Daily-Aware Personalized Recommendation based on Feature-Level Time Series Analysis

Yongfeng Zhang[†‡],Min Zhang[†],Yi Zhang[‡],Guokun Lai[†],Yiqun Liu[†],Honghui Zhang[†],Shaoping Ma[†]
[†]State Key Laboratory of Intelligent Technology and Systems, Dept. of CS, Tsinghua University, China
[†]Tsinghua National Laboratory for Information Science and Technology (TNList)
[‡]School of Engineering, University of California, Santa Cruz, CA 95060, USA
yongfeng14@mails.tsinghua.edu.cn, z-m@tsinghua.edu.cn, yiz@soe.ucsc.edu
{laiguokun,zhhthu11}@gmail.com, {yiqunliu,msp}@tsinghua.edu.cn

ABSTRACT

The frequently changing user preferences and/or item profiles have put essential importance on the dynamic modeling of users and items in personalized recommender systems.

However, due to the insufficiency of per user/item records when splitting the already sparse data across time dimension, previous methods have to restrict the drifting purchasing patterns to pre-assumed distributions, and were hardly able to model them rather directly with, for example, time series analysis. Integrating content information helps to alleviate the problem in practical systems, but the domain-dependent content knowledge is expensive to obtain due to the large amount of manual efforts.

In this paper, we make use of the large volume of textual reviews for the automatic extraction of domain knowledge, namely, the explicit features/aspects in a specific product domain. We thus degrade the product-level modeling of user preferences, which suffers from the lack of data, to the feature-level modeling, which not only grants us the ability to predict user preferences through direct time series analysis, but also allows us to know the essence under the surface of product-level changes in purchasing patterns. Besides, the expanded feature space also helps to make cold-start recommendations for users with few purchasing records.

Technically, we develop the Fourier-assisted Auto-Regressive Integrated Moving Average (FARIMA) process to tackle with the year-long seasonal period of purchasing data to achieve daily-aware preference predictions, and we leverage the conditional opportunity models for daily-aware personalized recommendation. Extensive experimental results on real-world cosmetic purchasing data from a major e-commerce website (JD.com) in China verified both the effectiveness and efficiency of our approach.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering; G.3 [**Mathematics of Computing**]: Probability and Statistic - *Time Series Analysis*

Keywords

Recommender Systems; Time Series Analysis; Collaborative Filtering; Sentiment Analysis

WWW 2015, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741087.

1. INTRODUCTION

The vast amount of online products in various e-commerce websites make it an essential task to develop reliable Personalized Recommender Systems (PRS) [24]. One of the most extensively investigated approaches for personalized recommendation is to model user preferences according to his/her historical choices, as well as those of the others', leading to the success of Collaborative Filtering (CF) techniques [27].

However, both user preferences and item profiles may change dynamically over time [16, 34, 11], thus treating the historical decisions of a user or the received comments of an item as static, fixed, or long-term influential information sources can be infeasible in real-world applications. In the product domain of cosmetics for temperate area users, for example, one may buy a sun cream product in summer when the UV radiation is strong, while purchase nourishing creams in winter when it is cold and dry, as exposited in Fig.1. If the system simply treats the user's historical purchasing records as a whole without dynamic modeling of his/her preference, then the user may receive sun cream recommendations frequently in the cold winter, and vise versa. Similarly, an item can receive new reviews from users continuously in practical systems, which makes its profile drift dynamically over time, as a result, it would be unwise for the system to recommend items that no longer fit the user needs any more.

The dynamic nature of users and items requires the ability of time-dependent modeling in recommender systems. Researchers have been putting efforts to capture this nature and integrate time factors into the recommendation process, giving rise to the research of time-aware recommendation [5], which is also generally categorized into the research topic of Context-Aware Recommender Systems (CARS) [1].

The most early approaches for time-aware recommendation mainly focus on the dynamic modeling of user and item profiles [6, 2, 9]. However, recommendation based merely on content profiles may lack the ability to take advantage of the wisdom of crowds, which is proved effective in many CF approaches. This leads to two concurrent research lines. Koren [16] and Xiang [34] adopted the approaches of dynamic modeling through time bins on explicit feedbacks, and sessions on implicit feedbacks, respectively, which were further combined by Dror [11] into Yahoo! Music recommendation and achieved superior results in KddCup-2011; At the same time, Lu [19] and Shi [25], as well as Karatzoglou [15] and Gantner [13] adopted matrix/tensor factorization to model time as context information. Recently, Wang [32, 31] investigated users' subsequent purchasing behavior with opportunity models for recommendation.

However, both the profile-based and the dynamic modeling approaches attempt to model user preferences in the produce space according to user purchasing and rating histories, which suffer from the problem of data sparsity due to the fact that a single user may be only related to a small number of products, compared with the vast amount of products in a system [32]. As a result, many of the current approaches have to assume pre-defined distributions over time and conduct parameter estimation thereon for purchasing prediction. It is shown that considering content information, such as popularity, product names and categories, helps to alleviate the problem [9], but such information is either difficult to obtain and structure, or expensive for manual generation.

Fortunately, the increasing availability of textual user reviews and the ever developing text summarization and sentiment analysis techniques have made it possible to take advantage of this important information source besides numerical ratings [38]. In this work, we extract and summarize the domain knowledge (i.e., domain-specific product features like *anti-UV* or *nutrition* for cosmetics) automatically from the textual user reviews, and conduct feature-level dynamic modeling to capture the drifting user preferences and item profiles, thus to avoid the lack of data in previous product-level modeling approaches. This allows us to conduct direct time series analysis to predict the importance of each feature and to provide pre-time favoured recommendations. More importantly, by analyzing the seasonal, cyclic and trend factors of different features, we are able to understand in detail how users' preferences change over time.

In order for rapid track of the fashion and provide timely recommendation, we choose to model and predict the time series in daily resolution. This leads to nearly year-long period (up to 365) time series, making them difficult to model with standard Auto-Regressive Integrated Moving Average (ARIMA) process due to expensive computational costs. As a result, we develop the Fourier-assisted ARIMA (FARIMA) process that leverages Fourier terms to model the seasonal patterns and short-term ARIMA process to model the error. Based on the product features that have been predicted of different importances, we further adapt the conditional opportunity model for daily-aware recommendation. We conduct extensive experiments to exposit both the intuition of how user preferences drift over time in e-commerce, and the superior performance of our proposed method.

The following of the paper is organized as follows: we review the related work in Section 2, and provide the preliminary studies of time series for e-commerce in Section 3; we propose our model for daily-aware time series prediction and personalized recommendation in Section 4 and Section 5, respectively; in Section 6, we present the experimental results from different aspects on real-world purchasing dataset; and we conclude the work in Section 7.

2. RELATED WORK

With the ability to help discover items of potential interests, Personalized Recommender Systems (PRS) [24] have been widely integrated into many online applications, such as e-commerce, social networks, and online review services. Early systems for personalized recommendation rely on content-based approaches [21], which make recommendations by analyzing the item features or user demographics. Recently, the Collaborative Filtering (CF) [27] based approaches have gained great popularity due to their free from human efforts,

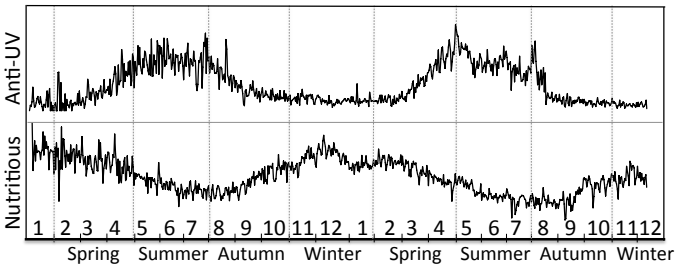


Figure 1: The frequency of two product features (Anti-UV and Nutritious) in user reviews over two consecutive years. The data is from JD.com, a major e-commerce website in China. One can see that Anti-UV seasonally prospers in summer, while Nutritious seasonally prospers in winter.

superior performance, and the ability to take advantage of the wisdom of crowds.

In real-world systems, however, it is usually observed that both user preferences and item profiles may dynamically drift over time [5], thus a static modeling of users/items that treats the historical purchases or reviews as fixed and long-term effective may be inappropriate to track users' interests. For example, a user may be less likely to buy an anti-UV suncream in winter, although she may well have bought one in summer; and a user who has purchased an SLR camera may not buy another one in a reasonably long period. As a result, the ability of time-aware modeling is of essential importance to practical systems [5].

In the early stage of the research on time-aware recommendation, researchers attempted to construct dynamic user or item profiles. Chen et al. [6] constructed a system that discovers, stores and updates private dynamic user profiles for personalized content recommendation; Baltrunas and Amatriain [2] proposed micro-profiling, which splits a user profile into several sub-profiles of different contexts; Chu and Park [9] combined both user and item profiles in a dynamic bilinear model for time-aware recommendation. A review of various user/item profiling techniques is provided in [14].

However, the profiling approaches usually require the presence of specific domain knowledge, which is usually expensive to obtain. Besides, it is difficult for them to leverage the wisdom of crowds. This drives researchers to investigate the CF approaches for time-aware recommendation.

Oku [20] and Yuan [37] integrated time factors into the similarity calculation of neighbour-hood based methods for time-aware CF; Koren [16] proposed timeSVD++ for dynamic CF with factor models by tracking the drifting user/item biases across different time bins; Xiang et al. [34] further extended the dynamic modeling to implicit feedback through random walk on sessions-based temporal graphs; Lu [19] and Shi [25] adapted the famous Matrix Factorization (MF) [28] approaches for time-aware CF, while Karatzoglou [15] and Gantner [13] leveraged tensor factorization to integrate the many contextual features. The topic modeling approaches are also investigated by Chen [7] and Vaca [30], while recently, Wang et al. [32, 31] borrowed the opportunity models from survival analysis to predict users' subsequent purchases for time-aware recommendation.

However, current approaches model the time-dependent user preferences in the product space directly, which suffers from the problem of data sparsity because a single user is

usually related to only a small number of products compared with the huge product space in a system. Incorporating content information like product categories helps to build more accurate models [1, 9, 32], but such information is either absent in practical applications or difficult to generate and structure manually.

Fortunately, the ever developing phrase-level sentiment analysis techniques [18, 10, 36, 29] have made it possible to extract, summarize and structure the rich product features and user preferences automatically from free-text reviews, and have begun to be leveraged for personalized recommendation tasks [38, 33]. This also sheds light on the feature-level dynamic modeling for recommendation. Compared with product-level modeling, feature-level modeling is finer-grained because a user who made only a single review may well have commented on several product features. Besides, by investigating into the detailed features, we can obtain an intuitional understanding of how and why the user purchasing behaviors drift over time.

We adopt Time Series Analysis (TSA) [3, 26] for daily prediction of the trend of product features. Time series analysis has long been used in many scientific data analysis tasks, such as econometrics, bioinformatics, physics, and astronomy [26]. Recently, it has also begun to be applied to data mining tasks, such as epidemic prediction [35], and economic indicators prediction in Google Trends [8]. However, to the best of our knowledge, it has not been investigated in personalized recommendation tasks, perhaps because of the lack of per-user purchasing data as mentioned above. Nevertheless, as we will exposit in the following, the information-rich textual reviews have made such analysis practical in real-world systems, which helps not only in time-aware recommendation, but also a better understanding of user behaviors.

3. TIME SERIES IN E-COMMERCE

Before introducing the models for daily-aware recommendation, we execute an initial study of the time series in e-commerce to exposit the intuition. The data is from JD.com, a major e-commerce website in China, and the data consists of 5,524,491 reviews from 1,844,569 users towards 53,188 products since Jan. 1^{st}, 2011 to Mar. 31^{st}, 2014. We introduce the automatic extraction of product features from free-text reviews fist, and then focus on the trend, seasonal, cyclic and random effects of feature-level time series.

3.1 Product Feature Extraction

We extract a set of product features $\mathcal{F} = \{f_1, f_2, \cdots, f_r\}$ based on a review corpus from a product domain, *e.g.*, cosmetic products. We leverage the state-of-the-art phrase-level sentiment analysis techniques described in [18, 39, 22]. Generally, we adopt the Stanford Parser[1] to conduct part-of-speech tagging, morphological analysis and grammatical analysis on each of the reviews to construct the corresponding dependency tree, and then retain those Noun Phrases (NP) whose frequency are above a specific threshold as feature candidates. The candidates are further pruned by Pointwise Mutual Information (PMI) [22] to form the final feature set of this domain. In cosmetics for example, the extracted features include *anti-UV*, *nutritious*, *price*, *feeling*, etc. These features serve as the feature space in the following part of time series analysis, prediction, and personalized recommendation.

[1] http://nlp.stanford.edu/software/lex-parser.shtml

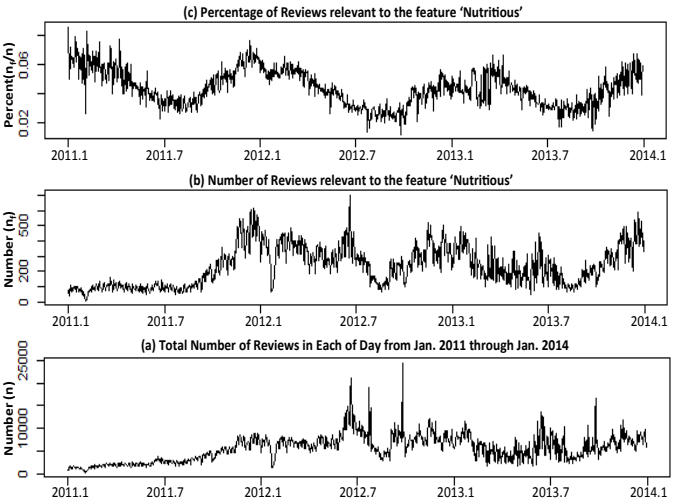


Figure 2: Three daily time series of the feature 'Nutritious': (a) The total number of reviews per day $N(t)$; (b) The number of reviews where 'Nutritious' is commented on per day $N_f(t)$; (c) The percentage of reviews $X_f(t) = N_f(t)/N(t)$ per day where 'Nutritious' is commented on.

The process of feature word extraction is fully automatic and requires no human efforts. Since the process is not the focus of this work, we refer the readers to the related literature [18, 39, 22] for more details. The quality evaluation of the extracted features will be reported in the experiments.

3.2 Feature-level Time Series

In this section, we investigate three candidate daily time series for each feature f to determine the appropriate one for dynamic user preference modeling and time-aware recommendation, as exampled in Fig.2 for the feature *nutritious*.

We first plot the total number of reviews per day $N(t)$ in Fig.2(a). This time series is an indicator of the total purchasing volume in the system, which is independent from a specific user, product or feature. After careful examination, we find that the peaks in $N(t)$ correspond to the online sales promotional campaigns carried out by the website. For example, on Oct. 1^{st} in the year of 2012 and 2013, the website conducted sales promotion to celebrate the National Day.

As we aim to investigate the different time series patterns of different features, we further plot the number of reviews $N_f(t)$ per day where the target feature f is commented on, as shown in Fig.2(b). We expect to discover regular seasonal or cyclic patterns from the series, because the features tend to gain different degrees of attention over time. Unfortunately, this set of time series do not exhibit clear seasonal or cyclic patterns. Generally, they exhibit similar trends with the total number of reviews $N(t)$. The reason could be the magnitude-level difference between the total number of reviews and the number of reviews relevant to a feature, which diminishes the fluctuation of per-feature series.

This observation further inspires us to investigate the percentage of reviews relevant to each feature, which we denote as *percentage time series* $X_f(t) = N_f(t)/N(t)$, shown in Fig.2(c). As expected, the percentage time series exhibit explicit seasonal and cyclic patterns over three consecutive years on the feature *nutritious*.

3.3 Validating Percentage Time Series

To validate the seasonal effect of all product features on average rather than *nutritious* alone, we conduct time series decomposition on the percentage time series $X_f(t)$ for each of the r features $f \in \mathcal{F} = \{f_1, f_2, \cdots, f_r\}$. Without loss of generality, we use $X(t)$ to denote the time series of an arbitrary feature. In line with the literature [3, 26], we decompose a time series $X(t)$ into three components in an additive manner, which are trend-cycle $T(t)$, seasonal component $S(t)$, and random effect $R(t)$:

$$X(t) = T(t) + S(t) + R(t) \tag{1}$$

The trend-cyclic component $T(t)$ captures long-term increase or decrease in the data, as well as the rises and falls that are not of fixed period, and it is reconstructed with the Moving Average (MA) [3] process:

$$T(t) = \frac{1}{2s+1} \sum_{j=-s}^{s} X(t+j) \tag{2}$$

The seasonal component $S(t)$ captures the fixed period influence of seasonal factors like quarter, month, or day of a week, and it is estimated by averaging the difference values $X(t) - T(t) = S(t) + R(t)$ for a year over the (three) years provided in the data set. This is because the seasonal variation is by definition constant from one year to the next, and the white noise residual $R(t)$ can be averaged out.

Finally, the random effect $R(t) = X(t) - T(t) - S(t)$ is constructed by substracting trend-cyclic and seasonal effects from the time series.

Fig.3 shows the decomposition for the percentage time series of *nutritious*, where the scale used for moving average is half a year ($s = 182$ in Eq.(2)). The trend-cyclic component indicates that the overall attention towards the feature *nutritious* decreases first and then rises, while the seasonal component presents a clear and regular seasonal fluctuation over years. The amplitudes of the seasonal signal over years turn approximately equal when the trend basis is removed.

The seasonal component $S(t)$ comprises the major part of energy in the time series, and this is of key importance in the following part of time series estimation and prediction. As a result, we fit the seasonal component of each feature f by a simple one-order Fourier series with a fixed period $m = 365$ to investigate the seasonality of the features:

$$S(t) \approx \hat{S}(t) = a + A \sin\left(\frac{2\pi t}{m} + \varphi\right) \tag{3}$$

We estimate the parameters by minimizing the Akaike Information Criterion corrected (AICc) [4], which is a corrected version of AIC for finite size samples, and has long been used as a measure of the relative quality of a statistical model for a given set of data in time series analysis:

$$\text{AICc} = \underbrace{2k + n \ln\left(\frac{1}{n}\sum_{t=1}^{n}\left(S(t) - \hat{S}(t)\right)^2\right)}_{\text{AIC}} + \frac{2k(k+1)}{n-k-1} \tag{4}$$

where k is the number of parameters, and n is the total number of data instances. Here we set $k = 3$ because we use a simple one-order Fourier series with three parameters.

For the seasonal series $S_f(t)$ of each feature f, we calculate the percentage of energy that a simple one-order Fourier can capture from the total amount of energy therein, which is:

$$p_f = \sum_{t=1}^{n} |\hat{S}_f(t)|^2 \Big/ \sum_{t=1}^{n} |S_f(t)|^2 \tag{5}$$

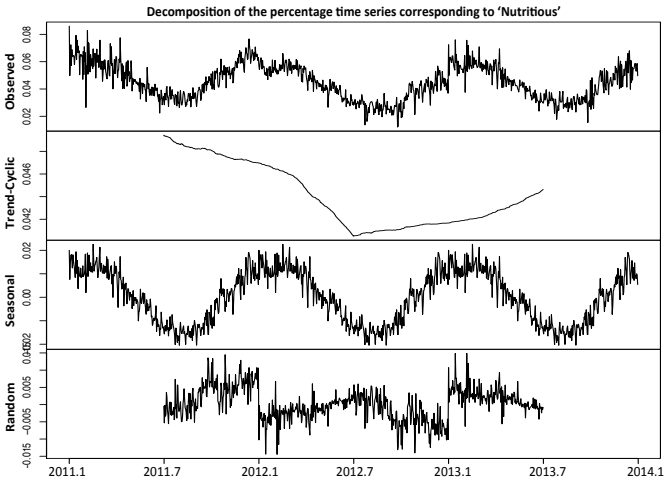


Figure 3: Decomposition of the percentage time series. From top to bottom are the original observed series, trend-cyclic component, seasonal component, and the random effect, respectively.

According to Parseval's theorem, a full-order Fourier transformation from time domain to frequency domain will preserve the energy, namely, $p_f = 1$, while $p_f < 1$ for an under-ordered Fourier due to the loss of information. In this study, we retain $r = 58$ features primarily by selecting the feature set with the highest F_1-score, which will be exposited with more detail in the experiments. The distribution of p_f for the 58 features in $\mathcal{F}$ is shown in Table 1.

Table 1: Statistical distribution of p_f of the features.

p_f	$0.7 \sim 0.8$	$0.8 \sim 0.9$	$0.9 \sim 1.0$
#Features	7	35	16
Percent(%)	12.1	60.3	27.6

We find that a simple one-order fixed-period Fourier is able to preserve more than 70% of the energy, and 87.9% of the features even got 80% or more energy preserved. The minimum and maximum p_f among the features are 0.753 and 0.966, respectively, and the average is 0.874.

The decomposation analysis above indicates that the percentage time series is capable of capturing the trend, cyclic, seasonal and random effects of a feature over time, and the decomposed seasonal component can be easily modeled with Fourier terms, which serves as an important basis for model estimation and prediction in the following. We thus adopt the percentage time series of each feature f primarily to investigate the feature-level dynamic preference of the users.

4. DAILY-AWARE TIME SERIES ANALYSIS

In this section, we present our model for daily-aware time series prediction. The Auto-Regressive Integrated Moving Average (ARIMA) process is widely adopted for monthly or yearly time series analysis, however, the year-long period (around 365) of daily time series makes the ARIMA models infeasible in practice due to the huge computational cost. To address the problem, we adopt Fourier series to capture

the seasonal effect in daily time series, and further leverage ARIMA process to model the residuals, which gives the Fourier-assisted ARIMA (FARIMA) model.

In the following, we briefly introduce the primary time series analysis (ARIMA) model first, and then integrate Fourier series into ARIMA to derive our FARIMA model.

4.1 Primary Time Series Analysis

The ARMA(p, q) process models a time series $X(t)$ with p Auto-Regressive (AR) terms and q Moving Average (MA) terms, which is given by:

$$\begin{aligned} X(t) &= \alpha_1 X(t-1) + \cdots + \alpha_p X(t-p) \\ &\quad + Z(t) + \beta_1 Z(t-1) + \cdots + \beta_q Z(t-q) \end{aligned} \tag{6}$$

where $Z(t) \overset{iid}{\sim} \Phi(0, \sigma^2)$ is independently identically sampled from a normal distribution with zero mean.

A simple example would be $X(t) \sim \text{ARMA}(1, 0)$, which is $X(t) = \alpha_1 X(t-1) + Z(t)$ that models the next data point as the previous one plus a white noise.

The successful application of ARMA(p, q) process requires the time series $X(t)$ be *stationary*, namely, for any positive integer $k \geq 1$ and any integer τ, the two discrete processes:

$$\begin{aligned} &\{X(t_1), X(t_2), \cdots, X(t_k)\}, \text{and} \\ &\{X(t_1+\tau), X(t_2+\tau), \cdots, X(t_k+\tau)\} \end{aligned} \tag{7}$$

have the same point distribution:

$$F_{t_1,t_2,\cdots,t_k}(X_1, X_2, \cdots, X_k) = P(X(t_1) < X_1, \cdots, X(t_k) < X_k) \tag{8}$$

Unfortunately, many time series in practice do not hold the stationary property, however, the d-order difference time series $W(t) \doteq \nabla^d X(t)$ may be stationary, where 1-order difference series is $\nabla X(t) = X(t) - X(t-1)$, and higher-order differences are iteratively defined as $\nabla^d X(t) = \nabla(\nabla^{d-1} X(t))$.

When $W(t) \doteq \nabla^d X(t)$ is modeled by an ARMA(p, q) process in Eq.(6), we denote $X(t)$ as modeled by an ARIMA(p, d, q) process.

4.2 Fourier-assisted ARIMA Model

The prediction of a future data instance in ARIMA(p, d, q) process requires to borrow information from those instances of at least a period ago to model the seasonal and cyclic effect in data fluctuation, which means that the number of parameters (especially the value of p) should be at least comparable to the period of the time series.

This is fine when processing monthly, quarterly, or yearly series because the periods (and thus the number of parameters) are small, which are usually 12 for monthly data and 4 for quarterly data. However, the approximate period of 365 of daily series makes it impractical for model estimation in ARIMA. Fortunately, the results on energy analysis in the previous section make it possible for us to leverage Fourier series to enhance ARIMA models on daily time series.

To do so, we use Fourier terms to model the seasonal pattern while using short-term ARIMA time series for the dynamic error, which is given by:

$$\begin{aligned} X(t) &\approx \underbrace{a + \sum_{k=1}^{K} \left[\alpha_k \sin\left(\frac{2\pi k t}{m}\right) + \beta_k \cos\left(\frac{2\pi k t}{m}\right) \right]}_{F(t)} \\ &\quad + \underbrace{\text{ARIMA}(p, d, q)(t)}_{E(t)} \doteq \hat{X}(t) = F(t) + E(t) \end{aligned} \tag{9}$$

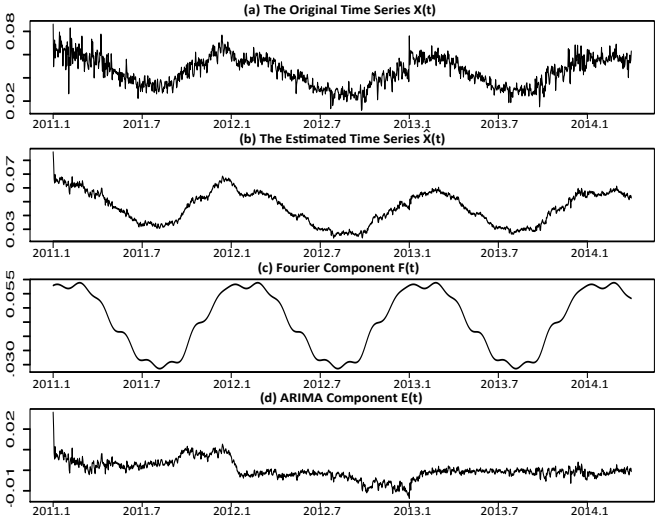


Figure 4: Fourier-assisted ARIMA estimation of a time series. (a) The original series $X(t)$; (b) The estimated series $\hat{X}(t) = F(t) + E(t)$; (c) The Fourier component $F(t)$; (d) The ARIMA component $E(t)$.

where $m = 365$ is the period, $F(t)$ is a K-order Fourier, and $E(t)$ is an ARIMA process that models the error of a Fourier approximation. The intuition here is that the annual seasonal shape is unchanged from year to year, so that Fourier terms can be used to model the annual seasonality, while the period of the residual series is reduced from $m = 365$ so that it can be modeled with a low-order ARIMA. This is shown in Fig.4, where the Fourier component $F(t)$ and ARIMA component $E(t)$ add up to the estimated series $\hat{X}(t)$.

The order of Fourier component K and ARIMA component (p, d, q) can be determined by minimizing the AICc, as shown by the procedure in Algorithm 1. The subroutines FOURIER and ARIMA fit the given time series with the specified orders by minimizing the AICc, and return the fitted series together with the corresponding AICc value achieved. In this work, we choose the FOURIER and ARIMA routines from the R language[2] in practical implementation.

5. DAILY-AWARE RECOMMENDATION

In this section, we leverage the predicted global preference (i.e., percentage time series) and the personal user/item preferences for daily-aware personalized recommendation. Table 2 lists the notations adopted for easy reference.

5.1 Feature Space Representation

The predicted percentage time series $\hat{X}_f(t)$ for each feature f is an indicator of the potential degree that users care about feature f at the given time t, which serve as the global preference of the users. There are many approaches in practice to describe the global preference of users towards the features based on the results of time series prediction. In this work, however, we adopt the approach of integrating $\hat{X}_f(t)$ into a vector of covariates directly to keep the simplicity, and also to fit with the conditional opportunity model in the following subsection. More precisely, we construct the *global covariate vector*:

$$\mathbf{x}^g(t) = [x_1^g(t), x_2^g(t), \cdots, x_r^g(t)]^{\mathrm{T}} \tag{10}$$

[2] http://www.r-project.org/

where the $k^{th}(1 \le k \le r)$ element $x_k^g(t) = \frac{1}{1+\exp(-\hat{X}_{f_k}(t))}$ is the sigmoid normalization of $\hat{X}_{f_k}(t)$ for feature $f_k \in \mathcal{F}$.

Besides, different users may concern about different features, and to profile a user u in the feature space, we consider all the historical reviews of the user. Suppose the k^{th} feature f_k is mentioned for $w_k^u(t)$ times by user u, then we profile this user as a *user covariate vector*:

$$\mathbf{x}^u(t) = [x_1^u(t), x_2^u(t), \cdots, x_r^u(t)]^{\mathrm{T}} \tag{11}$$

where $x_k^u(t) = \frac{1}{1+\exp(-w_k^u(t))}$ is the normalization of $w_k^u(t)$.

Similarly, we profile an item as an *item covariate vector* according to its corresponding reviews:

$$\mathbf{x}^i(t) = [x_1^i(t), x_2^i(t), \cdots, x_r^i(t)]^{\mathrm{T}} \tag{12}$$

where $x_k^i(t) = \frac{1}{1+\exp(-w_k^i(t))}$, and item i is commented for $w_k^i(t)$ times on feature f_k.

Finally, we construct the covariate vector $\mathbf{x}(t)$ for user-item pair (u, i) by integrating the global preference $\mathbf{x}^g(t)$, personal user preference $\mathbf{x}^u(t)$, and item profile $\mathbf{x}^i(t)$:

$$\mathbf{x}(t) = \left[\mathbf{x}^g(t)^{\mathrm{T}}, \mathbf{x}^u(t)^{\mathrm{T}}, \mathbf{x}^i(t)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{13}$$

In practical applications, our framework is flexible to integrate other available information into the covariate vector for model learning and prediction. Such information could be both user related (*e.g.* demographics), item related (*e.g.* content information), or even global factors.

5.2 Conditional Opportunity Model

To leverage the covariate vectors for prediction and recommendation in continuous rating scale, we adapt the conditional opportunity model used for follow-up purchase prediction in [32]. The probability density function of numerical rating y is modeled with a standard Weibull distribution:

$$p(y) = \gamma\lambda y^{\gamma-1}\exp(-\lambda y^{\gamma}) \tag{14}$$

Given the covariate vector $\mathbf{x}(t)$, we re-parameterize the scale parameter $\lambda = \exp(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}(t))$ based on the regression parameter $\boldsymbol{\theta}$ and the covariates $\mathbf{x}(t)$ in the following opportunity model:

$$p(y) = \gamma \exp\big(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}(t)\big) y^{\gamma-1} \exp\big(-\exp\big(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}(t)\big) y^{\gamma}\big) \tag{15}$$

For each user u, we sample the regression parameter $\boldsymbol{\theta}_u$ from a Gaussian distribution $\boldsymbol{\theta}_u \sim \Phi(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and sample the shape parameter γ_u from a Gamma distribution $\gamma_u \sim \Gamma(a, b)$. We use $D = \{(y|u, i, t)\}_{k=1}^n$ to denote a set of n user-item rating instances for model learning, where an instance $(y|u, i, t)$ means that user u rated item i with the rating y at time t. Based on this, we estimate the conditional probability of a single instance $(y|u, i, t)$ in the following conditional opportunity model:

$$\begin{aligned} p(y|u,i,t) &= p\big(y|\boldsymbol{\theta}_u, \gamma_u, \mathbf{x}(t)\big) \\ &= \gamma_u \exp\big(\boldsymbol{\theta}_u^{\mathrm{T}}\mathbf{x}(t)\big) y^{\gamma_u - 1} \exp\big(-\exp\big(\boldsymbol{\theta}_u^{\mathrm{T}}\mathbf{x}(t)\big) y^{\gamma_u}\big) \end{aligned} \tag{16}$$

Let $\phi = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, a, b)$, and let $\xi_u = \{\boldsymbol{\theta}_u, \gamma_u\}$ be the hidden variables for user u; $y_u = \{y_{u,1}, y_{u,2}, \cdots, y_{u,N_u}\}$ be the ratings rated by user u, then the likelihood of the observations in D can be written as a function of ϕ:

$$p(D|\phi) = \prod_{u=1}^{|\mathcal{U}|} p(y_u|\phi) = \prod_{u=1}^{|\mathcal{U}|} \int p(\xi_u, y_u|\phi) d\xi_u \tag{17}$$

where $\mathcal{U}$ is the set of users in the system.

Maximizing the likelihood $p(D|\phi)$ is equivalent to maximizing the log-likelihood function $L(\phi) = \ln p(D|\phi)$. In this work, we adopt the same EM approach for model learning as in [32], which the reader may refer to for detailed derivation of the EM updating rules.

5.3 Rating Prediction and Recommendation

With the predicted covariate vector $\mathbf{x}(t)$ given by the FARIMA model, as well as the estimated regression parameters $\boldsymbol{\theta}_u$ and shape parameter γ_u for each user given by the conditional opportunity model, we make daily-aware rating prediction by selecting the y that maximizes the conditional probability $p\big(y|\boldsymbol{\theta}_u, \gamma_u, \mathbf{x}(t)\big)$ in Eq.(16). The closed form solution can be easily found with the derivative of the Weibull distribution in Eq.(14):

$$p'(y) = \gamma\lambda y^{\gamma-2}\exp(-\lambda y^{\gamma})(\gamma - 1 - \lambda\gamma y^{\gamma}) \tag{18}$$

Table 2: A summary of the notations used.

$\mathcal{F}$	$\mathcal{F} = \{f_1, f_2, \cdots, f_r\}$ is the product feature set
$\hat{X}_f(t)$	The value of the predicted percentage time series for feature f at time t
$\mathbf{x}(t)$	The integrated covariate vector that contains the global, user, and item covariates
λ, γ	Scale/shape parameter of Weibull distribution
$\boldsymbol{\theta}$	The regression parameter for covariate vectors
$\boldsymbol{\mu}, \boldsymbol{\Sigma}$	Gaussian distribution parameters for sampling $\boldsymbol{\theta}$
a, b	Gamma distribution parameters for sampling γ
ξ_u	$\xi_u = \{\boldsymbol{\theta}_u, \gamma_u\}$ is the hidden variables for a user u
D	$D = \{(y\|u, i, t)\}_{k=1}^n$ is a set of n data instances, where $(y\|u, i, t)$ is a data instance
y_u	$y_u = \{y_{u,1}, y_{u,2}, \cdots, y_{u,N_u}\}$ is the set of ratings in D rated by user u
$\hat{y}_{ui}(t)$	The predicted rating of user u to item i at time t

Algorithm 1: FARIMA$(X(t), m, \bar{K}, \bar{p}, \bar{d}, \bar{q})$

Input: Time series $X(t)$, Period $m = 365$,
Maximum possible order of Fourier: $\bar{K}$,
Maximum possible order of ARIMA: $\bar{p}, \bar{d}, \bar{q}$
Output: Fitted time series $\hat{X}(t)$

```
minAICc ← ∞; F̂(t);
for K ← 1 to K̄ do
    {F(t), AICc} ← FOURIER(X(t), m, K);
    if AICc < minAICc then
        F̂(t) ← F(t);
        minAICc ← AICc;
    end
end
minAICc ← ∞; Ê(t);
for (p, d, q) in [0, p̄] × [0, d̄] × [0, q̄] do
    {E(t), AICc} ← ARIMA(X(t) − F̂(t), p, d, q);
    if AICc < minAICc then
        Ê(t) ← E(t);
        minAICc ← AICc;
    end
end
return X̂(t) ← F̂(t) + Ê(t);
```

Table 3: Cumulative statistical information in each quarter, where the statistics for a quarter represent the number of users, items and reviews in the system by the end of that quarter (included).

Year	2011				2012				2013				2014
Quarter	Q_1	Q_2	Q_3	Q_4	Q_1	Q_2	Q_3	Q_4	Q_1	Q_2	Q_3	Q_4	Q_1
#Users	72,403	164,256	280,918	500,619	701,975	925,649	1,070,542	1,239,967	1,355,395	1,417,551	1,575,223	1,710,040	1,844,569
#Items	3,559	5,474	7,961	11,235	14,265	20,444	28,746	37,380	42,184	46,877	49,710	52,117	93,243
#Reviews	119,517	305,974	571,602	1,108,673	1,667,607	2,409,277	3,015,849	3,656,338	4,100,668	4,253,294	4,656,628	4,963,927	5,524,491

By setting $p'(y) = 0$ and $\lambda = \exp(\boldsymbol{\theta}_u^{\mathrm{T}}\mathbf{x}(t))$, the predicted rating for user u towards item i at time t will be:

$$\hat{y}_{ui}(t) = \left(\frac{\gamma_u - 1}{\gamma_u \exp(\boldsymbol{\theta}_u^{\mathrm{T}}\mathbf{x}(t))}\right)^{\frac{1}{\gamma_u}} \tag{19}$$

In practice, we construct the time-aware personalized recommendation list on daily resolution for user u by ranking the unreviewed items in descending order of the predicted ratings $\hat{y}_{ui}(t)$.

6. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of product feature extraction, daily-aware time series prediction, dynamic personalized recommendation, and the performance of our framework in handling with cold-start users.

6.1 Experimental Setup

We obtained the real-world reviews in the cosmetic domain of JD.com, the second largest e-commerce website in China, from Jan. 1st, 2011 to Mar. 31st, 2014. The dataset covers all the reviews of three consecutive years and a quarter, which is sufficient for daily-aware time series modeling. The statistical information of cumulative per-quarter data is listed in Table 3, where the statistics of a specific quarter represents the total number of users, items, or reviews in the system until the end of that quarter.

To simulate the practical system operation over time, and also to evaluate the effectiveness of our framework over a whole year, we construct four time-dependent training-testing datasets $D_1 \sim D_4$ based on the last four quarters of data. Specifically, each of the dataset takes the reviews in one of the last four quarters (i.e. $Q_2 \sim Q_4$ of 2013, and Q_1 of 2014) as testing set, and takes the reviews in the previous quarters as training set, which is denoted in Table 4. In the following, we evaluate and report the performance of both time series prediction and personalized recommendation on each of the four datasets, as well as the average performance of the four quarters for the yearly performance.

In the following experiments, we set the maximum possible order of Fourier and ARIMA components as $\bar{K} = 10$ and $(\bar{p}, \bar{d}, \bar{q}) = (10, 3, 10)$ in Algorithm 1, respectively, as we find this is sufficient to achieve satisfactory performances. We introduce the comparative algorithms and evaluation metrics for each experimental task separately in the following,

Table 4: Four time-dependent datasets constructed for model training and testing.

Dataset	Train	Test
D_1	$2011 + 2012 + Q_1, 2013$	$Q_2, 2013$
D_2	$2011 + 2012 + Q_1, Q_2, 2013$	$Q_3, 2013$
D_3	$2011 + 2012 + Q_1, Q_2, Q_3, 2013$	$Q_4, 2013$
D_4	$2011 + 2012 + 2013$	$Q_1, 2014$

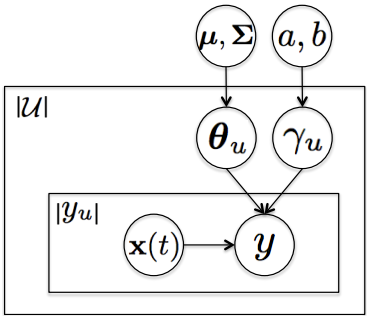


Figure 5: The dependency of variables for daily-aware recommendation. The i^{th} rating $y_{u,i}$ of user u is dependent on the conditional opportunity model $\xi_u = \{\boldsymbol{\theta}_u, \gamma_u\}$ of user u and the integrated covariate vector $\mathbf{x}(t)$. Models of different users share information through the same prior $\phi = (\boldsymbol{\mu}, \boldsymbol{\Sigma}, a, b)$.

because the evaluation of time series prediction and personalized recommendation require different settings.

6.2 Product Feature Extraction

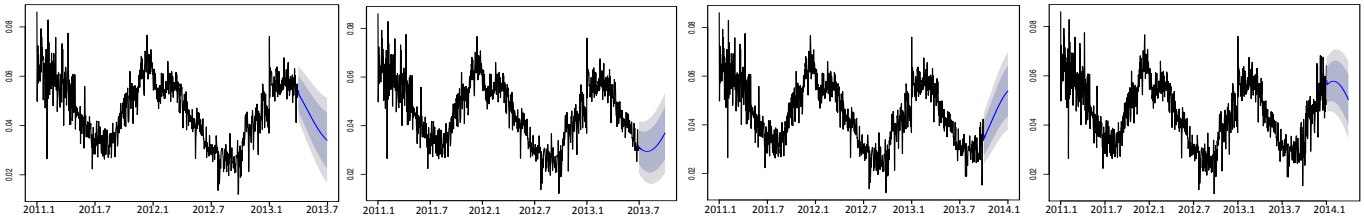
We first extract product features in cosmetic domain based on all the textual reviews in the whole dataset. Similar with many other text mining tasks, the automatic extraction of product features from free-text reviews is inherently difficult and have to trade off between precision and recall.

In this work, we tune the parameters in the method introduced in [22, 39] and record the precision and recall. In line with the literature [18, 39], we construct the gold-standard feature set by sampling 1,000 reviews and presenting them to 3 human annotators for manual feature extraction. The gold-standard feature set is constructed by selecting the common features of the three manually extracted sets, and it is further used to evaluate the automatically extracted features. The averaged pair-wise agreement between annotators is 82.46%, and the evaluation results for several trials of automatic feature set extraction are shown in Table 5.

Table 5: Evaluation of automatic feature extraction when trading off between precision and recall.

Trial	1	2	3	4	5	6
#Features	32	44	58	66	79	95
Precision	**0.9063**	0.8864	0.8621	0.7879	0.6962	0.6316
Recall	0.3867	0.5200	0.6667	0.6933	0.7333	**0.8000**
F_1-score	0.5421	0.6555	**0.7519**	0.7376	0.7143	0.7059

We select the result with the highest F_1-score primarily (Trial #3), which contains 58 features. In general, we tend to keep a high precision while choosing an acceptable recall to reduce the probability of introducing noise into time series prediction and personalized recommendation. We adopt the automatically constructed feature set without incorporating any human annotation information in the following experiments, thus to guarantee that no manual effort is included in our framework.



(a) $D_1 : K, p, d, q = 6, 4, 1, 3$ (b) $D_2 : K, p, d, q = 6, 4, 1, 5$ (c) $D_3 : K, p, d, q = 6, 3, 1, 2$ (d) $D_4 : K, p, d, q = 7, 3, 1, 2$

Figure 6: Exampled time series prediction results on the four datasets $D_1 \sim D_4$, where K is the final order of the Fourier component, and (p, d, q) is the order of ARIMA component. The orders are automatically selected by minimizing the AICc. The blue line is the predicted series for the testing set, and the dark grey and light grey boundaries are 80% and 95% prediction intervals of the random effect, respectively.

6.3 Daily-Aware Time Series Prediction

To evaluate the performance of FARIMA model in time series prediction, we conduct experiment on each of the four datasets $D_1 \sim D_4$. For each dataset, we adopt the training set for model learning based on Algorithm 1, and predict the time series of the following quarter in testing set. Figure 6 examples the time series prediction on the four datasets, where the listed Fourier order K and ARIMA order (p, d, q) are determined by minimizing the AICc automatically. We can see intuitionally that our FARIMA model is not only capable of capturing the future trends in rise and fall (*e.g.*, in Figure 6(a) and 6(c)), but also able to predict the future inflexions ahead of time (*e.g.*, in Figure 6(b) and 6(d)).

For numeral evaluation of the prediction accuracy, we adopt the Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), which are defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(X(t) - \hat{X}(t))^2}{n}}, \text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{X(t) - \hat{X}(t)}{X(t)}\right| \tag{20}$$

where n is the number of instances in the testing set, $X(t)$ is the actual value and $\hat{X}(t)$ is the forecasted value. The RMSE is a measure of the absolute error of prediction that has been widely adopted in many tasks, however, in the task of time series prediction the values of $X(t)$ are not restricted to a fixed range; as a result, we further adopt MAPE as an indicator of relative prediction deviation. The averaged results over the features for each dataset as well as the total averages across datasets are shown in Table 6.

Table 6: Evaluation of time series prediction.

Dataset	D_1	D_2	D_3	D_4	Average
#Instance	4,253,294	4,656,628	4,963,927	5,524,491	4,849,585
RMSE	0.00406	0.00453	0.00704	0.00603	0.00541
MAPE	0.08012	0.11962	0.12580	0.07770	0.10081
∼MAE	0.3205	0.4785	0.5032	0.3108	0.4032

The results on both RMSE and MAPE indicate that our FARIMA model is able to achieve quite accurate predictions of future time series. It may be easier for one to get a more intuitional understanding of the results by rescaling them into the frequently seen range of $1 \sim 5$ stars. By multiplying MAPE with the rating range (i.e., 4), we get the estimated MAE as shown in the last row of Table 6, which implies a superior result in prediction accuracy.

The results are statistically significant due to the huge amount of instances for testing. In the following experiments, we leverage the predicted percentage time series in each dataset for daily-aware recommendation.

6.4 Daily-Aware Rating Prediction

With the predicted daily time series on each feature, we preform daily-aware rating prediction and recommendation with the conditional opportunity model.

In real-world systems, the number of users and products are increasing with the business growth of the e-commerce. This is shown by the continuously growing number of users, item, and reviews in Table 3, which leads to the frequently noted problem of cold-start in recommender systems, especially when we are simulating the time effect by splitting training and testing sets according to time, because a user/item may have not even a single record in the training set, while we have to predict his/her ratings in the testing set, which may cause some comparative algorithms to fail.

As a result, for each dataset D_i, we only consider those users/items in the testing set that also exist in the training set. We will experiment on the performance of cold-start recommendation separately in the following sections.

We compare our daily-aware recommendation method based on Feature-level Time Series Analysis (FTSA) with the following comparative algorithms:

NMF: Non-negative Matrix Factorization [17] algorithm that is among the best MF algorithms in previous research [28]. This method is a baseline for time-independent CF.

timeSVD++: The state-of-the-art time-aware collaborative prediction method proposed by Koren [16]. For easy comparison, we adopt the open-source implementation provided by MyMediaLite [12].

Tensor: The tensor factorization approach [15] for time-aware prediction, which is widely adopted for context-aware recommendation. For daily prediction, we incorporate 366 days of a year as the time dimension of a tensor, and a user-item rating made on the i^{th} day of a year is placed in the rating matrix on the i^{th} layer.

EFM: The Explicit Factor Model for explainable recommendation proposed in [38], which is time-independent but it is the state-of-the-art CF method based on sentiment analysis on textual reviews as in this work.

To investigate the effect of incorporating different number of features r, we make use of the six sets of extracted features $\mathcal{F} = \{f_1, f_2, \cdots, f_r\}$ in the different trials on the experiment of product feature extraction (Table 5). Specifically, we tune $r = 32, 44, 58, 66, 79, 95$ in our FTSA approach by using the corresponding feature set for time series analysis and rating prediction. To ensure comparable model complexity, we take the same r as the number of latent factors in NMF, timeSVD++ and tensor factorization, and we use the same feature set in the EFM method.

We adopt the frequently used RMSE for evaluation. The hyper-parameters for each method are selected by grid search on the four datasets and retaining those with the best performance. The averaged RMSE of the four datasets is shown in Figure 7, and Table 7 shows the best RMSE achieved by each method on each dataset, as well as their averages.

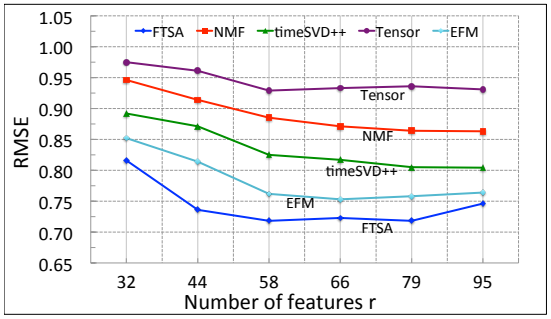


Figure 7: Averaged RMSE over $D_1 \sim D_4$ against the number of features/factors for different methods.

We see that for the latent factor based methods (NMF, timeSVD++ and Tensor), the RMSE decreases with the increase of the number of latent factors used r. However, we find that the time-aware tensor factorization approach did not gain superiority against the static NMF. This may be of the reason that splitting the rating records into the days of a year makes the training instances sparse, because the time-aware SVD (timeSVD++) gains superior results than NMF by integrating the advantages of time factors and latent factorization.

Surprisingly, a static model that takes advantage of textual reviews (EFM) is even better than a dynamic modeling approach (timeSVD++) that only makes use of the numerical ratings. This indicates the importance of incorporating textual reviews as an information source for CF. However, by further incorporating the time factors for dynamic CF through time series analysis on product features, our FTSA approach gains consistently better results than EFM.

We noticed that the performance of FTSA tends to drop slightly when the number of product features increases to 95, in which case too many noise features are incorporated (Table 5). As a result, it would be important to construct a reliable feature set for FTSA to gain the best performance in practical applications. However, our FTSA model achieves superior result in a large range of feature selections.

To validate the effect of our FARIMA model in time series prediction, we replace the predicted time series $\hat{X}_{f_k}(t)$ in Eq.(10) with the real value $X_{f_k}(t)$, and still conduct rating prediction with our conditional opportunity model. The results are shown by FTSA$'$ in Table 7. We find that our predicted percentage time series is even better than using the real values. This may indicate that the smoothing effect of our FARIMA model is important for noise filtering. For example, a feature may be commented for only a few times in a specific day due to the relatively fewer active users on that day. However, by leveraging the data accumulated through the previous years, we are able to model the dynamic user preferences more accurately.

6.5 Daily-Aware Top-K Recommendation

We also care about the performance of our method in top-K recommendation, as is important in real-world scenarios.

Table 7: The best RMSE achieved on four datasets and their averaged value for each method, standard deviations ≤ 0.015 for each experimental run.

Dataset	D_1	D_2	D_3	D_4	Average
NMF	0.857	0.866	0.851	0.878	0.863
timeSVD++	0.782	0.813	0.795	0.826	0.804
Tensor	0.912	0.937	0.916	0.951	0.929
EFM	0.733	0.768	0.742	0.769	0.753
FTSA	**0.706**	**0.729**	**0.711**	**0.726**	**0.718**
FTSA$'$	0.716	0.732	0.722	0.735	0.726

To evaluate top-K recommendation, we select those users who have 10 or more records in the testing set for each dataset D_i, and then conduct top-10 recommendation for them. Following [38, 23], we adopt the frequently used NDCG to evaluate top-10 recommendation, and we still use the NMF, timeSVD++, Tensor and EFM methods for performance comparison. We set $r = 58$ for all methods.

Table 8 shows the evaluation results on different datasets and their averages. It is shown that time-aware approaches (timeSVD++ and Tensor) are consistently better than static models (NMF and EFM). This indicates that considering the time factors may be more important for top-K recommendation than in rating prediction tasks. By taking advantage of both product features from textual reviews and dynamic time series analysis, our FTSA approach achieves the best NDCG among the comparative algorithms.

Table 8: The number of selected users and best NDCG on four datasets, standard deviations ≤ 0.01.

Dataset	D_1	D_2	D_3	D_4	Average
#Users	122,476	146,099	159,747	167,370	148,923
NMF	0.154	0.136	0.163	0.147	0.150
timeSVD++	0.228	0.241	0.236	0.233	0.235
Tensor	0.207	0.192	0.196	0.211	0.202
EFM	0.186	0.194	0.208	0.195	0.196
FTSA	**0.254**	**0.267**	**0.258**	**0.271**	**0.263**

This experimental result verifies our observations to some extent, that users tend to buy different categories/types of products at different time of a year, although they may make similar ratings towards the products. To further investigate users' purchasing behavior on different products, we conduct top-K recommendation for products of different popularities separately. To do so, we select those items that have more than L reviews in dataset D_i (including both training and testing set), and filter out the remaining 'low frequency' items. Based on the filtered items, we still select those users with 10 or more reviews in the testing set to conduct top-10 recommendation. The averaged NDCG of the four datasets under different filtering thresholds L is shown in Fig.8.

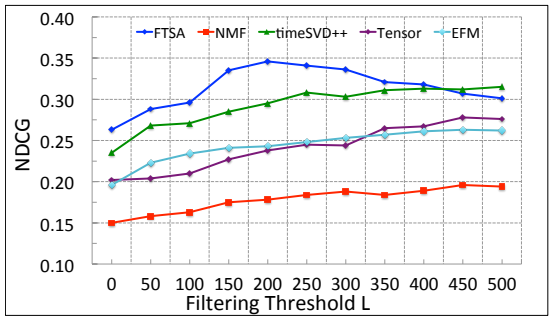


Figure 8: Averaged NDCG over $D_1 \sim D_4$ against the filtering threshold L for different algorithms.

We see that the results on NDCG for both dynamic algorithms (timeSVD++ and Tensor) and static methods (NMF and EFM) tend to rise gradually with the increase of filtering threshold L. This is not surprising because the training set becomes denser by selecting the popular items, which gives us more accurate predictions. Besides, it is generally easier to recommend popular items than long-tail ones. However, the results for FTSA is interesting because the performance tends to drop when L is too high. This indicates that our method achieves superior results against others for low- or mid-frequency items. After careful examination of the high frequency items, we find that they tend to be seasonally independent and are welcomed all over the year, *e.g.*, a major part of them are lipsticks. Thus forcing a seasonal factor on these products may result in inferior results. However, our method is still favoured in a large range of filtering thresholds. Besides, we usually need to consider all the products rather than only the popular ones in practical systems.

6.6 Recommendation for Cold-Start Users

Previous work of personalized recommendation usually filter out the cold-start users for evaluation, *e.g.*, those users that have less than 20 reviews. However, this may not be feasible for real-world systems because the most difficult users are not properly considered, although they may constitute a major part of the system. For example, [40] reported that about 49% of the users in Yelp made only a single review. As a result, we evaluate the performance on cold-start users separately in this section.

For each dataset D_i, we still conduct rating prediction and top-10 recommendation using the training set, but we only evaluate with the users that have at most one review in the training set. This setting fits with the real scenarios in practical systems, where we can adopt all the records in the system for model learning, and make recommendation for the cold-start users. The number of selected users for testing, as well as the RMSE for rating prediction and NDCG for top-10 recommendation are shown in Table 9.

Table 9: The number of users for evaluation, and the RMSE and NDCG on the four datasets, standard deviations ≤ 0.05 for RMSE and ≤ 0.02 for NDCG.

Metric	Dataset	D_1	D_2	D_3	D_4	Average
	#Users	527,328	637,965	613,904	712,003	622,800
R	NMF	1.729	1.693	1.732	1.706	1.715
M	timeSVD++	1.441	1.438	1.506	1.475	1.465
S	Tensor	1.763	1.779	1.725	1.769	1.759
E	EFM	1.383	1.362	1.360	1.389	1.374
	FTSA	**1.344**	**1.287**	**1.278**	**1.326**	**1.309**
N	NMF	0.067	0.073	0.061	0.066	0.067
D	timeSVD++	0.084	0.075	0.088	0.092	0.085
C	Tensor	0.046	0.054	0.044	0.048	0.048
G	EFM	0.093	**0.107**	0.095	0.087	0.096
	FTSA	**0.118**	0.098	**0.122**	**0.115**	**0.113**

In general, the prediction accuracy (RMSE) for cold-start users exhibits similar patterns with that of the total users (Table 7). The reviews-based approaches (EFM and FTSA) outperform the rating-based approaches (NMF, timeSVD++ and Tensor). The reason could be the inherent difficulty for rating-based methods to estimate accurate rating biases of a user with insufficient training data, thus they tend to predict the ratings of cold-start users around the global mean. For example, the regularization terms in NMF, timeSVD++ and tensor factorization actually force the predictions to be centred under a Gaussian prior, which makes the predictions less 'personalized'. However, by leveraging the product features embedded in reviews, we are able to expand the available information for prediction. We find that on average there are 0.66 ratings per user for the cold-start users, while the averaged number of features per user is 2.56, because a user may comment on more than one features in a single review. When taking time series factors into consideration, our FTSA approach achieves the best performance.

On the result of NDCG, we see that the static NMF approach is still better than the dynamic tensor factorization approach, which is different from the relatively 'warm-start' scenarios in Table 8. This is in fact not surprising because it is difficult for the tensors to learn time-shifting factors and even global time-specific mean ratings when so few as only a single rating is available on the time dimension. When the valid information is enriched with product features, our FTSA method gains superior results on most detests.

7. CONCLUSIONS

In this paper, we propose to leverage direct time series analysis for dynamic daily-aware recommendation. In order to overcome the problem of data insufficiency in previous product-level modeling approaches, we propose to extract product features automatically from user reviews, and thus to degrade product-level modeling to feature-level modeling. To make it computationally feasible for time series analysis on daily resolution, we develop the Fourier-assisted Auto-Regressive Integrated Moving Average (FARIMA) approach for percentage time series fitting and prediction, and we further adapt the conditional opportunity model for personalized rating prediction and recommendation.

For the first time, our analysis of time series in e-commerce characterizes the trend, seasonal, cyclic, and random effects of the product features that online users concern. The observation that a large amount of features exhibit clear seasonal and cyclic patterns may be inspiring for the construction of intelligent automatic marketing tools. Experimental results on both rating prediction and top-K recommendation, as well as cold-start performance verifies the superiority of our feature-level time series analysis approach.

This is a first step towards leveraging the rather mature technique of time series analysis in other research fields like Economics, to the research of personalized recommender systems, and there is much room for improvements. Apart from cosmetics, we will investigate other product domains with sufficient data, and even take geometrical information into consideration. Besides the conditional opportunity model, we can also develop other factorization, graphical, or even deep learning models to integrate feature-level time series for personalized recommendation. The cross-adaption of other models may even see more interdisciplinary research achievements between Economics and Artificial Intelligence.

Acknowledgement

This work was supported by National Key Basic Research Program (2015CB358700) and Natural Science Foundation (61472206, 61073071) of China, and Yi is sponsored by the National Science Foundation under grant CCF-1101741 and IIS-0953908. Yongfeng would also like to thank the Baidu and IBM Ph.D. Fellowship program, and the Siebel Scholar program for their kind supports.

8. REFERENCES

[1] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-Aware Recommender Systems. *Recommender systems handbook*, pages 217–253, 2011.

[2] L. Baltrunas and X. Amatriain. Towards Time Dependant Recommendation based on Implicit Feedback. *CARS*, 2009.

[3] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. Time Series Analysis: Forecasting and Control. *John Wiley & Sons*, 2013.

[4] K. P. Burnham and D. R. Anderson. Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. *Springer*, 2002.

[5] P. G. Campos, F. Díez, and I. Cantador. Time-aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols. *User modeling & user-adapted interaction*, 24:67–119, 2014.

[6] T. Chen, W. Han, H. Wang, Y. Zhou, B. Xu, and B. Zang. Content Recommendation System based on Private Dynamic User Profiling. *ICMLC*, 2007.

[7] W. Chen, W. Hsu, and M. Lee. Modeling User's Receptiveness Over Time for Recommendation. *SIGIR*, pages 373–382, 2013.

[8] H. Choi and H. Varian. Predicting the Present with Google Trends. *Economic Record*, 88(s1):2–9, 2012.

[9] W. Chu and S. Park. Personalized Recommendation on Dynamic Content Using Predictive Bilinear Models. *WWW*, pages 691–700, 2009.

[10] X. Ding, B. Liu, and P. S. Yu. A Holistic Lexicon Based Approach to Opinion Mining. *WSDM*, 2008.

[11] G. Dror, N. Koenigstein, and Y. Koren. Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. *RecSys*, 2011.

[12] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A Free Recommender System Library. *RecSys*, 2011.

[13] Z. Gantner, S. Rendle, and L. Schmidt-Thieme. Factorization Models for Context-/Time-Aware Movie Recommendations. *CAMRa*, pages 14–19, 2010.

[14] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User Profiles for Personalized Information Access. *The Adaptive Web*, pages 54–89, 2007.

[15] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering. *RecSys*, pages 79–86, 2010.

[16] Y. Koren. Collaborative Filtering with Temporal Dynamics. *KDD*, pages 447–455, 2009.

[17] D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. *NIPS*, 2001.

[18] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai. Automatic construction of a context-aware sentiment lexicon: An optimization approach. *WWW*, 2011.

[19] Z. Lu, D. Agarwal, and I. Dhillon. A Spatio Temporal Approach to Collaborative Filtering. *RecSys*, 2009.

[20] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, and H. Kato. A Recommendation Method Considering Users' Time Series Contexts. *ICUIMC*, 2009.

[21] M. J. Pazzani and D. Billsus. Content-Based Recommendation Systems. *The Adaptive Web LNCS*, pages 325–341, 2007.

[22] A. M. Popescu and O. Etzioni. Extracting Product Features and Opinions from Reviews. *EMNLP*, 2005.

[23] S. Rendle, C. Freudenthaler, Z. Gantner, and L. S. Thieme. BPR: Bayesian Personalized Ranking from Implicit Feedback. *UAI*, 2009.

[24] F. Ricci, L. Rokach, and B. Shapira. Introduction to Recommender Systems Handbook. *Springer US*, 2011.

[25] Y. Shi, M. Larson, and A. Hanjalic. Mining Mood-specific Movie Similarity with Matrix Factorization for Context-aware Recommendation. *CAMRa*, pages 34–40, 2010.

[26] R. H. Shumway and D. S. Stoffer. Time Series Analysis and Its Application. *Springer*, 2010.

[27] X. Su and T. M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 4, 2009.

[28] G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk. Investigation of Various Matrix Factorization Methods for Large Recommender Systems. *Proc. ICDM*, 2008.

[29] D. Tang, F. Wei, B. Qin, M. Zhou, and T. Liu. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. *COLING*, pages 172–182, 2014.

[30] C. Vaca, A. Mantrach, A. Jaimes, and M. Saerens. A Time-based Collective Factorization for Topic Discovery and Monitoring in News. *WWW*, 2014.

[31] J. Wang and Y. Zhang. Is It Time For a Career Switch? *WWW*, pages 1377–1387, 2013.

[32] J. Wang and Y. Zhang. Opportunity Models for E-commerce Recommendation: Right Product, Right Time. *SIGIR*, pages 303–312, 2013.

[33] Y. Wu and M. Ester. FLAME: A Probabilistic Model Combining Aspect Based Opinion Mining and Collaborative Filtering. *WSDM*, pages 199–208, 2015.

[34] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal Recommendation on Graphs via Long- and Short-term Preference Fusion. *KDD*, pages 723–731, 2010.

[35] D. Xu, Y. Liu, M. Zhang, S. Ma, A. Cui, and L. Ru. Predicting Epidemic Tendency through Search Behavior Analysis. *IJCAI*, pages 2361–2366, 2011.

[36] A. Yessenalina, Y. Yue, et al. Multi-level structured models for document-level sentiment classification. *EMNLP*, pages 1046–1056, 2010.

[37] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware Point-of-interest Recommendation. *SIGIR*, pages 363–372, 2013.

[38] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis. *SIGIR*, pages 83–92, 2014.

[39] Y. Zhang, H. Zhang, M. Zhang, Y. Liu, et al. Do Users Rate or Review? Boost Phrase-level Sentiment Labeling with Review-level Sentiment Classification. *SIGIR*, pages 1027–1030, 2014.

[40] Y. Zhang, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Understanding the Sparsity: Augmented Matrix Factorization with Sampled Constraints on Unobservables. *CIKM*, 2014.