Preliminary Report
CS-373

Liam Connery, Sam Javed, Theo Burkhart, Todd Griffin
16 November 2018

Project Goal

For our project, we will be using a dataset containing user's movie reviews. Each entry contains the user's unique ID, the movie's unique ID, and the rating the user gave the movie (1-5). Our goal is to be able to predict whether a given user will like a movie based off their reviews of other movies. We will user perceptron as well as perceptron with myopic forward fitting as our machine learning algorithms. We plan on choosing the movie with the most reviews in order to give us the largest subset of data.

Dataset: http://files.grouplens.org/datasets/movielens/ml-latest-small.zip

Cleaning the data

The main file we are analyzing is 'ratings.csv' which contains 3 columns, the user ID, movie ID, and rating. Our first step was to create new, unique move IDs since the given movie IDs did not follow numerical order. We wanted the movies to be numbered 1 through n, where n is the total number of movies. Our python script 'ratings_mod.py' did this by creating a dictionary where the key is the old movie ID and the value is the new movie ID, which we generate by incrementing a count by one for every new movie ID we encounter. We then just ran through all the movie IDs and replaced them with the value in our dictionary. From this script, we had a new dataset, 'updated_movie_rating.csv', with 3 columns (user ID, new movie ID, and rating). In the same script, we created a new CSV file ('new_movie_id_lookup.csv') that will allow us to lookup the old movie ID with our newly created ID. We needed this to be able to look up the movie titles from the provided, 'movies.csv'.

Lastly, we created a python script ('Transform.py') that transformed the data so each row represents a user, and each column a unique movie. We did this by creating a new matrix where the number of rows is the number of users, and the number of columns is the number of movies. We then filled in each column for each user with their rating of that movie. For example, the field (1, 3) would contain the rating user 1 gave movie 3. If a user has not reviewed a movie, we put a 0 in that field. In the end, our final dataset is 'final_test_data.csv' which we will use to train our perceptrons.

Based on our preliminary results, we don't expect to make anymore major changes to our dataset. However, we do plan on testing how our results are affected by taking a subset of our data. We want to see how removing movies with less than a certain number of reviews, or only using movies with the most reviews, affects our final results. This may yield better results given our data set contains only 610 users and over 9000 movies which will represent features to our perceptron.

Perceptron

We are implementing the perceptron learning algorithm. The two inputs are a n by d matrix X and a n by 1 matrix y. N is equal to the number of different individuals that have provided a review in our data set. D is equal to the number of different movie in the data set minus the specific movie perceptron will be training for. The matrix y is generated by removing the column of the movie we aim to train on from the original X. These two inputs give us one snapshot of a user's review history (X) and an indicator if they liked or dislike that movie (y).

Only one modification was made to the original Perceptron algorithm. The modification is made to prevent training on samples that have a y value of 0. This would mean that individual has not seen the movie we are training to predict. If we did not remove these samples they would cause theta to be wrongly skewed.

Early Testing

We were able to run our data using the perceptron algorithm that does not take advantage of myopic forward fitting. The algorithm worked on our data set as planned and provided good results. Our initial tests follow the steps taken in the case studies in lecture with first training and testing on the entire data set to see the upper bound on performance. The initial testing with this method had a mean error of 0.0. This was expected due not only to training and testing on the same data set, but also due to the number of features being much larger than the number of training samples. In our dataset, we have 610 users reviewing over 9000 movies. The model is likely too complex to train with this few of samples and demonstrates the value that myopic forward fitting with bring to our model.

Next Steps

Moving forward, our next step is to implement myopic forward fitting to find which movies act as best predictors for what rating a user will give a new movie. When this is complete, we plan to implement k-fold cross validation to determine the effectiveness of our two models. Finally, reporting results using scores from the k-fold cross validation and graphs to visualize our models.