Assignment 1: Music Genre Classification

Barbara Engelhardt, Princeton University out 02/03/2016; due 02/23/16

Background

When you scan stations on the radio, your brain is able to hear arbitrary snippets of music and classify them into a musical genre very quickly with only a small segment of audio. Inspired by this, we will use this first assignment to perform *multi-class classification*: given a song file, you will extract features and classify each song into one of ten music genres. There is a competition for music genre classification, with the best non-human classifiers getting 93% accuracy. Selecting songs for listeners is an important task related to the problem of collaborative filtering: given the listening preferences of a listener, what other songs might they enjoy? What type of songs should we avoid? Many music apps use these ideas to decide what music to play for us.

Project definition

Your goal in this homework project is to use a data set consisting of 1,000 30 second song files with genre labels (the GTZAN dataset from a standard challenge in MIREX[1]) to build a music genre classifier, or software that classifies a song snippet as one of 10 music genres, *blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, rock*. **Your main task is to build (multiple) multi-class classifiers that take as input the song feature sets and the song genre labels.** Feel free to use the classifiers we have or will discuss in class as well as others mentioned in our text books, described in the scientific literature, or implemented in software. You may also use more sophisticated classifiers (see *Extensions*). Because of the large number of possible features, we recommend using some type of feature selection to reduce the number of features. Finally, you should evaluate the classifiers, using a training dataset, test dataset, and validation dataset.

We provide an audio representation and a digital representation of each 30 second song snippet. All features for each song are depicted in Tables 1 and 2 along with their descriptions and dimensionality. Song-level features are described in Table 3. With the exception of MFCC, the features are extracted using the Matlab MIRToolbox Lartillot and Toiviainen [2007]. The data can be downloaded from Github[2] by typing the following command into your terminal.

```
git clone https://github.com/grgliner/voxResources.git
```

See the accompanying README.txt file on Github for the initial steps in the process.

[1] http://www.music-ir.org/mirex/wiki/MIREX_HOME

[2] You must have Github installed to clone a repository, otherwise the directory can be downloaded directly from Github

In addition to the data features described above, there is an **optional** Matlab script to generate a number of music features from each of the song snippets, creating, for each song, a bag-of-words representation from a vocabulary of musical features. A tutorial for the script is provided and will be presented at precept. You are also free to generate your own features from the data set or extend the existing feature set in interesting and well-motivated ways (see *Extensions*, below).

Essential to any data analysis task is the interpretation of the results. When completing the assignment consider the following. What features were most important for classification, and what do these features tell us about music and music genres? What is worse: classifying a song incorrectly, or not classifying a song at all? What types of genres were easy to classify for all approaches, and on what types of genres did the classifiers disagree? Simply building a machine learning approach to solve the problem does not constitute a data analysis; recovering and characterizing signal from these results does.

Deliverables

Your deliverables for this project include:

- A four page (not including citations) summary of the project work, which should contain (as described in the Example project write up on Piazza):
  - A title, authors' names, and abstract for the project;
  - an introduction to the problem being addressed;
  - a description of the data, and how they were split into training/test sets/validation sets;
  - a description of the methods developed and used, and how they were fitted using training data;
  - a presentation of the results of the methods applied to the test data;
  - a discussion of the results, including specific examples of songs and musical features that highlight the behavior of the classification models;
  - a short summary and conclusion, including extensions that you believe would be particularly valuable based on the results;
  - a *complete* bibliography to support the databases, feature selection, classifiers, code bases, and related work that are relevant to your project.
- please upload the code that you used to fit the classifiers as part of your assignment.

Please put your PDF write up of the project into
`https://dropbox.cs.princeton.edu/COS424_S2016/Assignment1`
by midnight on the assignment due date, with the file name `<author1PUID>_<author2PUID>_hw1.pdf`.
You should only submit one PDF and one tar file of code per pair of authors.

We strongly recommend *writing as you go* in the project, which means starting to write the project report as you are downloading and analyzing the data. That said, you should avoid speculative writing, and only write results once you have them.

Extensions

If you would like to extend this assignment to more interesting ground after first completing the basic deliverables for the project, you might consider the following:

- *Extend the data set*: The songs compiled here are not anywhere near comprehensive. There are a number of publicly available music databases. Processing and incorporating other data sets—including ones you personally compile—and releasing these data with appropriate permissions would be worthwhile. You can also check out Magnatune [3] for songs with crowd sourced genre labels.
- *More interesting features*: while we have provided a script for encoding the temporal information through Fisher vectors and exemplars, but there are many extensions to this to consider. This includes:
  - optimizing hyperparameters for the Fisher vector and exemplar representation
  - developing new methods to encode the temporal dimension of the feature space
  - developing more sophisticated feature selection methods
- *More classifiers*: there are a number of exciting classifiers that might be used for this task. You can try some methods that are not covered in class or something of your own design. *Ensemble classifiers* that combine music genre classifications from a number of classifiers to improve results may be built from a number of the more simple classifiers used in your basic analyses.
- *Better evaluation metrics*: what are better metrics that you might use to evaluate these classifiers? Time wasted listening to songs from genres you don't like (false positives)? Missing songs that obviously belong in specific genres (false negatives)?
- *Additional types of problems*: What about songs that do not have genre labels? You might consider developing an active learning method that will ask users to classify songs into music genes that will, in expectation, reduce uncertainty maximally across all unlabeled songs. What about adaptive music genre classifiers that can be refitted as new types of genres or new generations of specific genres arise?

Resources

Automatic genre classification dates back to 1997 when Dannenberg et. al. applied machine learning to musical style classification for interactive performing systems Dannenberg et al.

[3] http://musicmachinery.com/2009/04/01/magnatagatune-a-new-research-data-set-for-mir/

[1997]. In 2002, Tzanetakis and Cook at Princeton University set the milestone by using a large set of musical features, including timbre features, rhythmic variations, etc., to achieve a 58% accuracy on a 10-genre classification task Tzanetakis and Cook [2002]. Many new methods have been developed in this area, including AdaBoost classifiers Bergstra et al. [2006] (82.5% accuracy on GTZAN), high-level musical features McKay and Fujinaga [2004] and non-negative tensor factorization Panagakis et al. [2008] (78.2% accuracy on GTZAN). The current winning method in MIREX borrowed insights from computer vision Costa et al. [2012], Alam et al. [2015]. They treat the spectrogram as a texture image, extract visual patches such as local binary pattern textures (LBP) and then aggregate them into a single descriptor using, for example, a histogram. The reported performance on GTZAN is 88.60% Wu and Jang [2015] using this approach. However, higher accuracies on GTZAN data set exist using factorization methods such as compressive sampling Chang et al. [2010] and non-negative matrix factorization Panagakis and Kotropoulos [2010].

Click here for Python audio resources: https://wiki.python.org/moin/Audio/

References

Mohammad Rafiqul Alam, Mohammed Bennamoun, Roberto Togneri, and Ferdous Sohel. A confidence-based late fusion framework for audio-visual biometric identification. *Pattern Recognition Letters*, 52:65 – 71, 2015.

James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. Aggregate features and adaboost for music classification. *Machine Learning*, 65(2-3):473–484, 2006.

Kaichun K. Chang, Jyh shing Roger Jang, and Costas S. Iliopoulos. Iliopoulos: “music genre classification via compressive sampling. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR*, pages 387–392, 2010.

Y. M. G. Costa, L. S. Oliveira, A. L. Koerich, F. Gouyon, and J. G. Martins. Music genre classification using lbp textural features. *Signal Process.*, 92(11):2723–2737, November 2012.

Roger B. Dannenberg, Belinda Thom, and David Watson. A machine learning approach to musical style recognition. In *Proc. International Computer Music Conference*, pages 344–347, 1997.

Olivier Lartillot and Petri Toiviainen. A matlab toolbox for musical feature extraction from audio. In *International Conference on Digital Audio Effects*, 2007.

Cory McKay and Ichiro Fujinaga. Automatic genre classification using large high-level musical feature sets. In *ISMIR*, volume 2004, pages 525–530. Citeseer, 2004.

Ioannis Panagakis, Emmanouil Benetos, and Constantine Kotropoulos. Music genre classification: A multilinear approach. In *in Proceedings of ISMIR*, pages 583–588, 2008.

Y. Panagakis and C. Kotropoulos. Music genre classification via topology preserving non-negative tensor factorization and sparse representations. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 249–252, March 2010.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.

Ming-Ju Wu and Jyh-Shing R. Jang. Combining acoustic and multilevel visual features for music genre classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 12(1):10:1–10:17, August 2015.

Table 1: Frame-Level Audio Features

Name (Dim)	**Description**
MFCC ($\mathbb{R}^{24\times n}$)	Mel-Frequency Cepstrum Coefficients A timbrel representation (character of a sound)
Chroma ($\mathbb{R}^{12\times n}$)	Captures the distribution of energy along pitches
Energy ($\mathbb{R}^{1\times n}$)	Total intensity of each frame
Zero-Crossing ($\mathbb{R}^{1\times n}$)	Indicator of the noisiness
Spectral Flux ($\mathbb{R}^{1\times n}$)	Difference between the spectrogram of each successive frame

Table 2: Frame-Level Music Features

Name (Dim)	**Description**
Roughness ($\mathbb{R}^{1\times n}$)	An estimation of sensory dissonance
Key Strength ($\mathbb{R}^{12\times n}$)	A score between -1 and 1 for each key candidate via a cross-correlation of the chroma
Onsets ($\mathbb{R}^{1\times n}$)	Determines whether there is an onset in the frame
HCDF ($\mathbb{R}^{1\times n}$)	Harmonic Change Detection Function The flux of the tonal centroid
Inharmonicity ($\mathbb{R}^{1\times n}$)	Estimates the amount of partials that are not multiples of the fundamental frequency

Table 3: Song-Level Features

Name	**Description**
Key	The estimated key for the song
Tempo	An estimate of the tempo by detecting periodicity's from onset detection