12-10-2023

Improve the Serendipity in Recommender Systems

Amarajyothi Aramanda
University of Hyderabad, amarajyothi.aramanda@gmail.com

M. Kumara Swamy
CMR Engineering College Kandlakoya, Hyderabad, m.kumaraswamy@cmrec.ac.in

Improve the Serendipity in Recommender Systems

Completed Research Paper

Amarajyothi Aramanda
Department of CSE (AI & ML)
CMR Engineering College
Kandlakoya, Hyderabad 501 401
Telangana, India. &
SCIS, University of Hyderabad
Hyderabad 500 046.
amarajyothi.aramanda@gmail.com

M. Kumara Swamy
Department of CSE (AI & ML)
CMR Engineering College
Kandlakoya, Hyderabad 501 401
Telangana, India.
m.kumaraswamy@cmrec.ac.in

Abstract

Recommender system (RS) analyzes the purchase behavior of existing users and predicts relevant item(s) to a new user. In RS, collaborative filtering (CF) is a popular approach to suggest item(s) that are most similar to the new user's interests using item/user similarity. Generally, in CF-RS top rated items are recommended. However, in this approach the low and average rated items are neglected which may be liked by the user. As a result, the CF-RS approach is unable to improve the user satisfaction. In this paper, we propose an approach called *Serendipitous Recommender System* (SRS) to recommend the items which are liked by the users; however, the items need not be top-rated. Generally, the user ratings may not express the true opinions of the users. We observe that the users express their opinions in user-reviews through the emotional words and they may contain surprise emotion. These emotional words are considered to update the user-rating such that the items will be available for recommendations. We use the updated user-ratings for the final recommendations using the user-based and item-based CF approaches. We call such recommended items as serendipity items. This allows us to provide the recommendations that are nearer to the users' intent. We conducted experiments on real-world datasets, *Amazon* and *Yelp*. We evaluated the proposed approach using precision, recall, F1-Score, and unexpectedness metrics. The results show that the proposed approach performed better in recommending surprise items.

Keywords

Recommender System, Top Recommendations, Serendipity, Collaborative Filtering, Sentiment Analysis, K-Nearest Neighbors.

Introduction

Recommender system (RS) (Ricci et al. 2011; Sarwar et al. 2001) suggests relevant items to a new user by analyzing the purchase behavior of the existing users. In the literature, several approaches were proposed to analyze user preferences, viz., content-based filtering (CBF) (Pazzani and Billsus 2007; Son and Kim 2017), collaborative filtering (CF) (Kluver et al. 2018), and hybrid filtering (Bostandjiev et al. 2012; Burke 2002; Xiong et al. 2018) approaches. The CF approach is one of the popular approaches to generate recommendations. The CF approach analyzes the user purchase history of the existing users consisting of user-reviews, user-ratings, etc. User-based CF and item-based CF are commonly used approaches in top recommendations. Several CF approaches have been proposed to address RS accuracy (Aramanda et al. 2021; Kluver et al. 2018; Sarwar et al. 2001; Shen et al. 2019), serendipity (De Gemmis et al. 2015; Kotkov et al. 2018; Kotkov et al. 2016), diversity (Karakaya and Aytekin 2018; Kumara Swamy et al. 2017; Kumara Swamy and Krishna Reddy 2015; Ma et al. 2023), etc.

Generally, the CF approach analyzes the purchase behavior of the existing users. The item/user similarity of the new user and the existing users is computed from the purchase behavior. An item that is more likely to be interesting is provided as a recommendation to the new user. While making the recommendations, the top-rated items are considered. Due to this, the low and average-rated items are neglected even if a new user likes them. We explain this situation with the help of an Example 1 in the proposed approach. The example shows that there is a variation in user-reviews and user-ratings, and the user opinions are expressed more clearly in user-reviews. Considering only user-ratings, the RS always recommends the similar products and the surprise is missing in the recommendations. As a result, recommendations are missing the items which may be nearer to the user intent. It leads to user dissatisfaction.

In this paper, we propose an approach to improve user satisfaction in recommendations. We consider the user-reviews of the product to update the user ratings. In the literature (Aramanda et al. 2023; Shen et al. 2019), efforts were made to use the user-reviews to improve the accuracy of the RS. The users provide comments on the items in the form of user-reviews. We observe that the user provides their opinions in the form of emotions. The emotions help us to know what a user felt about an item. From these emotions, we can identify whether users are thrilled/surprised about item purchases. These surprising items make users feel more satisfied (Chen et al. 2019) than simply seeing the routine item purchases. Considering these emotions, it is possible to improve user satisfaction with recommendations that may be nearer to the user intent. Hence, by updating the user-ratings of an item using the user-reviews, it is possible to improve the user satisfaction in RS.

In the proposed approach, *Serendipitous Recommender System* (SRS), we consider both user-ratings and user-reviews to recommend the serendipity item(s) to the new users. We call an item as a serendipity item where a user feels the surprise on looking at the recommended item. The proposed approach extracts user opinions available in the form of emotional words (such as surprise words) from the user-reviews. These emotional words are used to update the user-ratings to find the serendipity items. The updated user-ratings data is used to find similarity between users as well as items. These similarities are used to generate the top recommendations using the K-Nearest Neighbor (KNN) algorithm. We conducted the experiments using the real-world *Amazon* and *Yelp* datasets to evaluate the proposed approach. We evaluated the proposed approach using precision, recall, F1-Score, and unexpectedness metrics. The results show that the proposed approach improves the user satisfaction as compared to the existing approaches.

The contributions of the proposed approach are as follows.

- Extracting the emotional words (such as surprise words) from the user-reviews.
- We update the user-ratings using emotional words to identify the serendipity items.
- Propose an approach called Serendipitous Recommender System (SRS) to recommend the serendipity items.
- Conduct experiments on real-world datasets, *Amazon* and *Yelp*.

Rest of the paper is organized as follows. Related work, proposed approach, experimental results are provided from section 2 to 4 respectively. Finally, Section 5 provides the conclusion.

Related Work

We provide the related work on CF and serendipity.

The CF (Deshpande and Karypis 2004; Kluver et al. 2018) is a widely applied technique for recommending items in the area of RSs. The CF approaches can be user-based CF (UCF) or item-based CF (ICF) (Koren 2010; Sarwar et al. 2000). The user-based CF approach (Koohi and Kiani 2017) suggests the relevant items on user similarity; Whereas in item-based CF approaches (Deng et al. 2019; Najafabadi et al. 2017) is based on the item similarity. To find the similarities between users/items, similarity measures such as cosine similarity, Pearson correlation, etc., are used. In the literature (Bhuvaneshwari et al. 2023; Coscrato and Bridge 2023; Fayezi and Golpayegani 2023), several CF approaches have been proposed to improve the accuracy of RS using ratings by applying similarity measures and machine learning algorithms. Sentiment analysis (Liu 2020) also have been used in CF to improve performance of RS (Aivazoglou et al.

2020; Aramanda et al. 2023). Apart from the traditional CF-RS to improve the accuracy, several factors are considering to recommend the items, such as diversity (Karakaya and Aytekin 2018; Kumara Swamy et al. 2017; Kumara Swamy and Krishna Reddy 2015; Ma et al. 2023), serendipity (De Gemmis et al. 2015; Kotkov et al. 2018; Kotkov et al. 2016) to enhance user satisfaction, etc.

Specifically, serendipity (Ziarani and Ravanmehr 2021) in RSs suggests items that are not obvious and unpopular, and still feel thrilled/surprised to a user. In (Kotkov et al. 2018), conducted a survey and found that serendipity broadens the user preferences. The survey also uncovered that different types of serendipity and unexpectedness have varying effects on the expansion of preferences and satisfaction of users. In (De Gemmis et al. 2015), studied the serendipity problem in terms of diversity and unexpectedness.

The proposed approach is different from the traditional CF approaches. The proposed approach applied user emotions to recommend the serendipity items. The existing approaches are limited to user-ratings for recommending serendipity items. In contrast, the proposed approach considered both user-ratings and user-reviews. We evaluate the proposed approach using the unexpectedness metric to evaluate serendipity. We compare the proposed approach with neighborhood based approaches (Koren 2010; Töscher et al. 2008), user-based KNN, item-based KNN, user-based KNN with means, and item-based KNN with means in experimental results.

Proposed Approach

In this section, we describe problem statement, basic idea, and proposed SRS approach.

Generally, in CF-RS, user/item similarities are computed to identify the user preferences using user-ratings and generate the recommendations. We explain the issue of the user-ratings and the importance of the proposed approach considering Example 1.

Example 1. *In this example, we consider two users,* Bob *and* Simson*, who purchased the items and provided their feedback in the user-ratings and user-reviews as shown In Table 1. This table consists of four attributes viz., user name, item identification (Id), user-rating and user-review.* Bob *gave five user-rating (top-rating) and positive user-review for* $item_1$. Bob *also gave three user-rating (average-rating) and a neutral user-review for* $item_2$*. Similarly,* Simson *gave three user-rating and neutral user-review for* $item_2$. Simson *also gave two user-rating (low-rating) and mixed (positive and negative) user-review for* $item_3$.

User Name	Item Id	User-rating	User-review
Bob	$item_1$	5	"This is definitely a good quality nozzle."
Bob	$item_2$	3	"It is ideal if you have a small lawn or want to get a little exercise while mowing."
Simson	$item_2$	3	"This is a cheap mower. It might be great for someone who has a small yard, but my yard isn't small."
Simson	$item_3$	2	"Pros: quick performer, favorable to small lawn. Cons: It does not have a gasket on the bottom thread so you'll need to add some plumber's tape to get a solid seal."

Table 1. Sample user-reviews along with ratings provided by the users, Bob and Simson

Now, we construct a rating matrix for *Bob* and *Simson* on three items from Table 1 as shown in Table 2 to find the similarities between them, where '?' entries are the items which are not purchased yet. From this table, we can say that *Bob* and *Simson* are similar because they gave the same rating for the $item_2$. Applying CF approach on this matrix, $item_1$ is suggested to *Simson* as *Bob* likes it more and hence *Simson* also likes it. Moreover, $item_3$ is not suggested to *Bob* by assuming that *Simson* gave a low rating, hence *Bob* does not like it. However, in reality, *Bob* may like the $item_3$. In this case, we may miss the user interest in $item_3$. Here, the issue is the item may be liked by the one user, but that item has not recommended by CF-RS approach.

To address this issue, we need to consider the user-reviews to check whether the user really likes any of its features so that *Bob* may like that feature. Considering user-reviews to update user–ratings using serendipity.

	$item_1$	$item_2$	$item_3$
Bob	5	3	?
Simson	?	3	2

Table 2. User-rating matrix for Bob and Simson

From Table 1, we can observe that users provide their opinions clearly in user-reviews. In the case of *Simson* on $item_3$, considering user-rating CF does not suggest to Bob. However, he mentions that some qualities of $item_3$ are good, like 'quick performer', and 'favorable to small lawn'. At the same time, he also mentioned that some qualities are not liked. It indicates that the $item_3$ can be suggested to users who are interested in its good qualities. Hence, $item_3$ is suggested to *Bob* if he is interested in its good qualities. The problem is checking whether *Bob* is interested in that item. It can be solved by extracting extra information from the user-reviews.

Here, we propose an approach to address the issue of not recommending the items to new user which may interested to him/her. Due to the fact that in CF-RS only top-rated items are recommended and low/average rated items are not considered at all.

Now, we define the basic idea as follows.

Basic Idea. *Let U be a set of users that provides their preferences in user-ratings, R, and user-reviews, D. The basic idea is to recommend the serendipity items by updating R with user emotions extracted from D.*

Serendipitous Recommender System

We propose an approach called *Serendipitous Recommender System* (SRS) to recommend the serendipity items using both user-ratings and user-reviews. In this approach, we extract the emotional words from user reviews to identify the surprise emotion of a user. The surprise emotion helps us to identify the serendipity items in the recommendations. These surprise words are used to update the user-ratings. The updated ratings are used in CF to generate the recommendations. Now, we explain the proposed approach as follows.

The proposed approach, SRS, is a two step process.

1. Update the user-rating with serendipity score, and
2. Top recommendation

1. Update the user-rating with serendipity score

In this step, we extract the emotional words from the user-reviews. From the emotional words, we identify the surprise emotions to update the user-ratings. Generally, users provide their opinions, also called emotions in user-reviews about the items which they purchased. The user emotions imply the user's intentions/preferences. Hence, analyzing user-reviews makes the users' preferences more nearer to their intent. Now, the process of identifying the emotions is explained as follows.

Let D be a z number of user-reviews from 1 to z given by users on items, where d_i is an i^{th} user-review. Similarly, each d_i consists of p number of words from 1 to p, where w_{ij} is the j^{th} word. We identify emotional words in each d_i.. We call a word an emotional word if it has an emotion; otherwise we call it a neutral word. We consider each w_{ij} , $1 \leq j \leq p$, in d_i to identify it as an emotional word or not. The w_{ij}'s emotion value is represented with e_{ij} . The e_{ij} value is one if w_{ij} is expressing the surprise emotion; Otherwise zero and it is defined in Eq. (1). The w_{ij} is an emotional word if e_{ij} value is one, otherwise w_{ij} is a neutral word.

$$e_{ij} = \begin{cases} 1, & \text{If } w_{ij} \text{ has } \textit{surprise} \text{ emotion;} \\ 0, & \text{Otherwise;} \end{cases} \tag{1}$$

In the next step, we accumulate all emotional words in d_i and compute the significance of surprise emotion to form emotion score, E_i. The E_i is defined in Eq. (2), where C is the total number of emotional words in d_i and λ is significance of surprise emotion, i.e., how important a surprise emotion is in D. The λ is defined in Eq. (3), where D_{sur} is the number of user reviews that have surprise emotion in D. Similarly, compute E_i for all the user-reviews in D.

$$E_i = \left(\frac{\sum_{j \in 1 \text{ to } \mathrm{C}} (e_{ij})}{C} \right) * \lambda \tag{2}$$

$$\lambda = log \left(\frac{\mathrm{z}}{D_{sur}} \right) \tag{3}$$

Later, we compute the ratings with serendipity score, S, by updating the ratings with surprise emotion. The updated user-rating for i^{th} user-review is defined in Eq. (4), where r_i is user-rating for i^{th} user-review in D given by user.

$$S_i = E_i + r_i \tag{4}$$

The resultant S is normalized as R' from the range 1 to 5 using min-max normalization. Finally, the R' is used in top recommendations.

2. Top recommendations

The SRS recommends top items using user-based and item-based recommendation approaches (Sarwar et al. 2000; Töscher et al. 2008). These approaches use the K-nearest neighbors (KNN) algorithm from updated ratings, R'. The updated ratings are computed with surprise emotion. For this, we map R' to the user-item matrix, $U_{m \times n}$, where m is the number of users and n is the number of items. Each element, r_{ik}, in this matrix is a rating value in between 1 to 5 if user 'i' gave rating on item 'k'; Otherwise zero.

In user-based SRS, we compute similarity between users using cosine similarity measure (Sarwar et al. 2001). Next, we find the K neighbors of a user using K-nearest neighbors (KNN) algorithm to predict the unknown ratings of a user. Later, we arrange all the predicted ratings in descending order and suggest the top items as the top recommendations to a new user. The computation of unknown user rating, $\tilde{r}_{ik}$, is defined in Eq. (5), where 'i' and 'j' are users, NB is a set of nearest neighbors of user, 'a' is number of neighbors, S is similarity score between users, and 'k' is item.

$$\tilde{r}_{ik} = \frac{\sum_{j \in NB_k{}^a(i)} S(i,j).r_{jk}}{\sum_{i \in NB_k{}^a(i)} S(i,j)} \tag{5}$$

Similarly, the items-based SRS computes the similarity between items using cosine similarity (Sarwar et al. 2001) and finds the K neighbors of an item to suggest the top-N items. The computation of unknown user rating using item-based SRS is defined in Eq. (6), where 'k' and 'l' are items.

$$\tilde{r}_{ik} = \frac{\sum_{l \in NB_i{}^a(k)} S(k,l).r_{il}}{\sum_{l \in NB_i{}^a(k)} S(k,l)} \tag{6}$$

Along with this, we have also considered mean values of ratings in SRS. The computation of unknown preferences considering the mean for user-based SRS is defined in Eq. (7), where 'μ_i' is the mean of 'i' ratings. Similarly, for item-based SRS with means is defined in Eq. (8).

$$\tilde{r}_{ik} = \mu_i + \frac{\sum_{j \in NB_k{}^a(i)} S(i,j).(r_{jk} - \mu_j)}{\sum_{j \in NB_k{}^a(i)} S(i,j)} \tag{7}$$

$$\tilde{r}_{ik} = \mu_k + \frac{\sum_{l \in NB_i{}^a(k)} S(k,l).(r_{il} - \mu_l)}{\sum_{l \in NB_i{}^a(k)} S(k,l)} \tag{8}$$

Experiment Results

This section gives data description and experimental results.

Data Description

We evaluated the proposed approach considering real-world datasets *Yelp* (Group 2013) and *Amazon* (*Cell* and *Clothing*) (McAuley 2014).

The data description is shown in Table 3, where the first to fourth represents the dataset name, the reviews/ratings, the number of users, and the number of items in a given dataset, respectively. The format of each dataset is json format. We considered stars/overall ratings as user-ratings and review text/text as user-reviews from datasets. We preprocessed the user-reviews by removing stop words and special symbols. We identify the surprise emotions using NRC emotion lexicon (Mohammad 2016).

Dataset	Reviews/ Ratings	Number of Users	Number of Items
Cell Phones and Accessories (Cell)	194,439	27,879	10,429
Clothing, Shoes, and Jewelry (Clothing)	278,677	39,387	23,033
Yelp	229,907	45,981	11,537

Table 3. Data description

Results

In this section, we evaluate the proposed approach using precision, recall, F1-Score and unexpectedness (to evaluate the serendipity) metrics (Sarwar et al. 2000; Silveira et al. 2019).

The precision, recall, F1-Score and unexpectedness are shown in Eq. (9), Eq. (10), Eq. (11), and Eq. (12), respectively, where size of hit set is number of recommended items that are relevant, S-top-N_i is top-N serendipity items for i^{th} user, and T-top-N_i is actual top-N items for i^{th} user.

$$\text{Precision} = \frac{\text{size of hit set}}{\text{size of top-N set}} \tag{9}$$

$$\text{Recall} = \frac{\text{size of hit set}}{\text{size of test set}} \tag{10}$$

$$\text{F1-Score} = \frac{2 * \text{precision} * \text{recall}}{\text{recall} + \text{precision}} \tag{11}$$

$$\text{Unexpectedness} = \frac{\sum_{i \in \text{test set}} \frac{(\text{S-top-N}_i)-(\text{T-top-N}_i)}{\text{size of S-top-N}_i}}{\text{size of test set}} * 100 \tag{12}$$

We compared the proposed approach, SRS, with the following existing neighborhood based approaches (Koren 2010; Töscher et al. 2008).

1. User-based KNN (UCF): This approach uses Eq. (5) to find the K-nearest neighbors of a user from user-ratings.
2. Item-based KNN (ICF): This approach used Eq. (6) to find the K-nearest neighbors of an item from user-ratings.
3. User-based KNN with means (UCF-m): This approach uses Eq. (7) to find the K-nearest neighbors of a user from user-ratings.
4. Item-based KNN with means (ICF-m): This approach uses Eq. (8) to find the K-nearest neighbors of an item from user-ratings.

Approaches	Datasets								
	Precision			Recall			F1-Score		
	Cell	*Clothing*	*Yelp*	*Cell*	*Clothing*	*Yelp*	*Cell*	*Clothing*	*Yelp*
UCF	**0.8566**	**0.8977**	0.7966	0.8566	0.8977	0.7966	**0.8748**	**0.9153**	0.8045
ICF	0.8455	0.8962	0.7723	**0.8789**	**0.9311**	0.7923	0.8619	0.9133	0.7822
UCF-m	0.8160	0.8718	0.7492	0.8407	0.9018	0.7546	0.8282	0.8865	0.7519
ICF-m	0.8354	0.8892	0.7667	0.8541	0.9137	0.7660	0.8446	0.9013	0.7663
SRS-1	0.8496	0.8971	**0.8828**	0.8777	0.9305	**0.8120**	0.8634	0.9135	**0.8459**
SRS-2	0.8351	0.8960	0.7721	0.8607	0.9281	0.7916	0.8477	0.9118	0.7817
SRS-3	0.7684	0.8368	0.7498	0.7777	0.8569	0.7545	0.7730	0.8467	0.7521
SRS-4	0.8062	0.8765	0.7666	0.7998	0.8828	0.7659	0.8030	0.8796	0.7662

Table 4. Performance comparison using precision, recall and F1-Score

We provide four variations of the proposed SRS as follows.

5. SRS-1: It is a user-based approach devised with Eq. (5) using updated ratings, R'.
6. SRS-2: It is an item-based approach devised with Eq. (6) using updated ratings, R'.
7. SRS-3: It is a user-based approach devised with Eq. (7) using updated ratings, R'.
8. SRS-4: It is an item-based approach devised with Eq. (8) using updated ratings, R'.

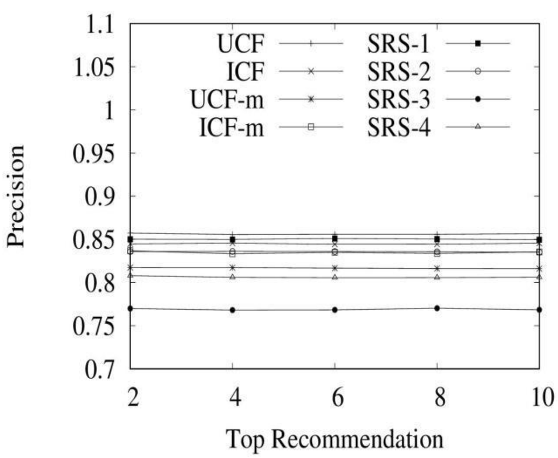
Approaches	Datasets		
	Cell	*Clothing*	*Yelp*
UCF	49.1384	49.3858	35.4340
ICF	49.2462	49.2478	35.5407
UCF-m	49.3440	49.2899	35.0816
ICF-m	49.2112	49.3564	35.0930
SRS-1	49.3305	**49.4055**	35.3553
SRS-2	49.2067	49.2638	**35.7207**
SRS-3	**49.3919**	49.3536	35.0805
SRS-4	49.3096	49.3491	35.0118

Table 5. Comparison proposed approach and existing approaches using unexpectedness

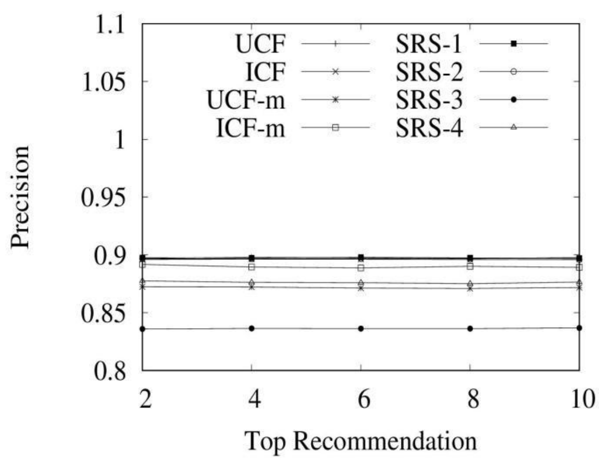
The user-rating data consists of 1 to 5 rating indicating 1 as low and 5 as high like. We consider the user rating above 2 as liked and 2 and below as not liked by the user. In all the experiments, we considered K as 20 neighbors and ≥3 rated items to generate top recommendations. We divided the dataset into 80% train and 20% test set. We performed 5-fold cross-validation, and the average is reported.

All approaches precision, recall and F1-Score on *Cell*, *Clothing*, and *Yelp* dataset shown in Table 4. From this table, we can observe the performance is almost similar to the existing approaches for *Cell* and *Clothing* datasets. In the case of *Yelp* dataset, the proposed approach performed comparatively better than existing approaches. Among all versions of the proposed approach (SRS), SRS-1 (User-based SRS) performs better.

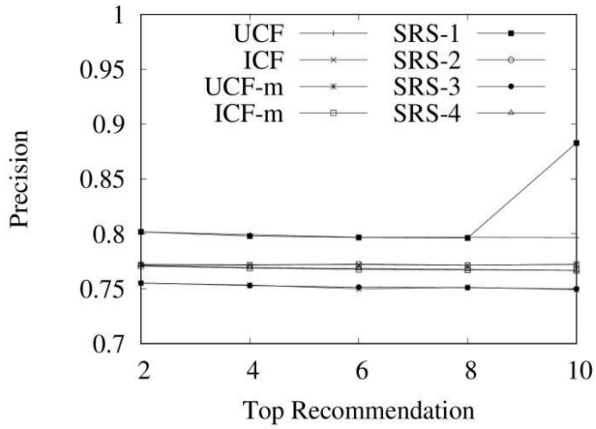
We also generate the recommendations varying the N value from 2 to 10 for the datasets *Cell*, *Clothing*, and *Yelp* and results shown in Figure. 1. From this figure, we can observe that the proposed approach performance is almost similar to the existing approaches.



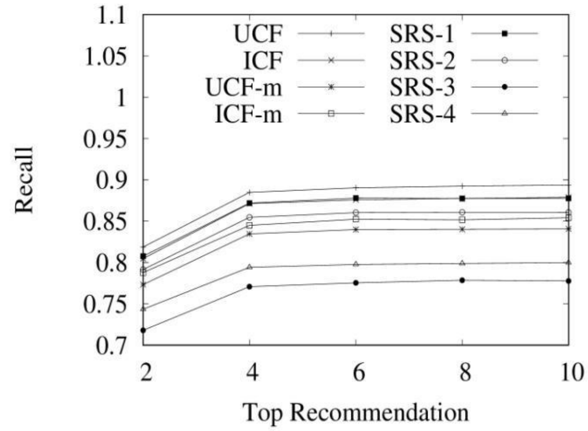
(a) *Cell* Precision



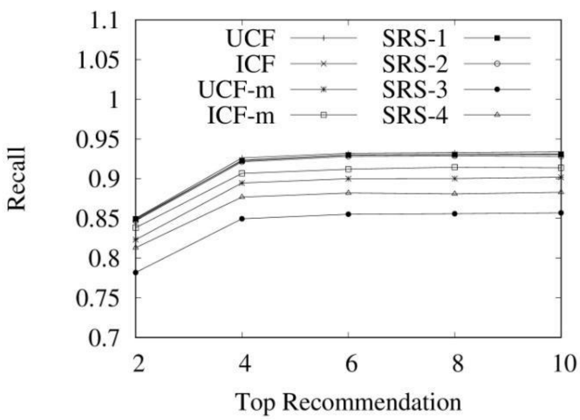
(b) *Clothing* Precision



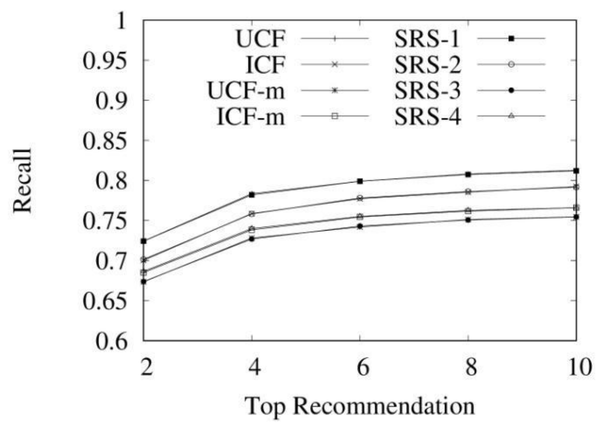
(c) *Yelp* Precision



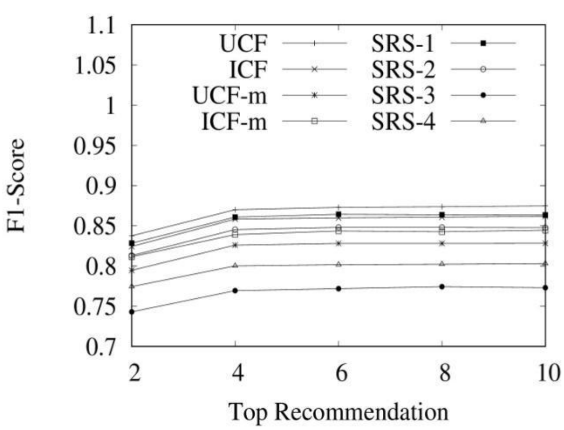
(d) *Cell* Recall



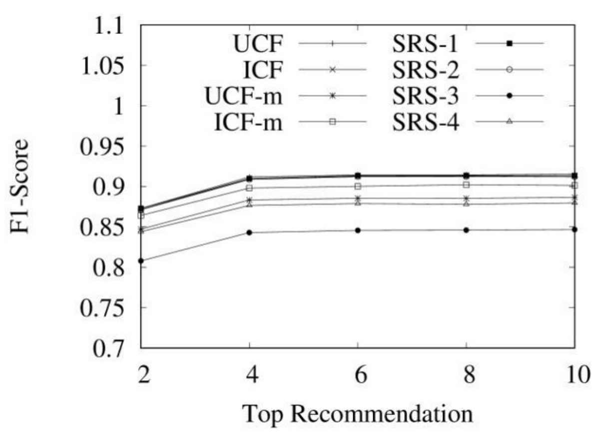
(e) *Clothing* Recall



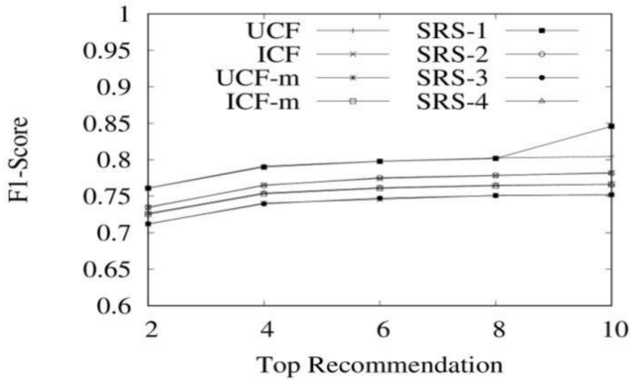
(f) *Yelp* Recall



(g) *Cell* F1-Score



(h) *Clothing* F1-Score



(i) *Yelp* F1-Score

Figure 1. Performance comparison of SRS with existing approaches at different Top for Cell, Clothing and Yelp datasets

To measure the serendipity in the recommended items, we compute unexpectedness in the recommended items. The comparison of proposed approach and existing approaches is shown in Table 5.

From Table 5, we can observe that the proposed approach performed better to recommend the unexpected items as compared to the existing approaches.

From these experiments, we conclude that the proposed approach performs almost similar and/or slight improvement to the existing approaches for *Cell* and *Clothing* datasets. However, the proposed approach recommended unexpected (also called serendipity) items compared to the existing approaches. Due to the fact that serendipity score has helped in recommendations. The proposed approach performed better as compared to the existing approaches for *Yelp* dataset. The reason for the better performance is that the proposed approach is able to understand the user emotions from the user-reviews. The emotions in user-reviews helped to update the user-ratings to reach user intent.

Overall, updating the user-ratings using the emotional words available in the user-reviews helps to improve the user satisfaction.

Discussion

In the proposed approach, we used both user-ratings and user-reviews. The user-reviews are used to update the user-ratings to recommend the serendipity items.

Using the proposed approach, we update the user ratings from the emotional words in the user reviews. In the existing approaches, the serendipity items are not recommended due to the low ratings. However, the proposed approach updates the user ratings for the serendipity items and makes the items to be available for recommendations.

In the proposed approach, our main aim is to improve the user satisfaction. However, it may lead to increase/decrease the RS performance. In the proposed approach, we have not considered the time and sequence of item purchases data. We consider this as a future work. Further, the proposed approach considered only surprise emotion. We consider the other emotions as a part of future work.

Conclusion

A recommender system (RS) suggests the relevant items to the new user by analyzing the purchase behavior of existing users. The CF approach is a popular approach in the area of (RS). The similarity based CF-RS recommends the top items based on top rated items and neglect the low/average rated items. This results into the missing of user interested items in recommendations and leads to user dissatisfaction. To address, we propose an approach called *Serendipitous Recommender System* (SRS) to recommend the items which are liked by users even though they are not top rated. To achieve this, the proposed approach uses user-reviews along with user-ratings. The SRS extracts the emotional words from user-reviews to update the user-ratings. The updated user-ratings are used to recommend the serendipity items using KNN algorithm. To evaluate the proposed approach, we conducted experiments using real-world datasets, *Amazon* and *Yelp*. We compared the proposed approach with existing approaches using precision, recall, F1-Score and unexpectedness metrics. The results show that the proposed approach performed better to recommend the serendipity items as compared to the existing approaches.

In a part of the future work, we would like to produce top recommendations considering negation words to extract the user emotions.

References

Aivazoglou, M., Roussos, A. O., Margaris, D., Vassilakis, C., Ioannidis, S., Polakis, J., and Spiliotopoulos, D. (2020). "A fine-grained social network recommender system," Social Network Analysis and Mining (10:1), pp. 1–18.

Aramanda, A., Abdul, S. M., and Vedala, R. (2021). "Refining User Ratings Using User Emotions for Recommender Systems," in Proc. of Int. Conf. on Information Integration and Web Intelligence, ACM, pp. 3– 10.

Aramanda, A., Abdul, S. M., and Vedala, R. (2023). "enemos-p: An enhanced emotion specific prediction for recommender systems," Expert Systems with Applications , p. 120190.

Bhuvaneshwari, P., Rao, A. N., and Robinson, Y. H. (2023). "Top-N Recommendation System Using Explicit Feedback and Outer Product Based Residual CNN," Wireless Personal Communications (128:2), pp. 967–983.

Bostandjiev, S., O'Donovan, J., and Höllerer, T. (2012). "TasteWeights: A visual interactive hybrid recommender system," in Proc. of ACM Conf. on Recommender Systems, ACM, pp. 35–42.

Burke, R. (2002). "Hybrid recommender systems: Survey and experiments," User modeling and user-adapted interaction (12:4), pp. 331–370.

Chen, L., Yang, Y., Wang, N., Yang, K., and Yuan, Q. (2019). "How serendipity improves user satisfaction with recommendations? a large-scale user evaluation," in Proc. of Int. Conf. on world wide web, ACM, pp. 240–250.

Coscrato, V. and Bridge, D. (2023). "Estimating and evaluating the uncertainty of rating predictions and top-n recommendations in recommender systems," ACM Transactions on Recommender Systems.

De Gemmis, M., Lops, P., Semeraro, G., and Musto, C. (2015). "An investigation on the serendipity problem in recommender systems," Information Processing & Management (51:5), pp. 695–717.

Deng, J., Guo, J., and Wang, Y. (2019). "A Novel K-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering," Knowledge-based Systems (175), pp. 96–106.

Deshpande, M. and Karypis, G. (2004). "Item-based top-n recommendation algorithms,"ACM Transactions on Information Systems (22:1), pp. 143–177.

Fayezi, M. M. and Golpayegani, A. H. (2023). "User structural information in priority-based ranking for top-N recommendation," Advances in Computational Intelligence (3:1), p. 3.

Group, Y. (2013). "Yelp Recruiting Competition," (available online at https ://www.kaggle.com/c/yelp -recruiting/data;).

Karakaya, M. Ö. and Aytekin, T. (2018). "Effective methods for increasing aggregate diversity in recommender systems," knowledge and Information Systems (56), pp. 355–372.

Kluver, D., Ekstrand, M. D., and Konstan, J. A. (2018). "Rating-based collaborative filtering: algorithms and evaluation," in Social Inf. Access, Springer, pp. 344–390.

Koohi, H. and Kiani, K. (2017). "A new method to find neighbor users that improves the performance of collaborative filtering," Expert Systems with Applications (83), pp. 30–39.

Koren, Y. (2010). "Factor in the neighbors: Scalable and accurate collaborative filtering," ACM Transactions on Knowledge Discovery from Data (4:1), pp. 1–24.

Kotkov, D., Konstan, J. A., Zhao, Q., and Veijalainen, J. (2018). "Investigating serendipity in recommender systems based on real user feedback," in Proc. of symposium on applied computing, pp. 1341–1350.

Kotkov, D.,Wang, S., and Veijalainen, J. (2016). "A survey of serendipity in recommender systems,"KnowledgeBased Systems (111), pp. 180–192.

Kumara Swamy, M., Krishna Reddy, P., and Bhalla, S. (2017). "Association rule based approach to improve diversity of query recommendations," in Proc. of Int. Conf. on Database and Expert Systems Applications, Springer, pp. 340–350.

Kumara Swamy, M. and Krishna Reddy, P. (2015). “Improving diversity performance of association rule based recommender systems,” in Proc. of Int. Conf. on Database and Expert Systems Applications, Springer, pp. 499–508.
Liu, B. (2020). Sentiment analysis: Mining opinions, sentiments, and emotions, Cambridge university press. Ma, T.-m., Wang, X., Zhou, F.-c., and Wang, S. (2023). “Research on diversity and accuracy of the recommendation system based on multi-objective optimization,” Neural Computing and Applications (35:7), pp. 5155–5163.
McAuley, J. (2014). “Amazon product data,” (available online at http://jmcauley.ucsd.edu/data/amazon/;).
Mohammad, S. M. (2016). “The Sentiment and Emotion Lexicons,” (available online at http://sentiment.nrc.ca/lexicons-for-research;).
Najafabadi, M. K., Mahrin, M. N., Chuprat, S., and Sarkan, H. M. (2017). “Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data,” Computers in Human Behavior (67), pp. 113–128.
Pazzani, M. J. and Billsus, D. (2007). “Content-based recommendation systems,” in The adaptive web, Springer, pp. 325–341.
Ricci, F., Rokach, L., and Shapira, B. (2011). “Introduction to recommender systems handbook,” in Recommender systems handbook, Springer, pp. 1–35.
Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). “Analysis of recommendation algorithms for ecommerce,” in Proc. of ACM Conf. on EC, ACM, pp. 158–167.
Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). “Item-based collaborative filtering recommendation algorithms,” in Proc. of Int. Conf. on World Wide Web, ACM, pp. 285–295.
Shen, R.-P., Zhang, H.-R., Yu, H., and Min, F. (2019). “Sentiment based matrix factorization with reliability for recommendation,” Expert Systems with Applications (135), pp. 249–258.
Silveira, T., Zhang, M., Lin, X., Liu, Y., and Ma, S. (2019). “How good your recommender system is ? A survey on evaluations in recommendation,” International Journal of Machine Learning and Cybernetics (10:5), pp. 813–831.
Son, J. and Kim, S. B. (2017). “Content-based filtering for recommendation systems using multiattribute networks,” Expert Systems with Applications (89), pp. 404–412.
Töscher, A., Jahrer, M., and Legenstein, R. (2008). “Improved neighborhood-based algorithms for large scale recommender systems,” in Proc. of KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition, pp. 1–6.
Xiong, R., Wang, J., Zhang, N., and Ma, Y. (2018). “Deep hybrid collaborative filtering for web service recommendation,” Expert Systems with Applications (110), pp. 191–205.
Ziarani, R. J. and Ravanmehr, R. (2021). “Serendipity in recommender systems: a systematic literature review,” Journal of Computer Science and Technology (36), pp. 375–396.