Aspect Based Recommendations: Recommending Items with the Most Valuable Aspects Based on User Reviews

Research Paper

Konstantin Bauman
Stern School of Business
New York University
kbauman@stern.nyu.edu

Bing Liu
University of Illinois
at Chicago (UIC)
liub@cs.uic.edu

Alexander Tuzhilin
Stern School of Business
New York University
atuzhili@stern.nyu.edu

ABSTRACT

In this paper, we propose a recommendation technique that not only can recommend items of interest to the user as traditional recommendation systems do but also specific aspects of consumption of the items to further enhance the user experience with those items. For example, it can recommend the user to go to a specific restaurant (item) and also order some specific foods there, e.g., seafood (an aspect of consumption). Our method is called *Sentiment Utility Logistic Model* (SULM). As its name suggests, SULM uses sentiment analysis of user reviews. It first predicts the sentiment that the user may have about the item based on what he/she might express about the aspects of the item and then identifies the most valuable aspects of the user's potential experience with that item. Furthermore, the method can recommend items together with those most important aspects over which the user has control and can potentially select them, such as the time to go to a restaurant, e.g. lunch vs. dinner, and what to order there, e.g., seafood. We tested the proposed method on three applications (restaurant, hotel, and beauty&spa) and experimentally showed that those users who followed our recommendations of the most valuable aspects while consuming the items, had better experiences, as defined by the overall rating.

CCS CONCEPTS

•Information systems → Recommender systems; Sentiment analysis; •Computing methodologies → *Factorization methods;*

KEYWORDS

Recommender systems, user reviews, sentiment analysis, user experience, aspects of user experience, user-controlled aspects.

ACM Reference format:
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 KDD '17, Halifax, NS, Canada, August 13-17, 2017,* 9 pages.
DOI: 10.1145/3097983.3098170

978-1-4503-4887-4/17/08…$15.00
DOI: 10.1145/3097983.3098170

1 INTRODUCTION

Over the last decade, there has been a great deal of interest in leveraging user reviews to provide personalized recommendations based on these reviews [6]. Much of the work in the area focuses on trying to improve estimations of user ratings of items based on the user reviews and other relevant information [6] and also to explain why some particular recommendations are given to the user based on the review information [25].

These approaches aimed at predicting and explaining ratings in terms of the user and item characteristics without taking into consideration of additional factors, such as circumstances and user's personal choices of consuming the item. For example, consider the user choosing between ordering Tiramisu or Cannoli in a cafe. Depending on what the user chooses to taste during her visit, she can give different ratings to the establishment. Therefore, user experience of a particular item can be further improved by recommending some additional aspects and personal user choices of consuming that item, such as ordering Tiramisu in that cafe.

Note that not all the aspects of the user experience can be selected by the user in order to improve her experience with the item. For example, in case of a movie, such aspects as the plot of the movie or the actors are beyond user control, which is in contrast to selecting particular dishes in the aforementioned restaurant example.

In this paper, we focus on the latter case of the *user-controlled aspects* by recommending not only particular items but also the most important aspects of consumption controlled by the user, such as ordering Tiramisu or Cannoli in a cafe. Furthermore, we can recommend certain actions to the management of an establishment (item) that can personalize experiences of the user when consuming the item (e.g., visiting the establishment). For example, we may recommend to the management of the spa salon to suggest a complimentary drink to a user because our method estimated that the user would particularly appreciate that drink in that salon which would enhance her experience there.

In this paper, we propose a method that identifies the most valuable aspects of possible user experiences of the items that the user has not tried yet and recommends the items *together* with suggestions to consume those most valuable user-controlled aspects that we have identified to be beneficial to the user. In particular, we have developed the *Sentiment Utility Logistic Model* (SULM) that takes user reviews and ratings, extracts aspects, and classifies sentiments on the aspects in the user reviews and recommends items together with the most important aspects that may enhance user experience with the items. To achieve this, the model learns how to predict the unknown ratings, sentiments that the user would express about various aspects of an item, and also identifies the impacts of these aspects on the overall rating of the item. Moreover, we use these estimated impacts to recommend the most valuable aspects to the users to enhance their experiences with the recommended items. SULM thus goes one step further and significantly enhances the functionality of the current recommender systems by providing all these additional capabilities to the traditional rating prediction and recommendation tasks.

We make the following contributions in this paper:

(1) Propose a novel approach to enhancing the functionality of the current recommender systems by recommending not only the item itself but also the specific aspects of consumption to further enhance the user experiences of the item.
(2) Develop a novel method (SULM) for identifying the most valuable aspects of future user experiences using fine-grained aspect-level sentiment analysis that automatically discovers aspects and corresponding sentiments specified by users in their reviews.
(3) Test the proposed approach on actual reviews across three real-life applications and show that our method performs well in these applications by providing recommendations of the most valuable aspects that improve user experiences. Moreover, we show that the proposed method also predicts unknown ratings of user reviews and the set of aspects that the user would mention in the reviews.

The rest of the paper is organized as follows. We discuss the related work in Section 2 and present the proposed method in Section 3. Experiments with the three real-life applications are described in Section 4 and the results are presented in Section 5. Section 6 summarizes our findings and concludes the paper.

2 LITERATURE REVIEW

Over the last few years, several papers tried to improve estimation of unknown ratings of items by extracting useful information from user reviews and leveraging this information to achieve improvements in estimation [6]. For example, the authors of [10] found six aspects in restaurant reviews and trained classifiers to identify them in the review to improve rating prediction quality. In [20], the authors calculated the sentiment of the whole review and incorporated this information into a Matrix Factorization technique.

In addition to these direct rating prediction approaches based on user reviews, there are also several proposed methods that predict user ratings relying on latent aspects inferred from user reviews. In particular, [26] presents a Latent Aspect Rating Analysis method to discover the relative importance of the topical aspects. [18] uses the LDA-based approach combined with Matrix Factorization for better prediction of unknown ratings. They obtained highly interpretable textual labels for latent rating dimensions, which helped them to "justify" particular rating values using texts of the reviews. More recently, [8], [16] and [27] went beyond [18] by using more complicated graphical models to predict unknown ratings based on collaborative filtering and topic modeling of user reviews. Their models are able to capture interpretable aspects and the sentiments on each aspect of a review. In [22] the authors presented Aspect-based Latent Factor Model (ALFM) that combines ratings and review texts to improve rating predictions. A recent work [28] presents a framework that incorporates both user opinions and preferences of different aspects. In particular, they apply the Tensor Factorization technique to the terms clustered using the LDA approach. Further, [28] uses LDA topics of terms in order to build user profiles and filter the reviews to be shown to the user.

Another research stream focuses on exploiting sentiment analysis techniques to extract useful aspects from the reviews. In particular, [31] presents the Explicit Factor Model (EFM) to generate recommendations according to the specific product aspects and [7] applies the Tensor Factorization technique to learn the ranking of the user preferences over various aspects of an item. Further, [12] applies a vertex ranking approach to the tri-partite graph of users-items-aspects to provide better recommendations of items using reviews. Finally, [29] developed an algorithm to infer the importance of aspects for the overall user opinion on the historical reviews. This method is not able to predict aspect importance for a new potential review as SULM does.

Moreover, our work is also related to Context-Aware Recommender Systems (CARS) [4]. Note that our aspects may also include contextual variables of user experiences, but they are not limited only to them. There has been much work done in CARS, including the papers dealing with user reviews [1, 11, 15]. Much of this work develops new methods for extracting contextual information from user-generated reviews and uses this information to estimate the unknown rating for an item. For example, in [1], the authors identified sentences in the review that contain contextual information based on classification and text mining techniques. They applied their method to a hotel application with the *Objective-of-trip* contextual variable. The authors of [15] proposed a method for extracting *Companion, Occasion, Time,* and *Location* contextual variables in restaurant applications based on NLP techniques. Further, [11] uses a Labeled-LDA method to categorize hotel reviews by their *Trip-type* contextual variable. All these papers showed how to improve rating predictions of items using the extracted contextual variables. Finally, [32] propose a "context suggestion" system, which is based on the collected data about contextual variables but does not work with user reviews as we do.

There is also an extensive literature on multi-criteria rating prediction [3], where such multi-criteria systems use a small number of ratings of predefined aspects (such as food quality, service quality, and ambiance in restaurant applications) to provide appropriate recommendations of items. In contrast to this type of research, we use a wide range of aspects automatically extracted from the reviews that change from one review to another and, therefore, we are not limited to the predefined fixed set of criteria. For example, review A may mention aspects x_1, x_2, x_3 and review B may mention aspects x_3, x_4, x_5, whereas the multi-criteria approach uses the same (usually small) fixed set of aspects across all the reviews.

In contrast to all the previous works, we not only predict unknown ratings of items based on user reviews as done in the prior work reviewed above, but also estimate the sentiments that a user would express on various aspects in the review and determine the impacts of the aspects on the overall predicted rating of the review about an item. Moreover, we use these estimated impacts to recommend the most valuable aspects to the users to enhance their experiences with the recommended items. Finally, we not only provide recommendations to the users but also recommend valuable aspects of user consumption to the *mangers* that can help them to run their businesses better and provide better services to their users.

In the next section, we present our proposed method.

3 OVERVIEW OF THE METHOD

In this section, we present the proposed method that identifies the most valuable aspects of possible user experiences of the items that the user has not tried yet and recommend the items together with those most important user-controlled aspects that we have identified for the user. In particular, our method consists of sentiment analysis of user reviews and subsequent training of our model called *Sentiment Utility Logistic Model* (SULM). The SULM model not only predicts the rating of a review but also identifies the impact of each aspect on the overall rating. More specifically, SULM builds user and item profiles that are used for estimating sentiment utilities and also the overall rating of the review. As a result, SULM can be used to provide recommendations of items with suggestions to experience its most valuable aspects that the model considers to be beneficial to the user.

The proposed SULM model relies on the *logistic function* to be used in Sections 3.2 and 3.3. Here we provide some background information about it before focusing on the model itself. The logistic function maps real numbers to the interval $[0, 1]$ and is defined as

$$g(t) = \frac{1}{1 + e^{-t}} \tag{1}$$

The logistic function can be applied to the classification problem as we can estimate the probability of vector x having one of the class labels $y \in \{0, 1\}$ as

$$\begin{cases} P(y = 1|x; \theta) = g(f(x, \theta)) \\ P(y = 0|x; \theta) = 1 - g(f(x, \theta)) \end{cases} \tag{2}$$

where $f(x, \theta)$ is a function, and θ is a set of parameters of the model that we will estimate. The linear case of $f(x, \theta) = a_0 \cdot x_0 + \cdots + a_n \cdot x_n$ constitutes logistic regression [5]. Assuming that the training examples x were generated independently, we can write down the likelihood of the parameters θ as:

$$\begin{aligned} \mathcal{L}(\theta) &= \prod_{x_j \in X} p(y_j|x; \theta) = \\ &= \prod_{x_j \in X} \Big(g(f(x_j, \theta))\Big)^{y_j} \cdot \Big(1 - g(f(x_j, \theta))\Big)^{(1-y_j)}. \end{aligned} \tag{3}$$

In order to find θ that maximizes $\mathcal{L}(\theta)$, we apply Stochastic Gradient Descent [30] to the log likelihood function, where the gradient step is computed based on the partial derivatives:

$$\frac{\partial}{\partial \theta_i}(\log(\mathcal{L}(\theta)) = \Big(y - g(f(x_j, \theta))\Big) \cdot \frac{\partial}{\partial \theta_i} f(x_j, \theta). \tag{4}$$

We will use equation (4) below for training the SULM model.

In the rest of this section, we describe the specifics of the proposed method.

3.1 Extracting Aspect-Sentiment Pairs

In this step we utilized the state-of-the-art "industrial-strength" sentiment analysis system *Opinion Parser (OP)* to extract aspect expressions from review text. OP is an unsupervised aspect-based sentiment analysis system. It performs two key functions, aspect extraction and aspect sentiment classification. Aspect extraction aims to extract sentiment targets on which some sentiments have been expressed. These targets are usually different aspects of entities (e.g., products or services), which are items in our context. Aspect sentiment classification classifies whether the sentiment expressed on an aspect is positive, neutral, or negative. For example, from the sentence "The food is great," "food" should be extracted as an aspect or target by the aspect extraction sub-system, and the opinion on "food" should be classified as positive by the aspect sentiment classification sub-system. The aspect extraction algorithm used in Opinion Parser is called Double Propagation (DP) [21]. It is based on the idea that a sentiment always has a target aspect, and their expressions in a sentence often have some syntactic relation. For example, the target aspect of the sentiment word "great" is "food." A dependency parser can be used to identify the relation for extracting "food" given the sentiment word "great." DP thus works as follows: Given a set S of seed sentiment words, a bootstrapping procedure is performed to extract aspects and also more sentiment words using S, and the resulting aspects and sentiment words can be used to extract more iteratively. The details of the algorithm can be found in [21]. Aspect sentiment classification is based on a set of sentiment expressions (called a sentiment lexicon), grammar analysis, and context analysis to determine whether a sentence is positive or negative about an aspect. Further details can be found in [17]. We will report its performances in Section 4.2.

In our study we apply OP to the set of reviews R at the *aspect level* for a given application (e.g., restaurant). OP builds a set of aspects $\mathbb{A}$ occurring in R and for each review $r \in R$, OP identifies a set of aspects A_r occurring in r and the corresponding sentiment opinions $o_{ui}^k \in \{0, 1\}$, where 1 is positive (like) and 0 is negative (dislike), that user u expressed about aspects $k \in A_r$ of item i.

We use the identified aspects and sentiments to train our model, as described in the rest of this section.

3.2 Aspect Sentiments

The Sentiment Utility Logistic Model (SULM) assumes that for each aspect k of the consumed item i, user u can give the *sentiment utility value* $s_{u,i}^k \in \mathbb{R}$ expressing the level of satisfaction with aspect k of item i. These utility values are not observable from user reviews. Instead of them, we observe the output of the OP which identifies only the binary value of the expressed sentiment $o_{ui}^k \in \{0, 1\}$. Therefore, we estimate the real sentiment utility values $s_{u,i}^k$ in such a way that they fit the binary sentiment values extracted from the reviews after applying the logistic function (1).

Further, SULM estimates the sentiment utility value for each of the aspects k using the matrix factorization approach [13] as

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

where μ^k is a constant pertaining to aspect k, b_u^k and b_i^k are the user's u and item's i biases of aspect k, p_u^k and q_i^k are m-dimensional latent vectors corresponding to user and item for aspect k. We denote all these coefficients by $\theta_s = (\mu, B_u, B_i, P, Q)$.

Further, we estimate parameters θ_s such that the estimated values of sentiments $\hat{o}_{u,i}^k(\theta_s) = g\left(\hat{s}_{u,i}^k(\theta_s)\right)$ fit the real binary sentiments extracted by OP as described above.

In particular, assuming that the training examples were generated independently, we search for θ_s maximizing log-likelihood

$$l_s(S|\theta_s) = \sum_{u,i,k} \left(o^k_{ui} \log \left(\hat{o}^k_{u,i}(\theta_s) \right) + (1 - o^k_{ui}) \log \left(1 - \hat{o}^k_{u,i}(\theta_s) \right) \right) \tag{6}$$

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

In this subsection, we described how to estimate the parameters of the model in order to fit the sentiments in user reviews. In the next subsection we focus on the rating estimation problem. Finally, we combine the two models into the overall SULM model for estimating both components in Section 3.4.

3.3 The Overall Satisfaction

As in the case of individual aspects, SULM assumes that user u can define the *overall level of satisfaction* with consuming item i that is measured by utility value $d_{u,i} \in \mathbb{R}$. We estimate this utility as a linear combination of the individual sentiment utility values for all the aspects in a review:

$$\hat{d}_{u,i}(\theta) = \sum_{k \in \mathbb{A}} \hat{s}^k_{u,i}(\theta_s) \cdot (z^k + w^k_u + v^k_i) \tag{7}$$

where z^k is the general coefficient expressing the relative importance of aspect k in an application, such as restaurants. Moreover, each user u may have personal preferences and specific values of importance of aspects for the overall level of satisfaction and, therefore, coefficient w^k_u represents such individual importance value of aspect k for user u. Similarly, each item i has its own specifics and coefficient v^k_i determines the importance value of aspect k for item i. We denote these new coefficients by $\theta_r = (Z, W, V)$ and the set of all coefficients in the model by $\theta = (\theta_r, \theta_s)$.

Further, in our model instead of estimating the rating that user would give to an item and minimizing the RMSE performance measure, we follow an alternative approach advocated in previous works (e.g. [2]) and classify the ratings into "like" and "dislike". In the traditional five-star rating settings, we would map "like" ratings to $\{4, 5\}$ and "dislike" to $\{1, 2, 3\}$. As a result, we transform the recommendation regression into a classification problem.

Finally, we estimate parameters θ such that the logistic transformation (1) of the overall utility value $\hat{d}_{u,i}(\theta)$ would fit binary rating $r_{u,i} \in \{0, 1\}$ that user u specified for item i

$$\hat{r}_{u,i}(\theta) = g\left(\hat{d}_{u,i}(\theta)\right). \tag{8}$$

In particular, assuming that the training examples were generated independently, we search for θ that maximizes the log-likelihood function on the training set of reviews:

$$l_r(R|\theta) = \sum_{u,i} \left(r_{ui} \cdot \log\left(\hat{r}_{u,i}(\theta)\right) + (1 - r_{ui}) \cdot \log\left(1 - \hat{r}_{u,i}(\theta)\right) \right) \tag{9}$$

In this subsection, we described how to estimate the parameters of the model in order to fit the binary ratings provided by the users. In the next subsection we combine the two models (6) and (9) into the overall SULM.

3.4 The SULM Model

The SULM model consists of two parts described in Sections 3.2 and 3.3. The main goal of SULM is to estimate the coefficients θ such that both parts of the model *simultaneously* fit the sentiments extracted from the reviews *and* the ratings provided by the users. More specifically, the SULM optimization criterion consists of the criterion from the sentiment utility part of the model (Equation (6)) and the rating prediction part of the model (Equation (9)). Moreover, we also apply regularization to avoid over-fitting. Combining all these considerations, we search for θ that minimizes:

$$Q(\theta) = -\alpha \cdot l_r(R|\theta) - (1 - \alpha) \cdot l_s(S|\theta_s) + \frac{\lambda_r}{2} \cdot \|\theta_r\|^2 + \frac{\lambda_s}{2} \cdot \|\theta_s\|^2 \tag{10}$$

where α is the parameter of the model defining the relative importance of the aspect and rating parts of the optimization criterion, and λ_r, λ_s are the regularization parameters.

3.5 Fitting the SULM Model

We apply the Stochastic Gradient Descent [30] to estimate parameters θ minimizing criterion (10). In particular, we calculate the partial derivatives $\frac{\partial Q}{\partial \theta_j}$ in order to perform the gradient step.

First, denote the difference between the real and the predicted values of rating as $\Delta^r_{u,i} = r_{u,i} - \hat{r}_{u,i}$, and the difference between the real and the predicted values of sentiment as $\Delta^s_{u,i,k} = o^k_{ui} - \hat{o}^k_{ui}$. Further, we denote the indicator function showing if user u expressed a sentiment about aspect k of item i in her review by $I^k_{u,i} \in \{0, 1\}$.

Based on (4), the partial derivative of Q by μ_k would be

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

and we denote this expression by $-\delta^k_{u,i}$. Further, we calculate the partial derivatives of Q for the rest of the parameters in θ and perform the gradient descent step as follows:

$$\begin{aligned}
\mu^k &:= \mu^k + \gamma \cdot \delta^k_{u,i} \\
b^k_u &:= b^k_u + \gamma \cdot (\delta^k_{u,i} - \lambda_s \cdot b^k_u) \\
b^k_i &:= b^k_i + \gamma \cdot (\delta^k_{u,i} - \lambda_s \cdot b^k_i) \\
p^k_u &:= p^k_u + \gamma \cdot (\delta^k_{u,i} \cdot q^k_i - \lambda_s \cdot p^k_u) \\
q^k_i &:= q^k_i + \gamma \cdot (\delta^k_{u,i} \cdot p^k_u - \lambda_s \cdot q^k_i) \\
z^k &:= z^k + \gamma \cdot (\alpha \cdot \Delta^r_{u,i} \cdot \hat{s}^k_{u,i}(\theta_s) - \lambda_r \cdot z^k) \\
w^k_u &:= w^k_u + \gamma \cdot (\alpha \cdot \Delta^r_{u,i} \cdot \hat{s}^k_{u,i}(\theta_s) - \lambda_r \cdot w^k_u) \\
v^k_i &:= v^k_i + \gamma \cdot (\alpha \cdot \Delta^r_{u,i} \cdot \hat{s}^k_{u,i}(\theta_s) - \lambda_r \cdot v^k_i).
\end{aligned} \tag{11}$$

As in the case of Matrix Factorization [14], iteratively, we first optimize the parameters in θ_s pertaining to the user by fixing the rest of parameters in θ, then optimize the parameters in θ_s pertaining to the item by fixing the rest of the parameters, and, finally, optimize the parameters in θ_r by fixing the parameters θ_s. We do it iteratively until convergence. As a result, we estimate all the parameters of the SULM model.

3.6 Aspect Impact on Ratings

In this step we apply the model trained in Section 3.5 to determine the most important aspects of user's potential experiences with the item that were discussed at the beginning of Section 3. In particular, we measure the importance of an aspect by *its weight* in the regression model (7). This means that for a potential experience of user u with item i we, first, predict sentiment utility values $\hat{s}_{ui}^k$ for each aspect $k \in \mathbb{A}$ in an application. After that, we compute the impact of each aspect k in the potential user review on the overall predicted level of satisfaction of user u with item i as a corresponding summand from the linear model (7):

$$impact_{ui}^k = \hat{s}_{ui}^k \cdot (z^k + w_u^k + v_i^k). \tag{12}$$

In other words, the impact of aspect k on the experience of user u with item i is calculated as a product of the predicted sentiment utility value $\hat{s}_{ui}^k$ and the corresponding coefficient representing the importance of aspect k of item i for user u.

These calculated aspect impacts reflect the importance of each aspect of a user review on the overall predicted rating. Note that they can be positive or negative, and we can use them to recommend positive and avoid negative experiences when users consume the recommended items, as explained in the next section.

3.7 Recommending Items and Aspects

Next, we manually identify two groups of aspects among all the aspects $\mathbb{A}$ in the application, over which (a) the user has control and (b) the management of the establishment has control. We call these groups of aspects *user-controlled* and *management-controlled* respectively. For example, aspect "gym" in a hotel application is under the user control, because she can decide whether to use it or not during her stay in the hotel. Furthermore, within these groups, we identify the most valuable aspects that we want to recommend to the user together with the item or to the management. These recommendations can be positive (suggestion to experience an aspect) or negative (suggestion to avoid an aspect). Finally, we recommend an item and the identified corresponding aspects to the user or the most important aspects to the management.

For example, if our system identified aspect "fish" as having high *positive* impact on the rating, we will recommend this restaurant *and* suggest to order fish in that restaurant to the user. Similarly, if aspect "dessert" has a strong *negative* impact on the rating, we may still recommend visiting that restaurant to the user with a suggestion *not* to order desserts there if we expect that the restaurant rating in this case to be high. Further, we can recommend such aspects to the management which are under their control and on which the management have influences. For example, we can recommend to the management of a beauty&spa salon to provide a complementary drink to the user (since it will improve her overall experience) and don't chat with her too much while in session.

In summary, we proposed a method for predicting whether a user would like an item, for estimating the sentiments that the user might express about different aspects of the item, and for identifying and recommending the most valuable user-controlled aspects of the potential user experience of the item. This method consists of sentiment analysis of user reviews, training the Sentiment Utility Logistic Model (SULM), predicting sentiment utility values, and calculating personal impact factors that each aspect contributes to the overall rating for the user. In Section 4, we show the experimental results of applying the proposed method to the real data from three applications.

	Restaurants	*Hotels*	*Beauty & Spas*
Initial	1,344,405	96,384	104,199
Filtered	602,112	5,669	5,065
Users	23,209	352	349

Table 1: Yelp Dataset Description.

Meat	Fish	Dessert	Money	Service	Decor
beef	cod	tiramisu	price	bartender	design
meat	salmon	cheesecake	dollars	waiter	ceiling
bbq	catfish	chocolate	cost	service	decor
ribs	tuna	dessert	budget	hostess	lounge
veal	shark	ice cream	charge	manager	window
pork	fish	macaroons	check	staff	space

Table 2: Examples of words pertaining to some of the aspects in the restaurant application.

4 EXPERIMENT

4.1 Dataset

To demonstrate how well our method works in practice, we tested it on the restaurant, hotel and beauty&spa applications based on the Yelp reviews[1] collected in several US cities over a period of 6 years. In this study, we selected only those users who have written at least 10 reviews. The numbers of reviews in the initial datasets, the users having more than 10 reviews, and the overall ratings generated only by those users (i.e., filtered ratings) are presented in Table 1 across the three applications.

Although Yelp uses a 5-star rating system, we transformed it into the binary "high" ($\{4, 5\}$) and "low" ($\{1, 2, 3\}$) classes, as explained in Section 3.3. Furthermore, we reformulated rating estimation as a classification problem where we estimate the probability that a user would "like" an item (by giving it a rating of 4 or 5).

4.2 Experiment Settings

We applied the method presented in Section 3 to the restaurant, hotel, and beauty&spa applications and extracted 69, 42, and 45 aspects for these applications respectively using *Opinion Parser*, as explained in Section 3.1. Table 2 presents several examples of aspects extracted from the reviews for the restaurant application, together with some of the examples of the words corresponding to those aspects. For each review, we also determine the set of aspects appearing in that review and their corresponding sentiments, as described in Section 3.1.

The aspect extraction part of OP was evaluated on 5 benchmark online review datasets [21] and it showed the *F-score* of 0.86. The evaluation of aspect sentiment classification part of OP based on 8 online review datasets [9, 17] showed the *F-score* of 0.90 on average for positive and negative sentiment classes. We also tested the performance of the OP system on our dataset. In particular, we

[1] http://www.yelp.com/dataset_challenge/dataset

selected a random sample of $3,000$ sentences from the reviews in the restaurant application and manually evaluated the aspect extraction and the sentiment classification parts of the OP. Our results are consistent with the previous studies, showing the *F-score* of 0.89 and 0.93 for the two parts of the system respectively.

All these evaluations show that OP performs well in general and specifically in our application. In this study, we focus on leveraging the output of OP[2] for providing recommendations of items and their most valuable aspects.

Further, for each application, the set of reviews R is partitioned into training and testing sets in the ratio of 80% to 20%. We also use cross-validation on the training set to find the best parameters of the SULM model, including α, λ_s, and λ_r parameters, that maximize the prediction performance measures to be introduced in the next section. In particular, we found that $\alpha = 0.5$ provides the best balance between the performances of predicting aspect sentiments and user ratings across the three applications.

We trained SULM on MacBook Air 1.4 GHz Intel Core i5. It took less than a minute to train the model for hotels and beauty&spa applications ($\sim 4,000$ reviews) and about one hour to train SULM for the restaurant application ($\sim 480,000$ reviews).

After training the model on the restaurant, hotel, and beauty&spas applications, we predict the unknown ratings and sentiments on the test data (reviews) and also determine the impacts of the aspects on the predicted ratings. Moreover, as explained in Section 3.7, among all aspects of the restaurant, hotel, and beauty&spa applications, we identified that the user has control over 49, 14, and 17 aspects respectively, and the managements of the establishments have control over 54, 29, and 31 aspects respectively. Further, SULM provides recommendations to experience positive or avoid experiencing negative aspects of the item from these identified sets.

The results of these experiments are presented in the next section.

4.3 Evaluation Methodology

When running the model as described in Section 4.2 on the data from Section 4.1, we measure its performance in terms of how the recommendations of the most valuable aspects affect the overall rating, how well the model predicts if the user would like (or dislike) the recommended items, and how well it predicts which of the aspects would appear in the potential user review.

Aspect Recommendations. The main point of SULM's performance is the effect it produces on the overall user experience with an item by providing recommendations of additional most valuable aspects. To estimate this effect we use the measure of how the recommendations of specific aspects (described in Section 3.7) affect the overall rating. In particular, we measure *how much the average overall rating is changed for those users who "follow" our recommendations.* We assume that the user *followed* our positive recommendations of additional aspects if he/she mentioned this aspect in the review. We expect that positive recommendations of aspects would increase the average rating, while not-following negative recommendations would decrease it. Note that in the case of negative recommendations we advise the user to *avoid* experiencing the specified aspects. Similarly, we measure how much the average overall rating is changed for those items where the managers follow our recommendations by suggesting the aspect consumptions to the user. As before, we assume that the management followed our positive recommendation of an additional aspect if the user mentioned this aspect in the review (e.g., provided a complimentary drink which was mentioned in the review). Note, that we calculate the described measure for users and management separately. It means that if the consumed aspect was recommended to the user *and* to the management, we count it as both the user and the management followed our recommendations. This point does not reduce the power of the results which show that the most valuable aspect should be consumed in order to enhance the overall user experience.

Aspect and Rating Predictions. In addition to measuring the effect of the recommendations of the most valuable aspects, we also evaluate how well the model predicts which of the aspects would appear in the user reviews. Although SULM does not focus on this type of predictions, we assume that users tend to discuss the most valuable aspects in their reviews. Therefore, we use the absolute values of aspects impacts on the predicted rating (described in Section 3.6) to predict the list of aspects that user would discuss in the review. In particular, for each potential user experience we first rank all the aspects $\mathbb{A}$ from the application (e.g. restaurants) according to these absolute values of aspects impacts. Then we select the *top-n* of the ranked aspects and examine how many of them appear in each review. In other words, we compute the precision measure of the $TopN$ most important aspects appearing in a review. Finally, we also calculate the rating prediction performance of the SULM model using the standard performance measures *Precision@Top3* and AUC [23].

We computed all these measures on the test data and present our results and their comparison with the baselines in the next section.

5 RESULTS

In this section, we present the results of how well the proposed method performed on the restaurant, hotel and beauty&spa applications across the three measures described in Section 4.3, each measure being described in a separate subsection (5.1-5.3). In particular, in Section 5.1 we compare the performance of SULM aspect recommendations with certain baselines. In Section 5.2 we compare aspect ranking performance of our algorithm with another baseline. Finally, in Section 5.3 we compare our method with the baselines in terms of rating prediction performance.

Note, that we compare the performance of our method with different baselines in the three subsections because our model in addition to the standard rating prediction provides a novel aspect recommendation functionality which is not supported by the baselines and it is thus not possible to compare our model uniformly with them across all the aforementioned performance metrics.

5.1 Recommendations of Aspects

In this subsection, we evaluate the recommendations of *user- controlled* aspects presented in Section 3.7 based on the performance measure described in Section 4.3. In particular, we measure how much the ratings have changed for those users who "follow" our recommendations on the test set by mentioning the recommended

[2]The result of applying OP to the Yelp dataset is available online: http://bit.ly/Yelp_aspect_dataset.

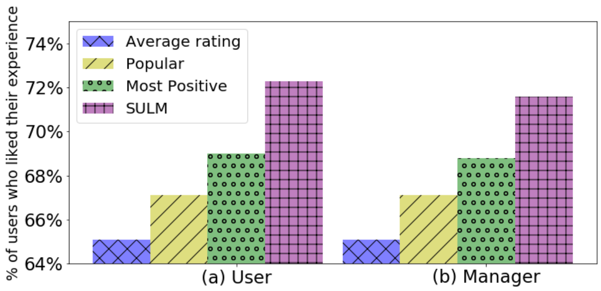

Figure 1: Average ratings for (a) Users and (b) Managers who followed positive recommendations of additional aspects in the restaurant application.

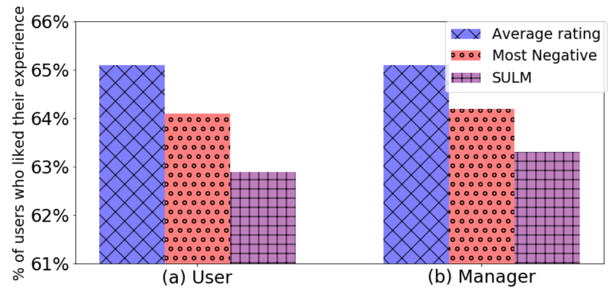

Figure 2: Average ratings for (a) Users and (b) Managers who did not follow negative recommendations of additional aspects in the restaurant application.

aspect in the review as described in Section 4.3. In addition to the *average rating* on the test set, we compare our results with three strong baseline approaches. These baselines basically indicate the strengths and weaknesses of the establishments based on their user reviews:

- *Popular Aspect:* this approach recommends users to experience the most popular aspect of an item, i.e. the most frequently mentioned aspect in the historical reviews of the item.
- *Most Positive Aspect:* this approach identifies and recommends the user to experience the most positive aspect of an item in the sense that it has the highest average sentiment rating taken over all the frequently occurring aspects of an item, i.e., those aspects that appear more than k times in the reviews of an item (we set $k = 5$ in our experiments).
- *Most Negative Aspect:* this approach identifies and recommends the user to *avoid* the most negative aspect of an item. Similarly to the most positive case, this approach identifies an aspect of an item with the lowest average sentiment rating among the set of frequent aspects, i.e. appearing in more than k historical reviews of an item.

The results of our experiments are presented in Table 3 that compares the performance of our method with the baselines across the three applications in terms of average overall rating for different groups of reviews. Furthermore, Figures 1-2 graphically show the same comparison results for the restaurants application. As Table 3 and Figures 1-2 demonstrate, our method significantly outperforms baseline approaches. In case of restaurants (Figure 1 (a)), users liked their experiences in 65.1% of the cases on average in the test set, whereas, those users who followed recommendations of the *most popular* aspect of an item liked their experience in 67.1% of the cases. Moreover, those users who followed recommendations of the *most positive* aspects, liked their experiences in 69.0% of the cases which is significantly higher that the performance result of 67.1% for the *most popular* case. Finally, our SULM model achieved the performance result of 72.3% which is again significantly higher than the previously described other baselines (based on t-test with p-value < 0.05). These comparisons show that recommendations of the aspects provided by our SULM model can help customers to get better experiences with items than the baseline approaches.

Similarly, those users who *did not* follow our *negative* recommendations[3] (and experienced the negative aspect of an item against our advice) gave lower ratings to the items than the average rating of the items given by all users in the application and those users who did not follow the recommendations provided with the baseline approach. For example, in the restaurant application (Figure 2 (a)) the users who did not follow the negative recommendations of the most negative aspects liked their experiences with the items in 64.1% of the cases, while on average users like the items in 65.1% of cases. Furthermore, those users who did not follow negative recommendations provided by our SULM model liked the items in only 62.9% of the cases, which is significantly lower (p-value < 0.05) than the result of the baseline approach. These comparisons demonstrate that negative recommendations provided by our SULM model can help customers to avoid more negative experiences with items than the baseline approach.

As described in Section 3.7, SULM provides similar recommendations of the most valuable aspects not only to users but also to managers (we call them *management-controlled* aspects). As for the users, Table 3 and Figures 1-2 also present the results of aspects recommendations to the management of the establishments. These results show that managers who "followed" our *positive* recommendations (as explained in Section 4.3), obtained higher ratings for the user experiences than the managers who followed recommendations provided with the baseline approaches. For example, in the restaurant application (Figure 1 (b)) when the managers followed SULM's positive recommendations of *additional aspects*, users liked their experiences in 71.6% of the cases, whereas, in those cases when the managers followed recommendations of the *most popular* or the *most positive* baseline approaches, users liked their experiences in only 67.1% and 68.8% of the cases respectively. These numbers are significantly lower than the result of the proposed SULM model (p-value < 0.05), which demonstrates that our method can help managers to provide better experiences for the users than the baseline methods.

Furthermore, as Table 3 also shows, these results hold not only for the restaurants but also across hotels and beauty&spas domains. In conclusion, our SULM method outperformed the baselines and help users to get (and managers to provide) better experiences with recommended items.

[3] "do not consume aspect k of item i"

	Restaurants		*Hotels*		*Beauty & Spas*	
	users	*managers*	*users*	*managers*	*users*	*managers*
Average	65.1%		58.0%		71.2%	
Followed *Popular*	67.1%	67.1%	62.2%	62.8%	72.0%	71.9%
Followed *Most Positive*	69.0%	68.8%	65.7%	65.2%	72.4%	72.7%
Followed *Positive SULM*	**72.3%**	**71.6%**	**68.0%**	**67.7%**	**76.1%**	**75.7%**
Not followed *Most Negative*	64.1%	64.2%	57.9%	57.9%	70.5%	70.4%
Not followed *Negative SULM*	**62.9%**	**63.3%**	**57.2%**	**57.6%**	**67.8%**	**67.5%**

Table 3: Average fraction of liked items for the users who followed (or not) our positive/negative recommendations of additional aspects.

	Precision@Top3			*Precision@Top5*		
Application	*R*	*H*	*B&S*	*R*	*H*	*B&S*
LRPPM	0.20	0.41	0.24	0.16	0.34	0.22
SULM	0.19	0.40	0.22	0.16	0.33	0.19

Table 4: Aspect Ranking Performance.

	Precision@Top3			*AUC*		
Application	*R*	*H*	*B&S*	*R*	*H*	*B&S*
LRPPM	0.801	0.822	0.845	0.694	0.725	0.637
HFT	0.821	0.821	0.842	0.714	0.756	0.651
SULM	0.818	0.849	0.862	0.707	0.745	0.663

Table 5: Rating Prediction Performance.

5.2 Aspect Ranking Performance

In addition to measuring the effect of recommendations of the most valuable aspects, we also evaluate how well SULM predicts which of the aspects would appear in the user reviews as described in Section 4.3. As a baseline, we use the following popular approach

(1) *Learning to Rank User Preferences Based on Phrase-Level Sentiment Analysis Across Multiple Categories (LRPPM)* [7], which is a model trained on the results of sentiment mining of user reviews. LRPPM predicts user ratings and ranks the aspects of user reviews according to the probability of these aspects to appear in possible future reviews. We download the LRPPM system from the authors' website.

The results of this comparison are presented in Table 4. As Table 4 shows, our method is comparable to the LRPPM baseline, even though our model does not optimize for the *Precision@TopN* ($N = 3$ and $N = 5$) aspect ranking performance, whereas the LRPPM model does. We explain this interesting result by conjecturing that users tend to discuss those aspects in the reviews that have the highest impact on the overall rating, and that this is the cause of the strong performance of the SULM model. We plan to explore this conjecture further as a part of future research. Note that LRPPM predicts only the importance of aspects for the user in an item, but it does not take into account user sentiments. It means that LRPPM cannot recommend experiencing the most valuable aspects and, therefore, SULM outperforms it in functionality by providing such additional capability as we discussed in the previous subsection.

Furthermore, as Table 4 shows, aspects are predicted significantly better for hotels than for restaurants and beauty & spas. This is the case because some aspects of hotels, such as "room" and "service," are very popular in the reviews and, therefore, are easily predictable.

5.3 Rating Prediction Performance

As explained in Section 4.3, we measure how well we predict if the user would like (or dislike) the recommended items. We compare the performance of our model with the following baseline approaches

(1) *LRPPM* model [7] that we described in Section 5.2
(2) *Hidden Factors as Topics (HFT)* [18] provides the state-of-the-art approach that incorporates user reviews into a rating prediction model. In particular, HFT combines the Matrix Factorization and the Latent Dirichlet Allocation (LDA) models to simultaneously train on the ratings and the texts of the reviews.

We selected these two baseline models because they constitute the state-of-the-art in combining rating predictions and user reviews. In particular, it is shown in [18] that HFT outperforms the classical Matrix Factorization (MF) approach [14]. We have also selected the LRPPM model because, as is shown in [7], this model outperformed several previously proposed rating and aspect ranking models, such as Probabilistic Matrix Factorization (PMF) [24], Explicit Factor Model (EFM) [31] and Rating-based Tensor Factorization (RTF) [7], and therefore also constitutes a strong state-of-the-art baseline. Comparing our proposed approach with these two models is sufficient for our purposes because these are strong baselines that outperformed various other approaches discussed in Section 2, including EFM, PMF, and RTF. Furthermore, the other models discussed in Section 2, such as [11, 20, 22, 27], are not directly comparable with SULM because they focus only on the rating prediction problem.

The results of the rating prediction performance of SULM and the two baselines are presented in Table 5. As Table 5 shows, SULM outperformed the LRPPM model across all the applications and performance measures and performed comparably with the HFT model. In particular, its performance is better than that of HFT for the hotel and beauty&spas application for the *Precision@Top3* measure and for the beauty&spas application in terms of AUC measure and is slightly worse in other cases.

Although the performance results of the SULM and HFT models are comparable for the rating prediction problem, SULM has more extensive functionality than HFT which only predicts ratings based on reviews and, therefore, SULM dominates HFT in general, as will be explained subsequently and further summarized in Section 6.

In summary, the SULM model performs well in predicting unknown ratings (at the level of the state-of-the-art HFT model [19]), estimating the aspects that a user would specify in a review (at the level of the state-of-the-art LRPPM model [7]), and determining the impacts of various aspects on the overall rating of the review. However, the main advantage of the proposed SULM model is the new additional functionality of providing not only recommendations of items to users, but also recommendations of the most valuable aspects that may enhance user experiences with items. Note, that none of the baseline approaches support all these capabilities in one system. We showed that those users who followed our recommendations of important aspects rated their experiences significantly higher than the users who followed recommendations from the baseline approaches. Furthermore, SULM provides recommendations not only to the users but also to the *mangers* which can help them to provide better services to the users. All this demonstrates that SULM significantly enhances the functionality of the current recommender systems by providing all these additional capabilities to the traditional rating prediction and item recommendation tasks.

6 CONCLUSION

In this paper, we presented a method that identifies the most valuable user-controlled aspects of possible user experiences of the items and recommends the items together with suggestions to consume those most valuable aspects. The paper makes the following contributions. First, it proposed a novel approach to enhance the functionality of recommender systems by recommending not only the item itself but also some positive aspects of the item to further enhance user experiences with the item. Second, in this paper we developed a method *Sentiment Utility Logistic Model (SULM)* for identifying the most valuable aspects of future user experiences that is based on the sentiment analysis of user reviews. Third, we tested our method on actual reviews across three real-life applications and showed that the proposed method performed well on these applications in the following sense. First of all, recommendations of a set of valuable *aspects* worked well as those users who followed our recommendations rated their experiences significantly higher than those who followed the baseline recommendations. Our method also managed to predict the unknown ratings of the reviews at the level commensurate with the state-of-the-art HFT model [18]. In addition, it predicted the set of aspects that the user would mention in a possible future review of an item at the level of the state-of-the-art LRPPM [7]. Moreover, SULM provides recommendations not only to the users but it also recommends valuable aspects of user experiences to the *mangers* of the establishments that can help them to provide better services to the users.

As shown in Section 2, most of the existing works focused on either leveraging user reviews to improve the rating prediction (e.g. HFT model [18]), or predicting the set of aspects that the user might include in her review (e.g. LRPPM [7]), or predicting the sentiments for each individual aspect of the user experience (e.g. [3]). SULM significantly enhances the functionality of these systems by providing recommendations of not only items of interest to the users but also additional aspects that may enhance user experiences with those items.

REFERENCES

[1] Silvana Aciar. 2010. Mining context information from consumers reviews. In *Proceedings of Workshop on Context-Aware Recommender System*. ACM.
[2] G. Adomavicius and Y. Kwon. 2007. New Recommendation Techniques for Multicriteria Rating Systems. *IEEE Intelligent Systems* 22, 3 (May 2007), 48–55.
[3] Gediminas Adomavicius and YoungOk Kwon. 2015. *Multi-Criteria Recommender Systems*. Springer US, Boston, MA, 847–880.
[4] Gediminas Adomavicius and Alexander Tuzhilin. 2015. *Context-Aware Recommender Systems*. Springer US, Boston, MA, 191–226.
[5] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.
[6] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. In *User Modeling and User-Adapted Interaction*.
[7] Xu Chen, Zheng Qin, Yongfeng Zhang, and Tao Xu. 2016. Learning to Rank Features for Recommendation over Multiple Categories. In *ACM SIGIR*. 305–314.
[8] 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 *(KDD)*.
[9] Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A Holistic Lexicon-based Approach to Opinion Mining *(WSDM '08)*. ACM, 10.
[10] Gayatree Ganu, Yogesh Kakodkar, and AméLie Marian. 2013. Improving the Quality of Predictions Using Textual Information in Online User Reviews. *Inf. Syst.* 38, 1 (March 2013), 1–15.
[11] N. Hariri, B. Mobasher, R. Burke, and Y. Zheng. 2011. Context-Aware Recommendation Based On Review Mining *(ITWP)*.
[12] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. TriRank: Review-aware Explainable Recommendation by Modeling Aspects. In *ACM CIKM*.
[13] Yehuda Koren and Robert Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, Boston, MA, 77–118.
[14] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (aug 2009), 30–37.
[15] Yize Li, Jiazhong Nie, Yi Zhang, Bingqing Wang, Baoshi Yan, and Fuliang Weng. 2010. Contextual Recommendation Based on Text Mining. In *COLING: Posters*.
[16] Guang Ling, Michael R. Lyu, and Irwin King. 2014. Ratings Meet Reviews, a Combined Approach to Recommend *(RecSys '14)*. ACM, New York, NY, USA, 8.
[17] Bing Liu. 2015. *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press. DOI:http://dx.doi.org/10.1017/CBO9781139084789
[18] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text *(RecSys '13)*. ACM, 8.
[19] Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning Attitudes and Attributes from Multi-aspect Reviews *(ICDM '12)*. 1020–1025.
[20] Štefan Pero and Tomáš Horváth. 2013. *Opinion-Driven Matrix Factorization for Rating Prediction*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–13.
[21] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction Through Double Propagation. *Comput. Linguist.* (2011).
[22] Lin Qiu, Sheng Gao, Wenlong Cheng, and Jun Guo. 2016. Aspect-based latent factor model by integrating ratings and reviews for recommender system. *Knowledge-Based Systems* 110 (2016), 233 – 243.
[23] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). 2011. *Recommender Systems Handbook*. Springer US.
[24] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo. In *ICML*. ACM, 880–887.
[25] Nava Tintarev and Judith Masthoff. 2015. Explaining Recommendations: Design and Evaluation. In *Recommender Systems Handbook 2nd edt.*
[26] Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. In *ACM SIGKDD*.
[27] Yinqing Xu, Wai Lam, and Tianyi Lin. 2014. Collaborative Filtering Incorporating Review Text and Co-clusters of Hidden User Communities and Item Groups. In *ACM CIKM*. 251–260.
[28] Chong Yang, Xiaohui Yu, Yang Liu, Yanping Nie, and Yuanhong Wang. 2016. Collaborative filtering with weighted opinion aspects. *Neurocomputing* 210 (2016), 185 – 196.
[29] Z. J. Zha, J. Yu, J. Tang, M. Wang, and T. S. Chua. 2014. Product Aspect Ranking and Its Applications. *IEEE Transactions on Knowledge and Data Engineering* 26, 5 (May 2014), 1211–1224.
[30] Tong Zhang. 2004. Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms. In *ICML*. 919–926.
[31] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit Factor Models for Explainable Recommendation Based on Phrase-level Sentiment Analysis. In *ACM SIGIR*.
[32] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2016. User-Oriented Context Suggestion. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization (UMAP '16)*. ACM, New York, NY, USA, 249–258.