CS 484 DATA MINING
JACOB BOISSEAU & ANDREA PEREZ ISLA
MINER USER: JCUP1 & YELLOWSUB12
RANK ON MINER: 20th 0.82
11/06/2018

HW4: Recommender Systems

1. INTRODUCTION

Data mining is used to develop new recommender systems nowadays in order to predict the preference that a user would have for a determined product. Streaming media platforms like Netflix or HBO use it to suggest their users to watch some type of movies or series based on previous visualizations. There are different types of recommender systems: collaborative, content-based, demographic-based, utility-based, knowledge-based and hybrid. In this case, considering the data we have available, we are going to develop a content-based recommender system. It learns a profile of the user's interests based on the features present, and the rates that this user gave to an item. For this assignment, we are required to predict the 5-star rating a movie will get for a user. We have different files with information about the movies seen by a user (genres, actors, directors and tags assigned by the platform and the user itself), in addition to the training dataset (with the user ID, movie ID and the rating) and the test data (without the rating).

2. METHODOLOGY

As I mentioned before, we are going to implement a content-based recommendation system to predict the rating a user is going to give to a determined movie based on previous products watched.

For this purpose, we have used the Surprise library in Python, as well as numpy and Pandas to provides easy-to-use data structures and analysis. We decided to implement our own K-

Nearest Neighbors class, using the surprise library for preprocessing and finding the similarity between user vectors.

KNN is really effective when the training set is large, which is the case for this assignment. The algorithm converts each user profile into a user vector, this user vector is then compared to all other user vectors to find similar user profiles. When predicting an unknown movie for a user, the algorithm checks other users with similar profiles and if these users have seen the movie in question. Based on the user rating and their similarity to the test user a weighted total is calculated for the final rating of the unknown movie. We are using the surprise library’s MSD similarity matrix as well as a K selection of 21, this gave us the final accuracy of 0.82. We decided on the surprise library after mocking this algorithm ourselves using skleran’s cosine similarity and found that our implementation took an enormous amount of time to run. The surprise library allowed us to cut our runtime down significantly.