École Polytechnique Fédérale de Lausanne

Report-223876-M2

CS-449 Milestone 2

Azouz Rami

30th April 2021

Q2.2.1

1 QUESTION

Is the prediction accuracy better or worst than the baseline (Eq. 5 from Milestone 1)?

2 ANSWER

The MAE of the cosine similarity based method is 0.7478. This is an improvement by -0.0191 comparing to the baseline method.

Q2.2.2

1 QUESTION

Provide the mathematical formulation of your similarity metric in your report. Compute the prediction accuracy and report the result. Compute the difference between the Jaccard Coefficient and the Adjusted Cosine similarity (Eq. 1, jaccard - cosine) Is the Jaccard Coefficient better or worst than Adjusted Cosine similarity?

2 ANSWER

To compute the Jaccard coefficient of a pair of users (u,v) is computed as follow:

$$\frac{|I(u) \cap I(v)|}{|I(u) \cup I(v)|} \tag{2.1}$$

Where I(u) stands for the items rated by the user u.

This coefficient measures how much two users have items in common and it is another measure for similarity. The MAE of this method is 0.7621. This method underperform the cosine similarity method and the difference in MAE is 0.0143.

Q2.2.3

1 QUESTION

In the worst case and for any dataset, how many su,v have to be computed, as a function of the size of U (the set of users), if every user rated at least one item in common with every other users? (Provide the formula in your report) How many would that represent if that was the case in the 'ml- 100k' dataset? (Compute the answer in your code from the number of users in the input dataset)

2 Answer

The Ml-100k data set contains $U = 943$ distinct users.In the worst case scenario we need to compute $U^2 = 889249$. However, we can do better by removing duplicates as $S_{u,v} = S_{v,u}$. This results into $U * (U - 1) + U = 440381$ computations. In the code, we reported only the worst case scenario.

Q2.2.4

1 Question

Compute the minimum number of multiplications required for each possible $S_{u,v}$ on the ml-100k/u1.base Do not consider (Tip: This is the number of common items, $|I(u) \cap I(v)|$, between u and v.) What are the min, max, average, and standard deviation of the number of multiplications? Report those in a table.

2 Answer

Min	Max	Mean	Std
0	685	12.205	18.175

Table 1
Similarity Statistics

We can see in Table 1, the statistics for the number of multiplications required for the computation of $S_{u,v}$. In the code, we computed that directly by getting number of common items between the pair of users (u,v).

Note: We also take into account auto-similarities ($S_{u,u}$).

Q2.2.5

1 Question

How much memory, as a function of the size of U, is required to store all possible $S_{u,v}$, both zero and non-zero values, assuming both require the same amount of memory? How many bytes are needed to store only the non-zero $S_{u,v}$ on the 'ml-100k' dataset, assuming each non-zero $S_{u,v}$ is stored as a double (64-bit floating point value)? (Tip: Do not include memory usage for intermediate computations, only for storing the final results.)

2 Answer

The total number of bytes to store all the similarities is equal to the number of similarities times 8 (bytes). This is equal to $889249 * 8 = 7113992$ bytes. The number of bytes required to store only the non-zeros is 6544136. This means that 92% of similarities are different than zero.

Q2.2.6

1 Question

Measure the time required for computing predictions (with 1 Spark executor), including computing the similarities $S_{u,v}$. Provide the min, max, average, and standard-deviation over five measurements. Discuss in your report whether the average is higher than the previous methods you measured in Milestone 1 (Q.3.1.5)? If this is so, discuss why.

2 Answer

t (μs)	Min	Max	Mean	Std
Similarities	3.3022E7	3.4641E7	3.3713E7	527176.501
Prediction	4.3881E7	4.5415E7	4.4680E7	584357.020

TABLE 2
Time required to compute similarities and prediction in μs

Table 2 summarizes the statistics of the computation time. We can clearly see that the average is much higher than previous methods and this is for the following reasons:

1. In previous methods, we just needed to compute the user deviation. We did that by replacing it by the average deviation of the user or the item. The prediction is almost Linear.
2. In the similarity method, the user deviation also depend on other users that rated the same item. This is obtained by computing a weighted sum that involves similarities with these users.
3. Similarity computations is heavy and takes 75% of the time.

Q2.2.7

1 Question

Measure only the time for computing similarities (with 1 Spark executor). Provide the min, max, average and standard-deviation over five measurements. What is the average time per $S_{u,v}$ in microseconds? On average, what is the ratio between the computation of similarities and the total time required to make predictions? Are the computation of similarities significant for predictions?

2 Answer

The time measurements for computing similarities are summarized in Table 2. The average time per $S_{u,v}$ is $37.9116\mu s$. The ratio between the time of computing similarities and the prediction time is 0.7545. This means that 3^{rd} of the time is spent computing similarities which is quite significant.

Q3.1.1

1 QUESTION

What is the impact of varying k on the prediction accuracy? Provide the MAE (on ml-100k/u1.test) for k=10, 30, 50, 100, 200, 300, 400, 800, 942. What is the lowest k such that the MAE is lower than for the baseline method (Eq. 5 of Milestone 1)? How much lower? (Use a baseline MAE of 0.7669, and compute lowestk - baseline). Do not include self-similarity in the k-nearest neighbours.

2 ANSWER

k	MAE	Memory Usage (Bytes)
10	0.8407	75440
30	0.7914	226320
50	0.7749	377200
100	0.7561	754400
200	0.7485	1508800
300	0.7469	2263200
400	0.7474	3017600
800	0.7472	6035200
943	0.7478	7106448

TABLE 3
Knn measurements

The MAE measurements for are summarized in Table 3. The lowest k such that the MAE is better than the baseline is k=100. It outperforms the latter in MAE by -0.0108.

Q3.1.2

1 QUESTION

What is the number of bytes required to store the k nearest similarity values for all users, i.e. top k $S_{u,v}$ for every u? Assume an ideal implementation that stored only similarity values with a double (64-bit floating point value) and did not use extra memory for the containing data structures (this represents a lower bound on memory usage to plan hardware capacities). In your report, provide a formula as a function of the size of U, assuming all users have exactly k neighbours. In your code, compute the total number of bytes for each value of k for the actual number of neighbours (i.e. $\leq$ k) of each user. Do not include self-similarity (the similarity of a user with themselves).

2 Answer

The number of bytes to store the k nearest similarities is $N = U * k * 8$ where U is the number of users in the data set. The measurements of the memory used for KNN are summarized in Table 3

Q3.1.3

1 question

Provide the RAM available in your laptop. Given the lowest k you have provided in Q.3.1.1, what is the maximum number of users you could store in RAM? Only count the similarity values, and assume you were storing values in a simple sparse matrix implementation that used 3x the memory than what you have computed in the previous section (2 64-bit integers for indices and 1 double for similarity values).

2 Answer

The RAM available in our laptop is 16 GB. The maximum number of users that could be stored in RAM is $N = \frac{TotalMemory}{lowestk*3*8} = 7158278$ users.

Q3.1.4

1 question

Does varying k has an impact on the number of similarity values $S_{u,v}$ to compute, to obtain the exact k nearest neighbours? If so, which? Provide the answer in your report.

2 Answer

Varying might have an effect on the number of similarities to compute. If k is bigger than the number of users v with which u has items on common, all the following similarities would be equal to zero and we do not need to compute them. However, in our implementation, there is no difference.

Q3.1.5

1 question

Report your personal top 5 recommendations with the neighbourhood predictor (Eq. 3) with k=30 and k=300. How much do they differ between the two different values of k? How much do they differ from those of the previous Milestone?

2 Answer

The top5 recommendations with the neighbourhood predictors for different k are:

- k = 30

    1. 136, "Mr. Smith Goes to Washington (1939)", 5.0
    2. 178, "12 Angry Men (1957)", 5.0
    3. 236, "Citizen Ruth (1996)", 5.0
    4. 253, "Pillow Book", 5.0
    5. 320, "Paradise Lost: The Child Murders at Robin Hood Hills (1996)", 5.0

- k = 300
    1. 361, "Incognito (1997)", 5.0
    2. 814, "Great Day in Harlem", 5.0
    3. 1114, "Faithful (1996)", 5.0
    4. 1127, "Truman Show", 5.0
    5. 1189, "Prefontaine (1997)", 5.0

As expected, these recommendations are user specific. In the previous milestone only the average rating matters. Varying k changes completely the top5 recommendations because we have more neighbors and thus have more items in common with them. However, the recommendations might be improved by implementing an additional popularity metric of items to the prediction equation.