PCML CS-433: Recommender System

Gael Lederrey, SCIPER 204874, gael.lederrey@epfl.ch
Stefano Savarè, SCIPER 260960, stefano.savare@epfl.ch
Joachim Muth, SCIPER 214757, joachim.muth@epfl.ch

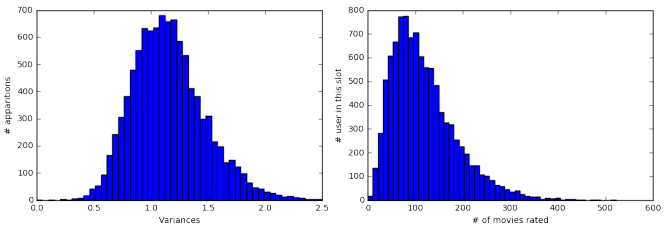
School of Computer and Communication Sciences, EPF Lausanne, Switzerland

***Abstract*—**

I. Data description

The data represent ratings from $10'000$ users on $1'000$ movies in an integer scale from 1 to 5. This scale represent the number of *stars* given by the users, 1 being the lowest grade and 5 the best.

The training set used to train our algorithm contains $1'176'952$ ratings which represent around 12% of possible filled ratings. An other $1'176'952$ ratings are hidden from us and must be predicted by our recommender algorithm.



(a) Distribution of variances of ratings per user.

(b) Number of movies rated per user.

Figure 1: Statistical description of data

II. Data preprocessing

A. Search for spammers

B. Search for inactiv users

C. Normalization of user behaviour

[To do: normalization of user mean and variance]

III. Model selection

A. Models

1) Global mean/median: The most simple model is to take all the ratings in the train set and apply the mean or the median value. We return this value as the prediction.

2) User/Movie mean/median: Another simple model is to compute the mean or median value for the users or the movies. Then we can return the corresponding value as the prediction.

3) Movie mean/median with User mood: The third set of model uses the mean or median value for each movie. We also compute the "mood" of the users this way:

$$d_u = \overline{U} - \overline{u} \quad \forall u \in U \tag{1}$$

where $\overline{U} = \frac{1}{\#U} \sum_{u \in U} \overline{u}$ and $\overline{u}$ being the average rating of the user u.

Then, we return the prediction of a user u on a movie m:

$$p_{m,u} = \overline{m} + d_u \tag{2}$$

where $\overline{m}$ is either the mean or the median of the ratings on the movie m.

4) Matrix Factorization using Stochastic Gradient Descent:

5) Alternativ Least Square:

6) kNN item-based:

7) Pareto Dominance and Collaborative Filtering Nearest Neighbors:

B. Models benchmark

[insert here a benchmark table for each method]

C. Blending

IV. Result

V. Discussion