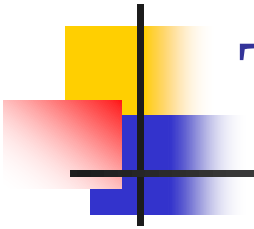
Movie Recommendations using Map-Reduce

Rakshith Muniraju (A20344305)

J V P S Avinash (A20344397)



The Problem

- Recommendation systems are quite popular among movie sites, and other social network systems these days.
- We need the user-data interaction details like items, movies watched and rating given and are available from various sites.
- To find the similarity between two pair of items, we need to find the correlation between them.
- Since the correlation data would be sparse and time-varying, we need the calculations to be done periodically so that the results are up to date. Moreover, the framework needs to handle a lots and lots of data.
- So, we need to have a *divide-and-conquer* pattern, to handle this scenario.
- **Map-Reduce** is the solution !!!!!!!

The Approach - MapReduce

- MapReduce is the frame useful for large scale distributed computations across various domains. It can handle petabytes of data.
- *mrjob* is a Python package useful for running Hadoop Streaming jobs. With this, we can test our code locally without installing Hadoop.
- We calculate how similar pairs of movies are, so we can know how likely person will watch that movie based on the recommendation.
- For every pair of movies A and B, we take the ratings given to both the movies and form vectors for both of them.
- Next, we find the similarity using following methods : Correlation, Cosine, Regularized Correlation and Jaccard Similarity measures.
- When someone watches a movie, you can now recommend him the movies most correlated with it.

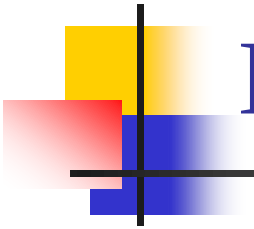
The Approach – MapReduce (Cont.)

- *Map Function*

  Takes user input from the file (user, movie, rating) and for each user emits a row of key-value pairs, where key is user and value is a list of (item, rating).

- *Reducer Function*

  Takes the input from the mapper and for each user emits a set of key-value pairs, where key is user and value is a list of (ratings sum, movies rated count).



Data

- At present, we are extracting the data from MovieLens database which contains the user, item, ratings information.
- We might club/merge the data from these files to make it meaningful.
- Initially, we thought of using NetFilx API, but we came to know that they don’t support the public development APIs anymore.
- Also we registered with “Rotten Tomatoes” and yet to receive the configuration file from them to extract the data set(This is the second option we are considering).

Timeline

Title	Time
Data Extraction and Analysis	10/25/2015
Pre-processing data to meet our needs	10/30/2015
Study on Map-Reduce and Similarity Measures	11/07/2015
Map and Reduce Function Implementation	11/14/2015
Similarity Measure Implementation	11/20/2015
Comparisons and Analysis	11/23/2015
Result Generation	11/30/2015