Asian Journal of Research in Computer Science

5(2): 10-16, 2020; Article no.AJRCOS.54759
ISSN: 2581-8260

Recommending Curated Content Using Implicit Feedback

Debashish Roy[1*]

[1]Department of Computer Science, Ryerson University, Toronto, ON, Canada.

Author's contribution

The sole author designed, analyzed and interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/AJRCOS/2020/v5i230130
Editor(s):
(1) Dr. G. Sudheer, GVP College of Engineering for Women, India.
Reviewers:
(1) Jackson Akpojaro, University of Africa, Nigeria.
(2) Marius Iulian Mihailescu, University of Bucharest, Romania.
Complete Peer review History: http://www.sdiarticle4.com/review-history/54759

Short Research Article

Received 23 December 2019
Accepted 28 February 2020
Published 11 March 2020

ABSTRACT

Matrix factorization (MF) which is a Collaborative filtering (CF) based model, is widely used in the recommendation systems (RS). For our experiment, we collected data from a company's internal web site where curated contents are published and pushed to the employees. However, the size of the dataset is small and interaction data is also limited. We got a sparse matrix when we generated a user-item rating matrix. We have used Multi-Layer Perceptron (MLP) to calculate the rating scores from the implicit feedbacks. However, on this sparse dataset traditional content only or CF-only RSs do not work well. Here, we propose ahybrid RS that incorporates content similarity scores into an MLP-based MF-model. To integrate the content similarity scores into the MF, we have defined an objective function based on a regularization term. The experimental result shows that our proposed model demonstrates a better result than the traditional MF-based models.

Keywords: Matrix factorization; LDA; TF-IDF; collaborative filtering; regularization; objective function; NLP.

1. INTRODUCTION

The problem of information overload has become prominent owing to the rapid growth in the amount of available digital information and the increasing number of visitors to the different web resources. Because of the huge volume of information, it is hard for the users to find items

**Corresponding author: E-mail: debashish.roy@ryerson.ca;*

of interest in time. To help users find items of their interest, recommender systems are widely used [1]. Recommender Systems (RS) are information filtering systems that recommend items to different users according to users' preferences, interests, or observed behavior on items. Depending on a user profile, RS can determine whether a particular user will like an item or not. To build an RS, various approaches have been developed, including collaborative filtering, content-based filtering and hybrid filtering [2]. Collaborative filtering (CF) is most commonly used and it recommends items by identifying other users with similar interest; it uses users' opinion to develop a rating matrix to recommend items to an active user. Content-based (CT) technique recommends items based on content similarity; they are based on a description of the item and a profile of the user's preferences. Hybrid filtering technique combines the CF and CT based filtering techniques [3,4] in order to increase the accuracy of the recommender systems.

Most of the CF-based recommender systems use a rating matrix to recommend items. The rating matrix is generated from the implicit or explicit ratings given by the users for different items; rating data retrieved from different content provider websites can be used as explicit feedbacks, user interaction data such as likes, comments etc. can be used as implicit feedbacks. However, most of the CF-based RSs do not include text information of the items in their rating matrix. The CF-based recommenders suffer a lot when the rating matrix is sparse, but the inclusion of text-information into the CF-models, improves the accuracy of the recommenders. In our research work, we have generated a user-item rating matrix based on the implicit feedbacks gathered from different users. We have used MLP to calculate the implicit feedbacks. To find similarity scores between items, we have used two data mining techniques, such as: TF-IDF and LDA [5,6]. To include the item similarity scores into the MF model, we have defined and implemented an objective function based on an item-similarity based regularization term. The experimental research shows that our MLP and TF-IDF based hybrid RS demonstrates a better result than the traditional MF-based models.

The contributions of this paper are summarized below:

- We have applied an MLP model to calculate the implicit feedbacks to generate the user-item rating matrix.
- Based on an item-similarity based regularization term, we have defined an objective function for matrix factorization.
- In a comparative study, we have found that MLP and TF-IDF based hybrid RS demonstrates a better result than the traditional MF-based models.

The rest of the paper is organized as follows: related work section reviews and analyzes the existing research work. The experiment design section explains the steps of our experiment. The result analysis section discusses the result of the experiment. Finally, in the last section, we conclude with a summary of results and analysis along with a future research direction.

2. RELATED WORK

Different types of matrix factorization techniques have been used widely in hybrid recommender systems. In [7] a two-level matrix factorization (TLMF) has been proposed. TLMF computes the semantic relations between items based on a novel approach - Weighted Textual Matrix Factorization (WTMF). In WTMF, a textual corpus is represented by a term document matrix, where the rows are words and columns are sentences. Each element in the term document matrix has the TF*IDF value for each word. TLMF uses the rating matrix along with the relations between items into account. In [8], homophily effect has been used to predict trust for online users. Homophily effect suggests that similar users have a higher likelihood to establish trust relations. Cosine similarity of users' rating vectors are used to measure their rating similarity and this rating similarity is a homophily co-efficient. Then a homophily based regularization term is used to diffuse homophily co-efficient into the matrix factorization model. In [9], implicit user feedbacks have been used to build a recommender system. Here, for any two given items, a similarity score is calculated. User feedback matrix and various item similarity matrices are combined by diffusing the item similarity information into the feedback matrix. Compared to these papers, our hybrid recommender is focused on: i) an MLP based learning model which is used to calculate the implicit feedback score; ii) an objective function, which is based on an item-similarity based regularization term.

3. MATERIALS AND METHODS

In this section, we will describe the different steps taken to design the experiment. For our experiment, we have collected data from a company's internal web site where curated contents are published and pushed to the employees. At first, on the collected dataset we have applied pre-processing to get rid of unwanted or noisy words. Then, we used the cleaned data set to continue the experiment.

3.1 Description of the Dataset

Different employees of the company can create different posts on their internal website. Other users are able to like a post or leave a comment on that post. Again, the post can be shared on different social network platforms. Their system also records social network interaction data for any post. So, in the dataset we have all the posts and user interaction statistics from their internal website and also from the social network platforms. In the collected dataset, we have 110 users and 4000 posts. The dataset is very sparse as we have found on an average 10 different users interact with each post.

3.2 Implicit Rating Calculation

The rating data retrieved from the company's website does not contain any explicit rating for different posts as the company collects only user interaction data. The interaction data is used to calculate an implicit rating for each post. We have used equation (1) to calculate the implicit rating. Here, n is the total number of different types of interactions, x is the total score for an interaction and w is the weight associated for each type of interaction.

$$\text{ImplicitRating} = \sum_{k=0}^{n} x^k w^k \tag{1}$$

At first, to calculate the implicit rating we assigned different weights to different types of user interactions. To assign the weights, we used user's involvement as a metric. If one user comments on a post and another user likes the same post, then we think the user who commented on the post is more attached to the post. For example, we used the following weights for different interactions: twitter share = 4, Facebook share = 4, Facebook Comment = 5, twitter reply = 5, Facebook like =3 etc. Then using these scores, we generated a user-post rating matrix. Later, to optimize the interaction weights we used an MLP-based learning model. To build the model, we asked the employees of the company to rate different posts explicitly in a scale of 1 to 5. Then, we used the implicit feedbacks and explicit feedback to design a regression model. A sample of the dataset is given in Fig. 1.

```
0,0,0,0,0,0,1,640,0,3,0,7,1,500,13,0,16,4
0,0,0,0,0,0,0,0,0,0,0,0,1,500,11,3,23,2
0,0,0,0,0,0,1,640,0,4,0,16,1,500,0,0,5,3
1,77,1,1,0,12,1,644,1,0,0,4,1,500,0,0,2,3
1,77,0,0,0,19,0,0,0,0,0,0,1,500,0,0,2,3
0,0,0,0,0,0,0,0,0,0,0,0,1,500,4,1,13,3
```

Fig. 1. A sample dataset for the regression model

In Fig. 1, each row represents different interaction scores for a post. The last number in each row is the explicit feedback provided by each user and this is also the value of the class for the regression model. We used this dataset to train the MLP-model shown in Fig. 2. After the training, the MLP generated weights for different interactions. For example, we got the following weights for different interactions: twitter share = 0.2, Facebook share = 0.2, twitter reply = 0.49, Facebook comment = 0.49, LinkedIn comment = 0.49 etc. Then, we used these weights in equation (1) to calculate the implicit rating for each post.

3.3 TF-IDF Based Content Similarity Calculation

To find out the similarity between posts, we need to build the content profiles for the posts. To build the content profiles for the posts, we have used two approaches: i) building of content profiles based on the titles of the posts, ii) building of content profiles based on the content of the posts. For the title-based approach, we developed a dictionary with all the words from the titles of all the posts. To remove the unnecessary words such as: a, an, the, etc. we have used a stop word list. A stop word list is a collection of words that are too frequent to be important. Then, we constructed the dataset based on the Inverse document frequency (IDF) values of each word of every title. We have used IDF values because IDF assigns less weight to most words occur in more documents [10,11]. We have used equation (2) to calculate IDF:

$$idf(t) = \log \frac{Total\ number\ of\ posts}{Number\ of\ posts\ with\ term\ t\ in\ it} \tag{2}$$

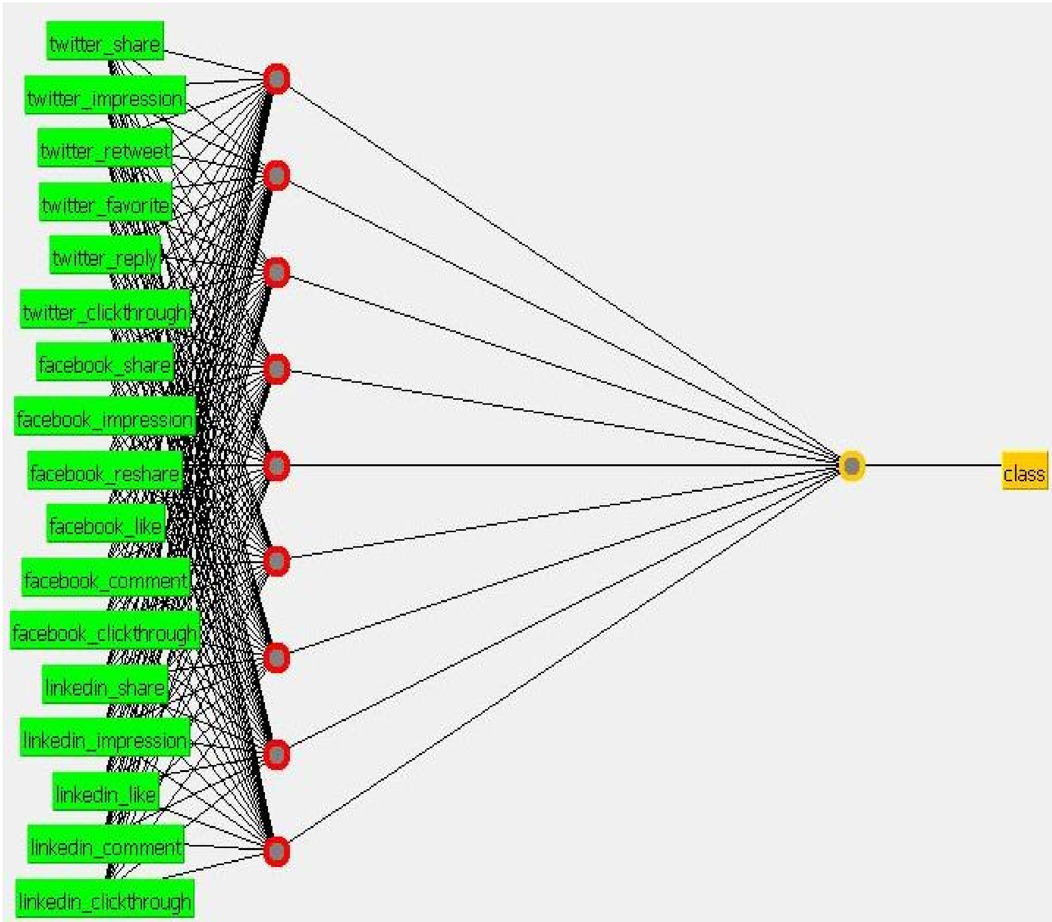

Fig. 2. Use of MLP to calculate implicit feedback

We didn't use the traditional TFIDF weighting scheme because usually a word occurs only once in the title and counting frequency is not that important in this case. Then, we use the IDF scores to build the content profiles for each post.

To build the content profiles based on the content of the posts, we developed a dictionary with all the words from the content of the posts. Again, we removed the stop words using the stop word list. To calculate the term frequency (TF) we calculated how often a term appears in a post and we have used equation (3) to calculate TF:

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ post}{Total\ number\ of\ terms\ in\ the\ post} \quad (3)$$

Then, we calculated the IDF score using equation (2). After the calculation of IDF, we derived TF*IDF for each term and we used the TF*IDF scores to build the content profiles for each post.

To find the similarities between different posts, we calculated cosine similarities for each of the post. To calculate the cosine similarity, we have used equation (4):

$$Sim(A, B) = \cos(\theta) = \frac{A.B}{||A||||B||} \quad (4)$$

3.4 LDA Based Content Similarity Calculation

We have used LDA to extract topics from different posts. We have extracted 15 topics from the posts. For each topic, the LDA defines a bag of keywords. Table 1 shows the keywords for some of the topics.

Then, for each post we calculated the similarity with each topic. As there are 15 topics, so got 15 similarity scores for each topic. For example, for post id: 160 we got the following 15 scores: 0.63, 0.012, 0.0613, 0.013, 0.012, 0.0613, 0.011, 0.012, 0.0312, 0.012, 0.014, 0.010, 0.001, 0.012, and 0.0611. In this way, we built the LDA based content profile for each post. Then, to find the similar posts, we calculated cosine similarities for each of the post.

Table 1. Keywords for different topics derived by the LDA

Topic ID	Keywords
1	Culture, company, startup, building, top, make, ways, corporate, talent, Canadian, article, reason, trust, innovation, google
2	Team, data, big, week, day, CEO, podcast, check, inside, updates, list, back, learn, women, click, meeting, weekend
3	Sales, business, management, strategies, email, lessons, communication, development, effective, sales, people, revenue, challenge
4	Time, community, part, open, make, share, network, live, read, join, support, year, important, event, claim, link, free

3.5 Matrix Factorization

To design a recommender system, matrix factorization is used widely. Matrix factorization (MF) is used to find out the latent features of items and users. MF uses known ratings to predict the unknown ratings. The process of MF is started with a user-item rating matrix R. The size of matrix R is: 'm x n'. Here, 'm' denotes the total number of users and 'n' denotes the total number of items. The MF methods divide the matrix R into two low rank matrices namely P and Q. Here, the size of matrix 'P' is: 'n x d' and the size of matrix 'Q' is: 'm x d'. Here, d is the rank of the matrices and defines dimension for the latent features. The rating matrix R is decomposed into matrices P and Q in such a way, so that:

$$R = PQ^T$$

Once the decomposition is done then matrix P and Q is used to predict the ratings for different items for any user. To predict a rating for a user-u for an item-i, an inner product between P_u and Q_i is done. To apply matrix factorization, different optimization techniques have been identified. To decompose a very sparse rating matrix, the following objective function is used by the traditional matrix factorization methods [12].

$$L = \min_{P,Q} \frac{1}{2} \sum_{(u,i)\in C} \left(R_{u,i} - p_u q_i^T\right)^2 + \frac{\lambda}{2}\left(\|P\|_F^2 + \|Q\|_F^2\right)$$

Here, we see to avoid over fitting, two regularization terms on the sizes of P and Q are added. However, this objective function is only based on users' ratings; it does not include the content similarity information into the model. To include the content similarity information, we have added the following regularization term to the traditional objective function for the positive only feedbacks.

$$\frac{\gamma}{2}\sum_{j=1}^{N} \frac{\beta}{2}\sum_{n=1}^{N} Q_j^T\, Q_n$$

Here, N is the total number of items, γ and β are two regularization parameters whose values are chosen as very small. The objective function adds item similarity information into the MF-model. Our hybrid recommender is developed based on this function.

In the next section, we will discuss the result of the experiment.

4. RESULTS AND DISCUSSION

We have implemented different types of recommender systems. We have designed content-based recommender systems based on just the title of the posts and also based on the content of the posts. To build the content profiles, we have used both LDA [13] and TF-IDF approach. Again, we ran the recommender system based on the user-item rating matrix. Here, we calculated the rating from the implicit feedback based on both the MLP weights and the company assigned weights and we have seen MLP weights out performs the company assigned weights. Because, the MLP weights are generated based on the explicit feedback given by the employees of the company. Again, for factorization, we have applied the traditional MF-model.

We made the system hybrid by combining the TF-IDF or LDA based recommender with the Matrix Factorization based recommender. To test the accuracy of the different types of recommender systems, we calculated the Root Mean Square Error (RMSE) for each system. We have used RMSE because RMSE works by measuring the difference between predicted values and the actual values. To calculate the RMSE, we have used 10-fold cross validation.

From Fig. 3, we see Collaborative Filtering (CF) based recommender system is the worst performer out of all the recommender systems. It is happening because the user-item rating matrix is really sparse. So, there is not enough data to train the model. We can see the best recommender is the content-based title only recommender, it has happened because the employees most of the times put a very detailed title for the posts and the content of the posts are mixed of links and texts. Here, we see TF-IDF based content recommender performs well comparing to the LDA based content recommender. The hybrid recommender also did not perform well comparing to the content-based recommenders but it performed better than the CF-based recommender.

Next, we implemented the new objective function to include the content similarity information to the MF-model. To build the content profiles from the titles of the posts, we have used both LDA and TF-IDF approach. From Fig. 4, we see the hybrid recommenders are performing well comparing to the recommenders from Fig. 3. It is also visible that TF-IDF based recommenders perform better than the LDA based recommenders, because TF-IDF mainly focuses on the intrinsic relationships between the posts and users.

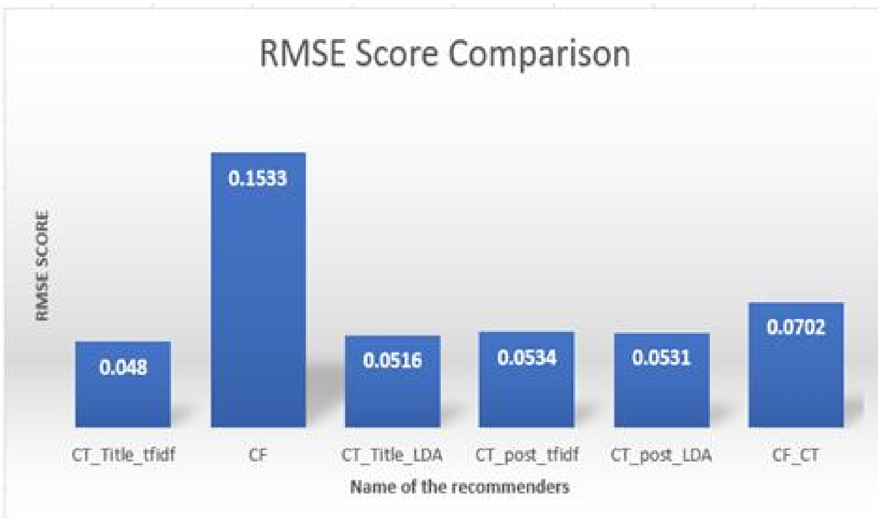

Fig. 3. RMSE comparison for different types of recommender systems using the traditional MF - model

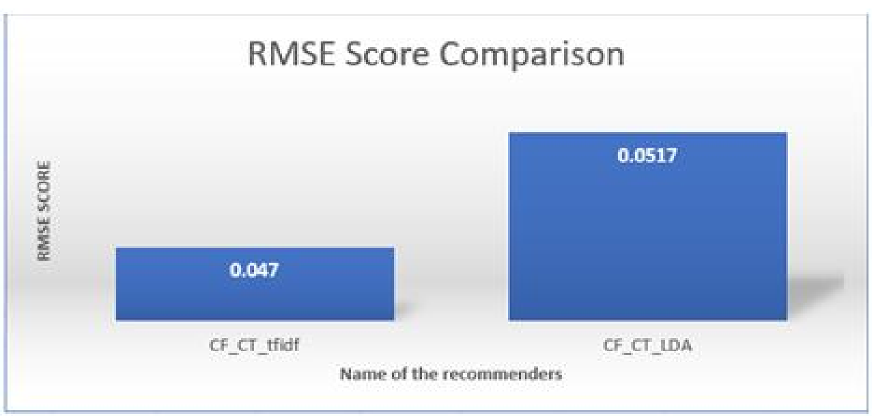

Fig. 4. RMSE comparison for different types of recommender systems using the new objective function

5. CONCLUSION

In this paper, we have used MLP-based training model to generate the weights to calculate the user-item rating from the implicit feedbacks. We have also defined an objective function based on the item-similarity base regularization term to develop a hybrid recommender. We have shown that our MLP and TF-IDF based hybrid recommender works better than the traditional MF-based recommenders. As a future work, we want to extend the hybrid recommender to include a group recommender system. We want to add a new regularization term that will focus on the different contents that are shared, liked, commented by the different group of users.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. Aggarwal CC. Recommender systems: The textbook (1st Ed.). Springer Publishing Company, Incorporated; 2016.
2. Isinkaye FO, Folajimi YO, Ojokoh BA. Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal. 2015;16(3):261-273.
3. Burke R. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction. Springer. 2002;12(4):331-370.
4. Lops P, De Gemmis M, Semeraro G. Recommender systems handbook. Springer. 2011;1-186.
5. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of Machine Learning Research. 2003;993-1022.
6. Zhao G, Liu Y, Zhang W, Wang Y. TFIDF based feature words extraction and Topic Modeling for Short Text. ICMSS. Wuhan, China; 2018.
7. Li F, Xu G, Cao L. Two-level matrix factorization for recommender systems. Neural Computing and Applications. 2016; 27(8):2267-2278.
8. Tang J, Gao H, Hu X. and Liu, H. Exploiting homophily effect for trust prediction. In the Proceedings of the Sixth ACM International Conference on Web Search and Data Mining. 2013;53-62.
9. Yu X, Ren X, Sun Y, Sturt B, Khandelwal U, Gu Q, Norick B, Han J. Recommendation in heterogeneous information networks with implicit user feedback. In Proceedings of the 7th ACM Conference on Recommender Systems. 2013;347-350.
10. Manning CD, Raghavan P, Schütze H. Introduction to information retrieval. Cambridge University Press. 2008;39:109-133.
11. Wei J, He J, Chen K, Zhou Y, Tang Z. Collaborative filtering and deep learning-based recommendation system for cold start items. Expert Systems with Applications. 2017;69:29-39.
12. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. IEEE Computer. 2009;42(8):30-37.
13. Cao Y, Li W, Zheng D. A hybrid recommendation approach using LDA and probabilistic matrix factorization. Cluster Computing. 2019;22(4):8811-8821.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle4.com/review-history/54759