Rating Prediction Based on Social Sentiment From Textual Reviews

Xiaojiang Lei, Xueming Qian, *Member, IEEE*, and Guoshuai Zhao

Abstract—In recent years, we have witnessed a flourish of review websites. It presents a great opportunity to share our viewpoints for various products we purchase. However, we face an information overloading problem. How to mine valuable information from reviews to understand a user's preferences and make an accurate recommendation is crucial. Traditional recommender systems (RS) consider some factors, such as user's purchase records, product category, and geographic location. In this work, we propose a sentiment-based rating prediction method (RPS) to improve prediction accuracy in recommender systems. Firstly, we propose a social user sentimental measurement approach and calculate each user's sentiment on items/products. Secondly, we not only consider a user's own sentimental attributes but also take interpersonal sentimental influence into consideration. Then, we consider product reputation, which can be inferred by the sentimental distributions of a user set that reflect customers' comprehensive evaluation. At last, we fuse three factors—user sentiment similarity, interpersonal sentimental influence, and item's reputation similarity—into our recommender system to make an accurate rating prediction. We conduct a performance evaluation of the three sentimental factors on a real-world dataset collected from Yelp. Our experimental results show the sentiment can well characterize user preferences, which helps to improve the recommendation performance.

Index Terms—Item reputation, rating prediction, recommender system (RS), reviews, sentiment influence, user sentiment.

I. INTRODUCTION

THERE is much personal information in online textual reviews, which plays a very important role on decision processes. For example, the customer will decide what to buy if he or she sees valuable reviews posted by others, especially user's trusted friend. We believe reviews and reviewers will do help to the rating prediction based on the idea that high-star ratings may greatly be attached with good reviews. Hence, how to mine reviews and the relation between reviewers in social networks has become an important issue in web mining, machine learning and natural language processing.

We focus on the rating prediction task. However, user's rating star-level information is not always available on many review websites. Conversely, reviews contain enough detailed product information and user opinion information, which have great reference value for a user's decision. Most important of all, a given user on website is not possible to rate every item. Hence, there are many unrated items in a user-item-rating matrix. It is inevitable in many rating prediction approaches, e.g., [1], [4], [14], [33], [65]. Review/comment, as we all know, is always available. In such case, it's convenient and necessary to leverage user reviews to help predicting the unrated items.

The rise like DouBan,[1] Yelp[2] and other review websites provides a broad thought in mining user preferences and predicting user's ratings. Generally, user's interest is stable in short term, so user topics from reviews can be representative. For example, in the category of Cups & Mugs, different people have different tastes. Some people pay attention to the quality, some people focus on the price and others may evaluate comprehensively. Whatever, they all have their personalized topics. Most topic models introduce users' interests as topic distributions according to reviews contents [10], [13], [17], [25], [31]. They are widely applied in sentiment analysis [37], travel recommendation [21], [27], [34], and social networks analysis [19].

Sentiment analysis is the most fundamental and important work in extracting user's interest preferences. In general, sentiment is used to describe user's own attitude on items. We observe that in many practical cases, it is more important to provide numerical scores rather than binary decisions. Generally, reviews are divided into two groups, positive and negative. However, it is difficult for customers to make a choice when all candidate products reflect positive sentiment or negative sentiment. To make a purchase decision, customers not only need to know whether the product is good, but also need to know how good the product is. It's also agreed that different people may have different sentimental expression preferences. For example, some users prefer to use "good" to describe an "excellent" product, while others may prefer to use "good" to describe a "just so so" product [20].

In our daily life, customers are most likely to buy those products with highly-praised reviews. That is, customers are more concerned about item's reputation, which reflects consumers' comprehensive evaluation based on the intrinsic value of a specific product. To obtain the reputation of a product, sentiment in reviews is necessary. Normally, if item's reviews reflect positive sentiment, the item may be with good reputation to a great

---

[1][Online]. Available: http://www.douban.com
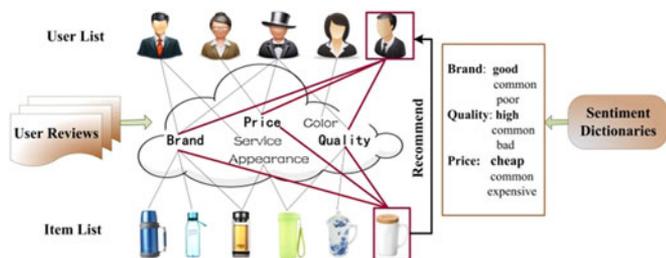[2][Online]. Available: http://www.yelp.com

Fig. 1. Product features that user cares about are collected in the cloud including the words "Brand," "Price," and "Quality." By extracting user sentiment words from user reviews, we construct the sentiment dictionaries. The last user is interested in those product features, so based on the user reviews and the sentiment dictionaries, the last item will be recommended.

extent. Oppositely, if item's reviews are full of negative sentiment, then the item is to be with bad reputation. To a given product, if we know user sentiment, we can infer the reputation and even the comprehensive ratings. When we search the net for purchasing, both positive reviews and negative reviews are valuable to be as reference. For positive reviews, we can know the advantages of a product. For negative reviews, we can obtain the shortcomings in case of being cheated. So it's worth to explore those reviewers who have obvious and objective attitude on items. We observe that reviewers' sentiment will influence others: if a reviewer has clear like and dislike sentiment, other users will pay much attention to him/her. However, user's sentiment is hard to predict and the unpredictability of interpersonal sentimental influence makes a great difficulty in exploring social users.

In addition to extracting user preferences, there is much work paying attention to the interpersonal interaction. Many approaches about the interpersonal influence in social networks have proved good performance in recommendation, which can effectively solve the "cold start" problems. However, the existing approaches [2], [3], [8], [9], [18], [51] mainly leverage product category information or tag information to study the interpersonal influence. These methods are all restricted on the structured data, which is not always available on some websites. However, user reviews can provide us ideas in mining interpersonal inference and user preferences.

To address these problems, we propose a sentiment-based rating prediction method (RPS) in the framework of matrix factorization. In our work, we make use of social users' sentiment to infer ratings. Fig. 1 is an example that illustrates our motivation. First, we extract product features from user reviews. Then, we find out the sentiment words, which are used to describe the product features. Besides, we leverage sentiment dictionaries to calculate sentiment of a specific user on an item/product. What is more, we combine social friend circle with sentiment to recommend. In Fig. 1, the last user is interested in those product features, so based on the user reviews and the sentiment dictionaries, the last item will be recommended. Compared with previous work [2]–[5], [8], [9], the main difference is that: we use unstructured information to recommend instead of other structured social factors. Compared with [6], [20], [39], [59], [60], the main difference is that: their work mainly focuses on

classifying users into binary sentiment (i.e., positive or negative), and they do not go further in mining user's sentiment. In our paper, we not only mine social user's sentiment, but also explore interpersonal sentimental influence and item's reputation. Finally, we take all of them into the recommender system.

The main contributions of our approach are as follows: 1) we propose a user sentimental measurement approach, which is based on the mined sentiment words and sentiment degree words from user reviews. Besides, some scalable applications are proposed. For example, we explore how the mined sentiment spread among users' friends. What is more, we leverage social users' sentiment to infer item's reputation, which showed great improvement in accuracy of rating prediction. 2) We make use of sentiment for rating prediction. User sentiment similarity focuses on the user interest preferences. User sentiment influence reflects how the sentiment spreads among the trusted users. Item reputation similarity shows the potential relevance of items. 3) We fuse the three factors: user sentiment similarity, interpersonal sentimental influence, and item reputation similarity into a probabilistic matrix factorization framework to carry out an accurate recommendation. The experimental results and discussions show that user's social sentiment that we mined is a key factor in improving rating prediction performances.

The remainder of this paper is organized as follows. In Section II, we present the related work about rating prediction in recommender systems. In Section III, the proposed sentiment-based RPS is described thoroughly. Experiments and discussion are given in Section IV. Conclusions and future work are drawn in Section V.

II. Related Work

In this section, we survey recent work related to our approach. Firstly, we review some approaches based on collaborative filtering (CF). Then, we review the often utilized rating prediction/recommendation methods based on matrix factorization. Also, the review based approaches as well as the sentiment mining and applications are provided in detail.

A. Collaborative Filtering

The task of CF is to predict user preferences for the unrated items, after which a list of most preferred items can be recommended to users. To improve recommendation performance, many CF algorithms have been proposed [7], [18], [22]–[24], [26], [28], [35]. One of the most well known CF algorithms is the user-based CF algorithm proposed in [35]. The basic idea is that people expressed similar preferences in the past will prefer to buy similar items in the future. Tso-Sutter *et al.* [18] propose a generic method that allows tags to be incorporated to standard CF algorithms and to fuse the 3-dimensional correlations between users, items and tags. Moreover, item-based CF algorithm [22] produces the rating from a user to an item based on the average ratings of similar or correlated items by the same user. It obtains better performance in computing the similarity between items. Gao *et al.* [24] propose a review expert collaborative recommendation algorithm based on the assumption that those projects/experts with similar topics have similar feature vectors.

Fletcher and Liu [26] propose a CF-based service recommendation method that considers users' personalized preferences on nonfunctional attributes.

B. Matrix Factorization-Based Approaches

1) Basic Matrix Factorization: Matrix factorization is one of the most popular approaches for low-dimensional matrix decomposition. Here, we review the Basic MF [1]. The rating matrix $R \in \boldsymbol{R}_{m \times n}$ (m is the number of users and n is the number of items) can be predicted according to (1), where $U_u \in \boldsymbol{U}_{m \times k}$ denotes the user Potential Eigen vectors matrix and $P_i \in \boldsymbol{P}_{n \times k}$ denotes item Potential Eigen vectors matrix, and k is the dimension of the vectors. $\hat{R}_{u,i}$ denotes the predicted objective star level of item i, $\bar{R}$ denotes the average value of all ratings

$$\hat{R}_{u,i} = \bar{R} + U_u P_i^T . \tag{1}$$

We learn Potential Eigen vectors of users and items on the observed rating data by minimizing the objective function. The objective function Ψ is defined as follows:

$$\Psi\left(\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{P}\right) = \frac{1}{2} \sum_{u,i} \left(R_{u,i} - \hat{R}_{u,i}\right)^2 + \frac{\lambda}{2}\left(||\boldsymbol{U}||_F^2 + ||\boldsymbol{P}||_F^2\right) \tag{2}$$

where $||\boldsymbol{X}||_F$ is the Frobenius norm of matrix $\mathbf{X}$, which is utilized to avoid over-fitting. The optimization of the objective function can be solved by gradient descent method [8].

2) Social Recommendation: Some matrix factorization based social recommendations are proposed to solve the "cold start" problems. Jamali and Ester [4] explore a matrix factorization based approach for recommendation in social networks. They incorporate the mechanism of trust propagation into the recommendation model. Trust propagation has been shown to be a crucial factor in social network analysis and in trust-based recommendation. Yang *et al.* [2] propose the concept of "Trust Circles" in social networks. Their model outperforms the Basic MF [1] and Social MF [4]. The trusted value between users is represented by a matrix $\mathbf{S}$, and directed and weighted social relationship of user u with user v is represented by a positive value $S_{u,v}^{c*} \in [0, 1]$. The basic idea is that the user latent feature should be similar to the average of his/her friends' latent features with weight of $S_{u,v}^{c*}$ in category c. Except for the factor of interpersonal influence in [2], Jiang *et al.* [3] propose another important factor, the individual preference. They conduct experiments on Renren dataset and Tencent Weibo dataset in China, and the results demonstrate the significance of social contextual factors (individual preference and interpersonal influence) in their model. Qian *et al.* [8] propose a personalized recommender model (*PRM*) combing with user interpersonal interest similarity, interpersonal influence and personal interest factor. They make use of categories of products, and user personal interest is the main contributions. Wang *et al.* [57] propose to use social propagation simulation and content similarity analysis to update the user-content matrix. They also construct a joint social-content space to measure the relevance between users and videos, which provides a high accuracy for video importing and re-sharing recommendation. However, some websites do not always offer structured information, and all of these methods do not leverage users' unstructured information, i.e., reviews. In addition, there also remain a few questions: some users may have no social relation with each other or even worse, explicit social networks information is not always available and it is difficult to provide a good prediction for each user. In this paper, we elaborate the sentiment factor to improve social recommendation.

C. Reviews-Based Applications

There are also many reviews based work for the task of recommendation. Qu *et al.* [30] propose a bag-of-opinions model to predict a user's numeric rating in a product review. And they develop a constrained ridge regression method for learning scores of opinions. Wang *et al.* [19] propose a review RPS by incorporating the social relations of a reviewer. In addition, they classify the social relations of reviewers into strong social relation and ordinary social relation. Zhang *et al.* [42] incorporate various product review factors including content related to product quality, time of the review, product durability and historically older positive customer reviews. They present a product ranking model that applies weights to product review factors to calculate the ranking score. Ling *et al.* [52] propose a unified model that combines content-based CF, and harnessing the information of both ratings and reviews. Luo *et al.* [43] define and solve a new problem: aspect identification and rating, together with overall rating prediction in unrated reviews. They propose a LDA-style topic model which generates ratable aspects over sentiment and associates modifiers with ratings.

D. Sentiment-Based Applications

Sentiment analysis can be conducted on three different levels: review-level, sentence-level, and phrase-level. Review-level analysis [29], [32], [47], [48] and sentence- level analysis [49] attempt to classify the sentiment of a whole review to one of the predefined sentiment polarities, including positive, negative and sometimes neutral. While phrase-level analysis [59], [53] attempt to extract the sentiment polarity of each feature that a user expresses his/her attitude to the specific feature of a specific product. The main task of phrase-level sentiment analysis is the construction of sentiment lexicon. Pang *et al.* [47] propose a context insensitive evaluative lexical method. However, they can not deal with the mismatch between the base valence of the term and the author's usage. Polanyi and Zaenen [44] describe how the base attitudinal valence of a lexical item is modified by lexical and discourse context and propose a simple implementation for some contextual shifters. They calculate user sentiment based on a finer grained method on all levels. Taboada *et al.* [46] present a semantic orientation calculator which uses dictionaries of words annotated with their semantic orientation (polarity and strength), and incorporates intensification and negation. Lu *et al.* [16] propose an optimization framework that provides a unified and principled way to combine different sources of information for learning a context- dependent sentiment lexicon. The proposed framework is quite general and applicable for opinionated text collection in any domain. Wang *et al.* [36]
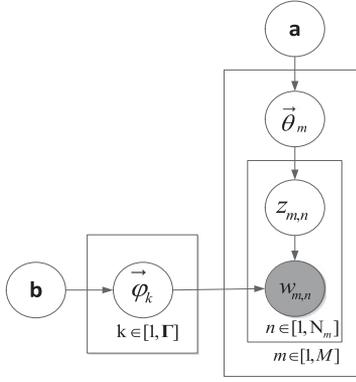
Fig. 2. Graphical model representation of LDA. The borders are representing replicates. The outer border represents user document, while the inner border represents the repeated choice of topics and words within a document.

analyze user opinions about an entity in a review at the level of topical aspects. They discover each individual reviewer's latent opinion on each aspect when forming the overall judgment of the entity.

There are many approaches leveraging sentiment analysis for personalized recommendation [12], [56], [59], [60]. Zhang *et al.* [12] propose a self-supervised and lexicon-based sentiment classification approach to determine sentiment polarity of a review that contains both textual words and emoticons. And they use sentiment for recommendation. Lee and Lee [56] propose a recommender system using the concept of Experts to find both novel and relevant recommendations. By analyzing the user ratings, they can recommend special experts to a target user based on the user population. Lei and Qian [60] leverage phrase-level sentiment analysis to infer a specific item's reputation. They also propose the concept of "virtual friends" to model items' relations, which can reduce time complexity while training. Zhang *et al.* [59] propose an Explicit Factor Model (EFM) to generate an explainable recommendation, they extract explicit product features and user opinions by phrase-level sentiment analysis on reviews.

III. THE PROPOSED APPROACH

The purpose of our approach is to find effective clues from reviews and predict social users' ratings. In this paper, we firstly extract product features from user review corpus, and then we introduce the method of identifying social users' sentiment. In addition, we describe the three sentimental factors. At last we fuse all of them into our sentiment-based RPS. The following sections describe more details about our approach.

A. Extracting Product Features

Product features mainly focus on the discussed issues of a product. In this paper, we extract product features from textual reviews using LDA [11]. We mainly want to get the product features including some named entities and some product/item/service attributes. LDA is a Bayesian model, which is utilized to model relationship of reviews, topics and words. In Fig. 2, the shaded variables indicate the observed variables

and the unshaded variables indicate the latent variables. The arrow indicates a conditional dependency between the variables and plates represented by the box. The definition of terminologies in LDA model is described as:

1) V: the vocabulary, it has N_d unique words. Each word is presented by the corresponding label $\{1, 2, \ldots, N_d \}$;
2) $w_i \in \{1, 2, \ldots, N_d \}$: the word, each word of a review is mapped to V whose size is N_d through character matching;
3) d_m: the document/review of a user, it corresponds to a word set of the review. A user with only one document. All documents denote as $D = \{d_1, d_2, \ldots, d_M \}$;
4) Γ: the number of topics (const scalar);
5) $\vec{\theta}_m$: the multinomial distribution of topics specific to the document m. One proportion for each document, $\Theta = \{\vec{\theta}_m\}_{m=1}^M (M \times \Gamma \text{ matrix})$;
6) $\vec{\varphi}_k$: the component for each topic, $\Phi = \{\vec{\varphi}_k\}_{k=1}^\Gamma (\Gamma \times k \text{ matrix})$
   $z_{m,n}$: the topic associated with the $n - th$ token in the document m; and
7) a, b: Dirrichlet priors to the multinomial distribution $\vec{\theta}_m$ and $\vec{\varphi}_k$.

1) Data Preprocessing for LDA: To construct the vocabulary, we firstly regard each user's review as a collection of words without considering the order. Then we filter out "Stop Words" [34], [41], "Noise Words" and sentiment words, sentiment degree words, and negation words. A stop word can be identified as a word that has the same likelihood of occurring in those documents not relevant to a query as in those documents relevant to the query. For example, the "Stop Words" could be some prepositions, articles, and pronouns etc.. After words filtering, the input text is clear and without much interference for generating topics. All the unique words are constructed in the vocabulary V, each word has a label $w_i \in \{1, 2, \ldots, N_d \}$.

2) The Generative Process of LDA: The input of LDA model is all users' document sets D, and we assign the number of topic Γ (we set 50 empirically). The output is the topic preference distribution for each user and a topic list, which contains at least 10 feature words under each topic. The generative process of LDA consists of three steps, as follows.

1) For each document d_j, we choose a dimensional Dirichlet random variable $\theta_m \sim$ Dirichlet (a).
2) For each topic z_k, where $k \epsilon [1, \Gamma]$, we choose $\phi_k \sim$ Dirichlet (b). For each topic z_k, the inference scheme is based upon the observation that:

$$p\left(\Theta, \Phi | D^{\text{train}}, \alpha, b\right) = \sum_z p\left(\Theta, \Phi | z, D^{\text{train}}, \alpha, b\right)$$
$$\times P\left(z, | D^{\text{train}}, \alpha, b\right). \quad (3)$$

We obtain an approximate posterior on Θ and Φ by using a Gibbs sampler to compute the sum over z.

1) Repeating the process above and eventually we get the output of LDA.

3) Extracting Product Features: From the three steps above, we obtain each user's topic preference distribution and the topic list. From each topic, we have some frequent words. How-

TABLE I
FREQUENT PRODUCT FEATURES OF THE TOP FIVE
TOPICS ON RESTAURANT DATASET OF YELP

Topics	Example of Product Features
Topic 1	prices, price, discount, worth, cash, card, queue, sell, pay, online
Topic 2	service, waiter, assistant, manager, waitress, servers, food, people, review, customer
Topic 3	attitude, kind, feeling, interior, feel, accessories, experience, environment, suit
Topic 4	wait, waiting, seat, location, hours, time, order, attitude, turn, minutes, phone
Topic 5	Seafood, lobster, dishes, shrimp, sauce, grouper, prawns, scallop, jellyfish, escargots, mussels

TABLE II
BRIEF INTRODUCTION OF THE SENTIMENT DICTIONARIES

Dictionaries	Representative Words
SD(8938)	**POS-Words(4379):** attractive, clean, beautiful, comfy, convenient, delicious, delicate, exciting, fresh, happy, homelike, nice, ok, yum . . . **NEG-Words(4605):** annoyed, awful, bad, poor, boring, complain, crowed, dirty, expensive, hostile, sucks, terribly, unfortunate, worse . . .
ND(56)	no, nor, not, never, nobody, nothing, none, neither, few, seldom, hardly, haven't, can't, couldn't, don't, didn't, doesn't, isn't, won't, . . .
SDD(128)	**Level-1 (52):** most, best, greatest, absolutely, extremely, highly, excessively, completely, entirely, 100%, highest, sharply, superb . . . **Level-2 (48):** awfully, better, lot, very, much, over, greatly, super, pretty, unusual . . . **Level-3 (12):** even, more, far, so, further, intensely, rather, relatively, slightly more, insanely, comparative. **Level-4 (9):** a little, a bit, slight, slightly, more or less, relative, some, some what, just. **Level-5 (7):** less, not very, bit, little, merely, passably, insufficiently.

ever, we need to filter the noisy features from the candidate set based on their co-occurrence with adjective words and their frequencies in background corpus. We have given an example of topics (cluster center of a review) and product features in Table I. After we obtained all product features in a review, we add tags (i.e., the symbol "/" before product features) to distinguish other words in reviews. From Table I, we can see that users in each topic care about a different subset of features, and each subset mainly reveals a different kind of product features.

B. User Sentimental Measurement

We extend HowNet Sentiment Dictionary[3] [12] to calculate social user's sentiment on items. In our paper, we merge the positive sentiment words list and positive evaluation words list of HowNet Sentiment Dictionary into one list, and named it as *POS-Words*; also, we merge the negative sentiment words list and negative evaluation words list of HowNet Sentiment Dictionary into one list, and named it as *NEG-Words*. Our sentiment dictionary *(SD)* includes 4379 *POS-Words* and 4605 *NEG-Words*. Besides, we have five different levels in sentiment degree dictionary *(SDD)*, which has 128 words in total. There are 52 words in the *Level-1,* which means the highest degree of sentiment, such as the words "*most*," and "*best*." And 48 words in the *Level-2*, which means higher degree of sentiment, such as the words "*better*," and "*very*." There are 12 words in the *Level-3*, such as the words "*more*," and "*such*." There are nine words in the *Level-4,* such as the words "*a little*," "*a bit*," and "*more or less*." And there are seven words in the *Level-5,* such as the words "*less*," "*bit*," and "*not very*." Also, we built the negation dictionary (*ND*) by collecting frequently-used negative prefix words, such as "*no*," "*hardly*," "*never*," etc. These words are used to reverse the polarity of sentiment words. The representative words and the sizes of all dictionaries are introduced in Table II.

We firstly divide the original review into several clauses by the punctuation mark. Then for each clause, we firstly look up the dictionary *SD* to find the sentiment words before the product features. A positive word is initially assigned with the score $+1.0$, while a negative word is assigned with the score -1.0. Secondly, we find out the sentiment degree words based

[3][Online]. Available: http://www.keenage.com/download/sentiment.rar

on the dictionary *SDD* and take the sentiment degree words into consideration to strengthen sentiment for the found sentiment words. Finally, we check the negative prefix words based on the dictionary *ND* and add a negation check coefficient that has a default value of $+1.0$. If the sentiment word is preceded by an odd number of negative prefix words within the specified zone, we reverse the sentiment polarity, and the coefficient is set to -1.0. Then for a review r that user u posts for the item i, we get the sentiment score as follows:

$$S(r) = \frac{1}{N_c} \sum_{c \in r} \sum_{w \in c} Q \cdot D_w \cdot R_w \qquad (4)$$

where c denotes the clause. N_c denotes the number of clauses. Q denotes the negation check coefficient. D_w is determined by the empirical rule in [63], [64]. When we have a level-1 sentiment degree word before the sentiment word, D_w is set a value of 5.0; when we have a level-2 sentiment degree word before the sentiment word, D_w is set a value of 4.0, etc. There is a one-to-one correlation between D_w and five sentimental degree levels, $D_w = [0.25, 0.5, 2, 4, 5]$. R_w denotes the initial score of the sentiment word w.

However, when we express positive sentiment by saying "*high quality*," but "*high price*" or "*high noise*" represents the negative sentiment. As a result, such direct rule may result in incorrect sentiment estimation. To improve the precision of sentiment mapping, we add two main linguistic rules as:

1) By applying conjunctive rules [16], [45], [50], [54], [58], [67].

"and" rule: Clauses that are connected with "and"-like conjunctives usually express the same sentiment polarity. For example, "*this mug has high quality and nice appearance*" implies that "*high*" for "*quality*" and "*nice*" for "*appearance*" are of the same polarity. Other "and"-like terms include: as well as, likewise.

"but" rule: Clauses that are connected with "but"-like conjunctives usually express the opposite sentiment polarity. For example, "*this mug has high price but nice appearance*" indicates that "*high*" for "*price*" and "*nice*" for "*appearance*" are of the opposite polarity. Other "but"-like terms include: however, nevertheless, though, and etc.

Original Reviews

Mark J.
Los Angeles, CA
4 friends
3 reviews

★★★★★ 10/19/2015

Such a great restaurant. Really friendly server, but not high price. It's tidy and delicate place, which makes really tasty food.

Reviews Analysis Result

Clause 1: Such /great /restaurant.
Clause 2: Really /friendly /server /but/not/high/price.
Clause 3: tidy /and/delicate /place /really /tasty /food.

Fig. 3. Example of review analysis for identifying user's sentiment on Yelp. Product features are denoted in red font, the sentiment words are denoted in green font, the sentiment degree words are denoted in blue font, the conjunction words like "and" and "but" are denoted in blank font, and the negation words are denoted in bright green font.

2) Distinguish between product features and sentiment words

Some features (i.e., noun) like "*noise*," "*mistake*," and "*stink*" are with clear negative sentiment polarity, while "*acclamation*," "*pleasure*," and "*happiness*" are with clear positive sentiment polarity. Here we treat these words as sentiment words and collect them into sentiment dictionary (*SD*). The words like "*acclamation*," "*pleasure*," and "*happiness*" will be collected into *POS-words* of *SD*, the words like "*noise*," "*stink*," and "*mistake*" will be collected into *Neg-words* of *SD*. When deciding the sentiment score of such a phrase (e.g., "*noise*") in a review, we will give an initial score of –1.0, and then we strengthen the sentiment by looking up the *SDD*, and reverse sentiment polarity by looking up the *ND*.

After we obtained the review *r*'s basic sentiment score and improved the sentiment mapping, we normalize the score as

$$E_{u,i} = \frac{10}{1 + e^{-S(r)}} - 5. \qquad (5)$$

Intuitively, we analyse a real user's review in Fig. 3. We can see that the user's original review is divided into three clauses ($N_c = 3$). And each clause only retains the more important words. In clause 1, "*restaurant*" is a product feature, "*great*" is a positive sentiment word ($R_w = 1$), "*such*" is a Level-3 sentiment degree word ($D_w = 2$), so the sentiment score of this clause is: $1 \times 2 = 2$. In clause 2, both of the words "*server*" and "*price*" are product features, "*friendly*" is a positive sentiment word ($R_w = 1$), "*high*" is a negative sentiment word ($R_w = -1$), because "*but*" is a twist conjunction after a positive word, so it has opposite polarity to "*friendly*." "*not*" is a negation word ($Q = -1$), and "*really*" is a Level-2 sentiment degree word ($D_w = 4$), so the sentiment score of this clause is: $1 \times 4 + (-1) \times (-1) = 5$. In clause 3, both of the words "*place*" and "*food*" are product features, "*tidy,*" "*delicate,*" and "*tasty*" are all positive words ($R_w = 1$), "*and*" is a coordinate conjunction to keep the sentiment words same polarity. At the same time, the word "*really*" is a Level-2 sentiment degree word ($D_w = 4$), so the sentiment score of this clause is: $1 + 1 + 1 \times 4 = 6$. To sum up the three clauses, we get user *u*'s sentiment

Symbols	Description	Symbols	Description
U	a set of users	P	a set of items
m	user numbers	n	item numbers
$R_{m \times n}$	the rating matrix expressed by users on items	$\hat{R}_{m \times n}$	the predicted rating matrix by users on items
$U_{m \times k}$	the user Potential Eigen vector	$P_{n \times k}$	the item Potential Eigen vector
k	the dimension of user latent feature and item latent feature	$E_{u,i}$	user u's sentiment on item i
$D(E_v)$	user v's sentiment variance	F_v	the set of user v's real friends
W_i	item i's sentiment distribution	F_i	item i's virtual friends
$S^*_{u,v}$	normalized user v's sentiment influence on user u	$C^*_{u,v}$	normalized user sentiment similarity of user u and user v
$I^*_{i,j}$	normalized item's reputation similarity of item i and item j	Ψ	the objective function of our rating prediction model
$\lambda, \alpha, \beta, \gamma$	the trade-off parameters in the objective function	ℓ, τ	step size and iteration number while training

$S(r) = \frac{1}{3}(2 + 5 + 6) = 4.33$. After normalizing the basic sentiment score, we get the normalized sentiment score $E_{u,i} = \frac{10}{1 + e^{-S(r)}} - 5 \approx 4.87$. Based on this method, we get all users' sentiment.

C. Three Sentimental Factors in Our Approach

This section describes the major components of the proposed approach, and the notations used in the rest of the paper are summarized in Table III. Each sentiment factor is described as follows:

1) User Sentiment Similarity: Generally, user's friends are trustworthy [2], [4], [8], [15], [38], [40], [55]. If a user has similar interest preferences with his/her friends, then he/she may hold similar attitudes towards the item. Based on this view, we firstly get all users' sentiment, and then calculate the sentiment similarity between the user and his/her friends.

On Yelp website, the items have been divided into a few fixed categories. We assume that the items rated by users have *M* categories, accordingly, we divide the users into *M* categories. Then we determine user *u*'s sentimental distribution $\Omega_u = \{E_u^1, E_u^2, \ldots, E_u^M\}$, where E_u^k denotes user *u*'s average sentiment score in *k*th category. After getting all users' sentimental distributions, we can calculate the sentiment similarity between user *u* and his/her friend *v*. We use cosine similarity to measure the relevance of user *u* and user *v*

$$C_{u,v} = \cosine\left(\Omega_u, \Omega_v\right). \qquad (6)$$

In order to fuse user sentiment similarity factor into matrix factorization model, we normalize $C_{u,v}$ as follows:

$$C^*_{u,v} = \frac{C_{u,v}}{\sum_{v \in F_u} C_{u,v}} \qquad (7)$$

where F_u denotes user *u*'s friends, and "∗" is a normalized symbol, and each row is normalized to unity $\sum_v C^*_{u,v} = 1$.

2) Interpersonal Sentiment Influence: When we search the internet for purchasing, we are more concerned with those users who posted five-star reviews or critical reviews. Especially, the critical reviews can reflect the deficiency of a product. In this case, we observe that reviewers' sentiment will influence others, if a reviewer expressed clear like or dislike sentiment, other users will obtain the specific advantages or weaknesses about a product. However, the middle evaluations have little useful information. In our paper, we argue that if a user always has explicit attitude about a product, his/her reviews will has a great reference value to others, and this user has a big influence on others. While a user always has neutral attitude will has a small reference value to others, and this user will has a small influence on others.

Generally, in mathematical statistics, the variance is used to measure the degree of deviation between random variable and its mathematical expectation (average). According to information theory, large variance means the giant information [66]. Therefore, the reviews with more information will have more influence. So we introduce the method of interpersonal sentiment influence by taking advantage of the concept of variance. The definition of variance is described as follows:

$$D\left(E_v\right) = \frac{1}{n}\sum_{i=1}^{n}\left(E_{v,i} - \bar{E}_v\right)^2 \qquad (8)$$

where $E_{v,i}$ denotes user v's sentiment on item i. $\bar{E}_v$ is the average sentiment score of the items user v has reviewed. Then we normalize the sentiment variance of all user u's friends as follows:

$$S_{u,v}^* = \frac{D\left(E_v\right)}{\sum_{v \epsilon F_u} D\left(E_v\right)} \qquad (9)$$

where F_u is the set of user u's friends, $S_{u,v}^*$ denotes the normalized user v's sentiment influence on user u.

3) Item Reputation Similarity: From typical item-based CF algorithms in [22], we know that similar items can help predicting ratings. Thus, it is important for us to find items that have similar features. In our work, we assume item's reputation can indirectly reflect its real ratings. We leverage users' sentiment distribution to infer item's reputation. Based on users' sentiment, we believe that if two items have similar sentiment distribution, then they may have similar reputation, and they will be posted with similar ratings. Based on the idea, we define that the user set is $U = \{u_1, u_{2,\cdots,}u_m\}$, where m is the number of users. After obtaining each item's normalized sentiment score $E_{u,i}$ in (5), we use the sentimental distribution among all users to denote the item i's reputation $W_i = \{E_{u_1,i}, E_{u_2,i}, \ldots, E_{u_m,i}\}$. Then we choose some items as "virtual friends" of the item, that has been rated by the same users. Item's virtual friends can be used to find the relevance between items, and it can reduce the time cost while training. After that, we calculate the reputation similarity between the item i and its virtual friend j. Here we hold that an item's latent feature P_i should be similar to its friends' latent feature with the weight of $I_{i,j}$. Then we use cosine similarity to measure the reputation similarity of item i and item j

as follows:

$$I_{i,j} = cosine\left(W_i, W_j\right). \qquad (10)$$

In order to fuse item reputation factor into matrix factorization model, we normalize $I_{i,j}$ as follows:

$$I_{i,j}^* = \frac{I_{i,j}}{\sum_{j \epsilon F_i} I_{i,j}} \qquad (11)$$

where F_i denotes item i's virtual friends, and $\sum_{j \epsilon F_i} I_{i,j}^* = 1$.

D. Sentiment-Based Recommender Model

After taking the three sentimental factors above into consideration, we have three important constrain terms in our rating prediction model. They are: 1) Normalized user sentiment similarity $C_{u,v}^*$. 2) Normalized interpersonal sentiment influence $S_{u,v}^*$. 3) Normalized item reputation similarity $I_{i,j}^*$. According to the matrix factorization, we fuse the three factors into the objective function as follows:

$$\Psi\left(\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{P}\right)$$

$$= \frac{1}{2}\sum_{u,i}\left(\hat{R}_{u,i} - R_{u,i}\right)^2 + \frac{\lambda}{2}\left(\boldsymbol{U}_F^2 + \boldsymbol{P}_F^2\right)$$

$$+ \frac{\alpha}{2}\sum_{u}\left(\left(U_u - \sum_{v}C_{u,v}^*U_v\right)\left(U_u - \sum_{v}C_{u,v}^*U_v\right)^T\right)$$

$$+ \frac{\beta}{2}\sum_{u}\left(\left(U_u - \sum_{v}S_{u,v}^*U_v\right)\left(U_u - \sum_{v}S_{u,v}^*U_v\right)^T\right)$$

$$+ \frac{\gamma}{2}\sum_{i}\left(\left(P_i - \sum_{j}I_{i,j}^*P_j\right)\left(P_i - \sum_{j}I_{i,j}^*P_j\right)^T\right) \quad (12)$$

where $\hat{R}_{u,i}$ is the predicted rating value according to (1). $R_{u,i}$ is user u's real ratings on item i, and $R_{u,i} \in \boldsymbol{R}_{m \times n}$. $\boldsymbol{U}_{m \times k}$ and $\boldsymbol{P}_{n \times k}$ denote user Potential Eigen vector and item Potential Eigen vector respectively. U_u and P_i are k-dimensional user-specific and item-specific latent feature vectors of user u and item i, and it is the rank of the latent matrices $\boldsymbol{U}_{m \times k}$ and $\boldsymbol{P}_{n \times k}$. They are obtained by the gradient descent method [8]. The first term of (12) denotes the deviation between the actual rating and prediction score, the second item of (12) is a regularization term, which plays a role in case of over-fitting. The idea of user sentiment similarity is enforced by the third term, which says that if two users have similar sentiment, they may have similar latent feature U_u. The factor of interpersonal sentiment influence is enforced by the forth term, which means if a friend of the user has clear like and dislike sentiment, the user may trust him/her more. The idea of item reputation similarity is enforced by the last term, which says that if two items have similar reputation, they may have similar latent feature P_i.

E. Model Training

We get the corresponding matrix factorization model as (12), from which we can obtain user latent profile U_u and item latent profile P_i by optimization. The objective function is minimized by the gradient decent approach. More formally, the gradients

of the objective function with respect to the variables U_u and P_i are shown as (13) and (14) respectively:

$$\frac{\partial \Psi}{\partial U_u} = \sum_i \left(\hat{R}_{u,i} - R_{u,i} \right) P_i + \lambda U_u$$
$$+ \alpha \left(U_u - \sum_{v \in F_u} C^*_{u,v} U_v \right) - \alpha \sum_{u \in F_v} C^*_{v,u}$$
$$\times \left(U_v - \sum_{w \in F_v} C^*_{v,w} U_w \right)$$
$$+ \beta \left(U_u - \sum_{v \in F_u} S^*_{u,v} U_v \right) - \beta \sum_{u \in F_v} S^*_{v,u}$$
$$\times \left(U_v - \sum_{w \in F_v} S^*_{v,w} U_w \right) \tag{13}$$

$$\frac{\partial \Psi}{\partial P_i} = \sum_u \left(\hat{R}_{u,i} - R_{u,i} \right) U_u + \lambda P_i$$
$$+ \gamma \left(P_i - \sum_{j \int F_i} I^*_{i,j} P_j \right) - \gamma \sum_{i \int F_j} I^*_{j,i}$$
$$\times \left(P_j - \sum_{k \int F_j} I^*_{j,k} P_k \right) \tag{14}$$

where F_v denotes user v's friends, similarly, F_i denotes item i's virtual friends. The initial values of U_u and P_i are sampled from the normal distribution with zero mean. The user and item latent feature vectors U_u and P_i are updated based on the previous values to insure the fastest decrease of the objective function at each iteration. We set the step size $\ell = 0.0002$ and the iteration number $\tau = 500$ to insure the decrease of the objective function in training.

IV. EXPERIMENTS

In this section, we conduct a series of experiments to evaluate the performance of our rating prediction model based on user sentiment. We have crawled nearly 60 thousand users' circles of friends and their rated items. We have subsistent social relationships and reviews to organize experiments. Some previous work [8], [9], [60] are all based on Yelp dataset.[4] The dataset contains eight categories: $\#^1$ *Active Life*, $\#^2$ *Beauty&Spa*, $\#^3$ *Home-Service*, $\#^4$ *Hotel&Travel*, $\#^5$ *Nightlife*, $\#^6$ *Restaurants*, $\#^7$ *Shopping*, and $\#^8$ *pets*. In total, there are 28 629 users, 96 974 items, 300 847 ratings, and we have every user's social relation. Each item has been posted by at least one comment/review. In the following experiments, we firstly evaluate our sentiment algorithm, and then investigate how to leverage review sentiment to achieve accurate rating predictions in various conditions.

A. Sentiment Evaluation

We shall note that, the task of phrase-level sentiment lexicon construction is inherently difficult. We need to tradeoff between precision and recall. As a primary step towards using sentiment

[4][Online]. Available:http://smiles.xjtu.edu.cn/Download/Download_yelp.html

TABLE IV
STATISTICS AND EVALUATION RESULTS OF OUR ALGORITHM

Dataset	Scale	Precision of Positive	Precision of Negative	Average Precision
Movie[61]	2,000	863/1000	592/1000	72.7%
SFU [62]	400	184/200	110/200	73.5%
Yelp [8]	66,992	52,474/57,193 (91.75%)	5,895/9,799 (60.16%)	87.1%

lexicon for RPS, we focus on the precision as we will only use the top-10 product features in our framework, primarily to avoid the negative effects of wrong features as much as possible. We expect as the research in sentiment analysis advances, the performance of our framework will further improve as well.

Similar to [16], we evaluate the sentiment by transforming each sentiment value $E_{u,i}$ into a binary value, namely, $E_{u,i} > 0$, a review will be regarded as positive; $E_{u,i} \leq 0$, a review will be regarded as negative. When testing in a labeled positive dataset, $E_{u,i} \leq 0$, this case is misclassification; When testing in a labeled negative dataset, $E_{u,i} > 0$, this case is also the misclassification. We firstly label all five-star Yelp reviews as positive reviews and label all one-star Yelp reviews as negative reviews. In total, we have 57 193 positive reviews and 9799 negative reviews. The statistics and evaluation results of our sentiment algorithm are shown in Table IV. From Table IV, we can see that the average precision on Yelp dataset is 87.1%. The precision on negative review corpus is 60.16%. However, our sentiment algorithm performance well on a larger positive review corpus, the precision is 91.75%. In order to better evaluate our sentiment algorithm, we test our sentiment algorithm on the other two public datasets [61], [62]. Both of the two public datasets have the same number of labeled positive reviews and labeled negative reviews, the average precision is 72.7% and 73.5% respectively. From Table IV, we can also see that our sentiment algorithm performs better on positive review corpus than negative review corpus.

B. Rating Prediction

1) Evaluation Metrics: In each dataset of Yelp, we use 80% of data as the training set and the remaining 20% as the test set. The evaluation metrics we use in our experiments are Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). They are defined as follows:

$$RMSE = \sqrt{\sum_{i \in \Re_{\text{test}}} \left(\hat{R}_{u,i} - R_{u,i} \right)^2 / |\Re_{\text{test}}|} \tag{15}$$

$$MAE = \sum_{i \in \Re_{\text{test}}} \left| \hat{R}_{u,i} - R_{u,i} \right| / |\Re_{\text{test}}| \tag{16}$$

where $R_{u,i}$ is the real rating value of user u to item i, $\hat{R}_{u,i}$ is the predicted rating value. $|\Re_{\text{test}}|$ denotes the number of user-item pairs in the test set.

2) Comparative Algorithms: We conduct a series of experiments to compare our rating prediction model based on user's sentiment (RPS) with the following existing models.

TABLE V
PERFORMANCE COMPARISONS FOR EIGHT CATEGORIES ON YELP (DIMENSIONALITY K = 10)

DATASETS	Basic MF		CircleCon2b		Context MF		PRM		EFM		RPS	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Active Life	1.633 *31.48%*	1.238 *29.25%*	1.477 *24.24%*	1.126 *22.2%*	1.285 *12.92%*	1.002 *12.57%*	1.265 *11.54%*	0.984 *10.98%*	1.215 *7.90%*	0.941 *6.91%*	1.119	0.876
Beauty&Spa	1.813 *29.01%*	1.390 *26.98%*	1.656 *22.28%*	1.272 *20.2%*	1.454 *11.49%*	1.147 *11.51%*	1.431 *10.06%*	1.128 *10.02%*	1.385 *7.08%*	1.086 *6.54%*	1.287	1.015
HomeService	1.981 *26.4%*	1.558 *24.45%*	1.844 *20.93%*	1.454 *19.05%*	1.624 *10.22%*	1.294 *8.89%*	1.611 *9.5%*	1.284 *8.33%*	1.583 *7.89%*	1.273 *7.54%*	1.458	1.177
HotelTravel	1.683 *33.75%*	1.318 *27.85%*	1.539 *27.55%*	1.201 *20.28%*	1.337 *16.6%*	1.055 *9.86%*	1.321 *15.59%*	1.042 *8.73%*	1.267 *11.99%*	1.024 *7.13%*	1.115	0.951
Night Life	1.408 *22.44%*	1.099 *20.47%*	1.311 *16.70%*	1.026 *14.81%*	1.176 *7.14%*	0.93 *6.02%*	1.150 *5.04%*	0.913 *4.27%*	1.134 *2.77%*	0.899 *2.78%*	1.092	0.874
Pets	1.873 *28.19%*	1.440 *25.76%*	1.715 *21.57%*	1.329 *19.56%*	1.499 *10.27%*	1.195 *10.04%*	1.481 *9.18%*	1.181 *9.48%*	1.436 *6.34%*	1.146 *6.72%*	1.344	1.069
Restaurants	1.261 *15.15%*	0.983 *13.22%*	1.202 *10.98%*	0.944 *9.64%*	1.149 *6.88%*	0.909 *6.16%*	1.094 *2.19%*	0.873 *2.29%*	1.113 *3.86%*	0.886 *3.72%*	1.070	0.853
Shopping	1.600 *24.81%*	1.228 *23.37%*	1.479 *18.66%*	1.138 *17.31 %*	1.321 *8.93%*	1.032 *8.82 %*	1.302 *7.60%*	1.016 *7.38 %*	1.278 *5.87%*	0.999 *5.81%*	1.203	0.941
Average	1.657 *26.92%*	1.282 *24.34%*	1.528 *20.75%*	1.186 *18.21%*	1.356 *10.69%*	1.071 *9.43%*	1.332 *9.08%*	1.053 *7.88%*	1.301 *6.92%*	1.032 *6.01%*	1.211	0.970

The percentage numbers in each cell are the relative improvements of RPS over the various baseline models.

1) *Basic MF*: This method is the baseline matrix factorization approach proposed in [1] without consideration of any social factors. We trained the model as (2).
2) *CircleCon*: This method is proposed in [2], which focuses on the factor of interpersonal trust in the social networks and infers the trust circles based on matrix factorization.
3) *Context MF*: This method [3] improves the accuracy of traditional item-based CF in [22], and SoRec in [53]. They take both interpersonal influence and individual preference into consideration.
4) *PRM*: This method is proposed in [8], which considers three social factors, including interpersonal influence, interpersonal interest similarity and personal interest. It is also based on matrix factorization to predict users' ratings.
5) *EFM*: This method is proposed in [59], which builds two characteristic matrixes: user-feature attention matrix and item-feature quality matrix. Each element in the user-feature attention matrix measures to what an extent a user cares about the corresponding product feature. Each element in the item-feature quality matrix measures the quality of an item for the corresponding product feature.
6) *RPS*: It's our sentiment-based RPS. Compared with above-mentioned models (e.g., EFM), we have built three sentimental dictionaries and added two linguistic rules to calculate users' sentiment, and some scalable sentimental applications are proposed. Such as interpersonal sentiment influence, it combines social networks and user sentiment preferences.

3) Performance Comparison: We compare the performance of our method with the existing models on Yelp dataset. In the objective function of RPS, k is the dimension of user and item latent feature vectors. λ is a coefficient for preventing over-fitting, α, β and γ are trade-off parameters. In all our compared algorithms, we keep the same initialization input and the same parameters set up. In RPS, we set $k = 10$, $\lambda = 1$, $\alpha = \beta = \gamma = 5$. Note that whatever these parameters are, it is fair for all comparative algorithms. To implement the comparative methods, we extract different features in the matrix factorization framework, and build the corresponding feature matrixes in EFM [59]. In Table V, we show the total performance evaluation in
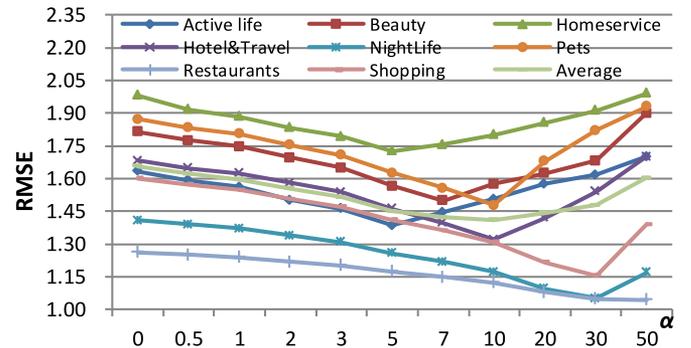

Fig. 4. RMSE line chart of impact of user sentiment similarity factor on eight categories of Yelp.

eight categories of Yelp dataset. The percentage numbers in each cell are the relative improvements of RPS over the various baseline models. It is clearly shown that our RPS model outperforms all the baseline models in each category of Yelp. For the baseline approaches, we decrease RMSE by 26.92%, 20.75%, 10.69%, 9.08%, and 6.92%. We decrease MAE by 24.34%, 18.21%, 9.43%, 7.88%, and 6.01%. The experimental results show the high accuracy of RPS. Meanwhile, we demonstrate the importance of social friend factors (i.e., CircleCon2b, PRM) and explicit features (i.e., EFM) in a recommender system.

C. Discussion

Besides the performance comparison in Table V, we discuss other five aspects in the experiments: the impact of user sentiment similarity, the impact of interpersonal sentiment influence, the impact of user friends' sentimental variance, the impact of item reputation similarity, and the impact of factors combination in all comparative models.

1) The Impact of User Sentiment Similarity: To discuss the impact of user sentiment similarity factor, we set $\beta = 0$, $\gamma = 0$, and let α ranges from 0 to 60. From Fig. 4, the RMSE drops in all testing categories when α ranges from 0 to 5. During this period, user sentiment similarity factor [the third term in (12)] can effectively help the objective function to optimize the user latent feature vectors. It leads to a fast decrease of prediction error [the
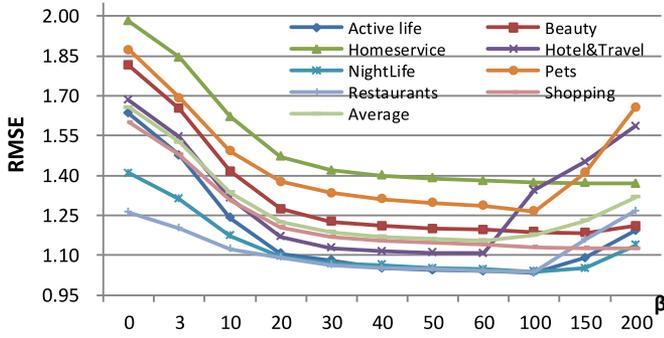
Fig. 5. RMSE line chart of impact of interpersonal sentimental influence factor on eight categories of Yelp.

Fig. 6. RMSE line chart of impact of item reputation similarity factor on eight categories of Yelp.

TABLE VI
IMPACT OF USER FRIENDS' SENTIMENTAL VARIANCE IN ALL COMPARED MODELS (THE EVALUATION METRIC IS RMSE)

MODELS	Basic MF	Circle Con	Context MF	PRM	EFM	RPS
$D(E_v) < 1$	1.645	1.496	1.383	**1.366**	1.442	1.440
$D(E_v) \geq 1$	1.591	1.481	1.305	1.289	1.203	**1.189**

Fig. 7. RMSE line chart of impact of factors combination in all comparative models in shopping dataset of Yelp.

first term in (12)]. However, when α is over 5, the focus of minimizing the objective function moves to the third term instead of the first term. The larger the third term's weight of fitting in training is, the smaller its fitting of the first term will be, which leads to a larger prediction error. This belongs to the problem of over-fitting. The average RMSE is 1.451 when $\alpha = 5$, $\beta = 0$, $\gamma = 0$. Compared with the Basic MF, the average RMSE decreases about 12.43%. The experiment results suggest that user sentiment similarity factor makes a good contribution to the accuracy of rating prediction.

2) The Impact of Interpersonal Sentiment Influence: To discuss the impact of interpersonal sentiment influence, we set $\alpha = 0$, $\gamma = 0$, and let β ranges from 0 to 200. From Fig. 5, the RMSE drops in all testing datasets when β ranges from 0 to 60. Besides, the average RMSE increases in different degrees from $\beta = 60$ to 200 because of over-fitting. The average RMSE is 1.155 when $\alpha = 0$, $\beta = 60$, $\gamma = 0$. Compared with Base MF, the average RMSE decreases about 30.30%.

The experiment results demonstrate interpersonal sentiment influence factor can improve the accuracy of rating prediction.

3) The Impact of User Friends' Sentimental Variance: We discuss the friends' sentimental variance in all comparative models. We set the same parameters as Table V, and we divide the testing users in *shopping* dataset into two parts: The first part consists of the users whose friends with almost neutral sentiment, i.e., $D(E_v) < 1$. The second part consists of the users whose friends with clear like and dislike sentiment, i.e., $D(E_v) \geq 1$.

From Table VI, we can see that RPS model outperforms all baseline models when user friends' sentimental variance $D(E_v) \geq 1$. Besides, our method performs not good when user friends' sentimental variance $D(E_v) < 1$, because our model mainly captures the sentiment influence of the testing users
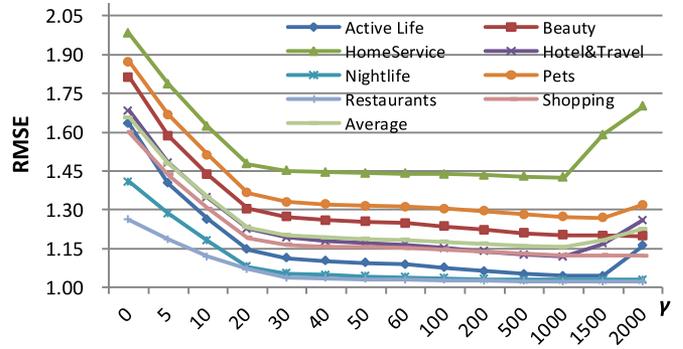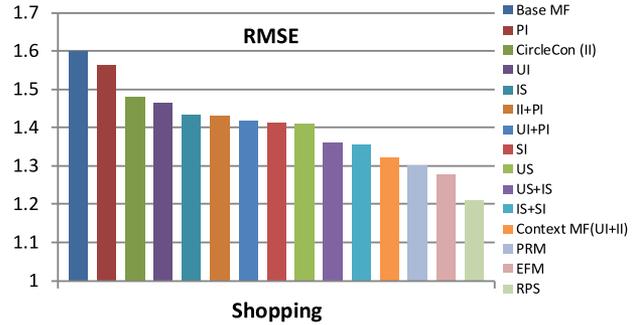
with clear like and dislike sentiment. This experiment shows a large degree of differentiation between the two kinds of users, which shows RPS is very special and effective. Similarly, EFM also considers user sentiment and product features, so we have similar result. Besides, PRM has a better performance when $D(E_v) < 1$, because it considers three important social factors, and it does not distinguish friends' sentiment. Hence, our model may be appropriate for users who have explicit sentiment or those users whose friends have clear like and dislike sentiment.

4) The Impact of Item Reputation Similarity: To discuss the impact of item reputation similarity, we set $\alpha = 0$, $\beta = 0$, and let γ ranges from 0 to 2000. From Fig. 6, we can see that the RMSE drops when γ ranges from 0 to 1000. Besides, the RMSE increases in different degrees from $\gamma = 1000$ to 2000 because of over-fitting. The average RMSE of our model under $\gamma = 1000$ is 1.156. Compared with Basic MF, the average RMSE decreases about 30.2%. The result suggests that the item reputation similarity can improve the performance of rating prediction.

5) The Impact of Factors Combination in All Comparative Models: We compare the performances in *shopping* dataset. In RPS model, we set $\alpha = 5$ and $\beta = \gamma = 0$ for the factor of user sentiment similarity, $\beta = 5$ and $\alpha = \gamma = 0$ for the factor of interpersonal sentiment influence, as well as $\gamma = 5$ and $\alpha = \beta = 0$ for the factor of item reputation similarity. EFM built two characteristic matrixes, PRM model has three social factors, so we set the same parameter for performance comparison, which is shown as Fig. 7.

The approaches using none, one, two, and all of the three factors are systematically compared, and the performance is shown in Fig. 7. *PI, II, UI* denote the personal interest factor, interpersonal influence factor, and user interest similarity in PRM. *US, IS, SI* denote user sentiment similarity factor, item reputation similarity factor and user sentimental influence factor in RPS. "+" denotes the factors combination. We can find that all factors have positive effect on improving performance. Deeply, each factor in our RPS model has a better performance than every social factor in PRM. In addition, the multifactor combination of our model is better than other approaches, such as context-MF, PRM, and EFM.

V. CONCLUSION

In this paper, a recommendation model is proposed by mining sentiment information from social users' reviews. We fuse user sentiment similarity, interpersonal sentiment influence, and item reputation similarity into a unified matrix factorization framework to achieve the rating prediction task. In particular, we use social users' sentiment to denote user preferences. Besides, we build a new relationship named interpersonal sentiment influence between the user and friends, which reflects how users' friends influence users in a sentimental angle. What is more, as long as we obtain user's textual reviews, we can quantitively measure user's sentiment, and we leverage items' sentiment distribution among users to infer item's reputation. The experiment results demonstrate that the three sentimental factors make great contributions to the rating prediction. Also, it shows significant improvements over existing approaches on a real-world dataset. In our future work, we can consider more linguistic rules when analyzing the context, and we can enrich the sentiment dictionaries to apply fine-grained sentiment analysis. Besides, we can adapt or develop other hybrid factorization models such as tensor factorization or deep learning technique to integrate phrase-level sentiment analysis.

REFERENCES

[1] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. NIPS*, 2007, pp. 1257–1264.

[2] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proc. 18th ACM SIGKDD Int. Conf. KDD*, Aug. 2012, pp. 1267–1275.

[3] M. Jiang *et al.*, "Social contextual recommendation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 45–54.

[4] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[5] Z. Fu *et al.*, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Trans. Commun.*, vol. 98, no. 1, pp. 190–200, 2015.

[6] G. Ganu, N. Elhadad, and A Marian, "Beyond the stars: Improving rating predictions using Review text content," in *Proc. 12th Int. Workshop Web Databases*, 2009, pp. 1–6.

[7] J. Xu, X. Zheng, and W. Ding, "Personalized recommendation based on reviews and ratings alleviating the sparsity problem of collaborative filtering," in *Proc. IEEE 9th Int. Conf. e-Bus. Eng.*, Sep. 2012, pp. 9–16.

[8] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, Jul. 2014.

[9] H. Feng and X. Qian, "Recommendation via user's personality and social contextual," in *Proc. 22nd ACM Int. Conf. Inform. Knowl. Manage.*, 2013, pp. 1521–1524.

[10] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Trans. Parallel Distrib. Syst.*, to be published.

[11] D. M. Blei, A.Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J.Mach. Learn. Res.* vol. 3, pp. 993–1022, 2003.

[12] W. Zhang, G. Ding, L. Chen, C. Li, and C. Zhang, "Generating virtual ratings from Chinese reviews to augment online recommendations," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, pp. 1–17, Jan. 2013.

[13] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 340–352, Feb. 2016.

[14] J. Weston, R. J. Weiss, and H. Yee, "Nonlinear latent factorization by embedding multiple user interests," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 65–68.

[15] J. Huang, X. Cheng, J. Guo, H. Shen, and K. Yang, "Social recommendation with interpersonal influence," in *Proc. 19th Eur. Conf. Artif. Intell.*, 2010, pp. 601–606.

[16] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai, "Automatic construction of a context-aware sentiment lexicon: An optimization approach," in *Proc. World Wide Web Conf. Ser.*, 2011, pp. 347–356.

[17] T. Kawashima, T. Ogawa, and M. Haseyama, "A rating prediction method for e-commerce application using ordinal regression based on LDA with multi-modal features," in *Proc. IEEE 2nd Global Conf. Consum. Electron.*, 2013, pp. 260–261.

[18] K.H. L. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme, "Tag-aware recommender systems by fusion of collaborative filtering algorithms," in *Proc. ACM Symp. Appl. Comput.*, 2008, pp. 1995–1999.

[19] B. Wang, Y. Min, Y. Huang, X. Li, and F. Wu, "Review rating prediction based on the content and weighting strong social relation of reviewers," in *Proc. Int. Workshop Mining Unstructured Big Data Using Natural Lang. Process.*, 2013, pp. 23–30.

[20] F. Li *et al.*, "Incorporating reviewer and product information for review rating prediction," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1820–1825.

[21] X. Wang, Y. Zhao, L. Nie, and Y. Gao, "Semantic-based location recommendation with multimodal venue semantics," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp.409–419, Mar. 2015.

[22] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.

[23] Z. Zhao, C. Wang, Y. Wan, Z. Huang, and J. Lai, "Pipeline item-based collaborative filtering based on MapReduce," in *Proc. IEEE 5th Int. Conf. Big Data Cloud Comput.*, Aug. 2015, pp. 9–14.

[24] S. Gao, Z. Yu, L. Shi, X. Yan, and H. Song, "Review expert collaborative recommendation algorithm based on topic relationship," *IEEE/CAA J. Automatica Sinica*, vol. 2, no. 4, pp. 403–411, Oct. 2015.

[25] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, and M. Steyvers, "Learning author-topic models from text corpora," *ACM Trans. Inform. Syst.*, vol. 28, no. 1, pp. 1–30, 2010.

[26] K. K. Fletcher and X. F. Liu, "A collaborative filtering method for personalized preference-based service recommendation," in *Proc. IEEE Int. Conf. Web Serv.*, Jun.-Jul. 2015, pp. 400–407.

[27] Y. Chen, A. Cheng, and W. H. Hsu, "Travel recommendation by mining people attributes and travel group types from community-contributed photos," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1283–1295, Oct. 2013.

[28] Y. Cai *et al.*, "Typicality-based collaborative filtering recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 766–779, Mar. 2014.

[29] Y. Lin, J. Zhang, X. Wang, and A. Zhou, "An information theoretic approach to sentiment polarity classification," in *Proc. 2nd Joint WICOW/AIRWeb Workshop*, 2012, pp. 35–40.

[30] L. Qu, G. Ifrim, and G. Weikum, "The bag-of-opinions method for review rating prediction from sparse text patterns," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 913–921.

[31] A. Daud, "Exploiting explicit semantics-based grouping for author interest finding," in *Proc. 13th Asia-Pacific Web Conf. Web Technol. Appl.*, 2011, pp. 41–52.

[32] S. Li *et al.*, "Sentiment classification and polarity shifting," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 635–643.

[33] Y. Seroussi, F. Bohnert, and I. Zukerman, "Personalized rating prediction for new users using latent factor models," in *Proc. 22nd ACM Conf., Hypertext Hypermedia*, 2011, pp. 47–56.

[34] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, "Author topic model based collaborative filtering for personalized POI recommendation," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 907–918, Jun. 2015.

[35] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Group-Lens: An open architecture for collaborative filtering of net news," in *Proc. ACM Conf. Comput. Supported Cooperative Work*, 1994, pp. 175–186.

[36] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: A rating regression approach," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 783–792.

[37] S. Tan, Y. Li, H. Sun, Z. Guan, and X. Yan, "Interpreting the public sentiment variations on twitter," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1158–1170, May 2014.

[38] B. Sun and V. Ng, "Analyzing sentimental influence of posts on social networks," in *Proc. IEEE 18th Int. Conf. Comput. Supported Cooperative Work Des.*, May 2014, pp. 546–551.

[39] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *J. Internet Technol.*, vol. 16, no. 2, pp. 317–323, 2015.

[40] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang, "Scalable recommendation with social contextual information," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 11, pp. 2789–2802, Nov. 2014

[41] V. Suresh, A. Veilumuthu, and A. Krishnamurthy, "A non-syntactic approach for text sentiment classification with stop words," in *Proc. 20th Int. Conf. Companion World Wide Web*, Mar.–Apr. 2011, pp. 137–138.

[42] K. Zhang, Y. Cheng, W. Liao, and A. Choudhary, "Mining millions of reviews: A technique to rank products based on importance of reviews," in *Proc. 13th Int. Conf. Electron. Commerce*, Aug. 2011, pp. 1–8.

[43] W. Luo, F. Zhuang, X. Cheng, Q. H, and Z. Shi, "Ratable aspects over sentiments: Predicting ratings for unrated reviews," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 380–389.

[44] L. Polanyi and A. Zaenen, "Contextual valence shifters," *Inform. Retrieval*, vol. 20, pp. 1–10, 2006.

[45] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, 2005, pp. 115–124.

[46] M. Taboada, J. Brooke, M. Tofiloski, K. Vol, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[47] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proc. ACL-02 Conf. Empirical Methods Natural Lang. Process.*, 2002, pp. 79–86.

[48] D. Tang, Q. Bing, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification," in *Proc. 53th Annu. Meeting. Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, Jul. 2015, pp. 1014–1023.

[49] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using CRFs with hidden variables," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 786–794.

[50] S. Mohammad, C. Dunne, and B. Dorr, "Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus," in *Proc. Empirical Methods Natural Lang. Process.*, 2009, pp. 599–608.

[51] P. Bedi, H. Kaur, and S. Marwaha, "Trust based recommender system for semantic web," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 2677–2682.

[52] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proc. 8th ACM Conf. Recommender Syst.*, 2014, pp. 105–11.

[53] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc.17th ACM Conf. Inform. Knowl. Manage.*, Napa Vally, CA, USA, 2008, pp.931–940.

[54] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2006, pp. 355–363.

[55] Z. Chen, J. R. Jang, and C. Lee, "A kernel framework for content-based artist recommendation system in music," *IEEE Trans. Multimedia*, vol. 13, no. 6, pp. 1371–1380, Dec. 2011.

[56] K. Lee and K. Lee, "Using dynamically promoted experts for music recommendation," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1201–1210, Aug. 2014.

[57] Z. Wang *et al.*, "Joint social and content recommendation for user-generated videos in online social network," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 698–709, Apr. 2013.

[58] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proc. Int. Conf. Web Search Data Mining*, 2008, pp. 231–240.

[59] Y. Zhang *et al.*, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2014, pp. 83–92.

[60] X. Lei and X. Qian, "Rating prediction via exploring service reputation," in *Proc. IEEE 17th Int. Workshop Multimedia Signal Process.*, Oct. 2015, pp. 1–6.

[61] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics*, 2004, pp. 271–278.

[62] M. Taboada, C. Anthony, and K. Voll, "Methods for creating semantic orientation dictionaries," in *Proc. Conf. Lang. Res. Eval.*, 2006. pp. 427–432.

[63] W. Xiong, Y. Jin, and Z. Liu, "Chinese sentiment analysis using appraiser-degree-negation combinations and PSO," *J. Comput.*, vol. 9, no. 6, pp. 1410–1417, 2014.

[64] H. Nie and Z. Rong, "Review helpfulness prediction research based on review sentiment feature sets," *New Technol. Library Inform. Serv.*, vol. 31, no. 7, pp. 113–121, 2015.

[65] G. Zhao, X. Qian, and X. Xie, "User-service rating prediction by exploring social users' rating behaviors," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 496–506, Mar. 2016.

[66] D. Arnaud and V. Martial, "Influence of emotion on memory for temporal information," *Emotion*, vol. 5, no. 4, pp. 503–507, 2005.

[67] D. Vilares, M. A. Alonso, and C. Gómezrodríguez, "A syntactic approach for opinion mining on Spanish reviews," *Natural Lang. Eng.*, vol. 21, no. 1, pp. 1–25, 2014.

Xiaojiang Lei is currently working toward the M.S.D. degree at the SMILES Laboratory, Xi'an Jiaotong University, Xi'an, China.

His current research interests include social recommendation and data mining.

Xueming Qian (M'09) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, in 2008.

He was a Visiting Scholar with Microsoft Research Asia, Beijing, China, from 2010 to 2011. He was also an Assistant Professor with Xi'an Jiaotong University, where he was an Associate Professor from 2011 to 2014, and is currently a Full Professor. He is also the Director of the SMILES Laboratory at Xi'an Jiaotong University.His research interests include social media big data mining and search.

Prof. Qian was the recipient of the Microsoft Fellowship in 2006 and outstanding doctoral dissertations of Xi'an Jiaotong University and Shaanxi Province in 2010 and 2011, respectively.
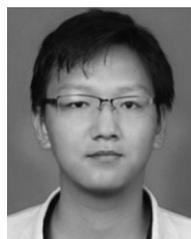
Guoshuai Zhao received the B.E. degree from Heilongjiang University, Harbin, China, in 2012, the M.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2015, and is currently working toward the Ph.D. degree at Xi'an Jiaotong University, Xi'an, China.

His current research interest includes social media big data analysis.