology Network².

For each dataset, we evaluate three kinds of behaviors, including *word-adoption*, *citation-adoption* and *coauthor-adoption* in the aforementioned three applications respectively. The statistics of the datasets are shown in Tbl. 1.

5.2.2 Synthetic Data

To evaluate to what extent our model can infer the temporal preference, we generated a synthetic dataset, because the real preferences cannot be observed in real-world data. The synthetic dataset was generated in the way of the ConTyor model. We generated 10,000 users and 10,000 items randomly, and set the number of interest areas $K = 10$. The parameters of DPS functions were randomly set. We generated the behaviors for consecutive 30 time slots. At each slot, we randomly selected 5-10 new friends for each user, and generated 5-10 behaviors according to ConTyor model.

5.3 Evaluation Methodology

We consider two evaluation tasks. In the first one, we evaluate to what extent our model can capture and infer the temporal preference variance; in the second one, we evaluate to what extent our model can predict the behavior at specific time.

¹<http://academic.research.microsoft.com>

²<http://clair.eecs.umich.edu/aan>

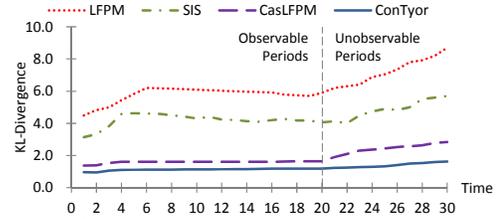


Figure 4: Preference inference performance on synthetic data.

5.3.1 Preference Inference Evaluation

We conducted this experiment on the synthetic data. To evaluate the capability of our model for recovering the past preferences and predicting their future variance, we assumed only the data between time 1-20 were observable and fed them to the models, and left the remaining data between 21-30 for testing.

We apply our ConTyor model and the baseline models to infer the temporal preference probability of each user at each time. Then we compare the inferred probability distribution with the ground-truth, which we can get easily given the original DPS functions. The difference at each time was evaluated using KL-divergence. A better model should produce a preference distribution which has smaller KL-divergence with the ground-truth.

5.3.2 Behavior Prediction Evaluation

Unlike traditional behavior prediction tasks which usually concern whether specific behaviors will happen in the near future, we make predictions for behaviors at specific time.

For each real-world dataset, the data in 1981–2000 were used for model training, and those in 2001–2005 for testing. Only the individuals with at least 5 friends by 2000 were considered. For discrete methods, because the data are quite sparse, we adopt the most natural and popular approach by splitting the behavior data by years (i.e. each year is an interval). To be fair, for our ConTyor, one year is adopted as the basic time unit. Then we make predictions for behaviors in each test interval. In the experiments we surprisingly found that even with the same “time granularity”, the proposed model can still outperform the discrete ones.

For each user at each year, her adopted items in each behavior category were regarded as the positive (relevant) items. The negative (irrelevant) items were selected randomly with the constraints that its amount was at most tenfold number of positive items for the sake of efficiency. It should be noted that we only predict the occurrence of each behavior in each interval and don’t consider the number of the occurrences.

We evaluate the prediction performance using MAP (Mean Average Precision) and AUC (Area Under the ROC Curve). Generally speaking, MAP measures how well the algorithm ranks positive items above negative items, and AUC measures how well the algorithm distinguishes positive items from negative items.

5.4 Comparison of Preference Inference

As the interests are interpreted as latent factors in all models, it’s difficult to align and compare the interests inferred by different models. Thus we focus on the preference in the DP-Space of influencers, i.e. one’s temporal preference for influencers. We

Table 2: Average performance of word-adoption prediction.

| Dataset | Metric | Static Models | | | Dynamic Models | | | ConTyor |
|-----------|--------|---------------|--------|--------|----------------|---------|---------------|---------------|
| | | pLSA | LFPM | SIS | TimeSVD++ | CasLFPM | RTM | |
| sci-comp | MAP | 0.6861 | 0.6799 | 0.7171 | 0.7139 | 0.6835 | 0.7209 | 0.7326 |
| | AUC | 0.8757 | 0.8740 | 0.8929 | 0.8882 | 0.8837 | 0.9047 | 0.9140 |
| comp-edu | MAP | 0.6409 | 0.6418 | 0.7210 | 0.7231 | 0.6582 | 0.7497 | 0.7522 |
| | AUC | 0.7929 | 0.7736 | 0.8450 | 0.8293 | 0.7953 | 0.8648 | 0.8950 |
| simu | MAP | 0.7114 | 0.7264 | 0.7817 | 0.7488 | 0.7471 | 0.7601 | 0.7906 |
| | AUC | 0.8358 | 0.8322 | 0.8819 | 0.8553 | 0.8439 | 0.8971 | 0.9020 |
| sec-priv | MAP | 0.6088 | 0.6465 | 0.6591 | 0.6557 | 0.6677 | 0.6698 | 0.6540 |
| | AUC | 0.8525 | 0.8255 | 0.8830 | 0.8481 | 0.8655 | 0.8852 | 0.8938 |
| comp-ling | MAP | 0.6637 | 0.6566 | 0.6998 | 0.6647 | 0.6840 | 0.6932 | 0.7083 |
| | AUC | 0.8312 | 0.8364 | 0.8627 | 0.8704 | 0.8435 | 0.8667 | 0.8674 |

Table 3: Average performance of citation-adoption prediction.

| Dataset | Metric | Static Models | | | Dynamic Models | | | ConTyor |
|-----------|--------|---------------|--------|---------------|----------------|---------|---------------|---------------|
| | | pLSA | LFPM | SIS | TimeSVD++ | CasLFPM | RTM | |
| sci-comp | MAP | 0.5522 | 0.5063 | 0.5354 | 0.5553 | 0.5555 | 0.5640 | 0.5431 |
| | AUC | 0.7609 | 0.7137 | 0.7487 | 0.7294 | 0.7207 | 0.7702 | 0.7829 |
| comp-edu | MAP | 0.5302 | 0.5203 | 0.5414 | 0.5759 | 0.5407 | 0.5419 | 0.6332 |
| | AUC | 0.6379 | 0.6277 | 0.6593 | 0.6604 | 0.6749 | 0.6697 | 0.7394 |
| simu | MAP | 0.7073 | 0.7193 | 0.7806 | 0.7765 | 0.7568 | 0.7776 | 0.7823 |
| | AUC | 0.8205 | 0.8495 | 0.8570 | 0.8872 | 0.8782 | 0.8776 | 0.8685 |
| sec-priv | MAP | 0.6585 | 0.6680 | 0.6881 | 0.6843 | 0.6715 | 0.6824 | 0.6849 |
| | AUC | 0.8649 | 0.8370 | 0.8809 | 0.8905 | 0.8495 | 0.9091 | 0.9034 |
| comp-ling | MAP | 0.7401 | 0.7357 | 0.8049 | 0.8221 | 0.7512 | 0.8284 | 0.7728 |
| | AUC | 0.8409 | 0.8201 | 0.8820 | 0.8724 | 0.8253 | 0.8788 | 0.8794 |

Table 4: Average performance of coauthor-adoption prediction.

| Dataset | Metric | Static Models | | | Dynamic Models | | | ConTyor |
|-----------|--------|---------------|--------|--------|----------------|---------------|---------------|---------------|
| | | pLSA | LFPM | SIS | TimeSVD++ | CasLFPM | RTM | |
| sci-comp | MAP | 0.7048 | 0.7190 | 0.6832 | 0.6798 | 0.7208 | 0.6908 | 0.6625 |
| | AUC | 0.8165 | 0.7706 | 0.7947 | 0.8175 | 0.7763 | 0.8200 | 0.7750 |
| comp-edu | MAP | 0.7737 | 0.7451 | 0.7745 | 0.7784 | 0.7506 | 0.7899 | 0.8017 |
| | AUC | 0.8284 | 0.7992 | 0.8120 | 0.8100 | 0.8095 | 0.8385 | 0.8429 |
| simu | MAP | 0.8136 | 0.8001 | 0.7922 | 0.8198 | 0.8025 | 0.8151 | 0.8214 |
| | AUC | 0.8683 | 0.8231 | 0.8387 | 0.8374 | 0.8344 | 0.8472 | 0.8865 |
| sec-priv | MAP | 0.7400 | 0.7516 | 0.7677 | 0.7738 | 0.7549 | 0.7886 | 0.7824 |
| | AUC | 0.8679 | 0.8519 | 0.8770 | 0.8817 | 0.8546 | 0.8857 | 0.8996 |
| comp-ling | MAP | 0.7561 | 0.7612 | 0.7190 | 0.7141 | 0.7675 | 0.7431 | 0.7145 |
| | AUC | 0.8427 | 0.7912 | 0.7985 | 0.8040 | 0.7975 | 0.8148 | 0.7922 |

compare our ConTyor model with three of the baselines, including LFPM, SIS and CasLFPM, which consider the social influence in their models and thus can produce the influencer preferences. It should be noted that LFPM and SIS are static models and can only produce static average preference distributions, but we'll compare them with the ground-truth at each time respectively.

We illustrate the average results in Fig. 4, where we see the temporal models (ConTyor and CasLFPM) can recover the original preferences better than static models (LFPM and SIS), and our ConTyor model performs best. SIS performs better than LFPM because it considers the effect of interests and the latter does not. ConTyor outperforms CasLFPM by modeling the continuous dynamics and thus capturing the global preference change better.

We can also observe that these models show different ability of predicting future preference change. The preference probability distributions inferred by the static models (LFPM and SIS) deviate the original ones quickly after time 20, because they cannot capture the dynamic change of preference. CasLFPM also becomes much worse in the unobservable intervals. With the cascade discrete modeling, CasLFPM can capture the dynamics in the past, but becomes static after time 20 when no new data are available. On the contrary, ConTyor can capture the global variance tendency of the dynamic preference with smooth continuous temporal modeling and thus can predict the future change better.

5.5 Comparison of Behavior Prediction

In the second group of experiments, we compare the performance of our model with that of other popular ones for TBP tasks. We used the *quadratic basis function* as the DPS function of our model and set the number of basis functions of each DPS function $C = 1$ for simplicity.

5.5.1 Comparison of Average Performance

To get an overview of the performance of different methods, we first present the average performance for each behavior prediction in the 5 test intervals. The prediction results for *word-adoption*, *citation-adoption* and *coauthor-adoption* behaviors on each dataset are illustrated in Tbl. 2, Tbl. 3 and Tbl. 4, respectively. The best result of each test is highlighted in bold.

The pLSA and LFPM are two fundamental models among our baselines. The former makes predictions based on the latent interest space, while the latter models the behaviors as latent *behavior propagations*. LFPM is not promisingly better than pLSA here because in our datasets we did not observe many direct behavior propagations. SIS outperforms both of them by incorporating personal preference and social influence.

Among our baselines, CasLFPM is the corresponding temporal model of LFPM, and TimeSVD++ can be regarded as the counterpart of pLSA in some sense. We observe that the performance of dynamic models exceeds that of the static ones in most occasions by modeling the temporal change.

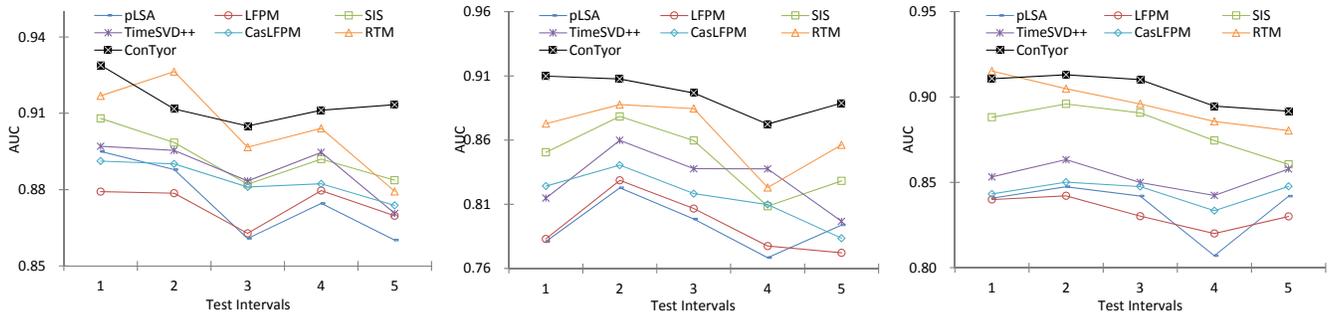
What SIS, RTM and the proposed ConTyor share in common is that all of them consider the joint effect of personal interests and social relationships on individual behaviors, though in different ways. The major distinction lies in their approach to temporal modeling. RTM adopts the similar way with LFPM. It splits the time into discrete intervals and assumes that the models evolving along the intervals form a Markov chain. Unlike them, ConTyor depicts the *continuous variance* of social influence and personal preference over time. We see that though RTM performs well on our datasets, the proposed ConTyor achieves significant improvement over the state-of-the-art methods including RTM in most situations.

When referring to the statistics in Tbl. 1, we surprisingly find our ConTyor usually wins when the data are sparse. For example, the densities of words, publications and friendships of the **comp-ling** dataset are larger than the counterparts of other datasets. Correspondingly, our ConTyor performs worse than some discrete models on **comp-ling**. This indicates that the ConTyor is better at capturing the latent global trends from sparse data, while discrete models are preferred when sufficient data are available so that the model can be fully optimized separately for each discrete interval locally.

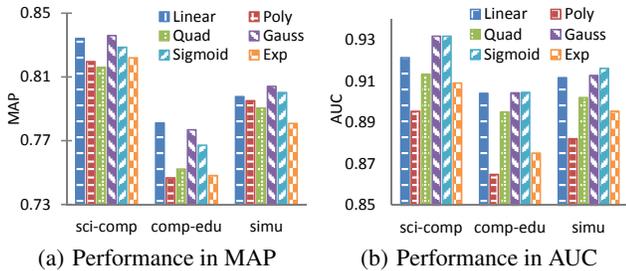
5.5.2 Comparison of Temporal Performance

Now we report the results of temporal behavior prediction in each test interval to observe the performance variance of the models with time. Due to the limit of space, we only present the results for the *word-adoption* behavior on three of our datasets in Fig. 5. Similar results were observed on other datasets and behaviors.

As shown before, our proposed ConTyor exhibits better performance than other baseline methods in most tests. Moreover, ConTyor is much more stable than others. Based on the Markov assumption, discrete approaches make predictions for each interval based on the behavior data in the last interval. Thus it may achieve relatively better performance for earlier test intervals, but become worse rapidly for later intervals. Both static and discrete approaches show more unstable performance than ConTyor because they cannot capture the dynamics and change tendency, and thus their prediction capability will inevitably decrease as time goes by. On the contrary, ConTyor can not only predict future behaviors based on its knowledge on previous data, but also predict the change tendency in the future with the learned continuous DPS functions. That is why we call ConTyor as a totally continuously dynamic behavior model: At different time, it adjusts its underlying parameters dynamically and makes the appropriate predictions accordingly.



(a) Performance comparison on **sci-comp** (b) Performance comparison on **comp-edu** (c) Performance comparison on **simu**
Figure 5: The temporal prediction performance of all models.³



(a) Performance in MAP (b) Performance in AUC
Figure 6: Effect of different DPS functions in ConTiyor.³

It should be noted that here we should study the temporal performance variance of models by studying their relative performance at each time point, rather than comparing the absolute AUC or MAP values at different time points. That is because the user behaviors at different time points are different, we cannot build the same evaluation dataset (especially the positive instances) for each time point, and thus the performances of a specific model at different time points are incommensurable.

5.6 Analysis of DPS Functions

The DPS functions are key components of the ConTiyor model as they depict the temporal dynamics of each factor. Here we study how to select appropriate DPS functions and control its complexity.

5.6.1 Comparison among DPS Functions

An interesting question about the DPS function is that what function can explain the temporal dynamics of social influence and personal preference better, and perform better in the prediction tasks. To figure this out, we adopt different DPS functions in the ConTiyor model and compare their performance for behavior prediction.

In Fig. 6 we show the average performance of the models using each DPS function for the *word-adoption* behavior on 3 datasets. We find that the *linear function*, *Gaussian function* and *sigmoid function* are shown better than other three, including *polynomial function*, *quadratic function* and *exponential function* in most situations. This is reasonable as the former three functions are gentler than the latter ones, and the social influence and personal preference usually change slowly and gradually. Thus the latter functions may be less appropriate for modeling such variance. For example, with the *exponential function*, the social influence is expected to grow/reduce increasingly rapidly after some specific time, which is usually rare in real world. On the contrary, *Gaussian function* describes the situation that one’s social influence from another increases firstly and then decreases slowly, or vice versa, while *Sigmoid function* depicts the monotonous variance of social influence.

³In Fig. 5-6 we report the results of *word-adoption* behavior prediction on *sci-comp*, *comp-edu* and *simu*.

Two traits make them the best functions for describing the temporal dynamics: 1) They are smoother and can model the gentle variance with less risk of over-fitting; 2) Both of them assume the variance will become slower with time goes by, and nearly stay at some value in far future. Besides, *Gaussian function* describes more complete lifecycle of social influence and personal preference including ups and downs. The *linear function* can be regarded as the simplified version of *sigmoid function*, however, it cannot model the decreasing speed of variance.

5.6.2 Complexity of DPS Functions

Now we study the relation between the performance of ConTiyor and the complexity of the inside DPS functions by observing how the prediction performance of ConTiyor varies when adjusting the complexity parameter C .

Experimental results of the *word-adoption* behavior prediction on **sci-comp** dataset are shown in Fig. 7. When $C = 0$, our ConTiyor degenerates to the basic static model and performs worse than that with larger C . Increasing C improves the performance at first and achieves the peak at around $C = 3$. After that the performance will decrease with the continuous increase of C due to over-fitting. The prediction for the 5th test interval suffers much more than that for the 1st one from that. It should be noted that increasing C also requires longer time for model inference, and thus in our experiments we find the most appropriate value for C is 2 or 3.

5.7 Analysis of Temporal Modeling

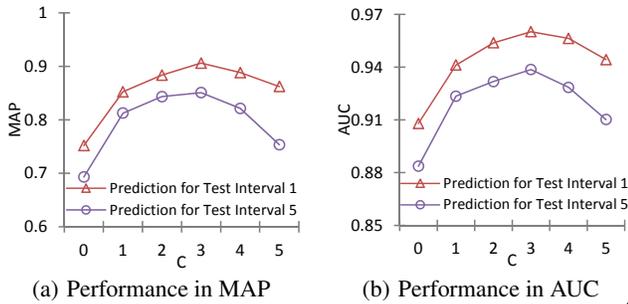
The most important characteristic of ConTiyor is the temporal modeling on the underlying factors. Although the performance of ConTiyor is verified in Sec. 5.5, we would like to examine the effect of temporal modeling from multiple aspects deeply. We seek answers for the following questions via experimental investigation:

- (1) Is the temporal modeling better than the static modeling?
- (2) Is the continuous modeling better than the discrete modeling?
- (3) Is the long-term modeling better than the short-term modeling?

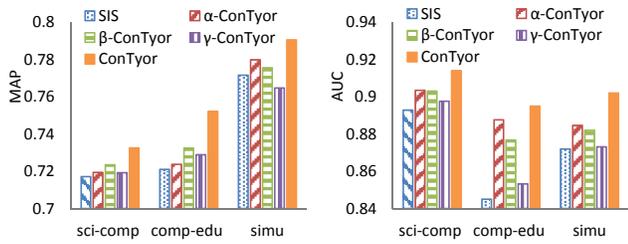
5.7.1 Temporal vs. Static Modeling

Here we try to answer the first question by investigating the effect of temporal modeling in detail. ConTiyor applies temporal modeling on all the three generative steps, including the influencer generation, interest generation and item generation. Alternatively, by only modeling the dynamics of each of them, we can get three variants: α ConTiyor, β ConTiyor and γ ConTiyor. We compare the performance of them with the static SIS and full dynamic ConTiyor model to evaluate the effect of temporal modeling on each step.

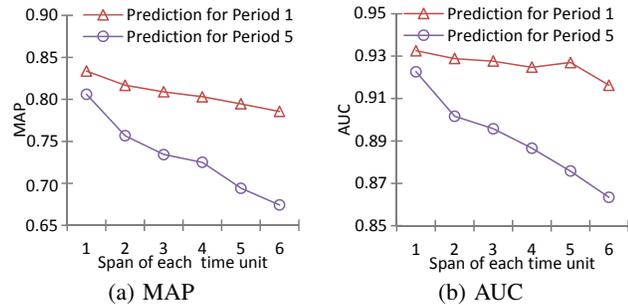
The experimental results are illustrated in Fig. 8. We can see the temporal modeling on any step achieves better performance over the basic SIS model in most situations. This demonstrates



(a) Performance in MAP (b) Performance in AUC
Figure 7: Effect of C (# of basis functions in DPS functions).⁴



(a) Performance in MAP (b) Performance in AUC
Figure 8: Comparison among the variants of ConTyor.⁵



(a) MAP (b) AUC
Figure 9: Effect of scale of time unit.⁵

that one’s preferences for influencers, interests and items are really evolving with time and appropriate modeling on such temporal dynamics can improve the inference for user profiles and prediction for user behaviors. We see that with temporal modeling on all steps, ConTyor achieves the best performance on all datasets. This proves the effectiveness of temporal modeling once again.

5.7.2 Continuous vs. Discrete Modeling

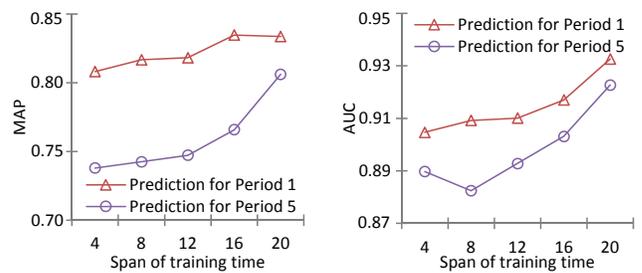
Unlike previous temporal models which usually model the drifts among discrete intervals based on the Markov assumption, ConTyor tries to capture the continuous and smooth variance of social influence and personal preference over time. Thus it is an interesting question that whether and to what extent the continuous modeling can improve the performance over discrete modeling?

Relative to the discrete models, the outperformance of ConTyor has been demonstrated in previous experiments in Sec. 5.5. Here we study it from another aspect by observing how its performance varies when we make ConTyor “discrete”.

As the minimal time unit of our training data is one year, we elongate the unit to 2, 3 or more year to see whether the performance would be affected. The results for *coauthor-adoption* behavior prediction on *sec-priv* dataset are shown in Fig. 9. The upper red curve

⁴In Fig. 7 we report the results of *word-adoption* behavior prediction on *sci-comp* dataset.

⁵In Fig. 8-14, we report the results of *coauthor-adoption* behavior prediction on *sec-priv* dataset.



(a) MAP (b) AUC
Figure 10: Effect of different training time.⁵

shows the prediction performance for the earliest test interval (Interval 1) and the lower purple curve for the last (Interval 5). We can see that with the elongation of the basic time unit, the prediction performance will decrease nearly monotonously. Furthermore, the performance in the last interval decreases much faster than that in the first one, which means the prediction for later intervals suffers more from the coarse-grained time unit. Similar results are observed on other datasets. Hence we can draw the conclusion that the continuous modeling can really improve the prediction performance by capturing more detailed and accurate temporal variance of latent social influence and personal preference. If we were given more fine-grained training data, we could expect much better performance of ConTyor.

5.7.3 Long-term vs. Short-term Modeling

Now Let us consider the third question: How long period of training data is sufficient for model learning? Although it has been shown that the *ancient* training data has little effect for future prediction in discrete temporal models [29], we still wonder whether there exists any difference when we meet the continuous ConTyor.

We conducted experiments to observe the performance variance with different length of training time. The results of the *coauthor-adoption* behavior prediction on *sec-priv* are shown in Fig. 10. We can see that the performance increases nearly monotonously with the elongation of training time. Furthermore, prediction for the last interval (Interval 5) benefits more from the longer training time than that for the first one. Unlike previous reports in discrete models, we find that in the continuous model the so-called *ancient data* is still quite useful because they can help to describe the temporal dynamics more finely and thus lead to better models.

5.8 Efficiency and Scalability Analysis

In this subsection we study the efficiency and scalability of our model. We implemented all the algorithms using Python 2.7.3, and conducted the experiments on a PC with a 12-threads CPU of Inter(R) Xeon(R) E5-2640 (2.50GHz) and 64 GB Memory. The operation system is Windows Server 2008. We report the results of *coauthor-adoption* behavior prediction on *sec-priv*.

Firstly we compare the time cost for model inference. As all of the compared methods are inferred iteratively, we report the running time of one iteration of each method in Fig. 11. We see the temporal models usually have to take more time for inference than static models, and more complex models require longer time. It should be noted that the model inferences are conducted off-line, thus it’s usually worthwhile to infer a better model to achieve more powerful predictive capability.

We also evaluated the convergence speed of the models. In Fig. 12 we compared ConTyor with the state-of-the-art static model (SIS) and dynamic model (RTM). We see SIS shows better performance at first, and converges after around 30 iterations. As a complex

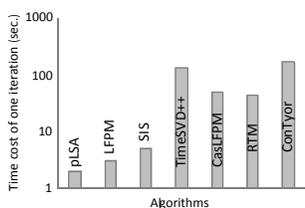


Figure 11: Running time.

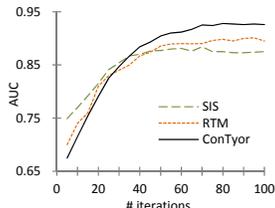


Figure 12: Convergence speed.

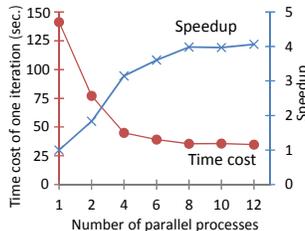


Figure 13: Parallel speedup.

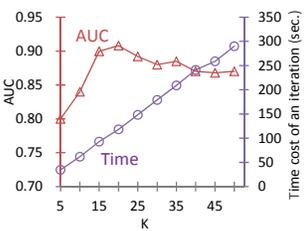


Figure 14: Effect of K .

model, ConTyor performs worst at the beginning, but its performance increases quickly after more iterations. Though SIS and RTM require less time to converge, the performance of ConTyor grows larger and increases faster with more iterations.

An important virtue of ConTyor is that its inference algorithm can be easily parallelized because the seeking for optimal DPS functions in the O-step is independent with each other. In this work we implemented the parallel inference algorithm, and evaluated to what extent the parallelism can accelerate the inference. We report our experimental results in Fig. 13. When we have 4 concurrent threads, the speedup ratio reaches 3; when we have 8, the speedup ratio achieves 4.

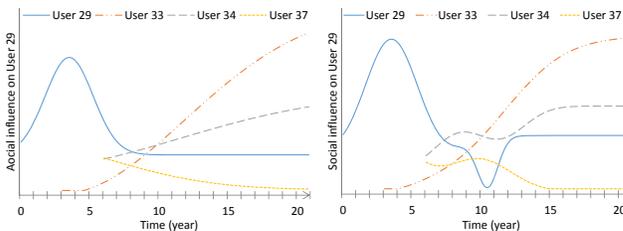
5.9 Parameter Analysis

Here we study the effect of the parameter K , i.e. the number of latent interest areas in Z . The performance variance with different K for *coauthor-adoption behavior prediction* on **sec-priv** dataset is shown in Fig. 14. Generally, increasing K moderately will lead to more detailed partition of the interest space, and further improve the model performance. Due to the size of the data, too large K will expose the sparsity problem, and result in an over-fitted model with worse performance. This has been verified in our experimental results, from which we find $K = 20$ performs best. Furthermore, increasing K also leads to nearly-linear growth of time for model inference, and thus we usually need to make a trade-off to choose the most appropriate K .

5.10 Case Study

Here, we show an example of social influence dynamics analysis with ConTyor. Fig. 15 visualizes the dynamics of social influence for user 29 in the **sec-priv** dataset. The results are from the inferred ConTyor model which utilized the Gaussian functions as basis of DPS functions and set the complexity parameter C as 1 and 3 respectively. Each curve represents the social influence from each of friends (including user 29 herself) of user 29 and starts when (s)he becomes a friend of the latter.

In Fig. 15(a), we see for user 29, she herself was the most important influencer at beginning when she had fewer friends. After the 3rd and 6th year, respectively, user 33 and user 34 had increasing influence on her gradually. User 37 became a friend of user 29 at the 6th year, but the influence of user 37 on user 29 is less than others. When we set a larger $C = 3$, as illustrated in Fig. 15(b), we can observe more detailed variance of the social influence.



(a) Learned curves when $C=1$ (b) Learned curves when $C=3$

Figure 15: The learned social influence curves of user 29 from the **sec-priv** dataset. We use the Gaussian basis functions and only show the most important influencers for user 29 for clarity.

6. RELATED WORK

In this section we survey lines of study related with ours.

Behavior Mining and Prediction. The mining and modeling of item-adoption behaviors has attracted many attentions. In the literature, it gets extensively studied in the area of recommender systems [1, 16], which focuses on recommending the most appropriate items to users based on their past adoption behavior data. The collaborative filtering technique [20] has been widely adopted and numerous methods have been proposed [10, 26, 18, 13, 4]. With the emergence of online social networks, the social relationships are found beneficial for such tasks [11, 9, 14, 27], because friends may influence each other and thus tend to exhibit similar behaviors [7]. Approaches have been proposed to incorporate the social relationships into predictive models [15, 14, 30, 28].

While focusing mainly on the item recommendation, less of above work studied the generation process of behaviors. Chua et al. proposed generative approaches to modeling item-adoption behaviors based on social correlation [6]. Ye et al. studied the direct impact of social influence on people’s behaviors and proposed the SIS behavior model [25]. However, they didn’t consider the temporal variation of social influence. Zhang et al. studied the temporal modeling for friend-making behaviors based on the friendship propagation by splitting the time into discrete intervals and proposed a propagation-based behavior model (LFPM) [29]. Similar idea was adopted by Chen et al. in their joint model (RTM) of user interest and social relationships over time [5]. Though simple and easy, their approaches by discretizing the time expose a number of deficiencies, such as the difficulty in selecting appropriate granularity, the incapability of capturing the correlation of behaviors in each interval, and the disability of capturing both the detailed variance and global tendency. In this paper, we propose to study the variance of social influence and personal preference from the perspective of continuous modeling, which overcomes the aforementioned drawbacks and enables us to capture both the microscopic and macroscopic variations simultaneously.

Temporal Modeling. Temporal modeling has also been studied in the topic models for documents. The DTM [3] and DMM [23] are two famous dynamic topic models for discrete data based on the Markov assumptions over state transitions in time. Unlike them, the TOT model [22] parameterizes a continuous distribution over time associated with each topic, but assumes the word distribution in each topic stays invariant. On the contrary, the cDTM [21] captures the continuous variance of the word distributions of each topic using a Markov chain modeling. The collaborative filtering with temporal dynamics [12] has also been studied. Our work differs from theirs in many aspects. First of all, our model is fully dynamic. Not only all the latent distributions are always changing all the time, but new factors (including new influencers (friends), new interest areas and new behavior items) also emerge at any

time. Furthermore, we exploit the DPS functions in DP-Spaces to model the continuous variances, enabling us to see the long-term variance of social influence and personal preference over time and further tendency in the future.

7. CONCLUSION

Much effect has been devoted to learning and modeling the item-adoption behaviors. Traditional approaches usually model the behaviors using static or discrete approaches based on the assumption that all related factors are invariant all the time or at least in a period. In this paper, we address the fully-continuous temporal dynamics of social influence and personal preference for behaviors. Unlike existing work, we argue the social influence and personal preference are changing and evolving continuously all the time, and we believe understanding and further modeling their temporal dynamics would be greatly beneficial. We propose ConTyor, a new temporal dynamic behavior model, which enables us to learn how the social influence and personal preference are continuously varying with time and how the underlying time-variant factors contribute to people's dynamic behaviors. To depict such dynamics, we propose the DP-Space, a dynamic preference probability space which exploits the flexible DPP and DPS functions to express various shapes of dynamic changes with controllable complexity. A comprehensive experimental study demonstrates the outperformance of ConTyor in modeling and predicting individual behaviors over state-of-the-art approaches, and verifies the superiority of continuous temporal modeling over static or discrete ones.

8. ACKNOWLEDGMENTS

We thank Heran Lin and Xiang Ying for collecting the experimental datasets. This work is supported in part by the National Natural Science Foundation of China (No. 61373023, No. 61170064, No. 61133002) and the National High Technology Research and Development Program of China (No. 2012AA011002).

9. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005.
- [2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*, chapter 3, pages 138–147. Springer-Verlag New York, Inc., 2006.
- [3] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *ICML '06*, pages 113–120, 2006.
- [4] Y. Cai, H. fung Leung, Q. Li, H. Min, J. Tang, and J. Li. Typicality-based collaborative filtering recommendation. *IEEE Trans. on Knowl. and Data Eng.*, 26(3):766–779, March 2014.
- [5] W. Chen, W. Hsu, and M. L. Lee. Modeling user's receptiveness over time for recommendation. In *SIGIR '13*, pages 373–382, 2013.
- [6] F. Chua, H. Lauw, and E.-P. Lim. Generative models for item adoptions using social correlation. *IEEE Trans. on Knowl. and Data Eng.*, 25(9):2036–2048, 2013.
- [7] F. C. T. Chua, H. W. Lauw, and E.-P. Lim. Predicting item adoption using social correlation. In *SDM '11*, 2011.
- [8] T. Hofmann. Probabilistic latent semantic analysis. In *UAI '99*, pages 289–296, 1999.
- [9] M. Jamali and M. Ester. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *KDD '09*, pages 397–406, 2009.
- [10] B. Kanagal, A. Ahmed, S. Pandey, V. Josifovski, J. Yuan, and L. Garcia-Pueyo. Supercharging recommender systems using taxonomies for learning user purchase behavior. *Proc. VLDB Endow.*, 5(10):956–967, June 2012.
- [11] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR '09*, pages 195–202, 2009.

- [12] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD '09*, pages 447–456, 2009.
- [13] Q. Liu, E. Chen, H. Xiong, Y. Ge, Z. Li, and X. Wu. A cocktail approach for travel package recommendation. *IEEE Trans. on Knowl. and Data Eng.*, 26(2):278–293, Feb 2014.
- [14] H. Ma. An experimental study on implicit social recommendation. In *SIGIR '13*, pages 73–82, 2013.
- [15] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR '09*, pages 203–210, 2009.
- [16] A. Machanavajjhala, A. Korolova, and A. D. Sarma. Personalized social recommendations: Accurate or private. *Proc. VLDB Endow.*, 4(7):440–450, Apr. 2011.
- [17] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.
- [18] Y.-J. Park. The adaptive clustering method for the long tail problem of recommender systems. *IEEE Trans. on Knowl. and Data Eng.*, 25(8):1904–1915, Aug 2013.
- [19] Y. Shen and R. Jin. Learning personal + social latent factor model for social recommendation. In *KDD '12*, pages 1303–1311, 2012.
- [20] X. Su and T. M. Khoshgoftar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:1–19, Jan. 2009.
- [21] C. Wang, D. Blei, and D. Heckerman. Continuous Time Dynamic Topic Models. In *UAI '08*, 2008.
- [22] X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *KDD '06*, pages 424–433, 2006.
- [23] X. Wei, J. Sun, and X. Wang. Dynamic mixture models for multiple time series. In *IJCAI'07*, pages 2909–2914, 2007.
- [24] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: Joint friendship and interest propagation in social networks. In *WWW '11*, pages 537–546, 2011.
- [25] M. Ye, X. Liu, and W.-C. Lee. Exploring social influence for recommendation: A generative model approach. In *SIGIR '12*, pages 671–680, 2012.
- [26] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen. Challenging the long tail recommendation. *Proc. VLDB Endow.*, 5(9):896–907, May 2012.
- [27] J. Zhang, C. Wang, and J. Wang. Who Proposed the Relationship?: Recovering the Hidden Directions of Undirected Social Networks. In *WWW '14*, pages 807–818, 2014.
- [28] J. Zhang, C. Wang, J. Wang, and P. S. Yu. LaFT-Tree: Perceiving the Expansion Trace of One's Circle of Friends in Online Social Networks. In *WSDM '13*, pages 597–606, 2013.
- [29] J. Zhang, C. Wang, P. S. Yu, and J. Wang. Learning Latent Friendship Propagation Networks with Interest Awareness for Link Prediction. In *SIGIR '13*, pages 63–72, 2013.
- [30] T. Zhao, C. Li, M. Li, Q. Ding, and L. Li. Social recommendation incorporating topic mining and social trust analysis. In *CIKM '13*, pages 1643–1648, 2013.

APPENDIX

A. THE STUDIED BASIS FUNCTIONS

- The linear basis function:

$$\varphi_i(t; \phi_i) = \sigma_i \cdot t + \mu_i. \quad (23)$$

- The quadratic basis function:

$$\varphi_i(t; \phi_i) = \left(\frac{t - \mu_i}{\sigma_i} \right)^2. \quad (24)$$

- The polynomial basis function:

$$\varphi_i(t; \phi_i) = \left(\frac{t - \mu_i}{\sigma_i} \right)^i. \quad (25)$$

- The Gaussian basis function:

$$\varphi_i(t; \phi_i) = \exp \left\{ -\frac{(t - \mu_i)^2}{2\sigma_i^2} \right\}. \quad (26)$$

- The sigmoidal basis function:

$$\varphi_i(t; \phi_i) = \frac{1}{1 + \exp \left\{ -\frac{t - \mu_i}{\sigma_i} \right\}}. \quad (27)$$

- The exponential basis function:

$$\varphi_i(t; \phi_i) = \exp \left\{ \frac{t - \mu_i}{\sigma_i} \right\}. \quad (28)$$