Music Rating Prediction: Milestone

Brian Doolittle and Pratap Luitel

COSC 174

Problem:

The goal of the project is to predict how a user will rate a song given their demographic, previous ratings, music preferences, and description of the artist. This project was originally posted on kaggle.com as a 24 hour hackathon taking place on July 21st, 2012.

Data:

The data was collected by EMI, a music recording and publishing company, from an online survey spanning two years. There are 3 data sets on Kaggle relating to the project, users.csv, words.csv and train.csv. Users.csv contains user demographics such as gender, age, importance of music in user's life, hours spent each day actively listening to music, hours spent each day listening to music in the background, etc. Words.csv contains information such as how much a user likes an artist, if the user owns the artist's music and a series of words used by the user to describe a track. The training data contains 50,928 user's rating of 184 tracks from 50 artists.

Preprocessing:

Two of our data sets, words.csv and users.csv, contained both numeric and string valued data. To incorporate these data sets into our model, we needed to convert the non-numerical fields into numerical values. Certain fields like gender were converted to a binary value. While other fields like region were classified such that each class (1,2,3,4) represented a region (North, Midland, Center, South). Fields like 'artist rating' had numerical data in the range of 0-100. A tabled description of data manipulation can be found in the appendix.

Furthermore, since our data was collected from a series of surveys, there were a number blank or missing fields. During preprocessing we filled these fields with a value of -1 in order to indicate that the data is missing. We plan to design our model to overlook these values and make a prediction using the data that is present.

Algorithms:

Baseline Model:

For a baseline model our algorithm predicted the user's rating by taking the average rating for the track in question. To test the performance of this model we set up a cross validation with 10 folds. For each iteration through the cross validation, we used a single fold as our validation set and the remaining 9 folds as our training set. For each example in our validation set, we calculated the average rating for the example track using our training data. In the case that there are no instances of the example track in our training set, our algorithm outputs the value of 50. Using this method we found a root mean square (RMS) error of 21.25.

Weighted Average Model:

For our next model we predicted the user rating with a weighted average. The features, *B*, include the average rating for the example artist, track, user, and time. For each user our algorithm learned a set of weights, θ. These weights are optimized with the formula for θ listed below.

$$\vec{B} = [Artist_{avg}, Track_{avg}, User_{avg}, Time_{avg}]$$

$$\theta_i = \frac{y_i}{B_i}$$

$$\vec{Y}_{pred} = \vec{\theta}\vec{B}$$

The sum of the weights, θ, is 1. This algorithm allows us to learn an individual set of weights for each user and use these weights to predict their rating for a test example.

We tested the performance of this algorithm in a cross validation with 10 folds. Given a user in a test set, we found the average ratings for the artist, track, user, and time in the training set. For each of the examples in our test set we learned a set of weights to apply to our features to predict the rating. This method gave us an RMS error of 18.69

Pitfalls:

Our algorithm has some pitfalls, first we are using a limited number of features. We currently only look at averages. Some additional features we can add are mode, median, maximum, and minimum. Furthermore, we can include features from users.csv and words.csv. These data sets include useful information about the users which will help us find similarities between users.

A second pitfall in our model is that the algorithm currently uses the rating of the test example to predict the weights. In our final model, we will need to predict these weights just using our test data. This will require us to write an algorithm that finds which users will rate most similarly. To achieve this goal, we will implement a kNN classifier.

Data Analysis:

We initially thought that the distribution of ratings would be best modeled by a gaussian distribution. We tested the log-likelihood of the data for a binomial, gaussian, and truncated gaussian. The maximum log-likelihood was found for the truncated gaussian. This result seems logical because the ratings can only assume values from 0 to 100. While the truncated-gaussian had the greatest likelihood out of the distributions we tried, it does not fit the actual distribution of ratings shown in fig. 1.

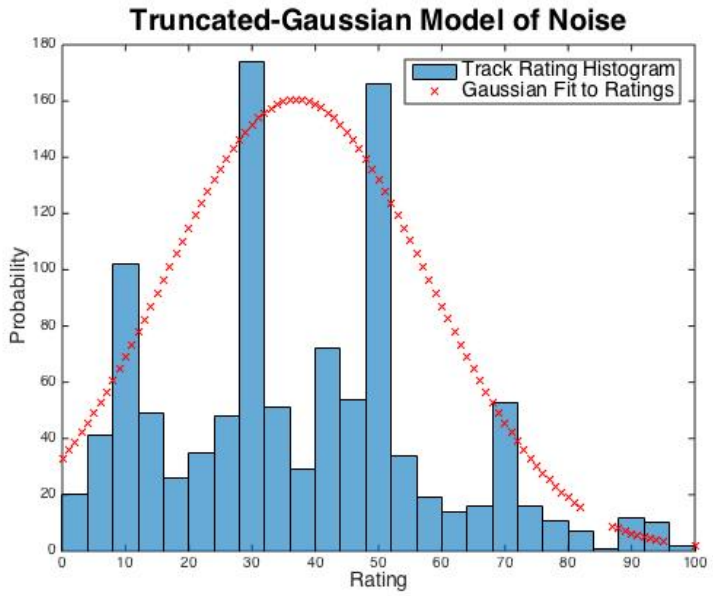
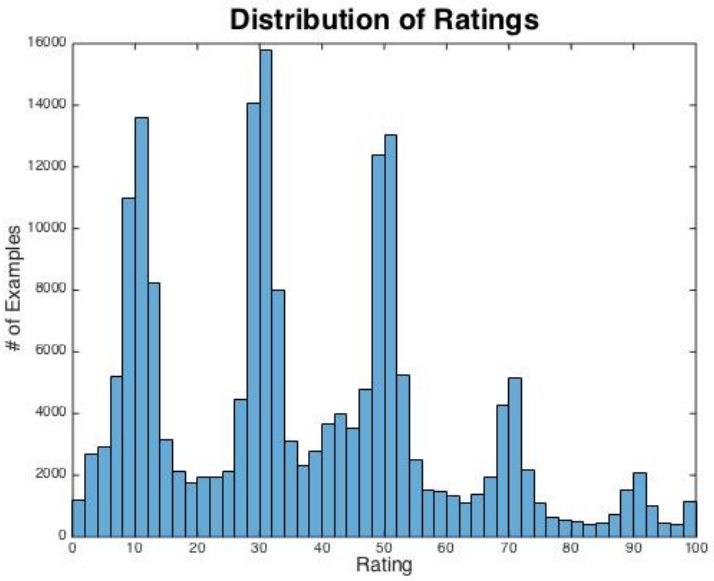


Fig. 1: Distribution of ratings for a single track. A gaussian curve was optimized and scaled to fit this data

The plot shows the distribution of ratings for a given track. We have scaled the optimized truncated-gaussian for comparison. The distribution shows peaks occurring at ratings of 10, 30, 50, 70, and 90. These peaks show that users tend to rate tracks to these values. This trend is also seen in the rating distribution for the entire data set fig 2.



.

Fig. 2: Total distribution of user ratings.

From these distributions, we can infer that the average may not be the most effective feature. For example, in fig. 1, the average rating of the track is given by the peak of the gaussian. However, this peak corresponds to a minimum in the actual distribution. There is a low probability that our average rating will be a good prediction. Using a feature such as the mode may be more effective in predicting the rating.

Furthermore, because these peaks are consistent throughout our data, it may be a good idea to use a classification algorithm to predict user ratings rather than a regression. The classification algorithm will predict a rating of 10, 30, 50, 70, or 90 based on the input features.

Conclusion:

So far, we have only used 4 features in our weighted average model. The relatively high RMS error of our baseline model suggests that we need a better model that incorporates new features. Incorporating features like, median, mode, user descriptions of artists and importance of music in user's life may help improve the model.

Since we have different types of data (binary, classified, numerical), we will need a model that utilizes the difference in the data types. We observed that user ratings tended to be clustered around values of 10,30,50,70 and 90. Our current model predicts a user rating on a continuous range of 0-100. We could instead approach the problem to classify user's rating into one of the commonly occurring values of 10,30,50,70 or 90.

Finally, we could predict ratings of a user by analyzing ratings from other users that have similar music profile. We will implement a kNN classifier algorithm for this purpose as discussed earlier.

Overview of Milestone Goals:

The goals for the milestone were to:

- Preprocess data
- Research Algorithms
- Implement a Baseline Model

We successfully preprocessed the data and implemented a baseline model. We did not perform too much research on algorithms that we can apply to our problem. As we move forward, we will begin researching recommender systems and how they can be applied to our problem.

Appendix A:

Table 1: Data After Preprocessing

Data field	Description	Type
Artist_id	unique identifier for artist	50 Classes
Track_id	unique identifier for track	184 Classes
User_id	unique identifier for user	50928 Classes
Track Rating	User's rating of track	Integer [0,100]
Time_id	month data was taken	24 Classes
Gender	Male/Female	Binary
Age	Age of user	Integer
Working	Employment status of user	9 Classes
Region	Where the user is from	4 Classes
List_Own	Hours spent listening to owned music	12 Classes
List_Back	Hours spent listening to music in background	12 Classes
Q_xx	19 questions about a user's music preferences	Integer [0,100]
Heard_of	Has the user heard of the artist	5 Classes
Own-Artist_Music	Does the user own any of the artists music	5 Classes
Like_Artist	User's rating of artist	Integer [0,100]
Words_xx	82 words used to describe the artist	Binary

Table 2: Classification of Non-Numerical Values

Data Field	File	Original Data	Processed Data
Heard_of?	Words.csv	● Heard of ● Never heard of ● Heard of and listened to music recently ● Heard of and listened to music RECENTLY	● 1 ● 2 ● 3 ● 4 ● 5
Own_Artist_Music?	Words.csv	● DonÍt know ● Own none of their music ● Own a little of their music ● Own a lot of their music ● Own all or most of their music	● 1 ● 2 ● 3 ● 4 ● 5
Gender	users.csv	● Male ● Female	● 1 ● 0
Working	users.csv	● Employed 30+ hours a week ● Employed 8-29 hours per week ● Full-time housewife /househusband ● In unpaid employment ● Other	● 1 ● 2 ● 3 ● 4 ● 5
Importance of Music?	users.csv	● Music means a lot to me and is a passion of mine ● Music is important to me but not necessarily more important ● I like music but it does not feature heavily in my life ● Music has no particular interest for me	● 1 ● 2 ● 3 ● 4