Description

Xiaoyu Zheng

1. Directory structure

In Solution directories, there are 2 folders, which contain the python one and the scala one separately. In the python folder, there are four python files, while in the scala folder, there are four scala files and a executable jar. In OutputFiles folder, there exist all the output txt files required in all three problems folders. As the output files from the python solution and the scala solution are different, I put five output text files in each folder.

2. Versions

Spark: 2.2.1 Python: 2.7.15 Hadoop: 2.7 Scala: 2.11.8 SBT: 0.13.8

3. Execution rules

1) Put my **executable files** (including all python files and the scala jar file) directly **in the root directory of Spark**.
2) To utilize my work, please **get into the root directory of spark** with terminal or in the command line.
3) Then to execute my work, you need the following commands. Note: These commands will overwrite the files which have the same name as my output files in the root directory of Spark.

   Task1:
   Python:
   bin/spark-submit Xiaoyu_Zheng_task1_Jaccard.py <rating file path>
   Scala:
   bin/spark-submit --class JaccardLSH Xiaoyu_Zheng_hw3.jar <rating file path>

   Task2:
   Model-Based:
   Python:
   bin/spark-submit Xiaoyu_Zheng_task1_ModelBasedCF.py <rating file path> <testing file path>
   Scala:
   bin/spark-submit --class ModelBasedCF Xiaoyu_Zheng_hw3.jar <rating file path> <testing file path>

   User-Based:
   Python:
   bin/spark-submit Xiaoyu_Zheng_task1_UserBasedCF.py <rating file path> <testing file path>
   Scala:
   bin/spark-submit --class UserBasedCF Xiaoyu_Zheng_hw3.jar <rating file path> <testing file path>

   Item-Based:

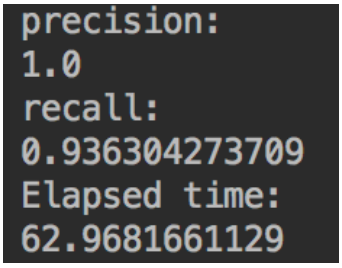
Python:
bin/spark-submit Xiaoyu_Zheng_task1_ItemBasedCF.py <rating file path> <testing file path>
Scala:
bin/spark-submit --class ItemBasedCF Xiaoyu_Zheng_hw3.jar <rating file path> <testing file path>

4) The output txt files will also be in the root directory of Spark.

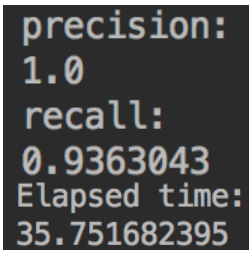
4. Evaluation result

1) Task1:

Python:
Precision: 1.0 Recall: 0.936304273709



Scala:
Precision: 1.0 Recall: 0.9363043



2) Task2:

Python:

	Model-Based		User-Based	Item-Based
	Small	Large	Small	Small
>=0 and < 1	13822	3243782	14590	13319
>=1 and < 2	4873	704590	4536	5587
>=2 and < 3	1248	93522	995	1132
>=3 and < 4	274	11725	128	209
>=4	39	832	7	9
RMSE	1.06919282307	0.828639671855	0.997322507642	1.03800625483
Time/Sec	22.4811880589	1826.14945006	48.4460120201	927.076499939

Scala:

	Model-Based		User-Based	Item-Based
	Small	Large	Small	Small
>=0 and < 1	13771	3233407	14291	13321
>=1 and < 2	4908	717699	4706	5589
>=2 and < 3	1278	91869	1081	1132
>=3 and < 4	252	10752	159	206
>=4	47	724	19	8
RMSE	1.069233111011328	0.8291243593404478	1.0280798659475594	1.0369691491962536
Time	11.418476848	435.810732413	29.033931053	336.570902438

5. Comparison for Item-based CF with and without LSH

With LSH, the item-based CF program runs much faster than the one without LSH. Because LSH finds out the similar movie pairs first, then to calculate the pearson correlation between the pairs. Though the item-based CF program also has the neighborhood part to only predict the results from several most similar items data, and to lower the tail effect from other items which do not resemble to the target item, so the precision for item-based CF with and without LSH should differ significantly. However, without LSH, we need to calculate all item pairs which have been rated by same user, and its number can be much larger than the number of similar movie pair found by LSH.