Hadoop Map-Reduce – Mini Project

Question

Implement Movie Recommendation System.

Objective

Using given dataset, find Movie Recommendations using Hadoop MapReduce program.

Dataset

This processing is to be done on the real world Movie Lens dataset. The data sets were collected over various periods of time, depending on the size of the set. It contains 100,000 ratings from 1000 users on 1700 movies. Users were selected at random for inclusion. Users are represented by its id and item are also represented by id. Format for the file is as follows.

Data File Structure

All ratings are contained in the DATA file. Each line of this file represents one rating of one movie by one user, and has the following format:

```
UserID,MovieID,Rating,Timestamp
17,70,3,0
35,21,1,0
49,19,2,0
49,21,1,0
49,70,4,0
87,19,1,0
87,21,2,0
98,19,2,0
```

The lines within this file are ordered first by UserID, then by MovieID.

Ratings are made on a 5-star scale, with half-star increments.

Timestamp represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970. In above example they are given 0.

Description

What is Recommendation System?

Recommendation system are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that user would give to an item. Recommender systems have become extremely common in recent years, and are applied in a variety of applications. The most popular ones are probably movies, music, news, books, research articles, search queries, social tags, and products in general.

Big Data – Hadoop

Imagine that you own an online movie business, and you want to suggest for your clients movie recommendations. Your system runs a rating system, that is, people can rate movies with 1 to 5 stars.

Our goal is to calculate how similar pairs of movies are, so that we recommend movies similar to movies you liked. Using the correlation we can:

- For every pair of movies A and B, find all the people who rated both A and B.
- Use these ratings to form a Movie X vector and a Movie Y vector.
- Calculate the correlation between those two vectors.
- When someone watches a movie, you can recommend the movies most correlated with it.

You want to compute how similar pairs of movies are, so that if someone watches the movie The Matrix, you can recommend movies like BladeRunner. So how should you define the similarity between two movies?

One possibility is to compute their correlation. The basic idea behind it is for every pair of movies X and Y, find all the people who rated both X and Y. Use these ratings to form a Movie X vector and a Movie Y vector. Then, calculate the correlation between these two vectors. Now when someone watches a movie, you can now recommend him the movies most correlated with it.

Our task is to find similarity between pair of item using correlation formula.

Similarity(X, Y) = Correlation(X, Y)

X and Y are items

Steps:

1. For pair of items find the users rated both the items X and Y.
2. Form two vectors X and Y. For Example:

User	Movie X	Movie Y
User 1	1.0	2.0
User 2	5.0	4.0
User 3	4.0	5.0
User 4	3.0	2.0
User 5	1.0	3.0

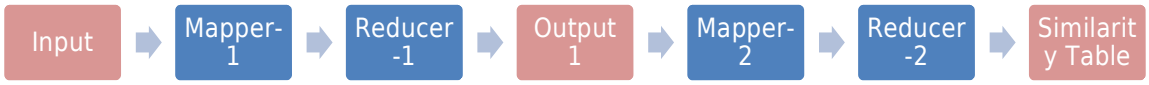
3. Calculate correlation between X and Y using following formula.
   Correlation(X, Y) =

$$\frac{n\sum xy - \sum x \sum y}{\sqrt{n\sum x^2 - \left(\sum x\right)^2}\sqrt{n\sum y^2 - \left(\sum y\right)^2}}$$

   Here n is number of users rated both the items, x and y are rating values between 1.0 to 5.0.

Solution:

To solve the problem we are using chaining of two MapReduce job. Output of the first job is given as input to the second.



MapReduce Job-1: Work of the first MapReduce job is to collect all the users rated both the items.

MapReduce Job-2: Second MapReduce job will find the similarity between items using correlation formula.

Input: This is a CSV file with columns as UserID, MovieID, Rating, Timestamp.

```
17,70,3,0
35,21,1,0
49,19,2,0
49,21,1,0
49,70,4,0
87,19,1,0
87,21,2,0
98,19,2,0
```

Mapper-1 Output: This is a map with UserID as key and (MovieID, Rating) pair as value.

```
17    70,3.0
35    21,1.0
49    19,2.0
49    21,1.0
49    70,4.0
87    19,1.0
87    21,2.0
98    19,2.0
```

Reducer-1 Output (Output1): This is a text file containing UserId as key and all (MovieID, Rating) pairs by that user.

```
17    70,3.0
35    21,1.0
49    70,4.0,21,1.0,19,2.0
87    21,2.0,19,1.0
98    19,2.0
```

Mapper-2 Output: This will drop UserId entirely. The output is map with (MovieId1, MovieId2) pair as key and It's (Rating1, Rating2) pair as value.

```
21,70	1.0,4.0
19,70	2.0,4.0
19,21	2.0,1.0
19,21	1.0,2.0
```

Reducer-2 Output: This is a text file containing similarity table. It contains (MovieId1, MovieId2) pair as key and (RatingCount, Similarity) as value.

```
19,21	2,-1.0
19,70	1,0.0
21,70	1,0.0
```