Project 9
Joe Zeimen

The goal for this project is to take data on users ratings of music tracks, do some preprocessing and mining of this data. Then, predict what users would say is their rating of some reserved test data. This test data has 6 tracks rated by users so we can see how well the predictions match and find the SSE.

Approach

To simplify the project I only looked at track data, users, genres, and ratings. I ignored artist and album data. This could be easily added later, but to get the process down I just used track and related data. I also used the sample data set instead of the entire dataset after finding out how prohibitively large it is.

I took the data and put it in a relational database. I then decided that since most songs had only been rated once, just looking at the nearest neighbors for their ratings of the songs would not work. I then decided that, since most tracks had genre data, I could then estimate how much a user likes a specific genre. So, I took the average rating of tracks for each specific genre a user rated. So, if they rated 2 different songs that were in genre 445 at a 90 and 80, the genre rating became 85. I then combined each user's genre rating with the 5 nearest neighbors. Then for each song that needed to be estimated I took the average of the ratings for the genre of the song. If there was no data I simply predict a 50.

Data/Preprocessing

For this project I pulled the data out of the text files and put it in an sqlite3 database. I used Active Record in Ruby to accomplish this. This also gave me my first taste of how much of a burden big data is. For just the sample data set it took tens of minutes to load the data into a database. This was only 43 users and about 5000 ratings. When trying to load the entire full dataset (250 million ratings) after many hours I my computers could not make a dent in creating a database.

K-Nearest Neighbors

Next I try to find the K-Nearest Neighbors for each user. I do this by finding the cosine similarity between the users and sorting them. I found predictions for tracks based on 1 through 5 different users.

Predictions

To predict an items rating I looked at the genres an item has. Then looked at the ratings for those genres based on the k-nearest neighbors. I averaged the all of the ratings for the related genres together to produce a prediction. I also produced predicitions based off of always guessing at 0 and 50 for the rating.

Results

Parameters	RMSE
K=0 (No neighbors)	33.23133880075095
K=1	34.42957327725322
K=2	34.72420404523897
K=3	35.495991225731515
K=4	35.980202311206995
K=5	36.771751522010064
Always predicts 50	36.46916505762094
Always predicts 0	72.73238618387272

It turns out that this method is not that much better than just guessing a 50 for each rating. Also taking the k-nearest neighbors into consideration did not help at all. This could be due to the fact that I only looked at track data instead of albums and artists. It also could be due to the fact that this is a very small data set for how many possible tracks and genres we have available. I think that these results would drastically improve with using all of the data, but the way I have it set up to load data and calculate it would take an impossibly long time.