Wang P, Zhang Y, Niu S *et al.* Modeling temporal dynamics of users' purchase behaviors for next basket prediction. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(6): 1230–1240 Nov. 2019. DOI 10.1007/s11390-019-1972-2

Modeling Temporal Dynamics of Users' Purchase Behaviors for Next Basket Prediction

Pengfei Wang[1], Yongfeng Zhang[2], Shuzi Niu[3], and Jiafeng Guo[4], *Member, CCF, ACM*

[1]*School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*

[2]*Department of Computer Science, Rutgers University, New Jersey 07450, U.S.A.*

[3]*Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*

[4]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

E-mail: wangpengfei@bupt.edu.cn; yongfeng.zhang@rutgers.edu; shuzi@iscas.ac.cn; guojiafeng@ict.ac.cn

Received November 30, 2018; revised September 10, 2019.

Abstract Next basket prediction attempts to provide sequential recommendations to users based on a sequence of the user's previous purchases. Ideally, a good prediction model should be able to explore the personalized preference of the users, as well as the sequential relations of the items. This goal of modeling becomes even more challenging when both factors are time-dependent. However, existing methods either take these two aspects as static, or only consider temporal dynamics for one of the two aspects. In this work, we propose the dynamic representation learning approach for time-dependent next basket recommendation, which jointly models the dynamic nature of user preferences and item relations. To do so, we explicitly model the transaction timestamps, as well as the dynamic representations of both users and items, so as to capture the personalized user preference on each individual item dynamically. Experiments on three real-world retail datasets show that our method significantly outperforms several state-of-the-art methods for next basket recommendation.

Keywords sequential recommendation, dynamic representation, next basket recommendation

1 Introduction

Next basket recommendation is becoming an increasingly important task for both retailers and customers, where many models have been devised. Among these models, sequential recommender[1–3] and time-aware collaborative filtering (CF)[4,5] are frequently integrated for this recommendation task. Sequential recommenders, e.g., the Markov approach[6], tend to explore the sequential relationships between items, and make next purchase prediction given a user's latest transaction, so as to capture the sequential patterns for recommendation, e.g., buying a phone may lead to accessory purchases. The time-aware collaborative filtering approach, on the other hand, tends to model the dynamic nature of a user's general preference by learning over his/her overall purchasing history for recommendation.

In real-world retail scenarios, however, both the user general interest and the item sequential relations could be time-dependent. For example, the general interest may shift as a consumer's age increases, while the sequential relationship between products may change over a season or year. As a result, it is essential to jointly consider the temporal dynamics of user preferences and item relations for informed recommendation.

However, existing approaches mostly consider the two types of dynamics separately. The sequential recommendation approach aims at modeling the item sequences from users, and attempts to extract sequential purchasing patterns between items for recommendation[6–10]; while the time-aware collaborative filtering approach integrates time factors into conventional CF models for dynamic user preference pre-

Regular Paper

This research work was supported by the National Natural Science Foundation of China under Grant Nos. 61802029, and 61602451, and the Fundamental Research for the Central Universities of China under Grant No. 500419741.

diction and recommendation[11–13]. In real-world scenarios, however, the dynamics of user preferences and item relations often depend on each other, and modeling each aspect respectively may not provide an accurate prediction of user preferences over items at the current time.

In this paper, we jointly model the dynamics of user preferences and item relationships, and cast the time-dependent next basket prediction task as a dynamic representation learning problem. To do so, we propose a dynamic representation model (DRM) for next basket recommendation. Specifically, DRM embeds a user or an item as a vector in a shared representation space. For each user and item, a time drifting function is used to model the dynamics of both user and the item representations on the feature level. A hybrid representation is then constructed by integrating both the user and item representations, and finally the hybrid representation is used to predict the items in the next basket. We conduct experiments over three real-world transaction datasets, and the empirical results verify the effectiveness of our approach compared with the state-of-the-art baselines.

In summary, the contributions of our work are as follows.

• We propose to consider the dynamic nature of user preferences and item sequential relationships jointly for next basket recommendation.

• We introduce a dynamic representation model for next basket prediction, and further design a time drifting function to model the temporal variations of both users and items on the feature level.

• Through empirical experiments we verify that our model can consistently outperform state-of-the-art baselines on the next basket recommendation.

The following part of the paper is organized as follows. We first discuss related work in Section 2, and then introduce our model in Section 3. We present the experiments in Section 4 as well as case studies in Section 5, and then make a conclusion with future research potentials in Section 6.

2 Related Work

In this section, we briefly review the two major approaches to next basket recommendation, i.e., the sequential recommendation approach, and the time-aware collaborative filtering approach.

2.1 Sequential Recommendation

The key idea of sequential recommender is to extract the sequential purchasing patterns between items from user transactions for next basket recommendation. Zimdars *et al.*[1] proposed a sequential recommendation model based on Markov chains, and studied how to extract sequential patterns to learn the next state using probabilistic decision tree models. Mobasher[3] studied different sequential patterns for recommendation, and found that contiguous sequential patterns are more effective for sequential prediction tasks than general sequential patterns. Yap *et al.*[14] introduced a new competence score measure for personalized sequential pattern mining and recommendation. Chen *et al.*[2] modeled playlists as a Markov chain, and proposed logic Markov embedding to learn the representations of songs for playlist prediction.

Recently, some hybrid methods have also utilized the sequential information to improve the recommendation performance. Rendle *et al.*[6] adopted tensor factorization to model user interest and sequential patterns, while Yin *et al.*[15] adopted topic modeling to capture user intrinsic interest and temporal contexts. With the development of representation learning techniques, Wang *et al.*[8] introduced a hierarchical representation model to consider the interactions between user general interests and sequential behaviors, while Yu *et al.*[10] adopted recurrent neural networks for sequential modeling and recommendation. [16] adopts a metric space learning approach to learn additive user-item relations for sequential recommendation, and the method is further generalized to factorization machines for sequential recommendation[17]. Recently, researchers have also explored memory networks[18] and knowledge graphs[19] for sequential recommendation.

2.2 Time-Aware Collaborative Filtering

Explicitly modeling the time factor has been shown to be beneficial for time-sensitive next basket recommendation. Existing work on this topic can be further categorized into time-aware general recommendation[4,12,20] and time-aware sequential recommendation[7,9,21].

Time-aware general recommendation mainly focuses on modeling the drifting nature of users' general preferences against time for recommendation. Koren[12] proposed TimeSvd++ for dynamic collaborative filtering by tracking the drifting user preferences and item biases across time bins. Lu *et al.*[22] proposed a spatio-temporal approach to collaborative filtering for dynamic recommendation. Xiong *et al.*[20] proposed a Bayesian Probabilistic Tensor Factorization algorithm to model the evolving relational data. Karatzoglou *et al.*[4] leveraged tensor factorization to model the dynamics of users' long-term interest against time, while Ahmed *et al.*[11] presented a time varying hierarchical user modeling approach that captures both the user's long-term and short-term interest.

Time-aware sequential recommendation conducts time series analysis to extract sequential patterns from different transactions. Based on the extracted sequential patterns, Wang *et al.*[7] modeled time context into recommendation by adding an exponential decay function on each sequential pattern; Wang and Zhang[9] adopted the opportunity model to capture the user's subsequent purchasing behaviors for recommendation; Zhang *et al.*[21] leveraged feature-level time series analysis to achieve daily-aware recommendation. However, these approaches require the presence of specific domain knowledge, which is usually expensive to obtain in practical systems.

3 Our Approach

In this section, we first formalize the next basket prediction problem when time factors are considered, and then describe our proposed DRM model. After that, we further present the model learning and recommendation procedures of DRM.

3.1 Problem Formalization

Let $I = \{i_1, i_2, \ldots, i_{|I|}\}$ be a set of items, and $|U|$ and $|I|$ be the total number of users and items, respectively. For each user $u \in U$, the purchase history B^u of user u is given by $B^u := (<B_1^u, t_1^u>, <B_2^u, t_2^u>, \ldots, <B_{b_u-1}^u, t_{b_u-1}^u>)$, where $B_m^u \subseteq I$, $m \in [1, b_u - 1]$ represents the m-th transaction, and t_m^u is the timestamp of transaction B_m^u.

Given the purchase history of a user u, our task is to learn a function F to recommend items that the user would most probably buy at the next (i.e., the b_u-th) timestamp $t_{b_u}^u$:

$$F : B^u, u, t_b^u \rightarrow B_{b_u}^u.$$

3.2 Dynamic Representation Model (DRM)

In this work, we learn a recommendation model that can integrate both the temporal dynamics of user preferences and item sequential relations. Fig.1 shows the architecture of DRM, where we learn both the dynamic user and item representations for next basket recommendation.

The model consists of three consecutive layers for embedding and recommendation: 1) the dynamic layer that learns the time-dependent user representations as well as item sequence representations from historical transactions; 2) the hybrid layer that aggregates user and item dynamic representations into a unified representation; 3) the output layer that summarizes the knowledge learned from data for prediction and recommendation. In the following, we present the design of each layer and the philosophy of such designs.

3.2.1 *Dynamic Layer*

Let $\boldsymbol{V}^U = \{\boldsymbol{v}_u^U \in \mathbb{R}^n | u \in U\}$ denote all the user vectors, and $\boldsymbol{V}^I = \{\boldsymbol{v}_i^I \in \mathbb{R}^n | i \in I\}$ denote all the item vectors. Based on the above definition, the dynamic representations of user $\boldsymbol{v}_u^U(\Delta t)$ and item $\boldsymbol{v}_i^I(\Delta t)$ over

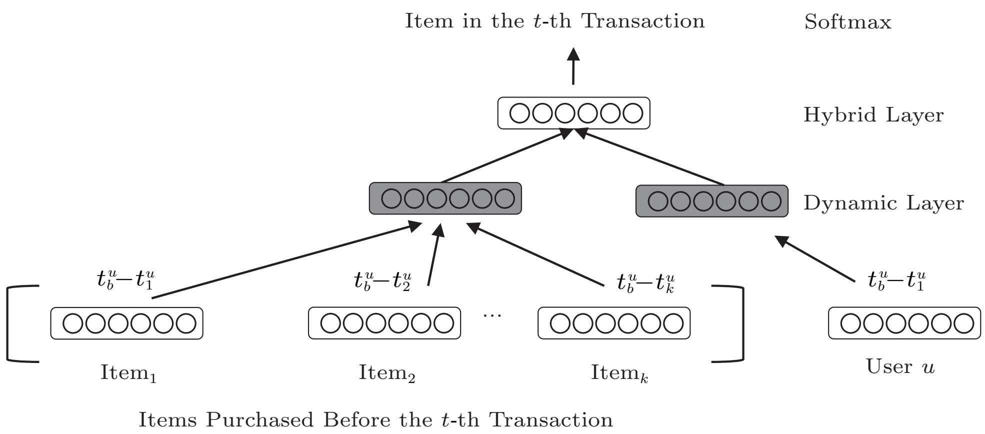


Fig.1. Architecture of DRM. User interests are aggregated from the user's dynamic preference and the dynamic sequential behaviors.

time are modeled as

$$\begin{cases} \boldsymbol{v}^U_{u,k}(\Delta t) = f(\Delta t, u, k)\boldsymbol{v}^U_{u,k} + \boldsymbol{s}^U_{u,k}, \\ \boldsymbol{v}^I_{i,k}(\Delta t) = f(\Delta t, i, k)\boldsymbol{v}^I_{i,k} + \boldsymbol{s}^I_{i,k}, \end{cases} \quad (1)$$

where $\boldsymbol{v}^U_{u,k}(\Delta t)$ is the k-th dimension of $\boldsymbol{v}^U_u(\Delta t)$ (same for $\boldsymbol{v}^I_{i,k}(\Delta t)$ as to $\boldsymbol{v}^I_i(\Delta t)$), $s^U_u \in \boldsymbol{S}^U$ and $s^I_i \in \boldsymbol{S}^I$ are the biases for users and items, respectively, and Δt represents the time interval of the purchase in consideration from the previous purchase action (e.g., three days).

Note that we model the dynamic representations of users and items as the aggregation of a static part and a time-variant part. Specifically, we take $\boldsymbol{S}^U$ and $\boldsymbol{S}^I$ to represent the static representations of users and items respectively, while in the time-variant part, $f(\cdot)$ is the time drifting function, which models the variations of different attributes against time. In this work, we adopt the Weibull function as the time drifting function:

$$f(\Delta t) = \gamma\theta\Delta t^{\gamma-1}e^{-\theta\Delta t^{\gamma}},$$

where θ is the scale parameter, and γ is the shape parameter. The Weibull function increases with time when $\gamma > 1$, decreases when $\gamma < 1$, and degenerates into an exponential model when $\gamma = 1$[9,21]. Due to this property, the Weibull function is very flexible to model the different variations of user and item attributes against time on the feature level. More specifically, we use $\varUpsilon^U = \{\boldsymbol{\gamma}^U_u \in \mathbb{R}^n | u \in U\}$ and $\varUpsilon^I = \{\boldsymbol{\gamma}^I_i \in \mathbb{R}^n | i \in I\}$ to represent the shape vectors of users and items, respectively and use $\boldsymbol{\varTheta}^U$ and $\boldsymbol{\varTheta}^I$ as the scale vectors for users and items, respectively. In (2), each dimension of $\boldsymbol{v}^U_u$ and $\boldsymbol{v}^I_i$ has their own shape and scale parameters. Based on the Weibull function, the drifting functions over time for each dimension are written as follows:

$$\begin{cases} f(\Delta t, u, k) = \boldsymbol{\gamma}^U_{u,k}\boldsymbol{\theta}^U_{u,k}\Delta t^{\boldsymbol{\gamma}^U_{u,k}-1}e^{-\boldsymbol{\theta}^U_{u,k}\Delta t^{\boldsymbol{\gamma}^U_{u,k}}}, \\ f(\Delta t, i, k) = \boldsymbol{\gamma}^I_{i,k}\boldsymbol{\theta}^I_{i,k}\Delta t^{\boldsymbol{\gamma}^I_{i,k}-1}e^{-\boldsymbol{\theta}^I_{i,k}\Delta t^{\boldsymbol{\gamma}^I_{i,k}}}. \end{cases} \quad (2)$$

In this way, we expect to model the different variations of each latent feature against time.

For simplicity, we use average pooling to obtain the item sequence representations in the dynamic layer:

$$\boldsymbol{v}^{\text{dyn}}_{\text{seq}} = \frac{1}{N(u)} \sum_{B^u_b \in B^u} \sum_{i \in B^u_b} \boldsymbol{v}^I_i(t^u_b - t^u_i),$$

where $N(u)$ is the number of items that user u purchased, and the dynamic representation of user is,

$$\boldsymbol{v}^{\text{dyn}}_u = \boldsymbol{v}^U_u(t^u_b - t^u_1).$$

3.2.2 Hybrid Layer

A hybrid representation $\boldsymbol{v}^{\text{Hybrid}}_u$ is then obtained from the aggregation of users' dynamic representation $\boldsymbol{v}^{\text{dyn}}_u$ and the dynamic item sequence representation $\boldsymbol{v}^{\text{dyn}}_{\text{seq}}$, so that DRM can benefit from both of the two dynamic natures:

$$\boldsymbol{v}^{\text{Hybrid}}_u = g(\boldsymbol{v}^{\text{dyn}}_u, \boldsymbol{v}^{\text{dyn}}_{\text{seq}}),$$

where $g(\cdot)$ denotes the aggregation operation. In this work, we take the commonly used average pooling operator to obtain the hybrid representation.

3.2.3 Output Layer

In the output layer, DRM computes the probability of buying the next item i via soft-max:

$$p(i \in B^u_b | u, B^u, t^u_b) = \frac{\exp(\boldsymbol{v}^I_i \cdot \boldsymbol{v}^{\text{Hybrid}}_u)}{\sum_{j=1}^{|I|} \exp(\boldsymbol{v}^I_j \cdot \boldsymbol{v}^{\text{Hybrid}}_u)}. \quad (3)$$

3.3 Model Learning and Recommendation

DRM maximizes the log probability defined in (3) over the whole transaction data of users for model learning:

$$\ell = \sum_{u \in U} \sum_{B^u_b \in B^u} \sum_{i \in B^u_b} \log p(i \in B^u_b | u, B^u, t^u_b) - \lambda \|\Phi\|^2_F,$$

where λ is the regularization coefficient and Φ is the model parameter. We adopt the negative sampling technique[23,24] to approximate the original objective function for training:

$$\ell_{NEG} = \sum_{u \in U} \sum_{B^u_b \in B^u} \sum_{i \in B^u_b} \Big(\log \sigma(\boldsymbol{v}^I_i \cdot \boldsymbol{v}^{\text{Hybrid}}_u) + k \times \mathbb{E}_{i' \sim P_I}[\log \sigma(-\boldsymbol{v}^I_{i'} \cdot \boldsymbol{v}^{\text{Hybrid}}_u)] \Big) - \lambda \|\Phi\|^2_F,$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, k is the number of negative samples, and i' is a sampled item, drawn according to the noise distribution P_I that is modeled by empirical unigram distribution over items.

Basically, the objective of DRM with negative sampling aims to derive the time-dependent ranking in a discriminative way by maximizing the probability of an observed item i and meanwhile minimizing the probability of an unobserved item i'.

With the learned dynamic user/item representations, given a user u, his/her historical transactions B^u, and a candidate item $i \in I$, we calculate the probability $p(i \in I | u, B^u, t^u_b)$ by DRM according to (3), and then recommend the top-n items to the user according to their probabilities.

4 Experimental Evaluation

4.1 Dataset Description

We evaluate different methods based on three real-world sequential transaction datasets.

• The Ta-Feng dataset is a public dataset released by RecSys conference, which covers a wide range of products from food, office supplies, to furniture. It contains 817 741 transactions from 32 266 users on 23 812 items.

• The BeiRen① dataset belongs to a large retail enterprise in China. We use the transactions of supermarket purchase history during the period from Jan. to Sept. in 2013.

• The Cell Phone dataset comprises a large corpora of reviews and timestamps related to phones and accessories. The dataset is from Amazon and spans from May 1996 to July 2014.

Similar to previous work[6,8], we conduct pre-processing on the three transaction datasets. For both Ta-Feng and BeiRen datasets, we remove those items bought by less than 10 users, and those users that have bought less than 10 items in total. Statistics of the three datasets after pre-processing are shown in Table 1. For each of the datasets, we reserve the last transaction (trans.) of each user for testing, the second last as the validating, and the rest transactions for training.

4.2 Baseline Methods

We evaluate our model by the comparison with both conventional but representative, and state-of-the-art methods for next basket recommendation.

• *TOP*: a non-personalized strategy that recommends the top-n most popular items to each user.

• *FPMC*: the Factorized Personalized Markov Chain model for sequential recommendation[6], which predicts the next purchase based on the latest transaction of the user with Markov assumptions.

• *NMF*: the Non-Negative Matrix Factorization method for collaborative filtering[25], where nonnegative MF is applied over the user-item binary interaction matrix constructed by discarding the sequential time information. We adopted the implementation in NMF:DTU Toolbox② for experiments.

• *HRM*: the state-of-the-art hierarchical representation model[8] for next basket recommendation, which adopts item-level static representation learning for user/item modeling. The implementation is publicly available③.

• *TimeSVD++*: one of the most successful models for dynamic user profiling[12], which considers the impact of time on users' general interest. This method achieved a success in the Netflix contest.

• *DREAM*: a hybrid model that learns a dynamic representation of a user but also captures global sequential features among baskets[10].

4.3 Evaluation Metrics

The performance is evaluated by predicting the last transaction $T^u_{t_u}$ for each user u in the testing dataset. For each recommendation method, we generate a list of n items ($n = 5$) for each user u, denoted by $R(u)$, where $R_i(u)$ stands for the item recommended on the i-th position. We use the following measures to evaluate the recommendation lists against the actually purchased items.

• F_1-*Score*: the harmonic mean of precision and recall.

• *HR*: the hit ratio, which is the percentage of recommendation lists that contain at least one correctly recommended item, i.e.,

$$HR = \frac{\sum_{u \in U} I(B^u_{t_u} \bigcap R(u) \neq \phi)}{|U|},$$

where $I(\cdot)$ is a binary indicator function, whose value is 1 when the condition is true, and 0 otherwise.

Table 1. Statistics of the Datasets Used in Our Experiments

Dataset	Number of Users $\lvert U\rvert$	Number of Items $\lvert I\rvert$	Number of Transactions $\lvert T\rvert$	Average Trans. Length	Average Time Interval (day)
Ta-Feng	9 238	7 982	67 964	7.4	16.9
BeiRen	9 321	5 845	91 294	9.7	9.8
Cell Phone	25 503	10 429	54 585	1.7	33.1

①http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Beiren, Oct. 2019.

②http://cogsys.imm.dtu.dk/toolbox/nmf/, Oct. 2019.

③http://www.bigdatalab.ac.cn/benchmark/bm/bd?code=HRM, Oct. 2019.

• *NDCG*: the normalized discounted cumulative gain that further takes into account the positions of correctly recommended items in the list[26], which is given by,

$$NDCG@n = \frac{1}{Z_n}\sum_{j=1}^{n}\frac{2^{I(R_j(u)\in B^u_{t_u})}-1}{\log_2(j+1)},$$

where $I(\cdot)$ is still the binary indicator function, and Z_n is a normalization constant that denotes the maximum possible value of NDCG given $R(u)$.

4.4 Parameter Settings

All the embedding vectors are randomly initialized in the range of (0, 1). For fair comparison, we select the best learning rate for each method in the range of {0.000 1, 0.001, 0.01, 0.05, 0.1}, and the vector dimension is tuned in the range of {50, 100, 150, 200}. We update them by conducting stochastic gradient descent (SGD). For HRM, we use max pooling strategy and set the drop rate to 0.5. For DRM, we set days as the basic unit to model the variations of users' interests and item properties, and shape parameters are initialized in the range of [0, 1.5].

4.5 Comparing Different Versions of DRM

We first compare the performances among different combinations of dynamic components. As can be seen in (1), both the user representations V^U and the item representations V^I can or cannot be modeled in a dynamic manner, which gives us four possible versions of DRM:

• $U+V$: a logic model, which assumes that both user and item representations are static;

• $U(t)+V$: a user-temporal model, which only considers the dynamics of user general interests;

• $U+V(t)$: an item-temporal model, which only considers the dynamics of item relations;

• $U(t)+V(t)$: both the temporal dynamics of users and items considered.

The results with different embedding sizes d are shown in Table 2–Table 4, where each best result (in bold) is significantly better than the second-best result on $p=1$ level, and d is the length of embedding vectors. Models 1–4 represent $U+V$, $U(t)+V$, $U+V(t)$, and $U(t)+V(t)$ respectively. We have the following observations. 1) The logic model ($U+V$) performs the worst, which verifies that considering the time dynamics for either users or items can improve the perfor-

Table 2. Performance (%) Comparison Among the Four Versions of Dynamic Representation Model on Ta-Feng Dataset

Model	$d=50$			$d=100$			$d=150$			$d=200$		
	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG
1	6.1	28.2	7.6	6.4	29.3	7.8	6.5	29.8	8.0	6.8	31.2	8.1
2	6.3	28.3	8.0	6.7	29.8	8.2	6.9	30.9	8.3	7.1	31.5	8.4
3	6.5	29.2	8.2	6.7	30.5	8.5	7.1	31.7	8.5	7.3	31.8	8.5
4	**6.7**	**29.4**	**8.6**	**6.9**	**30.7**	**8.7**	**7.2**	**31.8**	**8.8**	**7.5**	**32.3**	**9.1**

Table 3. Performance (%) Comparison Among the Four Versions of Dynamic Representation Model on BeiRen Dataset

Model	$d=50$			$d=100$			$d=150$			$d=200$		
	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG
1	11.1	50.1	14.8	11.5	50.7	15.3	11.7	51.5	15.4	11.8	51.6	15.6
2	11.3	50.6	15.4	11.6	51.3	15.5	11.8	51.2	15.6	11.9	51.9	16.0
3	11.6	50.9	15.9	11.8	51.8	16.0	12.0	51.6	16.1	12.0	51.9	16.4
4	**11.7**	**51.1**	**16.1**	**12.2**	**51.8**	**16.4**	**12.3**	**52.6**	**16.5**	**12.5**	**52.7**	**16.6**

Table 4. Performance (%) Comparison Among the Four Versions of Dynamic Representation Model on CellPhone Dataset

Model	$d=50$			$d=100$			$d=150$			$d=200$		
	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG	F_1-Score	Hit Ratio	NDCG
1	2.96	20.1	10.5	3.03	20.4	10.8	3.13	20.5	11.4	3.26	22.1	11.5
2	3.03	20.6	10.7	3.10	20.8	11.1	3.22	21.3	11.6	3.32	22.4	11.9
3	3.06	20.9	10.6	3.08	21.0	11.0	3.25	21.6	11.7	3.28	22.6	11.8
4	**4.09**	**21.2**	**11.6**	**4.12**	**21.8**	**11.9**	**4.14**	**22.6**	**12.1**	**4.28**	**23.2**	**12.6**

mance of recommendation. 2) When considering user or item dynamic representations, the performance on all of the three datasets improved; however, results show that $U+V(t)$ performs better than $U(t)+V$. The potential reason can be that the shifting of user preferences may span over a long period of time (e.g., across several years with age), which may not be well captured with several months of transactions as in the data, while the dynamics of item relations can be easily captured by frequent co-purchases from different users. 3) When considering both the dynamics of users and items, DRM achieves the best performance.

4.6 Comparison with Baselines

We further compare our DRM approach with other recommendation methods. Here we adopt the full dynamic model (i.e., $U(t)+V(t)$) for comparison. The results for F_1-score, HR, NDCG on all datasets under different choices of embedding sizes or the number of latent factors are shown in Fig.2.

We see the followings. 1) The non-personalized TOP method did not perform well on most datasets and measures. 2) Based on collaborative filtering, NMF performs better than TOP with personalized recommendations. 3) By modeling the dynamic user prefer-

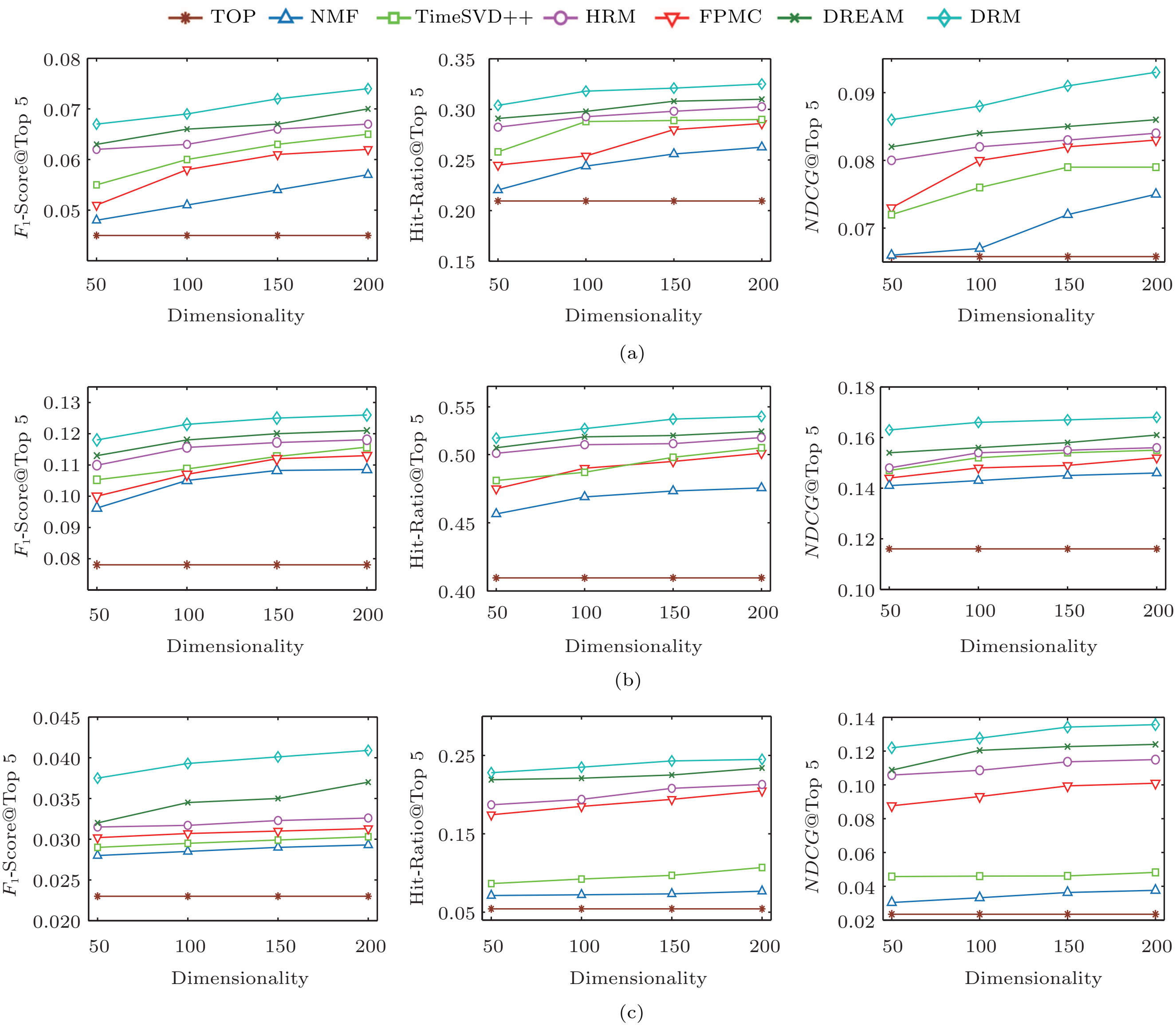


Fig.2. Performance comparison of DRM with TOP, FPMC, NMF, HRM, TimeSVD++, and DREAM over the three datasets. The embedding size (or the number of latent factors) increases from 50 to 200 with a step of 50 on all the datasets. (a) Ta-Feng. (b) BeiRen. (c) Cell Phone.

ences (TimeSVD++), or the item sequential relationships (FPMC), both TimeSVD++ and FPMC achieve better results than static NMF recommender on all cases, which implies the importance of dynamic modeling for next basket recommendation. Beyond this, further analyses will also be provided in Subsection 4.7. 4) By integrating both (static) user preferences and sequential patterns, HRM achieves better results than TimeSVD++ and FPMC. 5) By further capturing the temporal dynamics of both users and items, DRM outperforms all the baselines in terms of all the evaluation measures over the three datasets. Taking the Ta-Feng dataset with the embedding size of 50 as an example, when compared with the best baseline method (i.e., DREAM), our DRM approach achieves around 1.8% and 1.1% improvements in terms of hit ratio and NDCG, respectively. The improvements are statistically significant at $p = 0.01$.

4.7 Comparison Among Different User Groups

To further analyze the performance of different methods, we split the users into three groups — inactive, medium, and active — based on the time interval between the latest three transactions of a user. Taking the Ta-Feng dataset as an example, a user is empirically treated as active if the time interval is less than seven days, and inactive if the time is longer than 14 days. The remaining users are classified into the medium group. In this way, the proportions of active, medium, and inactive users are 29.1%, 14.9%, and 56.0%, respectively. Note that this is just an intuition study and the thresholds are selected empirically according to practical experience. We present the comparison results on the Ta-Feng dataset under the embedding size $d = 50$ (in Table 3), and the results on the BeiRen dataset as well as on other embedding size choices are similar.

From the results we can get the following findings: 1) The non-personalized TOP method still did not perform well on most cases, including active, medium, and inactive users. 2) By combining both sequential patterns and user general interest, FPMC obtains better performance than NMF, which is quite consistent with the observation in previous work[6,8]. 3) By modeling interactions among user general interest and item sequential patterns, HRM obtains better performance than FPMC on all three user groups, and performs better than NMF on inactive and medium users, which indicates its advantage in working on users with sparse records. 4) Finally, by considering the dynamic representations of both users and items, DRM obtains the best performance in terms of all the evaluation metrics, and benefits all the user groups, especially the active user group. Compared with the second-best model HRM, the relative performance improvements of DRM on the active users are 1.0%, 3.6%, and 1.1% in terms of F_1-score, hit ratio, and NDCG, respectively. The improvement is statistically significant ($p < 0.01$). It verifies that by further considering temporal dynamics of both users and items, DRM can improve the recommendation performance significantly.

Table 3. Performance Comparison on Ta-Feng Dataset over Different User Groups with Embedding Size $d = 50$

User Activeness	Method	F_1-Score	Hit Ratio	NDCG
Active	TOP	0.042	0.189	0.082
	FPMC	0.052	0.217	0.098
	NMF	0.049	0.211	0.093
	TimeSVD++	0.055	0.230	0.105
	HRM	0.056	0.236	0.115
	DREAM	0.058	0.246	0.118
	DRM	0.066	0.272	0.128
Medium	TOP	0.052	0.240	0.097
	FPMC	0.055	0.267	0.108
	NMF	0.048	0.235	0.087
	TimeSVD++	0.059	0.263	0.110
	HRM	0.065	0.299	0.115
	DREAM	0.066	0.305	0.120
	DRM	0.075	0.319	0.126
Inactive	TOP	0.044	0.201	0.084
	FPMC	0.050	0.242	0.102
	NMF	0.048	0.221	0.086
	TimeSVD++	0.054	0.234	0.098
	HRM	0.059	0.265	0.105
	DREAM	0.060	0.270	0.111
	DRM	0.065	0.288	0.122

5 Case Studies

To gain a deeper understanding of the temporal dynamics of users and items, we conduct case studies for further analysis of our model in this section.

5.1 Time Series Analysis of Single Item

We first analyze the item dynamics over time. To do so, we take the BeiRen dataset as an example and set the embedding size to 50. We select four items from the dataset for case study, which are biscuit, tissue, shopping bag, and milk. We then compare their shape parameter γ to analyze the variation of their dynamic parts in (1), and the details are shown in Fig.3.

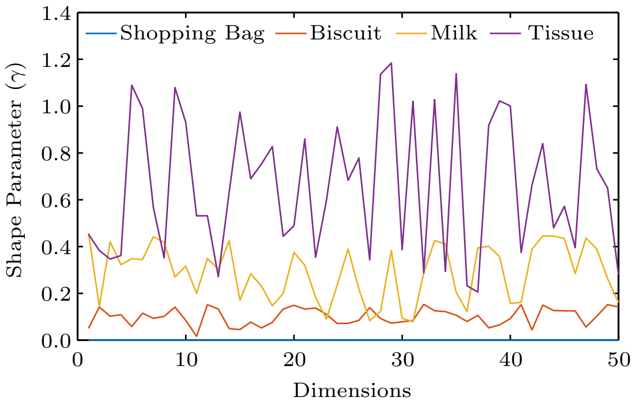
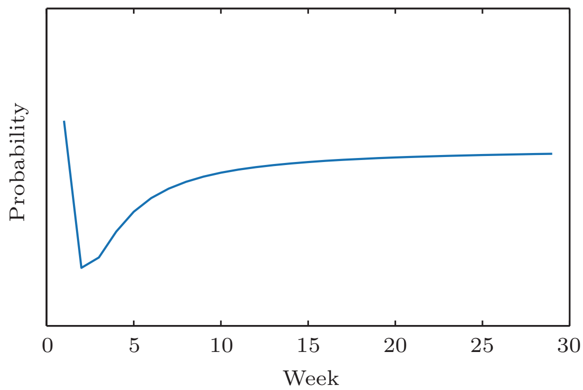


Fig.3. Visualization of the shape parameter for case-study items selected from the BeiRen dataset when embedding size $d = 50$. The y-axis represents the shape parameter γ, and the x-axis indicates each of the 50 embedding dimensions.

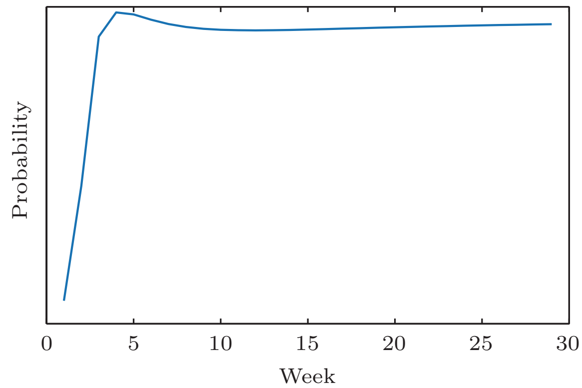
From the results we have the following observations. 1) For each item, the dynamic variations of different dimensions are very different — the value of the drifting function on some dimensions increases over time ($\gamma > 1$), while on some other dimensions the value decreases sharply over time ($\gamma < 1$). This observation is consistent with our assumption that the dynamic variation of different features may be different even for the same product. 2) The impact of dynamic part varies for different items — taking tissue and shopping bag as examples, we see that the shape parameters of tissue are greater than 1 on a lot of dimensions, making the value of the drifting function increase over time, which means that the representation (and thus the consumption) of tissue products may be quite dynamic over time. In contrast with tissue, the shape parameter of shopping bag is extremely small on most dimensions, and thus the drifting function value is close to 0, indicating that the consumption of shopping bags is relatively static over time, which is consistent with our intuitions in practice.

5.2 Time Series Analysis of Sequential Behaviors

In this subsection we analyze the item sequential relationships over time. Specifically, we choose two sequential patterns from the BeiRen dataset, which are $\langle toothpaste, toothpaste\rangle$ and $\langle toothpaste, toothbrush\rangle$, respectively. Given that a user has purchased a toothpaste, we analyze the probability that the user will purchase a toothpaste or toothbrush over time through (3) when the embedding size is 50, and the results are shown in Fig.4.



(a)



(b)

Fig.4. Probability distributions of (a) $\langle$*toothpaste*, *toothpaste*$\rangle$ and (b) $\langle$*toothpaste*, *toothbrush*$\rangle$ item relationship patterns over time.

We can see that, after the user has purchased a toothpaste, the probability of buying another toothpaste decreases over time at first, reaching a minimum after around two weeks, and then begins to increase over time, and after about 10 weeks the probability begins to converge. On the other hand, the probability of buying a toothbrush after a toothpaste increases over time, and reaches a maximum after about five weeks. This is consistent with our intuitions in daily life, and this phenomenon is also frequently referred to as the Law of Diminishing Marginal Utility[27] by economists.

5.3 Recommending Different Items over Time

We further check if our model can properly recommend time-dependent items over time, even for the same user. We still take the BeiRen dataset and set the embedding size to 50. Given that a user has purchased a toothbrush and biscuit, we provide top-3 item recommendation by DRM and DREAM for each of following weeks, as shown in Table 5.

Table 5. Top-3 Recommended Items of DREAM and DRM on Each Week After the User Has Purchased a Toothbrush and Biscuit

Week	DREAM	DRM
1	Toothpaste, toothbrush, soap	Fruit, fish, **biscuit**
2	Toothpaste, toothbrush, soap	Fruit, **cornmeal**, toothpaste
3	Toothpaste, **toothbrush**, soap	**Toothbrush**, biscuit, bread

Note: Words written in bold represent the items predicted correctly.

It is intuitive to see that DRM recommends the same user with different products on different time when an initial purchase has been made by the user. Besides, the recommended items not only are time-sensitive, but also accompany the already purchased items by mining the wisdom embedded in large-scale user purchasing transactions. In addition, by considering the time influence on users' purchase behaviors, DRM predicts that the user will buy the biscuit one week later, and the toothbrush three weeks later correctly, while DREAM can only recommend the same items against time as DREAM fails to consider the variations of users' interest and item properties.

6 Conclusions

In this paper, we proposed to model the dynamic nature of both user preferences and item sequential relations at the same time. To do so, we proposed the dynamic representation model (DRM) for next basket recommendation. Both quantitive experiments and empirical studies were conducted with three real-world datasets, which verified both the effectiveness of our model and the underlying intuition of our approaches.

This is our first step towards introducing the idea of dynamic representation learning for recommendation, and there is much room for further improvements. In the future, we would like to model the cyclical nature of user purchase behaviors as well as user repeating purchases for recommendation. The basic ideas of dynamic representation learning can also be applied to other tasks beyond next basket recommendation, e.g., dynamic explainable recommendation, group recommendation, and even personalized search tasks.

References

[1] Zimdars A, Chickering D M, Meek C. Using temporal data for making recommendations. In *Proc. the 17th Conference on Uncertainty in Artificial Intelligence*, August 2001, pp.580-588.

[2] Chen S, Moore J L, Turnbull D, Joachims T. Playlist prediction via metric embedding. In *Proc. the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2012, pp.714-722.

[3] Mobasher B, Dai H H, Luo T, Nakagawa M. Using sequential and non-sequential patterns in predictive web usage mining tasks. In *Proc. the 2002 IEEE International Conference on Data Mining*, December 2002, pp.669-672.

[4] Karatzoglou A, Amatriain X, Baltrunas L, Oliver N. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proc. the 4th ACM Conference on Recommender Systems*, September 2012, pp.79-86.

[5] Su X Y, Khoshgoftaar T M. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009, 2009: Article No. 421425.

[6] Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next basket recommendation. In *Proc. the 19th International Conference on World Wide Web*, April 2010, pp.811-820.

[7] Wang P F, Guo J F, Lan Y Y. Modeling retail transaction data for personalized shopping recommendation. In *Proc. the 23rd ACM International Conference on Information and Knowledge Management*, November 2014, pp.1979-1982.

[8] Wang P F, Guo J F, Lan Y Y, Xu J, Wan S X, Cheng X Q. Learning hierarchical representation model for next basket recommendation. In *Proc. the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2015, pp.403-412.

[9] Wang J, Zhang Y. Opportunity model for e-commerce recommendation: Right product; right time. In *Proc. the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2013, pp.303-312.

[10] Yu F, Liu Q, Wu S, Wang L, Tan T N. A dynamic recurrent model for next basket recommendation. In *Proc. the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2016, pp.729-732.

[11] Ahmed A, Low Y C, Aly M, Josifovski V, Smola A J. Scalable distributed inference of dynamic user interests for behavioral targeting. In *Proc. the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2010, pp.114-122.

[12] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proc. the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2008, pp.426-434.

[13] Xiang L, Yuan Q, Zhao S W, Chen L, Zhang X T, Yang Q, Sun J M. Temporal recommendation on graphs via long- and short-term preference fusion. In *Proc. the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 2010, pp.723-732.

[14] Yap G E, Li X L, Yu P S. Effective next-items recommendation via personalized sequential pattern mining. In *Proc. the 17th International Conference on Database Systems for Advanced Applications*, April 2012, pp.48-64.

[15] Yin H Z, Cui B, Chen L, Hu Z T, Huang Z. A temporal context-aware model for user behavior modeling in social media systems. In *Proc. the 2014 ACM SIGMOD International Conference on Management of Data*, June 2014, pp.1543-1554.

[16] He R N, Kang W C, McAuley J J. Translation-based recommendation. In *Proc. the 11th ACM Conference on Recommender Systems*, August 2017, pp.161-169.

[17] Pasricha R, McAuley J J. Translation-based factorization machines for sequential recommendation. In *Proc. the 12th ACM Conference on Recommender Systems*, October 2018, pp.63-71.

[18] Chen X, Xu H T, Zhang Y F, Tang J X, Cao Y X, Q Z, Zha H Y. Sequential recommendation with user memory networks. In *Proc. the 11th ACM International Conference on Web Search and Data Mining*, February 2018, pp.108-116.

[19] Huang J, Zhao W X, Dou H J, Wen J R, Chang E Y. Improving sequential Recommendation with knowledge-enhanced memory networks. In *Proc. the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, July 2018, pp.505-514.

[20] Xiong L, Chen X, Huang T K *et al*. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *Proc. the 2010 SIAM International Conference on Data Mining*, April 2010, pp.211-222.

[21] Zhang Y F, Zhang M, Zhang Y, Lai G K, Liu Y Q, Zhang H H, Ma S P. Daily-aware personalized recommendation based on feature-level time series analysis. In *Proc. the 24th International Conference on World Wide Web*, May 2015, pp.1373-1383.

[22] Lu Z D, Agarwal D, Dhillon I S. A spatio-temporal approach to collaborative filtering. In *Proc. the 3rd ACM Conference on Recommender Systems*, October 2009, pp.13-20.

[23] Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In *Proc. the 27th Annual Conference on Neural Information Processing Systems*, December 2013, pp.3111-3119.

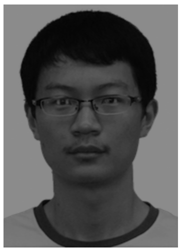
[24] Le Q V, Tomas M. Distributed representations of sentences and documents. In *Proc. the 31st International Conference on Machine Learning*, June 2014, pp.1188-1196.

[25] Lee D D, Seung H S. Algorithms for non-negative matrix factorization. In *Proc. the 14th Conference on Neural Information Processing Systems*, December 2001, pp.556-562.

[26] Järvelin K, Kekäläinen J. IR evaluation methods for retrieving highly relevant documents. In *Proc. the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2000, pp.41-48.

[27] Zhao G, Lee M L, Hsu W, Chen W. Increasing temporal diversity with purchase intervals. In *Proc. the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2012, pp.165-174.

Pengfei Wang received his B.S. degree in software engineering from the Xidian University, Xi'an, in 2008, his M.S. degree in software engineering from the Beihang University, Beijing, in 2011, and his Ph.D. degree in computer software theory from the Institute of Computing Technology, Chinese Academic of Sciences, Beijing, in 2017. He joined the Beijing University of Posts and Telecommunications, Beijing, as a lecturer in 2017. His research interests include data mining, machine learning, and recommendation. He has published more than 10 papers in refereed journals and conferences.



Yongfeng Zhang is an assistant professor in the Department of Computer Science, Rutgers University, New Jersey. His main research interests include recommender systems, information retrieval, machine learning, and data science.

Shuzi Niu is an associate researcher in the Institute of Software, Chinese Academic of Sciences, Beijing. Her research interests include data mining, machine learning, information retrieval and recommendation.

Jiafeng Guo is a researcher in the Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS), Beijing. His major research interests include Web search and data mining, user data modeling, machine learning, and social search. He has published more than 50 papers in international journals and conferences, including IEEE TKDE, Physical Review E, Journal of Statistical Mechanics, SIGIR, WWW, CIKM, ICDM, WSDM, and IJCAI. He has won the Best Paper Award in CIKM (2011) and the Best Student Paper Award in SIGIR (2012). He currently serves on the program committees of several international conferences and journals, including SIGIR, WWW, CIKM and TOIS. He was the recipient of the "Excellent Ph.D. Dissertation Award of CAS" in 2009.