Systems for Data Science - Project Milestone 1

Vincent Yuan
SCIPER 287639

June 11, 2021

Question 2.2.1

In this question and the next ones, I will round values at the 4th decimal.

The mean absolute error using adjusted cosine similarity is **0.7478**. As a reminder, the baseline MAE is **0.7669**. Hence, the difference between the Adjusted Cosine similarity and the baseline is **-0.0191**.

Result is logical, as the adjusted cosine similarity take into account the similarity between 2 users, while the baseline predicts using only item average and user average.

Question 2.2.2

The Jaccard Coefficient between 2 set U and V is:

$$Jacc(U, V) = \frac{U \cap V}{U \cup V} \tag{1}$$

The mean absolute error by computing jaccard similarity is **0.7626**. Hence, the difference between Jaccard Coefficient and Adjusted Cosine similarity is **0.0148**.

It makes sense, as the jaccard based prediction only takes into account the number of common movies watched between 2 users, while cosine similarity takes also into account the similarity between 2 users.

Question 2.2.3

In worst case situation, we will have to compute all pairs

$$u, v \ s.t \ u \in U, v \in U$$

Also, if we compute (u, v), we don't have to compute (v, u) - we can cache the value to divide the number of operations by 2.

Thus, the first user u_1 will compute his similarity with $|U|$ users, the second user u_2 will compute his similarity with $|U| - 1$ users (every similarity, except the one between u_1 and u_2), and so forth, until u_n which will compute only one similarity, namely the one with itself.

We will then have to compute:

$$\sum_{n=1}^{|U|} n = |U| * (|U| + 1)/2 \tag{2}$$

in the *ml-100k* dataset, we have 943 users. Hence, we will have 943 * 944 / 2 = **445096** similarities to compute.

Question 2.2.4

The number of multiplications required for each possible $s_{u,v}$ is equal to the number of common items $|I(u) \cap I(v)|$ between u and v.

Results of the computation of the minimum number of multiplications are shown below:

Metric	Number of multiplications
Min	0
Max	685
Average	12.2820
Standard deviation	18.5203

Table 1: Metric of the computation of the minimum number of multiplications for each possible $s_{u,v}$

Question 2.2.5

Firstly, we have to compute the number of similarities $s_{u,v}$ having non-zero values - we have computed the number of $s_{u,v}$ having zero values at Question 2.2.3. We compute this number by filtering our resulting dataset, and we arrive to **412489** similarities having a non-zero values.

Then, knowing that each similarity is stored as a double having a 64-bit floating point value, we know that each similarity is stored in 64/8 = 8 bytes.

Hence, the number of bytes needed to compute non-zero values is **3299912**.

Question 2.2.6

By benchmarking the total computing time on 5 different iterations, I obtained this table:

Metric	Time required for computing prediction in μs
Min	2.944E7
Max	2.9964E7
Average	2.9689E7
Standard deviation	2.0546E5

Table 2: Benchmark of prediction computing time on five measurements

It is roughly superior than previous milestone computation, which makes sense - we have more computations in the cosine similarity prediction than in the baseline. Indeed, we have to compute the preprocessing of the similarities, the similarities, and the user specific weighted sum in this milestone.

Question 2.2.7

Results are shown below:

Metric	Time required for computing similarities in μs
Min	2.0863E7
Max	2.1844E7
Average	2.1321E7
Standard deviation	3.6380E5

Table 3: Benchmark of similarities computing time on five measurements

The average time per $s_{u,v}$ is **47.9016** seconds.

On average, the ratio between computation of similarities and total time required to make predictions is **71.8144%**

The computation of similarities is largely significant for predictions, as it takes roughly three quarters of the time.

Question 3.1.1

I didn't succed on creating the knn. I might have exceeded the RAM size, thus having multiple I/O between disk and RAM, which is not optimal.

As I already submit the assignment late, I decided to not push further and to submit the given code.

Thus, I will not be able to answer to this question. However, I'll try to answer the best that I can to the next questions, which are theoretical.

Question 3.1.2

Assuming that we only store similarity values as a double and that we have $|U|$ users, the formula will be:

$$Number\ of\ bytes\ required = |U| * (64/8) * k = 8 * |U| * kbytes \tag{3}$$

Indeed, for all users, we have to store their similarities with k neighbours, and each similarities of 64 bits are stored in 8 bytes.

Question 3.1.3

I have 16 GB (16138216000 bytes) of Ram in my laptop.

As I don't have a lowest k as my computation didn't succeed, I will find the maximum k values such that the 943 users could be stored in RAM.

$$k_{max} = \frac{16138216000}{|U| * (64/8) * 3} = \frac{16138216000}{943 * 8 * 3} = 713070 \tag{4}$$

The maximum k numbers of my RAM is 713070 if I have 943 users - thus, my computer should be good enough to handle 943 users. So I have either not optimized the code to exceed the limit, either my problem is not on the RAM part.

3.1.4

In my code, I compute, for all user, the similarity they have with all other users, in a decreasing way. Thus, I compute all k for all users, and then select the k best. Thus, changing k will not change the number of similarities computed.

In general, as we have to sort the similarities, we have to compute them all before ordering them - thus, changing k will not change the number of computations.