Recsys Data Analysis

February 18, 2017

1 Results

Table 1: LastFM results

Algorithm	Avg. of RMSE.ByUser	Avg. of RMSE.ByRating	Avg. of Predict.nDCG	Avg. of MRR
ItemItem	1455	4469	0.7344	0.001738
PersMean	1225	3052	0.8117	0.000650
UserUser	1226	3251	0.7435	0.001728

Table 2: Movielens results

Algorithm	Avg. of RMSE.ByUser	Avg. of RMSE.ByRating	Avg. of Predict.nDCG	Avg. of MRR
ItemItem	0.8903	0.8969	0.9552	0.09501
PersMean	0.9208	0.9382	0.9499	0.00264
UserUser	0.9151	0.9249	0.9533	0.00377

Table 3: Jester results

Algorithm	Avg. of RMSE.ByUser	Avg. of RMSE.ByRating	Avg. of Predict.nDCG	Avg. of MRR
ItemItem	0.7906	0.8347	0.9510	0.6118
PersMean	0.8265	0.8768	0.9446	0.6177
UserUser	0.7960	0.8369	0.9512	0.7127

2 Datasets

2.1 LastFM

#Users: 1892 users
#Items: 17632 artists
This dataset contains 92.834 user-artist relations. The weight of the graph's edges represents *amount of plays*. We considered these weights as ratings.

In this iteration of the analysis we did not normalize the numbers into a $[1, 5]$ scale. It can be done by classifying each relation into its respective quintile. This approach, however, has the drawback of discretizing something that is very continuous; for example, relationships that are on the upper scale of a quintile are qualitatively similar to those on the lower scale of the next quintile, but we are losing that information. Another consequence (which may not be a problem, but it is an interesting result nonetheless) is that each class will have exactly the same amount of members (relationships). As datasets that include a normal rating system often do not have this property, it may be appealing to run the algorithms on this normalized dataset.

Another approach is inspired by the Mahalanobis distance; in essence, calculate the mean of the ratings and for each rating calculate how many standard deviations it strays from the mean. Those that are deviated to higher values are assigned a value near to 5 and ratings that deviate from the mean but by being lower are assigned values near to 1. This might have the effect of having ratings form a normal distribution, which is also not common in "true" rating datasets.

2.2 Movielens-1k

#Users: 943 users #Items: 1682 movies
This dataset contains 100.000 user-movie relations. Each relation represents the evaluation of a movie by a user (in a scale from 1 to 5).

2.3 Jester

#Users: 73.495 users
#Items: 100 jokes
This dataset contains 4.1 million of user-joke relations. Each relation represents the evaluation of a movie by a user (in a scale from -10.0 to 10.0). As the scale is different to the used in Movielens, we decided to normalized the evaluations to a scale of [1,5].

This dataset is really unique as the number of items, in this case *jokes*, is only 100. One of the consequences of this property is that the user-item matrix is not as sparse as it normally is elsewhere. It wouldn't be unusual for one user to have rated a majority of the items; in contrast, this would have been unthinkable in other datasets like MovieLens or Amazon-Books. The quantitative characterization of this dataset (and the rest) remains as future work (see **conclusions** for details).

2.4 Bookcrossing

#Users: 278.858 users
#Items: 271.379 books
This dataset contains 1.149.780 of user-book relations. Each relation represents the evaluation of a book by a user.

3 Metrics

3.1 RMSE

RMSE reflects the difference between the real values and those predicted by the algorithm. It is calculated in two ways:

- Grouped by user: RMSE.byUser

$$\frac{\sum_{\forall u \in U} \sqrt{\frac{\sum_{\forall r \in R} err_{ru}^2}{|R_u|}}}{|U|} \tag{1}$$

- Globally: RMSE.byRating

$$\sqrt{\frac{\sum_{\forall u \in U} \sum_{\forall r \in R} err_{ru}^2}{|R|}} \tag{2}$$

In general, both formulae give similar results.

In the case of LastFM, while grouping by users, the results are approximately a third of those obtained using the global expression.

3.2 MRR

This metric measures the ranking quality of the algorithm. MRR gives particularly good results for Jester. That seems to be caused by the reduced number of items in the dataset (only 100 jokes). So it could be easier to "guess" the correct ranking in Jester than in the other datasets because of it has less items to order, as diversity isn't guaranteed at all. This contrasts with other datasets where there are many *niche* genres and communities.

It is noteworthy that MRR result for Movielens using item-item CF are approximately 40 times better than the result for the same metric, the same dataset, but with other algorithms.

3.3 nDCG

nDCG also reflects the ranking performance of an algorithm. For Jester and Movielens, we can see that it has a similar behavior to that of MRR. But in LastFM dataset, its worst result co-occurs with the best results of MRR.

4 Algorithms

4.1 CF Item-item

For the rating evaluations (MRR), item-item CF is the algorithm with the best performance in LastFM and Movielens. But with Jester it is worse than CF User-user.

4.2 PersMean

This algorithm has the best RMSE performance for Jester and Movielens, while in LastFM the other algorithms perform better.

4.3 CF User-user

In general CF User-user has good, but not outstanding, RMSE results with Jester and Movielens. Also, in Jester the best MRR results are obtained by CF User-user.

4.4 FunkSVD

We have added FSVD to the repository but we have yet to run all the datasets with this algorithm.

5 Conclusions

Although we didn't do enough analysis to deduce solid conclusions, we can see that the behavior of the algorithms with LastFM is notoriously different to the one with the other datasets. This is probably a consequence of the fact that the relation between user-item is not an explicit evaluation, like in the other datasets, it is the number of times the user listened to the item (artist). We have also noted that the datasets are very different in their topology, so it may be interesting to look further into the properties of each one to be able to compare them more appropiately.

Future work can be summarized as:

More algorithms We have already added FunkSVD but we haven't yet compiled the results for all datasets.

More datasets We'll begin by adding small parts of big datasets (like Amazon). We have also tried to get the CiteULike dataset but the email listed on the site page appears to be non-reachable, we are currently searching for other options to get the data.

Analysis of each dataset As each dataset listed has very different properties (compare Jester to Amazon, for example), it might be worthwhile to see how they differ.

Grid-search Analysis of hyperparameters is needed in order to obtain stronger conclusions.