Exploiting Temporal Dynamics in Product Reviews for Dynamic Sentiment Prediction at the Aspect Level

PEIKE XIA and WENJUN JIANG, Hunan University
JIE WU, Temple University
SURONG XIAO, Hunan University
GUOJUN WANG, Guangzhou University

Online reviews and ratings play an important role in shaping the purchase decisions of customers in e-commerce. Many researches have been done to make proper recommendations for users, by exploiting reviews, ratings, user profiles, or behaviors. However, the dynamic evolution of user preferences and item properties haven't been fully exploited. Moreover, it lacks fine-grained studies at the aspect level. To address the above issues, we define two concepts of user maturity and item popularity, to better explore the dynamic changes for users and items. We strive to exploit fine-grained information at the aspect level and the evolution of users and items, for dynamic sentiment prediction. First, we analyze three real datasets from both the overall level and the aspect level, to discover the dynamic changes (i.e., gradual changes and sudden changes) in user aspect preferences and item aspect properties. Next, we propose a novel model of Aspect-based Sentiment Dynamic Prediction (ASDP), to dynamically capture and exploit the change patterns with uniform time intervals. We further propose the improved model ASDP+ with a bin segmentation algorithm to set the time intervals non-uniformly based on the sudden changes. Experimental results on three real-world datasets show that our work leads to significant improvements.

CCS Concepts: • **Information systems → Information systems applications**; *Data mining*;

Additional Key Words and Phrases: Review mining, opinion evolution, aspect level, temporal dynamics, sentiment prediction

ACM Reference format:
Peike Xia, Wenjun Jiang, Jie Wu, Surong Xiao, and Guojun Wang. 2021. Exploiting Temporal Dynamics in Product Reviews for Dynamic Sentiment Prediction at the Aspect Level. *ACM Trans. Knowl. Discov. Data* 15, 4, Article 68 (April 2021), 29 pages.
https://doi.org/10.1145/3441451

This research was supported by the NSFC (grant 61632009), Guangdong Provincial NSF (grant 2017A030308006), Open Project of Zhejiang Lab (2019KE0AB02), the Science and Technology Program of Changsha City (kq2004017), and in part by NSF (grants CNS 1824440, CNS 1828363, and CNS 1757533).

Authors' addresses: P. Xia, W. Jiang (corresponding author), and S. Xiao, College of Computer Science and Electronic Engineering, Hunan University, South Lushan Road #2, Changsha, Hunan Province, 410082, China; emails: {xiapeike, jiangwenjun, surong}@hnu.edu.cn; J. Wu, Department of Computer and Information Sciences, Temple University, SERC 362, 1925 N. 12th Street, Philadelphia, PA 19122; email: jiewu@temple.edu; G. Wang, School of Computer Science, Guangzhou University, Guangzhou, Guangdong Province, P. R. China, 510006, China; email: csgjwang@gzhu.edu.cn.

1556-4681/2021/04-ART68 $15.00
https://doi.org/10.1145/3441451

1 INTRODUCTION

In recent years, online shopping has become increasingly popular [14]. Meanwhile, more and more users like to express their opinions online [39, 52]. Online reviews and ratings play an important role in shaping the purchase decisions of customers in e-commerce. Many recommendation models have been proposed by exploiting reviews, ratings, user profiles, or behaviors [6, 8, 15, 44]. Existing works are usually based on user preferences and product attributes, which are usually changing over time. However, the evolution hasn't been fully exploited in recommendation. Moreover, it lacks the fine-grained analysis in product reviews. In this article, we strive to exploit fine-grained information at the aspect level and the evolution of users and items, for dynamic sentiment prediction and recommendation.

Existing researches on user sentiment prediction can be generally classified into three categories. One is to mine users' overall or aspect sentiments from product comments. In the past, some researchers tried to make rating predictions by exploiting users' historical scoring data. Such as the classic matrix factorization (MF) and collaborative filtering methods [18, 36]. However, their consideration is not fine-grained enough, ignoring users' aspect information. Some recent works strive to extract aspects and aspect ratings from product reviews to understand users' aspect preferences [43, 53]. For example, Bauman et al. [1] recommend more valuable aspects to users by extracting aspect ratings and aspect weights from comments. However, works in this category usually neglect the evolution of users and items. The second category is to dynamically mine user sentiment from user rating and time information, i.e., to analyze how ratings change over time and model temporal changes in customer preferences [11, 12]. For example, Koren [17] divides the rating data into different time bins and uses each bin to model user preferences and item properties. However, works in this category only consider ratings. The last category is to combine the idea of the above two categories, which is also our research direction. There are some recent works attempting to combine temporal dynamic information and reviews information [22]. For example, [41] improves the accuracy of the recommendation by fusing the time model TimeSVD++ and the topic model TopicMF. However, works in this category don't consider the fine grained aspects. In a word, although many achievements have been done, the evolution of users and items haven't been well studied and exploited in sentiment prediction and recommendation.

In real life, the user preferences and item properties keep changing over time. For example, a user may give a relatively high rating when his mood is good, and vice versa. When a new item comes out, it may receive a higher rating and then turn to steady over time. Therefore, it is unwise to statically treat the user's historical behavior information [6, 30, 33, 48]. Going more deeply into the aspect level, the user aspect preferences and item aspect properties are also changing dynamically over time. Figure 1 shows an example of changing aspect preferences: when a user does not have enough money, he tends to care more about the "price" aspect; as his economic capability increases, he may pay more attention to the "appearance" or the "quality" aspect. The changes of item aspect properties are similar to that. When a product is just released, its "price" aspect may be of great concern, and after a period of time, users will pay more attention to its "quality" aspect. Therefore, it is necessary to capture and explore those fine-grained dynamic changes for better recommendation.

Although many efforts have been done, three challenges are still open. (1) Existing works lack a deep understanding of the evolution (dynamic features) of users and items. (2) It lacks comprehensive time dynamic analysis for users and items at the aspect level. (3) There is a pressing need to integrate aspect dynamic features of users and items into sentiment prediction.

Our motivations. Keeping the above challenges and the task of sentiment prediction in mind, our motivations in this article are threefold: (1) exploring a way to measure the evolution and to capture the dynamic features of both users and items; (2) deeply studying the fine-grained change

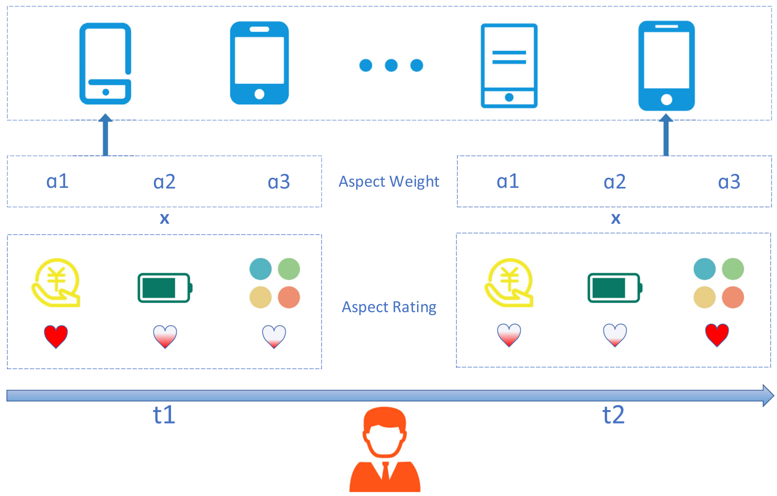


Fig. 1. The dynamic changes in user aspect preferences.

patterns of user's aspect preferences and item's aspect properties in real datasets; and (3) exploiting the evolution of aspect-level features of users and items to enhance the dynamic sentiment prediction.

In this article, we focus on fusing the temporal dynamics of user's aspect preferences and item's aspect properties from product reviews, so as to improve the accuracy of sentiment prediction for recommendation system. Our contributions are as follows:

(1) We identify a novel problem of aspect-based sentiment dynamic prediction (ASDP). To address the problem, we define the concepts of user maturity and item popularity to capture the dynamic features of users and items, and exploit them for better sentiment prediction.
(2) We deeply study the dynamic changes (i.e., gradual changes and sudden changes) in user aspect preferences and item aspect properties, with three real datasets from the overall-level and the aspect-level.
(3) We propose a new model of ASDP to dynamically capture the change patterns of user's aspect preferences and item's aspect properties with uniform time intervals. We also propose an improved ASDP+ model with a bin segment algorithm, which sets time intervals non-uniformly based on the sudden changes.
(4) We evaluate the proposed models on three real-world datasets and the results show that they lead to significant improvements compared with the baselines. For example, the improvements on the Beeradvocate dataset are about 2%–20% on the F1 score.

The rest of the article is organized as follows. In Section 2, we discuss the related work. Section 3 defines the ASDP problem and introduces the solution overview. In Section 4, we extensively study the change patterns of user aspect preferences and item aspect properties. Section 5 details the proposed method. Experiments with the three real-life datasets are described in Section 6. Section 7 concludes the article.

2 RELATED WORK

In this section, we briefly review the literature. Closely related works are on review text mining and dynamic evolution of user preferences and item properties.

2.1 Review Text Mining

Review text mining is essential to exploring the changes in user preferences and item properties. The review texts generally contain a lot of user and product information, which can tell us the reasons why users like or dislike this product [9, 14, 32]. By capturing these hidden information, the recommendation system can make more personalized recommendations. Therefore, in recent years, a large number of researchers are committed to mining useful information from review text [2, 24, 27, 34].

Opinion mining has received more and more attention in the past few years [9, 13]. Ravi et al. [35] present a comprehensive survey on opinion mining and sentiment analysis. Jiang et al. [13] propose the FluidRating model to predict multiple ratings in OSNs; it exploits the fluid dynamics theory to naturally reflect the time-evolving opinion formulation process. Luiz et al. [23] propose a general framework that allows developers to filter, summarize, and analyze user reviews written about applications on app stores. This framework can automatically extract relevant features from reviews of apps and analyze the sentiment associated with each of them. Poria et al. [31] present the first deep learning approach to aspect extraction in opinion mining. Musto et al. [29] propose a multi-criteria recommender system based on collaborative fltering techniques, which exploits the information conveyed by users' reviews.

With the development of opinion mining techniques based on aspect level, the relationship between ratings and aspect-based sentiment has become increasingly attractive. Wang et al. [38] use the review text information to mine the user's aspect preference. They first use an aspect segmentation algorithm to get a set of aspects for each review and use the word sentiment dictionary to calculate the aspect ratings of every review, and finally calculate the aspect weights by leveraging a regression method. Diao et al. [5] propose a probabilistic model based on collaborative filtering and topic modeling. It can capture the interest distribution of users and the content distribution of movies. Ling et al. [20] propose a model named RMR that applies topic modeling techniques on the review text and align the topics with rating dimensions to improve prediction accuracy. Lin et al. [19] uses a novel model to integrate ratings and review texts via latent factor model. Cheng et al. [4] first leverage the review text to model user preferences and item features from different aspects and estimated the aspect importance of a user towards an item, then use a model named ALFM to integrate the aspect importance. HernA et al. [7] exploit the extracted aspect opinion information to provide enhanced recommendations. Huang et al. [9] propose an aspect sentiment similarity-based personalized review recommendation model (A2SPR), which can recommend top-k reviews that contain the most related aspects for individuals.

Although many models have been proposed, existing works usually take a static way to mine user's opinions and item's attributes from reviews, ignoring the fact that both of them are changing over time. To address this issue, we try to deeply study the dynamic changes in user's aspect preferences and item's aspect properties from product reviews.

2.2 Dynamic Evolution of User Opinions

As is well known, both user preferences and item properties are changing dynamically over time [21, 49], so it is unadvisable to statically treat historical information of users and item. Over the last few years, there has been some works trying to improve the accuracy of the recommendation system by leveraging the time information [13, 40]. Koren [17] proposes a time-aware RS named TimeSVD++. It divides the rating data into different time bins, for each bin they separately model user preferences and item properties. Xiang et al. [42] further learn long-term and short-term factors through random walk on sessions-based temporal graphs. Koenigstein et al. [16] use the matrix decomposition model to combine the timestamps information with the rating data on the yahoo music dataset, exploiting the temporal characteristics of this dataset. Yin et al. [45] analyze user behaviors in social media systems and design a latent class statistical mixture model named TCAM, which considers the intentions and preferences behind user behaviors. Yuan et al. [46] develop a collaborative recommendation model that is able to incorporate time factors to recommend Points-of-interest (POIs) for a given user at a specified time in a day. Zhang et al. [47] consider evolving preferences and model user dynamics by introducing and learning a transition matrix for each user's latent vectors between consecutive time windows.

A novel approach with fluid dynamics has been proposed for personalized opinion formation [10, 12, 13, 37], in which the system is modeled simply with containers and pipes. To be specific, a user is seen as a "container," with their opinion as fluid in the container. The fluid has two dimensional information: the fluid temperature indicates the rating (e.g., opinion), and the fluid height as the persistence of their opinion. Several containers are connected through single directional pipes, corresponding to influence relations. Moreover, friends' influences are propagated via the fluid exchange among connected containers in several discrete steps.

Existing works usually focus the relationship between rating and time, ignoring the fine-grained dynamic changes of user preferences contained in user's reviews. In this article, we strive to combine review information and time factor. There have been some attempts to combine temporal dynamic information and reviews information [22]. Wu et al. [41] take temporal dynamics, reviews, and correlation into consideration and combine this information for accurate recommendation, but they don't take into account the aspect properties in user's comments. McAuley and Leskovec [26] believe that the user's aspect preferences may have changed over time, and they divide each user's experience into several levels and define a recommendation system for each level to achieve higher accurately. Mukherjee et al. [28] model the joint evolution of user experience and develop a generative HMM-LDA model to trace user evolution. Their difference with our method is that they only model user experience, while we also consider time information based on user experience. Zhang et al. [51] propose a daily-aware personalized recommendation based on feature-level time series analysis. It also differs from our work as they treat all reviews of a day as a whole to analyze the trend of product feature while we attempt to leverage aspect information to model the dynamic changes of user preferences and item properties.

In summary, our work is different from others at threefold: (1) we try to deeply study the evolution and understand the dynamics features of both users and items; (2) we strive to comprehensively capture the fine-grained change patterns of user's aspect preferences and item's aspect properties in real datasets; and (3) we try to exploit the evolution of aspect-level features of users and items to enhance the dynamic sentiment prediction.

3 PROBLEM DEFINITION

3.1 System Settings

In e-commerce platforms, a user u may give a rating after purchasing an item i, which represents his overall satisfaction with this item. For example, rating values can be integers ranging from 1 (star) indicating a strong unsatisfaction to 5 (stars) indicating a strong satisfaction. We convert the rating prediction task into a binary classification problem. Therefore, we mark the rating greater than 3 (stars) as 1, and the others as 0, where 1 is positive (like) and 0 is negative (dislike).

We use t to represent time (e.g., a week), so that a rating $\mathrm{r}_{ui}(t)$ indicates the satisfaction of user u on item i at time t. Meanwhile, user u may write a review text $d_{ui} \subset D$, where D represents the collection of all review texts. Formally, an aspect set $A = \{a_1, a_2, a_3, \ldots, a_k\}$ which contains k aspects is predefined in D. For example, in Beeradvocate dataset, four aspects are predefined for this domain, i.e., $A = \{$"appearance," "aroma," "palate," "taste"$\}$. A_d represents a set of aspects that

Table 1. Notations

Notation	Description
$r_{ui}/\hat{r}_{ui}$	Real/predicted rating that user u give to item i, (normalized into $[0,1]$)
$s_{ui}^k(t)$	Real aspect rating that user u give to aspect a_k of item i in time t, (normalized into $[1,5]$)
$o_{ui}^k/\hat{o}_{ui}^k$	Real/predicted aspect sentiment that user u give to aspect a_k of item i, (normalized into $[0,1]$)
p_u^k/q_i^k	The f-dimensional latent vectors corresponding to user u/item i for aspect a_k
$b_u^k(t)/b_i^k(t)$	The change of user's/item's kth aspect preferences/properties over time
$\mathrm{B}_u(t)/\mathrm{B}_i(t)$	Parameters on time t contained in user u/item i in the model
μ^k	A constant pertaining to aspect a_k
z^k	A general coefficient expressing the relative importance of aspect a_k
w^k	The impact of user maturity on aspect a_k.
v^k	The impact of item popularity on aspect a_k.
k	The number of aspects
D	The collection of all review text
A	the collection of all aspects
a_k	the kth aspect of collection A
f	The dimension of latent factors
λ_r/λ_s	The regularization parameters of the overall/aspect rating part
α	A parameter to combine the aspect rating part with the overall rating part

appears in a comment and we assume that $A_d \subset A$. The notations used in the article are listed in Table 1.

In order to measure the evolution of user preference and item property as well as the aspect level features, we present three concepts: user maturity, item popularity, and aspect impact. Details are as follows.

Definition 1: User Maturity. The user maturity represents the user's level of experience. The higher experience level a user has, the higher maturity he has.

We use the user review numbers to reflect user maturity. Specifically, we divide all users into several groups according to the number of reviews. Each group corresponds to different user maturity. The more reviews a user has posted, the higher maturity the user has. For example, in the Beeradvocate dataset, all users are divided into four groups $\{R_u < 50, 50 \leq R_u < 100, 100 \leq R_u < 200, R_u \geq 200\}$. For instance, $R_u < 50$ indicates the group of users who post less than 50 reviews.

Definition 2: Item Popularity. Item popularity represents how popular the item is (i.e., the degree that users like it). The more an item is purchased or commented on, the higher popularity it has.

We use the item review numbers to estimate the item popularity. Specifically, we divide all items into several groups according to the number of reviews they received. Each group corresponds to different item popularity. The more reviews an item has, the higher popularity the item has. For example, in the Beeradvocate dataset, all items are also divided into three groups $\{R_i < 30, 30 \leq R_i < 200, R_i \geq 200\}$. For instance, $R_i < 30$ indicates the group of items that receive less than 30 reviews.

Definition 3: Aspect Impact. The aspect impact indicates the importance of a particular aspect for a user or an item. It consists of three parts: aspect weight, user maturity, and item popularity.

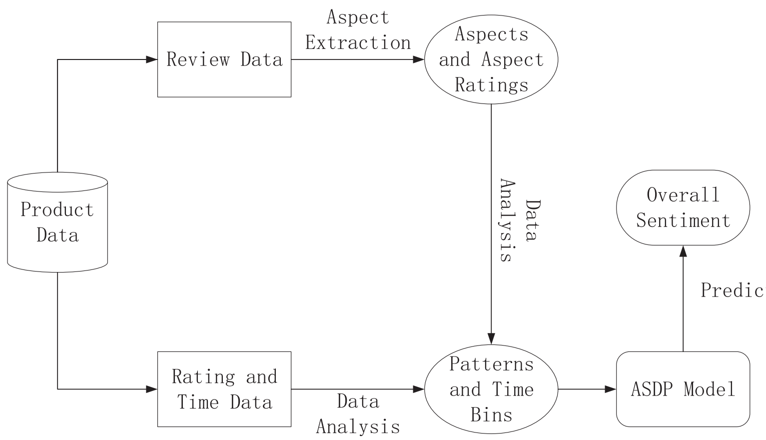


Fig. 2. Overview of our solution.

A user may have different preferences on different aspects. The higher impact an aspect has for a user, the higher preference he has on this aspect. Similarly, the higher aspect impact an aspect has for an item, the more popularity it has for this aspect.

3.2 The Problem of Aspect-based Sentiment Dynamic Prediction

We identify a novel problem of ASDP. We formulate it as a binary sentiment dynamic prediction task, to predict whether a user likes a target item based on dynamic historical behaviors. Specifically, we study how to capture the dynamic changes of user aspect preferences and item aspect properties from user's reviews. Given the historical behavior data of user u and a target item i, our task is to predict whether the user u likes item i by considering time evolving of user opinions in aspect level.

Sentiment classification helps estimate whether a user likes an item. According to the estimation, recommendation system can make proper recommendation. Therefore, we hope to improve the prediction accuracy of sentiment classification by capturing the dynamic features of user's aspect preferences. There are two main challenges:

(1) **How to capture the dynamic changes of user aspect preferences and item aspect properties?**
Specifically, we can define this objective as follows: for each user u and the target item i, our task is to figure out the change patterns of the aspect rating $s_{ui}^k(t)$ over time, k represents the number of aspects.
(2) **How to explore ASDP to improve recommendation accuracy?**
Specifically, we convert the sentiment dynamic prediction task into a binary classification problem. For each user u and the target item i, our task is to predict whether the sentiment is positive or negative by considering dynamic changes in aspect preferences.

3.3 Solution Overview

Keeping the evolution and aspect level feature in mind, we propose the ASDP. Our solution mainly consists of three steps, as shown in Figure 2. (1) Extract aspects and aspect ratings from the product reviews. Many methods can be exploited. Here we use the latent aspect rating analysis (LARA) method [38] as the basis. It can get specific aspects and aspect ratings for every review by using an aspect segmentation algorithm. These aspects and aspect ratings can help analyze the dynamic

Table 2. The Statistics of Three Datasets

Dateset	#users	#items	#reviews	time span
Amazon_Book	21,322	126,078	549,723	2012-01-01, 2012-12-31
Ratebeer	10,356	69,005	855,863	2010-01-01, 2011-12-31
Beeradvocate	22,406	45,084	848,623	2009-01-01, 2011-12-31

Table 3. Aspect Seed Keywords in Amazon-Book Dataset

Aspects	Seed words
value	value, price, worth, expensive, cheap
content	content, chapter, piece, story, series
paper	paper, typography, leather, size, shape, package
style	style, peaceful, wonderful, interesting, wisdom
writing	writing, language, emotion, thought
culture	culture, classic, literature, history

changes of users' emotion. Moreover, the aspect ratings can serve as the ground truth of the aspect rating prediction task, similar to [1]. (2) Identify the dynamic changes in user aspect preferences and item aspect properties via extensive data analysis. Here we mainly consider two types of changes: the gradual changes and the sudden changes. (3) Capture the change patterns of user preferences and item attributes in aspect level, and flexibly combine the temporal dynamic and product reviews.

3.4 Datasets and Preprocess

We use three real datasets (Amazon-Book, Beeradvocate, and Ratebeer) collected by McAuley et al. [25] to explore the dynamic changes in user's aspect preferences and item's aspect properties. A recordd in the datasets contains user ID, item ID, user review, review time, and rating. Considering the temporal dynamics factor, we divide the datasets in chronological order, and take the earliest 80% as the training set, and the rest 20% as a test set. The records for training occur before that for the test. The details of the datasets are shown in Table 2.

The original datasets use a 5-star rating system. We transform it into the binary "high" ({4, 5}) and "low" ({1, 2, 3}) classes. Furthermore, we reformulate sentiment estimation as a classification problem to predict whether a user would like an item.

For users' text comments, we first perform simple pre-processing on these reviews: (1) converting all the words in comments to lowercase; (2) removing the stop words according to the stop words list and removing the words that appear less than five times in all reviews. Because these rare words are not representative; (3) converting each word into its root by using NLTK [38], which is convenient for us to match the aspect of each user. Then we artificially set up a set of aspects and aspect seed keywords. These aspect seed keywords will be used to extract aspect and aspect ratings. For example, for Amazon-Book dataset, we set the number of aspects to be six, and the aspect seed words are shown in Table 3. Table 4 shows the aspect seed words of Beeradvocate and Ratebeer datasets. It is worth noting that we can extract any number of aspects according to the product properties. For the convenience of display and analysis, here we only use a few common aspects.

3.5 Preliminaries

In this subsection, we introduce the preliminaries of our work, i.e., the basic MF model [1], the TimeSVD model [41], and the TimeSVD++ model [17].

Table 4. Aspect Seed Keywords in Beeradvocate and Ratebeer Datasets

Aspects	Seed words
appearance	appearance, look, package, bottle, sheap
aroma	aroma, smell, savor, fragrant, bouquet
palate	palate, sense, feel, tang, lubrication
taste	taste, flavor, wonderful, sample, sweet

3.5.1 Matrix Factorization: A Basic Model. The MF model, as a state-of-the-art recommender method, maps both users and items to a joint latent factor space of dimensionality f. The latent space tries to explain ratings from user feedback. The final rating score r_{ui} for a user u and item i is calculated as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u. \tag{1}$$

MF is a classical model in recommendation systems and it performs well in accuracy and scalability. Moreover, it is very flexible to add side data sources for recommender. However, it assumes that all the historical behaviors are static and fails to capture the dynamic of user preferences and item properties. We adopt the MF method as a basis of our model.

3.5.2 TimeSVD and TimeSVD++: Time-aware Factor Models. User preferences and item properties are changing over time. To capture the dynamic features, Koren [17] proposed a TimeSVD++ method to dynamically process user behavior information. Based on MF, it adds implicit feedback, which provides an additional indicator of user preferences. To capture dynamic information, it splits the dataset into T bins, ratings for user u and item i in period $t \subset T$ are then defined as:

$$\hat{r}_{ui} = \mu + b_u(\mathrm{t}) + b_i(\mathrm{t}) + q_i^T \left(p_u(\mathrm{t}) + |\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \subset \mathrm{R}(u)} y_j \right), \tag{2}$$

where $b_u(\mathrm{t})$, $b_i(\mathrm{t})$ are time drifting bias for user u and item i in time period t, $p_\mathrm{u}(\mathrm{t})$ represents the latent factors of user u which change dynamically with time, and $|\mathrm{R}(u)|^{-\frac{1}{2}} \sum_{j \subset \mathrm{R}(u)} \mathrm{y_j}$ represents implicit feedback.

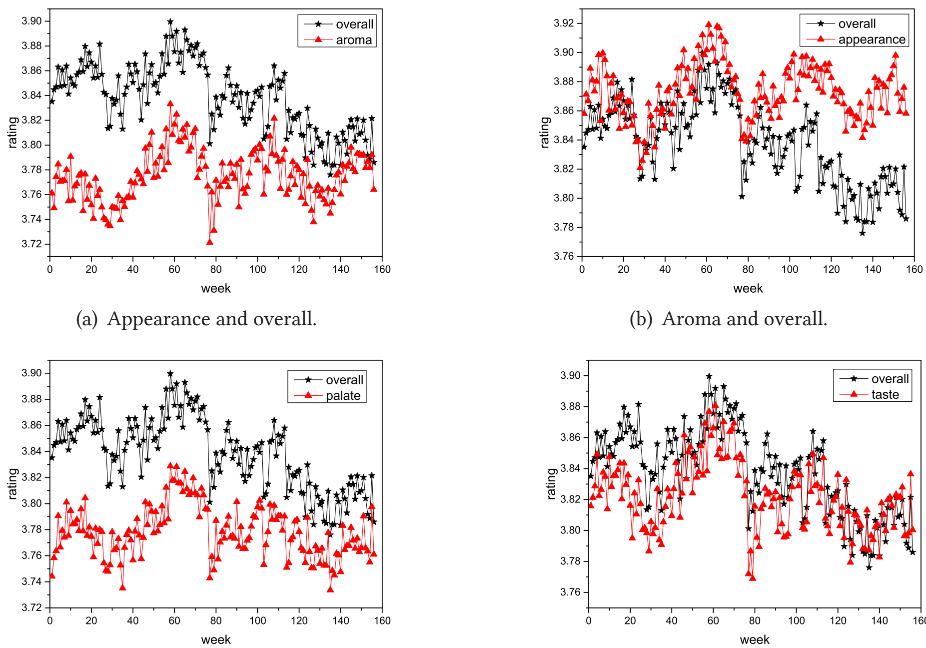
In our work, we focus on capturing the aspect-level dynamic features of users and items. We explore the idea of TimeSVD++ as the basis. To facilitate calculations, we remove the implicit part and simplify the TimeSVD++ model into TimeSVD. The modified model is as follows:

$$\hat{r}_{ui} = \mu + b_u(\mathrm{t}) + b_i(\mathrm{t}) + q_i^T p_u. \tag{3}$$

More details can be found in [17, 41]. Learning of the parameters is performed by minimizing the associated squared error function on the training set using a regularized stochastic gradient descent algorithm. We incorporate the time factor into our framework, and dynamically define the size of each bin by looking for mutation points.

4 DATA ANALYSIS FOR ASPECT-BASED DYNAMIC CHANGES

In this section, we analyze three real datasets (i.e., Amazon-Book, Ratebeer, and Beeradvocate as in Table 2) and explore the dynamic changes in overall ratings and aspect ratings, so as to deeply study the dynamic changes in user aspect preferences and item aspect properties. We first set up six aspects (“value,” “content,” “paper,” “style,” “writing,” and “culture,” as in Table 3) for Amazon-Book, and four aspects (“appearance,” “aroma,” “palate,” and “taste,” as in Table 4) for Ratebeer and Beeradvocate. Then we use the LARA method [38] to extract user’s aspect ratings from the comments. We deeply study the following four research questions (RQs):



(a) Appearance and overall.

(b) Aroma and overall.

(c) Palate and overall.

(d) Taste and overall.

Fig. 3. Dynamic changes in overall ratings and aspect ratings in Beeradvocate dataset.

RQ1: How do the overall and aspect ratings of the entire datasets change over time?
RQ2: What are the distributions of mutation points of all users and items change over time?
RQ3: How do the overall and aspect ratings of an individual user or a single item change over time?
RQ4: What are the distributions among user groups of different maturity and item groups of different popularity?

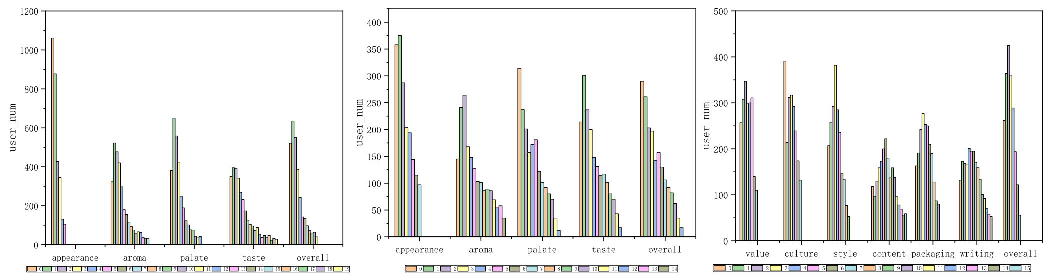
4.1 RQ1: How do the Overall and Aspect Ratings of the Entire Datasets Change Over Time?

To address RQ1, we divide the entire datasets into weekly intervals, and we calculate the average overall and aspect ratings of each week. Then we study their dynamic changes. Since the results in three datasets are similar, here we only display that in Beeradvocate, as shown in Figure 3. We can get the following findings:

(1) Both the overall ratings and the aspect ratings show clear change patterns. In Figure 3, [1, 52] weeks are for the first year, [53, 104] weeks are for the second year, and [105, 157] weeks are for the third year. It shows sudden changes around 27th, 62th, 79th, 107th, 135th, and 148th week, corresponding to the middle and the end of each year. We call them the mutation time point. At other times, there is a steady rise or fall change.

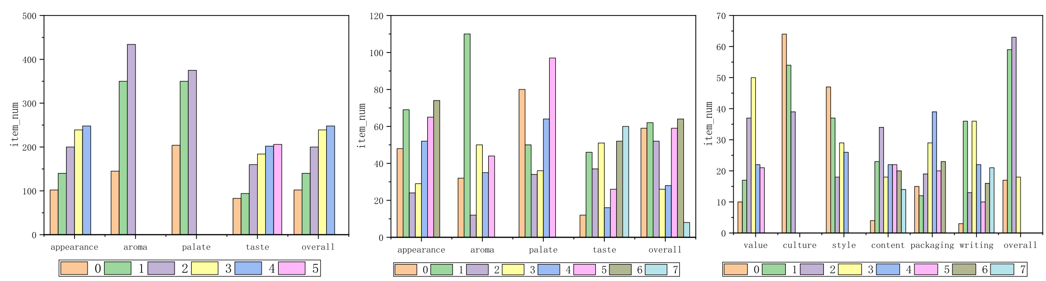
(2) The overall rating is related to each aspect rating because it shows a similar trend with each of the four aspects, as shown in the four sub-figures in Figure 3. It incites us that the overall rating can be calculated by summing up all aspect ratings.

(3) The impact of each aspect on the overall rating is different. It can be seen that the scores in "appearance" aspect are higher than the overall ratings (Figure 3(a)). The scores in "aroma" and "palate" aspects are lower than the overall ratings (Figure 3(b) and 3(c)), while the scores in "taste"



(a) Beeradvocate (b) Ratebeer (c) Amazon-Book

Fig. 4. The distribution of mutation points for users.



(a) Beeradvocate (b) Ratebeer (c) Amazon-Book

Fig. 5. The distribution of mutation points for items.

aspect are similar to the overall ratings (Figure 3(d)). Therefore, we can infer that the "taste" aspect has a higher aspect weight, while the "aroma" and "palate" aspects have lower weights.

4.2 RQ2: What are the Distributions of Mutation Points of all Users and Items Change Over Time?

We delve into the mutation time points of all users and all items, so as to deeply study their dynamic characteristics. To be specific, we count the numbers of users and items with respect to different mutation time points from the aspect level and the overall level, respectively. Figure 4 shows the results of users, and Figure 5 shows the results of items. The x-axis represents different aspects. The y-axis represents the number of users (items). Each column represents different mutation points. Taking Figure 4(a) for instance, we display the number of users who has 0, 1, 2 . . . , 19 mutation time points, at the aspect "appearance," "aroma," "palate," "taste," and the overall rating, respectively. We gain the following findings.

(1) In general, the number distribution of user mutation points presents a long tail effect. The greater the number of mutation points, the smaller the user count. It verifies the existence of mutation points for users. It also indicates that there are not too many mutation points for majority of users, usually a few to dozens.

(2) Some aspects take more impacts than others on the overall rating. To be specific, in Beeradvocate dataset (Figure 4(a)), the distribution of "palate" aspect is more similar to that of the overall rating. It means that, for the majority of users, the change of this aspect rating is the main factor affecting users' preference for Beeradvocate dataset. This also indicates that the weight of each aspect is different, "palate" aspects is more important in Beeradvocate. In Ratebeer dataset (Figure 4(b)), the distribution of "taste" aspect is more similar to that of the overall rating, which

Table 5. The Percentage Distribution of the Number of Mutation Points for All Users and Items

Aspect	Beeradvocate					
	User			Item		
	$p < 5$	$5 \leq p < 15$	$p \geq 15$	$p < 2$	$2 \leq p < 5$	$p \geq 5$
appearance	96.5%	3.5%	0%	26.1%	74.9%	0%
aroma	69.2%	29.7%	1.1%	53.3%	46.7%	0%
palate	76.7%	23.3%	0%	59.6%	40.4%	0%
taste	59.3%	34.9%	5.8%	19.1%	58.8%	22.1%
overall	79.2%	20.8%	0%	26.1%	73.9%	0%
Aspect	Ratebeer					
	User			Item		
	$p < 5$	$5 \leq p < 15$	$p \geq 15$	$p < 2$	$2 \leq p < 5$	$p \geq 5$
appearance	29.9%	46.1%	24%	32.4%	29.1%	38.5%
aroma	22.9%	49.6%	27.5%	50.2%	34.3%	15.5%
palate	27%	47.6%	25.4%	36.0%	37.1%	26.9%
taste	25.1%	47.5%	27.4%	19.3%	34.7%	46.0%
overall	24.8%	48.2%	27.0%	33.8%	29.6%	36.6%
Aspect	Amazon-Book					
	User			Item		
	$p < 5$	$5 \leq p < 15$	$p \geq 15$	$p < 2$	$2 \leq p < 5$	$p \geq 5$
value	72.9%	27.1%	0%	17.2%	69.4%	13.4%
culture	73.6%	26.4%	0%	75.2%	24.8%	0%
style	68.0%	32.0%	0%	53.5%	46.5%	0%
content	32.6%	67.4%	0%	17.2%	47.1%	35.7%
packaging	54.3%	45.7%	0%	17.2%	55.4%	27.4%
writing	40.0%	60.0%	0%	24.8%	45.2%	30.0%
overall	82.0%	18.0%	0%	48.4%	51.6%	0%

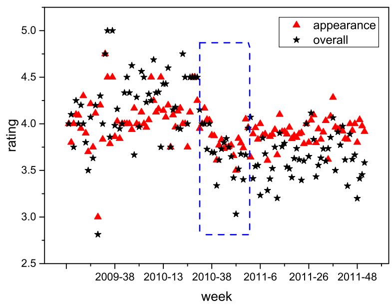
implies that this aspect is more important in Ratebeer. Similarly, in the Amazon-Book dataset (Figure 4(c)), the distributions of "style" aspect is more similar to that of overall rating. It means that for book products, "style" aspect has the higher impact.

We also analyze the distribution of mutation points for all items, as shown in Figure 5. The x-axis represents different aspects. The y-axis represents the number of items. Each column represents different mutation points. Taking Figure 5(a) for instance, we display the number of items who has 0, 1, 2, ..., 5 mutation time points, at the aspect "appearance," "aroma," "palate," "taste," and the overall rating, respectively. We gain the following findings.

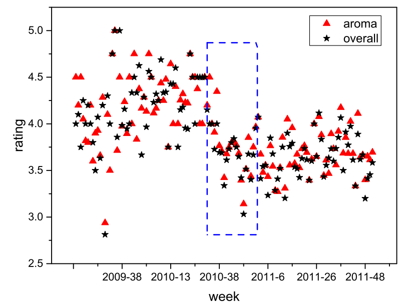
(1) Compared with the changes of mutation points in users, the changes in items are relatively small. That is to say, the changes in commodities are relatively stable.

(2) Again, some aspects take more impacts. For the two beer datasets (Figure 5(a) and 5(b)), the distribution of "taste" aspect is more similar to that of the overall rating. It indicates the "taste" aspect is the main one that affects beer products. For the book dataset (Figure 5(c)), the "content" aspect is more important.

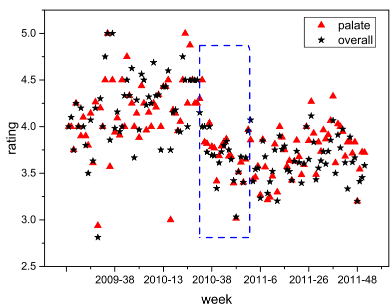
In order to show our findings more clearly, we divide users and items into three subsets according to the number of mutation time points. i.e., $p < 5$, $5 \leq p < 15$, $15 \geq p$ for users, and $p < 2$, $2 \leq p < 5$, $p \geq 5$ for items. Then we observe the percentage distribution of each group in each aspect and the overall rating. Table 5 shows the results. We gain two main findings.



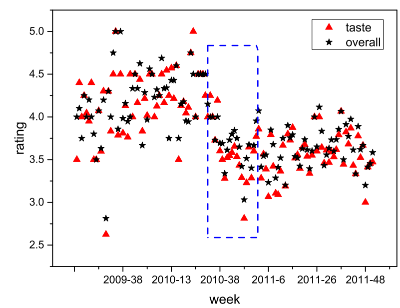
(a) Appearance and overall.



(b) Aroma and overall.



(c) Palate and overall.



(d) Taste and overall.

Fig. 6. Dynamic changes in overall ratings and aspect ratings for a single user.

(1) In general, over 20% of users have more than five mutation points considering all aspects and overall rating. Among them, this ratio exceeds 70% in Ratebeer, because each user has more consumption records in this dataset. Meanwhile, in beer products, there are more mutations in "taste"; and in book products, it is "writing" and "content."

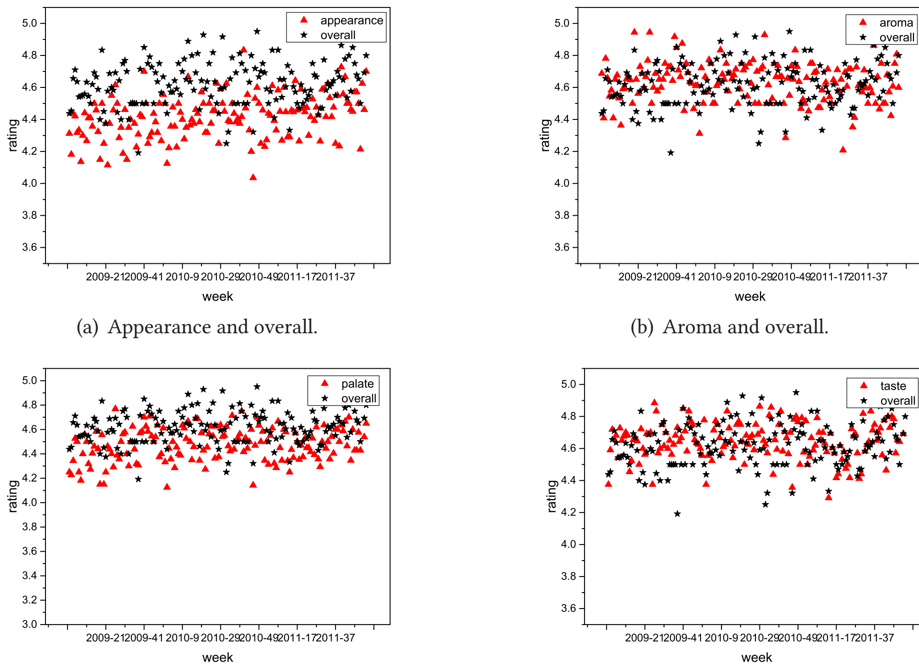
(2) The number of users mutation points is significantly higher than that of items. This indicates that the frequency of user preference changes is higher than that of commodities.

4.3 RQ3: How do the Overall and Aspect Ratings of Individual Users/Items Change Over Time?

We study RQ3 to go deeper into individual users/items. Without loss of generality, we randomly select some users and items to analyze. Here we only display the results of one user (user ID = 143) and one item (item Id = 4742) to show our findings. Again, we divide the data into weekly intervals, and we calculate the average overall and aspect rating of each week to observe their dynamic changes. First, we study the dynamic changes of a single user, as shown in Figure 6.

We can find that for this user, the change trend of aspect rating is closely related to the overall rating. A mutation time point is found around the 38th week of 2010, which may indicate that the user's preference or experience has changed greatly due to some factors. In other weeks, it shows a linear change in user's ratings, which reflects the gradual changes in user preferences.

Next, we study the dynamic changes of a single item, as shown in Figure 7. We find that for a single item, the change trend of aspect ratings is also closely related to that of the overall rating. Different from users, the change is relatively stable, which shows a slow linear change relationship, and no sudden change occurs.



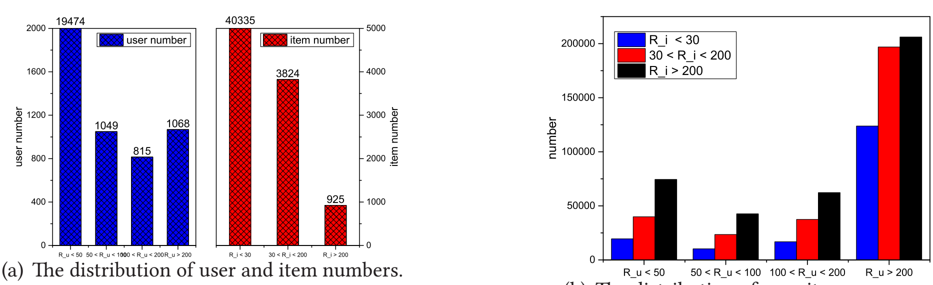
(a) Appearance and overall.

(b) Aroma and overall.

(c) Palate and overall.

(d) Taste and overall.

Fig. 7. Dynamic changes in overall ratings and aspect ratings for a single item.



(a) The distribution of user and item numbers.

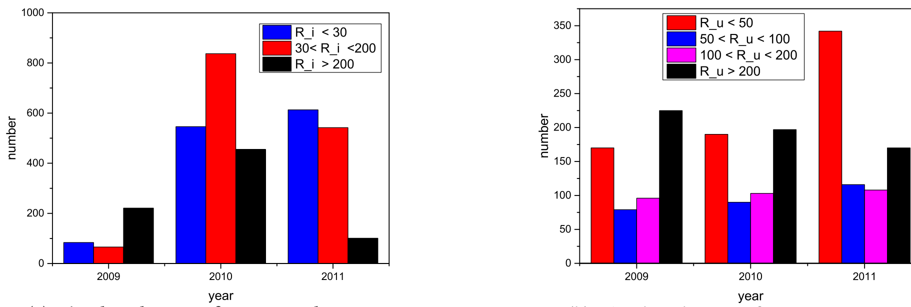
(b) The distribution of user-item group.

Fig. 8. The distribution of user/item numbers and user-item groups for overall items.

4.4 RQ4: What are the Distributions Among User Groups of Different Maturity and Item Groups of Different Popularity?

We study RQ4 to check the distributions of different user and item groups. Recall that we define four levels of user maturity and three levels of item popularity (in Section 3.1). We check the entire dataset to observe the number of users/items in each group, as shown in Figure 8(a). It can be found that most users/item are in low maturity/popularity, accounting for the proportion of 87% and 89%, respectively.

Next, we study the distribution of item groups in each user group, as shown in Figure 8(b). It can be found that all user groups like to purchase high-popularity items. However, for low maturity user group ($R_u < 50$), only about 14% people purchase low popularity items. For high maturity user group ($R_u > 200$), there are about 23% people purchase low popularity items. It indicates that



(a) The distribution of item popularity. (b) The distribution of user maturity.

Fig. 9. The distribution of review numbers and user maturity for a popular item.

experienced users (with high maturity) are more likely to make their own judgments than to be influenced by popular items.

Next, we check the item group distribution of an individual user (user Id = 143), as shown in Figure 9(a). It shows that the user prefers to buy popular items in three years. In 2009, only 22% of the products he purchased are of low popularity. However, by 2011, the proportion has increased to 48%. We can see that, with the improvement of user maturity (experience), he gradually focuses more on his own personal judgment than to be affected by item popularity.

Finally, we check the user group distribution of an individual item (item Id = 4742), as shown in Figure 9(b). From 2009 to 2011, the number of users with low maturity is increased from 30% to 46%, while that with high maturity is decreased from 39% to 23%. It indicates that with the increase of item popularity, it becomes more attractive to low maturity users, but less attractive to high maturity users.

4.5 Summary of Our Findings

In summary,we gain four main findings as follows, corresponding to the answers of the research questions, RQ1, RQ2, RQ3, and RQ4, respectively.

(1) Both the overall ratings and the aspect ratings show two kinds of change patterns: the sudden changes and the gradual changes. Moreover, the overall rating is related to each aspect rating and different aspects have different impact on the overall rating. It indicates that the overall rating can be calculated by summing up all aspect ratings with proper weights.
(2) In the three datasets, there do exist the mutation points in the aspect ratings and overall ratings. Moreover, the number of mutation points of users are clearly more than that of items, indicating that users' preference changes more frequently than that of items' properties.
(3) For a single user, the overall ratings and aspect ratings also show two kinds of change patterns, i.e., the gradual changes and the sudden changes. However, for a single item, their changes are relatively stable, showing a slow a gradual linear change.
(4) For different user groups, most users like high popularity items. However, with the improvement of user maturity, users are more likely to make their own judgments than to be influenced by popular items. For different item groups, with the increase of item popularity, it becomes more attractive to low maturity users, but less attractive to high maturity users.

Based on the above data analysis and findings, we can design our model to capture the dynamic changes (gradual changes and sudden changes) of overall ratings and aspect ratings.

5 ASDP: THE ALGORITHM DETAILS

In this section, we describe the ASDP model in detail. It captures temporal dynamic from product reviews based on equal intervals. We further propose a bin segmentation algorithm based on the mutation time points where sudden changes happen, which helps to set the size of intervals non-uniformly. The improved model is called ASDP+.

5.1 Aspect Extraction and Sentiment Analysis

In order to explore the evolution of user opinions from aspects level, we need to extract aspects and estimate the aspect ratings from users' reviews. Here we leverage a state-of-the-art method of LARA [38]. This is because LARA can obtain specific aspects and aspect ratings for every review. For example, for the review "The taste of this beer is great," the word "taste" can be extracted as an aspect and the aspect rating can be classified as positive by LARA. Generally, aspect segmentation algorithm is performed for extracting aspects of each review, and latent rating regression is used for computing aspect ratings. In particular, when performing the aspect segmentation algorithm, we manually set the number of aspects in each category of entities, and we select the most representative words as the aspect seed keywords. Then we iteratively expand the aspect keywords. Next, we match each review with these aspect keywords, and assign a set of corresponding aspects to each review. Finally, we use a latent rating regression method to calculate aspect ratings. More details can be found in [38].

Formally, for a set of reviews D of a given entity (e.g., beer), LARA builds a set of aspects A occurring in D, and identifies a set of aspects A_d occurring in $d \subset D$ for each review. The corresponding aspect ratings $s_{ui}^k \subset \{1, 5\}$ indicate that user u expresses rating about aspect $k \subset A_d$ on item i. And we use $o_{ui}^k \subset \{0, 1\}$ to indicate aspect sentiments, where 1 is positive (like) and 0 is negative (dislike). The relationship between aspect ratings and aspect sentiments is shown below.

$$\mathrm{o}_{ui}^k = \begin{cases} 1, & if\ s_{ui}^k > 3 \\ 0, & otherwise \end{cases} \tag{4}$$

5.2 ASDP: Based on Equal Intervals

In this subsection, we present the ASDP model that is able to capture the dynamic changes of user aspect preferences and item aspect properties from user's reviews. It has two parts. The first part is to capture the dynamic changes of aspect ratings by using TimeSVD method (Section 3.5). The second part is to predict the overall sentiment by combining aspect ratings with aspect weights. ASDP can predict whether a user is interested in an item at a specific time.

For capturing the dynamics of aspect ratings, ASDP estimates the sentiment utility value for each aspect k using the MF approach. We split the dataset into T intervals (denoted as bins), as is shown in Figure 10. We regard the user's aspect preferences and item's aspect properties as a function of time. Then, the aspect rating is calculated as follows:

$$\hat{s}_{ui}^k(\theta_s) = \mu^k + b_u^k(t) + b_i^k(t) + (q_i^k)^\top p_u^k, \tag{5}$$

where μ^k is a constant pertaining to aspect k, representing the average rating for each aspect. $b_u^k(t)$ represents the change of user's kth aspect preferences over time, which we use to capture the sudden changes in user aspect preferences. $b_i^k(t)$ represents the change of item's kth aspect properties over time, which we use to capture gradual changes in item aspect properties. p_u^k and q_i^k are f-dimensional latent vectors for aspect k. Here, for the convenience of calculation, we omit the dynamic change of latent vector. The two biases $b_i^k(t)$ and $b_u^k(t)$ to item i and user u for aspect k are calculated as follows:

$$b_i^k(t) = b_i^k + b_{i,Bin(t)}^k, \tag{6}$$

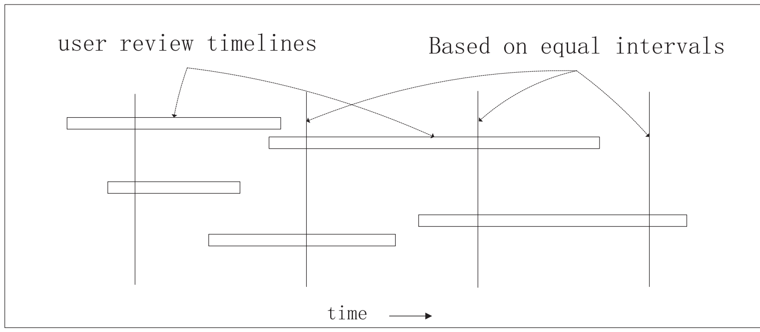


Fig. 10. Dividing bins based on equal intervals.

$$b_u^k(t) = b_u^k + \alpha_u^k * dev_u(t) + b_{u,t}^k. \tag{7}$$

The item's aspect bias $b_i^k(t)$ in Equation (6) consists of a fixed part b_i^k and a dynamic change part $b_{i,Bin(t)}^k$, so as to capture linear variation. Similarly, in Equation (7), for user's aspect bias, b_u^k represents a static feature vector for user u and $\alpha_u^k * dev_u(t)$ approximates a possible portion that changes linearly over time. $b_{u,t}^k$ absorbs the local variability (capture sudden changes) in period t.

We denote all these coefficients by $\theta_s = (\mu, B_u(t), B_i(t), P, Q)$, where $B_u(t)$ indicates all parameters about user u, $B_i(t)$ indicates all parameters about item i. P and Q are latent vectors. Then we use a logistic function $\hat{o}_{ui}^k(\theta_s) = g(\hat{s}_{ui}^k(\theta_s))$ to convert aspect ratings to {0, 1}, where 1 is positive (like) and 0 is negative (dislike). The function is defined as follows:

$$g(x) = \frac{1}{1 + e^{-x}}. \tag{8}$$

In particular, assuming that the training samples are generated independently, we use the maximizing log likelihood function method to estimate parameters θ_s.

$$l_s(\mathrm{S}|\theta_s) = \sum_{u,i,k} \left(o_{ui}^k \log\left(\hat{o}_{ui}^k(\theta_s)\right) + (1 - o_{ui}^k) \log\left(1 - \hat{o}_{ui}^k(\theta_s)\right)\right), \tag{9}$$

where S is the set of all aspect sentiments expressed by users in the training set.

Based on that, ASDP defines $\mathrm{r}_{u,i} \subset \Re$ as user u's overall satisfaction on item i. ASDP estimates the overall utility value as a linear combination of the individual sentiment utility values for all the aspects in a review, as follows.

$$\hat{e}_{ui}(\theta) = \sum_{k \subset \mathrm{A}} \hat{s}_{u,i}^k(\theta_s) \cdot \left(z^k + w_u^k + v_i^k\right), \tag{10}$$

where z^k is a general coefficient representing the relative importance of aspect k, w_u^k is the impact of user maturity on aspect k, and v_i^k represents the impact of item popularity on aspect k. Z, W, V refer to all the parameters z^k, w_u^k, v_i^k. We also use a logistic function $\hat{r}_{ui}(\theta) = g(\hat{e}_{ui}(\theta))$ to convert overall ratings to $\{0, 1\}$. We denote these new coefficients by $\theta_r = (Z, W, V)$ and the set of all coefficients in the model by $\theta = (\theta_r, \theta_s)$. Similarly, we use the maximizing log likelihood function method to estimate parameters θ.

$$l_r(\mathrm{R}|\theta) = \sum_{u,i} (r_{ui} \log(\hat{r}_{ui}(\theta)) + (1 - r_{ui}) \log(1 - \hat{r}_{ui}(\theta))). \tag{11}$$

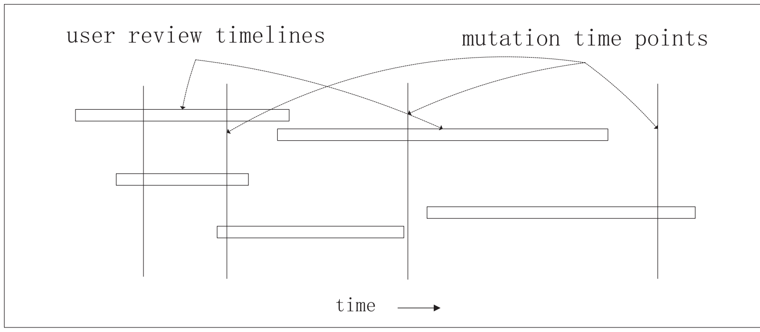


Fig. 11. Dividing bins based on mutation time point.

Finally, ASDP uses a parameter α to combine the aspect rating part with the overall rating part. Moreover, it also exlpoits regularization to avoid over-fitting, as follows.

$$Q(\theta) = -\alpha l_r(\mathrm{R}\,|\theta) - (1-\alpha) l_s(\mathrm{S}\,|\theta_s) + \frac{\lambda_r}{2}||\theta_r||^2 + \frac{\lambda_s}{2}||\theta_s||^2, \tag{12}$$

where λ_r and λ_s are the regularization parameters. ASDP model can explore the changes in user opinions from the perspective of aspects in combination with reviews.

It is worth noting that compared to KDD2017 [1], we are similar in the fitting part of the model. However, the difference is that they only fit each aspect rating, ignoring the temporal dynamic characteristics of aspect preferences. These dynamic characteristics such as gradual changes and sudden changes are exactly what the ASDP model is trying to capture.

5.3 ASDP+: Based on Mutation Time Points

In ASDP model, the size of each bin (i.e., time interval) is fixed. However, the changes of users sentiment are relatively complex in some time intervals, while in other time intervals, the changes are relatively trivial. So it is unadvisable to treat them equally. Therefore, we propose the ASDP+ model to set the size of each time interval non-uniformly based on mutation time points.

Our basic idea is to find the mutation time points in overall ratings and aspect ratings, and then divide the bins by the obtained mutation time points. In this way, we can set the size of each time interval non-uniformly, as is shown in Figure 11. Note that the mutation time points can be different for different datasets.

The details of the bin segmentation algorithm is as shown in Algorithm 1. Lines 1–7 search the mutation time points by setting a threshold value ε. Lines 8–16 divide the specific time intervals according to the obtained mutation time points.

5.4 Model Fitting

We use gradient descent to update parameters [50]. The difference between the real and the predicted values of the overall rating is denoted as $\Delta^r_{u,i} = r_{u,i} - \hat{r}_{u,i}$. Similarly, that of the aspect rating is denoted as $\Delta^s_{u,i,k} = o^k_{ui} - \hat{o^k_{ui}}$. We use an indicator function to indicate whether user u expresses an emotion for item i in his review by $I^k_{u,i} \leftarrow \{0,1\}$. Then we calculate the partial derivative of Q(Loss function) by μ_k as follows:

$$\left.\frac{\partial Q}{\partial \mu^k}\right|_{u,i} = -\alpha \cdot \Delta^r_{\mathrm{u,i}} \cdot (z^k + w^k_u + v^k_i) - (1-\alpha) \cdot I^k_{u,i} \cdot \Delta^s_{\mathrm{u,i,k}}. \tag{13}$$

ALGORITHM 1: Bin segmentation algorithm based on mutation time points

Input: The total number of weeks, n; The overall average ratings for each week, r_t (t represents the current week, from 1 to n);
Output: A set of divided bins, $Bins$;

/* Initialize a list T to record the mutation time points, $Bins$ to save each bin, j to represent the index of the ith mutation time point. */
$T \leftarrow \varnothing$, $Bins \leftarrow \varnothing$, $j \leftarrow 1$;
/* Compare the average rating difference between two adjacent weeks. */
for each $t \subset [1, n-1]$ **do**
/* If the difference is greater than a threshold ε, then we regard it as a mutation time point, otherwise it is not. */
  if $|r_{t+1} - r_t| >= \varepsilon$ **then**
    $T_j \leftarrow t$, $T \leftarrow T \cup T_j$, $j \leftarrow j+1$;
/* Divide the time bins by using mutation time point. */
$lenth \leftarrow j$, $j \leftarrow 1$;
for each $T_j \subset T$ **do**
  if $j = 1$ **then** $Bin_j \leftarrow [1, T_j]$;
  else if $j = lenth$ **then** $Bin_j \leftarrow [T_j, n]$;
  else
    $Bin_j \leftarrow [T_{j-1}, T_j]$;
    $Bins \leftarrow Bins \cup Bin_j$, $j \leftarrow j+1$;
return $Bins$;

The result of Equation (13) is denoted as $-\delta^k_{u,i}$, then we compute the rest parameters of Q. We first fix the parameter $\theta_r : (Z, W, V)$ and update $\theta_s : (\mu, B_u(t), B_i(t), P, Q)$, as follows.

$$\begin{aligned}
b^k_u &:= b^k_u + \gamma \cdot (\delta^k_{u,i} - \lambda_s \cdot b^k_u)\\
b^k_i &:= b^k_i + \gamma \cdot (\delta^k_{u,i} - \lambda_s \cdot b^k_i)\\
b^k_{u,t} &:= b^k_{u,t} + \gamma \cdot (\delta^k_{u,i} - \lambda_s \cdot b^k_{u,t})\\
p^k_u &:= p^k_u + \gamma \cdot (\delta^k_{u,i} \cdot q^k_i - \lambda_s \cdot p^k_u)\\
q^k_i &:= q^k_i + \gamma \cdot (\delta^k_{u,i} \cdot p^k_u - \lambda_s \cdot q^k_i)\\
\alpha^k_u &:= \alpha^k_u + \gamma_1 \cdot (\delta^k_{u,i} \cdot dev_u(t) - \lambda_s \cdot \alpha^k_u)\\
b^k_{i,Bin(t)} &:= b^k_{i,Bin(t)} + \gamma \cdot (\delta^k_{u,i} - \lambda_s \cdot b^k_{i,Bin(t)}).
\end{aligned} \tag{14}$$

Then we fix parameter θ_s to update θ_r, as follows.

$$\begin{aligned}
z^k &:= z^k + \gamma \cdot (\alpha \cdot \Delta^r_{\mathrm{u,i}} \cdot \hat{s}^k_{ui}(\theta_s) - \lambda_r \cdot z^k)\\
w^k_u &:= w^k_u + \gamma \cdot (\alpha \cdot \Delta^r_{\mathrm{u,i}} \cdot \hat{s}^k_{ui}(\theta_s) - \lambda_r \cdot w^k_u)\\
v^k_i &:= v^k_i + \gamma \cdot (\alpha \cdot \Delta^r_{\mathrm{u,i}} \cdot \hat{s}^k_{ui}(\theta_s) - \lambda_r \cdot v^k_i).
\end{aligned} \tag{15}$$

Iteratively, update θ_s and θ_r, until it is convergent.

5.5 Aspect Impact

In this subsection, we apply the model trained in Section 5.4 to determine the aspect impact, which indicates the aspect importance.

We measure the importance of an aspect by its weight in the regression model in Equation (10). That is, for a user u with item i, we first calculate the dynamic sentiment utility values $s_{ui}^k(t)$ for each aspect $k \in$ A. Next, we calculate the impact of each aspect k on the overall predicted level of u' satisfaction with i, as follows.

$$impact_{ui}^k = s_{ui}^k(t) \cdot \left(z^k + w_u^k + v_i^k\right), \tag{16}$$

where $s_{ui}^k(t)$ is the predicted sentiment utility value, z^k is a general coefficient representing the relative importance of aspect k, w_u^k is the impact of user maturity on aspect k, and v_i^k represents the impact of item popularity on aspect k.

6 EXPERIMENTS

In this section, we conduct extensive experiments on three real-world datasets to validate the performance of our work. We first introduce four baseline methods. Then, we deeply test and analyze our model components and their effects. Finally, we study the dynamic changes in aspect impacts for different user maturity and item popularity.

6.1 Baselines and Evaluation Metrics

We compare our model with four baseline methods.

(1) **Learning to Rank User Preferences Based on Phrase-Level Sentiment Analysis Across Multiple Categories (LRPPM)** [3], which models user preference on explicit features extracted from textual reviews, and is able to capture the implicit feed-backs in a direct way that promotes the accuracy. We download the LRPPM system from the authors' website.
(2) **Sentiment Utility Logistic Model (SULM)** [1], which only considers aspect features extracted from user reviews. SULM uses aspect sentiment to predict the overall sentiment. But it does not take into account the time factor.
(3) The TimeSVD model, which is a simplified model of TimeSVD++ [17], as explained in Section 3.4. It captures the temporal dynamics of user preferences by dividing the data into T bins equally, but it ignores the comment text information.
(4) TimeSVD+ model, which is an extended model of TimeSVD. It uses a bin segmentation algorithm that we proposed in Section 5.3. It can dynamically set the size of each time intervals according to the specific dataset.

We evaluate these models via the performance of sentiment prediction. Since this is a binary classification problem, so we use four common metrics to evaluate the performance: Precision(Pre), Recall(Rec), F1, Area Under the Curve (AUC). Among them, AUC avoids the supposed subjectivity in the threshold selection process, and can well measure the performance of the two-class prediction model. The larger value indicates the better performance. The parameters of all comparison models are individually tuned.

6.2 Parameter Sensitivity

There are five important parameters in our model: (1) the number of dimensions for latent factors f; (2) the parameter λ_s, which is the regularization coefficient of the the aspect rating; (3) the parameter λ_r, which is the regularization coefficient of the the overall rating; (4) the parameter

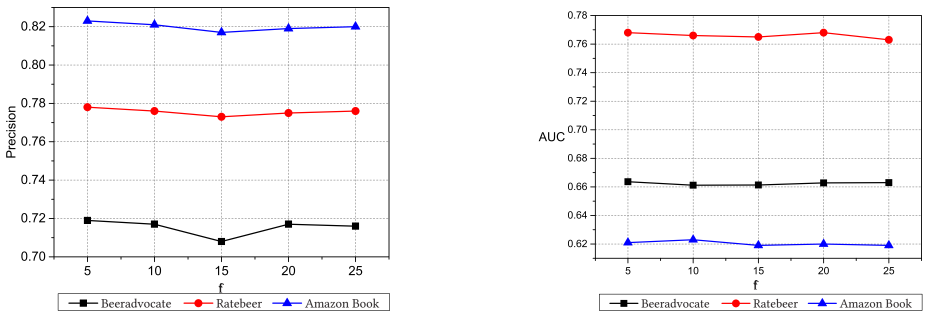


Fig. 12. Performance by varying number of latent factors f.

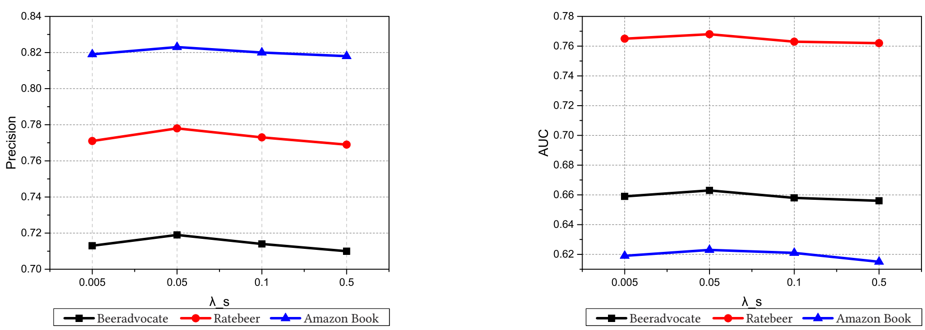


Fig. 13. Performance by varying the regularization coefficient of the aspect rating λ_s.

α, which balances the two parts of aspect rating and the overall rating; and (5) the parameter ε, which controls the size of each time interval. We set their default values to be: $f = 5$, $\lambda_s = 0.05$, $\lambda_r = 0.005$, $\alpha = 0.6$, $\varepsilon = \{0.03, 0.05, 0.06\}$. Without loss of generality, we only show the results on Precision and AUC due to the page limitation.

The effects of parameter f. We set λ_s, λ_r, α, and ε as the default values and change the values of f in $\{5, 10, 15, 20, 25\}$, The results are shown in Figure 12. It shows that the Precision and the AUC of ASDP are relatively stable for different values of f. It indicates that our method is insensitive to different dimensions. Note that existing models with latent factors usually tend to perform better as the dimension increases. From this point, we can say that our model is scaled on dimensions (i.e., the complexity keeps stable when the dimensions increase). The reason may be that each aspect only includes a limited number of dimensions.

The effects of parameter λ_s. We set f, λ_r, α, and ε as the default values and change the value of λ_s in $\{0.005, 0.05, 0.1, 0.5\}$. The results are shown in Figure 13. We find that the performance is also quite stable. Meanwhile, when $\lambda_s = 0.05$, our model ASDP achieves the best performance on both Precision and AUC, in all the three datasets. It may also benefit from the limited number of topics in each review text.

The effects of parameter λ_r. We set f, λ_s, α, and ε as the default values and change the value of λ_r in $\{0.005, 0.05, 0.1, 0.5\}$. The results are shown in Figure 14. We find that the performance decreases a little with the increase of $lamda_r$, indicating that the regularization coefficient of the overall rating cannot be too big. Meanwhile, when $\lambda_r = 0.005$, ASDP achieves the best performance.

The effects of parameter α. We set f, λ_s, λ_r, and ε as the default values and change the value of α in $\{0.5, 0.6, 0.7, 0.8\}$. The results are shown in Figure 15. We find that the performance is

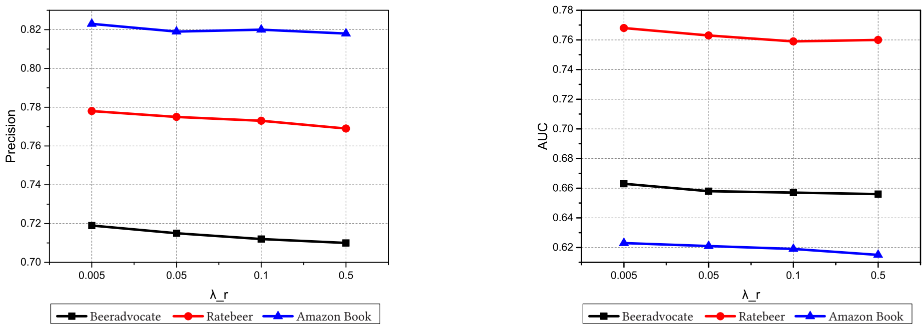


Fig. 14. Performance by varying the regularization coefficient of the overall rating λ_r.

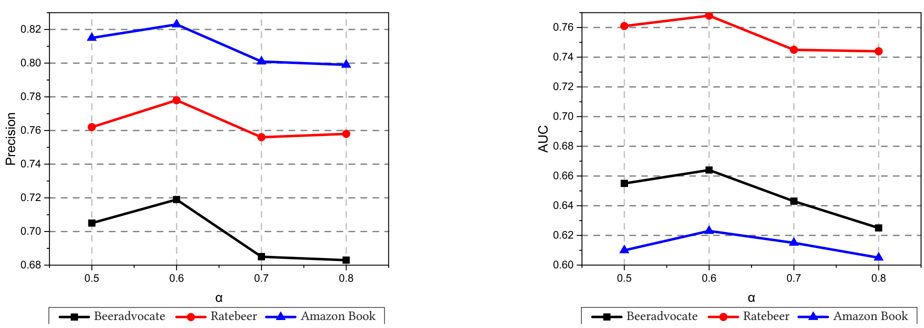


Fig. 15. Performance by varying parameter α.

Table 6. The Setting of Parameter ε in Three Datasets

Dateset	ε	Bins	Mutation time points
Amazon_Book	0.03	5	[2012-04, 2012-17, 2012-36, 2012-41]
Beeradvocate	0.05	6	[2009-47, 2010-25, 2010-51, 2011-10, 2011-20]
Ratebeer	0.06	6	[2010-09, 2010-24, 2010-44, 2011-15, 2011-50]

basically stable. Meanwhile, when $\alpha = 0.6$, ASDP achieves the best performance. It indicates that for rating prediction task, the overall rating takes more effect.

The effects of parameter ε. We set f, λ_s, λ_r and α as the default value. Through a large number of experiments, we find that the performance reaches the best when $\varepsilon = \{0.03, 0.05, 0.06\}$ for three datasets, respectively. The details are as shown in Table 6.

6.3 Comparative Studies

In this subsection, we compare our model with all baseline methods. For our method, we set $f = 5$, the balance parameter λ_s is set to be 0.05, λ_r is set to be 0.005, α is set to be 0.6, and ε is set to be $\{0.03, 0.05, 0.06\}$, respectively, for each dataset, and other parameters are fit with gradient descent. The numerical results on all of the benchmark datasets are displayed in Table 7.

Overall Performance. We can see that ASDP+, which incorporates temporal dynamics and reviews information, achieves the best performance on most cases in terms of both F1 score and AUC. The only exception is the Precision of the TimeSVD+ on Amazon-Book and Beeradavocate datasets. The reason may be that TimeSVD directly predicts the overall ratings, so it achieves

Table 7. Performance Comparison on the Rating Prediction Task

Model	Amazon-Book				Beeradvocate				Ratebeer			
	Pre	Rec	F1	AUC	Pre	Rec	F1	AUC	Pre	Rec	F1	AUC
LRPPM	0.811	0.941	0.871	0.596	0.695	0.522	0.596	0.647	0.712	0.679	0.695	0.651
SULM	0.814	0.939	0.872	0.60	0.707	0.857	0.775	0.65	0.774	0.852	0.811	0.763
TimeSVD	0.825	0.786	0.805	0.588	0.725	0.503	0.594	0.588	0.762	0.341	0.471	0.579
TimeSVD+	**0.83**	0.783	0.806	0.594	**0.733**	0.527	0.613	0.612	0.776	0.341	0.474	0.643
ASDP	0.82	0.945	0.878	0.618	0.712	0.875	0.785	0.653	0.763	**0.868**	0.812	0.743
ASDP+	0.823	**0.962**	**0.887**	**0.623**	0.719	**0.879**	**0.791**	**0.664**	**0.778**	0.853	**0.814**	**0.768**

better performance on Precision. The results validate the effectiveness of the proposed model. We conduct detailed analysis as follows.

Comparison with the basic models that only consider review text. We compare our models ASDP and ASDP+ with LRPPM and SULM, which ignore temporal dynamics information. We can see that our model ASDP+ achieves significant improvements on all the four metrics. We use the Beeradvocate dataset as an example to show the improvements. ASDP+ achieves 0.791 on F1, which is 1.6% higher than that of SULM and 19.5% higher than that of LRPPM. Meanwhile, for the performance on AUC, ASDP+ is 1.4% higher than that of SULM and 1.7% higher than that of LRPPM. It indicates that the time feature is an indispensable feature for the recommendation system, and it plays an important role in improving the accuracy of model.

Comparison with the basic models that only consider time factor. We compare our model ASDP+ with the TimeSVD+ model, which ignores review text information. It can be seen that although ASDP+ is slightly lower than TimeSVD+ on Precision for Amazon-Book and Beeradvocate datasets, it shows much better performance on the overall metrics of F1 and AUC. From Table 7, we can see that ASDP+ achieves significant improvements in all three datasets. Compare with TimeSVD+, it improves F1 score by 8.1%, 17.8%, and 34%, and improves AUC by 2.9%, 5.2%, and 12.5%, respectively. This may be because that, TimeSVD+ directly predicts the overall ratings, so it achieves better performance on Precision. However, it ignores the review text information, and thus cannot capture user's sentiment well; so the performance on F1 and AUC are not so good. The findings validate the importance of review text information.

The performance of bin segmentation algorithm based on mutation time point. We also compare the TimeSVD and TimeSVD+ models, as well as the ASDP and ASDP+ models to verify the effectiveness of our proposed bin segmentation algorithm. It can be seen from Table 7 that TimeSVD+ and ASDP+ perform better than TimeSVD and ASDP on both the F1 and AUC for all datasets. It indicates that setting the time intervals non-uniformly based on the sudden changes is better than directly fixing the bin size. It also validates the effectiveness of our bin segmentation algorithm.

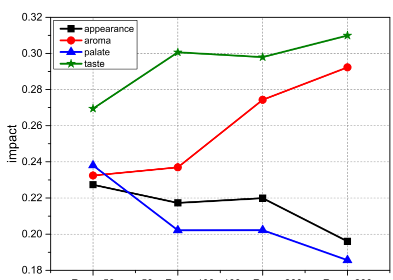
In summary, our models ASDP and ASDP+ which incorporate temporal dynamics and reviews information achieve better performance on three real datasets. It validates the effectiveness of our work. It also verifies that temporal dynamics and reviews information are very important for the recommendation system. Meanwhile, the experimental results also validate the effectiveness of our proposed bin segmentation algorithm.

6.4 Analysis of User Opinion Evolution

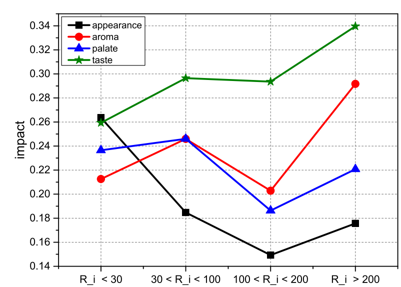
In this section, we analyze the effects of user maturity and item popularity on the model performance. Then we analyze opinion evolution for different user groups. Finally, we analyze opinion evolution for individual user.

Table 8. The Effects of User Maturity and Item Popularity

Model	Amazon-Book				Beeradvocate				Ratebeer			
	Pre	Rec	F1	AUC	Pre	Rec	F1	AUC	Pre	Rec	F1	AUC
ASDP(None)	0.752	**0.975**	0.849	0.562	0.705	0.860	0.775	0.612	0.741	0.828	0.782	0.730
ASDP(User)	0.789	0.947	0.861	0.582	0.693	**0.889**	0.779	0.638	0.748	0.839	0.791	0.733
ASDP(Item)	0.758	0.970	0.851	0.565	0.690	0.846	0.760	0.642	0.745	0.839	0.789	0.722
ASDP	**0.82**	0.945	**0.878**	**0.618**	**0.712**	0.875	**0.785**	**0.653**	**0.763**	**0.868**	**0.812**	**0.743**
ASDP+(None)	0.787	0.950	0.861	0.569	0.687	0.858	0.763	0.647	0.749	0.854	0.798	0.622
ASDP+(User)	0.801	**0.981**	0.882	0.597	0.695	**0.902**	0.785	0.624	0.752	**0.864**	0.804	0.754
ASDP+(Item)	0.806	0.964	0.878	0.572	**0.744**	0.820	0.780	0.652	0.751	0.856	0.800	0.760
ASDP+	**0.823**	0.962	**0.887**	**0.623**	0.719	0.879	**0.791**	**0.664**	**0.778**	0.853	**0.814**	**0.768**



(a) Aspect impact for different user maturity.



(b) Aspect impact for different item popularity.

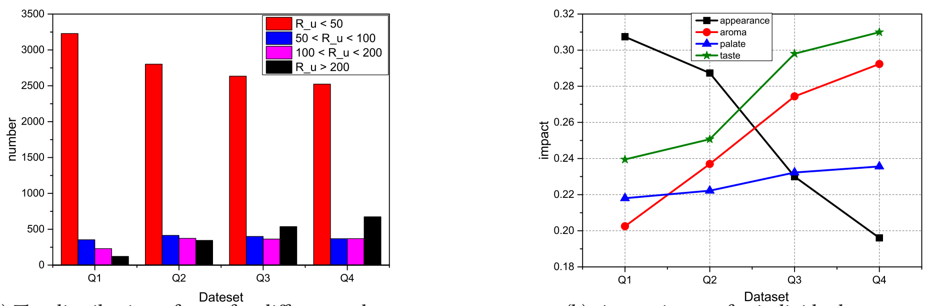
Fig. 16. Changes in aspect impacts for Beeradvocate dataset.

The effects of user maturity and item popularity. To check the effectiveness of user maturity and item popularity, we compare our model with several baseline methods, as shown in Table 8. Among them, ASDP(None) refers to ASDP that does not consider neither user maturity nor item popularity. ASDP(User) refers to ASDP that only considers user maturity but does not consider item popularity. ASDP (Item) refers to ASDP that only considers item popularity but does not consider user maturity. The same setting also applies for ASDP+. We can see that both ASDP model and ASDP+ model achieve the best performance on F1 and AUC after considering user maturity and item popularity.

We take the ASDP model on the Ratebeer dataset as an example for analysis. It can be seen that among all the four models, i.e., ASDP(None), ASDP(user), ASDP(item), and ASDP, the ASDP model achieves the best performance on F1 and AUC. ASDP(User) performs the second better; the third is ASDP(Item), and ASDP(None) performs the worst. It indicates that considering user maturity and item popularity helps to improve the performance. Meanwhile, user maturity takes more effects. ASDP(User) achieves 0.9% and 0.3% higher than ASDP(None). ASDP(Item) achieves 0.7% higher than ASDP(None) on F1, but 0.8% lower than ASDP(None). ASDP achieves 3%, 1.3% higher than ASDP(None) on F1 and AUC, respectively. Performance distribution is similar on ASDP+.

In a word, the two factors of user maturity and item popularity can help improve the performance.

Analysis of opinion evolution for different user groups. We analyze the dynamic changes in aspect impacts for different user groups. It can help us better understand the evolution of user opinions. Figure 16 displays the results in the Beeradvocate dataset. Figure 16(a) shows the dynamic changes in aspect impacts for different user maturities. Recall that, "$R_u < 50$" represents a group of users who post less than 50 reviews. In general, the more reviews a user writes, the



(a) The distribution of user for different subsets. (b) Aspect impact for individual user.

Fig. 17. Analysis of opinion evolution for individual users.

more experienced he is. As is shown in Figure 16(a), with the improvement of user experience, the user's aspect preferences are also changing over time. For the beer products, the user's preferences for "taste" and "aroma" aspects are getting higher, and that for the "appearance" and "palate" aspects are getting lower. This finding is consistent with real life: for a user, as he drinks more beer, he tends to pay more attention to the "taste" and "aroma" aspects and pay less attention to the "appearance" and "palate" aspects of the beer.

Figure 16(b) shows the dynamic changes in aspect impacts for different item popularities. Recall that "$R_i < 30$" represents a group of items that receive less than 30 reviews. In general, the more reviews an item has, the more popular it is. It can be seen that for "cold" items (i.e., "$R_i < 30$"), the "appearance" and "taste" aspect properties gain a larger impact. However, with the increase of item popularity, the impact caused by "appearance" aspect is on the decline, and the "taste" and "aroma" aspect impacts gain a big improvement. The reason may be that most users tend to be attracted by the appearance when they buy "cold" commodities. But when they buy popular commodities, they often will be attracted by the taste. This also explains why most people like popular items (as shown in Section 4.1), that is, the aspect properties of popular items are more in line with user preferences.

Analysis of opinion evolution for individual users. In order to analyze the evolution of user opinions more deeply, we also explore how individual user aspect preferences change with his maturity. Taking Beeradvocate as an example for analysis, we divide the data into four subsets in chronological order, which are denoted as Q1–Q4, and each subset contains half a year's records, as can be seen in Figure 17(a). We can see that more and more high-maturity users have emerged over time. Then we extract users with improving maturity in each subset (that is, the users posted more reviews in a bin than in a former bin, i.e., $R_u(Q1) < R_u(Q2) < R_u(Q3) < R_u(Q4)$). We regard these users as an integrated whole, and analyze his aspect preferences by taking the average value of aspect impact, as shown in Figure 17(b). We can see that as the user's maturity increases, his aspect preferences are also changing over time. For the beer products, the user's preferences for "taste" and "aroma" aspects are getting higher, and that for the "appearance" aspects is getting lower, which is also consistent with the results of our previous analysis. It also proves that our model can effectively capture the evolution of user opinions.

Through the above analysis, we can say that considering user maturity and item popularity helps to improve the performance. Moreover, it validates that our model can effectively capture the dynamic changes of user opinions.

6.5 Summary of Experiments

We conduct extensive experiments to validate the performance of our work, and check the effects of model components. In summary, we gain the following findings:

(1) Our models ASDP and ASDP+, considering dynamic changes in aspect preferences, achieve significant improvements in the accuracy for the rating prediction task.
(2) The proposed bin segmentation algorithm achieves better performance by setting the time interval non-uniformly based on the sudden changes.
(3) The two concepts of user maturity and item popularity help to improve the performance, because they can capture the dynamic changes in aspect preferences.

7 CONCLUSION

In this article, we deeply study the evolution of fine-grained user opinions in product reviews and explore it to improve personalized recommendation. To be specific, we define two concepts of user maturity and item popularity, to measure the fine-grained evolution of users and items, respectively. Next, we analyze three real datasets from both the overall-level and the aspect-level, and explore the dynamic changes (i.e., gradual changes and sudden changes) in user aspect preferences and item aspect properties. Then we propose a novel model called ASDP, to dynamically capture the change patterns of user aspect preferences and item aspect properties and to boost the performance of recommendation by mining review information and time factor. We also propose the improved ASDP+ model with a bin segmentation algorithm to set the time intervals non-uniformly based on the sudden changes. Extensive experiments validate the effectiveness of our work. In future work, we are interested in studying what causes users' opinions to change.

REFERENCES

[1] Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*. ACM, New York, NY, 717–725. DOI: https://doi.org/10.1145/3097983.3098170

[2] Muthusamy Chelliah, Yong Zheng, and Sudeshna Sarkar. 2019. Recommendation for multi-stakeholders and through neural review mining. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2979–2981.

[3] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16)*. ACM, New York, NY, 305–314. DOI: https://doi.org/10.1145/2911451.2911549

[4] Zhiyong Cheng, Ying Ding, Lei Zhu, and Mohan Kankanhalli. 2018. Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proceedings of the 2018 World Wide Web Conference (WWW'18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 639–648. DOI: https://doi.org/10.1145/3178876.3186145

[5] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*. ACM, New York, NY, 193–202. DOI: https://doi.org/10.1145/2623330.2623758

[6] Yunqi Dong and Wenjun Jiang. 2019. Brand purchase prediction based on time-evolving user behaviors in e-commerce. *Concurrency and Computation: Practice and Experience* 31, 1 (2019), e4882: 1–15. DOI: https://doi.org/10.1002/cpe.4882

[7] María Hernández-Rubio, Iván Cantador, and Alejandro Bellogín. 2019. A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. *User Modeling and User-Adapted Interaction* 29, 2 (2019), 381–441. DOI: https://doi.org/10.1007/s11257-018-9214-9

[8] Guang Neng Hu, Xin-Yu Dai, Feng-Yu Qiu, Rui Xia, Tao Li, Shu-Jian Huang, and Jia-Jun Chen. 2018. Collaborative filtering with topic and social latent factors incorporating implicit feedback. *ACM Transactions on Knowledge Discovery from Data* 12, 2 (2018), 1–30.

[9] Chunli Huang, Wenjun Jiang, Jie Wu, and Guojun Wang. 2020. Personalized review recommendation based on users' aspect sentiment. *ACM Transactions on Internet Technology* 20, 4, Article 42 (Oct. 2020), 26 pages. DOI: https://doi.org/10.1145/3414841

[10] W. Jiang and J. Wu. 2017. Active opinion-formation in online social networks. In *Proceedings of the IEEE Conference on Computer Communications*. 1440–1448.
[11] W. Jiang, J. Wu, F. Li, G. Wang, and H. Zheng. 2016. Trust evaluation in online social networks using generalized flow. *IEEE Transactions on Computers* 65, 3 (2016), 952–963.
[12] W. Jiang, J. Wu, G. Wang, and H. Zheng. 2014. FluidRating: A time-evolving rating scheme in trust-based recommendation systems using fluid dynamics. In *Proceedings of the IEEE Conference on Computer Communications*. 1707–1715.
[13] W. Jiang, J. Wu, G. Wang, and H. Zheng. 2016. Forming opinions via trusted friends: Time-evolving rating prediction using fluid dynamics. *IEEE Transactions on Computers* 65, 4 (2016), 1211–1224.
[14] Wenjun Jiang, Jing Chen, Xiaofei Ding, Jie Wu, Jiawei He, and Guojun Wang. 2021. Review summary generation in online systems: Frameworks for supervised and unsupervised scenarios. *ACM Transactions on the Web* (2021), 33. DOI: https://doi.org/10.1145/3448015
[15] Wenjun Jiang, Jie Wu, and Guojun Wang. 2015. On selecting recommenders for trust evaluation in online social networks. *ACM Transactions on Internet Technology* 15, 4 (Nov. 2015), 21 pages. DOI: https://doi.org/10.1145/2807697
[16] Noam Koenigstein, Gideon Dror, and Yehuda Koren. 2011. Yahoo! Music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys'11)*. ACM, New York, NY, 165–172. DOI: https://doi.org/10.1145/2043932.2043964
[17] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Communications of the ACM* 53, 4 (Apr. 2010), 89–97. DOI: https://doi.org/10.1145/1721654.1721677
[18] Chen Li and Wang Feng. 2013. Preference-based clustering reviews for augmenting e-commerce recommendation. *Knowledge-Based Systems* 50, C (2013), 44–59.
[19] Qiu Lin, Gao Sheng, Wenlong Cheng, and Jun Guo. 2016. Aspect-based latent factor model by integrating ratings and reviews for recommender system. *Knowledge-Based Systems* 110, C (2016), 233–243.
[20] Guang Ling, Michael R. Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14)*. ACM, New York, NY, 105–112. DOI: https://doi.org/10.1145/2645710.2645728
[21] Peng Liu, Yue Ding, and Tingting Fu. 2019. Optimal throwboxes assignment for big data multicast in VDTNs. *Wireless Networks* (March 2019), 1–11. DOI: 10.1007/s11276-019-01974-z
[22] Yining Liu, Yong Liu, Yanming Shen, and Keqiu Li. 2017. Recommendation in a changing world: Exploiting temporal dynamics in ratings and reviews. *ACM Transactions on the Web* 12, 1 (Aug. 2017), 20 pages. DOI: https://doi.org/10.1145/3108238
[23] Washington Luiz, Felipe Viegas, Rafael Alencar, Fernando Mourão, Thiago Salles, Dárlinton Carvalho, Marcos Andre Gonçalves, and Leonardo Rocha. 2018. A feature-oriented sentiment rating for mobile app reviews. In *Proceedings of the 2018 World Wide Web Conference (WWW'18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1909–1918. DOI: https://doi.org/10.1145/3178876.3186168
[24] Xuehui Mao, Shizhong Yuan, Weimin Xu, and Daming Wei. 2017. A fine-grained latent aspects model for recommendation: Combining each rating with its associated review. In *Proceedings of the International Conference on Web Information Systems Engineering*.
[25] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13)*. ACM, New York, NY, 165–172. DOI: https://doi.org/10.1145/2507157.2507163
[26] Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. ACM, New York, NY, 897–908. DOI: https://doi.org/10.1145/2488388.2488466
[27] Samaneh Moghaddam and Martin Ester. 2011. ILDA: Interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, NY, 665–674. DOI: https://doi.org/10.1145/2009916.2010006
[28] Subhabrata Mukherjee, Hemank Lamba, and Gerhard Weikum. 2016. Experience-aware item recommendation in evolving review communities. In *Proceedings of the IEEE International Conference on Data Mining*.
[29] Cataldo Musto, Marco de Gemmis, Giovanni Semeraro, and Pasquale Lops. 2017. A multi-criteria recommender system exploiting aspect-based sentiment analysis of users' reviews. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys'17)*. ACM, New York, NY, 321–325. DOI: https://doi.org/10.1145/3109859.3109905
[30] YanPing Nie, Yang Liu, and Xiaohui Yu. 2014. Weighted aspect-based collaborative filtering. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'14)*. ACM, New York, NY, 1071–1074. DOI: https://doi.org/10.1145/2600428.2609512
[31] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems* 108, C (2016), 42–49. DOI: https://doi.org/10.1016/j.knosys.2016.06.009

[32] Thiago R.P. Prado and Mirella M. Moro. 2017. Review recommendation for points of interest's owners. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media (HT'17)*. ACM, New York, NY, 295–304. DOI: https://doi.org/10.1145/3078714.3078744

[33] Li Pu and Boi Faltings. 2013. Understanding and improving relational matrix factorization in recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys'13)*. ACM, New York, NY, 41–48. DOI: https://doi.org/10.1145/2507157.2507178

[34] Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R. Zaiane. 2017. Current state of text sentiment analysis from opinion to emotion mining. *ACM Comput. Surv.* 50, 2, Article 25 (May 2017), 33 pages. DOI: http://dx.doi.org/10.1145/3057270

[35] Kumar Ravi and Vadlamani Ravi. 2015. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems* 89 (Nov. 2015), 14–46. DOI: https://doi.org/10.1016/j.knosys.2015.06.015

[36] R. Salakhutdinov. 2007. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*.

[37] Feng Wang, Jinhua She, Yasuhiro Ohyama, Wenjun Jiang, Geyong Min, Guojun Wang, and Min Wu. 2021. Maximizing positive influence in competitive social networks: A trust-based solution. *Information Sciences* 546 (2021), 559–572. DOI: https://doi.org/10.1016/j.ins.2020.09.002

[38] Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. ACM, New York, NY, 783–792. DOI: https://doi.org/10.1145/1835804.1835903

[39] Jingjing Wang, Wenjun Jiang, Kenli Li, and Keqin Li. 2021. Reducing cumulative errors of incremental cp decomposition in dynamic online social networks. *ACM Transactions on Knowledge Discovery from Data* 15, 3, Article 42 (February 2021), 33 pages. DOI: https://doi.org/10.1145/3441645

[40] Xin Wang, Roger Donaldson, Christopher Nell, Peter Gorniak, Martin Ester, and Jiajun Bu. 2016. Recommending groups to users using user-group engagement and time-dependent matrix factorization. In *Proceedings of the 13th AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 1331–1337. Retrieved from http://dl.acm.org/citation.cfm?id=3015812.3016008

[41] Ting Wu, Yong Feng, JiaXing Sang, BaoHua Qiang, and YaNan Wang. 2018. A novel recommendation algorithm incorporating temporal dynamics, reviews and item correlation. *IEICE Transactions on Information and Systems* 101, 8 (2018), 2027–2034. DOI: https://doi.org/10.1587/transinf.2017EDP7387

[42] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. 2010. Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*. ACM, New York, NY, 723–732. DOI: https://doi.org/10.1145/1835804.1835896

[43] Lina Yao, Quan Z. Sheng, Xianzhi Wang, Wei Emma Zhang, and Yongrui Qin. 2018. Collaborative location recommendation by integrating multi-dimensional contextual information. *ACM Transactions on Internet Technology* 18, 3 (Feb. 2018), 24 pages. DOI: https://doi.org/10.1145/3134438

[44] Hongzhi Yin, Bin Cui, and Chengqi Chen. 2015. Modeling location-based user rating profiles for personalized recommendation. *ACM Transactions on Knowledge Discovery from Data* 9, 3 (2015), 1–41.

[45] Hongzhi Yin, Bin Cui, Ling Chen, Zhiting Hu, and Xiaofang Zhou. 2015. Dynamic user modeling in social media systems. *ACM Transactions on Information Systems* 33, 3 (Mar. 2015), 44 pages. DOI: https://doi.org/10.1145/2699670

[46] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. ACM, New York, NY, 363–372. DOI: https://doi.org/10.1145/2484028.2484030

[47] Chenyi Zhang, Ke Wang, Hongkun Yu, Jianling Sun, and Ee-Peng Lim. 2014. *Latent factor transition for dynamic collaborative filtering*. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. 452–460. DOI: https://doi.org/10.1137/1.9781611973440.52

[48] Jifeng Zhang, Wenjun Jiang, Jinrui Zhang, Jie Wu, and Guojun Wang. 2021. Exploring weather data to predict activity attendance in event-based social network: From the organizer's view. *ACM Transactions on the Web* (2021), 25 pages. DOI: https://doi.org/10.1145/3440134

[49] S. Zhang, G. Wang, M.Z.A. Bhuiyan, and Q. Liu. 2018. A dual privacy preserving scheme in continuous location-based services. *IEEE Internet of Things Journal* 5, 5 (2018), 4191–4200. DOI: https://doi.org/10.1109/JIOT.2018.2842470

[50] Tong Zhang. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*. ACM, New York, NY, 116. DOI: https://doi.org/10.1145/1015330.1015332

[51] Yongfeng Zhang, Min Zhang, Yi Zhang, Guokun Lai, Yiqun Liu, Honghui Zhang, and Shaoping Ma. 2015. Daily-aware personalized recommendation based on feature-level time series analysis. In *Proceedings of the 24th International*

Conference on World Wide Web (WWW'15). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1373–1383. DOI: https://doi.org/10.1145/2736277.2741087

[52] Wayne Xin Zhao, Jinpeng Wang, Yulan He, Ji Rong Wen, and Xiaoming Li. 2016. Mining product adopter information from online reviews for improving product recommendation. *ACM Transactions on Knowledge Discovery from Data* 10, 3 (2016), 1–23.

[53] Peisong Zhu, Zhuang Chen, Haojie Zheng, and Tieyun Qian. 2019. Aspect aware learning for aspect category sentiment analysis. *ACM Transactions on Knowledge Discovery from Data* 13, 6 (2019), 1–21.

Received February 2020; revised November 2020; accepted December 2020