Exploiting Reviews to Generate Personalized and Justified Recommendations to Guide Users' Selections

Nevena Dragovic, Maria Soledad Pera
People and Information Research Team
Computer Science Department
Boise State University
Boise, Idaho, 83725

Abstract

We introduce RUS, a recommender that assists users by providing personalized and justified suggestions to facilitate the task of deciding which items, among the recommended ones, are best tailored towards their individual interests. We exploit users' reviews and matrix factorization to generate recommendations that include reviewers' opinions related to item characteristics that each individual user frequently mentions. To demonstrate the validity of RUS we use the Amazon dataset.

1 Introduction

Recommendation systems, which aid users in locating items of interest have been studied for the last few decades. Two of the most common issues that still affect them, which are the focus of this work, are lack of personalization and trust. As an attempt to more adequately tailor recommendations towards each user, researchers take advantage of different available data, such as reviews (Almahairi et al. 2015), to learn user preferences. The pursuit of further personalized strategies, however, continues, to improve both the performance and perceived usability of recommenders. While recent works focus on explaining recommendations to foster trust, justifying the reasons why an item has been suggested to a user is not an easy task. Thanks to the growth of online sites that archive reviews, researchers have suggested leveraging this data source to enhance the recommendation process (Zhang et al. 2014). Nonetheless, a better understanding of the features of a particular item that appeal the most to each individual user (e.g., $price$ in the case of restaurants), which can inform the recommendation justification process, is yet to be accomplished. To address the issues mentioned above, we present $Recommender\ Undeterred\ by\ Sentiment$ (RUS), which showcases suggested items in their real light. In developing RUS, we focus our efforts on using information collected from users' ratings and reviews to generate personalized suggestions with their corresponding explanations. By explicitly incorporating into the recommendation process items' characteristic frequently-mentioned by a user, as inferred from his reviews, we can get to know the user better than by simply considering his rating patterns (Liu, Wang, and Smola 2015). Consequently, the relevance and satisfaction on RUS' recommended items is increased. We strive for the development of a recommender users can trust by providing information they are interested in, no matter if it has a positive or negative connotation. This helps users make suitable decisions faster, in terms of selecting the most adequate item among the recommended ones, since users are spared the burden of performing additional search to find information about item traits they favor. To the best of our knowledge, no recommendation system takes advantage of users' reviews without including the sentiment to learn users' features of interest and generate explanations. RUS is domain independent, which is why its assessment is conducted on multiple item domains within the Amazon dataset.

2 Related Work

Recommendation systems have evolved from traditional content-based and collaborative filtering methodologies to strategies based on matrix factorization (Ricci, Rokach, and Shapira 2011).To improve their performance and ability to satisfy users' needs, different data sources have been leveraged to help better identify user preferences and thus further personalize recommendations. One of the data sources based on user-generated data that sparks the most interest among researchers is users' reviews. While Almahairi et al. (2015) rely on reviews to improve the performance of a prediction rating model, Zhang et al. (2014) analyze the sentiment of feature descriptions to understand how informal language used in reviews can improve rating predictions. Other than personalized suggestions, a powerful way to build a successful relationship between users and recommenders is by providing insights on how each system works and why a given item is recommended (Tintarev 2010). An attempt to do so involves including explanations that justify the generated suggestions. Among strategies used to generate explanations, those based on exploring previous activity of the user, information collected from user reviews, or content-based tag cloud explanations, are the most common. A number of the aforementioned strategies analyze the sentiment, i.e., positive or negative, of feature descriptions (Zhang et al. 2014), as many researchers believe that sentiment-based explanations can be more effective, trustworthy, and persuasive than the ones that ignore polarity (Chen and Wang 2014).

3 Our Proposed Recommender

The design methodology of each step of RUS addresses a research problem on its own, as discussed below.

3.1 Identifying User's Interest on Items

To provide any user U with personalized recommendations we need to identify item features (characteristics), such as the pace of a book or the battery lifetime in case of electronics, that can influence the degree to which U favors an item. We examine reviews written by U and focus on features that U cares the most about. In general, item traits addressed in reviews are expressed as nouns. Thus, we determine the part-of-speech of each word in U's reviews and create a list L of all identified nouns along with their frequencies of occurrence. We use Morpha Stemmer to lemmatize nouns in L, identify variant forms of each noun, and group them together (along with their frequencies). As different nouns can be used to express similar traits, we create clusters of words that refer to the same feature, which allow us to better capture the most frequent features U discusses in his reviews. We sort nouns in L according to their frequencies and treat the highest-ranked noun w_i as the seed that initiates the creation of cluster CU_i. Once created, nouns in CU_i are removed from L. This process is repeated until L is empty.

$$CU_i = \{x \in L \mid sim(w_i, x) =\geq 0.8\} \quad (1)$$

where x is a noun, sim (Equation 2) is captures the degree of similarity between x and w_i, and 0.8 is a threshold. This high threshold, which ensures that terms with similar meanings are clustered together, is based on our analysis of the degree of resemblance between synonym pairs in WordNet.

$$sim\,(w_1, w_2) = \frac{2 \times depth\,(LCS)}{(depth\,(s_{w1}) + depth\,(s_{w2}))} \quad (2)$$

where s_{w_1} and s_{w2} are the depths of the synsets associated with any two words w_1 and w_2 as defined in WordNet, $depth$ is the length of the hypernym path from a given synset to the root, and LCS represents the Least Common Subsumer (common ancestor deepest in the taxonomy). Each cluster CU_i is associated with its overall frequency of occurrence: $AF_{CU_i} = \frac{\sum_{x \in U_i} freq(x)}{|RU|}$, where RU is the set of U's reviews, $|\ RU\ |$ is size of RU, x is defined as in Equation 1, and $freq(x)$ is the frequency of occurrence of x in RU.

Clusters are sorted based on AF_{CU_i}, and the top-k are treated as the ones that best capture U's preferences. For setting k, we conducted preliminary experiments using the Office Products domain from the Amazon dataset (see Section 4). To quantify effectiveness, we used Normalized Discounted Cumulative Gain ($NDCG$) and Mean Reciprocal Rank (MRR), whereas for efficiency, we considered processing time. Even though $NDCG$ and MRR values shown in Table 1are not significant (p $<$ 0.05), processing time rapidly increases as k is increased. Consequently, we set k=2. Since the number of terms in a cluster can be large, we create L_{CU_i}, a label that captures the content of CU_i. Using WordNet, we generate a list of synonyms for each word in CU_i and the word that is part of all the lists is treated as the label for CU_i. (If there is no common word among the synonym lists, then L_{CU_i} is the most frequent term in CU_i.)

Table 1: RUS' performance on Office Products.

k Value	NDCG	MRR	Time (In Seconds)
2	**0.7450**	**0.5896**	384
4	0.7410	0.5818	505
6	0.7437	0.5717	650

3.2 Generating Candidate Recommendations

RUS depends upon the LensKit implementation of $FunkSVD$ algorithm for candidate item generation, i.e., a manageable set of items that likely appeal to U's interests and preferences. This estimation strategy represents each item i and user U as n-dimensional vectors of the form q_i and $p_u \in R^n$, where the vector components of q_i present the degree to which each factor applies to the corresponding item i and the vector components of p_u shows the degree of interest of u on items. We choose this as an adequate approach, since matrix factorization methods provide greater prediction accuracy and memory efficient compact models compared to other strategies (Ricci, Rokach, and Shapira 2011). We only consider candidate items for U those that have a predicted score above 3. In doing so, we eliminate items that are known to be less appealing to the user.

3.3 Identifying Most-Discussed Item Features

To have a deeper understanding of the characteristics that are often used to describe items, we examine features most-commonly addressed on public reviewspertaining to each candidate item I. This is done following the process defined in Section 3.1 for identifying features of interest to U.

3.4 Generating Top-n Recommendations

To determine if I is highly likely of interest for U, we calculate the degree of similarity between U's feature preferences and I's most-discussed features, as shown in Equation 3. We considered two traditional approaches for capturing this similarity, i.e., average and complete linkage, given that we represent U and I as two clusters, as defined in Sections 3.1 and 3.3. To select the optimal strategy, we conducted experiments using the Office Products domain of the Amazon dataset and concluded that complete linkage leads to better performance (for $p < 0.001$), as seen on Table 2.

Table 2: Performance of RUS on Office Products domain

	NDCG	MRR
Average Linkage	0.704	0.583
Complete Linkage	**0.745**	**0.589**

Using Equation 3, we generate a ranking score for each candidate item I, $Rank(I)$, which captures the degree to which U's preferred features are addressed in I's reviews.

$$Rank(I) \equiv \max_{\forall i,j=1..k} \left(sim\left(L_{CU_i}, L_{CI_j}\right)\right) \quad (3)$$

where L_{CU_i} and sim are defined as in Equation 2, L_{CU_j} is defined as in Section 3.3, k = 2 and, max is the function

that captures the maximum similarity scores between labels describing U's preferences and I's most-discussed features.

RUS treats the top-n candidate items with the highest rank scores, as the most relevant items to be suggested to U.

3.5 Generating Explanations

We pair each item to be recommended with an explanation that allows U to choose a single item among the suggested ones. To create informative explanations, we consider the features of interest to U and provide information related to these traits. Using Equation 4, we compute a similarity score for each review sentence S_i in reviews archived for I with respect to a feature of interest for U represented by the corresponding label L_{CU_i}.

$$RelScore(S_i, L_{CU_i}) = \max_{\forall s \in S_i} (sim(s, L_{CU_i})) \quad (4)$$

where s is a noun in S_i, sim is defined as in Equation 2, and max is the function that selects the highest degree of relatedness computed between the nouns in S_i and L_{CU_i}.

For each L_{CU_i}, RUS selects as a part of I's explanations the 3 sentences with the highest $RelScore$, providing U with sufficient information about the recommendations. As previously stated, we do not emphasize the sentiment of the features, since our intent is not to make U like one option more than another, but save U's time in identifying information important for him. RUS provides unbiased explanations solely based on users' features of preferences without involving sentiment. This leads to increasing trust, since users know RUS is making decisions tailored towards individual users based on their ratings and reviews.

4 Experimental Results

Offline Assessment. We conducted experiments using the Amazon dataset (McAuley and Leskovec 2013) (see Table 3). As shown in Figure 1 and Table 4, besides yielding high

Table 3: Statistics about the Amazon Dataset

Domain	# of users	# of items	# stars reviews
Baby	364	5176	4569
Kitchen	5405	49383	101379
Electronics	4195	55844	122380
Instruments	144	8331	1640
Patio	378	12566	6283
Software	148	7406	1971

$NDCG$ and MRR scores, RUS outperforms ($p < 0.01$) $Lenskit$' matrix mactorization (SVD) baseline, demonstrating that in general RUS's recommendations are preferred (and ranked higher) over the ones provided by SVD, which do not explicitly consider users' feature preferences.

We conducted another experiment to compare RUS's performance with state-of-the-art approaches detailed in (Mirbakhsh and Ling 2015). $Single$-MF is based on a single-domain matrix factorization strategy applied to unobserved ratings. $Cross$-MF and $Cross$-$CBMF$ are two cross-domain models for rating prediction. While the former enhances Single-MF by training the prediction model with ratings for multiple domains, the latter aggregates rating information from user-item connections and item clusters. Using the framework defined in (Mirbakhsh and Ling 2015), we evaluated arbitrarily-selected domains in the Amazon dataset based on $Recall = \frac{\#hits}{|T|}$, where #hits is the number of relevant items included by a recommender in a list of $top-N$ recommendation, $|T|$ is the number of items with a "5" rating score in a test set of items for a user, and N(=20) is the number of recommendation examined.

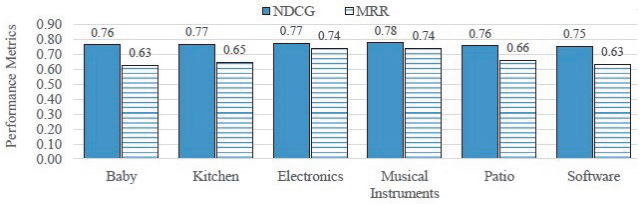

Figure 1: RUS's performance on the Amazon dataset

Table 4: RUS versus baseline

Domain	RUS	SVD	Domain	RUS	SVD
Baby	0.76	0.69	Instruments	0.80	0.62
Electronics	0.77	0.64	Patio	0.76	0.74
Kitchen	0.77	0.68	Software	0.75	0.71

As shown in Table 5, RUS outperforms the strategies considered ($p < 0.005$ and $p < 0.01$ indicated with + and *, respectively). These results are promising, especially given that RUS successfully makes recommendations by incorporating reviews, as opposed to cross-domain information.

Online Assessment of Explanations. As described in (Tintarev 2010), there are seven criteria to be achieved by explanations provided by recommenders: Transparency, Scrutability, Trust, Effectiveness, Persuasiveness, Efficiency and Satisfaction. RUS achieves five out of these seven criteria. By suggesting highly-rated items which are described based on users' frequently-mentioned features and showcasing opinions of other users on those features, RUS addresses transparency. RUS inspires trust, since it does not consider the sentiment connotation of the features to determine if they should be included in the explanations. Instead, RUS provides unbiased recommendations and explanations. With that, users' confidence increases knowing that RUS offers a real picture of each suggested item. Users can also make quick item selections, since they know what items are of their preferences based on provided explanations. Given that users' satisfaction with a recommender is related to the perceived quality of its recommendations and explanations (Gedikli, Jannach, and Ge 2014), we believe RUS users appreciate that they do not need to spend more time on researching items with traits important to them.

As reported in (Tintarev 2010) and based on examination of existing strategies for explaining recommendations (Hernando et al. 2013; Symeonidis, Krinis, and Manolopoulos 2013; Vig, Sen, and Riedl 2009; Zhang et al. 2014), we observe that, on average, only two (out of the seven) criteria

Table 5: Recall-based evaluation: RUS vs. state-of-the-art

	Single-MF	Cross-MF	Cross-CBMF	RUS
Electronics	0.22^{+}	0.21^{+}	0.29^{*}	0.49
Kitchen	0.15^{*}	0.15^{+}	0.18^{*}	0.29

are satisfied by any of these recommenders. The only system that is comparable with RUS is the one introduced in (Zhang et al. 2014) that fulfilled five of the aforementioned criteria. As this system explicitly considers sentiment, we argue that users' trust RUS more than this counterpart.

To quantify the effect personalized explanations have on the recommendation process, in terms of helping users to choose the right item among the recommended ones, we followed the framework presented in (Tintarev 2010).We provided a set of independent appraisers a survey including three different types of explanations, each with a different level of personalization, generated for a set of items.The three types of explanation-generation strategies considered for assessment purposes are: *Baseline (Type 1)* are not personalized and do not provide any descriptions of item features. For example: "The item is one of the top 15% purchased among Office Products". *Non-personalized (Type 2)* show an average rating value for each corresponding item. For example "The average rating of the item is 4.4". *Personalized (Type 3)* refer to strategies that consider individual users' interests. In this case, we refer to recommendation justifications generated by RUS. The survey included these questions: (1) Which type of explanation do you prefer?, (2) Which type of explanation is the most useful?, (3) Which type of explanation helped you make the fastest choice?, and (4) Which type of explanation is the most reliable?.

We surveyed 25 appraisers, who were shown ten different sets of recommended items. As illustrated in Figure 2, 80% of the appraisers preferred RUS explanations over remaining two strategies. Appraisers also indicated that RUS' explanations (i.e., Type 3) provide the most useful information about items. Furthermore, 60% of appraisers found RUS' explanations to be the most reliable. The results reported in this section provide supporting evidence to our claims which indicate that RUS increases satisfaction, effectiveness and trust. Based on the answers collected for question 3, we see that Non-personalized explanations were preferred by a significant majority. However, we believe the reason for this lies in their shorter that those provided by RUS.

5 Conclusion and Future Work

We introduced a new domain-independent recommendation strategy that relies on users' ratings and reviews to create personalized suggestions. The novelty of RUS consists in involving user-generated reviews into its recommendation process but without considering the sentiment expressed in the reviews. We also discussed a novel explanation-generation process that analyses other users' opinions on features of interest for a given user and pairs them with each suggested item. We conducted initial experiments to demonstrate the importance of considering data sources beyond users' ratings to enhance the recommendation process and verify the correctness and usefulness of our explanation-generation strategy. For future work, we plan to conduct user studies to further demonstrate that RUS generates relevant suggestions and helps users in making appropriate choices.

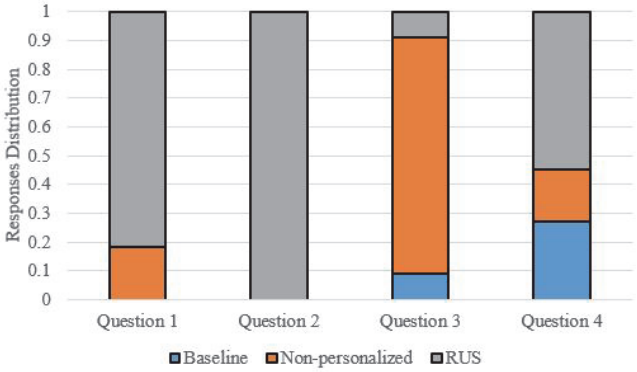

Figure 2: Evaluation of explanation strategies.

References

Almahairi, A.; Kastner, K.; Cho, K.; and Courville, A. 2015. Learning distributed representations from reviews for collaborative filtering. In *RecSys*, 147–154. ACM.

Chen, L., and Wang, F. 2014. Sentiment-enhanced explanation of product recommendations. In *WWW*, 239–240. ACM.

Gedikli, F.; Jannach, D.; and Ge, M. 2014. How should i explain? a comparison of different explanation types for recommender systems. *IJHCS* 72(4):367–382.

Hernando, A.; Bobadilla, J.; Ortega, F.; and Gutiérrez, A. 2013. Trees for explaining recommendations made through collaborative filtering. *Information Sciences* 239:1–17.

Liu, Z.; Wang, Y.-X.; and Smola, A. 2015. Fast differentially private matrix factorization. In *RecSys*, 171–178. ACM.

McAuley, J., and Leskovec, J. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*, 165–172. ACM.

Mirbakhsh, N., and Ling, C. X. 2015. Improving top-n recommendation for cold-start users via cross-domain information. *ACM TKDD* 9(4):33.

Ricci, F.; Rokach, L.; and Shapira, B. 2011. *Introduction to recommender systems handbook*. Springer.

Symeonidis, P.; Krinis, A.; and Manolopoulos, Y. 2013. Geosocialrec: Explaining recommendations in location-based social networks. In *ADBIS*, 84–97. Springer.

Tintarev, N. 2010. Explaining recommendations. PhD Dissertation. University of Aberdeen.

Vig, J.; Sen, S.; and Riedl, J. 2009. Tagsplanations: explaining recommendations using tags. In *IUI*, 47–56. ACM.

Zhang, Y.; Lai, G.; Zhang, M.; Zhang, Y.; Liu, Y.; and Ma, S. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*, 83–92. ACM.