MemoryLeak Iteration 2 Report

In this iteration, we implemented a precision-recall evaluator and we improved our mathematical model. To test those, we used the logs of the last 13 days from Argedor's dataset, which contains 5,225,092 logs, 144,292 different users and 109,147 different songs. This means that a user listened to an average of 36 songs.

Rather than predicting the ratings and compare them with the original ones (as our previous evaluator does), prediction-recall evaluator selects test users, hide some of their selections, and ask the recommender to predict a set of items that the user will use. Then, it calculates precision and recall values based on the following formulas;

$$Precision = \frac{Recommended\ Items\ \cap\ Relevant\ Items}{Recommended\ Items}$$

$$Recall = \frac{Recommended\ Items\ \cap\ Relevant\ Items}{Relevant\ Items}$$

Simply, precision is the proportion of top-recommendations that are good, and recall is the proportion of good recommendations that appear in top-recommendations.

To illustrate, let's consider the user 1000217 in Argedor's dataset. According to the logs in the dataset mentioned above, that user listened to 34 songs (11 different songs). Assume that the evaluator hides the songs {3408894, 3407844, 3408786, 3408899} from the recommender and the recommender recommends {3408894, 3409040, 3408786, 64557, 3409296}. Then, Recommended Items ∩ Relevant Items is equal to {3408894, 3408786}. Therefore, the precision value is equal to 2 over 5 which is 40% and precision value is equal to 2 over 4 which is 50%.

The evaluator calculates these rating and recall values for each user and the final precision and recall are the average of all precisions and recalls for all users.

Based on our research in the previous iteration, we came to a decision that we need to use the time that the user listened the songs in our rating calculations. So, we implemented the following formula;

$$R(u,s) = \frac{min\big(max, r(u,s)\big)}{max} \times 5$$

$$r(u,s) = \sqrt{\frac{\left(\sum_{i=1}^{n_s} d_{max} - d_i\right)^2}{n_s}}$$

$$max = d_{max}\sqrt{n}$$

, where $R(u,s)$ is the user u's rating on song s, n_s is the number of times that s is listened by u, d_i is the number of elapsed days since s was listened the i^{th} time, d_{max} is the number of days between the day of the first logged action in the system and the day of the ratings are calculated, and n is the number of times that the song needs to be listened on the current day to have a maximum rating, in our calculations n is equal to 5. For example if we are calculating the ratings of last 2 months and the user u listened the song s the first time 3 days ago, then $d_{max} = 60$ and $d_1 = 3$ in $r(u, s)$.

The factors that we take into consideration while implementing the model above includes but not limited to that a song which is listened to a day ago should get a lower rating than a song which is listened to both a day and two days ago. Moreover, a song listened twice in the same day should get a higher rating than a song listened once in that day.

Let’s show it in a real life example. If we are generating the ratings the day of the last log of the dataset above, the user 1000217, who again is the subject of our example, listened to the song 3407844 once today, and 4 times yesterday. So, the rating value of that song is equal to 4.7 out of 5.

This model has a precision of 23% and a recall of 21%. Our previous math model, which we introduced at the final demo of fall term, had a precision of 17% and a recall of 16%, which means that this new model has made a great improvement in the quality of our recommendations. Therefore, from now on we will be using the model introduced in this paper in our recommender system.

We have done everything that we planned for the iterations on time. We have a high precision value which indicates that statistically the users listened to one fourth of the songs we recommended them. On the next iteration we will speed up the processing time of the raw data (generation of ratings) by putting away MongoDB. As a result, the stakeholders will be able to generate ratings, more than once a day and can have up-to-date ratings any time of day.