Volume-6 | Issue-11 | November-2017 | ISSN No 2277 - 8179 | IF : 4.176 | IC Value : 78.46

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

IMPROVING NON PERSONALIZED RECOMMENDATIONS USING A NON-LINEAR WEIGHTED MEAN

Computer Science

Shitij Goyal

Grofers

ABSTRACT

Recommender Systems have become an important part of our day to day life. The goal of any recommendation system is to present users with a relevant set of items which would interest them. This paper showcases a new technique and implements a non-personalized recommender system using the proposed technique. It is shown how the modification can be used to improve the recommendation as compared to existing algorithms. The comparison is done with the widespread method of average ratings and conclusions are drawn based on these tests.

KEYWORDS

Recommendations; Ratings; Non-Personalized recommender system

Improving Non Personalized Recommendations using a Non-Linear Weighted Mean

There has been an information boom in the field of internet. Be it books, games, movies, TV shows, it has become very difficult for a user to choose one item out of the multitudes of choices. This is where a Recommender System comes to the rescue.

Recommender Systems (RSs) are the software techniques that provide propositions for items to be used by a user. Media companies have developed and deployed RSs as part of their services that they provide to the subscribers, such as Amazon (Linden, York, & Smith, 2003), YouTube (Davidson, et al., 2010), Netflix, Yahoo, IMDb, etc. along with other companies. The paper aims to build one such system and implements a technique for tackling the problems that are faced by such systems nowadays.

Content based recommenders are a great set of algorithms to recommend items and perform extremely well when the parameters of the item to be recommended are well defined, structured and limited. A good example would be the Pandora recommendation system (Howe, 2003). Techniques like Clustering (Jajoo, 2008) and Singular Value decomposition (Osinski, J.S, & Weiss, 2004) are used to reduce dimensions by selecting features based on some metric. These are sometimes used to filter the recommendations obtained after running collaborative algorithms based on some feature (Billsus & Pazzani, 2007). The best results are obtained when content-based recommenders are used along with collaborative recommenders to balance out the strengths and weaknesses (Campos, 2010).

A personalized recommender system entails recommending items based on the past history of the user and their similarity with other users. The opinions of users can be obtained explicitly from the users or by using implicit methods. Many modern systems use implicit models by recording clicks or by noting the time taken to read an article. A good example of such implicit behavior is Amazon's system (Linden, York, & Smith, 2003) or Google News (Liu, P.D, & Pederson, 2010), which shows news based on click history of a user. The main techniques used generally are User-based Collaborative filtering and Item-based collaborative filtering (Bogers & Bosch, 2009), (Sarwar, G.K, Konstan, & Riedl, 2001).

Personalized recommender systems are the most useful, being able to predict things suitable to a particular person's tastes but they require a lot of data beforehand, the condition commonly known as cold-start. Content-based recommenders may give more accurate results in situations when the attributes of an item are well known and not subject to change. These systems perform quite well when user context and data of user's interaction is available. Quite often due to security constraints or other reasons, it is not possible to gather data on a user level. This is where non personalized recommendations shine. Non - personalized recommender systems are the simplest of all recommender systems. These type of recommender systems do not take into account the individual preferences of a user. This can be done by manually selecting some popular items or by choosing the Top- N recommendations based on ratings given by users (Poriya, N.P, Bhagat, & Sharma, 2014). There are mainly two types of rating systems employed. The most common rating system employed is the mean of total rating by the total no. of users known as the average rating. Second is selecting the Top-N items based on items which have received the most number of ratings. This makes for an easy system to rate items but has a major drawback that less rated items never come out on top and there is no scope for surprise discovery as the items with more ratings are always shown on top. These are some issues which the proposed technique tries to alleviate.

METHOD

The theory behind the modification is to dampen the noise generated by the outlying ratings found in a User's vs Ratings graph. The final aggregated rating should be reflected by taking into account all the ratings albeit with less importance given to those ratings which have been given by less number of users. The average rating formula uses this theory by incorporating a linear weight function for each rating. This weight function is modified into a non-linear weight function to give more or less emphasis to a set of ratings depending upon the number of users.

$$K_x = \log_{10}\left(1 - \frac{U_x}{U}\right) \qquad (1)$$

Ux = Number of users who have given X rating
U = Total number of users

$$\text{New Rating} = \frac{\sum_x^n R_x K_x}{\sum_x^n K_x} \qquad (2)$$

R_x = Some rating given by Ux users. This is between 0 and 5 for our purposes with increments of 0.5

The new rating is the sum of each rating multiplied by the corresponding weight function. This is normalized by the sum of all the weight functions. Because the weight function is a non-linear function using the logarithmic scale, it penalizes the rating given by few users while the change is slight if U_x and U_y are not far apart. In cases where U_x equals U, the rating can be calculated directly without the need of the formula due to the absence of any sort of noise.

The testing methodology as described in (Cremonski, Koren, & Turrin, 2010) and (Koren, 2008) was followed to evaluate the algorithm. For testing purposes, MovieLens dataset of 10 million records was used. The dataset is split into two parts: probe set (P) and training set (T). The probe set contains the test set (Q) which consists of only 5-star movie ratings. Hence it can be reasonably assumed that Q contains items relevant to the users.

The probe set (P) contains 2% of the ratings from the dataset. The test set consists of all the 5-star ratings within the probe set. This gives a reasonably large testing set for evaluation. The training set contains all the ratings minus the probe set. The performance of the proposed technique is evaluated by comparing its recall and precision with that of the existing average rating algorithm. Following is the procedure used for evaluation:

In order to measure recall and precision, the model is trained over the ratings in M. Then, for each item i rated 5-stars by user u in Q:

(i) Randomly select 1000 additional items unrated by user u. It may be assumed that a lot of them will not be of interest to the user.

(ii) Predict the ratings for the test item i and for the additional 1000 items.

(iii) Form a ranked list by ordering all the 1001 items according to their predicted ratings. Let p denote the rank of the test item i within this list. The best result corresponds to the case where the test item i precedes all the random items (i.e. p = 1).

(iv) Form a top -N recommendation list by picking the N top ranked items from the list. If p ≤ N it's a hit (i.e., the test item i is recommended to the user). Otherwise, it's a miss. Chances of hit increase with N. When N= 1001 there is always a hit.

Recall is defined as the sensitivity or the number of relevant items among the calculated results. Thus it can be either a miss or a hit depending on whether the item i features in our top -N list. Precision is the number of hits with respect to the retrieved results. Thus precision and recall can be given by: -

$$\text{Recall} = \frac{(\text{No.of Hits})}{(\text{No.of Test cases(Q)})} \quad (3)$$

$$\text{Precision} = \frac{\text{Recall}}{\text{N}} \quad (4)$$

High scores for both precision and recall show that the system is returning accurate results (high precision), as well as returning a majority of all positive results (high recall). Thus precision is a standard to calculate the usefulness of the results while recall is a standard to calculate the completeness of the results.

RESULTS

|Q| = 30767
|T| = 9800053
|P| = 200001

The tests were done on different probe sets and training sets obtained randomly from the data set. The final results were obtained using the mean of all the readings. High consistency was seen in the results across all sets of data.

Table 1. Comparison of Recall and Precision

N	Average-Recall	Average-Precision	Non Linear Recall	Non Linear Precision
10	0.13006	0.013006	0.20724	0.020724
20	0.22855	0.011427	0.30491	0.015245
30	0.28678	0.009559	0.37855	0.012618
40	0.33511	0.008377	0.43868	0.010967
50	0.37537	0.007507	0.49019	0.009803

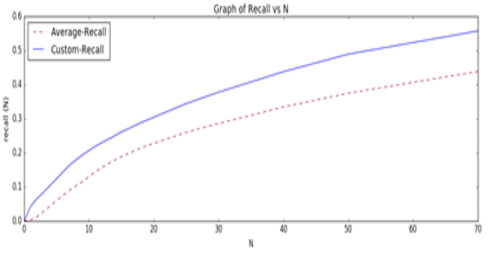

Fig. 1. MovieLens: recall-at-N

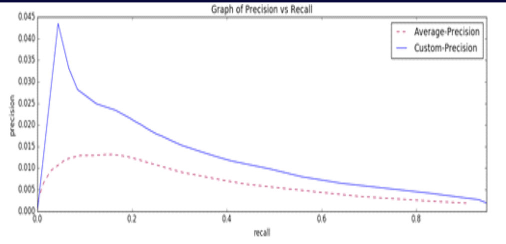

Fig. . MovieLens: precision-vs-recall

DISCUSSION

The recall values obtained by using the weight term is consistently higher than the recall value obtained from the average rating algorithm. The difference is approximately 0.06 for N = 5 and goes on increasing till N = 200. After that, the difference begins to taper off due to the fact that ratings matter less when the value of N is sufficiently larger as more items enter the list to be recommended. The maximum difference was found out to be 0.15. The median percentage increase in recall and precision, between N= 5 to N = 100 is 32% while the mean percentage increase is 38%. The percentage change is 73% when N=5 and goes on decreasing owing to the increasing value of average recall, reaching to 23% at N = 100. Thus, significantly higher relevant results are obtained when N is smaller. At higher values of N more number of irrelevant results enter the desired recommendations, increasing the noise. The precision value is also higher in the modified algorithm than in the average rating algorithm. It is markedly more for recall values up to 0.3 and then the difference gradually begins to decrease as seen in Fig. 2.

The higher precision values at lower recall values are significant from a practical viewpoint because lower value of recall means lower value of N. Top-10 and Top-15 lists are more common than Top-100 lists from a pragmatic perspective. Higher precision in lower N means more relevant results in the recommended records shown.

Non-Personalized systems come free of the baggage of being memory intensive or using complex algorithms like classifiers or probabilistic models. They are useful in situations which warrant recommendations where user information is not available and they have been known to give decently accurate results but a big drawback of this kind of system is the state of being repetitive and lack of relevant results to all users. The paper tries to improve upon this and shows an algorithm which would increase the relevant items shown by 30-35% when the number of items to be recommended is between 5 and 100, which is a significant number when we think about the amount of data available in this age of information. The non-linear weighted term is easy to understand and implement. The formula of averages has been modified to incorporate weights not in a linear fashion but as a non-linear function of the number of users who have given ratings. This has the effect of reducing the unwanted noise and the effect of outliers. Testing has been done on a large dataset and care has been taken to ensure accurate results.

Non-Personalized recommenders can serve as baseline models for personalized models. These recommenders could be further improved by using a more complex algorithm which gives individual consideration to each rating point. Bayesian modelling (Miller, 2012) may be a path which could be taken to further improve non-personalized recommendations.

REFERENCES

[1] Linden, G., York, J., & Smith, B. (n.d.). Amazon.com Recommendations. Item-to-Item Collaborative Filtering. Amazon. IEEE Computer Society.
[2] Davidson, J., Liebald, B., Liu, J., Nandy, P., Vleet, T. V., Gargi, U., Sampath, D. (2010). The YouTube video recommendation system. Proceedings of the fourth ACM conference on Recommender systems. Barcelona.
[3] Howe, M. (2003). Pandora's Music Recommender.
[4] Osinski, l. S., J.S, & Weiss, D. (2004). Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. In W. S. Kłopotek M.A., Intelligent Information Processing and Web Mining. Advances in Soft Computing (Vol. 25). Berlin: Springer.
[5] Jajoo, P. (2008). Document Clustering. Thesis, Indian Institute of Technology , Department of Computer Science & Engineering , Kharagpur.
[6] Billsus, D., & Pazzani, M. J. (2007). Content-based Recommendation Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl, The Adaptive Web. Lecture Notes in Computer Science (Vol. 4321, pp. 325-341). Berlin: Springer.
[7] Campos, L. d. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. International Journal of Approximate Reasoning, 785-799.
[8] Liu, J., P.D, & Pederson, E. R. (2010). Personalized News Recommendation Based on

Click Behavior. Proceedings of the 15th international conference on Intelligent user interfaces. Hong Kong.
[9] Sarwar, B., G.K, Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th international conference on World Wide Web, (pp. 285-295). Hong Kong.
[10] Bogers, T., & Bosch, A. v. (2009, January). Collaborative and Content-based Filtering for Item Recommendation on Social Bookmarking Websites. Proceedings of the ACM RecSys '09 workshop on Recommender Systems and the Social Web.
[11] Poriya, A., N.P, Bhagat, T., & Sharma, R. (2014). Non-Personalized Recommender Systems and User- based Collaborative Recommender Systems. International Journal of Applied Information Systems .
[12] Cremonski, P., Koren, Y., & Turrin, R. (2010). Performance of Recommender Algorithms on Top-N Recommendation Tasks. Proceeding RecSys '10 Proceedings of the fourth ACM conference on Recommender systems, (pp. 39-46).
[13] Koren, Y. (2008). Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, (pp. 426-434).
[14] Miller, E. (2012). Bayesian Average Ratings. Retrieved from evanmiller.com: http://www.evanmiller.com/bayesian-average-ratings.html