School of Computer Science
Dr. Ying Zhou

COMP5349: Cloud Computing **Sem. 1/2019**

Assignment 2: Online Review Analysis

Group Work: 20% **02.05.2019**

1 Introduction

Assignment 2 is a group project. In this project you will work on the online review data set released by Amazon and go through various analytic phases, including getting summary statistics, removing unwanted data and performing similarity analysis. The phases and sequence are typical in many real world data analytic workloads. The objective is to test your ability to apply big data framework in realistic setting.

2 Data Set

The complete data set contains product reviews and metadata from Amazon between 1995 and 2015. The data is available in multiple compressed TSV files in the `amazon-reviews-pds` S3 bucket in AWS US East Region. Files are separated by product category. The format and index of files can be found from Amazon Review Data format and File Index. Most category's data are saved in a single file. For instance, file `amazon_reviews_us_Watches_v1_00.tsv.gz` contains all reviews of watches on Amazon US site. Some category such as `book` contains a large number of reviews and they are stored in many files.

Many features of reviews are included in the data set. These include customer id, review headline, review body, rating, helpfulness votes and so on. Some meta data about the product are also included.

The project asks you to analyze data from a single category. In subsequent sections, data set refers to data from a single category, not the complete data set. You can choose **ONE** category from the following:

- `Music`: 1 file; zipped size is 1.4G
- `PC`: 1 file; zipped size is 1.4G
- `Video_DVD`: 1 file; zipped size is 1.4G
- `Wireless`: 1 file; zipped size is 1.6G
- `Digital_Ebook_Purchase`: 2 files; zipped size 2.6 and 1.2G each

- `Books`: 3 files; zipped file size 2.6G, 2.5G and 1.2G each

The file name has the format `amazon_reviews_us_<category>_v1_00.tsv.gz` for single file category. Multiple file category would have 00, 01, and/or 02 as the last two digits for various files.

The following features will be involved in the analysis: `customer_id`, `product_id`, `star_rating`, `review_id` and `review_body`.

3 Stage One: Overall statistics

In this stage you are asked to produce overall summary statistics of the data set, in particular, statistics related with distribution of reviews by user, and by product. The user and product are uniquely identified by `customer_id` and `product_id` respectively.
You are asked to find out the following numbers:

- the total number of reviews
- the number of unique users
- the number of unique products

For user-review distribution, you are asked to find out:

- the largest number of reviews published by a single user
- the top 10 users ranked by the number of reviews they publish
- the median number of reviews published by a user

For product-review distribution, you are asked to find out:

- the largest number of reviews written for a single product
- the top 10 products ranked by the number of reviews they have
- the median number of reviews a product has

4 Stage Two: Filtering Unwanted Data

In this stage, you are asked to filter reviews based on length, reviewer and product feature.
In particular, the following reviews should be removed:https://www.overleaf.com/project/5cbe60ce05f5

- reviews with less than two sentences in the `review_body`.
- reviews published by users with less than median number of reviews published

- reviews from products with less than median number of reviews received

Note that sentence segmentation can be done in various ways. You may use simple punctuation character like period or question mark to segment the review body, or use tools from NLP packages. We do not expect each group to generate the same output.

After filtering out the above, find out:

- top 10 users ranked by median number of sentences in the reviews they have published;
- top 10 products ranked by median number of sentences in the reviews they have received;

5 Stage Three: Similarity analysis with Sentence Embedding

In this stage, you are asked to perform similarity analysis on the review sentences. The analysis involves segmenting `review_body` into multiple sentences; encoding each sentence as vector so that the distance between pair of sentences can be computed.

5.1 Positive vs. Negative Reviews

For a given product, consider all reviews with `star_rating` 4 and above as positive reviews; and all reviews with `star_rating` 2 and below as negative reviews. You are asked to pick a product from the top 10 products you find in stage One. The positive class is constructed by

- extracting all reviews with rate 4 and above
- for each review, extracting the `review_body` part and segment it into multiple sentences.

The negative class is constructed in similar manner except that we extract all reviews with rate 2 and below.

Each sentence in the clusters is then embedded with Google Pre-trained universal sentence encoder. The result is a 512 dimension vector.

5.2 Intra-Class Similarity

We want to find out if sentences in the same category are closely related with each other. The closeness is measured by **average distance between points in the class**. In our case, point refers to the sentence encoding and pair-wise distance is measured by Cosine distance. Cosine distance is computed as "$1 - CosineSimilarity$". It has a value between 0 and 2.

5.3 Class Center Sentences

Find out the class center and its 10 closest neighbours for positive and negative class respectively. We define class center as the point that has the smallest average distance to other points in the class. Again in this case point refers to the sentence encoding and pair-wise distance are measured by Cosine distance.

The result should show the text of the center sentence, the `review_id` it belongs to and its 10 closest neighbouring sentences text and their respective `review_id`.

6 Stage Four: Similarity analysis with Spark supported Feature Extractors

In this stage, you are asked to perform the same similar analysis as in stage three, with different embedding mechanism. You are asked to use one of the Spark supported feature extractors, such as `TF-IDF` or `word2Vec` to convert review sentences into vectors before carrying out the similarity analysis.

7 Group Collaboration

This is a group assignment. Each group can have up to **FOUR** members. You are asked to use git to collaborate among members, you can use the university's hosting service, GitHub or BitBucket. Once you have created the repository on a hosting platform, you should give your tutor access to your repository.

Group members are expected to make fair contribution to the project. You are NOT recommended to divide the work based on stages as the amount of work required in each stage varies a lot. If members of your group do not contribute sufficiently you should alert your tutor as soon as possible. The tutor has the discretion to scale the group's mark for each member based on their contribution to the project.

8 Coding Requirements

All stages must be implemented with Spark. You can use a combination of Spark API. Your Spark task can load external packages or libraries to perform simple task such as sentence segmentation, or distance calculation.

We use stage to define various workload. This does not necessary mean you need to implement each in a separate script. It is possible to implement two stages in one script, or to implement one stage in two or more scripts.

You code must run on EMR cluster. You can decide on the capacity of the cluster based on your data set and implementation. You can store intermediate results back on S3, provided they are generated by Spark application.

9 Deliverable

There are two deliverables: **source code** and **project report** (less than or equal to 10 pages). Both are due on **Wednesday 22nd of May 23:59 (Week 12)**. There will be a **demo** in week 12 during tutorial time. **ALL** members of a group must attend the demo and explain individual contribution. Group members who do not attend the demo will not receive any mark, unless he(she) has been granted special permission for not attending from course coordinator.

There will be different links for source code and report submission to facilitate plagiarism detection. The marker may need to run your code on their environment, make sure you prepare one or multiple `read.me` files with details on how to run your workload. Put the source code and associated `read.me` into a zip file. Remember, only the source code and `read.me` file should be submitted. **No data file or compiled version should be included.**

Report should be submitted as a PDF file.

A hard copy of the report with signed group assignment cover page should be submitted during the demo. All member should sign the cover page.

Every student can only join one group, and each group can only demo in one tutorial.

10 Report Structure

The report should contain five main sections: one for each stage and a section to compare performance of stage 3 and 4. You may include an introduction section at the beginning and an appendix at the end to complete the look.

The sections represent stage one and two should be short. You only need to give a brief description of the implementation and show the result in tabular format. No diagram is required.

The sections represent stage three and four should contain details of the design and execution statistics. You should highlight performance optimization features you use in the implementation. These may include caching, partition number choice and so on. A performance analysis should be included in each section. Execution screen shots can be included for performance analysis. The result should be included as well.

There should be section comparing the performance of stage three and four. The two stages represent different implementations of the same workload. Each implementation consists of a few steps. The performance comparison should contain details on each step.

11 Reference

- Amazon Customer Reviews Dataset