PREDICTION OF RATING BASED ON SOCIAL SENTIMENTS AND REVEIWS OF THE USER

Vrishank N V[1], Rajarajeswari S[2]

[1] *Student , MCA, RRCE, Karnataka, India*
[2] *Assoc Professor, MCA, RRCE, Karnataka, India*

ABSTRACT

We have seen a flourish of review websites which presents a great chance to share our viewpoints for various products we purchase. We are facing the information overloading problem. How to mine valuable information from reviews to understand a user's preferences and make an accurate recommendation [1] is crucial. Traditional recommender systems (RS) considers some factors, such as user's purchase records, product category, and geographic location. In this work, we propose a sentiment-based rating prediction method 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. We also 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 help to improve the recommendation performance.

Keyword – *accurate recommendation,* *Reviews, Rating prediction, User sentiment , Recommender system, Item reputation Sentiment influence,*

1. 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.

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]. 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 DouBan1, Yelp2 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.

1: www.douban.comY [1].

2: www.yelp.com [2].

2. 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

.

2.1. 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 .One of the most well known CF algorithms is the user-based CF algorithm proposed. 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. 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, 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*. propose a review expert collaborative recommendation algorithm based on the assumption that those projects/experts with similar topics have similar feature vectors. Fletcher *et al*. propose a CF-based service recommendation method that considers users' personalized preferences on nonfunctional attributes.

2.2. Matrix Factorization based Approaches

Basic Matrix Factorization

Matrix factorization is one of the most popular approaches for low-dimensional matrix decomposition. Here, we review the Basic MF. 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 Eq. (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.

$\hat{R}_{u,i}=\bar{R}+U_uP_i^T$ (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:

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

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

Social Recommendation

Some matrix factorization based social recommendations are proposed to solve the "cold start" problems. Jamali 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 propose the concept of "Trust Circles" in social networks. Their model outperforms the Basic MF[1] and Social MF. The trusted value between users is represented by a matrix **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 $Su,vc*$ in category c. Except for the factor of interpersonal influence in , Jiang 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 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 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.

Reviews based Applications

There are also many reviews based work for the task of recommendation. Qu 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.* propose a review rating prediction method 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 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 propose a unified model that combines content-based collaborative filtering, and harnessing the information of both ratings and reviews. Luo 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.

Sentiment based Applications

Sentiment analysis can be conducted on three different levels: review-level, sentence-level, and phrase-level. Review-level analysis sentence- level analysis 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 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 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 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 present a semantic orientation calculator which uses dictionaries of words annotated with their semantic orientation (polarity and strength), and incorporates intensification and negation. Lu 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*. 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.

3. 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 rating prediction method (RPS). The following sub-sections describe more details about our approach.

3.1. 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 the 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:

V: the vocabulary, it has Nd unique words. Each word is presented by the corresponding label $\{1, 2, \cdots, Nd\}$.
$\boldsymbol{wi} \in \{1,2, \cdots, Nd\}$: the word, each word of a review is mapped to $\boldsymbol{V}$ whose size is Nd through character matching.
$\boldsymbol{d}m$: 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 $D = \{d1, d2, \cdots, dM\}$.
Γ: the number of topics (const scalar).
$\boldsymbol{\theta}^{\to\to}$ **m**: the multinomial distribution of topics specific to the document m. One proportion for each document, $\Theta = \{\theta\ m\}m{=}1\ M$ ($M \times \Gamma$ matrix)
$\boldsymbol{\varphi}^{\to\to\to}$ **k**: the component for each topic, $\Phi = \{\varphi^{\to}\ k\}k{=}1\ \Gamma$ ($\Gamma \times k$ matrix)
$\boldsymbol{zm}$,$\boldsymbol{n}$: the topic associated with the $n\text{-}th$ token in the document m.
$\boldsymbol{a}$, $\boldsymbol{b}$: Dirrichlet priors to the multinomial distribution $\theta\ m$ and $\varphi^{\to}\ k$.

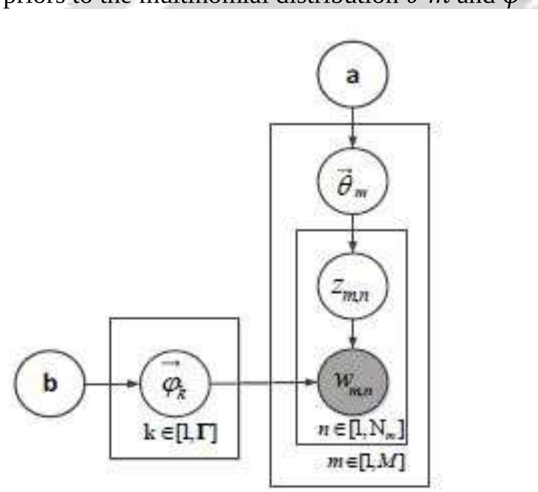

Fig-2:Graphical model representation of LDA. The borders are representing replicates. The outer border represents user document, while the inner border represents the repeated choice of topics and words within a document

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 $wi \in \{1,2, \cdots, Nd\}$.

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:
For each document $\boldsymbol{d}j$, we choose a dimensional Dirichlet random variable θm ~Dirichlet (a).
For each topic zk , where $k\epsilon[1, \Gamma]$, we choose ϕk ~Dirichlet (b). For each topic zk , the inference scheme is based upon the observation that:

$p(\Theta, \Phi|Dtrain , \alpha, b) = \Sigma\ p(\Theta, \Phi|z, Dtrain , \alpha, b)\ z\ P(z, |Dtrain , \alpha, b)$ (3)

We obtain an approximate posterior on Θ and Φ by using a Gibbs sampler to compute the sum over z.
Repeating the process above and eventually we get the output of LDA.

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. However, 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 shown in Table1. 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 Table1 , 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.

User Sentimental Measurement

We extend HowNet Sentiment Dictionary3 [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 9 words in the **Level-4,** such as the words "*a little*", "*a bit*", and "*more or less*". And there are 7 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 2.

TABLE 2.
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.

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 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:

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

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.

"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.

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 sentiment degree dictionary (**SDD**), and reverse sentiment polarity by looking up the negation dictionary (**ND**).

4. EXPERIMENTS

Here, we are conducting 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 works are all based on Yelp dataset4. The dataset contains 8 categories: 1.**Active Life**, 2.**Beauty&Spa**, 3.**HomeService**, 4.**Hotel&Travel**, 5.**Nightlife**, 6.**Restaurants**, 7.**Shopping**, and 8.**pets**. Totally, there are 28,629 users, 96,974 items, 300,847 ratings, and we are having 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.

4.1. Sentiment Evaluation

We shall note that, the task of phrase-level sentiment lexicon construction is inherently difficult. We need to trade off between precision and recall. As a primary step towards using sentiment 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 5-star Yelp reviews as positive reviews and label all 1-star Yelp reviews as negative reviews. In total, we have 57,193 positive reviews and 9,799 negative reviews. The statistics and evaluation results of our sentiment algorithm are shown in Table 4. From Table 4, 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 4, we can also see that our sentiment algorithm performs better on positive review corpus than negative review corpus.

TABLE 4.
THE 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%

4.2. Rating Prediction

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:

$RMSE=\sqrt{}\Sigma(R_{u,i}-R_{u,i})_{2i\in\Re test}|\Re_{test}|/$ (15)

$MAE=\Sigma|R_{u,i}-R_{u,i}|_{i\in\Re test}|\Re_{test}|/$ (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. $|\Re_{test}|$ denotes the number of user-item pairs in the test set.

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.

Basic MF: This method is the baseline matrix factorization approach proposed in [1] without consideration of any social factors. We trained the model as Eq.(2).

Circle Con: 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.

Context MF: This method improves the accuracy of traditional item-based collaborative filtering in , and So Rec . They take both interpersonal influence and individual preference into consideration.

PRM: This method is proposed 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.

EFM: This method is proposed, 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.

RPS: It's our sentiment-based rating prediction method. 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.

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 $\gamma$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, λ=1, α=β=γ=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. In Table 5, we show the total performance evaluation in 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.

5. CONCLUSIONS

Here From this recommendation model we propose a mining sentiment information from social user's reviews. We are combining user sentiment similarity, interpersonal sentiment influence ,and item reputation similarity into a unified matrix factorization framework to achieve the rating prediction task. To be particular we fused social user's sentiment to denote user preferences. Alongside we build a new relationship namely interpersonal sentiment influence between the user and friends, which reflects how user's friends influence users in a sentimental angle. In addition as long as we obtain user's textual reviews, we can quantitatively measure user's sentiment and leverage items' sentiment distribution among users to infer item's reputation. From the experimental results we demonstrate that three sentimental factors make huge contributions to the rating prediction. It also shows significant improvements over existing approaches on a real-world dataset. We further can adopt other hybrid factorization models such tensor factorization or deep learning technique for phrase level sentiment integration.

5. REFERENCES

[1] Z. Zhao, C. Wang, Y. Wan, Z. Huang, J. Lai, "Pipeline item-based collaborative filtering based on MapReduce," *2015 IEEE Fifth International Conference on Big Data and Cloud Computing*, 2015.
[2]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, 2013.
[3] Y. Cai, H. Leung, Q. Li, H. Min, J. Tang, and J. Li, "Typicality-based collaborative filtering recommendation," *IEEE Trans. Knowledge and data engineering*, Vol. 26, No. 3, 2014.
[4] Y. Lin, J. Zhang, X. Wang, A. Zhou, "An information theoretic approach to sentiment polarity classification," in *Proceedings of the 2nd Joint WICOW/AIRWeb Work shop*, Lyon, France. 2012.
[5]L. Qu, G. Ifrim, G. Weikum, "The bag-of-opinions method for review rating prediction from sparse text patterns," in *Proc. 23rd International Conference on Computational Linguistics*, 2010