COMP47580

Recommender Systems & Collective Intelligence

Recommender Systems Report

<Safiya Rehmat>

<18200494>

Declaration of Authorship

I declare that all material in this assessment is my own work except where there is clear acknowledgement and appropriate reference to the work of others.

Signed ….....Safiya Rehmat..... Date .....26/04/2019.....

1 Non-Personalized Recommendation Analysis

1.1 Discussion on Metrics for gauging item similarity

There are 4 performance metrics used in the experiments are discussed in this section.

Genre: All movies are represented as a string containing all genres applicable to it. Similarity is found using Jaccard index. This metric will consider movies similar only if they have common genre. Thus, recommended movies will be very similar and may not get enough diversity or novel items in the recommendation

Genome: All movies are represented as a vector containing values for a predefined set of tags. Similarity between two movies is the cosine of these vectors. Since every movie has a score assigned for all tags, it is very effective in comparing not just what the movie is about but also what it is not about, unlike the case with genre.

Genre and Genome metrics are based on the movie descriptions and metadata, so they are content based and do not consider how people rate these movies. They will thus generate recommendations which are similar to the target movies but cannot distinguish between movies perceived as good or bad.

Rating: This metric is based on user ratings. For ratings metric the idea is that movies similarity can be determined by how users rate different movie. For this idea to be effective, we need multiple users to have rated the movies and is suitable for popular movies that have been rated by many viewers.

Inc. **Confidence**: Movies are given a high score if they have more users which rate both the movies highly and less users that give them a low score. In this case, more popular movies win over controversial ones.

Cosine and Inc. Confidence metrics depend on user rating patterns and thus rely on the idea that a group of people that like a particular movie will also like another one, which may not always be the case. However, Inc. confidence takes into consideration both users who like the movie and users who dislike the movie, so the recommendations made should be more relevant.

1.2 Experiment Data Analysis

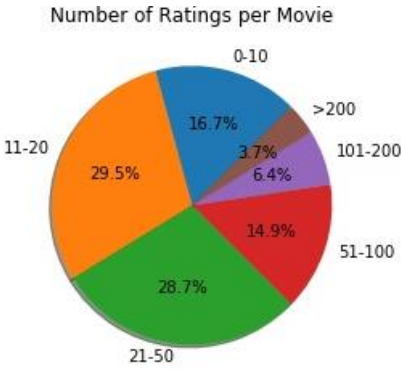

Figure 1 : Breakdown of number of ratings per movie

Figure 1 analyses the number of ratings for each movie in the dataset. The dataset contains 76k rows for ratings of 1661 movies. That averages to about 45 ratings per movie, however further analysis shows that is not the case. Few movies have more than 200 ratings each. 24.7% of the movies have more than the average number of ratings. More importantly, 46.2% of the movies have less than 20 ratings each. Keeping this in mind is important when we analyze the results.

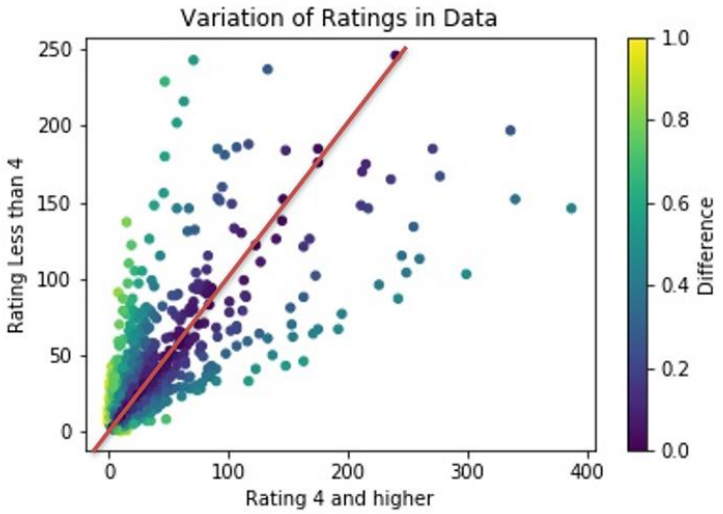

Figure 2: Analysis of controversy in Data

Figure 2 shows the variation in ratings for each movie. For each point on the graph, the x axis is the number of users who have given the movie a rating of 4 or more and the y axis the number of users who have the rated the movie less than 4. The color of each point specifies the difference in the number of higher and lower ratings for that movie. Thus, all movies which are plotted along the red line have minute difference in the number of high and low raters. That is, equal number of people like and dislike the movie. These are the controversial items and we can see many items in the dataset have the dark blue color corresponding to low difference. We can see that the training dataset contains many controversial movies. In the next section, performance of various evaluation criteria of the recommender system is analyzed taking controversy into account.

1.3 Results analysis

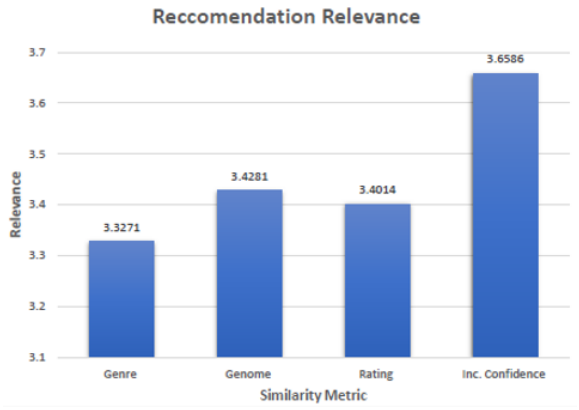

Figure 3: Performance of various metrics vs Relevance

Relevance in recommender systems: A recommender system that selects the popular or blockbuster items will have a higher relevance score. However, a recommender that makes novel and more interesting recommendations which are relevant to a niche group may not perform equally well. Even recommendations of controversial movies, which are equally liked and disliked by the people will score low because we are considering the mean rating. Figure 4 shows how various metrics perform with respect to relevance. Metrics which recommend popular movies with high ratings will performs well here. Inc confidence performs best as it is designed to recommend movies above a similarity threshold that accounts for how many users like that movie and penalized users who dislike the movie. The other 3 perform similarly with genome performing the best of the rest, which is again expected as there is maximum data to make recommendations in case of genome.

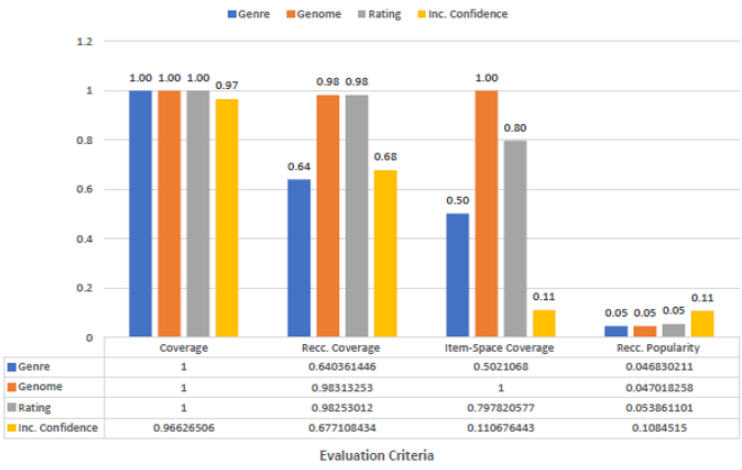

Figure 4: Performance of metrics Recommendation coverage, Item-space coverage and Coverage

We can see that all metrics perform well with respect to **Coverage**. All except Inc. confidence have 100% coverage. Inc confidence also has 97% coverage. Coverage is expected to be high since we are using a non-personalized approach. Inc. confidence might have slightly lesser coverage as the metric uses stricter rules to calculate similarity and there 299 movies in the dataset which have only a single rating.

Recommendation coverage scores are highly varied. Our hypothesis is that an inclusive recommender system would get a high score in this case. On the other hand, unlike in the case of Relevance, a system that only recommends the popular items should not get a high score. As expected, Genome metric performs well here as genome tag values are present for all movies, enabling connections between all pairs of movies and thus most of them may appear in the top k recommendations of other movies. Rating metric performs surprisingly well. 98% of the movies appear in the top k recommendations, I would have expected this value to be lesser as it is dependent on how many ratings are there for pairs of movies. Genre scores the least, which is expected as most movies tend to have the same few genre tags and other movies which don't contain that tag get excluded from the recommendation list.

Item-space coverage is the next criteria in the graph. The expectation is that in case of a relatively new system with few ratings for movies, the similarity metrics using ratings to calculate similarity will not perform well wrt. item-space coverage. On the other hand, for content-based recommenders, the item-coverage score will depend on what content is used to calculate similarity and how many movies having overlapping content.

Genome and Genre are the content-based recommenders in our experiment. As expected, Genome metric has the best score here. Since all movies have values for all tags, we get similarity greater than 0 when comparing two movies by genome. Contrastingly, genre does not perform well because most movies have a few common genre tags and movies that don't contain those tags will not have a positive similarity.

Amongst the ratings-based recommenders, Item-space coverage is only 11% for inc. confidence metric. Since there are many movies with few ratings, it is expected that not every pair of movies will have a positive similarity with a strict increased confidence metric. This should not be considered a drawback, as the metric is designed strict in a way to recommend the more relevant movies and all movies don't cross the threshold.

The next section of the graph shows **Popularity** scores. As the name suggests, popular movies that have been watched and reviewed by many people should be recommended to get a high popularity score. Like relevance, systems which recommend niche movies may not score well. However, unlike relevance, controversial movie recommendations do not suffer here as we only consider if a recommendation has been made or not, not the rating itself.

2 Item Based Collaborative Filtering Analysis

2.1 Evaluation Criteria for different similarity metrics

The following 2 criteria are analysed in this section of the report:

1. Coverage: 100% coverage means that a recommendation can be made for every item. Therefore, for an effective recommender system, a higher Coverage value is desirable.
2. RMSE: As the name suggests, a lower RMSE value is desirable as it means that the predicted ratings are closer to the actual ratings.

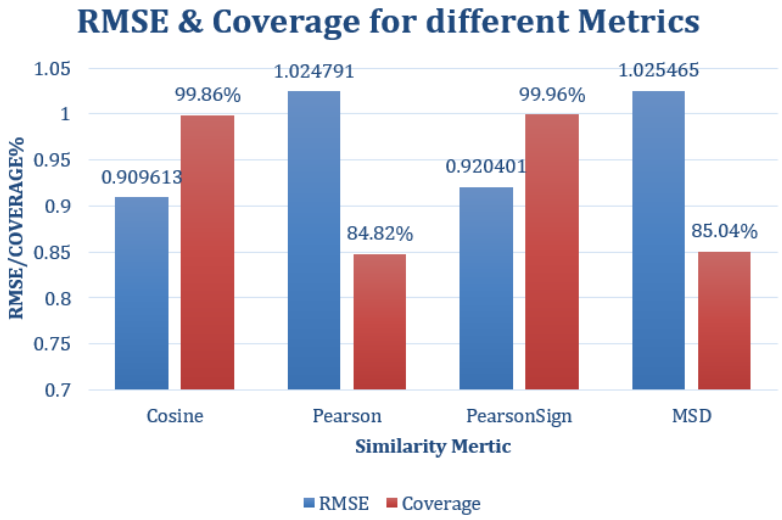

Figure 5: RMSE and Coverage for various similarity metrics

Figure 5 shows the performance of 4 similarity metrics: Cosine, Pearson, Pearson Weighted and Mean Squared Difference (MSD) wrt. the evaluation criteria.

We can see that cosine has the lowest RSME and almost the highest coverage, thus it is the best performing metric we examined. It is closely followed by Pearson significance weighting which has 2% higher RSME. Mean-squared-difference and Pearson correlation both have high errors and 85% coverage, thus not performing as well as their counterparts. It is expected behaviour that Pearson significance weighting performs better than the simple correlation metric as it takes into account the effect of number of common ratings on similarity.

2.2 Performance of various predictor strategies

The Simple Average, weighted average and deviation-from-item-mean predictors were analyzed in comparison with a non-personalized predictor. The general expectation is that the predictors will have lower errors than the non-personalized although the coverage may decrease a little.

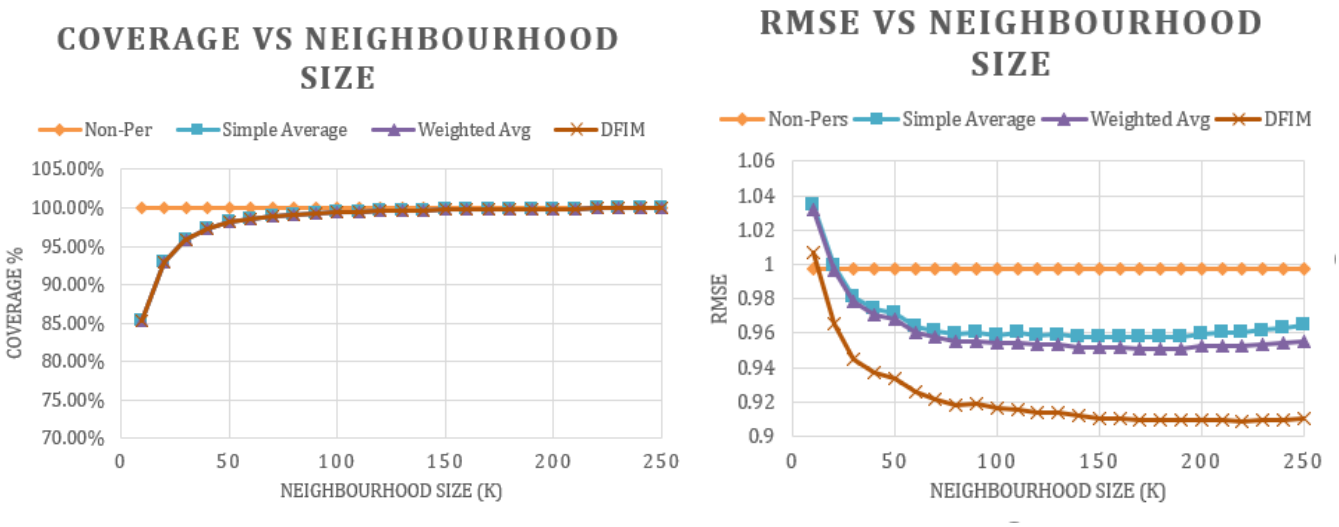

Figure 6: Coverage and RMSE for various predictors

The figure above shows the experiment results for performance of various predictors wrt. RMSE and coverage. The neighbourhood strategy used here is kNN and the effect of using different neighbourhood strategies is discussed in the next section. As it is expected, the coverage for all predictors other than the non-personalized one is lesser than 100% for small values of k. But as we increase the neighbourhood size, the coverage reaches 100% for all predictors.

Also, with increasing neighborhood sizes, the error decreases for all predictors, most significantly for DFIM predictor. This is the expectation because DFIM takes into consideration the fact that some items may be rated higher than others on average. **Weighted average predictor has slightly lower error than simple average**, but not much. Since we are weighting the averages by similarity, we should expect more decrease in the error, which is not the case. The hypothesis we present is that the performance of weighted average is not much better because there is very less variance in the ratings of movies. To verify this, we can plot a histogram with the similarities of all items in neighborhood of every target item and then see the distribution of pair-wise item similarities. The figure below shows this distribution.

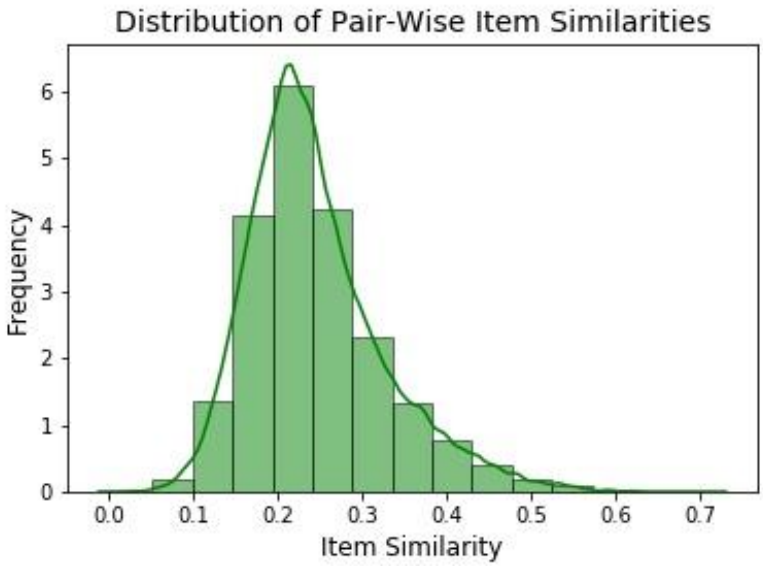

Figure 7: Distribution of Item Similarities in the MovieLens Dataset

We can see from Fig. 7 that most pair-wise item similarities fall between 0.15 and 0.25 and the spread is very skewed towards 0.2. Thus, the impact of weighting the average ratings by the similarities does not give us much higher accuracy. For a dataset where the item similarities are more evenly distributed, we should expect the accuracy of a weighted average predictor to be significantly better than simple average predictor.

2.3 Performance of neighborhood strategies

The experiments conducted used 2 neighborhood formation strategies: Nearest neighbor and threshold. The threshold approach suffers the drawback that if an item does not have close neighbors, they won't be included in the neighborhood, thus reducing coverage. So, we need to strike a balance between a threshold high enough to have less error but low enough so that items are included in the neighborhood of a target item.

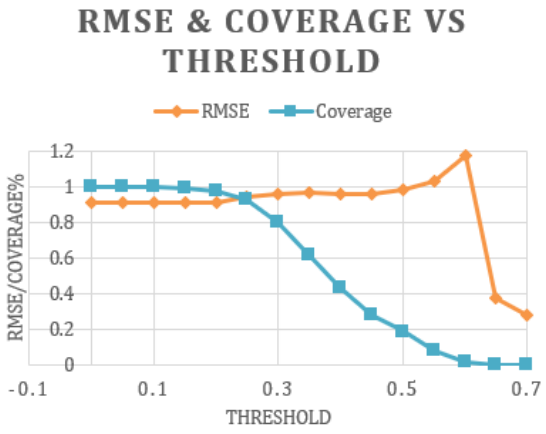

Figure 8: RMSE and Coverage vs Threshold

Figure 6 in the section 2.2 showed the performance of nearest neighbor approach with respect to the neighborhood size.

In this section we examine the performance of threshold approach of neighborhood formation and its impact on the RMSE and coverage. Figure 8 shows us that as the threshold increases, the error decreases but the coverage falls drastically as well. Thus, we need to have a tradeoff between coverage and error when selecting the threshold value. It is expected that the coverage decreases as the similarity threshold is increased. However, **the coverage reaches almost zero as the threshold is increased to 0.6** and above. This is because the item-similarity distribution we saw in Figure 7, there are very few pairs of items with similarity greater than 0.6. Thus, with the given spread of similarities in the data, having a threshold greater than 0.5 will not provide good coverage.

Now let us examine the RMSE in Figure 8. The error remains constant or increases very slightly and then drops drastically at 0.6. Let us have a closer look at the average neighborhood sizes for all threshold values:

Threshold	Average Neighbourhood Size
0	1510.85
0.05	1241.8
0.1	767.0842
0.15	393.64
0.2	177.53
0.25	76
0.3	34.256
0.35	17.6568
0.4	10.331
0.45	6.501
0.5	3.14
0.55	1.61
0.6	1.025
0.65	1.2
0.7	1

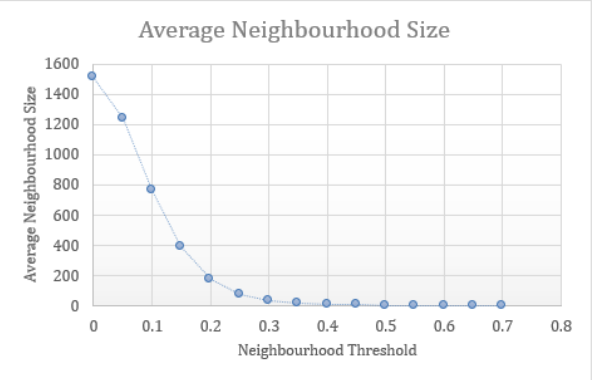

Figure 9: Average Neighbourhood Sizes for various Threshold Values (excluding size-zero neighbourhoods)

Figure 9 shows the average neighborhood size considering only those items for whom we have a neighbor. We can see that even for neighborhoods that are found at higher thresholds, the contain only 1 or 2 elements. It should be kept in mind that RMSE is only calculated over items where a prediction can be made. Thus, low coverage and low average neighborhood sizes combined make the reliability of RMSE performance at threshold values higher than 0.5 unreliable.