EECS 349 Fall 2015
Jason Brown - Homework 4

1) Collaborative Filtering: Looking at the Data

   a) The MovieLens 100K database was mapped into a numpy two dimensional array. This was a large dataset, and it dealt with a bug in python that made each list.append() a linear time operation (http://bugs.python.org/issue8023). I got around this by disabling garbage collection and pre-allocating the memory as a numpy array.
   Of the 943 users, in terms of pairs, there was a mean review in common of 44 and a median of 29. A histogram is attached.

   b) Movie 50 ("Star Wars") had the maximum number of reviews, 583.
   Movie 599 ("Police Story 4: Project S") had the minimum number of reviews, 1. A line plot is attached.

   While the number of reviews fits the spirit of Zipf's law, it doesn't fit it exactly. Here are the highest reviewed movie counts: [583, 509, 508, 507, 485, 481, 478, 452, 431, 429, 420, 413]. If it fit Zipf's law exactly, the second most reviewed movie would have 292 reviews, not 509. The first most frequently reviewed movie would have 194 reviews, not 508.

2) Collaborative Filtering: Distance measures

   a) Approach A is to compute the manhattan distance between two vectors, and if an attribute of one of the two vectors is blank, to put a zero there. Approach B is to compute the manhattan distance between two vectors, and if an attribute of one of the two vectors is blank, to put the average of the user's reviews there. In Approach A, the conclusion reached on distance between two items is based heavily on missing data. Approach B appears to be a better path. While putting in the average value for missing attributes distorts the data, it distorts the data far less than putting in zero. Here is an example: Bill likes horror movies and dislikes romantic comedies, and rated 4 movies [4,5,2,2]. Byron likes romantic comedies and rated 4 movies [2,1,4,4]. Blake likes horror movies, but hasn't seen many. His movie vector is [4 ? 2 3], where '?' denotes missing data. With Approach A, his manhattan distance from Bill and Byron is both 6, yet of the movies he's seen, his preferences are closely aligned with Bill's! When filling in a average value for missing data, his manhattan distance from Bill is 3 and from Byron is 7. Approach B yields a manhattan distance that is more reflective of Bill's preferences.

   b) Pearson's formula measures correlation between two vectors and returns a range of (-1, 1), with perfect positive correlation equal to 1, and a perfect negative correlation equal to -1. Euclidean distance measures distance in space between two vectors and is simpler to compute. The con is that two movies may have euclidean distance can be skewed by differences in scales. However, Euclidean distance will work fine for item-based collaborative filtering because for each attribute of the movie, the same user provided all ratings. Titanic had reviews [5,4,1]. Caddie Shack had reviews [1,3,2]. Braveheart had reviews [4,5,2]. Braveheart will show a strong postive Pearson correlation with

Caddie Shack, much stronger than with Titanic, yet it is much more similar to Titanic in terms of reviews. It's euclidean distance from Titanic is $\sqrt{3}$ and from Caddie Shack is $\sqrt{13}$, which correctly shows it as more similar to Titanic in terms of reviews.

3) -

4) -

5) Collaborative Filters

   a) Python scripts developed as described.