Building a recommender system with R

This paper is a deliverable of the capstone project in the 'Professional Certificate in Data Science' course hosted by Harvard University (HarvardX). The main task of a recommender system is to predict the users responce to different options. This document provides an example for predicting how a user would rate unseen movies. GroupLens Research has collected and made available rating data sets from the MovieLens web site. A data set with 10 million rows is split into a train (90%) and a 'unseen' test (10%) set for evaluation. The evaluation metric is RMSE.

Keywords: recommendation, collaborative filtering, R, recosystem, matrix factorization

Acknowledgements and further readings

Populair approaches for designing a recommender system

- *Content-Based systems* analyse the characteristics of items. Similarity of items is determined by measuring the similarity in their properties. Content-based filtering recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or tags given to the item by users. The user profile is represented with the same descriptors and built up by analyzing the content of items which have been seen by the user.
- *Collaborative-Filtering systems* analyse the relationship between users and items. The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person.

This paper

I've chosen collaborative-filtering approache, since the data, as specified within the capstone project, contains a limited set of meta data, does not contain the required user tags nor does it contain details about the users.

The MovieLens Dataset

GroupLens Research has collected and made available rating data sets from the MovieLens web site. A data set with 10 million rows is split into a train (90%) and a 'unseen' test (10%) set for evaluation.

Description

The training dataset contains 9000055 records and 6 columns with:

- 69878 users,
- 10677 movies,
- 797 unieke values in the genres column, muliple genres may apply to one movie,
- 10, ratings, in steps of 0.05 from 0 to 5 so 10 possible outcomes.

The data spreads a timespam from 1995-01-09 11:46:49 until 2009-01-05 05:02:16.

Table 1: Example rows

	userId	movieId	rating	timestamp	title	genres
1	1	122	5	838985046	Boomerang (1992)	Comedy | Romance
2	1	185	5	838983525	Net, The (1995)	Action | Crime | Thriller
4	1	292	5	838983421	Outbreak (1995)	Action | Drama | Sci-Fi | Thriller
5	1	316	5	838983392	Stargate (1994)	Action | Adventure | Sci-Fi
6	1	329	5	838983392	Star Trek: Generations (1994)	Action | Adventure | Drama | Sci-Fi
7	1	355	5	838984474	Flintstones, The (1994)	Children | Comedy | Fantasy

Notes:

- The title contains the release year of the movie.
- The genre column contains multiple genres seperated with a pipe.

Descriptive statistics

In addition to the quantative description of the dataset in the previous paragraph some basic statistics about ratings may be of interest.

Missing values

The dataset has 0 records with missing values.

Table 2: Percentage missing values per column

userId	movieId	rating	timestamp	title	genres
0	0	0	0	0	0

Feature engineering

A Collaborative-Filtering recommendation system uses a matrix of users, movies and ratings as values. Thus, further feature engineering is not required. One might consider the engineering of the following derived features when building a Content-Based recommendation system:

- Date attributes: year, month, day of week, hour.
- Release year (included in the title text).
- Number of genres per movie.
- One-hot encoding of genres.
- Add mean, median and or mode of rating per genre, movie and user.
- Add number of ratings per movie.
- Add number of rating per user.