Journal of Engineering and Applied Sciences 12 (Special Issue 3): 6487-6491, 2017
ISSN: 1816-949X
© Medwell Journals, 2017

Leveraging Personalized PageRank with Dynamic Initial Rank for Recommendation System Advances

[1]Hayder M. Naji and [2]Ghaidaa A. Al-Sultany
[1]Department of Software,
[2]Department of Information Network, Babylon University, Babil, Iraq

Abstract: Basically, the PageRank algorithms set the inputted items with initial ranks. They usually use a normal distribution of initial rank in the item's space with the ratio of 1/n for each item where n is the number of items in the dataset. Alternatively, a dynamic distribution of initial ranks according to the user preferences and item's features has been proposed for recommendation system improving. The adapted PageRank algorithm was evaluated in comparis on with the traditional algorithm. The results were extremely encouraging with respect to the recommendation system improvement. The evaluation measures ofthe suggested algorithm were examined on the MovieLens dataset.

Key words: Recommendation system, personalized PageRank, initial rank, improvement, algorithm, distribution

INTRODUCTION

Recommendation system is an information filtering system. The goal of a recommendation system is to discover irrelevant information and provided the users with relevant information corresponding to his/her personal preferences.

Recommendation systems predict recommendations using three basic steps: they obtain preferences from the user's raw data, compute recommendations using certain techniques and output a list of recommendations.

PageRank is Google's method of measuring the significance of the webpages. When all other factors such as title tag and keywords are taken into account, Google uses PageRank to adjust the results, so that, the sites with more vitalare moved up as the top resulted pages of a user's search accordingly. PageRank algorithm is convenient method for recommendation task. We used the PageRank algorithm to improve the representation of movies and users in our graph network in addition to the ranking movies for a target user in the recommendation activity. Additional to personalize the target user, we present a way to personalize each user with different weight according to them rating on movies.

Literature review: Full personalization capability is one of the most desired properties for any PageRank computation (Bahmani *et al*., 2010). The most popular of personalized recommendation methods based on PageRank doing the personalization activity in the users layer while the items space have a normal distribution with an equal initial ranks for items and there is no special process on items that is help to personalize the results. We analyze the content of items and personalize the items by set a different initial ranks which are produce a more personalized suggested items.

PageRank, also is the static distribution of a random walk which at each step with a certain probability ε jumps toa random node and with probability (1-ε) follows a randomly selected outgoing edge from the current node. Personalized PageRank is the same as PageRank, except that all the jumps are set to the target node for which we are personalizing the PageRanks (Bahmani *et al*., 2010). To a certain page i, we can calculate the pagerank value π (i) for it as follow:

$$\pi(i) = \begin{cases} (1-\varepsilon) \sum_{j:j \text{ links to } i} \frac{\pi(i)}{|out(j)|} & \text{if } i \neq x \\ \varepsilon + (1-\varepsilon) \sum_{j:j \text{ links to } i} \frac{\pi(i)}{|out(j)|} & \text{if } i = x \end{cases} \quad (1)$$

where x is the target node with probability ε jump to x, probability (1-ε) click on a random hyperlink within page i. In the hubs and authorities setting (Bahmani *et al*., 2010; Goel, 2009ab) present the following (Algorithm 1):

Algorithm 1; Iterative PageRank:
Algorithm which are initial with static ratio of item PageRank

Initial:

$$h_w^{(0)} = \frac{1}{M}, t = 0$$

Repeat:

Corresponding Author: Hayder M. Naji, Department of Software, Babylon University, Babil, Iraq

$$t \leftarrow t+1$$

$$a_{\upsilon}^{(t)} = \sum_{\omega \in Movies} A_{(\omega,\upsilon)} h_{\omega}^{(t-1)}$$

$$h_{\upsilon}^{(t)} = \sum_{\omega \in Movies} A_{(\omega,\upsilon)} a_{\omega}^{(t)} \text{Normalize } h_{w}^{(t)} \text{ to sum to 1}$$

Normalizing the PageRank values for each type entities are very important, according to the empirical result of (Bahmani *et al*., 2010), suggest that personalized PageRank with normalizedterms over-performs other methods while personalized PageRank without normalizing terms performs rather poorly.

G. Marco, present an "Item Rank", a random-walk based ranking algorithm (Gori and Pucci, 2007) which can be used to rank products according to expected user preferences he collect user preferences through the correlation graph they computed the item rank, iteratively in this way:

$$\begin{cases} IR_{ui}(0) = \dfrac{1}{|M|} \times 1_{|M|} \\ IR_{ui}(t+1)\ \alpha \times C \times IR_{ui}(t) + (1-\alpha) \times d_{ui} \end{cases} \quad (2)$$

They found user preferences according to his/her historical ratings and stay to use the static initial rank 1/M.

MATERIALS AND METHODS

The accuracy of the system that predicting user's preferences can be improved by modeling and acquiring user preferences, i.e., by letting the users with rates to indicate what does he think the most important features of the movies?

Item's features can lead to know the taste of users, therefore in this research, the user preferences was initiated with respect to the item's features. They are used to be as a part of structuring a user profile from both the past history of user rating andthe item's features that are preferred by a user using the way of vector approach. For each user we generate a vector of the length of selected feature values, each one denote to a certain type of the selected feature and the value represent how this type is preferred to the user and these values are sum to one.

If we select an item features to analyze the content of the items, the selected feature will haven types. Because of each item can has one or more types of the selected feature so, we assigned a boolean vector F_n to each item to hold the types of the selected feature for that item where n is the number of types of the selected feature.

Movies have more than one feature such as genre, actors, country, etc., the most effective feature in the research is the movie genre. Each movie m will be having a Boolean vector g with length of n where n is the number of movie genre types. g^m_i is 1 if the movie m has a type of ith genre, otherwise is 0. Figure 1 is an example for a movie genre feature.

As a part of user preferences we calculate user genres preferences from his past watched moviesthat were rated/tagged by himself. Then, the numbers of movies that belong to each type genre are counted and hold it in g^u vector as following:

$$\forall \{i, mv : i \in n, mv \in m\}\ g_i^u = \sum_{mv=1,}^{m} g_i^{mv}, \text{If } g_i^{mv} = 1 \quad (3)$$

After normalizing the g^u vector to be sum to one as shown in Fig. 2, the user preferences are aligned with the percentage of the movies genres that the user mostly likes them as presented in Algorithm 2:

Algorithm 2; Calculating genre preferences to the user:

```
calc_genr_pref()–void method
begin
    sum←0
    for each movie (mv) rated by the user
        for (i = 0→19)
            if (mv.genre(i) is true)
                user.genre(i)+1
                sum+1
            end if
        end for
    end for
    for (i = 0→19)
        divide user.genre(i) by sum
end begin
```

Mainly, the PageRank algorithms use a normal distribution of initial rank in item's space. It sets the ratio of 1/n for each item where n is the number of items in the dataset (Goel, 2009). Goel (2009ab) supposed the value 1/m for each movie in the initial iteration where m is the number of movies. Instead of that, we have used a dynamic distribution of initial ranks according to the user preferences and item's features as explained in the example of Fig. 3. Initial rank for a movie resulted by sum of user preferences values that corresponding with movie genres as following:

$$\pi_{u,m} = \sum_{i=1,}^{n} g_i^u, \text{If } g_i^m = 1 \quad (4)$$

where, $\pi_{u,m}$ is the initial rank of the movie m to the current user u.

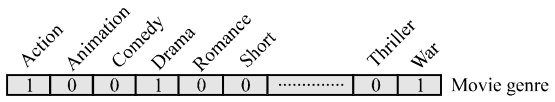

Fig. 1: Example for a movie genre vector

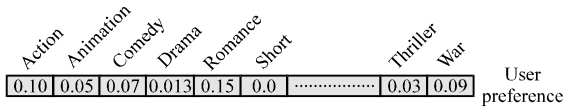

Fig. 2: Example of user preferences vector

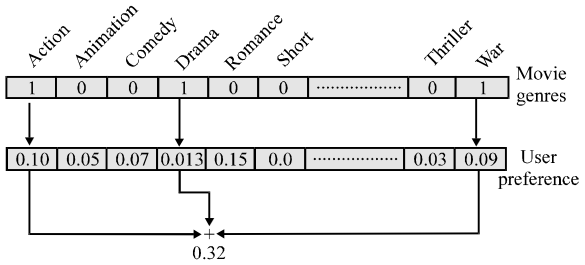

Fig. 3: Emple of calculation the initial rank for the movie

In the initial iteration of personalized PageRank algorithm, dynamic initial ranks have produced a personalization for the items according to their features. The dynamic initial ranks have supported the essential goal of recommendation task in which the user is suggested with the specifieditems based on his/her preferences.

RESULTS AND DISCUSSION

To evaluate the recommendation system's performance and compare it with the others we used precision, recall and F-measure metrics. To measure the accuracy ofour recommendation system we calculate hit counts of top-N named hitCount which are the number of recommended movies by system and they are already watched by the user in the test data. Movies are recommended to the user according to the movie's PageRank values resulted by the ranking stage. Movies aresorted according to their generated ranks and recommend the top-N movies tothe target user (Uluyagmur *et al*., 2012). Precision is the measure of howmany testing movies the system result it in the top-N movies:

$$\text{Precision} = \frac{\text{hitCount}}{N} \tag{5}$$

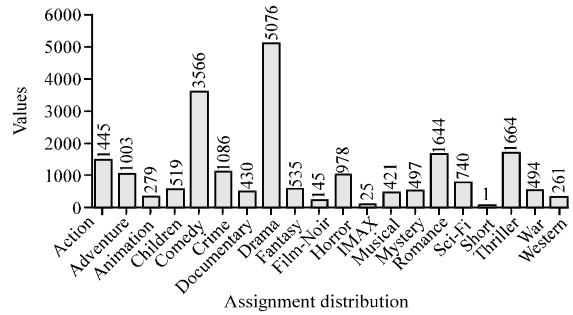

Fig. 4: Movies genre assignment distribution

Table 1: General statistics of MovieLens 2K dataset

Parameters	Values
Number of movies	10.197
Number of users	2.113
Number of movie genres	20
Number of movie genre assignments	20.809
Average genre assignments per movie	2.040
Number of ratings	855.598
Average ratings per user	404.921
Average ratings per movie	84.637

Recall is the measure of how many testing movies have a recommended movies resulted by the system:

$$\text{Recall} = \frac{\text{hitCount}}{U_{test}} \tag{6}$$

F-measure can be interpreted as a weighted average of the precision and recall as follows:

$$\text{F-measure} = \frac{2\times\text{hitCount}}{N+U_{test}} \tag{7}$$

For example, for N = 10 and the user have 20 movie in the test data so, if the recommendation system result 5 movies from the user test data among the top 10 then precision will be 0.5, recall will be 0.25 and F-measure will be 0.33.

We used MovieLens 2K dataset to evaluate the recommendation system. Table 1 showing some statistical information about the dataset.

The challenge of this dataset has a big item space in comparison with another datasets, represented by 10.197 movies that are rated by 2.113 users. So, we analyze the content of movies and select the feature of movie genre to test our method. Figure 4 show the distribution of 20 genres for the movies in the dataset.

We represented movies genres in a boolean vectors approach as displayed in Table 2 where columns are the movie genre (g_1, g_2 ..,g_{20}) while rows are the movies (M_1, M_2,..., $M_{10,197}$).

Table 2: Example of movie genre presentation in the system

Parameters	g_1 Action	g_2 Adventure	g_3 Animation	g_4 Children	g_5 Comedy	g_6 Crime	g_7 Documentary	g_8 Drama	g_9 Fantasy	g_{10} Film-Noir	g_{11} Horror	g_{12} IMAX	g_{13} Musical	g_{14} Mystery	g_{15} Romance	g_{16} Sci-Fi	g_{17} Short	g_{18} Thriller	g_{19} War	g_{20} Western
M_1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
M_2	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
M_3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
M_4	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
M_5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M_6	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
M_7	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
M_8	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M_9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M_{10}	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
M_{11}	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
M_{12}	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Table 3: Sample of user's preferences of movies genres

Users	g_1 (%)	g_2 (%)	g_3 (%)	g_4 (%)	g_5 (%)	g_6 (%)	g_7 (%)	g_8 (%)	g_9 (%)	g_{10} (%)	g_{11} (%)	g_{12} (%)	g_{13} (%)	g_{14} (%)	g_{15} (%)	g_{16} (%)	g_{17} (%)	g_{18} (%)	g_{19} (%)	g_{20} (%)
75	23.30	10.80	1.14	1.14	8.52	7.39	0.00	9.66	5.68	0.57	1.14	0.00	0.57	1.70	2.27	9.09	0.00	14.20	2.27	0.57
78	8.11	6.93	1.44	1.27	17.06	4.56	0.93	21.03	2.79	0.76	2.36	0.00	1.01	2.20	6.08	9.97	0.00	8.11	4.56	0.84
127	10.28	5.61	0.93	4.67	20.56	3.74	1.87	7.48	6.54	0.00	10.28	0.93	0.93	2.80	5.61	3.74	0.00	13.08	0.93	0.00
170	10.57	8.37	3.08	3.08	21.59	7.05	0.44	11.45	4.41	0.00	1.32	0.00	2.20	3.08	6.17	4.41	0.00	10.57	1.32	0.88
175	11.02	7.85	0.96	1.24	10.61	8.13	0.55	21.21	2.62	2.48	1.65	0.00	0.83	3.44	6.89	4.41	0.00	11.43	3.31	1.38
190	14.01	14.75	5.27	4.85	9.48	2.42	0.21	8.85	10.54	0.32	2.42	0.32	1.05	1.79	3.37	10.96	0.00	7.59	1.37	0.42
267	15.38	10.33	0.90	0.81	9.60	6.10	0.16	10.81	3.82	0.57	4.07	0.00	0.81	3.17	4.96	9.03	0.00	11.88	2.60	0.98
325	11.43	9.23	3.58	5.23	21.63	6.06	0.14	11.58	3.58	0.28	1.65	0.14	3.17	1.52	6.75	4.13	0.00	7.16	1.24	1.24
383	13.27	10.19	1.23	2.47	2.96	6.17	0.00	17.59	5.56	0.93	1.23	0.00	0.62	2.47	8.02	7.10	0.00	8.64	1.23	0.31
476	12.12	9.23	2.79	2.12	11.06	6.44	0.19	16.92	4.13	0.87	2.12	0.00	0.96	3.75	5.87	7.69	0.00	11.06	2.40	0.29
477	15.05	10.44	1.94	2.67	6.31	8.01	0.24	15.53	4.61	0.97	3.40	0.24	0.73	2.67	3.16	7.04	0.00	11.89	4.13	0.97
493	6.22	5.35	0.87	1.48	16.62	6.01	0.15	26.30	3.41	0.56	1.33	0.05	1.17	2.91	13.46	2.75	0.00	9.02	2.04	0.31

Table 4: Sample of initial ranks of movies for the users

Users	M_1	M2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}	M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}	M_{25}	M_{26}
U_1	0.27	0.18	0.11	0.20	0.09	0.45	0.11	0.12	0.23	0.48	0.20	0.10	0.02	0.10	0.36	0.17	0.20	0.32	0.09	0.63	0.41	0.34	0.45	0.19	0.12	0.10
U_2	0.29	0.11	0.23	0.44	0.17	0.21	0.23	0.08	0.08	0.23	0.44	0.19	0.03	0.21	0.21	0.26	0.44	0.46	0.17	0.59	0.46	0.38	0.21	0.31	0.27	0.21
U_3	0.38	0.17	0.26	0.34	0.21	0.27	0.26	0.10	0.10	0.34	0.31	0.06	0.06	0.07	0.21	0.11	0.34	0.41	0.21	0.55	0.38	0.37	0.27	0.11	0.13	0.07
U_4	0.41	0.16	0.28	0.39	0.22	0.28	0.28	0.11	0.11	0.30	0.39	0.23	0.06	0.11	0.25	0.19	0.39	0.44	0.22	0.61	0.44	0.33	0.28	0.16	0.18	0.11
U_5	0.23	0.12	0.17	0.39	0.11	0.31	0.17	0.09	0.11	0.30	0.39	0.12	0.02	0.21	0.26	0.29	0.39	0.43	0.11	0.62	0.43	0.46	0.31	0.26	0.28	0.21
U_6	0.45	0.30	0.13	0.22	0.09	0.24	0.13	0.20	0.14	0.36	0.22	0.12	0.10	0.09	0.32	0.11	0.22	0.26	0.09	0.42	0.32	0.23	0.24	0.20	0.12	0.09
U_7	0.25	0.15	0.15	0.29	0.10	0.33	0.15	0.11	0.15	0.38	0.29	0.14	0.02	0.15	0.31	0.21	0.29	0.36	0.10	0.58	0.40	0.40	0.33	0.24	0.20	0.15
U_8	0.43	0.18	0.28	0.40	0.22	0.25	0.28	0.14	0.11	0.28	0.40	0.23	0.09	0.12	0.27	0.18	0.40	0.41	0.22	0.58	0.45	0.28	0.25	0.16	0.19	0.12
U_9	0.32	0.18	0.21	0.39	0.13	0.28	0.21	0.13	0.13	0.32	0.39	0.14	0.04	0.18	0.31	0.24	0.39	0.39	0.13	0.59	0.44	0.36	0.28	0.25	0.26	0.18

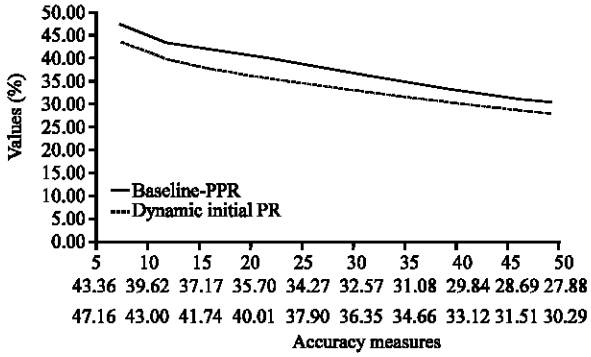

Fig. 5: Comparison of systems using precision accuracy measure

We used movie's genre vectors and historical user ratings to compute user preferences as genres preferences whichit is essential process for any content-based recommender system. Genre preferences are required to result the movie's initial rank in. Table 3 shows a sample of the resulted user preferences.

Initial rank of movies are computed based on user preferences and movie genre instead of equal assignment for each movie as we sampled in Table 4, each row represent a user (U_1, U_2, ..., $U_{2,113}$) and each column represent a movie (M_1, M_2, ..., $M_{10,197}$).

The same item (i.e., movie) can be getting from different initial rank values with different users according to their preferences. Each user needs a recommendations, the system will maintain the items ranking to suggest the user with the most preferred items from his view point. Initial ranks of items will be different according his/her preferences.

We evaluated the performance of our system on 500 users and splitting the dataset into 2:1, i.e., 66% for training and 33% for testing. The system was evaluated with different values of N for each evaluation measure so, we tested it 10 times with = {5, 10, …, 50}, each time the system recommended top- N movies for each user and the values of accuracy measures was calculated. Then the total accuracy measure from the average of user's accuracy values were computed and compared with the baseline recommender system as shown in Fig. 5-7 with the three evaluation measures.

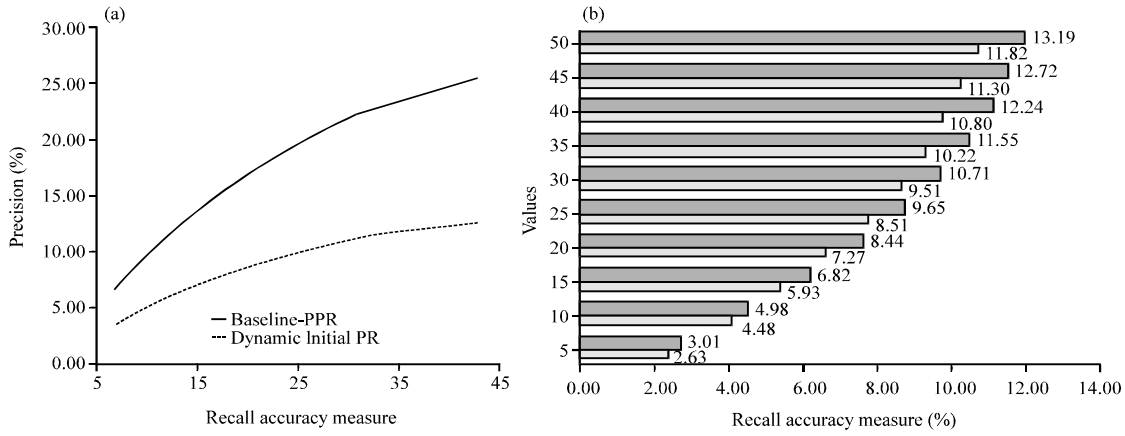

Fig. 6: Comparison of systems using recall accuracy measure

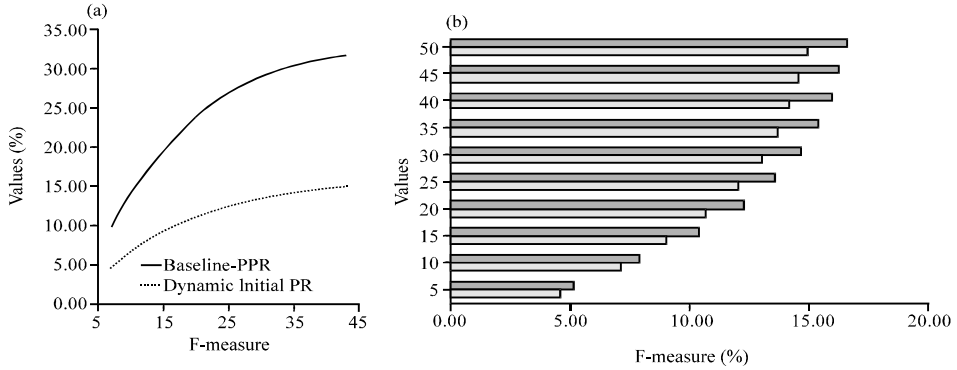

Fig. 7: Comparison of systems using F-measure

CONCLUSION

In this study, a dynamic distribution of initial ranks according to the user preferences and item's features has been proposed for recommendation system improving. The predicted out comes achieved a worthy results with respect to the evaluation measures on thetested dataset with comparison to the traditional recommendation system. The current research tries to combine more than one feature for the selected items to contribute in computing the initial rank with different weights.

REFERENCES

Bahmani, B., A. Chowdhury and A. Goel, 2010. Fast incremental and personalized PageRank. Proc. VLDB Endowment, 4: 173-184.

Goel, A., 2009a. More recommendation systems (Lectures). Stanford University Computer Science, California, USA.

Goel, A., 2009b. Reputation systems and page rank (Lectures). Stanford University Computer Science, California, USA.

Gori, M. and A. Pucci, 2007. ItemRank: A random-walk based scoring algorithm for recommender engines. Proceedings of the 20th International Joint Conference on Artificial Intelligences, Jan. 6-12, Hyderabd, India, pp: 2766-2771.

Uluyagmur, M., Z. Cataltepe and E. Tayfur, 2012. Content-based movie recommendation using different feature sets. Proceedings of the World Congress on Engineering and Computer Science Vol. 1, October 24-26, 2012, WCECS, San Francisco, USA., ISBN:978-988-19251-6-9, pp: 17-24.