WhatsOn: Recommendation System for Movies On Redbox And In Theatres

Author: Jamie Recas | Advisor: John V.E Ridgway | Trinity College Computer Science Department

Abstract

WhatsOn is a web application, built with Ruby on Rails, that suggests movies on redbox (a movie rental kiosk) and in theaters to it's users based on their calculated preferences. It accomplishes this using user-user (K-NN) collaborative filtering. K-NN Collaborative filtering was implemented using the K-Means algorithm and a weighted average. Using this method WhatsOn manages to predict, with reasonable accuracy, the estimated ratings for many of its users.

Motivation

There are a lot of movies to choose from. These movies may be popular or unpopular. Most of us have probably heard of the popular movies but what about the unpopular movies? The aim of this project is to suggest both popular and unpopular movies that are currently on redbox or in theatres.

Technology Overview

Front End

- HTML
- CSS
- jQuery

Back End

- Ruby on Rails

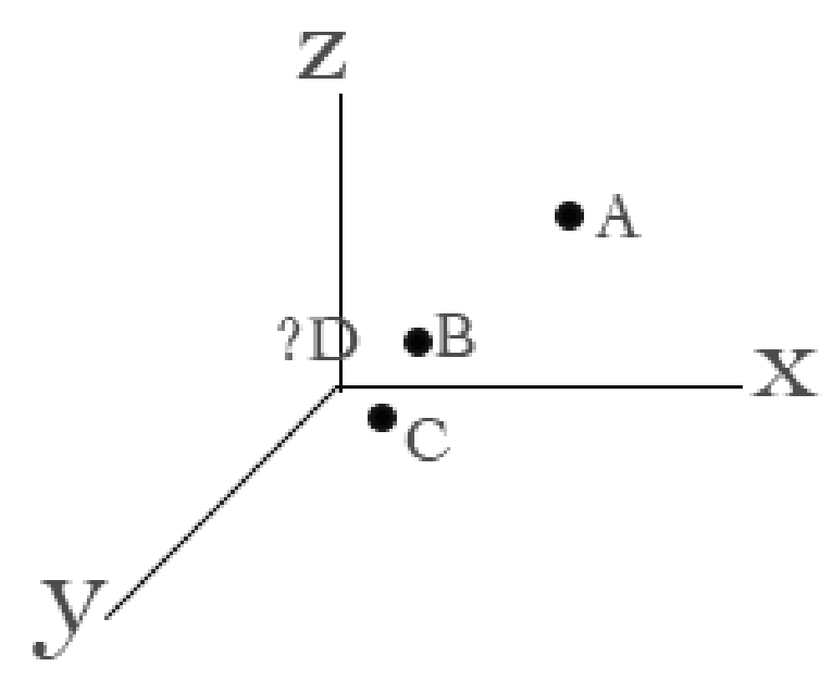
Ruby Gems: Nokogiri, GoogleClientAPI, Whenever

Collaborative Filtering

The type of collaborative filtering used was user-user collaborative filtering also known as K-NN collaborative filtering. K-NN collaborative filtering is a straightforward algorithmic interpretation of the core premise of collaborative filtering, which is the grouping of similar users and using their average ratings for a particular item as an indicator of the expected rating for other similar users. Collaborative Filtering is accomplished by first creating a recommendation matrix. For example:

	Movie 1	Movie 2	Movie 3
User A	4	5	5
User B	2	1	2
User C	1	2	1
User D	?	2	2

This matrix consists of four users and their ratings for three movies. Each row in the matrix is treated as a positional vector. This means that each user has a position in space. For example:

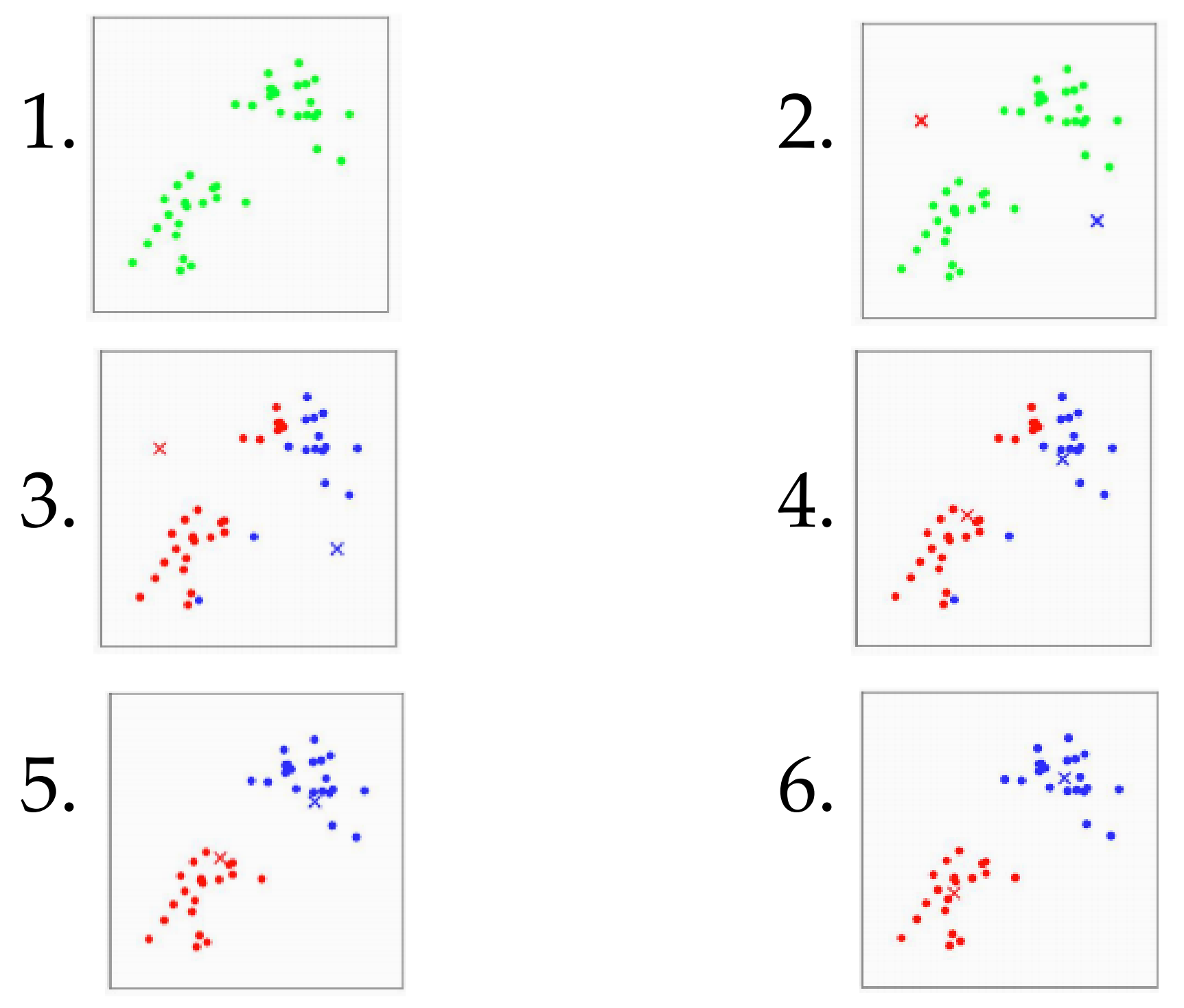


Users whose positions are close to each other are said be similar because they rate movies similarly. Hence, we use group B and C together and use a weighted average to estimate D's rating for movie 1.

K-Means

The K-Means algorithm was used to group similar users based on their position vector. The K-Means algorithm works as follows:

Step 1: Each node (user) is given a position.
Step 2: Cluster centroids are randomly placed within the range of the nodes positions.
Step 3: Each centroid is assigned the nodes that are closest (Euclidean Distance) to it.
Step 4: Centroids are then repositioned to the average position of their nodes
Step 5: Repeat steps 3 and 4 unless at equilibrium.



Weighted Average

$$p_{u,i} = \bar{r}_u + \frac{\sum_{u'\in N} s(u,u')(r_{u',i} - \bar{r}_{u'})}{\sum_{u'\in N} |s(u,u')|}$$

Where $P_{u,\ i}$ is the predicted rating of movie *i*, for user, *u*. r_u is the average rating for user u and S (*u*,*u′*) is the similarity(proximity in space) between user, *u*, and another user, *u′* , in the cluster. *(ru′,i -ru′,i)*, is the normalized rating for user.

Recommendations

After clustering, there are now groups of similar users but this does not give us any recommendations. To get recommendation for a user *j* and movie *k* we must go into the cluster that j belongs and find all the users that have rated the movie *k*. The weighted average equation is then used to give an estimate of user *j's* rating

Performance

WhatsOn manages to predict, with reasonable accuracy, the expected ratings for users of the application. It does, however, suffer from the "cold starting" problem that plagues many collaborative filtering applications and the K-Means algorithm needs to be optimized. The cold starting problem is as a result of a scanty recommender matrix (many users who have not rated items).

Future Work

- Optimization of the K-Means algorithm.
- Add a feature learning component.

References

Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan. "Collaborative Filtering Recommender Systems." *Foundations and Trends R in Human–Computer Interaction* 4.2 (2011): 81-173. Web..

Ozbal, G., H. Karaman, and F. N. Alpaslan. "A Content-Boosted Collaborative Filtering Approach for Movie Recommendation Based on Local and Global Similarity and Missing Data Prediction." *The Computer Journal* 54.9 (2011): 1535-546. Print.