A STEP TOWARDS AUTOMATIC IDENTIFICATION OF INFLUENCE: LICK DETECTION IN A MUSICAL PASSAGE

Anis Haron
Media Arts & Technology, University of California Santa Barbara

ABSTRACT

Using techniques borrowed from speech recognition, this paper presents an algorithm to detect known licks in a musical passage, in which its results could be used to identify who influenced a given improviser.

A lick can be understood as a known phrase, or a relatively short pattern consisting of a series of notes used in solos or melodic lines. Jazz trumpeter Clark Terry mentioned, the art to learning jazz (improvisation) consists of imitation, assimilation, and innovation. Ethnomusicologist, John Murphy explained in a paper on jazz improvisors and their precursors, "by invoking and reworking music that is familiar to the audience, the jazz performer involves the audience in the process and makes it meaningful for those who recognize the sources." As an analytical perspective on the concept of influence, Murphy went on with extensive analysis of several improvisations by saxophonist Joe Henderson, in which he quoted and transformed a melody segment (lick) by Charlie Parker.

1. ALGORITHM

1.1 Licks

For this demo we have arbitrarity choosen a lick consisting of a sequence of seven notes. The sequence starts on the tonic, followed by the major second, minor third and perfect fourth. It is then followed by descending three semitones from the perfect fourth back to major second, followed by a minor seventh, and finally resolving back to tonic. In the key of C major, this sequence translates to the note sequence C, D, Eb, F, D, Bb, C. It is important to note that a lick could be played in any key, at any speed and possibly with some degree of rhythmic variation and ornamentation.

1.2 Source separation

Since we will be implementing our algorithm on audio files, the first step is to separate unwanted sound sources. For example, if we were to analyze the singer (singer improvises), sources from the accompanying musicians should be excluded or the very least, suppressed as much as possible. For this task, we will use Flexible Audio Source Separation Toolbox (FASST) developed by Ozerov et. al. [1].

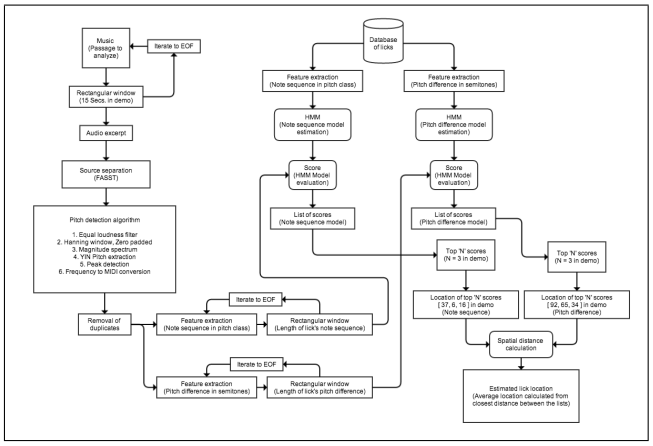

Figure 1. Algorithm overview.

The source we will be using in this demonstration is a 15 seconds excerpt from Melody Gardot's Baby I'm a Fool, taken from 1' 55" to 2' 10" mark in the song. This particular phrase is chosen as the lick we are looking for was quoted in this phrase. Futhermore, the song carries a calm mood hence its sparse accompaniment, making it an ideal candidate for good source separation results in our demo.

1.3 Pitch extraction

For pitch extraction, we will use Essentia, an open source audio analysis library, developed by Bogdanov et. al. of the Music Technology Group at Universitat Pompeu Fabra [2].

Firstly, an equal loudness filter is applied to our signal. As the human ear does not perceive all frequencies as the same loudness, the equal loudness filter compensates this known issue by applying an inverted approximation of the equal-loudness contour. A hanning window is applied to chunks of the signal, zero padded to increase frequency resolution, and its magnitude spectrum calculated. YIN algorithm is then applied to the magnitude spectrum, returning a list of pitch detected along with a list of values for confidence level. Next, we run a peak detection algorithm to detect local maximas. We arbitrarily use the mean of confidence level values as the thresholding level for peak detection. This way, only pitch which confidence are above average are considered correct, ignoring detected

pitches with below average confidence value. It should be noted that the chosen threshold value might have to be tweaked for use with other sources. Our peak detection algorithm returns position (time) along with the amplitude (frequency) of the peaks. Lastly, we convert our list of frequencies to MIDI note, rounding up to the nearest number.

1.4 Removal of duplicates

The next step is to remove any duplicates in our note list. This step is necessary as our algorithm runs in an overlapped, windowed segment. As our window moves across the signal, the algorithm is bound to repeat detection of the same note, provided it lasts longer than our window. Time corresponding to note duplicates should also be removed.

1.5 Pitch difference as an additional feature

As mentioned earlier, we know a lick could be played in any key, therefore it makes sense to use pitch difference (interval size) as an additional feature. In this demonstration, the unit for pitch difference will be in semitones.

First we will need to transpose our MIDI notes into their respective pitch classes, removing octaves. Pitch difference is calculated by subtracting current note (pitch class value) with the previous note. This is done for both the passage we are analyzing and our known lick.

1.6 Hidden Markov Models (HMM)

HMM is a statistical technique used in various fields, most notably in speech [3], hand-writing and gesture recognition. HMM is an unsupervised, generative probabilistic model whereby a sequence of observable variables (our known lick in this instance) is generated by a sequence of unobservable states. For this task, we will use scikit-learn, an easy to use, open source machine learning library for Python programming language.

At this point, we need to estimate the most likely series of states that could generate our observed data. The two features used are our observed data. Using scikit's 'fit' method, we estimate the most likely hidden states for each of our observed data, accumulating to a pair of models in this demonstration. One for the sequence of notes in pitch class, and another for the sequence of pitch difference.

Next, we compute the probability that the note sequence in the passage we are analyzing contains smaller sequences that could be generated using our two models. We will refer to this process as the evaluation. For this task, we first define a function to return 'n' number of top scores, arbitrarily chosen to be n=3 in this demonstration.

1.6.1 Evaluating the note sequence model and pitch difference sequence model

In this step we will be evaluating our known lick's estimated note sequence model against the passage we are analyzing (Melody Gardot's Baby I'm a Fool, 15 seconds excerpt). Note that we have already transposed each note in the passage we are analyzing into its respective pitch classes earlier.

To evaluate, we first apply a rectangular window with the length of our known lick note sequence (7 in this instance), to the passage we're analyzing. Note sequence of the analyzed passage inside the rectangular window are compared against our known lick note sequence model using scikit's 'score' method. The rectangular window is then iterated at every point up to the end of sequence. At every iteration, a probability score under the compared model is recorded. Lastly, we list the location of three highest scores, as we have chosen n=3 earlier.

We will also be evaluating our estimated pitch difference sequence model against the passage we are analyzing. Since pitch difference would remain the same even when the observed passage is transposed to any key, this additional feature is to assist in filtering out false positives. Just as before, evaluation for pitch difference sequence is carried out, listing location of three highest scores.

1.7 Distance computation and estimation of lick location in the analyzed passage

At this stage, we now have the top three location points for sequences that could be generated with our known lick note sequence model, and pitch difference model. We now calculate the spatial distance between the two list of vectors. The nearest distance between the two list would suggest the estimated location of our known lick in the analyzed passage.

In this demonstration, the estimated location of our known lick is estimated at around the 4.5 seconds mark in our 15 seconds excerpt. Since our 15 seconds excerpt started at the 1'55" mark, the final position of the lick is estimated at roughly around the 2'00" mark of the song. [1]

2. CONCLUSION

The algorithm presented have demonstrated the possibility of locating a known phrase embedded in a given musical passage. It suggests the possibility of tracing the influence of an improviser given the availability a relevant database. Additionally, to make the algorithm more robust, additional features should be extracted and studied to increase the probability of successful detection.

3. REFERENCES

[1] Alexey Ozerov, Emmanuel Vincent, Frdric Bimbot: "A general flexible framework for the handling of prior information in audio source separation," *IEEE Transactions on Audio, Speech and Signal Processing*, pp. 1118–1133, 2012.

[2] Bogdanov, Dmitry, et al: "Essentia: An audio analysis library for music information retrieval," *Proceedings of ISMIR*, 2013.

[3] Lawrence R. Rabiner: "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, pp. 257–286, 1989.

[1] Melody Gardot's Baby I'm a Fool is available for streaming on Melody Gardot's VEVO channel in YouTube.