Mobile Networks and Applications (2020) 25:1233–1248
https://doi.org/10.1007/s11036-020-01535-1

Mining consuming Behaviors with Temporal Evolution for Personalized Recommendation in Mobile Marketing Apps

Honghao Gao [1] · Li Kuang [2] · Yuyu Yin [3] · Bin Guo [2] · Kai Dou [2]

Published online: 6 June 2020

Abstract
Recently, more and more mobile apps are employed in the marketing field with technical advances. Mobile marketing apps have become a prevalent way for enterprise marketing. Therefore, it has been an important and urgent problem to provide personalized and accurate recommendation in mobile marketing, with a large number of items and limited capability of mobile devices. Recommendation have been investigated widely, however, most existing approaches fail to consider the stability or change of users' behaviors over time. In this paper, we first propose to mine the periodic trends of users' consuming behavior from historical records by KNN(K-nearest neighbor) and SVR (support vector regression) based time series prediction, and predict the next time when a user re-purchases the item, so that we can recommend the items which users have purchased before at proper time. Second, we aim to find the regularity of users' purchasing behavior during different life stages and recommend the new items that are needed and proper for their current life stage. In order to solve this, we mine the mapping model from items to user's life stage first. Based on the model, users' current life stage can be estimated from their recent behaviors. Finally, users will be recommended with new items which are proper to their estimated life stage. Experimental results show that it has improved the effectiveness of recommendation obviously by mining users' consuming behaviors with temporal evolution.

Keywords Recommender system · Time series prediction · Temporal E-volution · Mobile marketing apps

1 Introduction

With the wide adoption of mobile devices as well as technological developments, mobile apps have been used widely in our daily lives. In the field of enterprise marketing, mobile marketing has become a prevalent way. As the statistical data [1] tells, mobile marketing is the top priority task for technical users, since it has shown a high ROI; 71% of salesmen believe that mobile marketing is the core of the business; 58% of the companies involved in the survey have a specialized mobile marketing team, and they think having a team of people who are experts in mobile marketing can achieve maximum benefits from marketing efforts. 83% of B2B salesmen say that mobile apps are important to marketing and they plan to spend more money on developing their own marketing apps.

On one side, people enjoy the colorful and convenient life with rich items and information online; on the other side, people are feeling lost in such an online world. They find it very hard to get the required items which they are really interested in quickly and effectively. Therefore, recommender system is proposed to predict users' preference on items. Recommender systems can be roughly divided into three classes: content-based [2, 3], collaborative filtering [4–9] and context-aware approaches [10–15]. Content-based filtering approaches generate recommendations by comparing item descriptions with users' profiles. Collaborative filtering approaches predict users' preference from that of similar users, by mining a large amount of historical records. Whereas context-aware approaches consider the effect of context factors, such as user's profile (gender, age, profession and etc.) and natural situation (location, time and etc.), on users' preference, demand, and the selection and definition of neighbors.

Market researchers have been aware of the correlation between temporal factors and users' behaviors and begin to

✉ Li Kuang
kuangli@csu.edu.cn

[1] Computing Center, Shanghai University, Shanghai, China

[2] School of Computer Science and Engineering, Central South University, Changsha, China

[3] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China

employ their correlation to recommender system. There are various kinds of representation forms of such correlation. In this paper, in the context of marketing apps for maternal and child, we mainly consider two kinds of correlations: First, there exists a certain temporal frequency to buy fast-moving consumer goods of life kind, such as maternal vitamins, diapers, milk powder and body shampoo, therefore, we aim to mine the periodic regularity of purchasing a piece or a category of items by a user, and then for the items which the user has bought before, we can predict the next time when he will purchase it again. Second, there exists a certain temporal evolution regularity for parental population, as Figure 1 shows, a pregnant woman may buy folic acid and maternity dress. After the baby is born, they may buy milk powder and diapers. Later when the baby grows up, they may buy toys and books. Therefore, we aim to mine such temporal evolution regularity from the purchasement of several kinds of items by a category of similar users, and then we can estimate users' current life stage, and then recommend new items that are proper with his life stage.

In this paper, we propose a recommender system by mining consuming behaviors with temporal evolution. We consider two kinds of temporal factors in consuming behavior: periodic regularity of purchasement and demand evolution based on life stage. For mining the periodic regularity of purchasement, first, time series data is generated by ranking the time intervals of every two adjacent records of a user purchasing an item or a kind of items in chronological order. And then a time window is proposed for cutting the time series data into multiple feature vectors as training dataset. Next, a KNN [16] method is utilized to reduce the training set by selecting the top *k* feature vectors which have a high similarity with the input feature vector. Finally, we can predict the time interval of next purchasement by the user based on SVR [17] model. For life stage based recommendation, the system first mines the mapping model from items to user's life stage based on multiple-outlier detecting approach. Next, based on the model, users' current life stage can be estimated from their recent behaviors by dynamic allocation weight algorithm. Finally, new items are recommended to users based on UBN (unweighted bipartite networks) model [18] and Bayesian model [19]. Extensive experiments are conducted with offline data in the mum-baby domain provided by TIAN-CHI platform, and the experimental results show the effectiveness of our proposed recommender system.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes problem definition and approach overview. Section 4 presents periodic regularity of purchasement based recommendation. Section 5 illustrates the life stage based recommendation. Section 6 analyzes the experimental results. And finally, Section 7 gives conclusions and future work.

2 Related work

This section presents a brief review on three topics relevant to our study: traditional collaborative filtering, recommendations based on sequential behaviors and series prediction.

2.1 Traditional collaborative filtering

Collaborative Filtering (CF) is currently the most popular information filtering technology in recommender system, which generates personalized recommendations based on users' historical data. In the last few years, with the rapid growth of e-commerce, CF has become a research hotspot and many large e-commerce sites have used CF as an essential tool in their recommender systems, such as recommending books at Amazon [20], news at Google [21], movies at Yahoo [22], and CDs at Netflix [23]. Traditional collaborative filtering

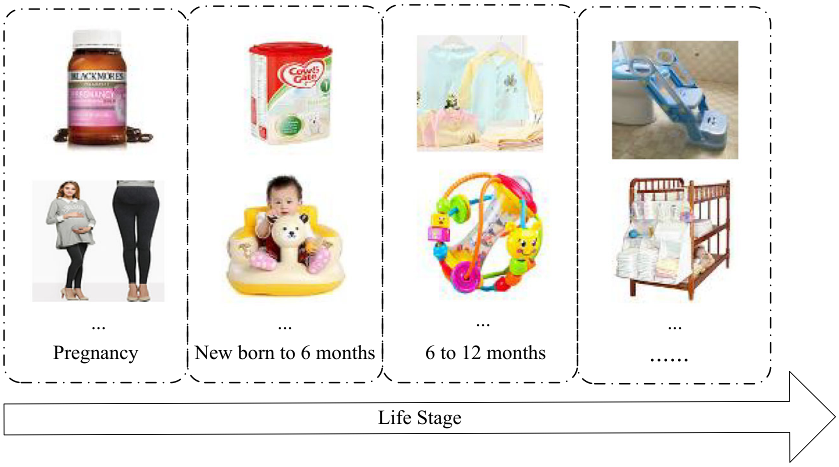


Fig. 1 An example in mum-baby domain

methods analyze a great deal of information on users' behaviors, activities or preferences and predict what users would like based on their similarity to other users. As collaborative filtering methods only rely on the information about user interactions and do not require the content information of items or user profiles, they are widely used in recommender system and studied by many researchers. These methods filter or evaluate items through the opinions of other users [24], which is based on a hypothesis that the user will prefer the items which the user's similar user preferred in the past [25].

In the literature, collaborative filtering methods can be mainly classified into three types. They are model-based, memory-based and hybrid methods respectively. Firstly, model-based methods generate recommendations by training a model. Algorithms of this type include the matrix factorization [26, 27], the graph-based approaches [28], etc. Secondly, most memory-based methods is first to obtain a set of neighbor users for a particular user according to the entire previously rated items by the users. After obtaining the set, the recommendations for the particularly user are generated based on the items that neighbor users like. These methods are known as user-oriented memory-based methods. Moreover, an analogous procedure, which builds item similarity groups using purchasing history, is known as item-oriented memory-based collaborative filtering [29]. At last, hybrid methods combine two or more types of collaborative filtering techniques to generate recommendations. They are helpful to address sparsity problem, in which external information can be used to predict for new users or new items. In 2004, Zhou D et al. [30] propose a hybrid collaborative filtering approach, which exploits bulk taxonomic information designed for exact product classification to address sparsity problem, based on the generation of profiles via inference of super-topics core and topic diversification. In 2002 Schein et al. [31] propose the aspect model latent variable method for cold start recommendation, which combines both collaborative and content information in model fitting.

2.2 Recommendations based on sequential behaviors

In recent times, several studies have highlighted the effect of time in recommender system [32–39]. In [32], Koren et al. take time information into account and propose the Time SVD++ algorithm to address interests drift problem. In [33], Khoshneshin et al. assign users (products) to different clustering dynamically based on evolutionary co-clustering, which is ready for further recommendation. In [34], it has been shown that user's historical ratings focus only on one or more aspects of user interest spanning a particular time period. According to this finding, Li et al. thus propose a cross-domain CF framework, which can track user's interests drift and effectively recommend. In [35], Ren et al. show that user preference patterns and preference dynamic effect has barely been exploited in the existing recommendation system. They thus formalize the user's preference as a sparse matrix, and use subspace to iteratively model the personalized and global preference pattern. In [36], Xiang et al. propose a Session-based Temporal Graph to simultaneously model users' long-term and short-term preferences over time. In [37], Rendel et al. present a tensor factorization modeling algorithm bringing both matrix factorization and Markov chain together for next-basket recommendation. In [38], Wang et al. adapt the proportional hazards modeling approach in survival analysis and explicitly incorporate time to recommend a particular product at a particular time. In [39], Peng Jiang et al. propose a new Maximum Entropy Semi Markov Model to segment and label consumer life stages based on observed purchasing data over time.

2.3 Time series prediction

Time series prediction is gaining popularity as a means of studying regularity that data changes with time. Generally, there are two modeling types for time series prediction. They are global modeling and local modeling. The former constructs a model which is independent from user queries, while the latter constructs a model for each different query from the user. Both Wu S F et al. [40] and Martínez-Rego D et al. [41] identified that local modeling is beneficial to improve estimation performance for time series prediction through multiple experiments. In many real-world applications where time series prediction techniques have been used, both Sapankevych N I et al. [42] and Thissen U et al. [43] demonstrate that the SVM outperforms the autoregressive moving average (ARMA) model and in most cases outperforms the best of several Elman neural networks by a comparison of these three methods based on their predicting ability. Huang Z et al. [44] proposed a k-nearest neighbor based least squares support vector machine (LS-SVM) framework. By selecting similar instances (i.e., nearest neighbors) in the training dataset for each testing instance based on the KNN approach, the complexity of training an LS-SVM regressor is reduced significantly and the prediction horizon of SVM is improved by taking into account the KNN. In time series prediction, even if the one-step prediction model is very exact, the iteration procedure would accumulate prediction errors when repeating one-step-ahead prediction, which results in bad prediction performance. Zhang L et al. [45] deals with iterated time series prediction problem by using multiple SVR models, which are trained independently based on the same training data with different targets. In other words, the n-th SVR model performs an n-step-ahead prediction.

Different from the above relevant studies, this paper proposes a new e-commerce recommender system based on mining consuming behavior with temporal evolution, namely periodic regularity of purchasement and demand evolution based on life stage. We have investigated the life stage based recommendation in [46], and now we extend the correlation between temporal evolution factors and users' consuming behavior with one more kind, i.e., periodic regularity of purchasement. We thus use proposed recommender system to recommend items for users.

3 Problem definition and approach overview

We present the problem definition and an overview of our proposed e-commerce recommender system in this section.

3.1 Problem definition

The notations to be used in the rest of this paper are:

1. U: the set of users.
2. I: the set of items.
3. $x_{u,\ i,\ t} = <u,\ i,\ t>$ is a 3-tuple denoting the purchase behavior of a user, where u is the user, i is the purchased item and t is the purchased time.
4. $x_{u,i}(n) = x_{u,i,t_{n+1}} - x_{u,i,t_n} = < u, i, t_{n+1} > - < u, i, t_n >$ is the time interval of two adjacent records for user u purchasing item i, and the unit of $x_{u,\ i}(n)$ is day.
5. $s_{u,t}$: the life stage of user u at time t.
6. Periodic regularity of purchasement: an obvious or potential habit that a user purchases an item.
7. Life stages: a predefined set of phases in life, and each life stage spans a period of time.

The main problems in paper are:

1. The first problem is, given the intervals of user u purchasing item i $\{x_{u,\ i}(1), \ldots x_{u,\ i}(n)\}$, we aim to investigate the periodic regularity of user u purchasing item i, so as to predict $x_{u,\ i}(n + 1)$, i.e., the time interval of next purchasement.
2. The second problem is, given all the purchasing records of user u buying various items at various time, we aim to predict the current life stage of the user u from a set of $s_{u,t}$, and then recommend to user u the items that are proper with his current life stage.

3.2 Approach overview

Figure 2 show the architecture of our proposed recommender system based on mining consuming behaviors with temporal evolution, including recommendation based on periodic regularity mining and that based on user's life stage.

Three main steps compose the solution to the first problem. First, in order to get training feature vectors and input feature vector, we divide each sample data into multiple subsequences with a fixed-size time window. Second, KNN is used to select the final training data from candidate feature vectors. Finally, the SVR model is trained and obtained according to the training data and it is used to predict the time interval of next purchasement. We can then recommend the user to buy the item again after the predicted interval.

Three main steps compose the solution to the second problem. First, we convert user's age information to a corresponding life stage which is defined by experts, and then count and analyze all purchase behaviors of users with age information, to get a mapping model which can tell the probabilities of users at each life stage when given a purchase behavior. Second, based on the mapping model, we propose an algorithm to predict user's current life stage if his purchase records are without age information. Finally, we can recommend users new items that are proper to his life stage.

4 Recommendation based on periodic regularity mining

In this section, we mainly consider to recommend users the items with periodic regularity of purchasement.

We first calculate the time intervals of every two adjacent records of a user u purchasing an item or a kind of items namely i, i.e. $\{x_{u,\ i}(1), \ldots x_{u,\ i}(t)\}$. We can deem the set of time intervals as an array, and show it in a picture. For example, Fig. 3 shows 4 sets of time intervals for different $<u,\ i>$ tuples, where x-axis denotes the array index, and y-axis denotes each time interval. From the upper two pictures of Fig. 3, we can see a straight line and a repeat wave line respectively, and we can easily conclude that there is obvious periodic regularity, however, we cannot see an obvious periodic regularity in the lower two pictures of Fig. 3.

In our dataset, we find that there is periodic regularity in most of purchasing records by statistical analysis. To judge whether there is periodic regularity in a set of time intervals, we transform the problem into a time series prediction problem. Recently time series analysis, especially on time series prediction, has gained much attention of many researchers. In order to solve the time series prediction, many methods have been applied to obtain a nonlinear and dynamic model, such as SVR, K nearest neighbor regression (KNNR). In this paper, we choose KNN + SVR after comparing two methods in the experiment.

Figure 4 shows the solution of to the first problem. The goal is to get the value at time t + 1, namely $x_{u,\ i}(t + 1)$.

4.1 Time window based data grouping

Consider a time series $\{x_{u,\ i}(1), \ldots x_{u,\ i}(l)\}$, where l is the length of the time series. As Fig. 5 shows, the SVR model that

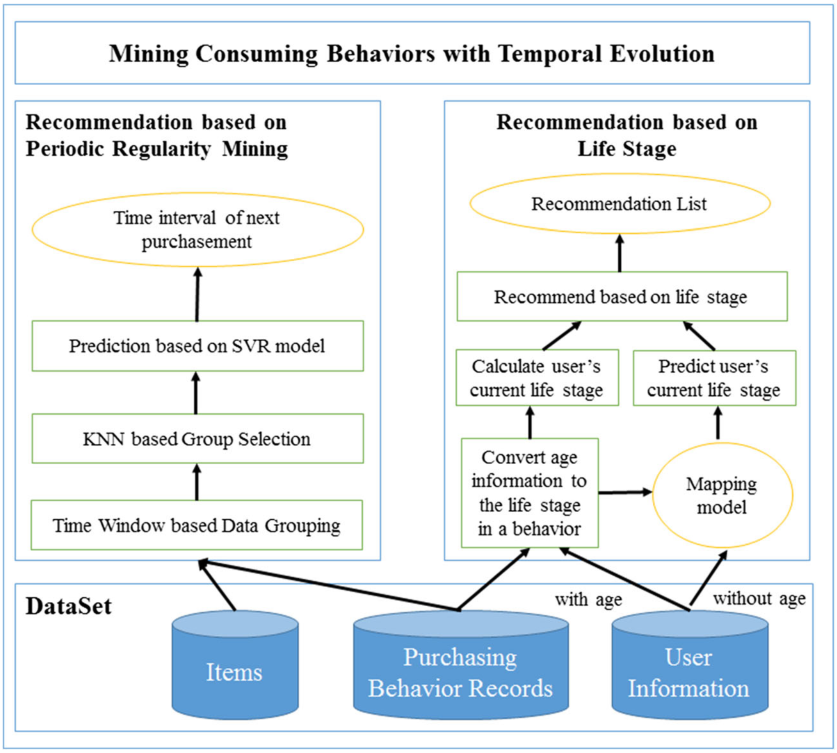

Mining Consuming Behaviors with Temporal Evolution
Recommendation based on Periodic Regularity Mining
Time interval of next purchasement
Prediction based on SVR model
KNN based Group Selection
Time Window based Data Grouping
Recommendation based on Life Stage
Recommendation List
Recommend based on life stage
Calculate user's current life stage
Predict user's current life stage
Convert age information to the life stage in a behavior
Mapping model
DataSet
with age
without age
Items
Purchasing Behavior Records
User Information

Fig. 2 The framework of temporal consuming behavior based recommendation

depends on the past d points has the form

$$x_{u,i}(l+1) = f\left(x_{u,i}(l), x_{u,i}(l\text{–}1), \ldots x_{u,i}(l\text{–}d+1)\right), \qquad (1)$$

where d is the size of time window.

For a time series prediction problem, it is necessary to divide the time series $\{x_{u,\ i}(1), \ldots x_{u,\ i}(l)\}$ to be samples which can be accepted by a SVR model. Thus let $\mathbf{x}_{j-\mathrm{d}+1} = [x_{u,\ i}(j), x_{u,\ i}(j-1), \ldots, x_{u,\ i}(j-d+1)]$ and$y_{j-\mathrm{d}+1} = x_{u,\ i}(j+1)$, where $d \le j \le l-1$. Each time the time window moves a step forward, we can get a piece of training data $(\mathbf{x}_{j-\mathrm{d}+1}, y_{j-\mathrm{d}+1})$, and finally we can get the training set $\cup_{j=d}^{l-1}\left\{\mathbf{x}_{j\text{–d}+1}, y_{j\text{–d}+1}\right\}$.

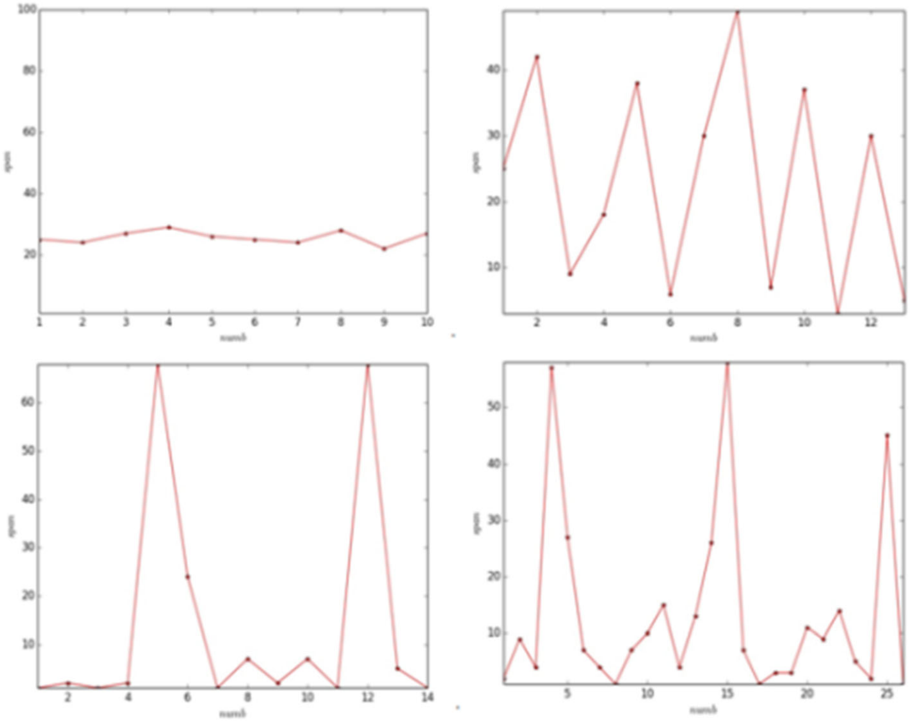


Fig. 3 Some examples of periodic regularity of purchasement

Moreover, in order to get $y_{l-\mathrm{d}+1} = x_{u,\ i}(l+1)$, the input testing feature vector is $\mathbf{x}_{l-\mathrm{d}+1}$. The d influences the final results and we will determine the optimal d in the experiment.

4.2 KNN based group selection

In the previous step, there are $l-d$ pieces of data in the training set. If l is too larger, the training time will be long and much noise data may be considered. Therefore in the second part, KNN is used to reduce the training dataset by selecting k feature vectors which are the most similar to the input testing feature vector $\mathbf{x}_{l-\mathrm{d}+1}$ from the candidate training dataset. Cosine similarity is used to calculate the similarity between the feature vectors.

Assume that the training sample set is $\cup_{j=d}^{l-1}\left\{\mathbf{x}_{j-\mathrm{d}+1}, y_{j-\mathrm{d}+1}\right\}$ and testing input feature vector is $\mathbf{x}_{l-\mathrm{d}+1}$. We first calculate its cosine similarity with each candidate feature vector $\mathbf{x}_{j-\mathrm{d}+1}$, denoted as:

$$\cos\theta = \frac{\mathbf{x}_{l-\mathrm{d}+1} \cdot \mathbf{x}_{j-\mathrm{d}+1}}{\left\|\mathbf{x}_{l-\mathrm{d}+1}\right\|\left\|\mathbf{x}_{j-\mathrm{d}+1}\right\|} \tag{2}$$

The feature vectors with the k largest cosine similarities are taken to be the nearest neighbors of the input feature vector. For example, suppose $\{x_{u,\ i}(1), \ldots x_{u,\ i}(l)\} = (58, 51, 126, 2, 51, 65)$ is a time series records. The size of time window is 3. Then we can get (58, 51, 126), (51, 126, 2), (126, 2, 51) as candidate training dataset. And the input feature vector is (2, 51, 65), we can get the cosine distances between input feature vector and candidate training vectors are 0.89, 0.59, 0.33. If $k = 2$ in KNN, we choose (58, 51, 126) and (51, 126, 2) as the final training feature vectors for the next step.

4.3 Time series prediction based on SVR

After obtaining the k training feature vectors, it is assumed that $\cup_{i=1}^{k}\{\mathbf{x}_i, y_i\}$ is the training sample set, and k denotes the number of samples. Given a set of training samples, SVM aims to find the optimal function from the set of hypothesis functions.

$$\left\{f \middle| f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b, \mathbf{w} \in \mathrm{R}^{\mathrm{d}}, b \in \mathrm{R}\right\} \tag{3}$$

where $\mathbf{w}$ is the weight vector and b is the bias term. In order to obtain weight vector $\mathbf{w}$ and bias term b, we must minimize the sum of the ε-insensitive loss function:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{\mathrm{i}=1}^{\tau} L(y_{\mathrm{i}}, f(\mathbf{x}_{\mathrm{i}})) \tag{4}$$

where $C > 0$ is the regularization factor, $\|\cdot\|$ denotes the 2-norm, and $L(\cdot,\cdot)$ is the loss function. SVR has different formulations when using different loss functions. Generally, the ε-insensitive loss function is used in SVR and has the form

$$L(y, \mathrm{f}(\mathbf{x})) = \begin{cases} 0, & \frac{|f(\mathbf{x}) - y| \le \varepsilon}{\text{otherwise}} \\ |f(\mathbf{x}) - y| - \varepsilon, & \end{cases} \tag{5}$$

We can convert formula (4) into an equivalent formula (6) by introducing slack variables ξ_i and ξ_i^*:

$$\frac{\min\limits_{\mathbf{w},\mathrm{b}} \frac{1}{2}\|\mathbf{w}\|^2 + \mathrm{C}\sum\limits_{i=1}^{\tau}\left(\xi_i + \xi_i^*\right)}{\begin{array}{l}\text{s.t. } \mathbf{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}} + \mathrm{b} - y_i \le \varepsilon + \xi_i \\ \quad y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i - \mathrm{b} \le \varepsilon + \xi_i^* \\ \quad \xi_i, \xi_i^* \ge 0, i = 1, \ldots, \tau\end{array}} \tag{6}$$

It is not easy to seek the solution of formula (6) because of high dimensional-feature space. So we convert it into the dual problem via a kernel function:

$$\begin{array}{l}\min\limits_{\alpha,\alpha^*} \frac{1}{2}\sum\limits_{i=1}^{\tau}\sum\limits_{j=1}^{\tau}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)k\left(\mathbf{x}_{i,}\mathbf{x}_j\right) - \sum\limits_{i=1}^{\tau}\left(\alpha_i - \alpha_i^*\right)y_i + \varepsilon\sum\limits_{i=1}^{\tau}\left(\alpha_i + \alpha_i^*\right) \\ \text{s.t.} \quad \sum\limits_{i=1}^{\tau}\left(\alpha_i - \alpha_i^*\right) = 0 \qquad 0 \le \alpha_i, \alpha_i^* \le \mathrm{C}, \quad \mathrm{i} = 1, \ldots, \tau,\end{array} \tag{7}$$

where α_i, α_i^*, $\mathrm{i} = 1, \ldots, \tau$ are Lagrange multipliers. Three of the most commonly used kernel functions include Gaussian radial basis function (RBF) kernel, linear kernel and polynomial kernel. The experimental results of those three kernel functions reveal the best performance of RBF kernel function:

$$k\left(\mathbf{x}_{i,}\mathbf{x}_j\right) = \exp\left(-\gamma\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right) \tag{8}$$

Namely, we have the regression function with the form

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{w}^{\mathrm{T}}\mathbf{x} + \mathrm{b} \\ &= \sum_{i=1}^{\tau}\left(\alpha_i - \alpha_i^*\right)\exp\left(-\gamma\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right) + \mathrm{b}, \end{aligned} \tag{9}$$

First, SVR maps original nonlinear data into a high-dimensional space with linear feature. And kernel function is used for solving optimization problem with a loss function. Then the prediction problem can therefore be cast as a linear regression problem in a high-dimensional space. In training phase, we are aimed at identification of the appropriate parameters for our prediction model, namely C, ε, γ. Finally the next purchase interval can be predicted using the well trained SVR.

5 Life stage based recommendation

We have introduced the architecture of life stage based recommendation in Fig. 2. Next, we will describe its detailed process including mapping model, prediction of life stage and recommendation in the following.

5.1 Mapping model for life stages

The first step of life stage based recommendation is to construct a mapping model between items and the life stages at which an item is most probably bought. Given all the relevant purchasing behaviors on item *i*, we first aim to count the purchasing probability distribution of item *i* at each life stage.

Table 1 shows the mapping model, where *M* is the number of items, *N* is the number of life stages, and $Z_{i,\ j}$ is purchasing probability of item *i* at life stage *j*.

A crucial aspect is how to calculate all the elements in the mapping matrix. We consider several approaches to constructing the mapping model, such as logistic regression, Student's t test, and Grubbs' test model. Inspired by Student's t test and Grubbs' test, we use Eq.10 as a key judgement condition:

$$\frac{Q_{i,j}-\min(Q_i)}{\mathrm{std}(Q_i)}\geq\beta, \tag{10}$$

where $Q_{i,\ j}$ denotes the purchasing quantity of items *i* at the life stage s_j, $Q_i=\{Q_{i,\ 1},Q_{i,\ 2},\dots Q_{i,\ N}\}$, $\min(Q_i)$is the minimum of Q_i, and $\mathrm{std}(Q_i)$ is the standard deviation of Q_i.

The rule for calculating $Z_{i,\ j}$ is, find all the $Q_{i,\ j}$ that satisfy Eq.10 and suppose they form a set $Q_i^{'}$, then $Z_{i,\ j}=0$ if $Q_{i,\ j}\notin Q_i^{'}$, else $Z_{i,\ j}=Q_{i,\ j}/\sum Q_{i,\ j}$ where $Q_{i,\ j}\in Q_i^{'}$. Please note we only consider the $Q_{i,\ j}$s that satisfy Eq.10 to ensure the sum of each row is 1.

The value of β influences the number of non-zeros in a row. If β is too small, there are more non-zeros entries in a row, which means the item may be bought at several life stages and there is no clear mapping relation between the item and the life stages at which it is most probably bought. On the other hand, If β is too large, some possible life stages may be ignored. The value of β determines the mapping model, and the mapping model further influences the accuracy of predicting users' current life stage. So we must adjust the value of β by experiment to minimize the error of predicting users' current life stage in the next step.

A case example is provided to illustrate how to construct mapping model. The sample data in Table 2 are a part of our experimental dataset. The original data of a purchasing behavior is labeled with the baby's age, and we have to transfer the labels into life stages first. We transfer the labels according to Table 3. The candidate life stages, as well as the mapping relations between life stages and age groups of babies are provided by marketing experts based on baby product standards and their domain knowledge. We do not consider life stage 0 in Table 3, and we denote the life stage set as $L=[1, 2, 3, 4, 5]$.

For each item, we count its purchasing quantities at each life stage. Table 3 shows the purchasing quantities of five items at each life stage as an example.

In order to show the purchasing distribution more visually, we normalize the purchasing quantities and show them as Fig. 6. The x-axis denotes the life stage and the y-axis denotes the normalized purchasing quantities.

Taking above five items as an example, according to the proposed rule for mapping model construction, we can get a mapping matrix $Z_{M\times 5}$ as shown in Table 4. Since the number of L is 5, the number of columns of the mapping matrix is 5.

When constructing mapping model, we find many users without age information in the dataset. Therefore, we will next predict the life stage of a user without age information according to his purchasing records and mapping model.

5.2 Prediction model for User's current life stage

For users without age information, we cannot directly transfer the label of baby's age into their life stage, so we need to predict their life stage based on their recent behaviors. We consider user's *n* latest different purchasing records. Since we have achieved the mapping model between items and the corresponding life stages from the last step, and each item may be mapped to several life stages, the key problem is how to predict a user's life stage by allocating weights to his *n* latest purchasing records.

In our solution, we first initialize the weight vector $W_{1\times n}$to $[1/n,1/n,\dots,1/n]$, and then adjust weights of each record according to the number of non-zeros of the purchased item of the record in the mapping matrix $Z_{m\times 5}$. If there is only one non-zeros value in a row, this item is mapped to only one life stage, i.e. the item is only purchased in this life stage. So the more zeros a row has, the more obvious the mapping result is, therefore, we increase the weight of the items, i.e., the rows with more 0, since once we observe the purchasing behavior with the item, it will be more definitely to tell the life stage of the user; otherwise we decrease the weight. On the other hand, if there are multiple items that can be mapped to one same life stage in *n* records, we increase their weights, since the items together provide more powerful evidence to tell the life stage of the user.

The Pseudo-code of allocating weights to each records is shown as follows:

```
Input: Initialized weight matrix W
Output: Allocated weight matrix W
1： w=1
2： for n=1 to N
3：     $W_n \leftarrow W_n - \frac{\mathrm{length}(Q_n{}') - \min\{\mathrm{length}(Q_n{}')\}}{\vartheta * N}$
4： for n=1 to N
5：     if($W_n$ ! = 1/N)
6：         $w \leftarrow w - W_n$
7： $n_1 \leftarrow$ Get the number of rows there are minimum number of non-zeros
8： for n=1 to N
9： $n_2 \leftarrow$ Get the number of the same rows in $n_1$ rows
10：    if($W_n$ == 1/N)
11：        if(there is not any row with the same value vector)
12：            $W_n \leftarrow w/n_1 - n_2/(\theta * N)$
13：            $w \leftarrow w - W_n$
14： for n=1 to N
15：     if($W_n$ == 1/N)
16：         $W_n \leftarrow w/n_2$
17： return W
```

In line 1–3, w is the total weight which is initialized to 1 and we first update the weight of items that need to decrease the weight. Length() is a function that counts the number of non-zeros in a row namely $\{Z_{i,j}\}_{j=1}^{5}$. w is then updated to the remnant weight in line 4–6. In line 7, n_1 is the number of rows where there are minimum number of non-zeros. In line 8–13, we update the weight of items whose value vector $\{Z_{i,j}\}_{j=1}^{5}$ is unique and the remnant weight *w*, and at last the remnant weight *w* is equally allocated to the n_2 rows. In line 3 and line 12, ϑ and θ adjust the speed of weight updating. The smaller the ϑ and θ are, the faster the weight updates. We set ϑ = 3 and θ = 6 to ensure the weights are allocated optimally according to repeated test.

An item may be mapped to multiple life stages according to proposed mapping methods. In order to multiply the mapping vector with the allocated weight vector $W_{1\times n}$ to get the life stage by considering user's *n* records, we need to ensure that each item is mapped to only one life stage. According to the mapped life stage at which $Z_{i,j}$ is the closest to the mathematical expectation value, we have:

$$\mathrm{Map}_{n\times 1} = \left\{\min\left(\left|j - \sum_{j=1}^{5}\left(Z_{i,j} \times j\right)\right|\right)\right\}. \tag{11}$$

Then we can get the predicted value of the life stage $s_{u,\ t}$ using formula (12):

$$s_{u,t} = W_{1\times n} \cdot \mathrm{Map}_{n\times 1}. \tag{12}$$

Each purchased record corresponds to a specific time point. In order to get a corresponding time point for the calculated life stage $s_{u,\ t}$, we propose the formula (13):

$$t = \left\lfloor W_{1\times n} \cdot Y_{1\times n}{}^{T} \right\rfloor \left| \left\lfloor W_{1\times n} \cdot M_{1\times n}{}^{T} \right\rfloor \right| \left\lfloor W_{1\times n} \cdot D_{1\times n}{}^{T} \right\rfloor, \tag{13}$$

where $Y_{1\times n}$ is a vector of years in *n* records, $M_{1\times n}$ is a vector of *n* months, and $D_{1\times n}$ is a vector of *n* days. We calculate the three parts, i.e. the year, month, day separately and combine the three parts using a symbol |. Given the estimated life stage $s_{u,\ t}$ and its corresponding time point, in order to predict the user's current life stage $s_{u,\ now}$, we have:

$$s_{u,now} = \left\lfloor s_{u,t} + (T_{\mathrm{now}} - t)/D_{s_{u,t}} \right\rfloor, \tag{14}$$

where $D_{s_{u,t}}$ is the duration of life stage $s_{u,\ t}$, T_{now} is the current time point, and the unit of $(T_{\mathrm{now}} - t)$ and $D_{s_{u,t}}$ is month.

We continue to use the example in the last step. Assume that we have a user *u* without age information and consider his 5 latest purchased records of different items. His purchasing behavior sequence is X = [(*u*,50,006,042,20,150,918),(*u*,50,007,011,20,151,001),(*u*,50,012,370,20,151,005,),(*u*,50,018,436,20,151,115,),(*u*,50,023,606,20,160,118,)] and corresponding mapping matrix $Z_{5\times 5}$ is the first 5 lines of Table 4.

Then the initialized weight matrix is $W_{1\times 5}$= [1/5,1/5,1/5,1/5,1/5]. According to the algorithm of dynamically allocating weights, we first calculate $W_{1,\ 1} = 1/5 - (1 - 1)/(3 * 5) = 1/5$, $W_{1,\ 2} = 1/5 - (2 - 1)/(3 * 5) = 2/15$, $W_{1,\ 3} = 2/15$, $W_{1,\ 4} = 1/15$, $W_{1,\ 5} = 1/5$, $w = 1 - 2/15 - 2/15 - 1/15 = 2/3$, the number of rows where there are

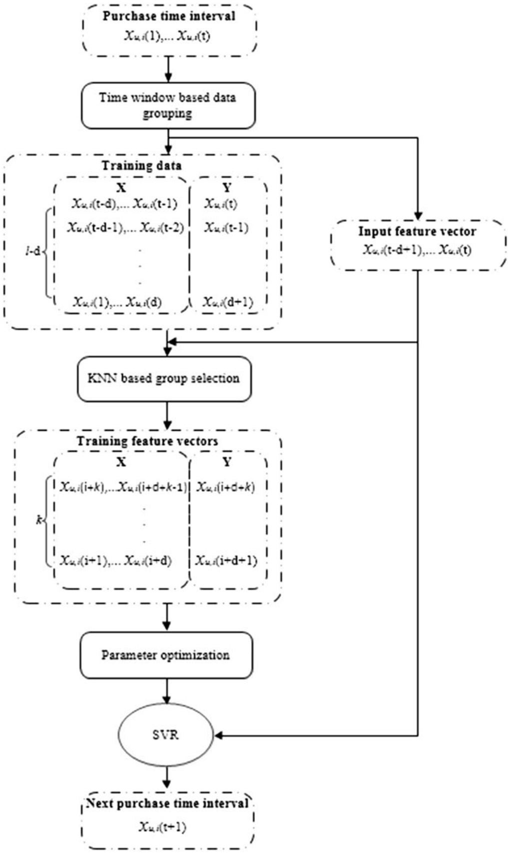

Purchase time interval
$X_{u,i}(1),\ldots X_{u,i}(t)$
Time window based data grouping
Training data
X
Y
$X_{u,i}(t-d),\ldots X_{u,i}(t-1)$
$X_{u,i}(t)$
$X_{u,i}(t-d-1),\ldots X_{u,i}(t-2)$
$X_{u,i}(t-1)$
l-d
$X_{u,i}(1),\ldots X_{u,i}(d)$
$X_{u,i}(d+1)$
Input feature vector
$X_{u,i}(t-d+1),\ldots X_{u,i}(t)$
KNN based group selection
Training feature vectors
X
Y
$X_{u,i}(i+k),\ldots X_{u,i}(i+d+k-1)$
$X_{u,i}(i+d+k)$
k
$X_{u,i}(i+1),\ldots X_{u,i}(i+d)$
$X_{u,i}(i+d+1)$
Parameter optimization
SVR
Next purchase time interval
$X_{u,i}(t+1)$

Fig. 4 The solution of first problem based on KNN + SVR

minimum number of non-zeros is 2, and the number of the same rows in these 2 rows is 2. Next, we get $W_{1,\ 1} = w/2 = 1/3$, $W_{1,\ 5} = w/2 = 1/$, namely the allocated weight vector is $W_{1\times 5} = [1/3, 2/15, 2/15, 1/15, 1/3]$.

According to Eq.11, we get$\mathrm{Map}_{5\times 1} = [4, 2, 3, 3, 4]^T$. According to Eq.12, the calculation process is $s_{u,\ t} = 4 \times 1/3 + 2 \times 2/15 + 3 \times 2/15 + 3 \times 1/15 + 4 \times 1/3 = 3.53$, namely this user is in the middle of the third life stage when he purchased the 5 items. According to Eq.13, the calculation process is $t = \lfloor 2015 \times 3/10 + 2015/15 + 2015/5 + 2015 \times 3/10 + 2016 \times 2/15 \rfloor \mid \lfloor 9 \times 3/10 + 10/15 + 10/5 + 11 \times 3/10 + 1 \times 2/15 \rfloor \mid \lfloor 18 \times 3/10 + 1/15 + 5/5 + 15 \times 3/10 + 18 \times 2/15 \rfloor = 20150610$. T_{now}=20160121, then we calculate: $s_{u,\ now} = \lfloor 3.53 + 7/6 \rfloor$=4, where 6 is the duration of life stage 3 according to Table 3.

5.3 Recommend new items based on life stage

Once we have estimated $s_{u,\ now}$, we can recommend the items that are appropriate to his current life stage $s_{u,\ now}$. By considering user's n latest different purchased behaviors, the current life stage $s_{u,\ now}$ may be the same as $s_{u,\ t}$, or may be in the following of $s_{u,\ t}$, therefore, we propose two recommendation strategies to differentiate the two cases.

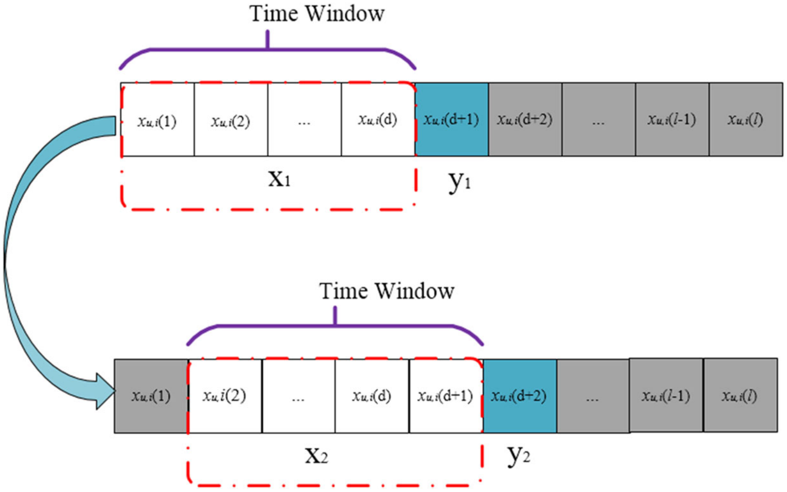
Figure 7 shows the two different cases. We use unweighted bipartite networks (UBN) model for (1) and Bayesian model for (2), because in the first situation, there are several records in the same life stage as a foundation for recommendation, but in the second, there are none for reference.

(1) $s_{u,\ now}$ is the same as $s_{u,\ t}$: we adopt an un-weighted bipartite networks model to present the relations between users and items, which is represented by an adjacency matrix $W = \{w_{ij}\}$ where $w_{ij} = 0$ if there is no edge between user U_i and item I_j, and $w_{ij} = 1$ otherwise. Assume that each item has a unit resource, a user distributes its resource to all his neighboring users, and then each user redistributes the received resource to all his/her purchased items. Accordingly, the resource that item I_m has received from item I_n is

$$R_{mn} = \frac{1}{k(I_n)} \sum_{i=1}^{U} \frac{w_{im} w_{in}}{k(U_i)}, \tag{15}$$

where $k(I_n)$ is the degree of I_n, namely the number of users who have purchased I_n, $k(U_i)$ is the degree of U_i,

Fig. 5 Time window



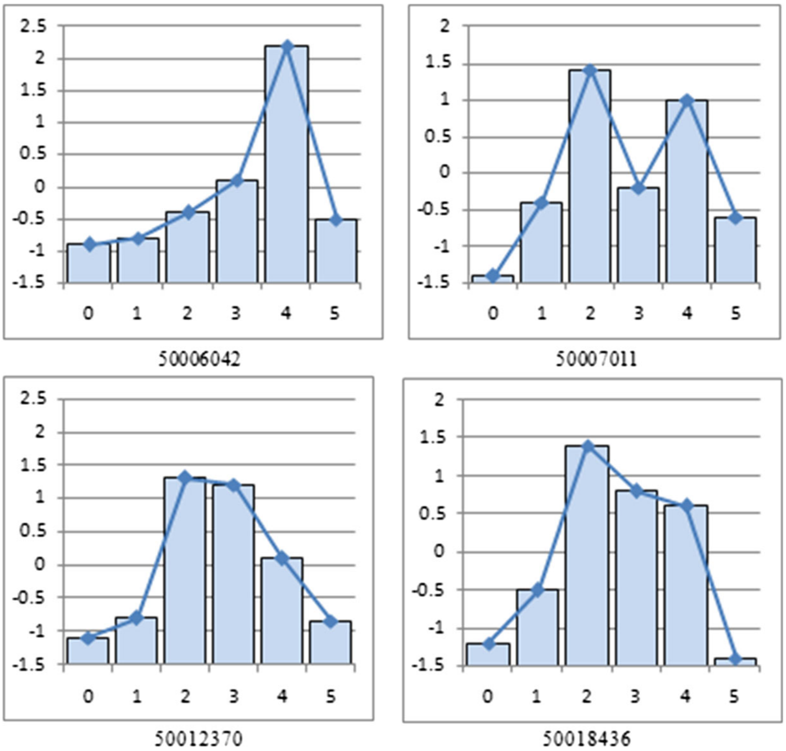


Fig. 6 The purchasing quantity of 4 items changes with life stage

namely the number of items the user U_i purchased. This process can be expressed by the matrix form $\overrightarrow{f'} = R \cdot \overrightarrow{f}$, where $R = \{R_{mn}\}$, $\overrightarrow{f}$ is the initial resource vector on items, and $\overrightarrow{f'}$ is the final resource vector. Given the target userU_i, the corresponding initial resource vector is defined as

$$f^i_m = w_{im} \tag{16}$$

According to the resource-allocation process discussed before, the final resource vector $\overrightarrow{f'}$ is

$$f'^i_m = \sum_{n=1}^{J} R_{mn} f^i_n = \sum_{n=1}^{J} R_{mn} w_{in} \tag{17}$$

Figure 8 provides a working process of un-weighted bipartite networks, where squares and circles present items and users, respectively. Red circle denotes target user and blue circles denote the others. Items purchased by target user and others at life stage $s_{u,\ t}$ denote red squares and blue squares, respectively. The left side of Fig. 8 depicts initial resource distribution and the right side presents final resource distribution. Then we can rank all items that hasn't been purchased by target user in life stage $s_{u,\ now}$ in descending order of final resource values, and recommend those items with the highest values to target user.

(2) $s_{u,\ now}$ is in the following of $s_{u,\ t}$: we adopt a Bayesian model to calculate the probability of a user purchasing an item at $s_{u,\ now}$, i.e. the joint probability$P(i, s_{u,\ now})$, where i is the item. Let $P(s_{u,\ now}|\ i)$ be the conditional probability of making the purchasement at $s_{u,\ now}$, given that the user has purchased item i. Let $P(i)$ be the probability of user purchasing item i. Based on the chain rule, we have:

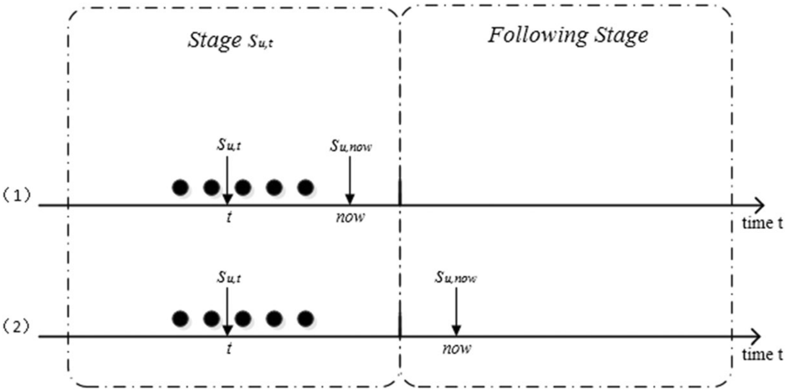


Fig. 7 Two different cases for $S_{u,t}$ and $S_{u,now}$

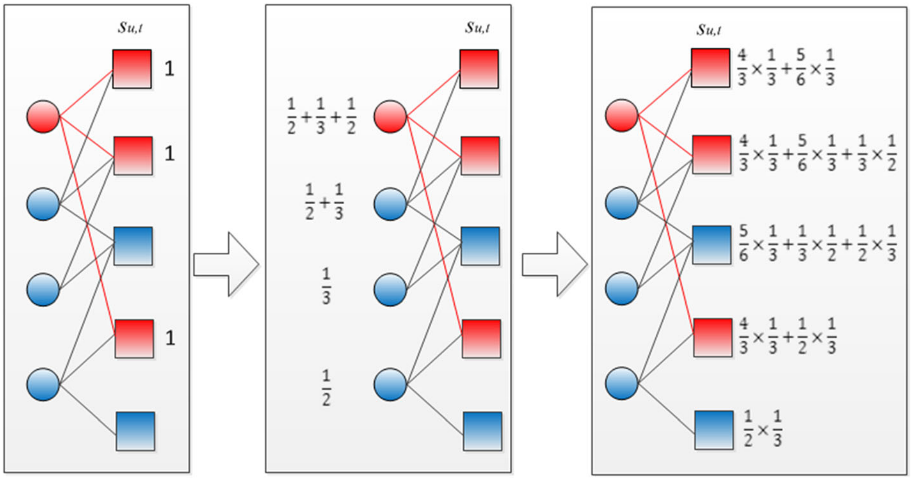

$s_{u,t}$
1
1
1
$\frac{1}{2}+\frac{1}{3}+\frac{1}{2}$
$\frac{1}{2}+\frac{1}{3}$
$\frac{1}{3}$
$\frac{1}{2}$
$\frac{4}{3}\times\frac{1}{3}+\frac{5}{6}\times\frac{1}{3}$
$\frac{4}{3}\times\frac{1}{3}+\frac{5}{6}\times\frac{1}{3}+\frac{1}{3}\times\frac{1}{2}$
$\frac{5}{6}\times\frac{1}{3}+\frac{1}{3}\times\frac{1}{2}+\frac{1}{2}\times\frac{1}{3}$
$\frac{4}{3}\times\frac{1}{3}+\frac{1}{2}\times\frac{1}{3}$
$\frac{1}{2}\times\frac{1}{3}$

Fig. 8 The working process of un-weighted bipartite networks

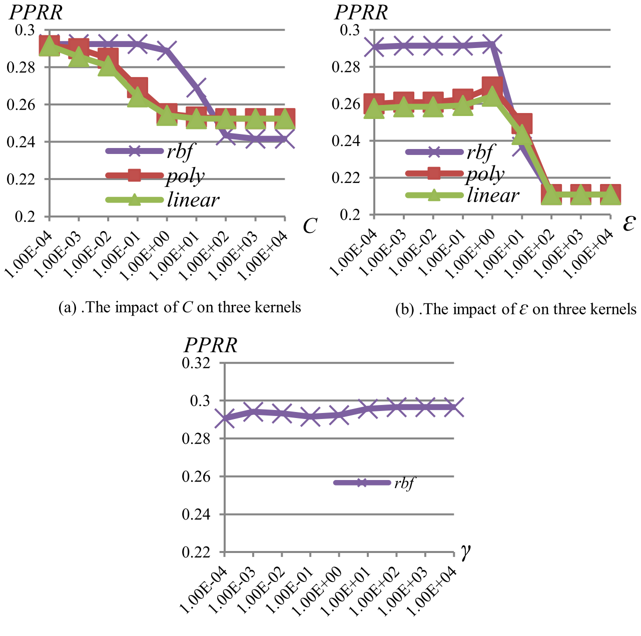
$$P\left(i, s_{u,now}\right) = P\left(s_{u,now}|i\right)P(i). \tag{18}$$

Then we can sort all purchased items at $s_{u,\ now}$ in descending order of $P(i, s_{u,\ now})$, and recommend those items with the highest values to target user. It is easy to understand that the most popular items at $s_{u,\ now}$ are recommended.

We continue to use the example in the last step and we have $s_{u,\ t} = 3$, $s_{u,\ now} = 4$.It is belonged to the second case. As Table 5 shown, taking the five new items that are mapped to the fourth life stage as an example, then we can get $P(121452056, 4) = \frac{16174}{45907} \times \frac{45907}{3415871} = 0.00047, P(50010557\ ,\ 4) = 0.0093, P(50012359\ \ \ ,4) = 0.0076, P(121408024, 4) = 0.0026, P\ (50008847, 4) = 0.0016$. So our final recommended list is [50,010,557, 50,012,359, 121,452,056, 121,408,024, 50,008,847]. If $s_{u,\ t} = 3$ and $s_{u,\ now} = 3$, it is belonged to the first case. First, we find all users who have purchased the five items from our dataset, and then find all items that those users purchased in previous step, we can therefore construct an un-weighted bipartite network similar to Fig. 8. Finally, a recommended list can be obtained according to the working process of the network. The final result is omitted here.

6 Experiments

In this section, we first introduce the experimental setup. And then, for the solution to the first problem, we determine the parameters of SVR model and analyze the effect of the size of time window d and the number of nearest neighbors k. Meanwhile, we compared the KNN + SVR with the other two methods: KNNR and SVR. For the solution to the second problem, we analyze the effect of β and m on life stage prediction and the effect of μ on recommending new items by using Two-way Analysis of Variance (ANOVA) [47], Meanwhile, we compare the our proposed recommendation model with the other two models: UBN and Bayesian.

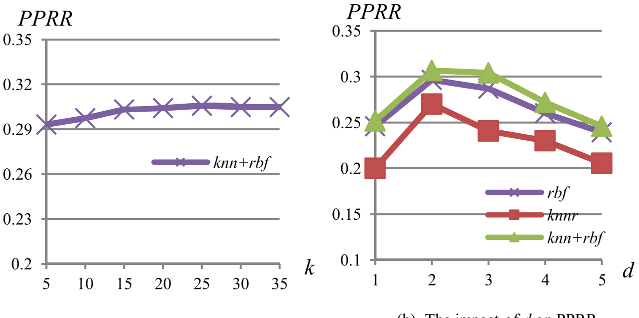


(a) .The impact of C on three kernels

(b) .The impact of ε on three kernels

(c) .The impact of γ on rbf kernel

Fig. 9 The experimental results of determining SVR model

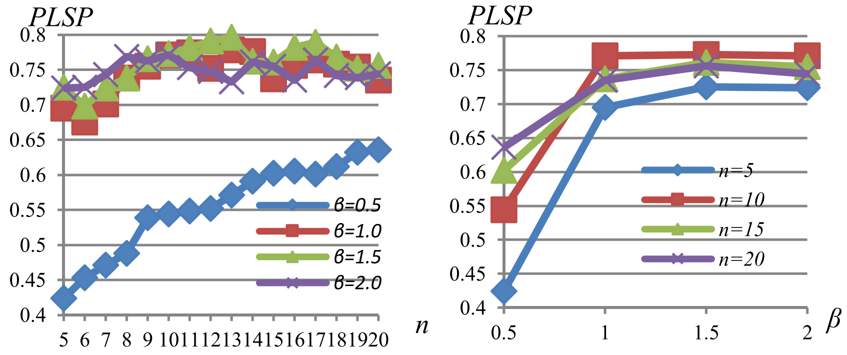


(a) .The impact of k on PPRR

(b) .The impact of d on PPRR

Fig. 10 The experimental results of PPRR

Fig. 11 The experimental results of PLSP



(a) The impact of *β* on *PLSP*

(b) The impact of *n* on *PLSP*

6.1 Experimental setup

To evaluate our proposed recommender system, we adopt the whole mum-baby dataset from TIANCHI, where those data was processed anonymously. Table 6 shows the description of dataset, where bolded columns are used in our experiment.

For each given purchasing record, it can be written as a 3-tuple <*user_id, item_id, day*>. Due to data sparsity problem in original dataset, we use parent category that an item belongs to instead, i.e., the 3-tuple <*user_id, cat_id, day*>.

In order to evaluate the precision of recommendation based on periodic regularity mining (PPRR), we divide all the time interval sets in the form of $\{x_{u,\ i}(1), \ldots x_{u,\ i}(n)\}$ into two parts. One part is the last time interval $x_{u,\ i}(n)$, which is used to compare with the predicted result. We then predict $x_{u,\ i}(n)$ based on the remainder part, i.e. $\{x_{u,\ i}(1), \ldots x_{u,\ i}(n-1)\}$, as training set to train the SVR model. Then we evaluate PPRR by verifying whether we predict $x_{u,\ i}(n)$ correctly, namely we have

$$\mathrm{PPRR}=\frac{\mathrm{N}_{|\mathrm{T}_{\mathrm{pre}}-\mathrm{T}_{\mathrm{real}}|\le 0.1*\mathrm{T}_{\mathrm{real}}}+\mathrm{N}_{|\mathrm{T}_{\mathrm{pre}}-\mathrm{T}_{\mathrm{real}}|\le 7}}{\mathrm{N}_{\mathrm{U}}}, \tag{19}$$

where $\mathrm{N}_{|\mathrm{T}_{\mathrm{pre}}-\mathrm{T}_{\mathrm{real}}|\le 0.1*\mathrm{T}_{\mathrm{real}}}$ is the number of predictions whose difference with real value $\mathrm{T}_{\mathrm{real}}$ is less than or equal to ± $0.1*\mathrm{T}_{\mathrm{real}}$. The criteria can stand for large $\mathrm{T}_{\mathrm{real}}$ values, but fails for small ones. For example, if$\mathrm{T}_{\mathrm{pre}} = 5$, and $\mathrm{T}_{\mathrm{real}} = 6$, the case does not satisfy $\mathrm{N}_{|\mathrm{T}_{\mathrm{pre}}-\mathrm{T}_{\mathrm{real}}|\le 0.1*\mathrm{T}_{\mathrm{real}}}$, but it should be a correct prediction. Because the shortest period we consider here is one week, and there is only 9.2% of time intervals which is less than 7 days, so, we think it is a correct prediction when $|\mathrm{T}_{\mathrm{pre}}-\mathrm{T}_{\mathrm{real}}| \le 7$. Therefore, we add $\mathrm{N}_{|\mathrm{T}_{\mathrm{pre}}-\mathrm{T}_{\mathrm{real}}|\le 7}$ in formula (19).

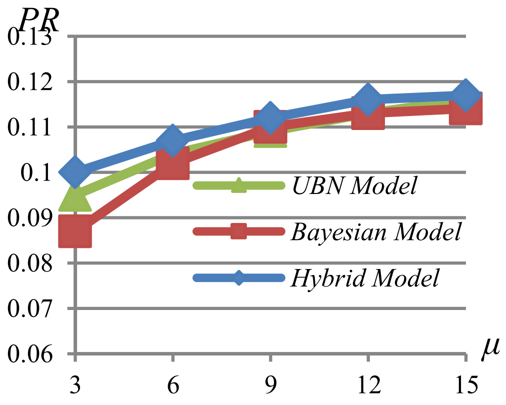


Fig. 12 The comparison of three models

For evaluating the precision of life stage prediction (PLSP), we divide dataset into two parts randomly. 80% are as the training sets, while 20% are as the test sets. The training set is used to learn the β、 *n* in the formula. We evaluate PLSP by comparing a user's real life stage and the predictive life stage, namely we have

$$\mathrm{PLSP}=\frac{\mathrm{N}_{\mathrm{S}_{\mathrm{pre}}=\mathrm{S}_{\mathrm{real}}}}{\mathrm{N}_{\mathrm{U}}}, \tag{20}$$

where $\mathrm{N}_{\mathrm{S}_{\mathrm{pre}}=\mathrm{S}_{\mathrm{real}}}$ is the number of users whose predictive life stage $\mathrm{S}_{\mathrm{pre}}$ equals to their real life stage $\mathrm{S}_{\mathrm{real}}$ in the test sets, N_{U} is the number of users in the test sets.

To evaluate the precision of life stage based recommendation (PLSR), each user's purchasing records will be classified into two parts according to the ascending order of their purchasing time. 20% of the data are removed to form a test set, and the remaining 80% of the data form a training set. We first use the training set to train parameterμ, and then examine the precision of PLSR through verifying whether purchased items of target user can be matched to the items we recommend, namely we have

Table 1 Mapping Matrix $Z_{M\times N}$

Item\Life stage	s_1	s_2	…	s_N
i_1	$Z_{1,\ 1}$	$Z_{1,\ 2}$		$Z_{1,\ N}$
i_2	$Z_{2,\ 1}$	$Z_{2,\ 2}$		$Z_{2,\ N}$
…			…	
i_M	$Z_{M,\ 1}$	$Z_{M,\ 1}$		$Z_{M,\ N}$

Table 2 Purchase distribution of five items

Item\Life stage	0	1	2	3	4	5	All
50,006,042	4316	6225	15,787	33,974	105,971	14,126	180,399
50,007,011	4137	9017	30,278	15,411	27,790	8968	95,601
50,012,370	3423	5236	25,443	25,775	14,111	4832	78,820
50,018,436	1359	4406	19,856	13,493	13,060	2199	54,373
50,023,606	1132	2863	4612	8535	17,466	3050	37,658
…							
All	248,533	436,837	539,745	517,606	1,111,720	561,430	3,415,871

$$\mathrm{PR} = \frac{\mathrm{N}_{\mathrm{L_{test}} \cap \mathrm{L_{rec}} \neq \Phi}}{\mathrm{N_U}}, \tag{21}$$

where $\mathrm{N}_{\mathrm{L_{test}} \cap \mathrm{L_{rec}} \neq \Phi}$ is the number of users whose list of purchased records $\mathrm{L_{test}}$ in the test sets and recommended list $\mathrm{L_{rec}}$ have intersection.

Due to the length limit of the paper, we do not describe the calculation formulas of Two-way ANOVA. In our Two-way ANOVA, we set $\alpha = 0.05$.

6.2 Determine the parameters of SVR model

In this part, we determine which kernel function is the most suitable and the parameters of the selective kernel function.

Each parameter of SVR code we used has a default value that is $C = 1\text{e}0$, $\varepsilon = 1\text{e-}1$, γ=0, *degree* = 3. γ is the parameter of RBF kernel. *Degree* is the parameter of polynomial kernel. C and ε are the common parameters of three kernel functions. Ensuring else parameters are default value, we first adjust C from 1e-4 to 1e4 to obtain experimental results as shown in Fig. 9(a), then adjust ε from 1e-4 to 1e4 to obtain experimental results as shown in Fig. 9(b). According to Fig. 9(a) and 9(b), we can see that the performance of RBF kernel function is better than poly kernel and linear poly in general. When >1e0, PPRR decreases with the growth of C. When $C \leq$ 1e0 , PPRR increases slowly with the decrease of C and when $C \leq$1e-1, PPRR is no longer changed. So we get appropriate value of C is 1e-1. Similarly, When $\varepsilon >$ 1e0, PPRR decreases with the growth of ε and when $\varepsilon >$ 1e2, PPRR is no longer changed. When $\varepsilon \leq$ 1e0 , PPRR decreases slowly with the decrease of ε and when $\varepsilon \leq$1e-1, the value of ε is no longer changed. So we get appropriate value of ε is 1e0.

Table 3 Life stages for Mom-baby domain

Life stages	Baby Age(Month)
1	−10-0
2	0–6
3	6–12
4	12–36
5	36–72
0	else

Next, we continue to determine the parameter γ of RBF kernel function, and Fig. 9(c) displays the experimental results. It shows that PPRR increases slowly with the growth of γ when $\gamma \leq$ 1e2, but PPRR is no longer changed with the growth of γ when $\gamma >$ 1e2. We get that most appropriate value of γ is 1e2.

6.3 Analysis of periodic regularity of purchase Recommen-dation

In this section, based on the determined SVR model in the previous section, we continue to determine the number of the nearest neighbors k of KNN + SVR (RBF) method. And then determine the size of time window d, and compare the performance of the three methods: KNNR, SVR (RBF), KNN + SVR (RBF).

We set d = 2, k as a variable (5 to 35) to obtain experimental results as shown in Fig. 10(a). It shows that PPRR increases with the growth of k when $k \leq$ 25, but PPRR decreases with the growth of k when $k >$ 25. So the most suitable value of k is 25.

Next, we determine the size of time window d, as shown in Fig. 10(b), all three lines reach its peak at d = 2. It means, when most user purchases the same item again, he/she considers the last two purchased time. Namely, users' last two purchased time has the greater influence on next time of purchasement. As shown in Fig. 10(b), the performance of KNN + SVR (RBF) is better than KNNR and SVR (RBF) in

Table 4 Mapping matrix in Mom-Baby domain

Item\Life stage	1	2	3	4	5
50,006,042	0	0	0	1	0
50,007,011	0	0.52	0	0.48	0
50,012,370	0	0.50	0.50	0	0
50,018,436	0	0.43	0.29	0.28	0
50,023,606	0	0	0	1	0
…	…				

Table 5 Purchase distribution of five new items

Item\Life stage	0	1	2	3	4	5	All
121,452,056	5027	2006	3241	4560	16,174	14,899	45,907
50,010,557	11,292	3224	5494	8661	31,720	24,324	84,751
50,012,359	15,828	4050	3427	3976	26,076	27,111	80,468
121,408,024	2633	677	1121	1871	8875	7923	23,100
50,008,847	2204	665	538	584	5439	7198	16,619
…							
All	248,533	436,837	539,745	517,606	1,111,720	561,430	3,415,871

general. By combining the conclusions of section B and C, we can get that when C = 1e-1, Ɛ = 1e0, γ = 1e2, k = 25 and d = 2, the performance of KNN + SVR is the best and the prediction precision PPRR = 30.68%.

6.4 Analysis of life stage prediction

In the second problem, we aim to find the appropriate β in formula (10) and n so that the prediction of life stage is the most accurate. n indicates the number of user's purchasing records that we observe.

Figure 11(a) presents the impact of regulatory factor β, where x-axis is the number of records and y-axis is the precision of life stage prediction. It is easy to see that whenβ = 0.5, the prediction precision is the minimum. When $\beta = 1.0$, $\beta =$ 1.5, $\beta = 2.0$, the precision is similar. But when $\beta = 2.0$, the trend of precision is the most stable, so we take $\beta = 2.0$ as the optimal value.

The impact of n is shown in Fig. 11(b). When $n = 10$ and $\beta \neq 0.5$, the prediction precision achieves to the maximum and its trend is the most stable. Therefore, we use the latest only 10 records to predict the life stage of a user. When $\beta = 2.0$ and $n = 10$, the prediction precision is 77.2%.

In our experiment, β and n are both studied according to Fig. 11(a), where factor β and n have 4 and 16 levels, respectively. In all, there are 64 cells. Because there only one observation in each cell, we do not consider the interaction between two factors. The Two-way ANOVA table of Fig. 11 is shown in Table 7,

Table 6 The description of dataset

Column	Description
user_id	User ID (Bigint)
birthday	Children's birthday (e.g. 20,130,423)
gender	Children's gender ('0' denotes female, '1' denotes male, '2' denotes unknown)
item_id	Item ID (Bigint)
cat_id	Category ID (Bigint)
cat1	Root category ID (Bigint)
property	Property of the corresponding item (String)
buy_mount	Purchase quantity (Bigint)
day	Timestamp

where SS denotes the sum of square, DF denotes the degree of freedom, MSS denotes the means sum of squares, F denotes the F-Value. According to Table 7 $F_\beta = 174.95840 > F(3, 45) = 2.81154$ and $F_m = 3.91805 > F(15, 45) = 1.89488$, so the two factors have statistically significant effect on PLSP.

6.5 Analysis of life stage recommendation

In order to observe the impact of μ, which denotes the length of recommendation list, on the precision of recommendation, we also adjust different μ as shown in Fig. 12.We compare the UBN model only, the Bayesian model only and the hybrid model stated in Section V.C.

Figure 12 shows that with μ increases, the precision of recommendation improves slowly, because a larger μ means a longer recommendation list. However, it will lose the original meaning of recommendation when μ is too large. So we set $\mu = 15$. From the experimental results, we can know that recommendation based on hybrid model is more effective than pure UBN model and Bayesian model, and its precision can achieve to 11.6%.

Similarly, the categories of model c and μ are both studied according to Fig. 12, where factor c and factor μ have 3 and 5 levels, respectively. In all, there are 15 cells.

Two-way ANOVA table of Fig. 12 is shown in Table 8. $F_c = 5.94186 > F(2, 8) = 4.45897$ and $F_\mu = 40.43605 > F(4,$

Table 7 The Two-Way Anova Table of Fig. 11

Source	SS	DF	MSS	F
Factor β	0.46649	3	0.15550	174.95840
Factor n	0.05223	15	0.00348	3.91805
Error	0.03999	45	0.00089	
Total	0.55871	63		

Table 8 The Two-Way Anova Table of Fig. 12

Source	SS	DF	MSS	F
Factor c	0.00007	2	0.00003	5.94186
Factor μ	0.00093	4	0.00023	40.43605
Error	0.00005	8	0.00001	
Total	0.00105	14		

8) = 3.83785, so the two factors have statistically significant effect on PLSR.

7 Conclusions

In this paper, we focus on mining consuming behaviors with temporal evolution to make accurate recommendation in mobile marketing apps for maternal and child. We consider two kinds of temporal factors in consuming behavior: periodic regularity of purchasement and demand evolution based on life stage. We first propose to mine the periodic trends of users' consuming behavior from historical records, and predict the next time when a user re-purchases the item, so as to recommend some items that users have purchased before at proper time. Second, we aim to find the regularity of users' purchasing behavior during different life stages and recommend the new items that are needed and proper for their current life stage.

For mining the periodic regularity of purchasement, first, a time window is proposed for dividing time sequence data into plenty of feature vectors as training dataset, next, a KNN approach is utilized to reduce the training dataset, finally, the time interval of user's next purchasement can be predicted based on SVR model. For life stage based recommendation, at the beginning, the system mines the relations between user's life stage and items. Then, according to mined relation model, it can predict user's current life stage using dynamic allocation weight algorithm. Eventually, according to the unweighted bipartite networks (UBN) model and Bayesian model, new appropriate items will be recommended to target user. The experimental results show that the proposed method is reasonable and effective.

This is the first step towards mining consuming behaviors with temporal evolution in mobile marketing apps for maternal and child. Furthermore, we will apply the recommendation approach into apps in other fields, such as pet feeding, health keeping and wedding. In addition, there are some other factors we can add into our system, such as item popularity and item property, to make the recommendations more accurate.

Acknowledgements This work has been supported by the National Key R&D Program of China under grant 2018YFB1003800, National Natural Science Foundation of China (No.61772560), Natural Science Foundation of Hunan Province (No. 2019JJ40388).

Compliance with ethical standards

Conflict of interests The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Mobile Marketing Statistics to Help You Plan for 2017. https://www.impactbnd.com/blog/mobile-marketing-statistics-for-2016
2. Brusilovsky P, Kobsa A, Nejdl W (2007) The adaptive web: methods and strategies of web personalization [J]. Lect Notes Comput Sci
3. Jian C, Jian Y, Jin H (2005) Automatic Content-Based Recommendation in e-Commerce[M]// Automatic content-based recommendation in e-commerce. 748–753
4. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Comput 7(1): 76–80
5. Adomavicius G, Tuzhilin A (2005) Tuzhilin, a.: toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering 17(6), 734-749[J]. IEEE Transactions on Knowledge & Data Engineering 17(6):734–749
6. Da-Xue LI, Xie ML, Zhao XB (2010) Collaborative filtering recommendation algorithm based on naive Bayesian method[J]. Journal of Computer Applications 29(10):2403–2411
7. Yin Y, Chen L, Xu Y, Wan J, Zhang H, Mai Z (2019) QoS Prediction for Service Recommendation with Deep Feature Learning in Edge Computing Environment. Mobile Networks and Applications. https://doi.org/10.1007/s11036-019-01241-7, QoS Prediction for Service Recommendation with Deep Feature Learning in Edge Computing Environment
8. Yin Y, Chen L, Xu Y, Wan J (2018) Location-Aware Service Recommendation With Enhanced Probabilistic Matrix Factorization. IEEE Access 6:62815–62825
9. Yin Y, Xu W, Xu Y, Li H, Yu L (2017) Collaborative QoS Prediction for Mobile Service with Data Filtering and SlopeOne Model. Mobile Information Systems 2017: 7356213:1–7356213: 14
10. Adomavicius G, Tuzhilin A (2008) Context-aware recommender systems[C]// ACM conference on recommender systems. ACM, 2175–2178
11. Wang S, Zheng Z, Wu Z, Sun Q, Zou H, Yang F (2014) Context-aware mobile service adaptation via a co-evolution eXtended classifier system in mobile network environments[J]. Mob Inf Syst 10(2):197–215
12. Rosa PMP, Rodrigues JJPC, Basso F (2013) A weight-aware recommendation algorithm for mobile multimedia systems[J]. Mob Inf Syst 9(2):139–155
13. Gao H, Zhang K, Yang J, Wu F, Liu H (2018) Applying improved particle swarm optimization for dynamic service composition focusing on quality of service evaluations under hybrid networks. International Journal of Distributed Sensor Networks(IJDSN) 14(2):1–14
14. Gao H, Chu D, Duan Y (2017) The probabilistic model checking based service selection method for business process modeling. Journal of Software Engineering and Knowledge Engineering 27(6):897–923
15. Gao H, Mao S, Huang W, Yang X (2018) Applying Probabilistic Model Checking to Financial Production Risk Evaluation and Control: A Case Study of Alibaba's Yu'e Bao. IEEE Transactions on Computational Social Systems(TCSS) 5(3):785–795

16. Safar M (2005) K nearest neighbor search in navigation systems[J]. Mob Inf Syst 1(3):207–224
17. Lin K, Lin Q, Zhou C, et al. (2007) Time Series Prediction Based on Linear Regression and SVR[C]// International Conference on Natural Computation. 688–691
18. Zeng W, Zhu YX, Lü L et al (2011) Negative ratings play a positive role in information filtering[J]. Physica A Statistical Mechanics & Its Applications 390(s 23–24):4486–4493
19. Barnard T, Prügel-Bennett A (2011) Experiments in Bayesian recommendation[J]. Advances in Intelligent & Soft Computing 86:39–48
20. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Comput 7(1): 76–80
21. Das AS, Datar M, Garg A et al. (2007) Google news personalization: scalable online collaborative filtering[J]. Www, 271–280
22. Park ST, Pennock DM (2007) Applying collaborative filtering techniques to movie search for better ranking and browsing[C]// Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. 550–559
23. Liu B, Bennett J, Elkan C et al (2007) KDD cup and workshop 2007[J]. Acm Sigkdd Explorations Newsletter 9(2):51–52
24. Schafer JB, Dan F, Herlocker J et al (2007) Collaborative filtering. Recommender system[J]. Lect Notes Comput Sci 9(3):46–45
25. Adomavicius G, Tuzhilin A (2005) Tuzhilin, a.: toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering[J]. IEEE Transactions on Knowledge & Data Engineering 17(6):734–749
26. Yin J, Lo W, Deng SG, Li Y, Wu Z, Xiong N (2014) Colbar: a collaborative location-based regularization framework for QoS prediction. Inf Sci 265(68–84):68–84
27. Yin Y, Aihua S, Gao M, Xu Y (2016) Wang Shuoping:QoS prediction for web service recommendation with network location-aware neighbor selection. Int J Softw Eng Knowl Eng 26(4):611–632
28. Fouss F, Pirotte A, Renders JM, Saerens M (2007) Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation[J]. Knowledge & Data Engineering IEEE Transactions on 19(3):355–369
29. Sarwar B, Karypis G, Konstan J, et al. (2001) Item-based collaborative filtering recommendation algorithms[C]// proceedings of the 10th international conference on world wide web. ACM, 285–295
30. Zhou D, Zhang C, Liu L, et al. (2004) Taxonomy-driven Computation of Product Recommendations[C]// Thirteenth Acm International Conference on Information & Knowledge Management. 406–415
31. Schein AI, Popescul A, Ungar LH, et al. (2002)Methods and metrics for cold-start recommendations[C]// proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval. ACM, 253–260
32. Koren Y (2010) Collaborative filtering with temporal dynamics[J]. Commun ACM 53(4):89–97
33. Khoshneshin M, Street WN (2010) Incremental collaborative filtering via evolutionary co-clustering.[C]// In Proc. of the Fourth ACM Conf. on Recommender Systems. 325–328
34. Li B, Zhu X, Li R, et al. (2011) Cross-Domain Collaborative Filtering over Time.[C]// Twenty-second International Joint Conference on Artificial Intelligence-volumethree. 2293–2298
35. Ren Y, Zhu T, Li G, et al. (2013) Top-N recommendations by learning user preference dynamics[M]// advances in knowledge discovery and data mining. Springer Berlin Heidelberg, 390–401
36. Xiang L, Yuan Q, Zhao S, et al. (2010) Temporal recommendation on graphs via long- and short-term preference fusion[C]// proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 723–732
37. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized Markov chains for next-basket recommendation.[C]// International Conference on World Wide Web. 811–820
38. Wang J, Zhang Y, Wang J, et al. (2013) Opportunity model for e-commerce recommendation[C]// proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. ACM, 303–312
39. Jiang P,Zhu Y,Zhang Y, Yuan Q (2015) Life-stage Prediction for Product Recommendation in E-commerce[J]. Proceedings of the 21th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1879–1888
40. Wu SF, Lee SJ (2015) Employing local modeling in machine learning based methods for time-series prediction[J]. Expert Syst Appl 42(1):341–354
41. Martínez-Rego D (2011) Efficiency of local models ensembles for time series prediction.[J]. Expert Syst Appl 38(6):6884–6894
42. Sapankevych NI, Sankar R (2009) Time series prediction using support vector machines: a survey[J]. IEEE Comput Intell Mag 4(2):24–38
43. Thissen U, Brakel RV, Weijer APD et al (2015) Using support vector machines for time series prediction[J]. Chemometrics & Intelligent Laboratory Systems 69(1–2):35–49
44. Huang Z, Shyu M L (2010) k-NN based LS-SVM framework for long-term time series prediction[C]// IEEE International Conference on Information Reuse and Integration. IEEE, :69–74
45. Zhang L, Zhou WD, Chang PC, Yang JW, Li FZ (2013) Iterated time series prediction with multiple support vector regression models[J]. Neurocomputing 99(1):411–422
46. Guo B, Dou K, Kuang L. Life stage based recommendation in e-commerce. International Joint Conference on Neural Networks (IJCNN). 3461–3468
47. Fisher RA (1958) Statistical methods for research workers[M]// statistical methods for research workers.. Госстатиздат, 66–70

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.