Systems for Data Science

CS-449

Movie Recommendation with k-NN - Milestone 2

Author:
Kamil Seghrouchni

Professors:
Erick Lavoie
Athanasios Xygkis

Spring Semester 2021

Contents

Run time note

Having started the milestone early on, i struggled some time to make the suggested implementation work under what i thought were the run time constraints (10 minutes). Looking at the forum, i understand it changed but i wanted to mention the total run time on my machine since it appears it can be different on the cluster. Therefore, my total run time (all questions to be answered is between 9-10 minutes).

Numerical Figures note

Every figure mentioned in this report has been, as instructed, rounded up to the forth decimal.

2 Similarity-based Predictions

2.1 Preprocessing Ratings

2.2 Questions

2.2.1 Cosine based mean absolute error and baseline difference :

The prediction accuracy for mean absolute error was computed and is reported in Table 1. It was found to be better than the previous milestone baseline with an improvement of 0.0046.

Table 1: mean absolute error for cosine based prediction

	mae	$\delta baseline = 0.7669$
CosineBasedMae	0.7478	-0.0046

2.2.2 Jaccard based mean absolute error and baseline difference :

In order to compute the prediction accuracy using the Jaccard similarity, the following formula was used :

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

A and B being the user the respective item sizes for user A and B, $A \cap B$ the size of their item intersection, $A \cup B$ the size of their item union.

The prediction accuracy using this similarity metric and difference with previous cosine based prediction was referenced in table 2

Table 2: Jaccard based similarity prediction accuracy

	mae	$\delta cosine = 0.7478$
JaccardBasedMae	0.7623	0.0145

The Jaccard based similarity was found to be worse than the cosine based one and by 0.0145, (but still better than previous milestone).

2.2.3 Number of similarity computations :

In the worst case and for any data set the number of $s_{u,v}$ with u $\neq$ v as a function of the size of U can be computed using the following formula and yields to :

$$\sum_{u \in 1,2,...,U} u - 1 = \frac{U * (U-1)}{2}|_{U=943} = 444153$$

This allows to compute the number of items in the lower triangle part of a 2d matrix containing all combinations of user u and v with u $\neq$ v:

$$S_{u,v} = \begin{pmatrix} \cdots & \cdots & \cdots & \cdots \\ s_{2,1} & \cdots & \cdots & \cdots \\ s_{3,1} & s_{3,2} & \cdots & \cdots \\ s_{4,1} & s_{4,2} & s_{4,3} & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ s_{U,1} & s_{U,2} & \cdots & s_{U,U-1} \end{pmatrix}$$

The idea behind this reasoning is to avoid computing duplicate similarity pairs such as $s_{u,v}$ and $s_{v,u}$. In light of the suggested implementation that would ease the scaling in the up coming milestone, one had to find ways to alleviate the computation cost in term of time and memory for computing all possible user similarities. With the lower matrix technique, we trade such burden with a much smaller cost endured by the search.

The average rating for each item was computed and one has found that all the items do not rate on average close to the global average. This result was obtained by assessing whether or not for all items, the difference between the minimum of items average rating and the global average as well as the difference between the maximum of items average rating and the global average are lesser than a given threshold of 0.5. The ratio of items with average rating close to global average was computed and one has founded it to be equal to :

2.2.4 Cosine similarity statistics :

The statistics for the minimum number of multiplications per similarity as a function of the size of U required for all possible $s_{u,v}$ are reported in table 3

Table 3: Cosine similarity statistics

	min	max	average	standard deviation
CosineSimilarityStatistics	0.0	332.0	12.1280	17.8222

2.2.5 Storing similarities :

Assuming that similarity $s_{u,v}$ is stored as a double in a (64 bit floating point system), the number of bytes required to store all possible $s_{u,v}$ without intermediate computation is equal to :

$$\#s_{u,v} * numberOfBit/(Bytesequivalent) = \#s_{u,v} * 64/8 = 444153 * 8 = 3553224$$

Now, if we discard the all zero value similarities, this number becomes : 3292368.

2.2.6 Prediction time measurements :

The measurements for computing prediction for this milestone as well as previous one is shown in table 4.

Table 4: Duration in Mirco seconds for computing predictions

Time duration	min(ms)	max(ms)	average(ms)	stddev(ms)
DurationInMicrosecForComputingPredictionsM2	2.1120E7	2.2639E7	2.1790E7	633661.9266
DurationInMicrosecForGlobalMethodM1	55237.447	151261.975	8.021010e+04	27888.381493
DurationInMicrosecForPerItemMethodM1	267706.293	927580.914	3.807835e+05	190350.431321
DurationInMicrosecForPerUserMethodM1	478819.894	902244.572	5.738828e+05	130482.172800
DurationInMicrosecForBaselineMethodM1	1175765.907	1252437.931	1.206466e+06	25642.552669

The time for computing this adjusted cosine similarity based similarity is much higher than before and this for many reasons. The major one being computing the different similarities. In regard to the suggested implementation (on which my work has extensively relied thinking it must of been an ideal implementation for later steps) could not be implemented using RDDs. In fact, nested RDDs create nested graphs rendering the computation extremely slow when executed. To that end, an implementation relying on scala collections was preferred with Maps more specifically. Each similarity computation involves finding the list of common items between two users and summing the product of their preprocessed ratings, this for all possible user combinations. Then, the prediction itself is slow because of that same use of collection. In essence, the operations realized for computing the prediction are similar to previous milestone baseline method. Yet, here, in order to compute product between similarities and normalized ratings, one has to hover over each user to get his similarities and each item to get the user rating it. A join operation on RDD with appropriate structure might have been faster.

Technical specifications :

- Machine : MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports)
- Processor : 2.3 GHz Intel Core i5 four cores

2.2.7 Similarities time measurements :

Time for computing similarities was measured and referenced in table 5 :

Table 5: Duration in Mirco seconds for computing similarities

Time duration	min(ms)	max(ms)	average(ms)	stddev(ms)
DurationInMicrosecForComputingSimilarities	1.2176E7	1.3190E7	1.2634E7	424186.5862

The average time per $s_{u,v}$ in micro seconds is equal to : 28.4454 (ms/ $s_{u,v}$). Then looking at the ratio between time measurment average computing similarities and prediction, one found that ratio to be equal to : 0.5773. This means that approximately half of the prediction time is due to

the computation of similarities which can therefore be considered as a significant impact on the total computation time.

3 Neighbourhood-Based Predictions

3.1 Questions

3.1.1 Varying the number of nearest neighbors k :

The prediction for the following values of nearest neighbors k were computed : 10, 30, 50, 100, 200, 300, 400, 800, 943. The accuracy (mean absolute error) the also compute and is reported in table 7

Table 6: Prediction accuracy varying the nearest neighbors k:

Number of nearest neighbors	Mean absolute errors
k = 10	0.8407
k = 30	0.7914
k = 50	0.7749
k = 100	0.7561
k = 200	0.7485
k = 300	0.7469
k = 400	0.7474
k = 800	0.7473
k = 943	0.7478

Looking at these results, one notices that the lowest value for the number of nearest neighbors which improves our model compared to the baseline is k =100. The difference in accuracy is of :-0.0108.

To be honest, i have never heard of any of the movies suggested and therefore have no particular opinion about them. After a quick look up on the internet, i think both Star kid and Prefontaine are movies i might actually really like and want to see in the future. For the other ones, i will have to watch to be sure of my opinion.

3.1.2 Storing similarities for k nearest neighbors :

In order to compute the number of bytes required to store only to top k similarities for each user u as a function of the tolal number of users, one relies on the following formula :

$$U * k * numberOfBit/(Bytesequivalent) = 943 * k * (64/8)$$

. One has compute the total number of bytes needed for such storage for the previous values of nearest neighbors k and has reported the result in the following table :

Table 7: Prediction accuracy varying the nearest neighbors k: . Note : looking at the forum it is k = 942 that was used

Number of nearest neighbors	Number of bytes for storage
k = 10	75440
k = 30	226320
k = 50	377200
k = 100	754400
k = 200	1508800
k = 300	2263200
k = 400	3017600
k = 800	6035200
k = 942	7106448

3.1.3 Maximum storage for k nearest neighbors :

Technical specifications :

- Machine : MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports)
- Ram: 8 Go, 8.5899E9 Bytes

To compute the maximum numbers of users one could store in his Ram, the following formula was used (as suggested) :

$$\frac{totalBytesRam}{BytesPerUser} = \frac{totalBytesRam}{k * \#valueStored * numberOfBit/(Bytesequivalent} = \frac{8.5899E9}{k|_{=100} * 3 * 8} = 3579139$$

Therefore, with this Ram capacity and for 100 neighrest neighbors, one can store up to 3579139 users.

3.1.4 Nearest neighbors and similarities :

Varying the number of nearest neighbors will not have any impact on the number of similarities to compute. In order to the return the top k users which maximises the similarity with a given user u, one has to compute the similarity with all other users.

3.1.5 Top 5 recommendations:

Table 8: Top 5 recommendations with k = 30

item number	movie title	rate
20	Angels and Insects (1995)	5.0
165	Jean de Florette (1986)	5.0
166	Manon of the Spring (Manon des sources) (1986)	5.0
236	Citizen Ruth (1996)	5.0
272	Good Will Hunting (1997)	5.0

Table 9: Top 5 recommendations with k = 300

item number	movie title	rate
247	Turbo: A Power Rangers Movie (1997)	5.0
835	Gay Divorcee	5.0
1085	Carried Away (1996)	5.0
1138	Best Men (1997)	5.0
1144	Quiet Room	5.0

It can be observed that the recommendations are completely different from moving from k = 30 to k = 300. None of the item in k = 30 is present in k = 300. I must say that this time, the recommendation include movies i have seen, enjoyed but did not rate. Good will Hunting and Power Rangers being one of them. These predictions are also very different from the previous ones, as they have no item in common. For ease of reference, i put the previous recommendations bellow.

Table 10: Top 5 recommendations baseline

item number	movie title	rate
814	Great Day in Harlem	5.0
1122	They Made Me a Criminal (1939)	5.0
1189	Prefontaine (1997)	5.0
1201	Marlene Dietrich: Shadow and Light (1996)	5.0
1293	Star Kid (1997)	5.0

Table 11: Top 5 recommendations after smoothing

item index	movie title	rate
1189	Prefontaine (1997)	4.679291
1293	Star Kid (1997)	4.679291
1449	Pather Panchali (1955)	4.652425
318	Schindler's List (1993)	4.630964
408	Close Shave	4.627302