Product Adoption Rate Prediction in a Competitive Market

Le Wu, Qi Liu, *Member, IEEE*, Richang Hong, *Member, IEEE*, Enhong Chen, *Senior Member, IEEE*, Yong Ge, Xing Xie, *Senior Member, IEEE*, and Meng Wang, *Senior Member, IEEE*

Abstract—As the worlds of commerce and the Internet technology become more inextricably linked, a large number of user consumption series become available for online market intelligence analysis. A critical demand along this line is to predict the future product adoption state of each user, which enables a wide range of applications such as targeted marketing. Nevertheless, previous works only aimed at predicting if a user would adopt a particular product or not with a binary buy-or-not representation. The problem of tracking and predicting users' adoption rates, i.e., the frequency and regularity of using each product over time, is still under-explored. To this end, we present a comprehensive study of product adoption rate prediction in a competitive market. This task is nontrivial as there are three major challenges in modeling users' complex adoption states: the heterogeneous data sources around users, the unique user preference and the competitive product selection. To deal with these challenges, we first introduce a flexible factor-based decision function to capture the change of users' product adoption rate over time, where various factors that may influence users' decisions from heterogeneous data sources can be leveraged. Using this factor-based decision function, we then provide two corresponding models to learn the parameters of the decision function with both generalized and personalized assumptions of users' preferences. We further study how to leverage the competition among different products and simultaneously learn product competition and users' preferences with both generalized and personalized assumptions. Finally, extensive experiments on two real-world datasets show the superiority of our proposed models.

Index Terms—User modeling, product adoption, user interest modeling, product competition

1 Introduction

With the help of information technology, users' digital footprints over a long time period have been easily collected by various online service providers, such as blogs, forums, and social-networking services. Such contents have accumulated an increasing interest in data-driven business intelligence research, whose goal is to collect and analyze users' behavior, and then provide insightful guidance to facilitate business management and strategizing [9], [17]. Particularly, the problem of predicting users' product adoption probability has been one of the emerging fields in this area. Accurately predicting users' product adoption tendency is beneficial for a broad range for applications, such as targeted marketing and marketing strategy development for product providers [4], as well as personalized services for customers [2], [29].

In the literature, much of the active research has been devoted to the product adoption prediction problem [15]. Specifically, these works usually classified users into two categories: the adopters that already consumed this product and the non-adopters that have not consumed it till now. In other words, these methods described users' product adoption states with a binary buy-or-not representation. Then some learning algorithms are proposed to model the future adoption possibilities of those non-adopters. E.g., the popular recommender systems deal with the task of predicting users' preferences to the products that they have not consumed before [2], [40]. In contrast to these products that are usually consumed only once (e.g., books and movies), there are plenty of products users may use frequently after buying them, such as smart devices. Fig. 1 shows an illustrating example of users' preferences to two different smart devices over time. As shown in this figure, the traditional buy-or-not binary-valued adoption representation only captures the fact that both users have consumed the two smart devices in the past. Actually, in a specific competitive market (e.g., *mobile devices*), it is nature for a user to switch among different products over time after she consumes these products (e.g., *iPhone*, *Samsung*, *and Windows*). Compared to the traditional static buy-or-not adoption representation, the merchants care more about users' loyalty and commitment to the products over time after users consume the products. To better capture users' loyalty to the frequently used products after puchase over time, we argue, the measure of *adoption rate*, i.e., the usage rate and regularity that consumers use a product at a particular time, is more appropriate to describe users' preference changes to different products. As each

- *L. Wu, R. Hong, and M. Wang are with the School of Computer and Information, Hefei University of Technology, Hefei, Anhui 230009, China. E-mail: {lewu.ustc, hongrc.hfut, eric.mengwang}@gmail.com.*
- *Q. Liu and E. Chen are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: {qiliuql, cheneh}@ustc.edu.cn.*
- *Y. Ge is with the Department of Management Information Systems, University of Arizona, Tucson, AZ 85721. E-mail: yongge@email.arizona.edu.*
- *X. Xie is with Microsoft Research, Beijing 100080, China. E-mail: Xing.Xie@microsoft.com.*

Manuscript received 28 Mar. 2017; revised 24 Sept. 2017; accepted 4 Oct. 2017. Date of publication 17 Oct. 2017; date of current version 9 Jan. 2018. (Corresponding author: Qi Liu.)
Recommended for acceptance by P. Cui.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2763944

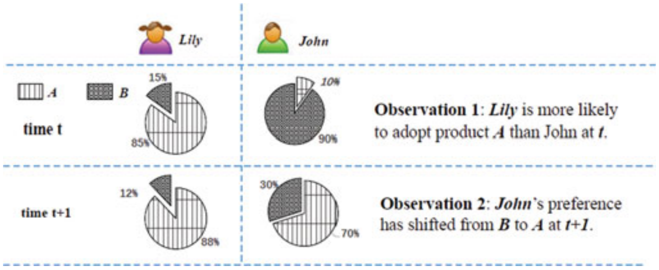


Fig. 1. An example of the adoption frequency of two users in a smart device market at time t and $t+1$. For ease of illustration, suppose both users have two smart devices of *A* and *B*. The percentage shows the frequency of adopting this product after purchase, i.e., the usage frequency of this product. In this figure, the adoption rate of *Lily* to *A* is 0.85 at t. Nevertheless, traditional methods would assign a static value 1 to both users at these two time slices as they purchased the two smart devices while neglecting their detailed preference changes over time after purchase.

user's adoption rate over time could be summarized into an adoption series, by capturing each user's adoption rate series, the two observations in Fig. 1 can be easily obtained.

After introducing the product adoption rate measure, the problem we study in this paper is how to predict the future product adoption rate of each user in a competitive market. Unfortunately, none of the existing models (e.g., models for recommender system [25], the time-series forecasting models [3], [37]) could be directly applied to this problem due to the following challenges. First, a user's decision making process is very complex as many heterogeneous sources around her may contribute to the final decision, e.g., the users' own profiles [45], and the social network structure [15]. How to design a flexible prediction model that can leverage many heterogeneous data sources in a unified framework remains pretty much open. Second, based on the heterogeneous data sources around users, the adoption decision process varies from person to person. For example, some users may weight more on social neighbors' opinions while others are unlikely to change their decisions. Thus, from a limited adoption rate series of each user, how to explore users' unique preferences becomes another challenge. Last but not least, in a competitive market, the fierce competition among different products is a significant factor to track the transitions of users' adoption rates over time. In fact, in the marketing domain, product competition is well recognized as a focal part that influences a company's market performance [12], [14]. How to mine the competitive relationships among products in a competitive market to improve the product adoption rate prediction results? *In summary, the data heterogeneity, the unique user preference and product competition compose three main challenges of the problem we study.*

To address the challenges of data heterogeneity and user uniqueness, we provided a preliminary study on the product adoption rate prediction problem from a multi-factor view [42]. Specifically, we first introduced a flexible factor-based decision function to capture users' product adoption rate changes over time, where various factors from heterogeneous data sources that may influence users' decisions can be leveraged. Using this factor-based decision function, we then provided a Generalized Adoption Model (*GAM*) and a *B*ayesian *P*ersonalized *A*doption *M*odel (*BPAM*) to learn the parameters of the decision function with both generalized and personalized assumptions of users' preferences.

In this paper, we further extend our previous work and study the product adoption rate prediction problem with multiple products in a competitive market. A naive method is to divide the multi-product adoption rate prediction problem into a set of independent single product prediction problems, then our previous proposed GAM and BPAM models could be applied directly [42]. This independent assumption among products enjoys the advantage of simplicity, however, it fails to consider the competition among products in real-world adoption decisions. In a competitive market, products turn to compete with each other to attract the attention of users. Take the competition among smart devices as an example, as shown in Fig. 1, since *John* turns to adopt product A more frequently at $t+1$, the adoption rate of B decreases at that time. Therefore, we argue, in order to predict users' adoption rate more accurately, it is essential to take the competition effect among products into consideration. Specifically, given a competitive market with multiple products compete with each other, we study how to incorporate product competition into the proposed GAM and BPAM models, and jointly learn user preference and product competition in a unified framework. The extended models are termed as GAM with Competition (*GAM-C*) and BPAM with Competition (*BPAM-C*) respectively. We argue that the joint modeling of users' preferences and product competition is significant as users interact with multiple products at the same time, and product competition is considered as an indispensable part for users to transit commitment to different products.

In summary, by extending the problem definition from predicting the adoption rate of a particular product [42] to multiple products in a competitive market, we further address the technical challenge of how to model product competition as a factor in the decision function, and how to jointly learn the parameters of users' preferences and product competition in our proposed models of GAM-C and BPAM-C (Section 6). Finally, we conduct extensive experiments on two markets: a smartphone device market and a internet access technology market. The experimental results on these two real-world datasets show the effectiveness of our proposed models.

2 Related Work

We summarize the related work as follows.

Recommender Systems. At a first glance, our research topic resembles the recommender systems. Recommender systems infer each user's preferences to products that she has not rated before, and then recommend those products that have the largest predicted ratings [18], [18], [19], [19], [35]. The models in this area can be As users' preferences evolve over time, time-aware recommender systems exploited how to leverage users' temporal dynamics to further improve recommendation accuracy [22], [24]. All these recommendation systems saliently assumed that users would adopt the products once (e.g., movies and travel attractions), thus they focused on predicting the preferences of users to products that have not been adopted yet [2], [29], [43]. We borrow the idea of modeling users' unique preferences to improve product adoption rate prediction performance. Nevertheless, instead of discovering products users are unfamiliar with, there are plenty of products that users may buy or use frequently. Our proposed problem is complementary to the recommender systems as we put emphasis on predicting the future likelihood of adopting the products users have already adopted in the past.

Product Adoption Prediction in Social Networks. With the proliferation of online social networks, a hot research topic is how to leverage the social network for better product adoption prediction performance. A distinct characteristic of the social network is the existence of the social influence, which usually presents in two forms: the global crowd influence shows the herding effect among the population level while the local social neighbors influence argues that users are more likely to be influenced by the social neighbors' decisions than others [5], [21]. Thus, researchers focused on how to promote the product adoption prediction performance for non-adopters in social platforms with the above theories. The main algorithms could be summarized into the following two categories: the feature-based methods and the social recommendation models. The feature-based methods designed features from users' social profiles [45], the social network [15], [23] or the hybrid of the above two [6] for prediction task. On the other hand, the social recommendation models leveraged the social network information into recommendation algorithms [10], [32], [41]. Nearly all these works classified users into two categories: the adopters that have already been familiar with the product and the non-adopters that have not adopted the product yet, thus they represented users' adoptions with static binary buy-or-not values. Nevertheless, there are many products that users may frequently adopt and their adoption states evolve over time. To better characterize users' adoption dynamics, we measure each user's adoption state for each product as a rate (i.e., in the range of [0,1]) that changes over time.

Time-Series Forecasting. Our proposed problem is also closely related to time-series forecasting, which builds models to predict future values based on previously observed sequences of discrete-time data. Among them, the auto regression (AR) model described the process of a single time series, where the current value linearly depends on its previous values [3]. The vector auto regression (VAR) model generalized AR and captured the linear interdependencies among multiple time series. Some superior models, such as the hidden Markov model [37], the conditional random fields [26], and the linear dynamic systems [30] are statistical Markov models in which the system being modeled is assumed to be a Markov process with unobserved states. We borrow the ideas of temporal dependency modeling of these models. However, these methods could not work well under our scenario. The proposed adoption rate forecasting task displays specific challenges of data heterogeneity, preference uniqueness as well as product competition in a market, which restrains the utility of techniques in time-series forecasting.

Product Competition Mining. As competition analysis serves as a pivotal role in companies for strategy formulation, monitoring, and adjustment, researchers from the marketing and management community have long studied this problem from an empirical view [12], [14]. With the availability of large online data, researchers have motivated to design data-driven competition intelligence analysis [4], [27]. Another research line focused on the competitive relationship identification by assuming that competitors usually co-occur in the web data sources, such as online reviews [34], online news [33] and search log analysis [4]. Recently, Zhang et al. designed a topic model to monitor market competition by jointly modeling online text and image data [44]. To summarize, nearly all these previous works focused on mining product competition relationship by leveraging the text-based knowledge. Different from these text-based competition relationship mining, we focus on learning the competitive degree between different products directly from users' behaviors.

3 Concepts and Problem Formulation

In this section, we first introduce some basic concepts, followed by the formulation of the product adoption rate prediction problem.

3.1 Product Adoption Rate Measure

Let us consider a competitive market with a set of N users $U = [u, v, \ldots]$ and a set of M products $B = [a, b, c, \ldots]$. Users form a social network $G = \langle U, A\rangle$, with U denotes the same set of users in the market and the edge set A represents the relationship between users. E.g., if user u follows user v, then $(u, v) \in A$. The product set B is application dependent that contains all the products that compete with each other in the competitive market, which can be obtained by the merchants or the domain experts. E.g., in a smart device market, the products include *iPhone*, *Windows*, *Androiod* and so on. Here, to track users' preferences and loyalty to products in a competitive market, we introduce a *product adoption rate* notion to measure the frequency and regularity of users' preferences to products at each time, which is defined as:

Definition 1 (Product Adoption Rate). *The adoption rate of user u to product b at time t, denoted as r^t_{ub}, is defined as the percentage or the normalized frequency of using b (e.g., usage times) among the whole product set B at t: $r^t_{ub} = \frac{c^t_{ub}}{\sum_{a\in B} c^t_{ua}}$, where c^t_{ub} records the usage times of product b at that time.*

Based on the above definition, we have $0 \leq r^t_{ub} \leq 1$ and $\sum_{b\in B} r^t_{ub} = 1$. For ease of future explanation, we group all users' adoption decision at time t $(t = 1, \ldots, T)$ into a matrix $\mathbf{R}^t \in \mathbb{R}^{N\times M}$, and $\mathbf{R} = [\mathbf{R}^1, \ldots, \mathbf{R}^t, \ldots, \mathbf{R}^T]$ denotes users' adoption rate sequence over time.

3.2 Factor-Based Adoption Rate Function

In a real-world market, a user's adoption rate to a product is very complex as many heterogeneous sources around her may contribute to the final decision, such as the user's profile [45], historical preference [3], [25], and the influence from the social network [15]. These previous works usually focused on a particular aspect from a single data source to determine users' adoption tendency. However, they are far from comprehensive since other kinds of rich data sources have not been well exploited. To fill this gap, in this section, we present a factor-based adoption rate function that combines various heterogeneous data sources to describe each user's adoption rate series over time. Specifically, let p^t_{ubd} denote the propensity score of user u's tendency to adopt product b at time t from the dth factor, which is extracted from a particular data source. With heterogeneous data sources, we group all the factors that may influence u's adoption rate r^t_{ub} into a D dimensional *factor vector* $\mathbf{p}^t_{ub} = (p^t_{ub1}, \ldots, p^t_{ubd}, \ldots, p^t_{ubD}) \in \mathbb{R}^{D\times 1}$. Then all users' factor vectors at time t can be represented as a factor tensor $\boldsymbol{P}^t \in \mathbb{R}^{U\times B\times D}$. We leave the details of how to construct the factor tensor $\boldsymbol{P}$ in the next section. To combine all these factors from heterogeneous data sources, we define the factor-based adoption function as:

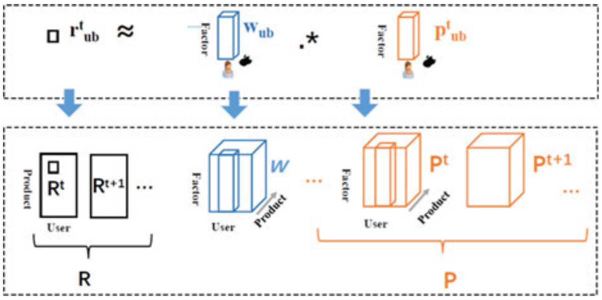


Fig. 2. The factor-based adoption function representation, with each part in the top row represents an element in the bottom row.

Definition 2 (Factor-based Adoption Rate Function). *Given a user u, a product b, and a set of factors that influence a user's product adoption decision, the factor-based adoption rate function models each user's predicted adoption rate $\hat{r}_{ub}^t$ as a weighted combination of all factors:*

$$\hat{r}_{ub}^t = \sum_{d=1}^{D} w_{ubd} \times p_{ubd}^t = \boldsymbol{w}_{ub}^{'} \times \mathbf{p}_{ub}^t. \tag{1}$$

In the above definition, $\mathbf{w}_{ub} = [w_{ub1}, \ldots, w_{ubd}, \ldots, w_{ubD}]'$ is a column vector that shows all the factor weights that influence user u's adoption of product b, and $\mathbf{p}_{ub}^t \in \mathbb{R}^{D\times 1}$ is a same-dimensional column vector that depicts the factors.

In fact, the weight w_{ubd} can be explained from the following two assumptions. The first one is a *generalized assumption* that presumes *all users are influenced equally by these factors*, i.e., $\forall u, v \in U, w_{ubd} = w_{vbd}$. Though simple, this assumption may be not realistic in practice, as different users may have their unique preferences by balancing these factors. e.g., *Alice* is easily influenced by friends' opinions while *Bob* is unwilling to be swayed by others. Obviously, these two users have different weights on friends' adoption decisions. Thus, instead of sharing the same weights for all users in the first assumption, we propose a second *personalized assumption* that argues *each user would balance all these parameters based on their own choices*, i.e., w_{ubd} is personalized and varies among people.

3.3 Problem Definition

With the above factor-based decision function, the problem we study in this paper can be formally defined as:

Definition 3 (Product Adoption Rate Prediction Problem). *Given a product adoption rate matrix sequence $\mathbf{R}^t$ $(1 \le t \le T)$ of user set U to product set B in a competitive market, i.e., the detailed product adoption rate of users to products in this market from time slice 1 to T, and the factor-based decision function (Eq. (1)), our goal is to predict the future product adoption rate of any user u to any product b in this competitive market at the future time slice $T+1$, i.e, $\mathbf{R}^{t+1}$.*

We illustrate the product adoption rate prediction problem with the factor-based adoption function in Fig. 2. As shown in this figure, in order to predict users' future product adoption rate with the factor-based adoption function, we need to figure out two issues. First, how to identify the key factors that influence each user's adoption rate and construct the factor tensor P based on the available heterogeneous data sources? Second, by considering users' preference uniqueness, how to design models that could learn the weight tensor W with both generalized and personalized assumptions effectively and efficiently? We would resolve these two issues in the following of this paper. Specifically, we show the overview of our proposed framework in Fig. 3. The following of this paper is organized as follows. In Section 4, we provide constructing the factor tensor P. In Section 5, we show how to learn the weight tensor under the generalized and personalized assumptions. After that, we further devise algorithms to incorporate product competition into both GAM and BPAM in Section 6. Section 7 presents the experimental results, followed by the conclusions in Section 8. Foe ease of explanation, Table 1 lists the notations used in this paper.

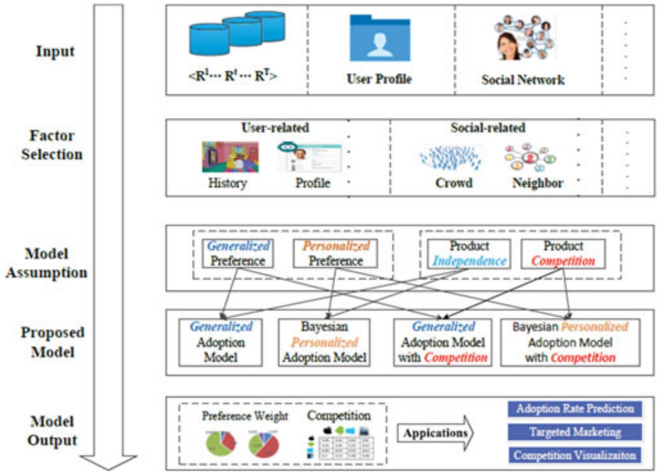


Fig. 3. Overview of our proposed framework. The first block shows the heterogeneous data sources. In the second block, we show four representative factors for adoption rate modeling. Based on the preference and competition assumption in the third block, we learn the corresponding models in the fourth block. At last, the model outputs can be used for various applications as shown in the last block.

4 Key Factors for Adoption Prediction

Here, we introduce how to select key factors and construct the factor tensor P^t from heterogeneous data sources. In general, these factors can be divided into two categories: the user-related factors and the social-related factors.

4.1 User-Related Factors

For many time-series data, the Markov property is widely used to model a stochastic process. Specifically, the Markov property assumes that the probability of being in a state at time t depends only on the state at previous time $t-1$, not on the sequences that before precede it. The intuition underlying this assumption is that the state at time t represents "enough" summary of the past to reasonably predict the future. Given this Markov assumption, it is natural to assume that for each user, the *historical product adoption rate* $r_{ub}^{(t-1)}$ is an important factor that influences the current adoption rate r_{ub}^t

$$p_{ub1}^t = r_{ub}^{(t-1)}. \tag{2}$$

Besides, users' individual profile is also an important factor that considers users' tendency to adopt a product [45]. Without loss of generality, we group each user u's profile features at time t (e.g., gender, location, age and number of friends) as a vector $\mathbf{x}_u^t$. Then, the *individual profile factor* can be modeled as

$$p_{ub2}^t = \mathbf{y}_b' \times \mathbf{x}_u^{t-1}, \tag{3}$$

where $\mathbf{y}_b$ stores the weights over user profile features on product b. Here, the coefficient $\mathbf{y}_b$ can be experimentally

TABLE 1
Mathematical Notations

Symbol	Description
U	user set, $\|U\| = N$
B	product set, $\|B\| = M$
F	the factor set that influences users' decision, $\|F\| = D$
$\mathbf{x}_u^t$	the profile feature vector of user u at time t
$\mathbf{R}^t = [r_{ub}^t]$	real product adoption matrix at time t, $\mathbf{R}^t \in \mathbb{R}^{N\times M}$
$\hat{\mathbf{R}}^t = [\hat{r}_{ub}^t]$	predicted product adoption matrix, $\hat{\mathbf{R}}^t \in \mathbb{R}^{N\times M}$
$\mathbf{P}^t = [p_{ubd}^t]$	adoption rate factor tensor at time t, $\boldsymbol{P}^t \in \mathbb{R}^{N\times M\times D}$
$\mathbf{p}_{ub}^t$	a D-dimensional vector in $\mathbf{P}^t$ indicating all factors that influence u's adoption decision
$\boldsymbol{W} = [w_{ubd}]$	the weight factor tensor, $\boldsymbol{W} \in \mathbb{R}^{N\times M\times D}$
$\mathbf{S} = [s_{ab}]$	the transition matrix of products

learned based on the users' adoption history. A simple learning approach is by minimizing the following loss function from the training data with profile related features as

$$\min_{\mathbf{y}_b} L = \sum_{t=2}^{T}\sum_{u=1}^{N}(r_{ub}^t - \mathbf{y}_b'\mathbf{x}_u^{t-1})^2 + \lambda_c||\mathbf{y}_b||_{Fro}^2,$$

where the first term tries to fit the training data. The second term denotes the Frobenius norm with the regularization term λ_c that controls the model capacity. With the learned feature weight $\mathbf{y}_b$, we can construct the user profile factor as shown in Eq. (3).

4.2 Social-Based Factors

It is well known that users' decisions highly rely on the the aggregated opinions of others, with the belief that the aggregations over a large population can successfully harness the *crowd wisdom* [38]. E.g., most of the product advisory websites display a ranking list of *the most popular products* that show the overall choices of the crowd. We define the *crowd factor* as

$$\forall u, b \quad p_{ub3}^t = \frac{\sum_{u=1}^{N} r_{ub}^{(t-1)}}{N}. \tag{4}$$

In addition to the crowd factor, researchers have converged that direct neighbors' influence in a social network represents an important force affecting users' adoption behaviors [23], thus the prediction results improve by incorporating social neighbors' decisions [6]. Here, we model the *neighbor influence factor* as

$$p_{ub4}^t = \sum_{(u,v)\in A} t_{vu} \times r_{ub}^{(t-1)}, \tag{5}$$

where t_{vu} denotes the influence of neighbor v to u. For simplicity, we set the influence strength as $t_{vu} = \frac{1}{out\_degree(u)}$.

In summary, we present four key factors ($D = 4$) as examples to construct factor tensor $\boldsymbol{P}$. However, we should note that the proposed factor-based adoption rate function is flexible enough and can be easily extended to incorporate other factors when appropriate.

5 Adoption Rate Prediction Models

In this section, we propose solutions to the adoption rate prediction problem. With the extracted factor tensor $\boldsymbol{P}$, our goal turns to learn the weight tensor $\boldsymbol{W}$. Specifically, to deal with users' preference uniqueness, we introduce two assumptions of users' weight tensor, i.e., generalized assumption and personalized assumption. We would propose the solutions to these two assumptions in the following two sections.

Given the real adoption rate sequence $\boldsymbol{R}$, we model the likelihood of the observed product adoption rate as a Gaussian distribution with precision α

$$p(\boldsymbol{R}|\boldsymbol{W}, \boldsymbol{P}) = \prod_{t=2}^{T}\prod_{u=1}^{N}\prod_{b=1}^{M}\mathcal{N}\left(r_{ub}^t \Big| \sum_{d=1}^{D} w_{ubd} \times p_{ubd}^t, \alpha^{-1}\right). \tag{6}$$

5.1 GAM: Generalized Adoption Model

In this section, we introduce the *G*eneralized *A*doption *M*odel with the generalized assumption. The generalized assumption assumes each user is influenced equally by various factors, i.e., $\forall u, v \in U, w_{ubd} = w_{vbd}$. Given this assumption, each row in the weight tensor $\boldsymbol{W}$ turns to be the same, enabling it to be reduced to a weight matrix $\mathbf{W} = [w_{bd}] \in \mathbb{R}^{M\times D}$. In this reduced weight matrix, the element w_{bd} in the bth row and dth column represents the general weight of product b on factor d for all users. As a usual practice for machine learning, we add a Gaussian prior on the weight matrix $\mathbf{W}$ as

$$p(\mathbf{W}) = \prod_{b=1}^{M}\prod_{d=1}^{D}\mathcal{N}(w_{bd}|0, \alpha_w^{-1}). \tag{7}$$

By combining the prior (Eq. (7)) and the likelihood (Eq. (6)), maximizing the log-posterior is equivalent to minimizing the sum-of-square error

$$\min_{\mathbf{W}} L = \sum_{t=2}^{T}\sum_{u=1}^{N}\sum_{b=1}^{M}\left(r_{ub}^t - \sum_{d=1}^{D} w_{bd} \times p_{ubd}^t\right)^2 + \lambda_G tr(\mathbf{W} \times \mathbf{W}'), \tag{8}$$

where $\lambda_G = \frac{\alpha_W}{\alpha}$ and $tr(\mathbf{X})$ denotes the trace of matrix $\mathbf{X}$. In the above loss function, the first term captures the training loss and the second term regularizes the parameters.

In fact, as L is a convex quadratic function, a global minimum could be achieved by updating the gradient of each parameter of $\mathbf{W}$ iteratively until convergence

$$\frac{\partial L}{\partial w_{bd}} = 2\sum_{t=2}^{T}\sum_{u=1}^{N}(\hat{r}_{ub}^t - r_{ub}^t)p_{ubd}^t. \tag{9}$$

Algorithm 1 shows the procedure of the GAM model.

Algorithm 1. The Procedure of GAM

Input: Product adoption sequence $\boldsymbol{R}$, factor tensor $\boldsymbol{P}$.
Output: The product adoption rate for each user at $T + 1$.

```
1: Initialize W with small positive values
2: while Not converged do
3:    for b = 1; b <= M; b++ do
4:       for d = 1; d <= D; d++ do
5:          update w_bd = w_bd − step_size × ∂L/∂w_bd (Eq. (9))
6:       end for
7:    end for
8: end while
9: For each u and b, calculate the predicted adoption rate at T + 1.
10: Return the predicted adoption rate.
```

5.2 BPAM: Bayesian Personalized Adoption Model

We model the personalized assumption of users' preferences in this section. By considering users' preference uniqueness, it is assumed that the weight factors contribute differently to each user, i.e., w_{ubd} is personalized and varies among users. Given this assumption, the weight tensor W is of large size with $N \times M \times D$ elements, thus estimating it solely based on Eq. (6) may lead to serious over-fitting problem. To deal with this issue, we follow the Bayesian approach by placing priors on W [7], [11]. Specifically, let $\mathbf{W_{::d}} \in \mathbb{R}^{N \times M}$ denote the frontal two-dimensional factor weight matrix of tensor W. Alternatively, without confusion, $\mathbf{W}$:: $\mathbf{d}$ is denoted more compactly as $\mathbf{W_d}$. For each factor matrix $\mathbf{W_d}$, a Gaussian prior is assumed

$$p(\mathbf{W_d}) = \prod_{u=1}^{N} \prod_{b=1}^{M} \mathcal{N}(w_{ubd}|\mu_d, \alpha_d^{-1}). \tag{10}$$

It turns out that by combing the prior (Eq. (10)) and the likelihood function (Eq. (6)) with the personalized assumption, *m*aximizing *a* log *p*osterior (MAP) is equivalent to minimizing

$$\begin{aligned}\min_{\boldsymbol{W}} L = &\sum_{t=2}^{T} \sum_{u=1}^{N} \sum_{b=1}^{M} (r_{ub}^t - \hat{r}_{ub}^t)^2 \\ &+ \sum_{d=1}^{D} \frac{\alpha_d}{\alpha} \sum_{u=1}^{N} tr((\mathbf{W_d} - \mu_d) \times (\mathbf{W_d} - \mu_d)'),\end{aligned} \tag{11}$$

where $\forall d \in F, \frac{\alpha_d}{\alpha}$ is a regularization parameter.

Though intuitive, the main drawback of the above MAP estimation is the need for manual complexity control that is essential to making the model generalize well. Actually, the performance is tied carefully to the manual tuning of the hyperparameters to avoid overfitting by the MAP estimation. In practice, we need to tune $2 \times D + 1$ hyperparameters according to Eq. (11) (i.e., $\{\boldsymbol{\Theta_W} = \{\Theta_{W_d}\}_{d=1}^{D} = [\mu_d, \alpha_d]_{d=1}^{D}, \alpha\}$) on the validation set to get the best performance, which is computationally expensive.

Instead of the MAP estimation of the unique user preference assumption that relies on the careful tuning of hyperparameters, we introduce a method named *B*ayesian *P*ernsonlized *A*doption *M*odel that automatically controls model complexity given the observational data. The core idea of BPAM is to further add priors for hyperparameters and maximizes the log posterior over both the parameters and the hyperparameters simultaneously. We add conjugate priors for the hyperparameter set $\{\boldsymbol{\Theta_W} = [\Theta_{W_d}]_{d=1}^{D} = [\mu_d, \alpha_d]_{d=1}^{D}, \alpha\}$ as

$$\begin{aligned}&p(\alpha|a, b) = \mathcal{G}(\alpha|a, b), \\ &\forall d, p(\Theta_{W_d}|\mu_d, \alpha_d) = \mathcal{N}(\mu_d|\mu_0, (\beta\alpha_d)^{-1})\mathcal{G}(\alpha_d|a, b),\end{aligned} \tag{12}$$

where $\mathcal{G}(\alpha_d|a, b)$ is a gamma distribution with a shape parameter a and a rate parameter b. Gamma distribution is widely used as the conjugate prior for univariate Gaussian distribution [36]. For convenience, we also define $\Theta_0 = \{a, b, \mu_0, \beta\}$. Θ_0 depicts our prior understanding of the data, which has little impact on the final results if the data is large enough.

Learning by Gibbs Sampling. Given the observed data sets, the fully Bayesian treatment integrates out all model parameters W and hyperparameters $\{\boldsymbol{\Theta_W}, \alpha\}$, arriving a predictive distribution of future observations. Specifically, the predictive distribution of $\hat{r}_{ub}^t$ is modeled as

$$p(\hat{r}_{ub}^t|\boldsymbol{R}, \Theta_0) = \int p(\hat{r}_{ub}^t|W, \alpha)p(W, \boldsymbol{\Theta_W}, \alpha|\boldsymbol{R}, \Theta_0)d(W, \boldsymbol{\Theta_W}, \alpha). \tag{13}$$

Since the exact inference of the above predicted distribution is analytically intractable, we exploit Gibbs sampling to approximate the true posterior distribution of $p(W, \boldsymbol{\Theta_W}, \alpha|\boldsymbol{R}, \Theta_0)$ [16]. For this method, each step involves replacing the value of one variable by a new value drawn from distributions conditioned on all the other variables. The procedure is repeated by cycling through all the variables until converges to the desired distribution. Then we collect a number of samples and approximate the integral in Eq. (13) by

$$p(\hat{r}_{ub}^t|\boldsymbol{R}, \Theta_0) \approx \frac{1}{L} \sum_{l=1}^{L} p(\hat{r}_{ub}^{t,l}|\mathbf{w}_{\mathbf{ub}}'^{(\mathbf{l})} \times \mathbf{p_{ub}^t}), \tag{14}$$

where L denotes the total number of samples and $\mathbf{w_{ub}^l}$ is a factor preference vector that samples from the lth iteration. Correspondingly, we have

$$\hat{r}_{ub}^t \approx \mathbb{E}\left[\frac{1}{L} \sum_{l=1}^{L} \mathcal{N}\left(\hat{r}_{ub}^{t,l}|\mathbf{w}_{\mathbf{ub}}'^{(\mathbf{l})} \times \mathbf{p_{ub}^t}, \alpha^{-1}\right)\right] = \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{w}_{\boldsymbol{ub}}'^{(\boldsymbol{l})} \times \mathbf{p_{ub}^t}. \tag{15}$$

Now we show how to sample the posterior distribution of each variable in each iteration. Due to the introduction of the conjugate priors, the conditional distributions derived from the posterior distribution share the same form as the prior distributions. As to the weight hyperparameters $\boldsymbol{\Theta_W} = \{\Theta_{W_d}\}_{d=1}^{D}$, the posterior distribution is estimated as

$$\begin{aligned}&p(\mu_d, \alpha_d|\boldsymbol{w_d}, \Theta_0) = \mathcal{N}(\mu_d|\mu_d^*, (\beta_d^*\alpha_d^*)^{-1})\mathcal{G}(\alpha_d|a_d^*, b_d^*), \\ &u_d^* = \frac{\beta u_0 + K\bar{w}_d}{\beta + K}, \quad \beta_D^* = \beta + K, \quad a_d^* = a + \frac{K}{2}, \\ &b_d^* = b + \frac{1}{2} \sum_{u=1}^{N} \sum_{b=1}^{M} (w_{ubf} - \bar{w_f})^2 + \frac{K \times \beta}{2(K + \beta)} (\bar{w_f} - u_0)^2,\end{aligned} \tag{16}$$

where $K = N \times M$, $\bar{w_d}$ is the mean of the weight matrix $\mathbf{W_d}$ of all users' weights for factor d. As to precision parameter α, the posterior distribution follows Gamma distribution

$$\begin{aligned}p(\alpha|\boldsymbol{R}, \boldsymbol{W}) &= \mathcal{G}(\alpha|a^*, b^*), \\ a^* &= a + \frac{K}{2} \times (T - 1), \\ b^* &= b + \frac{1}{2} \sum_{t=2}^{T} \sum_{u=1}^{N} \sum_{b=1}^{M} (r_{ub}^t - \hat{r}_{ub}^t)^2.\end{aligned} \tag{17}$$

For each element of user u's weight vector $\mathbf{w_{ub}}$, i.e., w_{ubd}, the conditional distribution given other relevant parameters is Gaussian

$$\begin{aligned}&\mathcal{N}(w_{ubd}|u_d^*, [\alpha_d^*]^{-1}) \propto \prod_{t=2}^{T} [\mathcal{N}(r_{ub}^t|\hat{r}_{ub}^t, \alpha^{-1})]\mathcal{N}(w_{ubd}|\mu_d, \alpha_d^{-1}), \\ &\text{where} \quad \alpha_d^* = \alpha_d + \alpha \sum_{t=2}^{T} p_{ubd}^t \times p_{ubd}^t, \\ &u_d^* = [\alpha_d^*]^{-1} \left\{ \alpha \sum_{t=2}^{T} p_{ubd}^t (r_{ub}^t - \hat{r}_{ub}^t + w_{ubd} p_{ubd}^t) + \alpha_d \mu_d \right\}.\end{aligned} \tag{18}$$

Tracking the Factor Preference of Users. An important assumption of the BPAM model is that each user has personalized factor preference in the factor-based adoption decision function. In fact, according to the approximation of the predicted adoption rate function as shown in Eq. (15), we can track the factor preference for user u to product b as

$$\mathbf{w_{ub}} = \frac{\hat{r}_{ub}^t}{\mathbf{p_{ub}^t}} \approx \frac{\sum_{l=1}^{L} \boldsymbol{w}_{\boldsymbol{ub}}'^{(\boldsymbol{l})} \times \mathbf{p_{ub}^t}}{L \times \mathbf{p_{ub}^t}} = \frac{\sum_{l=1}^{L} \boldsymbol{w}_{\boldsymbol{ub}}^{\boldsymbol{l}}}{L}. \tag{19}$$

We summarize the Gibbs sampling process of BPAM in Algorithm 2, where we can track the factor preference after the sampling process has completed (as shown in Line 9 of the algorithm).

Algorithm 2. The Gibbs Sampling Process of *BPAM*

Input: Product adoption sequence $\boldsymbol{R}$, factor tensor $\boldsymbol{P}$.
Output: The product adoption rate for each user at $T+1$.
1: The product adoption rate for each user at $T+1$.
2: **for** $l=0; l <= L; l++$ **do**
3: Sample hyperparameter $\boldsymbol{\Theta}^{(l+1)}{}_W \sim p(\boldsymbol{\Theta}_W|\boldsymbol{W}^l, \Theta_0)$.
4: Sample hyperparameter $\alpha^{l+1} \sim p(\alpha|\boldsymbol{R}, \boldsymbol{W}^l, \Theta_0)$.
5: **for** $u \in U, b \in B, d \in F$ **do**
6: Sample $w_{ubd}{}^{(l+1)} \sim p(w_{ubd}|\boldsymbol{R}, \boldsymbol{\Theta}_{\boldsymbol{W}}^{l+1})$.
7: **end for**
8: For each u and b, predict adoption rate at $T+1$.
9: For each u and b, track the factor preference (Eq. (19)).
10: **end for**
11: Return the predicted adoption rate and factor preference of users.

5.3 Time Complexity

As shown in Algorithms 1 and 2, both the two proposed algorithms involve iterations. Specifically, in each iteration, the time complexity of all models is $O(T \times N \times M \times D)$. In practice, as both algorithms would converge after several iterations, the time complexity of these two proposed models increase linearly with the useset size.

6 Adoption Rate Prediction Models with Competition

In a competitive market with multiple products, both GAM and BPAM tackle the multi-product rate prediction problem by dividing it into a set of independent single product adoption rate prediction problems. However, it fails to consider the competition among different products. In this section, we address the problem of how to incorporate product competition process into the proposed GAM and BPAM, and jointly learn user preferences and produce competition in a unified framework. .

Introducing Product Competition as a Factor. As shown in Eq. (2), the historical product adoption factor depicts that for each user u and each product b, the previous adoption rate of this product r_{ub}^{t-1} is a key factor for the current adoption rate r_{ub}^t. This factor treats each product separately. To leverage the competition among multiple products in a fierce market, for each user, it is reasonable to assume that the current product adoption rate of a particular product b is not only influenced by the historical adoption rate of this product, but also the transitions from other products. We introduce a transition matrix $\mathbf{S} \in \mathbb{R}^{M\times M}$ to model the competition, with element s_{ab} denotes the transition probability from a to b. Thus, *the factor of the historical adoption rate with competition* among different products is defined as

$$\begin{aligned} p_{ub1}^t = \sum_{a=1}^{M} s_{ab} \times r_{ua}^{(t-1)} &= \mathbf{S_b}' \times \mathbf{r_u^{(t-1)}} \\ s.t. \quad \forall a, b \in B, \quad & s_{ab} \geq 0 \\ \forall a \in B, \sum_{b=1}^{B} s_{ab} = 1&, \end{aligned} \tag{20}$$

where $\mathbf{r_u^{(t-1)}}$ is a column vector that represents u' adoption rates of all products at time $t-1$. $\mathbf{S_b}$ denotes the bth column of the transition matrix $\mathbf{S}$. $s_{ab} \geq 0$ ensures the transition probability is not negative and $\sum_{b=1}^{B} s_{ab} = 1$ constraints that the total transitions from a to all products equals one. The diagonal elements of $\mathbf{S}$ denote the probability of the state of self transformation. The larger the diagonal value of s_{aa}, the higher loyalty value of the product. The larger the non-diagonal elements of s_{ab}, the higher transition probability from product a to b. Note that if $\mathbf{S}$ is a pre-defined identity matrix with ones on the main diagonal and zeros otherwise, then there exits no competition between different products, i.e, each product is considered independently.

By combining the flexible factor-based adoption rate function (Eq. (1)) and competition (Eq. (20)), we have the *competitive factor-based adoption rate function*

$$\begin{aligned} \hat{r}_{ub}^t &= w_{ub1} \times \mathbf{S_b}' \times \mathbf{r_u^{t-1}} + \sum_{f=2}^{D} \boldsymbol{w}_{ub}' \times \mathbf{p}_{ub}^t \\ &= \begin{bmatrix} w_{ub1} \\ \mathbf{w}_{ub[2:D]} \end{bmatrix}' \begin{bmatrix} \mathbf{S_b' r_u^{(t-1)}} \\ \mathbf{p^t}_{ub[2:D]}, \end{bmatrix} \end{aligned} \tag{21}$$

where $\mathbf{w}_{ub[2:D]}$ denotes the 2th to Dth element of the weight vector $\mathbf{w}_{ub} = [w_{ub1}, w_{ub2}, \ldots w_{ubD}]$.

6.1 GAM-C: Generalized Adoption Model with Competition

The GAM-C shares the generalized assumption of users' preferences in GAM, and it extends GAM by considering the competition among products. Thus, by combing Eq. (21) of competitive adoption rate function and the loss function of GAM in Eq. (8), the optimization goal of GAM-C is

$$\begin{aligned} \min_{\mathbf{W,S}} L &= \sum_{t=2}^{T}\sum_{u=1}^{N}\sum_{b=1}^{M}\left(r_{ub}^t - \sum_{d=1}^{D} w_{bd} \times p_{ubd}^t\right) + \lambda_G tr(\mathbf{W}\times\mathbf{W}') \\ &= \sum_{t,u,b}\left(r_{ub}^t - w_{b1}\mathbf{S_b}' \times \mathbf{r_u^{t-1}} - \sum_{d=2}^{D} w_{bd} \times p_{ubd}^t\right)^2 + \lambda_G tr(\mathbf{W}\times\mathbf{W}'). \end{aligned} \tag{22}$$

In fact, the coupling between $\mathbf{W}$ and $\mathbf{S}$ makes the above loss function non-convex. However, it is convex when either the general user preference matrix $\mathbf{W}$ or the product transformation matrix $\mathbf{S}$ is fixed. This leads us to resort to an alternating-least-square (ALS) optimization technique, where we alternate between re-computing the general user preference matrix $\mathbf{W}$ and the product transformation matrix $\mathbf{S}$. Each alternating step is guaranteed to decrease the loss function of Eq. (22) until convergence.

A. Computing $\mathbf{W}$ *Given* $\mathbf{S}$. With $\mathbf{S}$ given, we can figure out the first historical competition factor as $p_{ub1}^t = \mathbf{S_b}' \times \mathbf{r_u^{(t-1)}}$. Then this problem resembles the updating step in GAM, i.e., for each parameter w_{bd}, we can update it according to Eq. (9).

B. Computing $\mathbf{S}$ *Given* $\mathbf{W}$. It is non-trivial to optimize $\mathbf{S}$ given $\mathbf{W}$ due to the constraints of the transition matrix $\mathbf{S}$. Here, we introduce a same-sized auxiliary matrix $\mathbf{Z}$. Specifically, for each element z_{ab} in the auxiliary matrix, the correspondence between s_{ab} and z_{ab} is

$$s_{ab} = \frac{h(z_{ab})}{\sum_{c\in B} h(z_{ac})}, \tag{23}$$

where $h(x)$ is a positive increasing function. In this paper, we adopt the widely used logistic function $h(x) = \frac{1}{1+exp(-x)}$

for simplicity. Thus, with the auxiliary matrix **Z**, the constrains of **S** can be automatically satisfied.

By introducing the auxiliary matrix, our problem turns to figure out **Z** given **W**. Specifically, the gradient of z_{ab} is

$$\begin{aligned}\frac{\partial L}{\partial z_{ab}} &= \frac{\partial L}{\partial s_{ab}} \times \frac{\partial s_{ab}}{\partial z_{ab}},\\ \frac{\partial L}{\partial s_{ab}} &= \sum_{t=2}^{T}\sum_{u=1}^{N}(\hat{r}_{ub}^{t} - r_{ub}^{t})w_{b1} \times r_{ua}^{t-1},\\ \frac{\partial s_{ab}}{\partial z_{ab}} &= \frac{\sum_c h(z_{ac}) - h(z_{ab})}{\left(\sum_c h(z_{ac})\right)^2} \times h(z_{ab})(1 - h(z_{ab})).\end{aligned} \tag{24}$$

We summarize the GAM-C model in Algorithm 3.

6.2 BPAM-C: Bayesian Personalized Adoption Model with Competition

A key idea of BPAM-C is that it shares the personalized assumption of users' preferences in BPAM with users' unique preferences, and it further extends BPAM by considering competition among products. Following this Bayesian treatment, we can regard the personalized weight tensor $\boldsymbol{W}$ as latent variables, then the goal of training BPAM-C is to find product competition matrix **S** that maximizes the incomplete data likelihood by marginalizing over the parameter set $\boldsymbol{W}, \boldsymbol{\Theta_W}, \alpha$

$$\begin{aligned}\max_{\mathbf{S}} p(\hat{\boldsymbol{R}}|\boldsymbol{R}, \theta_0) = \int & p(\boldsymbol{R}|\boldsymbol{W}, \mathbf{S}, \alpha)\\ & p(\boldsymbol{W}, \boldsymbol{\Theta_W}, \alpha|\boldsymbol{R}, \Theta_0)d(\boldsymbol{W}, \boldsymbol{\Theta_W}, \alpha).\end{aligned} \tag{25}$$

Algorithm 3. The Procedure of GAM-C

Input: Product adoption sequence $\boldsymbol{R}$, factor tensor $\boldsymbol{P}$.
Output: The product adoption rate for each user at $T+1$.
1: **for** $l = 0; l <= L; l++$ **do**
2: Fix **S**, compute the historical preference with competition factor according to Eq. (20)
3: Fix **S**, update each parameter of **W** (Algorithm 1)
4: Fix **W**, update each parameter of **S** (Eq. (24))
5: **end for**
6: For each u and b, calculate the predicted adoption rate at $T+1$
7: Return the predicted adoption rate of users at $T+1$.

Given the above analysis, it is natural to develop the Expectation-Maximization (EM) algorithm to train BPAM-C. Specifically, EM is an iterative method for computing maximum likelihood estimation in problems with missing data. In each iteration, it consists of an E-step: estimate the complete-data likelihood(i.e., $p(\hat{\boldsymbol{R}}|\boldsymbol{R}, \theta_0, \mathbf{S})$) given the current parameter setting of **S** and an M-step: maximize the expected complete data likelihood from the E-step to obtain updated parameter values (i.e., Eq. (25)). The process is repeated until convergence [13]. However, by introducing the Bayesian treatment, the E-step is analytically intractable in BPAM-C. Thus, we resort to the Monte Carlo EM algorithm, which works by a Monte Carlo approximation of the E-step [39]. The details of the E-step and M-step of BPAM-C are listed as follows.

A. Monte Carlo E-Step. In Monte Carlo E-step, we fix **S** and approximate the expectation of the posterior distribution $p(\hat{\boldsymbol{R}}|\boldsymbol{R}, \theta_0, \mathbf{S})$ by marginalizing over $\boldsymbol{W}$. With the fixed **S**, we can figure out the first historical competition factor as $p_{ub1}^{t} = \mathbf{S_b'} \times \mathbf{r_u^{(t-1)}}$. Then this problem resembles BPAM, i.e., the posterior distribution is shown in Eq. (15) and can be approximated with Gibbs sampling process (Algorithm 2).

B. M-Step. In M-step, we need to figure out the updated parameter **S** that maximizes the log likelihood expectation computed in the E-step

$$\max_{\mathbf{S^{(l+1)}}} E(logp(\hat{\boldsymbol{R}}|\boldsymbol{R}, \mathbf{S}^{l}, \theta_0)). \tag{26}$$

By replacing Eqs. (14) and (19) learned from the Monte Carlo E-step into the above equation, maximizing the expected log likelihood is equal to minimizing the following error function:

$$\begin{aligned}\min_{\mathbf{S}} L &= \sum_{t=2}^{T}\sum_{u=1}^{N}\sum_{b=1}^{M}\left(r_{ub}^{t} - \sum_{d=1}^{D} w_{ubd} \times p_{ubd}^{t}\right)\\ &= \sum_{t=2}^{T}\sum_{u=1}^{N}\sum_{b=1}^{M}\left(r_{ub}^{t} - w_{ub1}\mathbf{S_b}' \times \mathbf{r_u^{t-1}} - \sum_{d=2}^{D} w_{ubd} \times p_{ubd}^{t}\right)^2,\end{aligned} \tag{27}$$

With the constrains of **S**, we also introduce an auxiliary matrix of **Z** as shown in Eq. (23). Then we could also compute the gradient of **S** similarly as illustrated in Eq. (24).

We summarize the Monte-Carlo EM algorithm for training BPAM-C in Algorithm 4.

Algorithm 4. The Monte Carlo EM Procedure of BPAM-C

Input: Product adoption sequence $\boldsymbol{R}$, factor tensor $\boldsymbol{P}$.
Output: The product adoption rate for each user at $T+1$.
1: **while** Not converged **do**
2: Fix **S**, compute the historical preference with competition factor according to Eq. (20)
3: Fix **S**, update the predicted adoption rate with Gibbs sampling according to Algorithm 2
4: Track the factor preference tensor $\boldsymbol{W}$ of each user according to Eq. (19)
5: Fix $\boldsymbol{W}$, update each parameter of **S** according to Eq. (24)
6: **end while**
7: Return the predicted adoption rate and factor preference of users.

6.3 Time Complexity

As shown in Algorithms 3 and 4, both the two proposed algorithms need to iteratively update the weight tensor $\boldsymbol{W}$ and the competition matrix **S**. In each iteration, as the time complexity for updating each element of S takes $O(T \times N)$, the total complexity is $O(T \times N \times M^2)$ for matrix **S**. By combining the time complexity of GAM and BPAM in Section 5.3, the total time complexity is $O(L \times (T \times N \times M \times \ (M + D)))$ for both GAM-C and BPAM-C, where L denotes the outer iteration times of these two algorithms. In practice, as the number of competing products are much smaller than the user size, GAM-C and BPAM-C cost more time than the corresponding simplified models of GAM and BPAM. However, the time complexity of these two models still linearly increase with the user size, thus they are applicable to real-world product adoption prediction tasks with hundreds of millions of users.

7 Experiments

In this section, we conduct experiments on two real-world datasets.

7.1 Experimental Setup

Datasets. With users' digital footprints have been accumulated by online service providers, we provide an innovative online social media, i.e., the leading Chinese social network and microblog platform *Weibo*[1] for tracking users' adoption of products over time. Specifically, when a user posts a message, *Weibo* would forward an enriched message to all of the user's followers, including the post, the timestamp and the sending device. As shown in Fig. 4, the sending device presents rich information about how a user accesses this platform, i.e., through a PC client (Fig. 4a) or a mobile device (Fig. 4b). If this user accesses *Weibo* through a mobile device, the detailed mobile brand information is displayed directly (e.g., iPhone as shown in Fig. 4b). These enriched message streams provide valuable sources to track users' Internet access patterns (mobile access or traditional PC client) and the smart device adoption (the brand information is displayed directly over the mobile access). We devise two datasets based on this platform:

Mobile Device (*MD*): It depicts users' preferences of adopting mobile products. In fact, there are many Android devices with different brands (e.g., *Samsung*, *XiaoMi*, *Huawei* and so on), with each takes less than 5 percent of the the total market share. We group all Android devices into the device type *Android*. Then we have four mobile device products in the MD dataset: *iPhone*, *Android*, *Windows* and *Tablet*. After that, we consider the active users that have at least two products in the training data to ensure users have alternatives to transit between different products. For each user, the adoption rate of a product at a particular time is computed as the number of posts sent by this device divided by the total number of mobile posts the active user sends at that time.

Internet Access Technology (*IAT*): This dataset provides the choice of users' different access patterns to the Internet. In this market, there are two products: traditional *PC access Technology* pattern through PC client and *Mobile Technology* access. Specifically, the mobile technology access describes users prefer to use the mobile devices to access Internet (compared to traditional PC client). In practice, the mobile technology adoption rate is computed as the number of posts the target user sends by mobile devices divided by the total number of messages she sends at that time slice. In data preprocessing step, to track users' transitions among different products, we filter out users that only adopt one access pattern to *Weibo* in the training data.

For preparing these two datasets, we tracked and crawled about 230 thousand users and their profiles (e.g., age, location and occupation), with corresponding 15 million social relations and 30 million post streams in year 2012 from *Weibo*. We treat each month as a time slice, thus every user has 12 adoption rate records among year 2012(one adoption rate record for each time slice). We filter out users that post less than 5 messages at any time slice and only select those active users. After data pruning, the detailed statistics of these two datasets are summarized in Table 2. To better illustrate users' adoption rate changes over time, for each user u and product b, we calculate the adoption change at time $t\,(2 \le t \le T)$ as: $|r^t_{ub} - r^{t-1}_{ub}|$. Fig. 5 displays the distribution of the adoption change over time, where the center point shows the mean of all users' adoption change at that time, and the bar shows the standard deviation. We observe that most users change product adoption rates over time, and the variance of the change rate between different users is very large. This also empirically validates the soundness of the proposed product adoption rate concept as users change preferences quickly over time.

1. weibo.com

(a) A post from PC client (b) A post from mobile device

Fig. 4. Two sample posts from Weibo.

Please note that, our proposed problem of product adoption rate prediction differs from the traditional recommender system tasks as we focus on predicting the future likelihood of adopting the durable products that users have already bought in the past, while the recommender systems infer each user's preference to the nondurable products that have not been rated before. Thus, the traditional datasets for recommender systems, such as the movie dataset [35], the travel package dataset [29], and the TV dataset [20], could not be used for this particular task. In the meantime, though someone may argue that it seems limited to use the Weibo data to capture users' adoption rate of the MD and IAT datasets, we do so for the following reason: One one hand, more than 90 percent smarthpone owners use a social networking service on mobile devices at lease weekly [1]. On the other hand, *Weibo* is the leading and largest social network platform in China. Thus, we claim, the Weibo data provides a good approximation of users' adoption rates of the MD and IAD datasets.

Baselines. As our proposed problem could be regarded as a prediction task, we borrow some classical baselines from time series analysis and product adoption prediction modeling for comparison. For time series analysis, we first adopt the *Auto Regressive* model that assumed a user's future decision is a linear combination of her previous adoption history [3]. Besides, we also leverage the one-order Markov property that presumed the conditional probability distribution of the future state only depends on the current state, i.e., $\hat{r}^{T+1}_{ub} = r^T_{ub}$. We call this baseline as the *N*earest *H*istory (NH) method. These two time series baselines neglected the correlation between different products, thus we also select *V*ector *A*uto *R*egression to capture the interdependencies among multiple products [31]. For product adoption prediction models, a common practice is to first construct the features of each user and then train a classification model based on these features [6]. Here we choose the Classification *A*nd *R*egression *T*ree (CART) [8] that was used in [6] for product adoption prediction. Note that as nearly all of these previous works assumed users' adoption preferences are binary values, thus the historical product adoption rates of users are not available in these traditional feature-based prediction models. For fair comparison, we add the historical adoption rate (Eq. (2)) as a feature in the CART model.

We split each dataset as follows: we use the adoption rate records from time 1 to T for training and the data at time $T+1$ for testing. As we have 12 records for each user, $T = 11$ on both datasets. Among all users, we randomly select 5 percent of users as the validation set. There are several parameters in the baselines, e.g., the order of time slices in AR and VAR. For fair comparison, we tune all the parameters in the validation dataset to ensure the best performance. In our

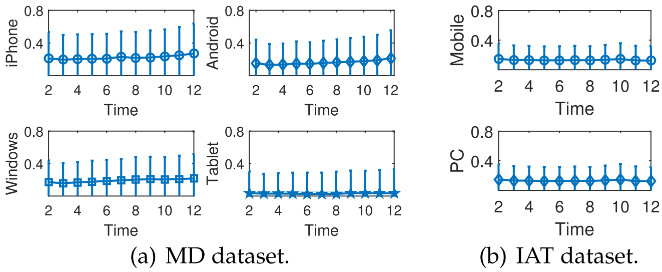


Fig. 5. The mean and standard deviation of users' product adoption rate change over time. In each figure, the center dot shows the mean of the product adoption rate change of all users at that time, and the horizontal bar displays the standard deviation.

proposed GAM, λ_G controls the complexity of the model. As the training records are much larger than the parameters, setting λ_G in a reasonable range (e.g., from 0.1 to 10) would not impact the results. In practice, we set $\lambda_G = 1$. In our proposed Bayesian models of BPAM and BPAM-C, the hyperparmeters (θ_0) are set with small values without tuning. We initialize the parameters of W in BPAM and BPAM-C with a mean from the learned parameters of GAM and stop the model learning when the validation error increases. All the experiments are performed on a 2.5 GHZ4-Core CPU with 8G main memory PC and the programs are implemented in C++.

Evaluation Metrics. As our goal is to predict users' adoption rate at time $T+1$ as accurate as possible, a natural metric is to calculate the *R*oot *M*ean *S*quared *E*rror (*RMSE*) [35] as

$$RMSE = \sqrt{\frac{\sum_{u\in U, b\in B}(\hat{r}_{ub}^{T+1} - r_{ub}^{T+1})^2}{N \times M}}. \tag{28}$$

Besides, instead of directly comparing the prediction error, one of the most important applications of the product adoption rate prediction task is targeted marketing, i.e., for each product manufacture, the company would like to identify a small group of customers that are highly likely to adopt this product. Thus, the company could put marketing efforts on these targeted users. To evaluate the performance of targeted marketing efforts, we adopt the following two metrics: *Top-1* and *D*egree *o*f *A*greement (*DOA*).

Top-1 measure captures the marketing efforts by selecting the top-1 ranked product for each user among all products. At time $T+1$, for each user, there are M product adoption rate prediction results. For a particular product manufacture b, he cares about whether the corresponding product in his company has the leading product adoption rate among all products. Top-1 measures the fraction of users that the predicted top-ranked product with the largest predicted product adoption rate is equal to the real top-ranked product. Specifically, for each user u, let R_u (P_u) be the product that has the largest real (predicted) adoption rate at $T+1$ for u, then $Top-1$ is defined as

$$Top-1 = \frac{\sum_{u\in U}\delta(P_u = R_u)}{N}, \tag{29}$$

where $\delta(x)$ is an indicator function that equals 1 if x is true and 0 otherwise. In other words, $\delta(P_u = R_u)$ equals 1 if $P_u = R_u$. The larger the Top-1 value, the better the ranking performance.

DOA measure captures the marketing efforts by selecting the top ranked users for each product b that have the largest product adoption rate among all users. Specifically, for each product b, we select the top 10 percent of users that have the largest product adoption rates at the test time $T+1$ as the candidate targeted userset P_{ub} and the remaining users as the negative userset N_{ub} ($\forall b \in B, \forall i \in P_{ub}, j \in T_{ub}, r_{ib}^{T+1} > r_{jb}^{T+1}$). Then we use DOA measure to calculate the percentage of user pairs that are correctly ranked with respect to these two usersets [28]. Here, for each product b, the DOA_b measure is defined as

$$DOA_b = \frac{\sum_{i\in T_{ub}, j\in N_{ub}}\delta(\hat{r}_{ib}^{T+1} - \hat{r}_{jb}^{T+1})}{|T_{ub}| \times |N_{ub}|}, \tag{30}$$

where $\delta(x)$ is an indicator function. Then the DOA measure is averaged among all products. The DOA value ranges from 0 to 1 and the larger the better.

7.2 Effectiveness Comparison

In this section, we show the overall effectiveness comparison of all models under different metrics. We plot the the predictive performance on MD dataset in Fig. 6, where the T value ranges from $[11, 7]$ with a decrease of 1. As can be seen from this figure, our proposed four models perform better than all baselines. Specifically, among all baselines, NH and AR only take each user's adoption history for the prediction task. By leveraging the correlation among different products, VAR improves over NH and AR. By combing all heterogeneous data sources, CART shows the best performance among all baselines. As to our proposed four models, BPAM-C shows the best performance compared to all other proposed models, followed by GAM-C and BPAM. Based on these observations, we conclude that it is effective to model users' adoption sequences with the factor-based adoption rate function. Thus, GAM always performs better than all baselines. GAM-C and BPAM improve over GAM by considering product competition and users' uniqueness. The results of BPAM-C indicate the superiority of combining both product competition and

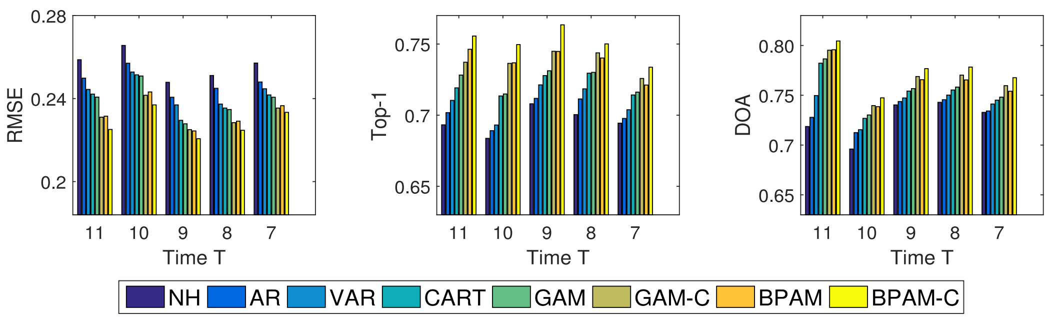


Fig. 6. The overall performance on MD dataset.

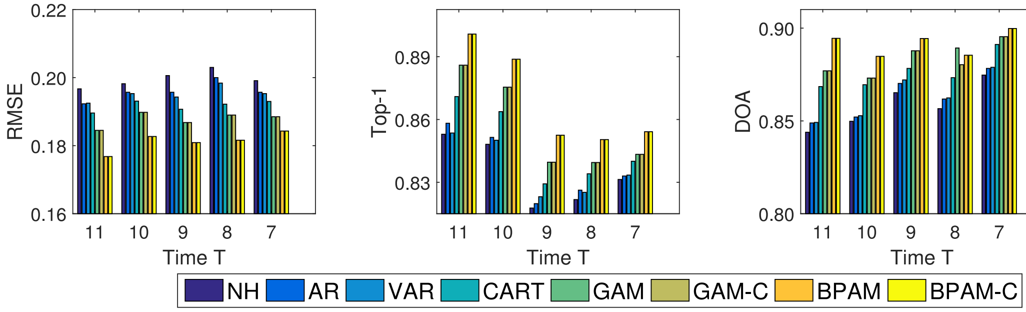


Fig. 7. The overall performance on IAT dataset.

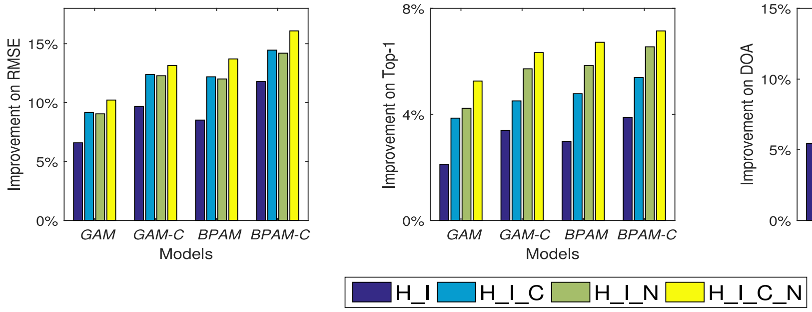


Fig. 8. The relative improvement over the factors on MD dataset (T=11).

TABLE 2
The Statistics of the Two Datasets

DataSet	#Users	#Products	#Social Edges	#Time	Detailed Products
MD	14,121	4	334,471	12	*iPhone, Android, Windows, Tablet*
IAT	120,608	2	3,794,295	12	*Mobile Technology, PC Access Technology*

preference uniqueness in the modeling process. Though the detailed metric values vary with T, the overall trend is that the relative improvement of our proposed models increase as T increases. A possible reason is that we have more data to train our model. E.g., the average RMSE improvement of BPAM-C improves from 9.8 to 16.09 percent over NH T increases from 7 to 11.

The effectiveness comparison for the IAT dataset is shown in Fig. 7. We observe that our proposed models also perform the best among all models, which is consistent with the results on MD. Nevertheless, there is one major distinction. The results of GAM_C is similar to GAM, so as the comparison between BPAM_C and BPAM. That is to say, there is no improvement after adding the product competition factor. The reason is that, there are only 2 products on the IAT dataset (as shown in Table 2). For a target user and each product a, the historical adoption rate of the remaining product b can be easily calculated, thus adding the competition among these two products do not add any information gain. In summary, based on the above experimental results, we conclude that our proposed models have better performance than the baselines with varying T values. In the following, without loss of generality, we set $T = 11$ and make further analysis based on this setting.

7.3 Model Analysis

In this section, we further study some important properties of our proposed models. In fact, we have introduced three main challenges in designing our proposed models, i.e., the data heterogeneity, product competition and the unique user preferences. In the following of this section, we would show the model analysis from the three aspects.

Factor Importance. To deal with the data heterogeneity challenge, we introduce a factor-based adoption function to capture users' adoption rate changes over time. Specifically, we consider four factors from heterogeneous data sources: the *H*istorical rate (H), the *I*ndividual characteristics (I), the *C*rowd wisdom (C) and the *N*eighbor influence (N). Here, we analyze the effect of these four factors underlying people's adoption decisions. E.g., H_I denotes considering the historical rate and the individual factor. Figs. 8 and 9 show the relative gain of these factors for all proposed models compared to the naive NH model, which can be seen as only considering the active user's previous adoption rate (Eq. (2)). Note that, in GAM-C and BPAM-C, the historical factor is modeled with the competition among different products (Eq. (20)). From these two figures, we find that all these factors can improve the final prediction results to some extent for all of our proposed models. In particular, there are about 3 to 8 percent improvement by using both the historical and individual factors with RMSE metric. Moreover, both the crowd wisdom and neighbor influence generate a relative 1 percent improvement. Further combining these two social factors give about 2 percent relative improvement over the user-related factors, indicating the social related factors can complement the user-related factors to some extent. Also, the improvement on the RMSE metric is larger than the remaining two ranking metrics. We guess a possible reason is that all of our proposed models do not optimize the ranking results directly. In summary, all the chosen four factors contribute to the final prediction

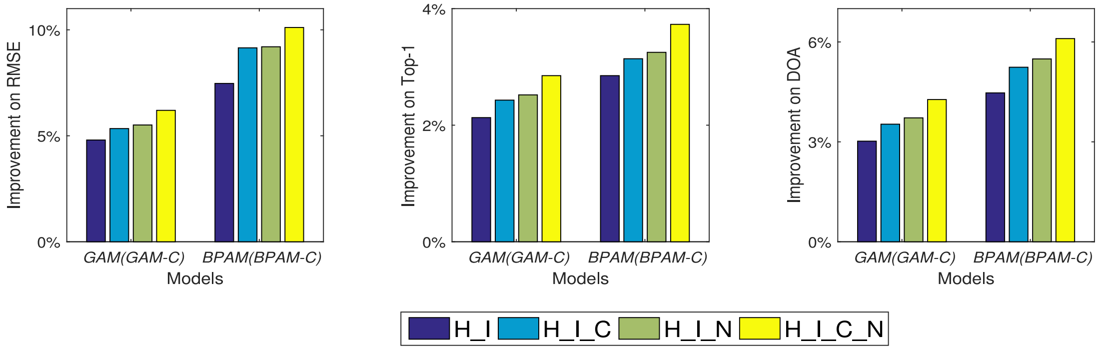


Fig. 9. The relative improvement over the factors on IAT dataset (T=11).

task, and the user-related factors give more improvement than the social-related ones.

Product Competition Analysis. After introducing the product competition factor, BPAM-C performs better than BPAM, and GAM-C also has better prediction power than GAM on the MD dataset. As BPAM-C has the best performance among all of our proposed models, we visualize the learned competition matrix S from BPAM-C on MD dataset in Fig. 10. In this figure, each row shows the transition from a particular product to all the products in the market, and each column depicts the transition from all the product to this particular product. The lighter the color between a product in the ath row and the bth column, the larger the transition probability from product a to b. As can be seen from this figure, for each row of a particular product, the largest transition probability all lies at the diagonal element, indicating each product has the largest transition probability to itself. The transitions from a particular product to other products are all very small with the exception from the transition from *Windows* to *iPhone*, which is about 0.13. Please note that we do not report the competition results on IAT dataset as there are only two products in this market, and we have also explained why there is no performance improvement on this dataset after introducing the competition factor.

Personalized Preference Visualization. We would visualize the learned factor weights of several typical users to get a more straightforward observation. Among all our proposed models, we select BPAM-C as it shows the best performance. Specifically, we first calculate the personalized weight with each factor for each user by Eq. (19) of the BPAM-C model. Then, we depict the normalized factor weights of several typical users manually chosen from the two datasets in Fig. 11. As can be seen, the actual preference to each factor varies among users, i.e., some users may be influenced easily by the crowd factor (e.g., about 30 percent in the third part of Fig. 11a) while others are more likely to be influenced by the individual characteristics. (e.g., about 30 percent in the second part of Fig. 11b). Based on this case study, we could empirically understand that users are influenced differently by these factors and the proposed personalized models (e.g., BPAM and BPAM-C) can help capture the personalized aspects in decision-making.

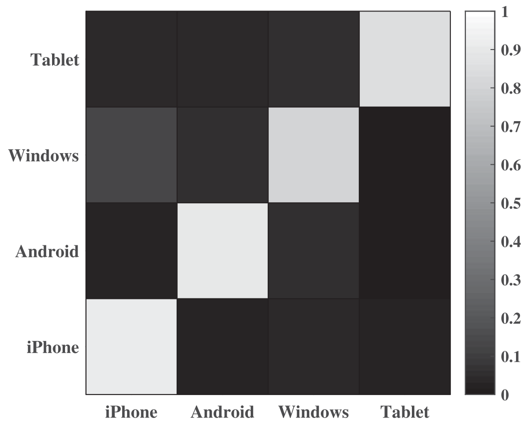


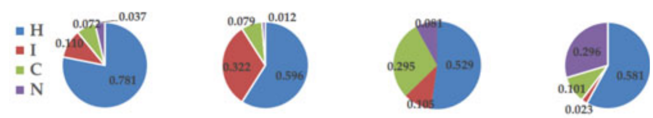
Fig. 10. Visualization of competition matrix S on MD dataset.

7.4 Efficiency and Scalability

To evaluate the efficiency and scalability of the proposed models, we test the running time of each model on different segmentations of the whole user set (i.e., 20, ...,100 percent). Fig. 12 shows the total running time of each proposed model of the two datasets. We observe that all the models are very time-efficient. After selecting the relevant features, the average running time is less than 200 seconds for both GAM and BPAM. The GAM-C and BPAM-C cost more time as these two methods need iteratively update the user preference factor and the product competition matrix. Nevertheless, the computational costs are still linear with the size of the users for all proposed models, which shows that our proposed models are fast enough to be applied to the real-world production adoption rate prediction tasks.

8 Conclusions and Future Work

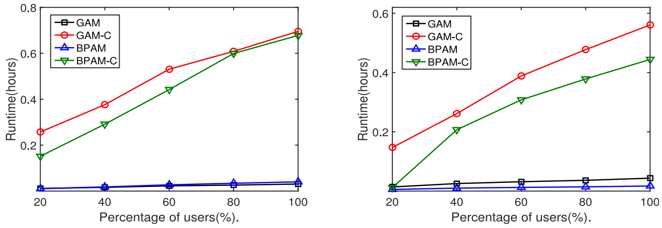
In this paper, we studied the problem of tracking and predicting users' adoption rates of products in a competitive market. We first introduced a flexible factor-based decision function to capture the change of users' product adoption rate over time, where various factors from heterogeneous data sources can be generally leveraged. Using this factor-based decision function, we developed the GAM and



(a) MD dataset.

(b) IAT dataset.

Fig. 11. Case study of the learned weights of typical users.



(a) MD dataset. (b) IAT dataset.

Fig. 12. The runtime of different models.

BPAM models by assuming the generalized and personalized user preference respectively. Furthermore, we presented how to leverage product competition effect into the GAM and BPAM models, and designed the GAM-C and BPAM-C models by simultaneously learning product competition and users' preferences with both generalized and personalized assumptions. Finally, the experimental results on two real-world datasets clearly validated the effectiveness and efficiency of our proposed models.

Since predicting the product adoption rate in a competitive market is an emerging research topic following the map from data collection, to problem definition, and to model design, we notice that there are still some open directions for future research. On one hand, though our proposed adoption rate function is flexible to incorporate various factors that may influence users' decision, our current solution mainly focused on the human designed factors. Is it possible to design machine learning algorithms to automatically select important factors from heterogeneous data sources? On the other hand, in the problem definition process, the competitive market with several competing products are known as prior knowledge obtained by merchants or the domain experts. We would like to explore how to automatically mine the competition relationships from a large collection of products in the future.

Acknowledgments

This work was supported in part by grants from the National Key Research and Development Program (Grant No. 2017YFB0803301), and the National Science Fund for Excellent Young Scholars (Grant No. 61722204), and the National Natural Science Foundation of China (Grant No. 71490725, 61472116, and 61602147, and U1605251), and the Natural Science Foundation of Anhui Province (Grant No. 1708085QF155). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association of CAS (No. 2014299). This paper was an expanded version of [42], which appeared in *SDM 2015* as "Best of SDM".

References

[1] Lightspeed research. [Online]. Available: https://econsultancy.com/blog/9792–73-of-smartphone-owners-use-a-social-networking-app-on-a-daily-basis, accessed Mar. 28, 2017.
[2] 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. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[3] H. Akaike, "Fitting autoregressive models for prediction," *Ann. Inst. Statistical Math.*, vol. 21, no. 1, pp. 243–247, 1969.
[4] S. Bao, R. Li, Y. Yu, and Y. Cao, "Competitor mining with the web," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 10, pp. 1297–1310, Oct. 2008.
[5] F. M. Bass, "Comments on a new product growth for model consumer durables the bass model," *Manage. Sci.*, vol. 50, pp. 1833–1840, 2004.
[6] R. Bhatt, V. Chaoji, and R. Parekh, "Predicting product adoption in large-scale social networks," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1039–1048.
[7] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.
[8] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
[9] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact," *MIS Quart.*, vol. 36, no. 4, pp. 1165–1188, 2012.
[10] F. C. T. Chua, H. W. Lauw, and E.-P. Lim, "Generative models for item adoptions using social correlation," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 9, pp. 2036–2048, Sep. 2013.
[11] K. M. Cremers, "Multifactor efficiency and Bayesian inference," *J. Bus.*, vol. 79, no. 6, pp. 2951–2998, 2006.
[12] G. S. Day, A. D. Shocker, and R. K. Srivastava,"Customer-oriented approaches to identifying product-markets," *J. Marketing*, vol. 43, pp. 8–19, 1979.
[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statistical Soc.*, vol. 39, pp. 1–38, 1977.
[14] K. Eliaz and R. Spiegler, "Consideration sets and competitive marketing," *Rev. Econ. Stud.*, vol. 78, no. 1, pp. 235–262, 2011.
[15] X. Fang, P. J.-H. Hu, Z. Li, and W. Tsai, "Predicting adoption probabilities in social networks," *Inf. Syst. Res.*, vol. 24, no. 1, pp. 128–145, 2013.
[16] A. Gelfand and A. Smith, "Sampling-based approaches to calculating marginal densities," *J. Amer. Statist. Assoc.*, vol. 85, no. 410, pp. 398–409, 1990.
[17] X. He, M. Gao, M.-Y. Kan, Y. Liu, and K. Sugiyama, "Predicting the popularity of web 2.0 items based on user comments," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 233–242.
[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. Int. Conf. World Wide Web*, 2017, pp. 173–182.
[19] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 549–558.
[20] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. IEEE Int. Conf. Data Mining*, 2008, pp. 263–272.
[21] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, and S. Yang, "Social recommendation with cross-domain transferable knowledge," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3084–3097, Nov. 2015.
[22] M. Jiang, P. Cui, F. Wang, X. Xu, W. Zhu, and S. Yang, "FEMA: Flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1186–1195.
[23] J. Kleinberg, "Cascading behavior in networks: Algorithmic and economic issues," *Algorithmic Game Theory*, vol. 24, pp. 613–632, 2007.
[24] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, pp. 89–97, 2010.
[25] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Comput.*, vol. 42, no. 8, pp. 30–37, 2009.
[26] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Mach. Learning*, 2001, pp. 282–289.
[27] T. Lappas, G. Valkanas, and D. Gunopulos, "Efficient and domain-invariant competitor mining," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 408–416.
[28] Q. Liu, E. Chen, H. Xiong, C. H. Ding, and J. Chen, "Enhancing collaborative filtering by user interest expansion via personalized ranking," *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)*, vol. 42, no. 1, pp. 218–233, Feb. 2012.
[29] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong, "Personalized travel package recommendation," in *Proc. IEEE Int. Conf. Data Mining*, 2011, pp. 407–416.
[30] Z. Liu and M. Hauskrecht, "Learning linear dynamical systems from multivariate time series: A matrix factorization based framework," in *Proc. SIAM Int. Conf. Data Mining*, 2016, pp. 810–818.

[31] H. Lutkepohl, "New introduction to multiple time series analysis," *Econometric Theory*, vol. 22, no. 5, pp. 961–967, 2005.

[32] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 203–210.

[33] Z. Ma, G. Pant, and O. R. Sheng, "Mining competitor relationships from online news: A network-based approach," *Electron. Commerce Res. Appl.*, vol. 10, no. 4, pp. 418–427, 2011.

[34] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 785–794.

[35] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.

[36] K. P. Murphy, "Conjugate Bayesian analysis of the gaussian distribution," Technical Report, 2007, Available: https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf.

[37] L. Rabiner and B.-H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 4–16, Jan. 1986.

[38] J. Surowiecki and M. P. Silverman, "The wisdom of crowds," *Anchor Books*, 2006.

[39] G. C. Wei and M. A. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *J. Amer. Statistical Assoc.*, vol. 85, no. 411, pp. 699–704, 1990.

[40] L. Wu, E. Chen, L. Qi, L. Xu, T. Bao, and Z. Lei, "Leveraging tagging for neighborhood-aware probabilistic matrix factorization," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1854–1858.

[41] L. Wu, et al., "Modeling the evolution of users' preferences and social links in social networking services," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1240–1253, Jun. 2017.

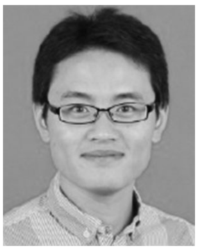
[42] L. Wu, Q. Liu, E. Chen, X. Xie, and C. Tan, "Product adoption rate prediction: A multifactor view," in *Proc. SIAM Int. Conf. Data Mining*, 2015, pp. 154–162.

[43] L. Wu, Q. Liu, E. Chen, N. J. Yuan, G. Guo, and X. Xing, "Relevance meets coverage: A unified framework to generate diversified recommendations," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 3, 2016, Art. no. 39.

[44] H. Zhang, G. Kim, and E. P. Xing, "Dynamic topic modeling for monitoring market competition from online text and image data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1425–1434.

[45] Y. Zhang and M. Pennacchiotti, "Predicting purchase behaviors from social media," in *Proc. Int. Conf. World Wide Web*, 2013, pp. 1521–1532.

Le Wu received the PhD degree in computer science from the University of Science and Technology of China (USTC). She is currently a faculty member with the Hefei University of Technology (HFUT), China. Her general area of research is data mining, recommender system, and social network analysis. She has published several papers in referred journals and conferences, such as the *IEEE Transactions on Knowledge and Data Engineering*, the *ACM Transactions on Intelligent Systems and Technology*, AAAI, IJCAI, KDD, SDM, and ICDM. She is the recipient of the Best of SDM 2015 Award.



Qi Liu received the PhD degree in computer science from USTC. He is an associate professor with USTC. His general area of research is data mining and knowledge discovery. He has published prolifically in refereed journals and conference proceedings, e.g., the *IEEE Transactions on Knowledge and Data Engineering*, the *ACM Transactions on Information Systems*, the *ACM Transactions on Knowledge Discovery from Data*, the *ACM Transactions on Intelligent Systems and Technology*, KDD, IJCAI, AAAI, ICDM, SDM, and CIKM. He is a member of the ACM and the IEEE. He received the ICDM 2011 Best Research Paper Award and the Best of SDM 2015 Award.

Richang Hong (M'12) received the PhD degree from USTC, in 2008. He is currently a professor with USTC. He has co-authored more than 60 publications in the areas of his research interests, which include multimedia question answering, video content analysis, and pattern recognition. He is a member of the Association for Computing Machinery. He was a recipient of the best paper award in the ACM Multimedia 2010. He is a member of the IEEE.

Enhong Chen (SM'07) received the PhD degree from USTC. He is a professor and vice dean of the School of Computer Science, USTC. His general area of research includes data mining and machine learning, social network analysis, and recommender systems. He has published more than 100 papers in refereed conferences and journals, including the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Mobile Computing*, KDD, ICDM, NIPS, and CIKM. He was on program committees of numerous conferences including KDD, ICDM, and SDM. His research is supported by the National Science Foundation for Distinguished Young Scholars of China. He is a senior member of the IEEE.

Yong Ge received the PhD degree in information technology from Rutgers, the State University of New Jersey, in 2013. He is an assistant professor of management information systems, University of Arizona. His research interests include data mining and business analytics. He received the ICDM-2011 Best Research Paper Award. He has published prolifically in refereed journals and conference proceedings, such as the *IEEE Transactions on Knowledge and Data Engineering*, the *ACM Transactions on Information Systems*, the *ACM Transactions on Knowledge Discovery from Data*, ACM SIGKDD, SIAM SDM, IEEE ICDM, and ACM RecSys.

Xing Xie (SM'09) is currently a senior researcher in Microsoft Research Asia, and a guest PhD advisor with USTC. His research interest include spatial data mining, location-based services, social networks, and ubiquitous computing. In recent years, he was involved in the program or organizing committees of more than 70 conferences and works. Especially, he initiated the LBSN workshop series and served as program co-chair of ACM Ubicomp 2011. He is a senior member of the ACM and the IEEE, and a distinguished member of the China Computer Federation (CCF).

Meng Wang (SM'17) received the BE and PhD degrees from USTC, in 2003 and 2008, respectively. He is a professor with HFUT. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored more than 200 book chapters, journal, and conference papers in these areas. He is the recipient of the ACM SIGMM Rising Star Award 2014. He is an associate editor of the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Circuits and Systems for Video Technology*, and the *IEEE Transactions on Neural Networks and Learning Systems*. He is a senior member of the IEEE.