Tool: Hive, R

Favorite movie list: 5000 movies which viewed by most accounts.

**1) Collect data from UVH**:

Generate a data set which contains a pair of movie (tmsid) and account(accountId) for every Watch event with durationviewed > 5m in the UVH data. The 5 minutes is a configurable parameter to filter out watch too-short events which should be considered as not watching.

Output size: 2 x 4830510

Output example:

|  |  |
| --- | --- |
| Tmsid | accountId |
| MV003648300000 | 00041b718724b74ed046948656e7cad211 |
| MV003648300000 | 0007e7524c64c56d59ffaf0927dd08b73bb |
| MV000320830000 | 00041b718724b74ed046948656e7cad211 |

Estimated time: ~1h

**2) Init event data**:

Convert tmsid and accountId to tmsIndex and accIndex to be used with the movie-user matrix later. Only keep the movies which belong to the 5000 favourite movie list.

Output size: 2 x 4434326

Output Example:

|  |  |
| --- | --- |
| tmsIndex | accIndex |
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |

Estimated time: ~5m

**3) Compute Cosine Similarity Matrix**:

Step 1: Init movie-user matrix with R’s sparseMatrix:

|  |  |
| --- | --- |
| *Method* | *Estimated time* |
| Read each row of event data and assign the 1 to the corresponding cell in the matrix:  M[tmsIndex,accIndex] <- 1 for each row in event.data. | ~5h |
| Group event.data by tmsIndex, and assign 1 cell as bellow:  M[tmsIndex,accountIndexVector] <- 1 for each tmsIndex. | ~2h |
| Init the matrix with values in construction:  M <- sparseMatrix(I = event.data$tmsIndex, j = event.data$accIndex) | <1m |

Step 2: Compute movie-movie matrix.

|  |  |
| --- | --- |
| *Method* | *Estimated time* |
| Compute cosine similarity for each movie pair with intersect function (don’t need movie-user matrix from step 1). | ~5h |
| Compute similarity by multiply each row with the transpose of the matrix. | ~20m |
| Compute cosine with crossproduct function. | 4s |

Scale-up experiments with dummy data (because of the environment unstable until May 21st ) for the last method in step 2:

|  |  |  |
| --- | --- | --- |
| *User-movie matrix* | *Movie-movie matrix* | *Estimated time* |
| 5000 x 300000 (~19Mb) | 5000 x 5000 (~100Mb) | ~4s |
| 5000 x 1000000 (~57Mb) | 5000 x 5000 (~100Mb) | ~15s |
| 15000 x 300000 (~54Mb) | 15000 x 15000 (~900Mb) | ~37s |

The calculation performance is very good now and scaling up is very feasible now.