Experiment Design, Group 04 – Reproducibility - Paper 2

Helmuth Breitenfellner
e8725866@student.tuwien.ac.at

László Király
e9227679@student.tuwien.ac.at

Gerald Weber
e0125536@student.tuwien.ac.at

Abstract

This documents the activities and results performed by Group 04 during *Exercise 2: Reproducibility* of the lecture *188.992 Experiment Design for Data Science.*

A) Paper selection

We selected *Option 2 - Prediction of music genre across different taxonomies* which is based on the paper *MediaEval 2018 AcousticBrainz Genre Task: A Baseline Combining Deep Feature Embeddings Across Datasets* written by Sergio Oramas, Dmitry Bogdanov and Alastair Porter for the MediaEval conference 2018 in France. We have choosen the paper because we were curious to reproduce the output of the Neural Network and see what the problems in difference to regular Machine Learning papers are.

(1) Paper Introduction

The paper describes a baseline approach for the MediaEval 2018 AcousticBrainz Genre Task based on a deep neural network. *The task is focused on content-based musicgenre recognition using genre annotations from multiple sources and large-scale music features data available in the AcousticBrainz database* [BPUS17].

(2) Datasets

The conference provides a website for the classification task[1] which describes the task, the schedule and the dataset on a subpage[2] in detail. This subpage contains multiple links:

- Zenodo[3] uploaded by Dmitry Bogdanov, Alastair Porter et al.
- Google Drive[4] containing train/test/validation folders provided by Dmitry Bogdanov and Alastair Porter

The Zenodo page contains all the required data (allmusic, discogs, lastfm and tagtraum) for the classification task. The second Zenodo page contains restricted access to the AllMusic ground truth, which has been granted after a simple request on the Zenodo page. The compressed file size of all downloads is about 47GB.
After decompressing the dataset is structured as followed:

- acousticbrainz-mediaeval-train folder 83GB, 1.458.447 files
- acousticbrainz-mediaeval-validation folder 18GB, 313.860 files
- acousticbrainz-mediaeval-allmusic-train.tsv 260MB (restricted dataset)
- acousticbrainz-mediaeval-allmusic-validation.tsv 55MB (restricted dataset)
- acousticbrainz-mediaeval-discogs-train.tsv 127MB
- acousticbrainz-mediaeval-discogs-validation.tsv 26MB
- acousticbrainz-mediaeval-lastfm-train.tsv 62MB
- acousticbrainz-mediaeval-lastfm-validation.tsv 14MB
- acousticbrainz-mediaeval-tagtraum-train.tsv 59MB
- acousticbrainz-mediaeval-tagtraum-validation.tsv 13MB

Total size of the extracted files: 102GB
The folders contains subfolders ranging from 00-ff containing JSON files which itself contain precomputed Audio features extracted with Essentia[5] from the community-built AcousticBrainz database. The .tsv files contains features in the format: recordingmbid releasegroupmbid genre1 genre2 genre3 ... genre39 genre40.
The field *recordingmbid* is the MusicBrainz identifier of the particular recording, whereas *releasegroupmbid* is a MusicBrainz identifier of a release group (an album, single, or compilation) that it belongs to. The related genres and subgenres are encoded into the columns *genre1-genre40*. For each recordingmbid there exists a file in the *acousticbrainz-mediaeval-train* folder.

B) Reproduction - Original Repository

The authors of the paper provided a link to a Github repository[6] which contains a data-preparation folder and a link to another repository[7]. This linked repository contains three experiments written by Oramas et al. . Based on the description of the experiments and on the number of features described in the paper (2669) we could identify experiment 3: *TISMIR Experiments (Singlelabel Classification)*[8] as the baseline approach.
With this information we inspected the *data-preparation/create_h5.py* file to find out how to prepare the data. The existing file *data-preparation/create_h5.py* requires some *.features.clean.std.csv* and *.clean.std.csv* as well as *.genres.csv* as input. None of them is available in the dataset.
Tries to contact the main author via two mail addresses (soramas@pandora.com, sergio.oramas@upf.edu) were not successful.

C) Reproduction - Additional Repository

Some online research led us to a recently created Github repository[9] which deals with the same paper. Additionally it has a preprocessing step included. Naturally, we cloned the repository and started inspecting *preprocessing.py*. The script describes the creation of the files as we described it as input for the original paper repository in section B).

(1) Datasets

With the code from Section C) we provided the dataset according to the script and started preprocessing, which consists of 3 steps:

- creating CSV feature file and a genres file per dataset

[1] https://multimediaeval.github.io/2018-AcousticBrainz-Genre-Task/, seen on 2020-01-30
[2] https://multimediaeval.github.io/2018-AcousticBrainz-Genre-Task/data/, seen on 2020-01-30
[3] https://zenodo.org/record/2553414, https://zenodo.org/record/2554044, seen on 2020-01-30
[4] https://drive.google.com/drive/folders/0B8wz5KkuLnI3RjFYSFY5TkJVU1U
[5] https://essentia.upf.edu/, seen on 2020-01-30
[6] https://github.com/MTG/acousticbrainz-mediaeval-baselines, seen on 2020-01-30
[7] https://github.com/sergiooramas/tartarus/tree/b214f66dd4e61e83edc45ffc5c280efe7318a1b6, seen on 2020-01-30
[8] denoted in the run_experiments.py as *ISMIR 2019 Experiments (Baseline for AcousticBrainz Genre Dataset Classification)*
[9] https://github.com/nikuya3/acousticbrainz-mediaeval-baseline, seen on 2020-01-30

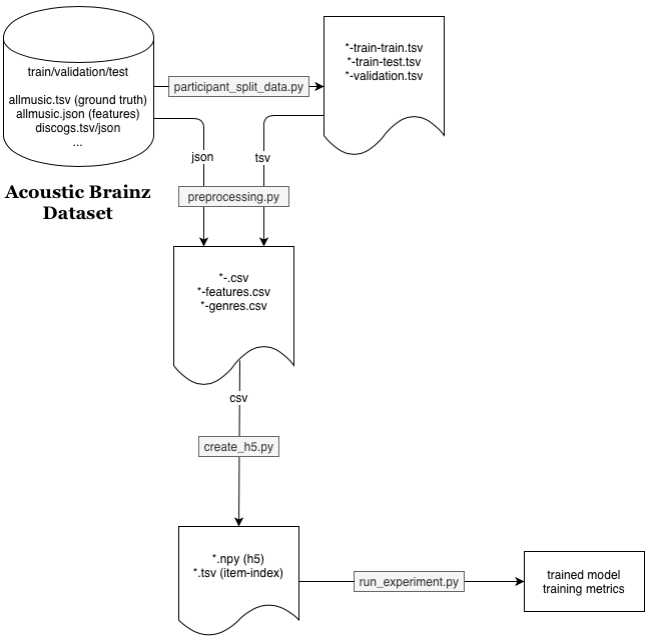

Figure 1: Preprocessing Data.

- calculating the mean and standard deviation of each dataset
- scaling each dataset

The output of the steps is written into *processed* folder. Even though the repository is very new and seems to handle the missing preprocessing step, an error occured in step 3:

```
Preprocessing train mode of allmusic dataset
Preprocessing validation mode of allmusic dataset
Preprocessing train mode of tagtraum dataset
Preprocessing validation mode of tagtraum dataset
Preprocessing train mode of discogs dataset
Preprocessing validation mode of discogs dataset
Preprocessing train mode of lastfm dataset
Preprocessing validation mode of lastfm dataset
Finished first preprocessing pass
Calculated means and standard deviations
Scale preprocessed datasets
Scaling processed/train/allmusic
Traceback (most recent call last):
...
ValueError: Unable to coerce to Series,
  length must be 2668: given 2669
```

Therefore *preprocessing.py* has been adapted. The idea is to use the preprocessing step of the second repository to generate the files for the data-preparation step of the original repository. Afterwards the training should have all required files to start (see figure 1).

(2) Preprocessing

- preprocessing.py python3
- participant_split_data.py
- create_h5.py python2 with h5py

Problems:

- TypeError: No conversion path for dtype: dtype('<U38') https://github.com/h5py/h5py/issues/1131
  -> solved with using python2

(3) Train

all python3

- python run_experiments.py genre_allmusic
- python run_experiments.py genres_discogs
- ...
- python run_experiments.py genres_allmusic_multimodal part of our Reproducibility run???

Problems:

- ValueError: Error when checking target: expected dense_5 to have shape (766,) but got array with shape (1,)

(4) GPU

Problems:

- Without GPU one Epoch with only few 100 items takes hours
- cuda on windows 10 fails: https://github.com/Theano/libgpuarray/issues/587

(5) Sources

(6) Doing

References

[BPUS17] Dmitry Bogdanov, Alastair Porter, Julián Urbano, and Hendrik Schreiber. The mediaeval 2017 acousticbrainz genre task: content-based music genre recognition from multiple sources. 01 2017.