How Similar is Rating Similarity to Content Similarity?

Osman Başkaya
Department of Computer Engineering
Bahçeşehir University
İstanbul, Turkey
osman.baskaya@computer.org

Tevfik Aytekin
Department of Computer Engineering
Bahçeşehir University
İstanbul, Turkey
tevfik.aytekin@bahcesehir.edu.tr

ABSTRACT

The success of a recommendation algorithm is typically measured by its ability to predict rating values of items. Although accuracy in rating value prediction is an important property of a recommendation algorithm there are other properties of recommendation algorithms which are important for user satisfaction. One such property is the diversity of recommendations. It has been recognized that being able to recommend a diverse set of items plays an important role in user satisfaction. One convenient approach for diversification is to use the rating patterns of items. However, in what sense the resulting lists will be diversified is not clear. In order to assess this we explore the relationship between rating similarity and content similarity of items. We discuss the experimental results and the possible implications of our findings.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation, Measurement

Keywords

diversity, recommender systems, collaborative filtering

1. INTRODUCTION

Recommender systems help users to pick items of interest based on explicit or implicit information that users provide to the system. One of the most successful and widely used technique in recommender systems is called collaborative filtering (CF) [7]. CF algorithms try to predict the ratings of a user based on the ratings of that user and the ratings of other users in the system. The performance of collaborative filtering algorithms is typically measured by the error they make in predicting the ratings of users for items. Although accuracy of predictions is an important aspect of recommender systems, it is not the only one. Recently, increasing the diversity of recommendation lists have gained attention among researchers in the field [8, 2]. To be able to recommend a diverse set of items to a user is important with respect to user satisfiability because a recommendation list consisting of one type of item (e.g., movies only from the same genre) might not be very satisfactory even if the accuracy of rating prediction is high. But here there is one issue. We need to define a metric for measuring the diversity of a recommendation list. Then we can try to optimize the recommendation list based on this metric. One possible metric for measuring the diversity of a recommendation list of a particular user is described in [2]. This metric measures the diversity as the average dissimilarity of all pairs of items in a user's recommendation list. Formally, it can be defined as follows:

$$D(R) = \frac{1}{N(N-1)} \sum_{i \in R} \sum_{j \in R, j \neq i} d(i,j), \qquad (1)$$

where R is the recommendation list of a user and $N = |R|$. $d(i,j)$ is the dissimilarity of items i and j which is defined as one minus the similarity of items i and j.

We think that average dissimilarity is a reasonable way to measure the diversity of a list of items. However, the important part is how to define $d(i,j)$, i.e., the dissimilarity of two items which is unspecified in equation (1). The problem is not to choose a similarity metric such as Pearson or cosine. The problem is whether we can use the rating patterns (vectors) of items in order to measure their similarity. And if we use these rating patterns, in what respect the recommendation lists will be diversified? For example, if it is a movie recommender system, will the recommendation lists contain more movies from different genres or will the content of the movies get diversified?

In order to answer these questions we will compare rating similarity with two types of content similarities which we will define below. We hope that the results we discuss will shed some light on these types of questions and stimulate discussion on diversification.

2. RELATED WORKS

In hybrid recommendations content information is used in order to increase the accuracy of rating predictions especially for items whose ratings are too sparse. For example [3, 5, 6] use content information collected from sources such as

Wikipedia and IMDB in order to improve the accuracy of rating predictions. These works indirectly show that there is indeed some positive relationship between rating similarity and content similarity. Otherwise, it was not possible to increase the prediction accuracy using content information.

Another paper which comes close to our concerns is [1] Here, the authors propose a new algorithm for diversifying recommendation lists. Their algorithm uses rating patterns of movies for diversification. They evaluate the results by looking at how well the recommendation lists are diversified with respect to genre and movie series they belong. They report that the resulting lists' diversity increase in both respects (genre and series). However, to the best of our knowledge there are no direct comparisons between rating and content similarity. In this paper we examine directly these two types of similarities.

3. ITEM CONTENT GENERATION

In our experiments we use Movielens[1] (1M) data set. In order to compare movies' rating patterns to their contents we first need to generate movie content information. We use two sources of information to this end. One source of content information comes from Wikipedia articles corresponding to movies in the Movielens dataset. The other source of content information comes from genre information which are provided in the dataset. Details of content generation are given below.

3.1 Content Generation from Wikipedia

The Movielens dataset contains 3883 distinct movies and 6040 users. Some of these movies are not rated by any user. Also some of the movies have no corresponding entries in Wikipedia. After discarding these movies we are able to fetch 3417 (approximately 88% of all movies) movie articles from Wikipedia.

In this work we only use the text of each Wikipedia article (we do not use link structure or category information of articles). The text of a Wikipedia article consists of parts such as "Plot", "Cast", and "Release". We do not include "References" and "See also" parts of the text since they may contain information which is unrelated to the content of the movies. After extracting the text of each document we apply some basic preprocessing steps such as stemming and stopwords removal. We use a vector space model to represent text documents.

3.2 Genre Information

As a second source of content we use the genre keywords (such as adventure, action, comedy, etc.) provided by the Movielens dataset. Each movie in the dataset is associated with one or more genre keywords. We define the genre similarity between two movies using the Jaccard metric given below:

$$J(i,j) = \frac{|G_i \cap G_j|}{|G_i \cup G_j|} \quad (2)$$

where G_i and G_j are genre sets of items i and j.

4. EXPERIMENTS

In the first set of experiments we try to understand the relation between movie rating patterns and content generated from the corresponding Wikipedia articles. We have two matrices: one is the Movie-User matrix which holds the ratings of users on movies and the other is the Movie-TFIDF matrix which holds the *tf-idf* weights for each document. For evaluation we use the following methodology. For each movie we find the most similar 100 movies using the Movie-User matrix (rating neighborhood) and the most similar 100 movies using Movie-TFIDF matrix (content neighborhood). We then find the number of common items in these two neighborhoods. It turns out that on average there are 14.74 common movies in the two neighborhoods. If we generate the neighborhoods randomly this value turns out to be around 2.80. Randomization tests show that this difference is significant ($p < 0.01$).

We run the same experiment with different neighborhood sizes (20 and 50) but the percentages of the number of common items in the rating and content neighborhoods turn out to be similar to the percentages we get when we use a neighborhood of size 100.

We also test whether there is a relationship between the number of ratings and the correspondence between rating and content similarity. To see this we find the rating and content neighborhoods of those movies which have similar number of ratings. To do this we divide the movies into rating intervals according to the number of ratings they have: movies which have ratings between 1-100, between 101-200, and so on. If an interval has less than 20 movies, we merge it with the previous one in order to increase the significance of the results. Figure 1 shows the average number of common items in the rating and content neighborhood sets of movies as a function of rating intervals. Interestingly, Figure 1 shows a clear linear correlation, i.e., as the number of ratings increases the number of common items in the content and rating neighborhood of movies also increases. One possible explanation of this positive linear correlation might be this. Generally, there is a positive relationship between the number of ratings and the popularity of a movie. This means that popular movies receive ratings from many different people with different tastes. Hence the rating patterns of popular movies reflect a diverse set of characteristics. Wikipedia movie articles also have rich contents reflecting different characteristics of movies. This might explain why a movie's rating neighborhood approaches to its content neighborhood as the number of ratings increase.

In the next set of experiments our aim is to understand the relationship between movie rating patterns and movie genres provided in the Movielens dataset. Genre keywords provide limited information compared to Wikipedia articles. Because Wikipedia articles contain terms that give information not only about the genre of a movie but also about the director, starring, musical composition, etc.

In order to measure the relationship between movie rating patterns and genres we applied a similar methodology. For each movie m we find the most similar 100 movies using the Movie-User matrix (that is the rating neighborhood) and find the Jaccard similarity (as defined in equation 2) between movie m and movies in its rating neighborhood. The average Jaccard similarity value turns out to be 0.43. If we generate the rating neighborhood randomly we find a Jaccard value around 0.17. Randomization tests show that this difference is significant ($p < 0.01$).

We also test whether there is a relationship between the number of ratings and genre similarity. Similar to the experiment we described above we divided the movies into rat-

[1] http://www.grouplens.org/node/73

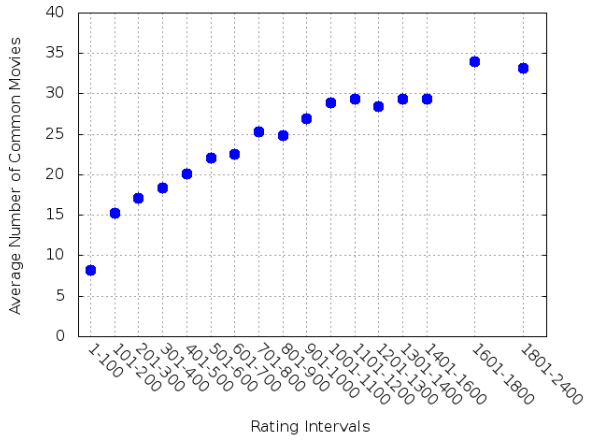

Figure 1: Average number of common movies as a function of rating intervals.

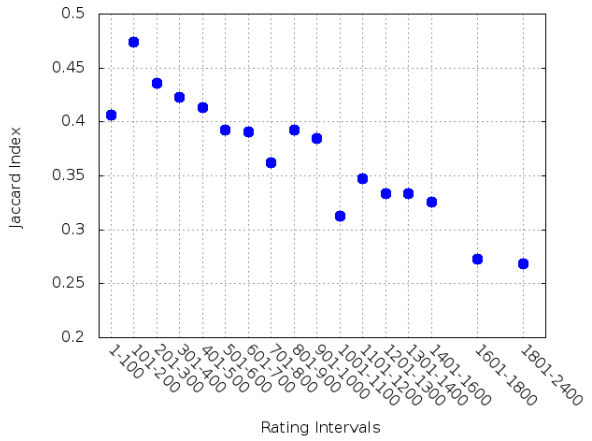

Figure 2: Average Jaccard index as a function of rating intervals.

ing intervals according to the number of ratings they have. Then for each movie m in a rating interval we calculate the Jaccard similarity value between the movie m and its rating neighborhood of 100 movies then calculate the averages per rating interval. Figure 2 shows these average values as a function of rating intervals. Here, we again have an interesting case. There is a *negative* linear correlation which means that the more a movie has ratings the more its rating similarity diverges from its genre similarity.

The reason underlying these results might be this. Movies which have limited number of ratings (unpopular movies) are generally watched by the fans of that genre. For example, a fan of sci-fi movies may also watch an unpopular sci-fi movie. So, unpopular movies generally get ratings from the same set of users who are fans of that movie's genre. And this makes the rating vectors of those movies (same genre movies) similar to each other. On the other hand if a movie is popular than it gets ratings from a diverse set of users which causes their rating neighborhoods diverge from its genre.

5. CONCLUSION

We should note at the outset that the conclusions presented here are not conclusive. Different experiments on different datasets and with different item types need to be done in order to drive more firm conclusions. However, we hope that these experiments and results will stimulate discussion and further research.

In this work we examined the relationship between rating similarity and content similarity of movies in the Movielens dataset. We examined two kinds of content: one of them is the *tf-idf* weights of movie articles in Wikipedia and the other is the genre keywords of movies provided by the Movielens dataset.

We found that to a certain degree there is a similarity between rating similarity and Wikipedia content similarity and also between rating similarity and genre similarity. However, we leave open to discussion the magnitude of these similarities. We also found that as the number of ratings of a movie increases its rating similarity approaches to its Wikipedia content similarity whereas its rating similarity diverges away from its genre similarity.

According to these results if diversification is done based on the rating patterns of movies then the recommendation lists will likely be diversified with respect to the content of movies to some extent. So, if no content information is available or it is difficult to get it, it might be useful to use rating patterns to diversify the recommendation lists.

To this analysis we plan to add latent characteristics of items generated by matrix factorization methods [4]. We plan to explore the correspondences among similarities defined over rating patterns, contents, and latent characteristics of items.

6. REFERENCES

[1] R. Boim, T. Milo, and S. Novgorodov. Diversification and refinement in collaborative filtering recommender. In *CIKM*, pages 739–744, 2011.

[2] N. Hurley and M. Zhang. Novelty and diversity in top-N recommendation - analysis and evaluation. *ACM Trans. Internet Techn*, 10(4):14, 2011.

[3] G. Katz, N. Ofek, B. Shapira, L. Rokach, and G. Shani. Using wikipedia to boost collaborative filtering techniques. In *RecSys*, pages 285–288, 2011.

[4] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[5] A. Loizou and S. Dasmahapatra. *Using Wikipedia to alleviate data sparsity issues in Recommender Systems*, pages 104–111. IEEE, 2010.

[6] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.

[7] J. B. Schafer, D. Frankowski, J. L. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The Adaptive Web*, pages 291–324, 2007.

[8] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys*, pages 123–130, 2008.