Practical 3: Preferences in Streaming Music

Writeup due 11:59pm on Friday 7 April 2017
Kaggle submission closes at 11:59pm on Thursday 6 April 2017

You will do this assignment in groups of three. You can seek partners via Piazza. Course staff can also help you find partners. Submit one PDF writeup per team via the Canvas site.

For this practical, you are tasked with predicting people's tastes in music. Specifically, you will be predicting the number of times (a non-negative integer) different users listened to tracks of different artists over a span of time. You will have some basic self-reported demographic information about many, but not all, of the users, such as sex, age, and location. You will also have the name of the artist and their MusicBrainz[1] ID, if available. There are about 233,000 users and 2,000 artists. The training set has over 4.1M user/artist pairs and the test set is of a similar size. Your objective is to predict how many times a user will listen to a new artist.

Data Files

There are five files of interest, which can be downloaded from the Kaggle page:

- `profiles.csv` – This file contains information about the users. There is a header row and then four columns in basic CSV format with comma delimiters and double-quote escaping where appropriate. The first few rows are:

  ```
  user,sex,age,country
  fa40b43298ba3f8aa52e8e8863faf2e2171e0b5d,f,25,Sweden
  5909125332c108365a26ccf0ee62636eee08215c,m,29,Iceland
  d1867cbda35e0d48e9a8390d9f5e079c9d99ea96,m,30,United States
  63268cce0d68127729890c1691f62d5be5abd87c,m,21,Germany
  02871cd952d607ba69b64e2e107773012c708113,m,24,Netherlands
  0938eb3d1b449b480c4e2431c457f6ead7063a34,m,22,United States
  e4c6b36e65db3d48474dd538fe74d2dbb5a2e79e,f,,United States
  b97479f9a563a5c43b423a976f51fd509e1ec5ba,f,,Poland
  3bb020df0ff376dfdded4d5e63e2d35a50b3c535,m,,United States
  f3fb86c0f024f640cae3fb479f3a27e0dd499891,,16,Ukraine
  ...
  ```

  The `user` column is a unique alphanumeric identifier. The other columns may be blank if those data were not provided.

- `artists.csv` – This file contains information about the 2,000 artists that have been listened to in these data. There is a header row and then five columns. The first several rows are:

[1] https://musicbrainz.org/

```
artist,name
03098741-08b3-4dd7-b3f6-1b0bfa2c879c,Liars
7a2e6b55-f149-4e74-be6a-30a1b1a387bb,The Desert Sessions
7002bf88-1269-4965-a772-4ba1e7a91eaa,Glenn Gould
dbf7c761-e332-467b-b4d9-aafe06bbcf8f,G. Love & Special Sauce
a3cb23fc-acd3-4ce0-8f36-1e5aa6a18432,U2
8b0f05ce-354e-4121-9e0b-8b4732ea844f,Juanes
8363f94f-fd86-41b8-a56b-26eacb34f499,Summoning
2e41ae9c-afd2-4f20-8f1e-17281ce9b472,Gwen Stefani
c17f08f4-2542-46fb-97f3-3202d60c225a,Fear Factory
4bd95eea-b9f6-4d70-a36c-cfea77431553,Alice in Chains
f467181e-d5e0-4285-b47e-e853dcc89ee7,Ratatat
...
```

The first column is the MusicBrainz ID. The second is the name, if available.

- `train.csv` – This file contains the training data, which are 4.1M artist/user pairs with numbers of plays. It is a standard CSV file with comma delimiters and a header row. The first column is the user identifier, followed by the artist identifier, and then the number of plays. Example rows:

```
user,artist,plays
eb1c ... af03,5a8e ... 9c94,554
44ce ... bb5d,a3a9 ... 84df,81
da9c ... 08e3,eeb1 ... 8e43,708
8fa4 ... 7d81,a141 ... eabc,265
b85f ... b2cf,a3cb ... 8432,220
feed ... 08f7,1cc5 ... f506,2113
cbb8 ... 324b,9c9f ... d090,127
5641 ... 3e9b,832a ... 1c24,305
...
```

- `test.csv` – This file is a CSV file which contains users and artists, but without the `plays` column. Your objective is to predict these values and create a prediction file, which is described below. Each user/artist pair has a distinct `Id`; you'll need to match this to your predictions. There are 4,154,805 pairs to predict. About half of these are used to compute the visible leaderboard. The other half are used to compute the true results. This separation is to prevent overfitting to the leaderboard, and is standard for these kinds of prediction contests. The first couple of rows of the test file are:

```
Id,user,artist
1,306e ... 22d2,4ac4e32b-bd18-402e-adad-ae00e72f8d85
```

```
2,9450 ... 27ac,1f574ab1-a46d-4586-9331-f0ded23e0411
3,8019 ... 53cc,3eb72791-6322-466b-87d3-24d74901eb2d
4,e3ed ... 5d82,61604b45-8a91-4e33-a1b6-45d7b1fec4e5
5,a73f ... 44aa,5dfdca28-9ddc-4853-933c-8bc97d87beec
6,55f1 ... c0af,ef58d4c9-0d40-42ba-bfab-9186c1483edd
7,7ad7 ... bbee,a3cb23fc-acd3-4ce0-8f36-1e5aa6a18432
...
```

- **global_median.csv** – This is an example of how you submit predictions. It is a standard comma-delimited CSV file with two columns. The `Id` column corresponds to entries in the **test.csv** file above, i.e., specific user/artist pairs. The `plays` column is where you specify your best guess. Although the true number of plays are integers, you can produce floating point numbers in your predictions. An example is below:

```
Id,plays
1,118.0
2,118.0
3,118.0
4,118.0
5,118.0
6,118.0
7,118.0
8,118.0
9,118.0
10,118.0
11,118.0
12,118.0
13,118.0
14,118.0
...
```

Evaluation

After you upload your predictions to Kaggle (which you can do at most four times per day), they will be compared to the held-out true number of plays. The score is computed via mean absolute error (lower is better). If there are N test data, where your prediction is $\hat{y}_n$ and the truth is y_n, then the MAE is

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^{N} |\hat{y}_n - y_n|$$

Sample Code

Two Python files are available from the course website. The files `global_median.py` and `user_median.py` implement rudimentary predictions based on simple ideas.

Helpful Hints

As in the previous practicals, you have a lot of flexibility in what you might do. You could focus on feature extraction (the MusicBrainz API may be helpful); feature engineering, i.e., coming up with fancy inputs for your method; model building; or some combination. It may be useful to consider an ensemble of methods. You could also really drive the practical with unsupervised learning and use clustering techniques or matrix factorization. Here are some ideas to get you started:

- How may you be able to cluster the artists and users (given feature extraction from MusicBrainz API) to give you more information beyond that of per-user median counts? Would you be able to group similar users together and leverage that information to give you a better result?
- Perhaps the median of a user's play counts is not the best predictor of how many times someone will listen to a new artist. Can you think of other ways to use the distribution of known play counts for a "best estimate" of future play counts?
- Something we might want to take into account is the fact that even the shape of the distribution of play counts might look very different for each individual and might sometimes be multimodal. (One non-parametric approach to estimating the probability density of a random variable, for instance, is kernel density estimation.)

Another point of consideration is that, like the first practical, this practical's dataset is rather large. There are a multitude of ways to deal with this, including starting with a small sample of the data in order to iterate quickly. Be careful with memory allocation, e.g. by reading in data line-by-line, not constructing and reconstructing the entire matrix, etc.