PCML Project 2 - Recommender System for Netflix Prize Dataset

Manana Lordkipanidze, Aidasadat Mousavifar, Frederike Dümbgen
EPFL, Switzerland

I. Introduction

In the age of online platforms for entertainment, shopping and knowledge transfer, the automatic characterization of the available items and the involved users has become a crucial factor of success. It is used to increase the ease of use (bypass exhaustive searching of the database by suggesting appropriate pages), to maximize the effect of the platform (by placing appropriate recommendations, a user might buy/watch/read more than he originally intended) and to increase profit by intelligent ad placement and user screening for personalized marketing. The tools created for such purposes are commonly called "recommender systems". In some cases, the recommendations can be done using additionally provided information on users and items, such as demographic measures like age and gender or official popularity measures and item classifications. This content-based approach tends to be too simplistic and cumbersome for large databases and adds a dependeny on third-party data of unknown accuracy. Methods grouped under the term "Collaborative filtering" overcome these difficulties by predicting behaviour based on past feedback. Feedback can either be of implicit form (how many times did a user click on an item, how long did he watch a TV series, etc.) or of explicit form, such as rankings or feedback form answers. [1] When users and items are grouped by their similarity and recommendations are done based on similar users (user-oriented models) or similar items (item-oriented models) respectively, one talks about neighborhood models. If, on the other hand, the items and users are characterized by a set of K features whose signification is a priori unknown, one talks bout latent factor models The goal of this project is to apply collaborative filtering to the Netflix dataset, which consists of explicit feedback in terms of ratings of $D = 1000$ users and $N = 10000$ items. The main focus is on latent factor models, but neighborhood models can be taken into account to improve predictions.

II. Models and Methods

A. Preprocessing

1) Data analysis: The data provided is of the form

$$\mathbf{X} = (x_{nd}), \tag{1}$$

where x_{nd} corresponds to the rating of user d for movie n. the ratings are discrete and go from 1 to 5. Since every user only rates a very small subset of movies, the matrix is sparse.

- ratio of non-zero entries
- how many ratings does each user make on average?
- by how many users is each item rated on average?

2) Preprocessing: Since users and movies can vary a lot in terms of their average ratings, i.e. some users might give generally give more positive ratings than others and some movies might be more negative than others, there might be some implicit bias in the data provided. This biais was removed as follows by subtracting a correcting term from each element,

$$\widetilde{x}_{nd} = x_{nd} - \mu_{nd} \tag{2}$$

$$\mu_{nd} = \begin{cases} \mu_n = \frac{1}{|R(n)|} \sum_{d \in R(n)} x_{nd}, & \text{for item bias only} \\ \mu_d = \frac{1}{|R(d)|} \sum_{n \in R(d)} x_{nd}, & \text{for user bias only} \\ \mu = \frac{1}{|R(n,d)|} \sum_{n,d \in R(n,d)} x_{nd}, & \text{for global bias only} \\ \mu_n + \mu_d - \mu & \text{for combined biais} \end{cases} \tag{3}$$

The entries of the residual matrix, $\widetilde{x}_{nd}$ are then factorized as explained in II-B and the bias is added again for the final predictions. TODO: insert the simple matrix visualization for this. As expected, the combined bias with underlying assumption that each rating can be composed of a contribution by the user and one by the item, is the most accurate model. (TODO: insert super pretty plot of errors by Manana here)

B. Machine learning methods

The goal is to factorize the given ratings matrix using two low-rank matrices,

$$\mathbf{X} = \mathbf{W}\mathbf{Z}^T, \qquad \text{with} \quad \mathbf{W} \in \mathbb{R}^{N \times K}, \mathbf{Z} \in \mathbb{R}^{D \times K}, \tag{4}$$

where K is the number of latent features, and $\mathbf{Z}$ and $\mathbf{W}$ are in the following referred to as the user and feature matrix respectively.

Find more about these methods in [2]

1) Stochastic Gradient Descent:

2) Bias Stochastic Gradient Descent:

3) Alternating Least Squares:

C. Evaluation

1) **Cross valdiation** A 10-fold cross validation is implemented for the whole training dataset to fix the hyperparameters.

2) **Performance prediction** The training data was split into two sets of ratio 1:2, emulating the actual data ratio between Kaggle's training and test set. Doing the matrix factorization with this training/test set pair enables us to predict if we can expect a better performance on the kaggle dataset. TODO: what is this ratio for this dataset?

REFERENCES

[1] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets Yifan," *IEEE International Conference on Data Mining*, pp. 263–272, 2008.

[2] C. R. Aberger, Y. Koren, R. Bell, and C. Volinsky, "Recommender : An Analysis of Collaborative Filtering Techniques," *Computer*, vol. 42, no. 8, pp. 42–49, 2009.