INF553
Bufan Zeng
3/25/18

Description

Environment Requirement

Scala: 2.11 Spark: 2.2.1

Dataset: Amazon rating data (*video_small_num.csv, video_small_testing_num.csv*)

Command Line Instructions

The jar file (*Bufan_Zeng_hw3.jar*) contains 4 main classes (*JaccardLSH, ModelBasedCF, UserBasedCF, ItemBasedCF*).

In terminal, get to the spark/bin directory (*spark-2.2.1-bin-hadoop2.7/bin*)

- LSH

To run the Jaccard similarity based LSH algorithm to find the similar items, run the following command in terminal:

./spark-submit --class JaccardLSH Bufan_Zeng_hw3.jar <input file path> < output file path>

In our case:

./spark-submit --class JaccardLSH

Bufan_Zeng_hw3.jar ./video_small_num.csv ./Bufan_Zeng_SimilarProducts_Jaccard.txt

- Collaborative Filtering

To run the collaborative filtering algorithm, run the following command in terminal:

./spark-submit --class <class name> Bufan_Zeng_hw3.jar <rating file path> <testing file path> <output file path>

An example of running model based collaborative filtering algorithm is:

./spark-submit --class ModelBasedCF

Bufan_Zeng_hw3.jar ./video_small_num.csv ./video_small_testing_num.csv ./Bufan_Zeng_ModelBasedCF.txt

Precision and Recall of LSH

Expressions:

Precision = true positives / (true positives + false positives)

Recall = true positives / (true positives + false negatives)

- Jaccard based LSH:

  Precision = 51111 / (51111 + 0) =1

  Recall = 51111 / (51111 + 0) =1

- Cosine based LSH:

  Precision = 58547 / (58547 + 3724) = 0.940196881373352

  Recall = 58547 / (58547 + 17668) = 0.7681821163812897

Baseline of CF

- User-based CF
  >=0 and <1: 4093
  >=1 and <2: 2413
  >=2 and <3: 833
  >=3 and <4: 331
  >=4: 30
  RMSE: 1.4062215447360011
  Time: 15 sec

- Model-based CF
  >=0 and <1: 4717
  >=1 and <2: 2086
  >=2 and <3: 533
  >=3 and <4: 364
  >=4: 0
  RMSE: 1.288234875454538
  Time: 9 sec

- Item-based CF
  >=0 and <1: 4117
  >=1 and <2: 2311
  >=2 and <3: 808
  >=3 and <4: 401
  >=4: 63
  RMSE: 1.4193201273235432
  Time: 34 sec

- Item-based CF without LSH
  >=0 and <1:3501
  >=1 and <2:2280
  >=2 and <3:880
  >=3 and <4:470
  >=4:569
  RMSE: 2.2018845666007594
  Time: 50 sec

Implementing LSH in CF

Comparing the results of item-based CF, we observe a significant improvement in RMSE and processing time when using LSH. The LSH helps to identify the similar items first and thus can save the time of computing all the Pearson similarities between all the product (only need to calculate those similar candidates). Therefore, it

not only helps to reduce the computation cost, but also performs the function of filtering out the pairs that are not similar so that the RMSE and processing time can be improved.