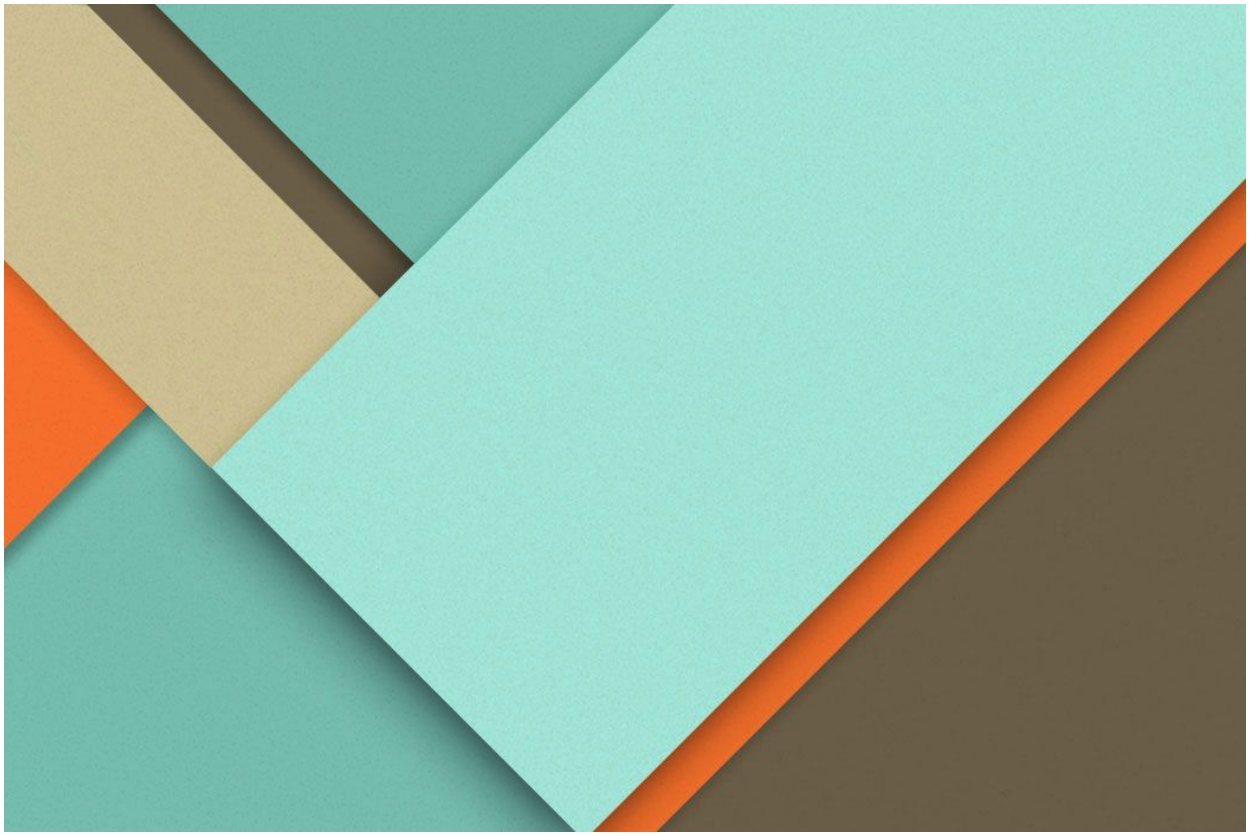

Music Pace Compatibility

02.09.2019

Andrew Harless
89 Pleasant Street
Watertown, MA 02472

Overview

A project to estimate multiple musical tempos from an audio segment. For example, a song marked at ♩=160, in 4/4 time with secondary stress on beat 3 and a lot of ♫, could be timed as 40, 80, 160, or 320 bpm depending how you define a beat, and depending on the particular sound of the song, which may or may not lend itself to any or all of these timings.

Beat detection and tempo estimation are standard problems, for which (imperfect) solutions exist. But one thing I've found trying to jog to music is that a single tempo estimate (even if it's accurate) doesn't necessarily solve the relevant problem. For example, if a song has ♩=105 but has a lot of 8th notes, you can jog with very quick steps (210 per minute) or with very slow steps (105). And a song compatible with a normal jogging pace (say 160) may not be recognizable as such from a single tempo estimate (e.g. maybe 80).

The problem is complicated by the fact that, while most songs have an even time signature (2 or 4), many have odd (usually 3), and some have both even and odd factors (6 or 12). So the relation of the official tempo to the possible alternatives differs from song to song.

Applications

Tempo estimation has numerous applications, but what particularly interests me is keeping pace for jogging, walking, marching, running, and so on. The present project adds a new dimension by allowing for multiple tempos.

A hypothetical client could be someone developing smartphone software to choose music for jogging/walking/running. Once a model is developed, it can be used to produce an ever-expanding database of songs with compatible tempos. The database could then be used to choose music compatible with an intended pace.

Data

The data will consist of a set of audio files, each containing a single song, and a human-labeled database specifying the tempos that are compatible with each song. I will provide the labels, at least initially. Presumably if the project continues, it could include other labelers.

Approach

1. Choose a set of songs with roughly constant tempos.
2. Hand-label songs with compatible paces.
3. Divide each song into segments of about 25 seconds each. Ignore the first and last segment of each song.
4. Using LibROSA, calculate a tempogram for each song segment, and find the peaks of the tempogram's time-averaged means. Collect a set of all the peaks (rounded to nearest integer) from all the segments of each song. Use these as "candidate tempos" for the song.
5. A candidate tempo is "correct" if it is within a certain tolerance (initially 5%) of one of the hand-labeled paces. The rest are "incorrect."
6. For use in initial training data, generate the Mel spectrogram (as a function of time, produced by LibROSA) for each song.
7. To generate each training case,
    - choose a labeled song (at random or according to a process later to be decided),
    - choose one of its candidate tempos (at random or...),
    - take a random clip (with a specified length in *number of beats* according to the candidate tempo, but without attempting to choose a phase relative to the beat) from the song (but not from the beginning or end of the song),
    - resample the spectrogram of the clip to make it a standard size,
    - reshape the spectrogram, adding a dimension, so that the time segment associated with each beat has a separate row, and
    - associate the result with the binary label as to whether the candidate tempo is correct for the song.
8. Train a binary classifier to distinguish correct and incorrect tempos.

Deliverables

1. Code for data preparation
2. Code for model fitting
3. Database of compatible paces for test data
4. Slide deck presentation describing the project
5. A report describing the project, including EDA and data story