Advances in Data Mining: Assignment III.

Haoran Ding
bitdhr@hotmail.com

Petra Kubernatova
pkubernatova@gmail.com

Introduction

The purpose of this assignment was to find similar users of Netflix with the help of LSH. The task was to implement the LSH algorithm with minhashing in Python and apply it to the input data. We were aiming to find pairs of users whose similarity is at least 0.5.

1 Input Data

The data comes from the original Netflix Challenge (www.netflixprize.com). To reduce the number of users (originally: around 500.000) and to eliminate users that rated only a few movies, only users who rated at least 300 and at most 3000 movies were selected.

Additionally, original user ids and movie ids were recoded by consecutive integers, so there are no 'gaps' in the data. The dataset contains 65.225.506 records, each record consisting of two integers: userid and movieid, indicating that the user userid rated the movie movieid.

2 Experiments and Conclusions

We used the banding technique in our implementation of LSH.

We have been experimenting with three parameters. The number of hash functions h, the number of bands b and the number of rows in each band r. We have tried out different values of these three parameters and in the end tested with $h = 25, b = 5, r = 5$. We recorded the time it took to get results (pairs of similar users) and how many results we achieved.

We also added a column which represents an estimate of the real number of similar pairs. For this we used the equation

$$\frac{number\ of\ results}{(1 - (1 - 0.5^r)^b}$$

We based our values of b and r on the assumption that the threshold

$$t \cong (\frac{1}{b})^{\frac{1}{r}}$$

We have noticed that for a certain number of hash functions, the more rows we had in each band, the higher the threshold was and the less time it took to run the pair check. This is consistent with the theory that the less bands you have, the less candidate pairs you get.

h	b	r	time taken	no. of pairs	estimated no. of pairs
25	5	5	22 minutes	49	334
25	5	5	25 minutes	105	715
25	5	5	26 minutes	155	1055
25	5	5	24 minutes	42	286
25	5	5	24 minutes	109	743

Table 1: Results of the experiments

The time it took to run our implementation was usually just around 25 minutes. We found out that to fit in the 30 minute run time limit we have to use 25 hash functions and 5 bands with 5 rows each. The threshold in this case is 0.72, which might be a bit far from the original 0.5, but we were able to achieve a time under 30 minutes, which is what we were aiming for. We also did experiments with 24 hash functions, 6 bands and 4 rows, but it took too much time to run, although the threshold was more close to 0.5 and we got 104 similar pairs.

The mean amount of pairs that we found within our experiments was 92. And the median of the time it took to complete the run was 24 minutes.