Tools for audio analysis of Indian music

Xavier Serra

Music Technology Group

Universitat Pompeu Fabra, Barcelona

Software tools

- Python 3.0: https://www.python.org
- Jupyter: https://jupyter.org
- Docker: https://www.docker.com
- Docker image: https://github.com/MTG/MIR-toolbox-docker
- Essentia: http://essentia.upf.edu
- Dunya API: https://github.com/MTG/pycompmusic
- MIR course: https://github.com/MTG/MIRCourse

Data

- Dunya corpora: https://dunya.compmusic.upf.edu
- Compmusic datasets: https://compmusic.upf.edu/datasets

T2 & T3 set up

$ git clone https://github.com/MTG/MIRCourse.git
$ cd MIRCourse/
$ docker-compose up

! Mac users should launch Docker before this command

Access http://localhost:8888 using a browser, password: *mir*

Docker image: https://github.com/MTG/MIR-toolbox-docker

Contains:

Common data-science packages: numpy, scikit-learn, pandas, etc.

MIR tools: mir-eval, music21

MTG tools: Essentia, freesound-python, pycompmusic, Jupyter server

ESSENTIA

Software library and tools for audio and music analysis, description and synthesis

- Extensive collection of reusable algorithms
- Written in C++ and optimized for computational speed
- Python bindings for fast prototyping
- Feature extractors for **large-scale audio analysis**
- **Cross-platform** (Linux, Mac OS X, Windows, iOS, Android, and JavaScript)
- Support for mobile platforms and **real-time** processing

ACM Multimedia '13 Best Open Source Software Award

```
from essentia.standard import *
audio = MonoLoader(filename='audio.mp3')()
beats, bconfidence = BeatTrackerMultiFeature()(audio)
audio = EqualLoudness()(audio)
melody, mconfidence = PitchMelodia(frameSize=2048, hopSize=128)(audio)
```

D. Bogdanov, N. Wack, et al. 2013. "ESSENTIA: an Audio Analysis Library for Music Information Retrieval", ISMIR.

ESSENTIA

Over 200 algorithms for audio signal processing and analysis, and sound and music description, developed at MTG

- Standard audio IO & DSP
- Sound and music descriptors
  - spectral features
  - rhythm and tempo
  - tonality, pitch and melody
  - loudness/dynamics
  - sound envelope
  - audio segmentation
  - fingerprinting
- Machine-learning based descriptors
  - genres, moods, instrumentation, …
  - SVM classifiers
  - TensorFlow deep learning models

Music extractor

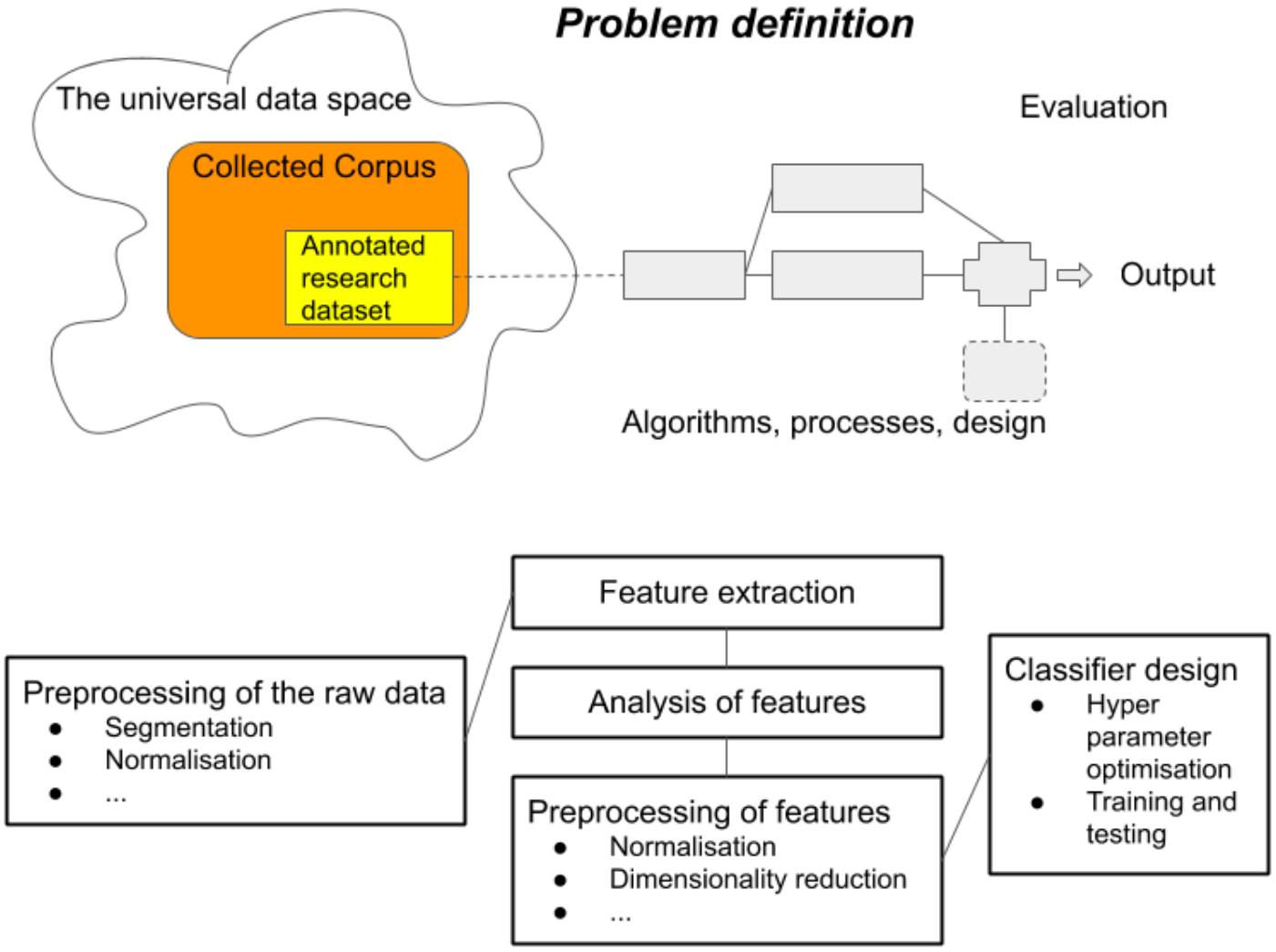
`essentia_streaming_extractor_music` is a configurable command-line feature extractor that computes a large set of spectral, time-domain, rhythm, tonal and high-level descriptors. Using this extractor is probably the easiest way to get many common music descriptors out of audio files using Essentia without any programming. The extractor is suited for batch computations on large music collections and is used within AcousticBrainz project. The prebuilt static binaries of this extractor are available via Essentia website and AcousticBrainz website.

It is possible to customize the parameters of audio analysis, frame summarization, high-level classifier models, and output format, using a yaml profile file (see below). Writing your own custom profile file you can specify:

- output format (json or yaml)
- whether to store all frame values
- an audio segment to analyze using time positions in seconds
- analysis sample rate (audio will be converted to it before analysis, recommended and default value is 44100.0)
- frame parameters for different groups of descriptors: frame/hop size, zero padding, window type (see FrameCutter algorithm)
- statistics to compute over frames: mean, var, median, min, max, dmean, dmean2, dvar, dvar2 (see PoolAggregator algorithm)
- whether you want to compute high-level descriptors based on classifier models (not computed by default)

https://essentia.upf.edu/documentation/streaming_extractor_music.html

MIR classification task



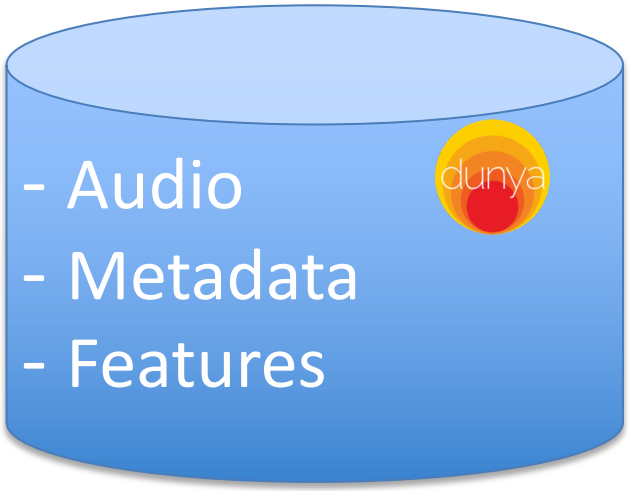
T1: Instrument classification

Understand (modifying and running):

https://github.com/MTG/MIRCourse/blob/master/notebooks/Lecture1_IntroWithACaseStudy.ipynb

Apply it to the automatic classification the different mridangam strokes using this dataset: https://compmusic.upf.edu/mridangam-stroke-dataset, or table strokes using this dataset: https://compmusic.upf.edu/tabla-solo-dataset

Dunya Server



pycompmusic

Python tools for accessing audio, annotations and features.

https://github.com/MTG/pycompmusic

Features extracted using culture-specific tools

Accessing Dunya corpora using the API

Hands on: `DownloadDataFromDunya_noToken.ipynb`

This notebook demonstrates downloading data using pycompmusic API.

- downloading a single file using recording's MusicBrainz ID
- downloading files of a CompMusic dataset
(https://github.com/MTG/otmm_makam_recognition_dataset)

Music Brainz ID for a recording is simply the code you can retrieve from a MusicBrainz link to a recording.

Example: the recording
https://musicbrainz.org/recording/e666ec52-b752-492d-9423-24e1c7bffbc7
has a Music Brainz ID : 'e666ec52-b752-492d-9423-24e1c7bffbc7'

Accessing Saraga content

Hands on: `downloadAllSARAGAContent.ipynb`

This notebook demonstrates the use of Dunya api for downloading Saraga dataset files which includes manual annotation files accompanying recordings with Creative Commons licence (in mp3 format).

Saraga dataset is also available for direct download on
: https://doi.org/10.5281/zenodo.1256126

Saraga dataset is composed of two collections:

- Hindustani collection
- Carnatic collection

The notebook creates two subfolders and saves all data in these folders. Each annotation is saved in a separate text file.

T3: Corpus analysis

References

- MIR course: https://github.com/MTG/MIRCourse
- Essentia: http://essentia.upf.edu
- Dunya API: https://github.com/MTG/pycompmusic
- S. Gulati thesis code: