Product Adoption Rate Prediction: A Multi-factor View

Le Wu* Qi Liu* Enhong Chen* Xing Xie† Chang Tan*

Abstract

As the worlds of commerce and Internet technology become more inextricably linked, a large number of user consumption series become available for creative use. A critical demand along this line is to predict the future product adoption for the merchants, which enables a wide range of applications such as targeted marketing. However, previous works only aimed at predicting if one user will adopt this product or not; the problem of adoption rate (or percentage of use) prediction for each user is still underexplored due to the complexity of user decision-making process. To that end, in this paper we present a comprehensive study for this product adoption rate prediction problem. Specifically, we first introduce a decision function to capture the change of users' product adoption rate, where various factors that may influence the decision can be generally leveraged. Then, we propose two models to solve this function, the Generalized Adoption Model (GAM) that assumes all users are influenced equally by these factors and the Personalized Adoption Model (PAM) that argues each factor contributes differently among people. Furthermore, we extend the PAM to a totally Bayesian model (BPAM) that can automatically learn all parameters. Finally, extensive experiments on two real-world datasets not only show the improvement of our proposed three models, but also give insights to track the effects of the various factors for product adoption decisions.

1 Introduction

With the help of information technology, the digital records of users' daily routines have provided an unprecedented opportunity to track the product adoption series of users. As a trend, leveraging these series for future adoption prediction has attracted increasing attention from both academy and industry [12, 5]. Accurate prediction not only helps to understand the human decision process, but is also crucial to a wide range of business applications ranging from personalized recommendation [2], targeted marketing [4] to customer churn prediction [22].

In the literature, many efforts have been devoted to the product adoption prediction problem, where the product could be a particular brand, a technology service or an opinion [10, 23, 8]. These works usually classified users in two categories, the adopters that have already consumed this product and the non-adopters that have not consumed it till now. Then classification methods were proposed to model the future adoption possibilities to those non-adopters. Though intuitive, this binary representation of users' product adoption preferences is too coarse to characterize a user's adoption status. E.g., *Alice* frequently uses the *mobile technology* to access Internet (i.e., access Internet through mobile devices) while *Bob* only tries to use this *mobile technology* once at time t, indicating *Alice* is more likely to adopt this technology service than *Bob*. Nevertheless, previous works would assign a value 1 of adoption status to both users while neglecting their detailed preferences. Actually, in the real world, a consumer usually has several alternatives among a particular kind of products and she may switch among these products due to many internal and external factors. Instead of the binary buy-or-not adoption values, the merchants care more about users' commitment to the product. Thus, we argue, the *adoption rate*, i.e., the usage rate and regularity at which consumers use a product, is more appropriate to describe users' preferences to the products. Based on this measure, *Alice* and *Bob* could have a large difference for the *mobile technology* product. Then, the problem turns to how to predict the future product adoption rate of each user based on the adoption rate series.

Unfortunately, none of the existing models (e.g., the time-series forecasting models [3] and the hidden markov model [18]) can be applied directly due to the unique characteristics of this problem. First, a user's decision making process is very complex as many factors around her may contribute to the final decision, e.g., the users' own profiles [23], the social network structure [8] and the overall crowd awareness [20]. How to design a flexible prediction model that can leverage many different factors remains pretty much open. Second, users have their own preferences by considering the various factors underlying the decision process. E.g., some users may weight more on social neighbors' opinions while others are unlikely to change their decisions. Thus, from a limited consumption series of each user, how to explore users' preferences by balancing various factors becomes another challenge.

To address the challenges mentioned above, in this paper, we propose to study product adoption rate prediction from a multi-factor view. Specifically, we first introduce a factor-based function to capture a user's product adoption rate decision along time. The function

*University of Science and Technology of China,{wule,tanchang}@mail.ustc.edu.cn,{qiliuql,cheneh}@ustc.edu.cn
†Microsoft Research. Xing.Xie@microsoft.com

is flexible enough to generally leverage various factors that may influence the decision. Then, we propose two models to solve this function, the GAM that assumes all users are influenced equally by these factors and the PAM that argues each factor contributes differently among people. Furthermore, to deal with the parameter-tuning process, we extend PAM to a totally Bayesian model (BPAM) that can automatically control model complexity. Finally, we validate our proposed models on two real-world datasets. Extensive experimental results show that our proposed three models could better predict each user's product adoption rate in the future. Meanwhile, the user's personalized preference on various factors can be also well captured.

2 Problem Definition

In this section, we first introduce the formal definition of the product adoption rate prediction problem, and then present the factor-based prediction function.

We study the problem of further product adoption rate prediction of a particular product v for any user i in a social network $G =< U,A >$ given a time horizon $t = 0, 1, 2, ...T-1, T$ with 0 being the start of the time horizon. Here, the node set $U=1, 2, , ..., N$ is users and the edge set A represents the relationship between users. The graph can be either directed or undirected based on the property of the social network. The product v (e.g., new technology innovation, a particular brand or opinion) and the unit of time slice (e.g., quarter, month or week) is application-dependent. For product v, the competing products C_v are a set of products that are competitors of v in the market. These competing products can be easily obtained by the product merchants, and $V = C_v \cup v$ is the product set. E.g, if v refers to the mobile technology to access Internet, then the competing products C_v are the traditional web access technologies. Similarly, if v denotes a smartphone brand (e.g., iPhone), then the product set V contains all the mobile phone brands in the market. After giving the product set, the adoption rate for each user can be defined as:

Definition 1. (**Product Adoption Rate**) *The adoption rate of user i to product v at time slice t, denoted as r_{iv}^t, is the percentage of using v (e.g., usage times) among the whole product set V at t.*

We have $0 \leq r_{iv}^t \leq 1$ and $\sum_{v' \in V} r_{iv'}^t = 1$. Compared to the binary-valued product adoption, this measure captures users' commitments to products more accurately. As in this paper, we focus on predicting the adoption rate of a particular product, we omit the subscript of the product and simply refer r_i^t as r_{iv}^t. We summarize all the users' adoption history into a matrix $\boldsymbol{R}^{N \times T}$, where r_i^t is the element in the i's row and the t's column. Then, the problem we study is defined as:

Definition 2. (**Problem Definition**) *Given a product adoption rate matrix $\boldsymbol{R}$, i.e., the detailed product adoption rate of each user from 0 to T, our goal is to predict the further product adoption rate of each user i at time $T+1$, which is denoted as $\hat{r}_i^{T+1}$.*

Researchers from both marketing science and social science have converged that there are various factors that may determine a user's product adoption [4], such as the user's profile [23], historical preference [3] and the influence from the social network [8]. To generally leverage all the factors that may influence a user's decision, we introduce a adoption rate function as a combination of various factors:

Definition 3. (**Factor-based Adoption Rate Function**) *Given a user i and a set of factors F that can influence a user's product adoption decision, let $p_f(t-1|i)$ denote the propensity of product adoption rate based on factor f ($f \in F$) till time $t-1$ and w_{if} the weight of factor f on user i for the decision-making process. The final predicted product adoption rate $\hat{r}_i^t$ is modeled as:*

$$\hat{r}_i^t = \sum_{f \in F} w_{if} \times p_f(t-1|i) = \boldsymbol{w_i} \times \boldsymbol{p}(t-1|i), \tag{2.1}$$

where $\boldsymbol{w_i}$ is the i's row of weight matrix $\boldsymbol{W}^{\boldsymbol{N} \times |\boldsymbol{F}|}$ and $\boldsymbol{p}(t-1|i)$ is the $|F|$ dimensional vector of $p_f(t-1|i)s'$.

In the above Equation, each user's product adoption rate at t is weighted by various factors in $t-1$. The weight w_{if} can be explained from the following two assumptions. The first one is the **generalized assumption** that presumes *all users are influenced equally by these factors*, i.e., $\forall i, j \in U, w_{if} = w_{jf}$. However, this assumption may be not realistic in practice, as different users may have their own preferences by balancing these factors. E.g., some users are easily influenced by friends while others are unwilling to be swayed by others. Thus, instead of sharing the same weights for all users in the first assumption, we propose the second **personalized assumption** that argues *each user would balance all these parameters based on their own choices*, i.e., w_{if} is personalized and varies among people. For notation convenience, we summarize the $p_f(t-1|i)$ into a tensor $\boldsymbol{P}^{N \times T \times |F|}$, where the element in the $(i, t-1, f)$ position is denoted as $p_f(t-1|i)$. We call $\boldsymbol{P}$ as the adoption rate factor tensor.

Given the factor-based adoption rate decision function, there are still two issues. First, identify key factors influencing a user's product adoption rate. Second, figure out the models that can solve both assumptions effectively and efficiently. We would illustrate these two issues in the following two sections. Table 1 lists the notations used in this paper.

Table 1: Mathematical Notations

Symbol	Description
U	user set, $\|U\| = N$
A_i	the set of users that i follows
$\boldsymbol{x_i^t}$	characteristics of user i at time t
$\boldsymbol{R}^{N\times T}$	the product adoption matrix
r_i^t	an element in $\boldsymbol{R}$, i's adoption rate at time t
$\hat{r}_i^t$	i's predicted adoption rate at time t
$\boldsymbol{P}^{N\times T\times \|F\|}$	the adoption rate factor tensor
$p_f(t\|i)$	the (i,t,f)'s element in $\boldsymbol{P}$, the adoption rate of i based on the factor f till time t
F	the key factors that influence users' decisions
$\boldsymbol{W}^{N\times\|F\|}$	the weight factor matrix
w_{if}	the weight of factor f on user i's adoption rate
$\boldsymbol{w_i}$	the i-th row of $\boldsymbol{W}$, weight vector on user i
$\boldsymbol{w_f}$	the f-th column of $\boldsymbol{W}$, vector on the f's factor

3 Key Factors for Adoption Prediction

In this section, we introduce the selected key factors that may influence a user's decision. In general, the presented factors can be divided into two categories: user-related factors and social-related factors.

3.1 User-related Factors We consider two types of factors, which are the user's historical adoption decisions and the characteristics of her, respectively.

3.1.1 Historical Product Adoption Rate Since user i's product adoption usually changes slowly along time, the elementary factor that influences the current adoption rate r_i^t is the historical rate. For many sequential data prediction tasks, a well understood and widely used theory is the Markov property that assumes the next state depends only on the current state and not on the sequences that precede it [18]. Given this theory, we assume that for each user i, the product adoption rate is closely related to her preference at $t-1$. Then, we can define the first factor, i.e., the historical product adoption rate factor as:

$$p_1(t-1|i) = r_i^{(t-1)}. \tag{3.2}$$

3.1.2 Individual Characteristics A user's adoption behavior is also correlated to her characteristics [23]. Without loss of generality, we denote user i's characteristics (e.g., gender, location, age and number of friends) at time t as $\boldsymbol{x}_i^t$. Then, this factor can be modeled as:

$$p_2(t-1|i) = \boldsymbol{c}' \times \boldsymbol{x}_i^{t-1}, \tag{3.3}$$

where $\boldsymbol{c}$ stores the weights over user characteristics. Here, the coefficient $\boldsymbol{c}$ can be experimentally learned based on the users' adoption history, i.e., by minimizing the negative log-likelihood in Eq. (3.3) as:

$$\min_{\mathbf{c}} L = \sum_{t=1}^{T}\sum_{i=1}^{N}(r_i^t - \boldsymbol{c}'\boldsymbol{x}_i^{t-1})^2 + \lambda_C||\boldsymbol{c}||_{Fro}^2, \tag{3.4}$$

where the first term tries to fit the training data, the second term $||\cdot||_{Fro}$ denotes the Frobenius norm that controls the model capacity. $\lambda_C > 0$ is a regularization coefficient.

3.2 Social-based Factors Besides the inherent user-related factors, people are usually affected by the influences from external social sources. To incorporate such social effects into product adoption predictions, we consider two kinds of widely accepted factors, i.e., the opinions of the social crowd and the direct neighbors' influence, respectively.

3.2.1 Crowd Wisdom 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* [21, 20]. E.g., most of the product advisory websites have a ranking list of *the most popular products* that shows the overall choices of the crowd. Similarly, we use the average product adoption rates among all social entities to define the crowd factor:

$$\forall i, p_3(t-1|i) = \frac{\sum_{j=1}^{N} r_j^{(t-1)}}{N}. \tag{3.5}$$

3.2.2 Neighbor Influence Researchers have converged that social influence, or the impacts created through the neighbors of a user represent an important force affecting users' adoption behaviors [12]. Thus, we also leverage the aggregate statistics of neighbors' product adoption rates to get a neighbor factor:

$$p_4(t-1|i) = \sum_{j\in A_i} t_{ji} \times r_j^{(t-1)}, \tag{3.6}$$

where A_i is the neighbor set of user i. For simplicity, we set the tie strength $t_{ji} = \frac{1}{|A_i|}$.

In summary, we identified four key factors ($|F| = 4$) for users' adoption rate decisions. However, we should note that the proposed factor-based adoption rate prediction function (Definition 3) is general enough and can be easily extended to incorporate many more factors in the future.

4 Adoption Rate Prediction Models

In this section, we propose solutions for the adoption rate prediction function. Specifically, with the extracted factors F, our goal turns to learn the weight matrix $\boldsymbol{W}$ for all users. We present the solutions with respect to both the generalized assumption and the personalized assumption of the weight vector $\boldsymbol{w_i}$ for each user i.

Specifically, given the real adoption rate matrix $\boldsymbol{R}$, the conditional distribution over the observed product adoption rate is assumed to be:

$$p(\boldsymbol{R}|\boldsymbol{W},\boldsymbol{P}) = \prod_{t=1}^{T}\prod_{i=1}^{N}\mathcal{N}(r_i^t|\sum_{f=1}^{|F|} w_{if} \times p_f(t-1|i), \alpha^{-1}). \tag{4.7}$$

where $\mathcal{N}(.|.,.)$ stands for the Gaussian distribution and α is the observational precision.

4.1 GAM: Generalized Adoption Model With the generalized assumption, each user is influenced equally with various factors, i.e., $\forall i,j \in U, w_{if} = w_{jf}$. Based on this assumption, each row of the weight matrix $\boldsymbol{W}$ is the same, enabling it reduces to a vector $\boldsymbol{w}$. Then

the f-th element w_f represents the general weight of factor f for all users. By minimizing the log posterior in Eq.(4.7), $\boldsymbol{w}$ can be learned through:

$$\min_{\boldsymbol{w}} L = \sum_{t=1}^{T}\sum_{i=1}^{N}(r_i^t - \sum_{f=1}^{|F|} w_f \times p_f(t-1|i))^2 + \lambda_G \parallel \boldsymbol{w} \parallel_{Fro}^2, \tag{4.8}$$

where λ_G is a regularization parameter. As the number of dataset records is much larger than the parameters, setting λ_G in a reasonable range (e.g., $[0.001, 100]$) has little impact on the final results. To optimize the generalized loss function in Eq.(4.8), we can explore gradient descent based approach directly. The detailed learning process of GAM is omitted due to its simplicity.

$$\frac{\partial L}{\partial w_f} = \sum_{t=1}^{T}\sum_{i=1}^{N}(\hat{r}_i^t - r_i^t)p_f(t-1|i) + \lambda w_f. \tag{4.9}$$

4.2 PAM: Personalized Adoption Model In this PAM, it is assumed that the factors contribute differently to each user, i.e., w_{if} varies among users. Given this assumption, the weight matrix $\boldsymbol{W}$ is of large size with $N \times F$ elements, thus estimating it solely based on Eq.(4.7) may lead to serious overfitting problem. To deal with this issue, we follow the usual Bayesian approach by placing priors on $\boldsymbol{W}$ [6]. Specifically, for each factor, we assume the corresponding weights in the f-th column of $\boldsymbol{W}$, denoted as $\boldsymbol{w_f}$, follows Gaussian distribution:

$$p(\boldsymbol{w_f}) = \prod_{i=1}^{N}\mathcal{N}(w_{if}|\mu_f, \alpha_f^{-1}) \tag{4.10}$$

It turns out that by combing the priors (Eq.(4.10)) and the likelihood function (Eq.(4.7)) for the personalized adoption model, *m*aximizing *a* log *p*osterior (MAP) is equivalent to minimizing:

$$\min_{\boldsymbol{W}} L = \sum_{i=1}^{N}\sum_{t=1}^{T}(r_i^t - \hat{r}_i^t)^2 + \sum_{f=1}^{F}\frac{\alpha_f}{\alpha}\sum_{i=1}^{N} \parallel w_{if} - \mu_f \parallel_{Fro}^2, \tag{4.11}$$

where $\frac{\alpha_f}{\alpha}$ is a regularization parameter. To optimize the objective function in Eq.(4.11), we could also turn to the gradient descent approaches. To be specific, for each element w_{if} in $\boldsymbol{W}$, we can compute the gradient of each user i on factor f by

$$\frac{\partial L}{\partial w_{if}} = \sum_{t=1}^{T}(\hat{r}_i^t - r_i^t)p_f(t-1|i) + \frac{\alpha_f}{\alpha}(w_{if} - \mu_f). \tag{4.12}$$

Then each element in $\boldsymbol{W}$ can be updated iteratively based on its gradient until the convergence.

Actually, the performance of PAM is tied carefully to the manual tuning of the hyperparameters to avoid overfitting by the MAP estimation. In practice, we need to tune $2|F|+1$ hyperparameters in Eq. (4.12), i.e., $\{\boldsymbol{\Theta_W} = \{\Theta_{W_f}\}_{f=1}^{|F|} = [\mu_f, \alpha_f]_{f=1}^{|F|}, \alpha\}$, on the validation set to get the best performance, which is computationally expensive. To avoid this, similar existing works usually set all parameters to be the same when there are several regularization parameters [15], i.e., $\forall f, \mu_f$ is the same and α_f is the same. However, the performance may be limited due to this simple setting.

4.3 BPAM: Bayesian Personalized Adoption Model To avoid the the carefully tuning of hyperparameters in PAM, we would introduce a totally bayesian treatment of PAM. We further introduce priors for the hyperparameters and maximize the log posterior over both the parameters and the hyperparameters simultaneously, arriving an automatical complexity control given the observation data. We call this solution as *B*ayesian *P*ernsonlized *A*doption *M*odel (BPAM). Specifically, based on the likelihood function (Eq.(4.7)) and the prior distributions (Eq.(4.10)) over the hyperparameter set: $\{\boldsymbol{\Theta_W} = [\Theta_{W_f}]_{f=1}^{|F|} = [\mu_f, \alpha_f]_{f=1}^{|F|}, \alpha\}$, we add conjugate priors for these hyperparameters as:

$$p(\alpha|a,b) = \mathcal{G}(\alpha|a,b),$$
$$\forall f, p(\Theta_{W_f}|\mu_f, \alpha_f) = \mathcal{N}(\mu_f|\mu_0, (\beta\lambda_f)^{-1})\mathcal{G}(\lambda_f|a,b),$$

where $\mathcal{G}(\lambda_f|a,b)$ is the 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 with unknown precision [16]. For convenience, we also define $\Theta_0 = \{\mu_0, a, b\}$. 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 could integrate out all model parameters $\boldsymbol{W}$ and hyperparameters $\{\boldsymbol{\Theta_W}, \alpha\}$, arriving a predictive distribution of future observations. Specifically, the predictive distribution of $\hat{r}_i^t$ is modeled as:

$$p(\hat{r}_i^t|\boldsymbol{R}, \Theta_0) = \int p(\hat{r}_i^t|\boldsymbol{w_i}, \alpha)$$
$$p(\boldsymbol{W}, \boldsymbol{\Theta_W}, \alpha|\boldsymbol{R}, \Theta_0)d(\boldsymbol{W}, \boldsymbol{\Theta_W}, \alpha) \tag{4.13}$$

Since the exact inference of the above predicted distribution is analytically intractable, a variety of approximation models have been proposed. In this paper, we exploit the Markov chain Monte Carlo procedure based on Gibbs sampling [9] to approximate the true posterior distribution of $p(\boldsymbol{W}, \boldsymbol{\Theta_W}, \alpha|\boldsymbol{R}, \Theta_0)$. 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.(4.13) by

$$p(\hat{r}_i^t|\Theta_0) \approx \frac{1}{L}\sum_{l=1}^{L} p(\hat{r^{(l)}}_i^t|\boldsymbol{w_i^{(l)}}, \alpha^l), \tag{4.14}$$

where L denotes the total number of samples and $\boldsymbol{w^l}$ is a vector that samples from the l-th iteration.

Now, we show how to sample the posterior of each latent variable in each iteration. Due to the conjugate priors, the conditional distributions derived from the posterior distribution have the same form as the prior distributions. As to the weight hyperparameters $\boldsymbol{\Theta_W} = \{\Theta_{W_f}\}_{f=1}^{|F|}$, the posterior distribution is estimated as:

$$p(\mu_f, \alpha_f|\boldsymbol{w_f}, \Theta_0) = \mathcal{N}(\mu_f|\mu_f^*, (\beta_f^*\lambda_f^*)^{-1})\mathcal{G}(\lambda_f|a_f^*, b_f^*),$$
$$u_f^* = \frac{\beta u_0 + N\bar{w_f}}{\beta + N}, \quad \beta_f^* = \beta + N, \quad a_f^* = a + \frac{N}{2},$$
$$b_f^* = b + \frac{1}{2}\sum_{i=1}^{N}(w_{if} - \bar{w_f})^2 + \frac{N\beta}{2(N+\beta)}(\bar{w_f} - u_0)^2, \tag{4.15}$$

where $\bar{w_f}$ is the mean of the weight vector $\boldsymbol{w_f}$ of all users' weights for factor f. As to the parameter α, the

posterior distribution follows the Gamma distribution:

$$p(\alpha|\boldsymbol{R},\boldsymbol{W}) = \mathcal{G}(\alpha|a^*,b^*),$$

$$a^* = a + \frac{1}{2}N \times T,$$

$$b^* = b + \frac{1}{2}\sum_{t=1}^{T}\sum_{I=1}^{N}(r_i^t - \hat{r}_i^t)^2. \tag{4.16}$$

For each element of user i's weight vector, i.e., w_{if}, the conditional distribution given other relevant parameters is also Gaussian:

$$\mathcal{N}(w_{if}|u_f^*,[\alpha_f^*]^{-1}) \propto \prod_{t=1}^{T}[\mathcal{N}(r_i^t|\hat{r}_i^t,\alpha^{-1})]\mathcal{N}(w_{if}|\mu_f,\alpha_f^{-1}),$$

$$\text{where} \quad \alpha_f^* = \alpha_f + \alpha\sum_{t=1}^{T}(p_f(t-1|i))^2,$$

$$u_f^* = [\alpha_f^*]^{-1}\alpha\sum_{t=1}^{T}[(r_i^t - \hat{r}_i^t + w_{if}p_f(t-1|i)] + \alpha_f\mu_f. \tag{4.17}$$

In summary, the whole process for Gibbs sampling of the BPAM is shown in Algorithm 1.

Algorithm 1 The Gibbs sampling process of *BPAM*

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

```
Initialize model parameters W with small values.
for l = 0; l <= L; l++ do
    Sample hyperparameter Θ_W^(l) ~ p(Θ_W | W^l, Θ_0).
    Sample hyperparameter α^(l) ~ p(α | R, W, Θ_0).
    for i = 1, ..., N do
        For f = 1, ..., |F|, sample w_if^(l+1) ~ p(w_if | R, Θ_W^l).
    end for
    For each user i, calculate the predicted adoption rate at T.
end for
Return the predicted adoption rate.
```

4.4 Time Complexity All the proposed three models involve iterations. Specifically, in each iteration, the time complexity of all models is $O(T \times N \times |F|)$. Then the total complexity of GAM and BPAM is $O(L \times T \times N \times |F|)$, where L is the number of iterations for convergence. However, for the PAM, we need to try the hyperparameter set to reach convergence, thus the total complexity of PAM is $O(K \times L \times T \times N \times |F|)$, where K is the total number of the different parameter sets. In summary, the time complexity of the proposed three models linearly increase with the userset size, thus they are applicable to real-world production adoption prediction tasks with hundreds of millions of users.

4.5 Factor Preference In both PAM and BPAM, it is assumed that the factors contribute differently for each user. Actually, we can track the personalized factor preference easily after we get the optimal learning results, e.g., the output w_{if} of PAM. Similarly, for BPAM, after the Markov chain Monte Carlo procedure, the factor preference for user i on factor f can be approximated as:

$$w_{if} \approx \frac{1}{L}\sum_{l=1}^{L} w_{if}^{(l)}. \tag{4.18}$$

5 Experiments

In this section, we conduct experiments on real-world datasets to evaluate: (1) The performance of our proposed three models; (2) The effectiveness of each factor; (3) The learned factor preference (i.e., w_{if}) for users.

Table 2: The statistics of the two datasets.

DataSet	Product	Users	Social Edges
MT	Mobile Technology	120,608	3,794,295
SB	iPhone	20,807	530,088

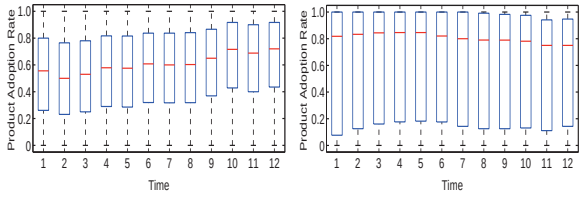
5.1 Experimental Setup It is not easy to find a dataset which records both a massive number of user adoption series in a social network. As an alternative, we refer to an online social media ,i.e., the leading Chinese social network and microblog platform **Weibo.com**, to collect user behaviors and the factor information. When a user posts a message, *Weibo* would forward an enriched message to all of the user's followers, e.g., the message, the time information and the sending device. Here, the sending device presents how a user accesses this platform, i.e., either through the PC client or a mobile device with the brand information (e.g., iPhone). These enriched message streams provide valuable sources to track users' Internet access patterns (mobile access or traditional PC client) and the smartphone brand adoption (the brand information is displayed directly over the mobile access). Thus, we crawled a dataset from Weibo containing 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 the year of 2012. From this original data, we devise two kinds of datasets (product adoption tasks).

Mobile **T**echnology (MT) Dataset. It describes how users would prefer to use the mobile devices (compared to traditional PC client) over time, i.e., for each user i at time t, the MT adoption rate r_i^t refers to the percentage that the user uses mobile devices to access Internet. Here, r_i^t can be computed as the number of messages i posts by mobile devices divided by the total number of messages she sends at that time slice (e.g., one month).

Smartphone **B**rand (SB) Dataset. It depicts users' preference of a particular brand over other smartphone brands. Here, we consider the active users who have used multiple smartphones in our Weibo data. Among all the smartphone brands, we choose iPhone as the product for prediction. As nearly 50% active mobile users have ever adopted iPhone to post messages, while the records for the remaining brands are too sparse to be analyzed for research purpose.

For these two datasets, we treat each month as a time slice, thus every user has 12 adoption rate records among year 2012(one for each time slice). For better

evaluation we only select the active users and filter out the ones that post less than 5 messages at any time slice. The detailed statistics of these two datasets are summarized in Table 2, and Figure 1 displays the boxplot of the adoption rate over time. We could observe that the overall adoption rate changes slowly over time for both datasets, but the variance among users is very large.



(a) MT dataset. (b) SB dataset.

Figure 1: Product Adoption Rate over Time.

Baselines. To compare the performance of our proposed three models, we borrow some baselines from time series analysis and product adoption prediction. For time series analysis, we first adopt the **A**uto **R**egression (AR) model that assumes 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 describes the conditional probability distribution of the future state only depends on the current state, i.e., $\hat{r}_i^{(T+1)} = r_i^T$. We call this baseline as the **N**earest **H**istory (NH). For previous 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 [5, 10]. Here we choose the **C**lassification **a**nd **R**egression **T**ree (CART) [7] that was used in [5] for product adoption prediction. Note that as nearly all of these previous works assume users' adoption preferences are binary values, thus the historical product adoption rates of users are not available in the CART baseline.

Evaluation Metrics. Our goal is to predict users' adoption rate in time T as accurate as possible. Thus, we evaluate the performance of each approach by calculating the *R*oot *M*ean *S*quared *E*rror (RMSE) as: $RMSE = \sqrt{\frac{\sum_{i \in U}(\hat{r}_i^{(T+1)} - r_i^{(T+1)})^2}{N}}$ [15]. The smaller the RMSE value, the better performance of the model.

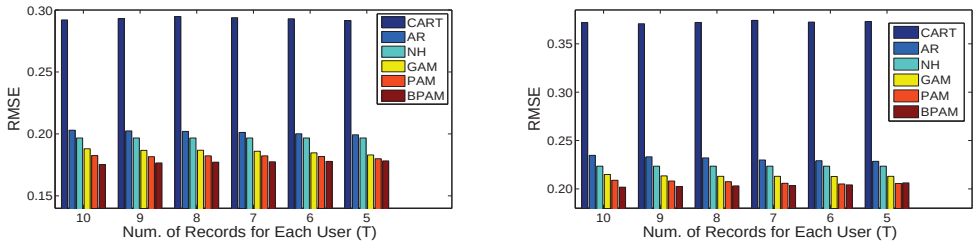
Besides, one of the most important applications of the product adoption rate prediction is the targeted marketing, i.e., identifying a small group of customers that are highly likely to adopt this product. To evaluate this ranking performance, we select the 10% of users that have the largest product adoption rates as the candidate targeted userset P_u and the remaining users as the negative userset N_u ($\forall i \in P_u, j \in T_u, r_i^{(T+1)} > r_j^{(T+1)}$. Then we use the *D*egree *o*f *A*greement (DOA) measure to calculate the percentage of user pairs that are correctly ranked with respect to these two usersets [14]. Here, the DOA measure is defined as:

$$DOA = \frac{\sum_{i \in T_u, j \in N_u} \delta(\hat{r}_i^{(T+1)} - \hat{r}_j^{(T+1)})}{|T_u| \times |N_u|}, \tag{5.19}$$

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

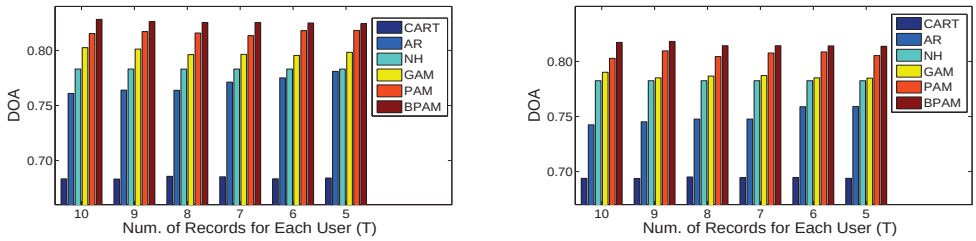
For tuning the parameters in the baselines and our proposed models, we use the adoption rate records from 1 to $T-1$ for training and the data in T for validation. We tune all the parameters in the validation dataset to ensure the best performance and omit the detailed setting of these models for space constraint. Specifically, in our proposed PAM, the μ_fs are set to be the values from GAM and the α_f values are tuned to have the best performance on the validation data. For BPAM, the parameters (Θ_0) are set with small values without tuning. All the experiments are performed on a 2.3GHZ4-Core CPU with 8G main memory PC and the programs are implemented in C++.

5.2 Performance Comparison In this subsection, we show the performance comparison of prediction models. As we take a sequence of users' adoption history for future prediction, an important parameter in our model is the T value, i.e., the number of records of each user's adoption history. In both datasets, we have at most 12 records for each user, among them the last two records are used for validation and test, respectively. Then each user has at most 10 records ($T = 10$) for training. We plot the the overall RMSE performance of each method with different history length values in these two datasets in Figure 2, where the T value ranges from $[10, 5]$ with a decrease of 1. We observe that the overall performance of different models are the same for these two datasets, where our proposed three models always outperform the baselines. Among the baselines, the CART performance changes slowly with different T values as the historical adoption rate of the user is neglected in this model. Similar trends can be found on AR as the adoption history too long ago does not have an impact on a user's current decision. Thus the NH model, i.e., only taking the previous behavior, always performs better than the AR model. As to our own three models, the BPAM always performs the best, followed by PAM and GAM. Based on the observation, we conclude that GAM helps to model the decision process of users by leveraging multiple factors. Further considering the personalized influence weights of users better captures the uniqueness of users, thus PAM improves over GAM. By automatically learning the parameters in PAM from a Bayesian perspective, this BPAM model has better performance than PAM. E.g, when $T = 10$, the average improvement of BPAM, PAM and GAM over the best baseline (i.e., NH) are about



(a) MT dataset. (b) SB dataset.

Figure 2: The Overall RMSE Comparison.

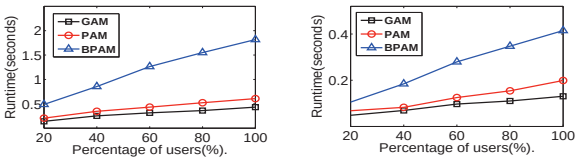


(a) MT dataset. (b) SB dataset.

Figure 3: The Overall DOA Comparison.

4%, 7% and 10% respectively. Nevertheless, when the T value begins to decrease, the BPAM performance also decreases slowly. A possible reason is that decreased training records limit the performance of BPAM.

The overall DOA values of different models are shown in Figure 3. We can observe the similar trends as the RMSE measure, i.e., our proposed models outperform the baselines to a large extend. On average, our proposed models can improve 6% to 8% over the best baseline of both datasets. Based on the above two evaluation metrics, we conclude that our proposed models have better overall performance than the baselines.



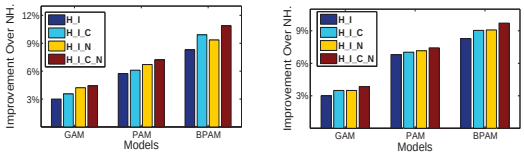
(a) MT dataset. (b) SB dataset.

Figure 4: The Runtime of Each Iteration.

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%). Figure 4 shows the running time of each iteration with respect to all proposed models of the two dataset. We can see that all the models are very fast. After selecting the relevant features, the average time is less than 2 seconds with more than 100 thousand users. Generally, the GAM costs the least time, followed by the PAM and the BPAM as the latter two models need to calculate the personalized factor preference for each user. Though BPAM is more time consuming, it has the same time complexity with GAM as illustrated in Section 4.4. Meanwhile, the computation times are almost linear with the size of the users for all proposed models. In summary, our approach is fast enough to be applied to the real-world production adoption prediction tasks.

5.3 Effectiveness of the Factors Here, we analyze the effect of different factors underlying people's adoption decisions. We have introduced four factors: the historical rate (H), the individual characteristics (I), the crowd (C) wisdom and the neighbor (N) influence. E.g., H_I denotes considering the historical rate and the individual factor. Figure 5 shows the relative gain of these factors compared to the best baseline, i.e., the N-H model. From this figure, we find that all these factors can improve the final prediction results to some extent with regard to the three proposed models. In particular, there are about 3% to 6% improvement by using both the historical and individual factors. These results imply a big impact of these two user-related factors for product adoption rate prediction. Moreover, both the crowd wisdom and neighbor influence generate a relative 1% improvement. Further considering both of these two social factors give about 2% relative improvement over the user-related factors, indicating the social related factors can complement the user-related factors to some extent. In summary, all the chosen four factors contribute to the final prediction task, and the user-related factors give more improvement than the social-related ones.

5.4 Personalized Factor Preference As shown in previous empirical studies, both PAM and BPAM out-



(a) MT dataset. (b) SB dataset.

Figure 5: The Relative Improvement over The Factors.

perform the GAM, and thus we could conclude that it is more reasonable to assume the factors contribute differently among users. In this subsection, we would visualize the learned factor weights of several typical users to get a more straightforward observation. Specifically, we first calculate the personalized weight with each factor for each user by Eq.(4.18) of the BPAM model [1]. Then, we depict the normalized factor weights of several typical users in Figure 6. For comparison, we also visualize the learned weight w.r.t to GAM in the left most part, where the factor weights for all users are the same. From the results of GAM, the historical factor plays an important role for all the users' adoption rate prediction, accounting about 80% to 90% of the total contributions. However, the actual preference to each factor varies among users, i.e., some users may be influenced easily by the crowd factor (e.g., 60% in the second leftmost of Figure.6(a)) while others are more likely to be influenced by the individual characteristics (e.g., 51% in the rightmost of Figure 6(a)). 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) can help capture the personalized aspects in decision-making.

6 Related Work

Generally, the related work of our research could be classified into the following three categories.

The first category is about the product adoption prediction. A typical task in this area is to predict the future adoption probabilities for non-adopters till now. Generally, nearly all these models relied on extracting features from individual characteristics [23], the social network [12, 8] or the hybrid of the above two [10, 5]. Then various traditional classification models can be further utilized to solve the binary prediction task after feature engineering. Different from these works, we characterize users' adoption status as a rate (i.e., in the range of [0,1]) rather than the binary buy-or-not values, thus our proposed models could easily capture more factors underlying people's adoption decisions. Moreover, we introduce both generalized assumption and personalized assumption to balance these factors in users' decision making process, whereas the previous works seldom modeled the personalized aspects.

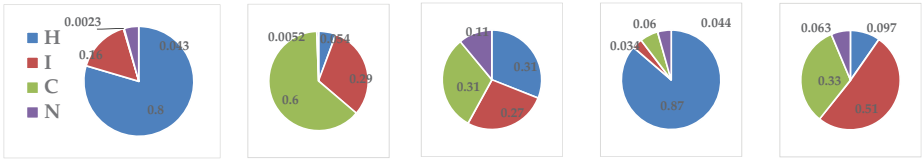
If regarding the product adoption prediction as a recommendation task with the goal to provide personalized services for users, various recommendation models are also related to our task. The recommendation models can be grouped into two categories: the collaborative filtering models [19, 15, 24] that solely relied on users' historical preference to generate recommendations and the content-based models [17] that explored users' profiles for recommendation. Recently, the social based recommendation has emerged to boost recommendation accuracy by leveraging social network structure among users [11]. Borrowing ideas from these works, our proposed models explicitly integrate both the historical preference, the users' profiles and the social network in the factor construction process. Meanwhile, our models are flexible to leverage more factors for product adoption prediction.

Our work is also closely related to time-series forecasting, which tried to capture the internal structure of data over time and usually have many forms. Among them, the moving average (MA)[1] modeled the future state as an average smoothing of previous sequences. The autoregressive model (AR) [3] extended MA by assuming the current value linearly depends on its previous values. Some superior models, such as the hidden markov model [18] and the conditional random fields [13], are assumed to be a Markov process with unobserved hidden states. However, these models could not work well for the product adoption task. Different from these works, we are interested in predicting future adoption rate of users, which depends heavily on the decision-making process of users underlying various factors around them. Meanwhile, users' personalized preferences for balancing these factors also plays an important role for final adoption, which restrains the traditional time series forecasting techniques. Therefore, in this paper, we introduce a decision function to leverage these factors for product adoption rate prediction.

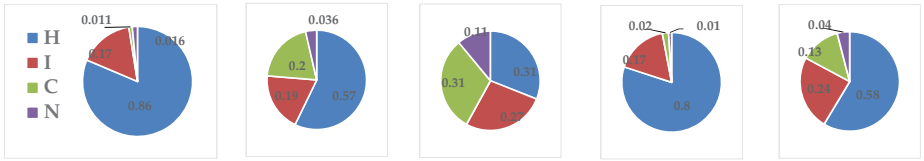
7 Conclusions and Future Work

In this paper, we studied the problem of product adoption rate prediction based on the historical adoption series. We first introduced a factor-based decision function that could leverage various factors underlying users' adoption decisions. Then, we proposed two solutions, i.e., GAM and PAM, based on the generalized assumption of the factor weights and the personalized assumption respectively. Furthermore, we extended PAM to a total Bayesian approach to achieve automatically complexity control. Though we described this work with only four factors, all the approaches could be easily extended to leverage more factors. Finally, experiments on

[1] Similar results could be observed by PAM.



(a) MT dataset.



(b) SB dataset.

Figure 6: Case study of the learned weights of typical users. The leftmost depicts the learned weights w.r.t. GAM.

real-world datasets showed the superiority of our proposed models with about 10% precision improvement compared to the baselines.

In the future, we will incorporate some other factors for more effective and efficient product adoption modeling. Moreover, we would like to apply the discoveries (e.g., the personalized factor preferences) to several real-world applications, e.g., personalized product recommendation and customer churn prediction.

8 Acknowledgements

This research was partially supported by grants from the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61325010), the Natural Science Foundation of China (Grant No. 61403358), the Fundamental Research Funds for the Central Universities of China (Grant No. WK0110000042) and the Anhui Provincial Natural Science Foundation (Grant No. 1408085QF110). Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association, CAS.

References

[1] The moving average model. http://en.wikipedia.org/wiki/Moving_average.

[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 TKDE*, 17(6):734–749, 2005.

[3] H. Akaike. Fitting autoregressive models for prediction. *AISM*, 21(1):243–247, 1969.

[4] G. Armstrong, M. H. Cunningham, and P. Kotler. *Principles of marketing*. Pearson Australia, 2012.

[5] R. Bhatt, V. Chaoji, and R. Parekh. Predicting product adoption in large-scale social networks. In *CIKM*, pages 1039–1048. ACM, 2010.

[6] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[7] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[8] X. Fang, P. J.-H. Hu, Z. Li, and W. Tsai. Predicting adoption probabilities in social networks. *Information Systems Research*, 24(1):128–145, 2013.

[9] A. Gelfand and A. Smith. Sampling-based approaches to calculating marginal densities. *JASA*, 85(410):398–409, 1990.

[10] S. Guo, M. Wang, and J. Leskovec. The role of social networks in online shopping: information passing, price of trust, and consumer choice. In *EC*, pages 157–166. ACM, 2011.

[11] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, pages 135–142. ACM, 2010.

[12] J. Kleinberg. Cascading behavior in networks: Algorithmic and economic issues. *Algorithmic game theory*, 24:613–632, 2007.

[13] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[14] Q. Liu, E. Chen, H. Xiong, C. H. Ding, and J. Chen. Enhancing collaborative filtering by user interest expansion via personalized ranking. *IEEE TSMCB*, 42(1):218–233, 2012.

[15] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[16] K. P. Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, 1(2σ2):16, 2007.

[17] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[18] L. Rabiner and B.-H. Juang. An introduction to hidden markov models. *ASSP Magazine*, 3(1):4–16, 1986.

[19] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM, 2001.

[20] J. Surowiecki. The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business. *Economies, Societies and Nations*, 2004.

[21] J. Surowiecki, M. P. Silverman, et al. The wisdom of crowds. *AJP*, 75(2):190–192, 2007.

[22] W. Verbeke, D. Martens, C. Mues, and B. Baesens. Building comprehensible customer churn prediction models with advanced rule induction techniques. *ESWA*, 38(3):2354–2364, 2011.

[23] Y. Zhang and M. Pennacchiotti. Predicting purchase behaviors from social media. In *WWW*, pages 1521–1532, 2013.

[24] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian. Mining mobile user preferences for personalized context-aware recommendation. *ACM TIST*, 2013.