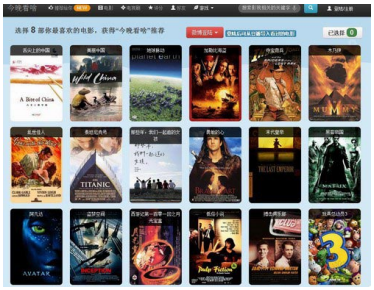
Collaborative Filtering: Movie Recommandation

Nicolas Pécheux
Alexandre Allauzen
Guillaume Wisniewski

January 2015



This assignment will be graded and is due Sunday February 1 at 23h59. You should write a short report (up to 4 pages) explaining what you have done, showing your results, providing answers to the questions and insightful comments for any difficulty you encountered. You may work by pairs and submit joint work. Send your report only as a *Portable Document Format* `lastname1_lastname2.pdf`. Your code will not be assessed, but should be provided packaged as a bitstream archive `lastname1_lastname2.tar.gz`. Send your submission to your lab instructor, either `nicolas.pecheux@limsi.fr` or `elena.knyazeva@limsi.fr`.

1 Principle

The goal of this exercise is to implement a recommendation system for movies using the corpus MovieLens. The corpus is available on the website of the course. Each line of this corpus contains one judgement from a specific user for one specific movie. The version you will use is simplified and preformated as follows :

- one judgement per line
- for each line, three fields in this order: the user id, the movie reference, the rating;
- fields are separated by the character `|`.

2 Preliminary corpus analysis

1. Extract the total number of judgment, the total number of different user, and the number of different movies. How do you handle equal judgment lines ?
2. What is the most recent movie ? and the oldest one ?
3. Compute the mean and the standard deviation of the ratings.
4. Represent the distributions of the ratings.[1]
5. Compute the mean and the standard deviation of the number of judgements per user, along with the maximum and minimum number of judgment per user.

3 Recommandation system

The proposed recommendation system is based on the estimated similarity among movies. To recommend a movie to a user, we therefore can select a movie that is similar to movies that the user liked or viewed. The similarity beween two movies can be estimated as follows:

- extract the set of n users that have ranked at least these two movies;
- each movie is then represented a vector of $\mathbb{N}^n$, the i^{th} component of this vector corresponds to the rating given by the i^{th} selected user;
- the similarity is finally estimated as the correlation between these two vectors.

6. Why the correlation can be used as a similarity ? Can you propose another similarity measure ? What are the advantage and drawbacks of the correlation as a similarity ?
7. Implement this similarity estimate and compute the similarity between `Scream (1996)` and `Stargate (1994)`.
8. Give the five most similar movies to `Scream (1996)` and `Stargate (1994)` along with similarity values.
9. What is the complexity of such computation ? How many time does it take to compute the similarities between all movies ?
10. How can we evaluate these results ?
11. Instead of extracting the users that have ranked both movies, we may extract all users that have ranked at least one, and fill the i^{th} component of the vector with a 0 rating if the i^{th} selected user did not rate the movie. What do you think of such approach ? How does it works in practice ?

[1] You may use the function `hist` from the python library MATPLOTLIB.