README.md

PySpark Recommender System

Project Description

The purpose of this project is to become familiar with big data processing tools, specifically, PySpark. Through the large-scale collaborative-filtering project and mini-applications implemented in this repo, I was able to gain a solid understanding of data handling and model formulation using PySpark.

In this repository, I implement simple map-reduce programs for counting odd/even integers, counting words in a text document, and calculating simple properties present in a supplied large data-frame.

I also implement a large-scale collaborative-filtering recommender system. This system takes in user rating data pertaining to various movie IDs and models the relationship between the users by utilizing the collaborative filtering approach with alternating least squares (ALS).

Setup

Organization

The scripts for each of the PySpark applications are present in the `scripts/` folder. The results for each of the implementations are in the `results/` folder. These results have also been appended to the end of the README in image format. The outputs from each script have been included in the `outputs/` folder.

Conda Environment Setup

Create environment from environment.yml:
From base directory:
```
conda env create -f ./environment.yml
```

Update environment from environment.yml:
From base directory and after activating existing environment:
```
conda env update --file ./environment.yml
```

Execution

After creating a conda environment using the supplied `environment.yml` file, inspect the files within the `scripts/` folder to understand the purpose of each PySpark application. Then, simply run each of the scripts and observe the output.

NOTE: the `recommender_sys.py` file has a `question` flag that enables different branches of the script to run. Be sure to set this flag if you wish to control the flow of the script

Insights

RMSE vs. MSE

The mean-squared error (MSE) is the mean of the squared difference between the predictions and the targets. The root mean-squared error (RMSE) is the square root of the MSE. The difference between these two metrics is subtle. When the error gets large, the MSE penalizes this difference more aggressively than the RMSE because the RMSE is the square root of the MSE. Furthermore, when the error is less than 1, the MSE penalizes the difference between the actual and the predicted less aggressively than the RMSE.

In our context, we wish to heavily punish the model if it predicts outlandish ratings that are far away from the true value. Furthermore, we also wish to "go easy" on the model when it is within a star. Therefore, the MSE would be a good choice in this situation. Also, since RMSE requires taking the square root of the MSE, this may result in minor timing disadvantages.

ALS Parameter Tuning

The parameters that I focused on for tuning were rank, regParam, and alpha. Rank is the total number of latent factors used by the model. The default value for rank is 10. After reading online that rank can vary from 5-200, I chose to test the following rank values: `[5, 10, 20, 40, 80]`. The regParam is the regularization parameter used by the ALS model. I decided to vary the regularization parameter using the following values: `[0.1, 0.01, 0.001]`. These are typical values used for regularization in other applications. Furthermore, after reading a little bit about other PySpark recommender systems, these values seemed to be commonplace. I also decided to vary the alpha parameter. This parameter controls the implicit feedback variant of ALS. It determines the model's baseline confidence with respect to its observations. I decided to vary this model using these values: `[2, 3]`. Using the built-in PySpark libraries, grid-search was conducted and an optimal model was extracted from these parameters for train/test splits of 75/25 and 80/20.

User-Specific Movie Recommendations

Using the optimal model from the grid-search, movie recommendations were made for the user with ID 11 and the user with ID 23. Note that the movies that these two users had already reviewed were removed from the input data so that already seen movies weren't recommended.

Result Images

A.1 Odds/Evens

```
Number of odd numbers = 496
Number of even numbers = 514
```

A.2 Salary Department

```
The individuals in the Sales department were paid a total of $3,488,491
The individuals in the Research department were paid a total of $3,328,284
The individuals in the Developer department were paid a total of $3,221,394
The individuals in the QA department were paid a total of $3,360,624
The individuals in the Marketing department were paid a total of $3,158,450
```

A.3 & A.4 MapReduce Word Count

```
The word "GUTENBERG" appeared 100 times
The word "COLLEGE" appeared 98 times
The word "LIBRARY" appeared 99 times
The word "SHAKESPEARE" appeared 101 times
The word "THIS" appeared 104 times
The word "WORLD" appeared 98 times
The word "WILLIAM" appeared 128 times
```

```
The top 20 words are as follows:
#1: "the" appeared 11412 times
#2: "I" appeared 9714 times
#3: "and" appeared 8942 times
#4: "of" appeared 7968 times
#5: "to" appeared 7742 times
#6: "a" appeared 5796 times
#7: "you" appeared 5360 times
#8: "my" appeared 4922 times
#9: "in" appeared 4803 times
#10: "d" appeared 4365 times
#11: "that" appeared 3864 times
#12: "And" appeared 3735 times
#13: "is" appeared 3722 times
#14: "not" appeared 3595 times
#15: "me" appeared 3448 times
#16: "s" appeared 3398 times
#17: "his" appeared 3278 times
#18: "with" appeared 3221 times
#19: "it" appeared 3078 times
#20: "be" appeared 2986 times
```

```
The bottom 20 words are as follows:
#1: "anyone" appeared 1 time
#2: "restrictions" appeared 1 time
#3: "License" appeared 1 time
#4: "online" appeared 1 time
#5: "www" appeared 1 time
#6: "gutenberg" appeared 1 time
#7: "org" appeared 1 time
#8: "COPYRIGHTED" appeared 1 time
#9: "Details" appeared 1 time
```

```
#9:  "Details" appeared 1 time
#10: "guidelines" appeared 1 time
#11: "Title" appeared 1 time
#12: "Author" appeared 1 time
#13: "Posting" appeared 1 time
#14: "September" appeared 1 time
#15: "EBook" appeared 1 time
#16: "Release" appeared 1 time
#17: "January" appeared 1 time
#18: "Character" appeared 1 time
#19: "encoding" appeared 1 time
#20: "START" appeared 1 time
```

B.1 Describe + Top 10 Movies/Users

```
+-------+------------------+------------------+------------------+
|summary|           movieId|            rating|            userId|
+-------+------------------+------------------+------------------+
|  count|              1501|              1501|              1501|
|   mean| 49.40572951365756|1.7741505662891406|14.383744170552964|
| stddev|28.937034065088994| 1.187276166124803| 8.591040424293272|
|    min|                 0|                 1|                 0|
|    max|                99|                 5|                29|
+-------+------------------+------------------+------------------+


+-------+------------------+
|movieId|       avg(rating)|
+-------+------------------+
|     32|2.9166666666666665|
|     90|            2.8125|
|     30|               2.5|
|     94| 2.473684210526316|
|     23| 2.466666666666667|
|     49|            2.4375|
```

```
|     49|             2.4375|
|     18|                2.4|
|     29|                2.4|
|     52| 2.357142857142857|
|     62|               2.25|
+-------+-------------------+
only showing top 10 rows
```

```
+------+------------------+
|userId|       avg(rating)|
+------+------------------+
|    11|2.2857142857142856|
|    26| 2.204081632653061|
|    22|2.1607142857142856|
|    23|2.1346153846153846|
|     2|2.0652173913043477|
|    17|1.9565217391304348|
|     8|1.8979591836734695|
|    24|1.8846153846153846|
|    12|1.8545454545454545|
|     3|1.8333333333333333|
+------+------------------+
only showing top 10 rows
```

B.2 Collaborative Filtering Initial Implementation

```
RMSE = 1.021 & Accuracy = 45.23% with [0.75, 0.25] Train/Test split

Prediction Summary:
+-------+------------------+------------------+------------------+------------------+
|summary|           movieId|            rating|            userId|        prediction|
+-------+------------------+------------------+------------------+------------------+
|  count|               409|               409|               409|               409|
|   mean| 49.97066014669927|  1.80440097799511|14.163814180929096|1.5014046736057018|
| stddev|28.196076764441454|1.1593102224011254| 8.436037375026123|0.7623964320002299|
|    min|                 0|                 1|                 0|      -0.068828106|
|    max|                99|                 5|                29|         4.6778164|
```

```
+-------+------------------+------------------+------------------+------------------+

Prediction Samples:
+----------+
|prediction|
+----------+
|0.45544553|
| 1.5363644|
| 1.1170142|
| 1.0456172|
| 1.8243765|
| 1.0101541|
|0.97592133|
| 1.0299788|
|   1.39473|
| 1.0714704|
| 0.9807167|
| 0.9704875|
| 1.4610391|
| 2.4848142|
| 2.4638247|
| 0.8294847|
|0.08758587|
| 1.3699824|
|0.63840675|
| 1.7418337|
+----------+
```

```
RMSE = 1.077 & Accuracy = 46.15% with [0.8, 0.2] Train/Test split

Prediction Summary:
+-------+------------------+------------------+------------------+------------------+
|summary|           movieId|            rating|            userId|        prediction|
+-------+------------------+------------------+------------------+------------------+
|  count|               299|               299|               299|               299|
|   mean| 45.32107023411371| 1.862876254180602|13.538461538461538|1.5772097781549728|
| stddev|29.141818084444605|1.2576983430932391| 8.571512770931427|0.8170460983012805|
|    min|                 0|                 1|                 0|        0.15136692|
|    max|                99|                 5|                29|         4.8983245|
+-------+------------------+------------------+------------------+------------------+
```

```
Prediction Samples:
+----------+
|prediction|
+----------+
| 1.2023145|
| 1.7353871|
| 1.9104246|
| 1.4926156|
|  2.484839|
|0.87266463|
| 1.3010404|
| 0.6278953|
| 1.0593654|
| 1.7290851|
| 2.3464236|
|0.79799676|
| 0.8437349|
| 0.6881512|
| 0.7799513|
|0.50047743|
| 0.4663005|
| 1.8974373|
|  2.089991|
| 1.3488197|
+----------+
```

B.3 RMSE vs. MSE

```
RMSE = 1.047 with [0.75, 0.25] Train/Test split
MSE = 1.013 with [0.75, 0.25] Train/Test split
RMSE = 1.085 with [0.8, 0.2] Train/Test split
MSE = 0.974 with [0.8, 0.2] Train/Test split
```

B.4 Collaborative Filtering CV

```
Optimal Model:
rank = 40
RMSE = 1.009
Train/Test split = [0.75, 0.25]

Optimal Model:
rank = 40
RMSE = 1.03
Train/Test split = [0.8, 0.2]
```

B.5 Top 15 Movies for User 11 and User 23

```
User ID = 11 Predictions:
+-------+------+----------+
|movieId|userId|prediction|
+-------+------+----------+
|     55|  11.0|  3.141249|
|     49|  11.0| 3.0094433|
|     46|  11.0| 2.9218156|
|     33|  11.0| 2.8364775|
|     65|  11.0| 2.7801595|
|     87|  11.0| 2.6918771|
|     93|  11.0| 2.6707416|
|     17|  11.0| 2.6287627|
|     34|  11.0| 2.4312875|
|     74|  11.0| 2.2958617|
|     73|  11.0|  2.278268|
|      8|  11.0| 2.1717615|
|     96|  11.0| 2.1711764|
|      7|  11.0| 2.0749292|
|     44|  11.0| 2.0611234|
+-------+------+----------+
only showing top 15 rows

User ID = 23 Predictions:
+-------+------+----------+
|movieId|userId|prediction|
+-------+------+----------+
|     17|  23.0|  4.451287|
|     46|  23.0| 4.3385634|
|     90|  23.0| 3.8812928|
|     94|  23.0| 3.4361098|
|     19|  23.0| 2.7770422|
|     16|  23.0| 2.6142771|
```

```
|      7|  23.0| 2.5311217|
|     35|  23.0| 2.5044208|
|     81|  23.0| 2.4542487|
|     79|  23.0| 2.4086957|
|     56|  23.0| 2.3643012|
|     91|  23.0| 2.2758536|
|     51|  23.0| 2.0649421|
|      1|  23.0| 2.0324259|
|     98|  23.0| 1.9882916|
+-------+------+----------+
only showing top 15 rows
```