Dataset creation using Last.fm tags

Audio and Music Processing Lab

Roman Tsukanov
SMC

March 15, 2016

1 Introduction

The goal of this lab was to use Last.fm tags (other than genre) to create a dataset. Dataset then needs to be evaluated in AcousticBrainz project[1].

Last.fm tags come from a data dump, which contains tags mapped to MusicBrainz IDs (MBIDs)[2] of recordings. Each tag is associated with normalized count (0 being the least common tag and 100 the most common tag). Count indicates how many times users assigned a particular tag to a recording (track) on Last.fm. All recordings in that data dump have at least one low-level data submission on AcousticBrainz, so no filtering is required.

All scripts (Jupyter Notebooks[3]) and other materials are hosted at `https://github.com/gentlecat/amp-lab/tree/master/8_acousticbrainz`.

2 Preparing and analyzing the data

I used Python programming language for working with data and generating datasets.

2.1 Parsing

Data dump was provided to us in CSV format[4]. Each row contains MBID of a recording, followed by pairs of values: tag value, count for that tag.

To simplify further processing I created two Python dictionaries. First maps recording MBID to a list of {*tag, normalized count*} tuples. Second maps tags to a list of {*recording MBID, normalized count*} tuples.

[1] `https://acousticbrainz.org/`
[2] `https://musicbrainz.org/doc/MusicBrainz_Identifier`
[3] `https://jupyter.org/`
[4] `https://tools.ietf.org/html/rfc4180`

2.2 Looking at tags

To see what tags are more popular I sorted them by number of occurrences. As a result I got a set of tags, which was used to define what kinds of datasets it is possible to create[5].

- **Mood**: [”happy”], [”sad”]
- **Female/Male**: [”female vocalists”, ”female vocalist”, ”female vocals”, ”female”], [”male vocalists”, ”male vocalist”, ”male vocals”, ”male”]
- **Quality of content**: [”good”, ”awesome”, ”amazing”, ”great”], [”bad”, ”awful”, ”terrible”, ”garbage”]
- **Origin**: [”american”, ”usa”], [”british”, ”uk”], [”german”, ”deutsch”, ”germany”], [”spanish”, ”spain”]
- **Rating (out of 10)**: [”0 of 10 stars”], [”1 of 10 stars”] ... [”10 of 10 stars”]
- **Decade**: [”20s”], [”30s”] ... [”90s”], [”2000s”, ”00s”], [”2010s”, ”10s”]

This list describes structure of datasets that I built. Each class can include recordings with one or more of specified labels. For example, it makes sense to combine recordings with tags “male vocalists” and “male vocalist”, they mean pretty much the same thing in our context (Female/Male classification).

3 Creating datasets for AcousticBrainz

3.1 Generating datasets from tags

There are a couple of things to keep in mind when generating datasets:

1. Classes need to have roughly the same number of items to prevent bias.
2. There shouldn’t be more than 1,000 instances in a class, because evaluation might take too much time.

After looking at total number of instances I picked one that had lowest count or 1,000 (if it’s more than that) and used it as a limit for all classes in a dataset.

Then I sorted all items in a class by normalized count. This was done to make sure that only items with higher count are used, assuming that they are better candidates for a particular class.

[5] I decided to create multiple datasets.

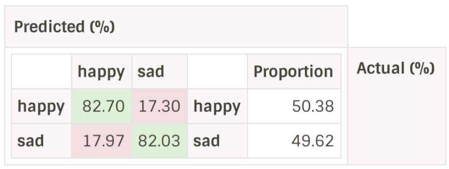
3.2 Importing datasets into AcousticBrainz

AcousticBrainz provides a way to import datasets in CSV format, so this is what I used to save them. Each row contains {*recording MBID, class name*} pairs.

All datasets can be viewed on my AcousticBrainz profile page at `http://acousticbrainz.org/user/Gentlecat`. I submitted all of them for evaluation and already got some results.

4 Results

I got evaluation results for two datasets: "Mood" and "Quality of content". "Mood" classifier (Figure 1) has accuracy of 82.37%. "Quality of content" classifier (Figure 2) has accuracy of 71.11%.

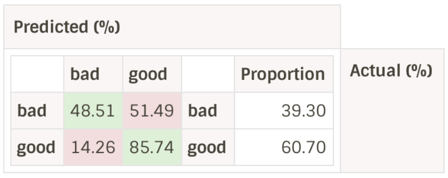


Predicted (%)

Actual (%)

	happy	sad		Proportion
happy	82.70	17.30	happy	50.38
sad	17.97	82.03	sad	49.62

Figure 1: Confusion table for "Mood" classifier



Predicted (%)

Actual (%)

	bad	good		Proportion
bad	48.51	51.49	bad	39.30
good	14.26	85.74	good	60.70

Figure 2: Confusion table for "Quality of content" classifier

Both results seem to be pretty good. It would be nice to check them against the rest of data that wasn't included into these datasets. Unfortunately, right now there's no easy way to do that.

Evaluation of other datasets has failed because of some issues in evaluation script.