Machine Learning Engineer Nanodegree

Capstone Proposal

Simon Jackson March 20th, 2017

Proposal

Domain Background

The use of recommender systems is popular on e-commerce sites, which are interested in predicting the ratings/preferences of users for products[1] to personalise and improve the user experience. For example, Netflix wants to suggest movies you will like, Amazon ought to recommend books you're likely to enjoy, and so on.

Recommender systems often take the form of collaborative or content-based filtering. The former involves predicting what a user will like based on their similarity to other users[2]. The latter involves matching content to a user based on similarities among the content[3]. A more advanced approach that will be investigated in this project is a hybrid recommender system, which leverages and combines collaborative and content-based filtering.

Problem Statement

Imagine you work at Netflix and have added new movies to the service. You'd like to recommend these to people who will like them. To determine whether a particular user might like one of these movies, it's impossible to see if other, similar users like the movie (because it's new and hasn't been rated). It's also challenging to see how the user rated other similar movies, because user ratings are relatively sparse. This project will attempt to solve this problem by investigating a hybrid recommender system for predicting the 5-star rating a person will give a new, unrated movie. A collaborative filtering component will link a given user to other similar users, and movie preferences can be pooled over the entire group of similar users. This can be used to estimate the preferences of many already-existing movies, which can be used by a content-filtering component to make the final prediction.

Datasets and Inputs

The data used in this project will come from two open-source projects:

- MovieLens 20M Dataset: Over 20 Million Movie Ratings and Tagging Activities Since 1995
- IMDB 5000 Movie Dataset: 5000+ movie data scraped from IMDB website

The MovieLens dataset is to be used as the key source for the collaborative filtering component of the model. It contains individual user ratings of movies on a 5-star scale (with 5 being the best and 1 being the lowest).

The IMDB dataset is to be used as the key source for the content-based filtering component of the model. It contains public information about 5000 movies and includes the following variables:

> "movie_title" "color" "num_critic_for_reviews" "movie_facebook_likes" "duration" "director_name" "director_facebook_likes" "actor_3_name" "actor_3_facebook_likes" "actor_2_name" "actor_2_facebook_likes" "actor_1_name" "actor_1_facebook_likes" "gross" "genres" "num_voted_users" "cast_total_facebook_likes" "facenumber_in_poster" "plot_keywords" "movie_imdb_link" "num_user_for_reviews" "language" "country" "content_rating" "budget" "title_year" "imdb_score" "aspect_ratio"

Combined, these two data sets can be used to train and test a hybrid recommender model for predicting the ratings that users will give "new" movies.

Solution Statement

The solution should:

1. Take a person who has rated movies in the MovieLens dataset
2. Take a movie that hasn't appeared in the MovieLens dataset, but has features about it available in the IMDB dataset.
3. Return a predicted 5-star rating.

Benchmark Model

Two benchmark models for estimating the solution (5-star ratings):

- The mean rating of all users for all movies.
- The mean rating of the user for whom a prediction is being made.

Evaluation Metrics

Given a data set of known movie ratings, models investigating this problem can be evaluated by the accuracy of their predictions. If 5-stars are treated as categorical outcomes, then metrics such as precision, recall, and F1 are appropriate measures of performance. If 5-stars are treated as continuous, then metrics such as root mean square error are appropriate (this metric was used in the famous Netflix Prize). Different algorithms will be tested, therefore making it possible that both of these evaluation approaches will be used.

Project Design

My expected approach will attempt to do the following:

- Develop *collaborative filtering model*: For a particular user, use unsupervised methods to find similar users and how they would rate the movies being considered.
- Develop *content-based filtering model*: For any new movie, use unsupervised methods to find similar (rated) movies and link these to user ratings.

- Combine information in a supervised manner to make a prediction.

The expected workflow for generating these components will be to iteratively address the following tasks:

- Import data
- Clean data, including:
  - Investigating outliers
  - Handling missing values
  - Normalising where appropriate
  - one-hot encoding where appropriate
- Feature engineering
- Develop unsupervised method(s) for clustering movies
- Develop unsupervised method(s) for clustering users
- Develop supervised method(s) to make movie prediction
- Assess performance of final model(s) using validation and test sets by:
  - Evaluating performance with relevant metrics (as described earlier)
  - Comparing to benchmark model performance
  - Using k-fold cross validation

---

1. Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1-35
2. Prem Melville and Vikas Sindhwani, Recommender Systems, Encyclopedia of Machine Learning, 2010.
3. R. J. Mooney & L. Roy (1999). Content-based book recommendation using learning for text categorization. In Workshop Recom. Sys.: Algo. and Evaluation.