Indian Journal of Science and Technology, *Vol 10(28), DOI: 10.17485/ijst/2017/v10i28/113441, July 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Sentiment Analysis and Prediction using Text Mining

K. Prasanna Lakshmi, V. Shraddha, V. Abhinava, K. Kavya and R. Gayathri

Department of Information and Technology, GRIET, Hyderabad - 500090, Telangana, India; prasannakompalli@gmail.com, shraddhavasi@gmail.com, abhinavavoruganti@gmail.com, kavyakonda95@gmail.com, rangagayatri@gmail.com

Abstract

Objectives: The main aim of the proposed system is to predict the ratings of a textual review using the concept of sentiment analysis. Prediction is an important process to know about the user sentiment. **Methods/Statistical Analysis**: This work has a sentiment-based rating prediction method (RPS) to upgrade the prediction accuracy in any recommender system. It basically constitutes of a factor used in predicting the rating. Initially, we calculate user's sentiment on an item/product based on user sentimental approach. We apply cosine similarity to find user's sentiment similarity between the users. By taking user's sentiment similarity into consideration we can fill the missing values and predict the rating for the products that have not been reviewed by the users. **Findings:** We assess the above two sentimental factors on a sample dataset collected. Eventually the results show that the sentiment distinguishes user preferences which let a helping hand to enhance the performance of the recommender system. The proposed system is executed and the results show 80% accuracy. This work is efficient in terms of the factors used. **Application/Improvements:** The system can be enhanced with the addition of other factors and fusing them to the recommender system. The use of matrix factorization can, however, be more efficient while using 2 or more factors.

Keywords: Cosine Similarity, Prediction, Reviews, Sentiment Score, User Sentiment

1. Introduction

The online textual reviews play a very important role on decision processes. For example, a customer may tend to buy a product after he or she sees useful reviews, especially a trusted friend. In general, humans believe that reviews and reviewers help in prediction and assume that a high rated product has good reviews. Therefore, in the field of web mining, machine learning and language processing the process of knowing the relationship between reviewers and how to mine reviews has become a very important and basic issue.

Let us focus on the work of predicting the ratings. Nevertheless, on many review websites, the user's rating star-level information is not available. Contrarily, the product information and user opinion information contained in a review has a great reference value for a user's decision. Above all, it is not possible to rate every item for a given user on a website. Consequently, in a user-item-rating matrix, we find many unrated items[1,2]. As review or comment option is available in many sites, it is important for us to influence user reviews to help predicting the items that were unrated.

The rise of review websites gives a vast thought in mining user preferences and predicting users' ratings. Usually, user's interest is stable for a short term, so the user topic can be representative from reviews. For example, considering a category like shoes and boots, different users have a variety of tastes. Some focus on quality, whereas others focus on the price and few may evaluate comprehensively. Regardless, they all have their personalised topics.

Sentiment analysis can be called as the most fundamental and essential work in deriving user's interest preferences. In general terms, the user's own attitude on items is well described through sentiment. Practically, it is important to have numerical scores rather than binary

*Author for correspondence

decisions. We can broadly divide reviews as positive and negative. However, the review given by any user may not be a clear positive or negative sentiment. Customers not only know whether the product is good, but also know how good the product is, to make a purchase decision. We can agree that different users have different sentimental expression preferences. For example, a user can use the word "good" to say that the product was "just so so", while other user may take the word "good" as "excellent"[3].

We often see that the reviews can affect the user in making the decision of buying the product. Usually, if an item's reviews show positive sentiment, we can conclude that the item may be with good reputation. Exceptionally, if the item's reviews are full of negative sentiment then we can say that the item is to be with bad reputation. In the case of purchasing, it is important for us to refer both the positive and negative reviews. The positive reviews gives us the advantages of the product and the negative reviews can give us the shortcomings in case of being cheated. We can notice that the sentiment given by a reviewer can influence others. However, it is hard to predict the user's sentiment and makes a great difficulty in exploring social users. The user reviews can provide us ideas in mining interpersonal inference and user preferences.

We propose a sentiment based prediction method to address these problems. We make use of social users' sentiment to infer ratings in our work. Figure 1 is an example that can illustrate our work. Firstly, we find out the product features from the review and then find out the words used to describe the product. These words are known as the sentiment words. In addition, we leverage the sentiment dictionaries to calculate the sentiment score given by a user on an item/product. We also combine social friend circle with sentiment to recommend. In Figure 1, the last user is interested in those product features, so based on the user reviews and sentiment dictionaries, we recommend the last item[4-8] the main difference is that, we use unstructured data and find the reviews' sentiment score[9-12]. While comparing with the previous works, the proposed approach doesn't just choose either positive or negative sentiment of a review but also focuses on the neutral review.

In our approach the main contributions would be as follows:

- A user sentimental measurement is used which is based on the mined sentiment words and the sentiment degree words from the reviews given by the users.
- The sentiment is used for rating prediction of the review. User sentiment similarity focuses on the user interest preferences. Through this we can know how the user similarity is spread among the friends.

The remainder of this paper is as follows. In Section II, we present the related work, i.e. with the review and sentiment based applications. The proposed approach is present in Section III. It is then followed by the experimentation and discussion in Section IV. Conclusions and future work is drawn in Section V.

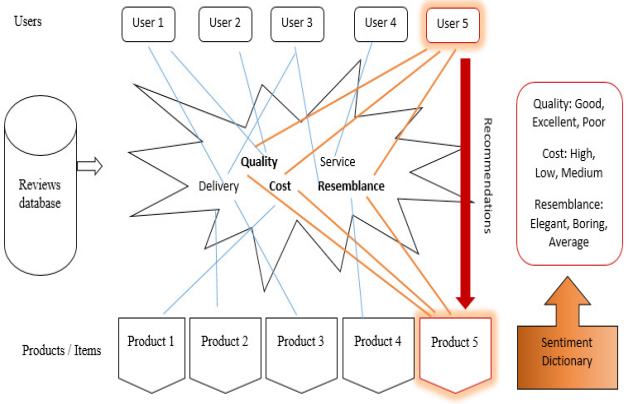

Figure 1. The product features that user cares about are collected in the cloud including the words "Quality", "Cost", and "Resemblance", etc.

1.1 Exertion on Reviews

For the task of recommendation, we have many review based work. And so as to predict a user's numeric rating in a review for a given product, Q_u proposed a bag-of-opinions model[13]. They also developed a constraint regression method for the learning scores of all the opinions given by a user. In[14] proposed a review based RPS by consolidating all the social relations of a reviewer. Moreover, the social relations of the reviewers are divided into strong social relations and ordinary social relations[15]. We consolidate various product review factors including all the content related to quality of the product, time of the review, durability of the product and past historic reviews of the customers. A product ranking model is given that applies weights to all the product review factors so as to calcu-

late the ranking score. A unified model is proposed in[16] that integrates content-based CF and by rendering useful information of both reviews and ratings. In[17] defined and resolved a new complication namely aspect recognition and rating jointly with final rating prediction in unrated reviews. A new LDA-style topic model is introduced which produces ratable features over sentiment and links modifiers with ratings.

1.2 Exertion on Sentiment

We conduct sentiment analysis on three distinct levels namely review-level, sentence-level and phrase-level. In review-based[18,19] and sentence-based analysis[20] we make effort to distinguish the sentiment of a complete review to one of the predefined sentiment polarities comprising of positive, negative and sometimes neutral. In phrase-level analysis[11,21] we make an effort to draw out the sentiment polarity of each feature that a user conveys his/her attitude towards a particular feature of a distinct product. The predominant responsibility of phrase-level sentiment analysis is building of sentiment lexicon. In Pang proposed a context based insensitive evaluative lexical method. They cannot deal with the conflict between base valence of the term and the author's usage. Narrate how the base attitudinal valence of a lexical item is revised by lexical and discourse context and introduce a simple implementation for some contextual shifters. Then calculate a user sentiment on all the levels based on an exquisite grained method[22] presented a semantic orientation calculator which utilizes all the dictionaries of words annotated with their semantic orientation and includes intensification and negation. Later, in[23] introduced an optimized framework that provides an integrated and principled way to merge various sources of information for studying a context-dependent sentiment lexicon. The proposed framework is very simple and appropriate for any opinion oriented collection of data under any domain. We find in[24] examines all the user opinions about an entity in a review at the level of topical aspects. They find each independent reviewer's latent opinion on each and every characteristic for forming the final understanding of the entity. We have various approaches leveraging sentiment analysis for individualized recommendation[25,26]. Self-supervised and lexicon-based sentiment classification approach introduced by[25] used to direct sentiment polarity of a review that holds both emoticons and textual words. And each sentiment is used for recommendation. In[26] introduced a recommender system using the idea of professionals to find both relevant and novel recommendations. By examining all of the ratings given by the user, they can recommend special experts to a target user based on the user community. In[12] leverage phrase-level sentiment analysis to infer reputation of a specific item. This allows us to introduce the concept of "virtual friends" to model reputation of an item which can minimize time complexity while training. In proposed an Explicit Factor Model (EFM) to produce an explainable recommendation; they bring out explicit features of the product and opinions of the users by phrase-level sentiment analysis on reviews.

2. Proposed Approach

The main agenda of our approach is to discover essential hints from reviews and forecast the ratings. We firstly extract product features from the reviews given by users. Then a method of identifying social user's sentiment is inaugurated. Additionally we describe two sentimental factors and ultimately use them into our sentiment-based rating prediction method (RPS). The following sections outline further details about our approach.

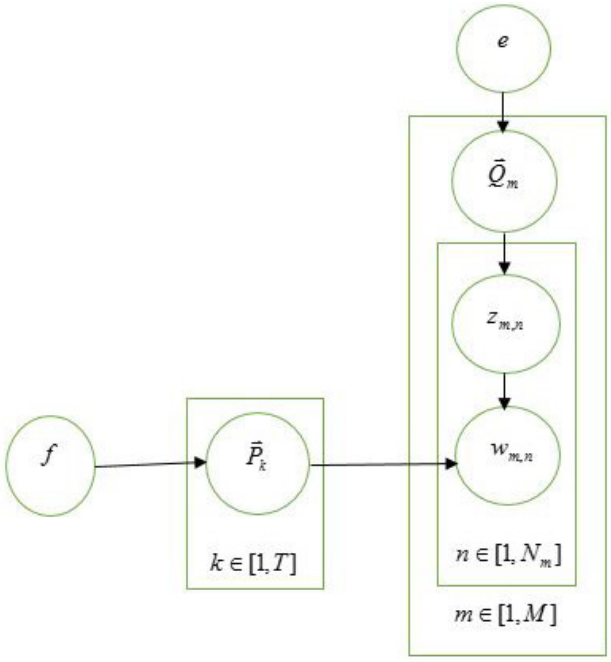

Figure 2. Graphical model representation of LDA using the algorithm.

2.1 Extracting Product Features

Product features are those which are obtained after a focus on the discussed issues of a product. We extract product

features from textual reviews using Latent Dirichlet allocation (LDA)[27] in this paper. We want to get the product features in addition with 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 Figure 2, the observed variables are represented as shaded variables and the variables that are not shaded represent the latent variables. Further definitions of terminologies in LDA model is described as[28]

- **V**: the vocabulary, it has N_d unique words as {1, 2... N_d}.
- $\mathbf{w_i} \in \{1, 2, \ldots, N_d\}$: the word, each word that is to be mapped with **V** and the size is N_d.
- $\mathbf{d_m}$: the document which the user uses. Corresponds to a set of words. All documents denote as $D = \{d_1, d_2, \ldots, d_M\}$.
- T : represents the number of topics.
- $\vec{Q}_m$: Multinomial distribution of topics. One proportion for each document, $\Theta = \{\vec{Q}_m\}_{m=1}^{M} (M \times \Gamma \text{ matrix})$
- $\vec{P}_k$: component for each topic, $\phi = \{\vec{P}_k\}_{k=1}^{\Gamma} (\Gamma \times k \text{ matrix})$
- $\mathbf{Z_{m,n}}$: topic associated with *n-th* token in the document *m*.

e, f: Dirrichlet priors to the multinomial distribution $\vec{Q}_m$ and $\vec{P}_k$.

Begin
- *Store* **reviews** *in an array.*
- *Split the reviews with delimiter as "space" using*
- **split***("space");*
- *Compare strings with the topics in a* **for** *loop.*
- *Select* **for** *every topic.*
- *Select* **for** *every word.*
- *Compare the word using*
- word. equalsIgnore(str);
- *End* **for.**
- *End* **for.**
- *End* **loops.**
- End.

Table 1. A sample of product features of few topics

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

Algorithm for the implementation of LDA algorithm

- Data Preprocessing

Firstly, to build the vocabulary we check the reviews and remove all the stop words[29,30], noise words, sentiment words, sentiment degree words and negative words. For example, stop words can be explained as the prepositions, articles, pronouns etc. After this filtration, the words would be clear and can be stored in the vocabulary**V**, where each word could be labeled as $\mathbf{w_i} \in \{1, 2, \ldots, N_d\}$.

- The Generation Process of LDA

For LDA model, the input is the document sets *D,* and assignsthe number of topic as Γ. We get the output as the topic preference of each user and each topic contains at least 10 words. We consider three steps: [34]

- We choose a dimensional Dirichlet random variable θ_m ~ Dirichlet (a), for each document $\mathbf{d_j}$.
- For each topic z_k, where $k \in [1, \Gamma]$, we choose $\phi_k \sim Dirichlet(b)$. For each topic z_k, the inference scheme is based upon the observation that:

$$p(\theta, \phi \mid D^{train}, a, b) = \sum_z p(\theta, \phi \mid z, D^{train}, a, b) P(z, \mid D^{train}, a, b)$$

- (1)
- Repeating the process will give us the output of LDA.
- Extracting Product Features

From the above process we get each user's topic preference distribution and the topic list. We have few fre-

quent words from each topic. We use Table 1 which has an example of topics and words required. We distinguish words with a '/' between words in the clauses.

2.2 User Sentimental Measurement

We are using HowNet Sentiment Dictionary[31] for calculating a social user's sentiment on the products. In our paper, we consolidate the list of positive sentiment words and positive evaluation words of HowNet Sentiment Dictionary into one list named as POS-words. Similarly, we also consolidate negative sentiment words list and negative evaluation words list of HowNet Sentiment Dictionary into one list named as NEG-words. Now our sentiment dictionary (SD) consists of 4379 POS-words and 4605 NEG-words. Apart from that, we have five distinct levels in sentiment degree dictionary (SDD) which constitutes of 128 words on the whole. There are about 52 words in Level-1 with highest degree of sentiment words such as "best" and "greatest", 48 words in Level-2 with higher degree of sentiment words such as "lot" and "super", 12 words in Level-3 with words such as "even" and "more", 9 words in Level-4 with words such "a little" and "relative" and 7 words in Level-5 with words such as "bit" and "little". Also, we construct a negation dictionary (ND) by gathering frequently used negative prefix words such as "no", "none", "neither" etc. All these words are used to reverse the polarity of sentiment words. The characteristic words and sizes of all the dictionaries are found in the Table 2.

Firstly, the original review is split into several clauses by a punctuation mark. For each clause, firstly we look into dictionary SD to find sentiment words before determining the product features. Initially, a positive word is assigned with the score +1.0 and negative word is assigned with the score -1.0. Secondly we discover sentiment degree words based on the dictionary SDD and take hold of all sentiment degree words to strengthen sentiment for the found sentiment words. In the end, we check negative prefix words based on dictionary ND and also add a negation check coefficient with a default value of +1.0. For suppose the sentiment word is preceded by an odd number of negative prefix words within the confined zone then we reverse the sentiment polarity and also we set the coefficient to -1.0. For an instance, a user 'u' posts a review 'r' for an item 'i' we get the sentiment score as follows:

$$S(r) = \frac{1}{N_c} \sum_{c \in r} \sum_{w \in c} Q \cdot D_w \cdot R_w \qquad (2)$$

Table 2. A sample of the sentiment dictionaries

Dictionaries	Representative words
SD (8938)	**POS-Words(4379):** attractive, clean, beautiful, comfy,convenient, delicious, delicate, exiting, 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…
	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 …
SDD(128)	**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, somewhat, just
	Level-5 (7): less, not, very, bit, little, merely, passably, insufficiently

where c is the clause, N_c is the number of clauses, Q is the negation check coefficient, D_w is determined by the empirical rule[32,33]. The value of D_w is set to 5.0 if we have a level-1 sentiment degree word before the sentiment word. The value of D_w is set to 4.0 if we have a level-2 sentiment degree word before the sentiment word. It is said that there is one-to-one correlation between D_w and all the five sentiment degree levels, D_w= [0.25, 0.5, 2, 4, 5]. R_w is the initial score of sentiment word w.

Although when a positive sentiment is expressed by saying "good quality" but "high price" represents a negative statement. To enhance the accuracy of sentiment mapping, we additionally attach two fundamental linguistic rules such as:

- By Applying Conjunctive Rules[34,34]

"And" rule: When clauses are connected with "and" -like conjunctives normally convey same sentiment polarity. For instance, "This dress has high quality and great appearance" infers that "high" for "quality" and "great" for "appearance" are of same polarity. Few other "and"-like terms are likewise, as well as etc.

"But" rule: When clauses are connected with "but" -like conjunctives normally convey different sentiment

polarity. For instance, "This dress has high price but great appearance" infers that "high" for "price" and "great" for "appearance" are of opposite polarity. Few other "and"-like terms are however, nevertheless, though etc.

- Differentiate between Product Features and Sentiment Words

Figure 3. Example of review analysis for identifying user's sentiment.

Any features of the products (i.e., noun) such as "annoyed", "stink" and "awful" have clear negative sentiment polarity. Besides that we have few other words with clear positive sentiment polarity such as "happiness", "comfy" and "pleasure". Here the words with both positive and negative sentiment polarities are accumulated into a sentiment dictionary (SD). The words with positive sentiment polarity are collected under POS-words category and the words with negative sentiment polarity are collected under NEG-words category of SD. When we decide the sentiment score of a phrase (e.g. "clumsy") in a given review, we initially give the score as -1.0 and then we make the sentiment stronger by looking up the SDD and reverse the sentiment polarity by looking up the ND. Once we have obtained the basic sentiment score of a review 'r', we improve the sentiment mapping and then normalize the sentiment score by the formula given below,

$$E_{u,i} = \frac{10}{1+e^{-S(r)}} - 5 \quad (3)$$

Instinctively, we examine review of a user Figure 3. Here we can say that, 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 grey font, and the negation words are denoted in orange font. We can clearly see that the user's original review is partitioned into three clauses and so $N_c = 3$. And each clause holds only the most essential words. In clause 1, "place" is termed as a product feature, "attractive" is a positive sentiment word ($R_w = 1$), "such" is a Level-3 sentiment degree word ($D_w = 2$) and hence the sentiment score of this clause is $1 \times 2 = 2$. In clause 2, both of the words "staff" and "price" are termed as product features where as "cordial" 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, hence it has opposite polarity to "cordial". Here "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 termed as product features, where as "neat", "beautiful," and "delicious" are all positive words ($R_w = 1$), "and" is a coordinate conjunction so that the sentiment words remain with same polarity. At the same time, the word "really" is a Level-2 sentiment degree word ($D_w = 4$), hence the sentiment score of this clause is $1 + 1 + 1 \times 4 = 6$. Then we add up the sentiment score of all the three clauses, we get user u's sentiment score as $S(r) = 13$ $(2 + 5 + 6) = 4.33$. After the normalization of the basic sentiment score, we get the normalized sentiment score as

$$E_{u,i} = \frac{10}{1+e^{-S(r)}} - 5 \approx 4.87$$

Based on the method, we can get the sentiment score of a given user.

2.3 Sentimental Factors in Our Approach

In this segment, we discuss about the main factors proposed in our system. Each of the terms used are outlined in Table 3.

- User Sentiment Similarity

By considering the view that if a user has similar interest preferences with his/her friends then we can say that he/she may have similar attitude towards that particular item and hence they are termed as trustworthy. Firstly we obtain all user's sentiment, and then compute the sentiment similarity between the user and his/her friends.

The items on our website have been split into a few fixed categories. Let us presume that the items assessed by

the users have M categories. Appropriately, we split all the users into M categories. Later we determine user u's sentimental distribution as $\Omega_u = \{E_u^1, E_u^2, \cdots, E_u^M\}$ where E_u^k denotes user u's average sentiment score in k-th category. Subsequently we fetch all the user's sentimental distributions, and then calculate the sentiment similarity between a user u and his/her friend v. We make use of cosine similarity to estimate the relevance of user u and user v.

$$C_{u,v} = \cos ine(\Omega_u, \Omega_v) \tag{4}$$

With the intention of fusing user sentiment similarity factor into our model, we normalize Cu,v as the following:

[34] (5)

Where F_u denotes user u's friends, and "*" is a normalized symbol, and each row is normalized to unity

$$\sum_v C_{u,v}^* = 1.$$

- Item Sentiment Similarity

When you explore internet for purchasing, we are more anxious about the users who have posted five-star reviews or critical reviews on the items. Predominantly, the critical reviews can reveal the deficiency of a product. Using this we can notice that the reviews given by the users can influence other users. For instance, if a reviewer have conveyed through his/her reviews clear like or dislike sentiment then all the other users will fetch particular benefits and flaws about a product.

However, the middle calculations consist of very beneficial information. In our paper, we hold that a user always possess explicit attitude regarding a particular item and also the reviews given by other users on an item will state as a great reference value to others, so this user will has immense impact on others. While a user always possess neutral attitude will have small reference value to others and this user will have very small influence on others.

We propose a method called item sentiment similarity by considering the advantage of concept of variance. As generally in mathematical statistics, the concept of variance is used to measure the degree of deviation between its random variable and its mathematical expectation. According to information theory, large variance is termed as the giant information. Hence, the reviews with more information will have more influence. The definition of variance is described as below:

$$D(E_v) = \frac{1}{n}\sum_{i=1}^{n}(E_{v,i} - \overline{E}_v)^2 \tag{6}$$

Where $E_{v,i}$ denotes user is the v's sentiment on item i. $\overline{E}_v$ 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 below:

$$S_{u,v}^* = \frac{D(E_v)}{\sum_{v \in F_u} D(E_v)} \tag{7}$$

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.

Table 3. Table of notations in recommender framework

Symbols	Description
U	A set of users
S	A set of items
M	User numbers
N	Item numbers
$E_{u,k}$	User u's sentiment on item i
K	The dimension of user latent feature and item latent feature
$D(E_v)$	User vs. sentiment variance
F_v	The set of user vs. real friends
W_i	Item if's sentiment distribution
$S_{u,v}^*$	Normalized user vs. mutual sentiment influence on user u
$C_{u,v}^*$	Normalized user sentiment similarity of user u and user v

3. Experiment

In this section, evaluating the performance is important and hence we conduct a series of experiments. We've taken a sample of datasets to perform experiments. The dataset includes categories of a sample shopping site and the categories are: Clothing, Footwear, Accessories and Cosmetics. Each item in the category is reviewed or commented at least once. We firstly do our sentiment score calculation.

3.1 Sentiment Score Calculation

We do the sentiment score calculation by using *(3)*. We consider few reviews of a category. For example, a review which is given by a customer on an item from our sample shopping site.

It gives elegant and beautiful look. Extremely magnificent but slightly expensive

Probably, we examine review of a user. We can clearly see that the user's original review is partitioned into two clauses and so $N_c = 2$. And each clause holds only the most essential words. In clause 1, "*elegant*" is a positive sentiment word (R_w =1), "*beautiful*" is a positive sentiment word (R_w =1) "*and*" is a coordinate conjunction so the sentiment words remain with same polarity and hence sentiment score of this clause is *1+1=2*. In clause 2, "*extremely*" is a Level-1 sentiment degree word (D_w =1), "*magnificent*" is a positive sentiment word (R_w =1), "*but*" is a twist conjunction after a positive word. Again, "*slightly*" is a Level-1 sentiment degree word (D_w =1), "*expensive*" is a negative sentiment word (R_w =-1), hence the sentiment score of this clause is *5 x 1 + 3 x -1 =2*. Then we add up the sentiment score of all the two clauses, we get user *u*'s sentiment score as

$$S(r) = \frac{1}{2}(2+2) = 2 .$$

After the normalization of the basic sentiment score we get

$$E_{u,i} = 4.62 .$$

Very cozy but not high price. It is a bright and beautiful dress which makes you look very allureing. Totally worth it.

The user review is divided into Three clauses and so $N_c = 3$. In clause 1, "*very*" is a Level -2 sentiment degree word (D_w =4), "*cost*" is a positive sentiment word (R_w =1) "*but*" is a twist conjunction "*not*" is a negation word (Q =-1), "*high*" is a negative sentiment word (R_w =-1), so sentiment score of this class is $4 \times 1 + (-1) \times (-1) = 5$. Similarly by calculating sentiment score for clause 2 in which "*bright*", "*beautiful*" and "*alluring*" are positive sentiment words (R_w =1) , "*very* " is a Level -2 sentiment degree word (D_w =4) by calculating sentiment score it is $1 + 1 + 4 \times 1 = 6$. In clause 3 "*totally*" is a Level -1 sentiment degree word (D_w =5), "*worth*" is a positive sentiment word (R_w =1), so sentiment score is $5 \times 1 = 5$. Then we add up the sentiment score of all the two clauses, we get user u's sentiment score as

$$S(r) = \frac{1}{3}(5+6+5) = 5.3$$

After the normalization of the basic sentiment score we get $E_{u,i} = 4.82$

It gives a glossy as well as radiant finish which looks very natural on the face. Super smooth on the skin. Little expensive.

The user review is divided into three clauses and so N_c =3.

In clause 1, "*glossy*", "*radiant*", "*natural*" are positive sentiment words (R_w =1), "*very*" is a Level -2 sentiment degree word (D_w =4). So sentiment score is $1 + 1 + 4 \times 1 = 6$. In clause 2, "*super*" is a Level -2 sentiment degree word (D_w =4), "*smooth*" is a positive sentiment word (R_w =1). Sentiment score is $4 \times 1 = 4$. In clause 3, "*little*" is a Level -5 sentiment degree word (D_w =0.25), "*expensive*" is a negative sentiment word (R_w =-1), sentiment score is -0.25. Then we add up the sentiment score of all the two clauses, we get user $u's$ sentiment score as

$$S(r) = \frac{1}{3}(5+9+5) = 6.33$$

After the normalization of the basic sentiment score we get $E_{u,i} = 4.98$.

Immensely beautiful bag. Very convenient to carry and has the best quality. Definitely worth the price.

The user review is divided into two clauses and so N_c = 2.

In clause 1, "*super*" is Level -2 sentiment degree word (D_w =4). "*Comfy*" is a positive sentiment word (R_w =1). "*Awful*" is a negative sentiment word (R_w =-1), sentiment score is $4\times1+(-1)=3$. In clause 2, "*absolutely*" Level-1 sentiment degree word (D_w =5), "*comfortable*" is a positive sentiment word (R_w =1), sentiment score is $5\times1=5$. Then we add up the sentiment score of all the two clauses, we get user *u*'s sentiment score as

$$S(r)=\frac{1}{2}(3+5)=4$$

After the normalization of the basic sentiment score we get $E_{u,i}$ = 4.98.

Immensely beautiful bag. Very convenient to carry and has the best quality. Definitely worth the price.

The user review is divided into three clauses and so N_c = 3. In clause 1, "*immensely*" is a Level -1 sentiment degree word (D_w =5). "*Beautiful*" is a positive sentiment word (R_w =1), so sentiment score is $5\times1=5$. In clause 2, "*very*" is a Level-2 sentiment degree word (D_w =4), "*convenient*" is a positive sentiment word (R_w =1), "*best*" is a Level-1 sentiment degree word (D_w =5), sentiment score is $4\times1+5=9$. In clause 3, "*definitely*" Level -1 sentiment degree word (D_w =5). Sentiment score is 5 x 1 = 5. Then we add up the sentiment score of all the two clauses, we get user *u*'s sentiment score as

$$S(r)=\frac{1}{3}(5+9+5)=6.3$$

After the normalization of the basic sentiment score we get

$E_{u,i}$ = 4.98.

3.2 User Sentiment Similarity Calculation

As soon as we got the sentiment scores for all the reviews given by the users, we calculate the sentiment similarity between the user and his/her friends. Then we deduce sentimental distribution for a user *u* as $\Omega_u=\{3.81,4.95,4.62,4.82,4.98\}$ and for user *v* as $\Omega_v=\{4.52,3.80,-4.70,4.89,4.87\}$. The first step is to measure the relevance of user *u* and user *v* using cosine similarity.

$$C_{u,v}=\cos ine(\Omega_u,\Omega_v)$$

$$\Omega_u\cdot\Omega_v=((3.81\times4.52)+(4.95\times3.80)+(4.62\times(-4.70))+(4.82\times4.89)+(4.98\times4.87))$$

$$=17.2212+18.81+23.5698+24.2526$$

$$=83.7736$$

$$\|\Omega_u\|=\sqrt{3.81^2+4.95^2+4.62^2+4.82^2+4.98^2}$$

$$=\sqrt{14.5161+24.5025+21.3444+23.2324+24.8004}$$

$$=\sqrt{108.3958}$$

$$=10.226$$

$$\|\Omega_v\|=\sqrt{4.52^2+3.80^2+(-4.70)^2+4.89^2+4.87^2}$$

$$=\sqrt{20.4304+14.44+22.09+23.9121+23.7169}$$

$$=\sqrt{104.5894}$$

$$=10.226$$

$$\cos ine(\Omega_u,\Omega_v)=\frac{\Omega_u\cdot\Omega_v}{\|\Omega_u\|\|\Omega_v\|}$$

$$=\frac{83.7736}{10.411\times10.226}$$

$$=\frac{83.7736}{106.4628}$$

$$=0.8150$$

Therefore, we get $\cos ine(\Omega_u,\Omega_v)=0.8150$.

3.3 Calculation of Missing Values

A user may not sometimes give a review for a product. Hence it is important for us to find out the missing values.

$$\Omega_u = (3.81, 4.95, 4.62, 4.82, 4.98)$$
$$\Omega_v = (4.52, 3.8, -4.70, 4.89, 4.87)$$
$$\Omega_w = (3.63, 4.96, 3.72, 2.56, 4.19)$$

We take a sample set of users where we predict a value considering it to be a missing value.

$$\Omega_u = (3.81, 4.95, 4.62, 4.82, 4.98)$$
$$\Omega_v = (4.52, 0, -4.70, 4.89, 4.87)$$
$$\Omega_w = (3.63, 4.96, 3.72, 2.56, 4.19)$$

We calculate cosine similarity among the 3 users. We can see that user *v* has a missing value. It can be found out through few simple steps.

$$\cos ine(\Omega_u, \Omega_v) = 0.6571$$
$$\cos ine(\Omega_v, \Omega_w) = 0.3854$$
$$\cos ine(\Omega_u, \Omega_w) = 0.9798$$

As we are considered with the values of user *v* we take the cosine similarity of *v* with *u* and *w*. We take the average of the values present in the required product category. Here in this example, the value in the second place is missing. We take the average of the values in the second place of user *u*'s and user *w*'s values.

$$\frac{4.95 + 4.96}{2} = 4.955$$

We multiply the average value with the highest value of the cosine similarity between user *u* and *v* and user *v* and user *w*.

$$4.955 \times 0.6571 = 3.25$$

We predicted the missing value to be 3.25 and the original value is 3.8. Hence we have an 80% of accuracy.

4. Conclusion

In this paper, we propose a recommendation model by digging sentiment information from the reviews given by all the social users. We make use of user sentiment similarity and item sentiment similarity to attain the rating prediction task. We utilize social user's sentiment to signify the preferences of the user. Even though we attain textual reviews of the users, we can determine user's sentiment and can hold item's sentiment distribution among all the users to deduce item's similarity. The results of the experiment depict that the user's sentiment similarity contribute for rating prediction. It also reveals enhancement over existing approaches on a real-world dataset.

By taking user's sentiment similarity into consideration we can fill the missing values and predict the rating for the products that have not been reviewed by the users. In future work, we can assess additional rules when studying the context, and we can upgrade the sentiment dictionaries to appeal fine-grained sentiment analysis. We can also use an alternative method called Matrix Factorization to find the missing values in a given matrix. It basically distinguishes both items and users by vectors of factors deduced from item rating patterns. The prime purpose of applying this method into the user item rating matrix is to locate inferior latent factors and to predict the missing values of the matrix.

5. References

1. Salakhutdinov R, Mnih A. Probabilistic matrix factorization. India: Proceeding in NIPS. 2008; p. 1-8.
2. Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. Barcelona, Spain: Proceedings of 4th ACM conference RecSys'10. 2010; p. 135-42. Crossref.
3. Li F, Liu N, Jin H, Zhao K, Yang Q, Zhu X. Incorporating reviewer and product information for review rating prediction. Proceedings of the 22nd International Joint Conference on Artificial Intelligence. 2011; p. 1820-5.
4. Yang X, Steck H, Liu Y. Circle-based recommendation in online social networks. Proceedings of 18th ACM SIGKDD International Conference KDD. 2012; p. 1267-75. Crossref.
5. Jiang M, Cui P, Liu R, Yang Q, Wang F, Zhu W, Yang S. Social contextual recommendation. Proceedings of 21stACM International CIKM. 2012; p. 45-54.
6. Fu Z, Sun X, Liu Q. Achieving Efficient Cloud Search Services: Multi-Keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing. IEICE Transactions on Communications. 2015; 98(1):190-200. Crossref.
7. Qian X, Feng H, Zhao G, Mei T. Personalized recommendation combining user interest and social circle. IEEE Transactions Knowledge and Data Engineering. 2014; p. 1763-77. Crossref.
8. Feng H, Qian X. Recommendation via user's personality and social contextual. Proceedings of 22nd ACM International Conference on Information and Knowledge Management. 2013; p. 1521-4. Crossref.

9. Ganu G, Elhadad N, Marian A. Beyond the stars: Improving rating predictions using Review text content. Proceeding of 12th International Workshop on the Web and Databases. 2009; p. 1-6.
10. Ren Y, Shen J, Wang J, Han J, Lee S. Mutual Verifiable Provable Data Auditing in Public Cloud Storage. Journal of Internet Technology. 2015; 16(2):317-23.
11. Zhang Y, Lai G, Zhang M, Zhang Y, Liu Y, Ma S. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2014; p. 83-92. Crossref.
12. Lei X, Qian X. Rating prediction via exploring service reputation 2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP). 2015; p. 1-6.
13. Qu L, Ifrim G, Weikum G. The bag-of-opinions method for review rating prediction from sparse text patterns. Proceedings of 23rd International Conference on Computational Linguistics. 2010; p. 913-21. PMid: 20473716.
14. Wang B, Min Y, Huang Y, Li X, Wu F. Review rating prediction based on the content and weighting strong social relation of reviewers. Proceedings of the 2013 International Workshop of Mining unstructured big data using natural language processing ACM. 2013; p. 23-30. Crossref.
15. Zhang K, Cheng Y, Liao W, Choudhary A. Mining millions of reviews: a technique to rank products based on importance of reviews. Proceedings of the 13th International Conference on Electronic Commerce. 2011; p. 1-8.
16. Ling G, Lyu RM, King I. Ratings meet reviews, a combined approach to recommend. New York: Proceedings of 8th ACM Conference RecSys'14. 2014; p. 1-8. Crossref.
17. Luo W, Zhuang F, Cheng X, Shi OHZ. Ratable aspects over sentiments: predicting ratings for unrated reviews. IEEE International Conference on Data Mining (ICDM). 2014; p. 380-9. Crossref.
18. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. Proceedings of EMNLP. 2002; p. 79-86.
19. Tang D, Bing Q, Liu T. Learning semantic representations of users and products for document level sentiment classification. Proceedings of 53th Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. 2015; p. 1014-23. Crossref.
20. Nakagawa T, Inui K, Kurohashi S. Dependency tree-based sentiment classification using CRFs with Hidden Variables. 2010; p. 786-94.
21. Ma H, Yang H, Lyu MR, King I. SoRec: Social recommendation using probabilistic matrix factorization. Proceedings of 17th ACM CIKM. 2008; p. 931-40.
22. Taboada M, Brooke J, Tofiloski M, Voll K, Stede M. Lexicon-based methods for sentiment analysis. Computational Linguistics. 2011; 37(2):267-307. Crossref.
23. Lu Y, Castellanos M, Dayal U, Zhai C. Automatic construction of a context-aware sentiment lexicon: an optimization approach. World Wide Web Conference Series. 2011; p. 347-56. Crossref.
24. Wang H, Lu Y, Zhai C. Latent aspect rating analysis on review text data: a rating regression approach. New York: Proceedings of KDD'10. 2010; p. 783-92. Crossref.
25. Zhang W, Ding G, Chen L, Li C, Zhang C. Generating virtual ratings from Chinese reviews to augment online recommendations. ACM TIST. 2013; 4(1):1-17. Crossref, Crossref, Crossref.
26. Lee K. Using dynamically promoted experts for music recommendation. IEEE Transactions on Multimedia. 2014; 16(5):1201-10. Crossref
27. Blei DM, Ng AY, Jordan MI. Latent Dirichlet Allocation. Journal of machine learning research 3. 2003; p. 993-1022.
28. Jiang S, Qian X, Shen J, Fu Y, Mei T. Author topic model based collaborative filtering for personalized POI recommendation. IEEE Transactions Multimedia. 2015; 17(6):907-18. Crossref.
29. Suresh V, Veilumuthu A, Krishnamurthy A. A Non-syntactic approach for text sentiment classification with stop words. Proceeding of ACM WWW. 2011; p. 137-8.
30. Zhang W, Ding G, Chen L, Li C, Zhang C. Generating virtual ratings from Chinese reviews to augment online recommendations. ACM TIST. 2013; 4(1):1-17. Crossref, Crossref, Crossref.
31. Xiong W, Jin Y, Liu Z. Chinese sentiment analysis using appraiser-degree-negation combinations and PSO. Journal of Computers. 2014; 9(6):1-8. Crossref.
32. Nie H, Rong Z. Review helpfulness prediction research based on review sentiment feature sets. New Technology of Library and Information Service. 2015; 31(7):113-21.
33. Kanayama H, Nasukawa T. Fully automatic lexicon expansion for domain-oriented sentiment analysis. Proceedings of EMNLP'06. 2006; p. 355-63.
34. Ding X, Liu B, Yu PS. A holistic lexicon-based approach to opinion mining. Proceedings of WSDM '08. 2008; p. 231-40.