**CS6350: BIG DATA ANALYTICS and MANAGEMENT**

**Fall 2015**

**HW #4**

**Related to: Spark, Data Analytics and Recommendation System**

**Due: Nov 29, 2015 (11.55 p.m.)**

This homework consists of two parts. Here, we focus on K-means clustering (data analytics) and recommendation systems.

You will be using the movie dataset for this homework. The dataset is attached with the homework. The readme.txt file contains more information about the dataset.

**Part A:**

**Q1**

Implement the k-means algorithm from the scratch using SCALA or PYTHON on SPARK. Using your k-means program, **cluster the movies using the ratings given by the user,** that is, use the item-user matrix from **itemusermat File provided** as input to your program.

**Dataset description.**

**Dataset: Itemusermat File.**

The **itemusermat file contains** the ratings given to each movie by the users in **Matrix format.** The file contains the ratings by users for 1000 movies.

Each line contains the movies id and the list of ratings given by the users.

A rating of 0 is used for entries where the user did not rate a movie.

From the sample below, user1 did not rate movie 2, so we use a rating of 0.

A sample **Itemusermat file** with the item-user matrix is shown below.

|  |  |  |
| --- | --- | --- |
|  | user1 | user2 |
| movie1 | 4 | 3 |
| movies2 | 0 | 2 |

Set the number of clusters (**k**) to 10

Your Scala/python code should produce the following output:

* For each cluster, **print any 5 movies in the cluster. Your output should contain the movie\_id, movie title, genre and the corresponding cluster** it belongs to. **Note:** Use the **movies.dat** file to obtain the movie title and genre.

For example

**cluster: 1**

**123,Star wars, sci-fi**

**Q2.**

Given any movieid as input, write a scala or python code to output the 5 most similar movies (include **movie\_id, movie title, genre**) using the item-user matrix provided in **itemusermat File**. Use **Pearson correlation coefficient** as your similarity measure. **Please include in your output, the movie title and genre of the movie id used as input to your program. Your program should take input from the command line.**

**Part B**

**Q3a.**

**Recommendation system using Matrix factorization with Alternating Least Squares (ALS).**

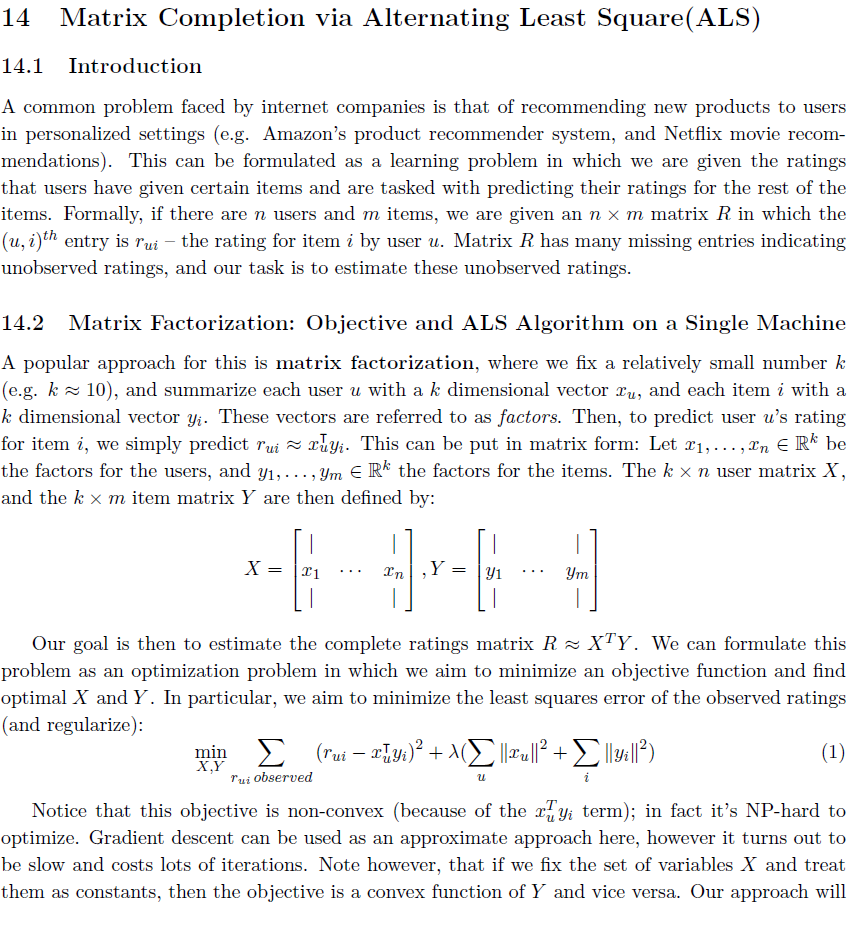
**Description of the approach can be found at:**

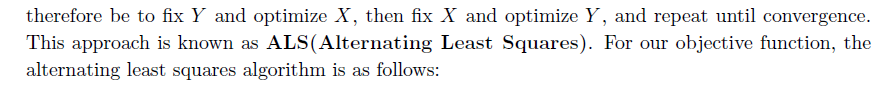
**Source: Stanford Distributed Algorithms and Optimization class.**

http://stanford.edu/~rezab/dao.

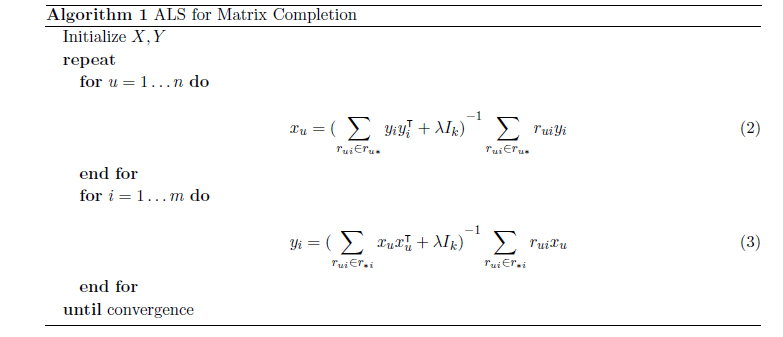
Instructor: Dr. Reza Zadeh, Databricks and Stanford.

Here is the relevant part:

****

****

**Given below is the ALS algorithm.**

****

Equation 2 updates the latent vectors for each user. Equation 3 updates the latent factors for each item.

A Scala implementation of Equation 2 has been provided in a file named **als.scala** (attached and description using the distributed algorithm is provided in **section 14.3** –above). Each line of the code is commented and it provides reference to the provided document.

**You will be required to fill in the code that implements Equation 3 in the given als.scala file to update the latent vectors for each movies/item.**

**You will be using ratings.dat as input to your program.**

Given below is the Dataset description:

**ratings.dat**

**UserID::MovieID::Rating::Timestamp**

- UserIDs range between 1 and 6040

- MovieIDs range between 1 and 3952

- Ratings are made on a 5-star scale (whole-star ratings only)

- Timestamp is represented in seconds since the epoch as returned by time(2)

- Each user has at least 20 ratings

**movies.dat**

**MovieID::Title::Genres.**

**Output:**

**1. Print out the learned latent vector for user id 1,1757 and 1759.**

**2. Print out the learned latent vectors for movies 914, 1777 and 231.**

**Note: Your code should be able to retrieve any of the latent vectors by taking the user\_id or movie\_id from command line argument.**

**Q3b**

**Using the factorized matrix obtained from Q3a, Provide the predicted rating for the given users and movies below.**

user 1 and movieid 914,

user 1757 and movieid 1777,

user 1759 and movieid 231.

***Submission***:

Please upload your submission via e-learning. Please upload the following to eLearning:

1. Submit the answer to Q1 as Q1.txt
2. Q2 as Q2.txt
3. Submit the answer to Q3 as completed als.scala file and Q3b answer.

If you use python/scala, then submit all source files.