Individual Project Report

Pedro Uria Rodriguez

April 22, 2019

Individual Work

- Wrote the code for converting the `.txt` files into `.csv` files: `txt_to_csv.py`. This code was later improved by Zach.
- Wrote the code to combine all data into one massive `.csv` file: `combine_csv.py`. This was later also implemented by Zach on `txt_to_csv.py` for improved efficiency.
- After some discussion with the team, wrote the code to downsample the dataset due to size issues: `downsample.py`. This code consists in removing users and movies with few reviews, and then on subsetting the dataset by choosing a 0.5% random sample of users. A random seed was used for reproducibility.
- Came up with a "clustering" approach and implemented it after discussion with the team: `get_similar_users.py`. This scripts computes the rating distances from each user to each other and then selects the most similar users in terms of ratings, for each user. This code takes about one and a half hours to run and saves both the distances and the clusters in `.json` format. See the distance metric below:

$$d(A,B) = \frac{1}{n \cdot 5^2} \sum_i (r_{A_i} - r_{B_i})^2$$

  where r_{Ai} is the rating of user A to movie i, and n is the number of movies both users A and B have rated. The clustering was done by selecting a threshold of $d(A,B) \leq 0.04$, which means that on average, the ratings of A and B_j for each common movie are within a unit distance. For example, $r_{A_i} = 5$ and $r_{B_{j_i}} = 4$. This threshold can be experimented with later on.
- Came up with the modeling approach by making use of the clusters of users. Wrote an example of the dataset to be used for training a model for a particular user: `user_to_model_example_729846.py`.

movie	cluster_avg_ratingt	movie_feat_1	movie_feat_2	...	user_A_rating
movie_1	Known	Known	Known	...	Maybe Known
movie_2	Known	Known	Known	...	Maybe Known
...	....	...	...	...	...

The table above shows an idea of the process. Using the average rating of the groups of similar users for a movie, together with features taken from IMDb, we would train a Data Mining model to predict the rating for user A. Some rating will be known, and others will not, so we will drop the unknown ones and split the rest into training and testing. Due to a lack of time, we decided to not include the IMDb features, and will leave this as a potential future improvement.

- Wrote the code for the different models and clustering thresholds, located in `models.ipynb`.
- Wrote a brief comparison between the approach on `models.ipynb` and the collaborative filtering approach (`comparison.ipynb`).
- Described all of the above on `README.md` under the main code subdirectory.
- Helped to write the project proposal, project report and project slides. This was mostly a team effort, with each of us revising and improving each others work.

Percentage of non-self written code

- `downsample.py`: 4.77%. (`https://stackoverflow.com/questions/613183/how-do-i-sort-a-dictionary-by-value`)
- `get_similar_users.py`: 4.55% (`json.dump`)
- `user_to_model_example_729846.py`: 0%
- `models.ipynb`: 3 lines out of many lines (`https://stackoverflow.com/questions/613183/how-do-i-sort-a-dictionary-by-value`)
- `comparison.ipynb`: 0%, although used some of Zach's and Aaron's code.