Ratings vs. Reviews in Recommender Systems: A Case study on the Amazon Movies Dataset

Maria Stratigi, Xiaozhou Li, Kostas Stefanidis, and Zheying Zhang

Tampere University, Finland
{maria.stratigi,xiaozhou.li,konstantinos.stefanidis,zheying.zhang}@tuni.fi

Abstract. Together with the prevalence of e-commerce and online shopping, recommender systems have been playing an increasingly important role in people's daily lives in terms of discovering their potential preferences. Therein, users' preferences are mostly reflected by their online behaviors, specially their evaluation towards particular items, e.g., numeric ratings and textual reviews. Many existing recommender systems focus on using item ratings to determine users' preferences, while others provide approaches using textual reviews instead. In this work, via a case study on the Amazon movies data, we compare the recommendation results when using ratings or reviews, as well as that of combining both.

1 Introduction

Recommender systems facilitate the selection of data items by users by issuing recommendations for items they might like. In particular, they aim at providing suggestions to users by estimating their item preferences and recommending those items featuring the maximal predicted preference. Typically, recommendation approaches are classified as content-based and collaborative filtering approaches. In content-based approaches, information about the features/content of the items is processed, and the system recommends items with features similar to items a user likes. In collaborative filtering approaches, we produce interesting suggestions for a user by exploiting the taste of other similar users.

Nowadays, recommendations have more broad applications, beyond products, like news recommendations [9], links recommendations [15, 13], and more innovative ones like query recommendations [3], health recommendations [14], open source software recommendations [7] and diverse venue recommendations [4]. For achieving efficiency, there are approaches that build user models for computing recommendations. For example, [10] applies subspace clustering to organize users into clusters and employs these clusters, instead of a linear scan of the database, for making predictions.

In this study, we investigate the effectiveness of using the sentiment analysis of users textual reviews as the rating of the users for the target items [8]. Comparing to the traditional collaborative filtering approach, when calculating the similarity between users and the relevance of them towards items, we use the ratings obtained based on the sentiment scores of their textual reviews. The

effectiveness of this textual review sentiment-based recommender system is evaluated via a case study on the Amazon movie review dataset. According to the findings, using sentiment score-based rating mechanism can provide more reasonable numeric score for the target items and therefore a more intuitive view of the item quality. In addition, the effectiveness of the recommender system based on sentiment rating is as high as that with regular numeric ratings.

The remainder of the article is organized as follows. Section 2 introduces the recommendation model used in this study. Section 3 presents the method of processing textual reviews towards producing sentiment scores. Section 4 provides the general information regarding the data used in the case study. Section 5 evaluate the recommender system of sentiment score by comparing to that with regular numeric ratings based on the previously introduced dataset. Section 7 concludes the article with a summary of our contributions.

2 The Recommendation Model

Assume a recommender system, where I is the set of data items to be rated and U is the set of users in the system. A user $u \in U$ might rate an item $i \in I$ with a score $p(u, i)$, i.e., $p(u, i)$ reflects the numeric rating (range in $[1, 5]$) u gave directly to i. Comparatively, $p(u, i)$ can also reflect the user's evaluation for an item via a textual review (more in Section 3). Let R be the set of all ratings recorded in the system. Typically, the cardinality of I is high and users rate only a few items. The subset of users that rated an item $i \in I$ is denoted by $U(i)$, whereas the subset of items rated by a user $u \in U$ is denoted by $I(u)$.

For the items unrated by the users, recommender systems estimate a relevance score, denoted as $r(u, i)$, $u \in U$, $i \in I$. There are different ways to estimate the relevance score of an item for a user. In the content-based approach (e.g., [11]), the estimation of the rating of an item is based on the ratings that the user has assigned to similar items, whereas in collaborative filtering systems (e.g., [12]), this rating is predicted using previous ratings of the item by similar users. In this work, we follow the collaborative filtering approach. First, similar users are located via a similarity function that evaluates the proximity between two users. Then, items relevance scores are computed for users taking into account their most similar users. For computing the similarities between user u and u', denoted as $s(u, u')$, we use the Pearson Correlation, that is defined as follows:

$$s(u, u') = \frac{\sum_{i \in X} (p(u, i) - \mu_u)(p(u', i) - \mu_{u'})}{\sqrt{\sum_{i \in X} (p(u, i) - \mu_u)^2} \sqrt{\sum_{i \in X} (p(u', i) - \mu_{u'})^2}} \quad (1)$$

where $X = I(u) \cap I(u')$ and μ_u is the mean of the ratings in $I(u)$, i.e., the mean of the ratings for user u. Given a user u and the group of users that are considered similar to him/her, P_u, if u has expressed no preference for an item

i, the relevance of i for u, denoted as $r(u, i)$, is estimated as:

$$r(u,i) = \frac{\sum_{u' \in (P_u \cap U(i))} s(u,u') p(u',i)}{\sum_{u' \in (P_u \cap U(i))} s(u,u')} \tag{2}$$

After estimating the relevance scores of all unrated user items for a user u, namely A_u, the items A_u^k with the top-k relevance scores are suggested to u.

3 Processing Textual Reviews

3.1 Sentiment Analysis with VADER

To assign a sentiment score to each textual review, we adopt a robust tool for sentiment strength detection on social web data, namely the Valence Aware Dictionary for sEntiment Reasoning (VADER) approach [5]. Compared with other sentiment analysis tools, VADER has a number of advantages. Firstly, the classification accuracy of VADER on sentiment towards positive, negative and neutral classes is even higher than individual human raters in social media. In addition, its overall classification accuracy on product reviews from Amazon, movie reviews, and editorials from NYTimes also outperform other sentiment analysis approaches, such as SenticNet [2], SentiWordNet [1], and Word-Sense Disambiguation [6], and run closely with the accuracy of individual human.

As any text can be seen as a list of words, the approach first selects a lexicon that will determine the sentiment score of each word in the given list. The lexicon for sentiment analysis is a list of words used in English language, each of which is assigned with a sentiment value in terms of its sentiment valence (intensity) and polarity (positive/negative). To determine the sentiment of words, we assign a rational value within a range to a word. For example, if the word "okay" has a positive valence value of 0.9, the word "good" must have a higher positive value, e.g., 1.9, and the word "great" has even higher value, e.g., 3.1. Furthermore, the lexicon set shall include social media terms, such as Western-style emoticons (e.g., :-)), sentiment-related acronyms and initialisms (e.g., LOL, WTF), and commonly used slang with sentiment value (e.g., nah, meh).

With the well-established lexicon, and a selected set of proper grammatical and syntactical heuristics, we to determine the overall sentiment score of a review, which is in the range of $[-1, 1]$ with VADER. The grammatical and syntactical heuristics are seen as the cues to change the sentiment of word sets. Therein, punctuation, capitalization, degree modifier, and contrastive conjunctions are taken into account. For example, the sentiment of "The book is EXTREMELY AWESOME!!!" is stronger than "The book is extremely awesome", which is stronger than "The book is very good.". With both the lexicon value for each word of the review, and the calculation based on the grammatical and syntactical heuristics, we can then assign unique sentiment values to each review.

3.2 Sentiment-based Ratings

In order to analyze the relation between the users rating and their text review sentiments, we use the VADER approach to calculate the corresponding sentiment score for each user review text. Due to the fact that VADER is a lexicon-based method which is more suitable for sentence-level sentiment analysis [5], we firstly divide each review text r_i (i.e., the review of user u to item i) into k sentences $s_{i,1}, s_{i,2}, ...s_{i,k}$ using the tokenize function of NLTK. Then the sentiment score of each sentence (i.e., $sent(s_{i,j})$) is calculated accordingly. Therefore, the overall sentiment of review text r_i from user u (i.e., $sentO(u, r_i)$) is calculated by the mean of the sentiment scores of all its review sentences.

$$sentO(u, r_i) = \frac{\sum_{j=1}^{k} sent(s_{i,j})}{k}, \quad where \quad r_i = \{s_{i,1}, s_{i,2}, ...s_{i,j}...s_{i,k}\} \tag{3}$$

To ease the comparison, we further transform the obtained sentiment score $sentO(s_{i,j})$ into quintiles (i.e., sentiment-based ratings), corresponding to the original numeric 5-star ratings with the following equation.

$$p(u,i) = \begin{cases} 1, & \text{if } sentO(u, r_i) \geq -1 \text{ and } sentO(u, r_i) < -0.6 \\ 2, & \text{if } sentO(u, r_i) \geq -0.6 \text{ and } sentO(u, r_i) < -0,2 \\ 3, & \text{if } sentO(u, r_i) \geq -0.2 \text{ and } sentO(u, r_i) < 0,2 \\ 4, & \text{if } sentO(u, r_i) \geq 0.2 \text{ and } sentO(u, r_i) < 0,6 \\ 5, & \text{if } sentO(u, r_i) \geq 0.6 \end{cases} \tag{4}$$

By doing so, for each textual review given by a user u, we obtain the new ratings based on the sentiment score of the texts, namely, $p(u, i)$.

4 The Amazon Movies Reviews Dataset

In this case study, we use the Amazon movies reviews dataset[1]. The dataset spans a period between August of 1997 and October of 2012. It consists of 889.176 users and 253.059 movies. The users have given 7.911.684 reviews, and their median word count is 101. Each entry in the dataset consists of a user identification number, the movie id that the user reviews, the plain text review and the rating that the user ultimately gave to the product.

To better determine the sentiment score from a user review, we eliminate all entries that the reviews were composed with less than 101 words. After this step the dataset is reduced to 487.134 users, 211.903 movies and 4.086.968 reviews. Also, we eliminate all movies with less than 20 reviews. This resulted in our final dataset that contained 320.451 users, 46.421 movies and 3.301.125 reviews. The distribution of the ratings per user follows a power law distribution. To better demonstrate that, in Figure 1, we show the 600 users with the most ratings

[1] https://snap.stanford.edu/data/web-Movies.html

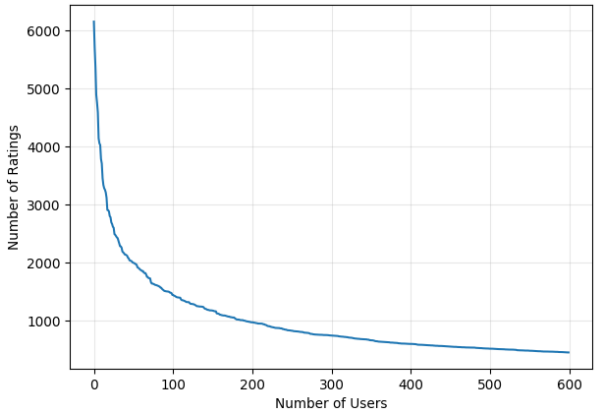

Fig. 1: The distribution of ratings for the 600 users with the most ratings given.

given. All other users belong in the long tail of the power law distribution with less than 1000 ratings per user.

Figure 2(a) presents the distribution of the original numeric ratings, while Figure 2(b) presents that of the sentiment-based ratings. Figure 2(a) shows that the majority of the users give 5 stars to the movies on Amazon.com, while the number of 4-star ratings or below is much lower. Comparatively, from Figure 2(b), we could observe that the majority of the users mean to give 3 or 4 star ratings according to the sentiment of their textual reviews. It is rare that users choose to give completely positive or negative reviews. Figure 3 visualizes the number of user reviews along with their difference between numerical ratings and sentiment scores.

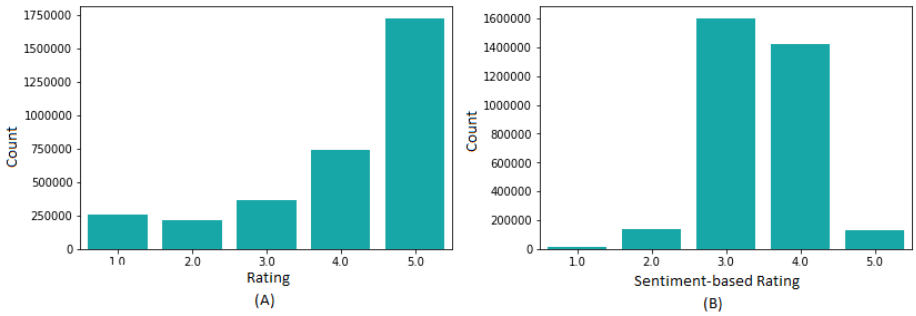

Fig. 2: The Distribution of (a) ratings, and (b) sentiment-based ratings.

Fig. 3: The difference between the ratings and sentiment scores.

5 Evaluation

To calculate the MAE and RMSE errors, we hidden k items from $I(u)$ for 20 different users. Then applied the recommender algorithm twice and tried to predict them. The fist time we utilized the ratings and the second the sentiment scores. Finally, for each different input dataset we average the errors over all users. Figure 4 shows the results for $k = 10, 20, 30, 40, 50$.

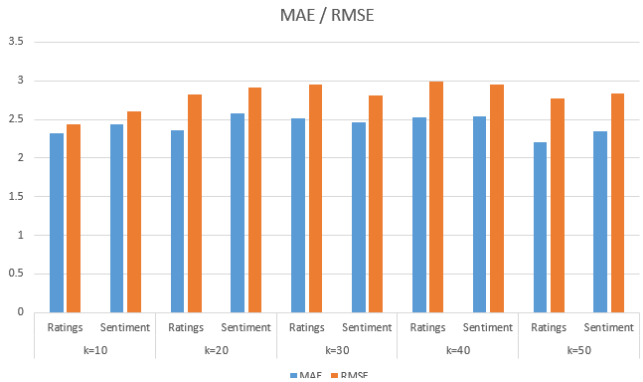

Fig. 4: MAE and RMSE errors for the rating and sentiment scores, calculated for different numbers of hidden items.

To calculate the distance between the recommendation lists produced by the ratings and the sentiment scores, we applied the recommendation algorithm twice on 20 different users. Afterwards, we calculated the distance between these lists for each user and finally we averaged them.

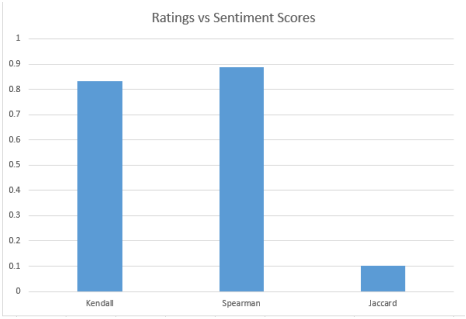

Fig. 5: Kendall, Spearman and Jaccard for the 20 top users. Comparisons between ratings and sentiment scores.

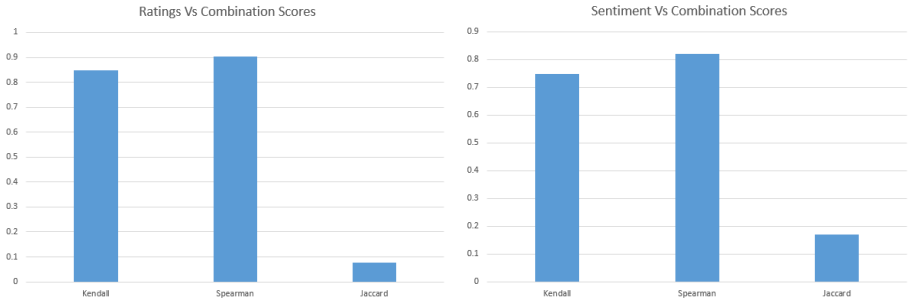

Fig. 6: Kendall, Spearman and Jaccard for the 20 top users. Comparisons between ratings and combination scores (left), and sentiment and combination scores (right).

Figures 5 and 6 shows the calculated distances among the obtained sentiment-based ratings, the combination ratings and the original user-provided ratings. We apply three different ways of distance-based evaluation, including Spearman's footrule distance, Kendall tau distance, and Jaccard distance. From the shown figures, we observe that both Spearman and Kendall distances, for any two recommendation lists, are significant while the Jaccard similarity of such is relevantly low. Such result interestingly demostrates that the recommendation results for a particular user based on the sentiment of his/her textual reviews can be largely different from that based on his/her numeric ratings. Furthermore, averaging the results from both numeric ratings and sentiment-based ratings shall lead to new recommendation lists which are different from either of the results from the two. Without proper evaluation, this study does not indicate any recommendation result is more accurate and preferred than the other two. It instead presents that the preference shown from users' textual reviews towards particular items can be different from the ratings they give to them, which results in largely different recommendation results.

6 Summary

In this study[2], we provide a pilot investigation on the recommendation results using sentiment analysis on users textual reviews. Compared with the recommendation based on users' numeric ratings, as well as that of the combination of both, we find that the similarity among such results are very low. The results of the case study on Amazon users' ratings and reviews for movie items suggests the possibility of users' potential inconsistency in showing preferences with numeric ratings and textual reviews. In our future work, we will evaluate the obtained recommendation results and further investigate the usefulness of applying review sentiment analysis towards recommendation systems.

References

1. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC (2010)
2. Cambria, E., Speer, R., Havasi, C., Hussain, A.: Senticnet: A publicly available semantic resource for opinion mining. In: Commonsense Knowledge (2010)
3. Eirinaki, M., Abraham, S., Polyzotis, N., Shaikh, N.: Querie: Collaborative database exploration. IEEE Trans. Knowl. Data Eng. **26**(7), 1778–1790 (2014)
4. Ge, X., Chrysanthis, P.K., Pelechrinis, K.: MPG: not so random exploration of a city. In: MDM (2016)
5. Hutto, C.J., Gilbert, E.: VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: ICWSM (2014)
6. Júnior, E.A.C., de Andrade Lopes, A., Amancio, D.R.: Word sense disambiguation: A complex network approach. Inf. Sci. **442-443**, 103–113 (2018)
7. Koskela, M., Simola, I., Stefanidis, K.: Open source software recommendations using github. In: TPDL (2018)
8. Li, X., Zhang, Z., Stefanidis, K.: Mobile app evolution analysis based on user reviews. In: SoMeT (2018)
9. Liu, J., Dolan, P., Pedersen, E.R.: Personalized news recommendation based on click behavior. In: IUI (2010)
10. Ntoutsi, E., Stefanidis, K., Rausch, K., Kriegel, H.: Strength Lies in Differences: Diversifying Friends for Recommendations through Subspace Clustering. In: CIKM (2014)
11. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: The Adaptive Web, Methods and Strategies of Web Personalization (2007)
12. Sandvig, J.J., Mobasher, B., Burke, R.D.: A survey of collaborative recommendation and the robustness of model-based algorithms. IEEE Data Eng. Bull. **31**(2), 3–13 (2008)
13. Stefanidis, K., Ntoutsi, E., Kondylakis, H., Velegrakis, Y.: Social-based collaborative filtering. In: Encyclopedia of Social Network Analysis and Mining, 2nd Edition (2018)
14. Stratigi, M., Kondylakis, H., Stefanidis, K.: Fairgrecs: Fair group recommendations by exploiting personal health information. In: DEXA (2018)
15. Yin, Z., Gupta, M., Weninger, T., Han, J.: LINKREC: a unified framework for link recommendation with user attributes and graph structure. In: WWW (2010)

[2] This work has been partially supported by the Virpa D project funded by Business Finland.