Automatic Rhythmic Notation from Single Voice Audio Sources

Jack O'Reilly, Shashwat Udit

Introduction

In this project we used machine learning technique to make estimations of rhythmic notation of a sung melody from its raw audio waveform. Notating music by hand is a tedious and time consuming task, and the benefits from automation in allowing musicians to easily collaborate and transfer their ideas would be substantial. However, of the various instruments employed to produce melodies in music, the human voice is among one of the most difficult instruments in which to determine note onsets. While some instruments which mechanically produce consistent pitches corresponding to each note, the human singing voice exhibits continuous variation in pitch even during the middle of a note, and these variations are frequently more pronounced towards the beginning and end of a note. There is no absolute physical or mathematical definition for note onset with a human voice. Instead, listeners are frequently able to discern the onset of new note through the use of contextual clues. With our machine learning techniques, we attempted to match what a skilled human listener would transcribe if they were listening to a performance. The data input for are algorithm is the waveform itself and the output is the rhythmic notation that indicates the note onsets and durations. To accomplish this task, we divided the problem into two sub-problems. The first is estimation of note start and stop times from the audio waveform. The second sub-problem is to fit this result, given in a set of onset times and note durations, to the likeliest musical notation.

Requirements of the Dataset

Before searching for an appropriate data set, we determined the type of sample data and ground truth data that would be necessary for tackling the problem. An appropriate data set would need to include raw audio recordings of a single instrument or singer. It would need to have note onset and offset times in order to be appropriate for the first problem. To be useful for the notation problem, it would need to include musical notation for the monophonic melody.

We found two data sets that fulfilled some of these requirements. The first is the SVNOTE data set, which contains monophonic melodies and provides onset and offset times as well as estimated note pitch values for ground truth. The melodies are sung without accompaniment by non-professional singers. The TONAS data set provides a similar set of samples and ground truth values for unaccompanied vocal performances of flamenco music by trained flamenco singers. These data sets have an unfortunate shortcoming – the ground truth values for note onset and duration come from a computer-aided manual transcription process.

Dataset Generation

After spending some time working with both the TONAS and the SVNOTE datasets, we became increasingly suspicious that the nature by which the datasets were generated was resulting in relatively noisy ground truth data for training our algorithms. Here, we decided to investigate generating our own dataset. Taking inspiration from the MAPS dataset, which used computer-controlled acoustic pianos to generate extremely accurate onset and offset data, we decided to use a professional virtual software instrument to generate monophonic voice signals and accurate ground truth data simultaneously. Using Sibelius, we generated a set of musical notation files for single voices. Using the Sibelius Manuscript scripting language, we then generated synthesized audio files via playback through the Voxos Epic Virtual Choir plugin and simultaneously created metadata files describing precise onset, offset, and fundamental frequency values. In an attempt to reduce overfitting to a particular sound, we employed

the full functionality of the virtual instrument; i.e., the dataset included staccato and legato passages, and every available syllable was represented.

For algorithm training and evaluation, we reserved 70% of the dataset for training and 30% for evaluation. In order to ensure that both portions of the dataset were representative of the signals likely to be encountered throughout, each portion was randomly shuffled. The random seed used for shuffling was fixed for all feature types and learning strategies.

Feature Selection

Qualities of Audio Signals and Singing

Before diving into particular machine learning algorithms, it was important to understand what features are typically considered significant when analyzing audio data, particularly with regard to note onset and offset. Immediately rising to mind were the energy envelope, the instantaneous pitch, and the autocorrelation of the waveform. In particular, a high rate of change in the signal energy for a given signal window may signify a note onset or offset. Likewise, a change in the estimated fundamental frequency for a given signal window could indicate the same.

Naïve Windowing

The first approach we took was one we came to call the 'naïve windowing' method. In this method, our feature vectors are simply the raw waveform values of a subsection of the audio signal. Intuitively, this is unlikely to be an optimal approach to the problem for a few reasons. By creating an n-sample sequence into an n-dimensional vector, we may be throwing away information that adjacent samples are likely to be highly correlated. The results from this approach were poor, as expected, and they do not appear in the results section of this document.

Windowed DFT

The next approach we took was to window the signal sample and compute the discrete time Fourier transform. We used the Kaiser window and a sample length of 100ms. Intuitively, this corresponds somewhat closer to the musical features of an audio waveform than the naïve windowing method, as it approximately corresponds to energy represented in frequency bins (and therefore musical pitch). However, this approach still produces a vector of high dimensionality proportional to the sample length, and it suffers from many of the same problems as naïve windowing.

MFCC with Derivatives

One approach to feature selection was inspired by classical audio processing techniques, as discussed in [1]. Here, we took windows of the audio signal and computed the mel-frequency cepstral coefficients (MFCC). These coefficients can intuitively be thought of as representing the energy represented within a signal sample mapped to discrete frequency bins on the log-frequency scale. This approach is psychoacoustically motivated, as humans perceive musical pitch on a logarithmic basis. We used 23 cepstral coefficients. In addition, we appended the feature vector with approximations the first and second derivatives of these coefficients, resulting in a feature dimension of 69.

For computing the cepstral coefficients, we concerned ourselves with two frequency ranges. The choice of the first was motivated by the fundamental frequency range we saw in the data set: approximately 250 Hz to 1000 Hz. For the second range, we doubled the upper end in order to incorporate greater harmonic content and effects of signal transients.

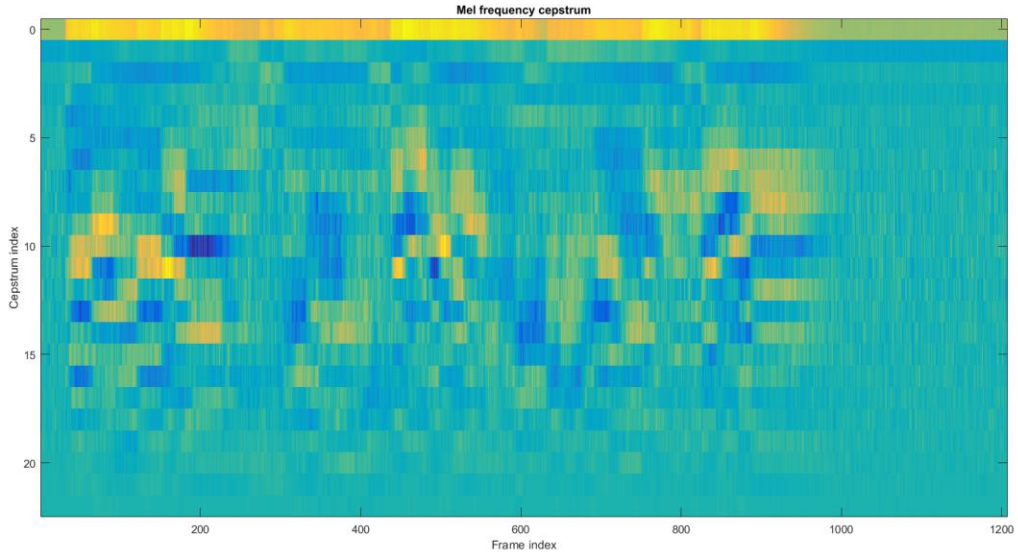

Figure 1 - Mel Frequency Cepstrum for a Sample File

PCA

For some feature selection strategies, we reduced the feature dimension by employing principal component analysis. In particular, we performed analysis using PCA on feature vectors produced by the windowed DFT approach as well as the MFCC approach. The number of principal components used was determined experimentally by searching through powers of two up to the original feature vector dimension, and better performing values were selected.

Algorithmic Approaches

Onset Detection

SVM

For each feature selection approach, we trained SVMs (using libSVM) with three different kernels: the linear kernel, polynomial kernel with degree 3, and the radial basis kernel. The data set was normalized to have zero mean and values bounded within [-1, 1]. Feature vectors were classified into two classes: those corresponding to signal windows containing a note onset and those without. However, almost every single feature selection type produced a severely optimistic or pessimistic model, either classifying every signal window as containing an onset or classifying no windows as such. The only exception was using the linear kernel with the DFT + PCA (k = 16) feature set, which, although it produced a model that successfully classified the test data into more than a single class, otherwise performed poorly.

EM with Gaussian Mixture Model

This approach, which proved the most successful, involved training two Gaussian Mixture Models (GMMs) with the EM algorithm, each corresponding to a classification type (onset-containing or non-onset-containing). Given a test vector input, we compared prior probabilities calculated according to each of the two models in order to decide the likelihood of an onset's presence.

Best Musical Notation Fit with Naïve Bayes

To generate rhythmic notation, determining the note onset and offset is the only the first part of the problem. We must also turn the raw intervals into note lengths. This is difficult, however—in addition to varying pitches during a note, vocalists often slightly vary the length of a note. When we transcribe the notation we aim to convey the intended length, not the note that is actually sung.

One potential advantage we had as we attempted to generate rhythm notation is that there are certain psychological and cultural bases of music that come into play, as music assigns high probability to certain note durations. In particular, notes in western music almost always have a duration which is a linear combination of products of reciprocals of 2 or 3 where we define a measure to have duration of exactly one. If a note as marked by the onset and offset times appears to be between a quarter and a fifth of measure, it is almost certainly a quarter note -- quint notes are rarely used in western music. The machine learning technique used to take advantage of our knowledge about the prior likelihood of note lengths is the naïve Bayes algorithm which is shown below:

$$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x)}$$

Here y=1 is the chance that the note is question is equivalent to a particular note value; say a quarter note, and x is observed interval between note start and stop. Use of the naïve Bayes algorithm requires knowledge of the tempo of the melody in beats per minute or equivalent. This is a reasonable assumption and necessary for precise transcription, since there's no other way to distinguish between half notes at 60 beats per minute and quarter notes at 120 beats per minute. If we used correct note onset and offset times and a reasonable training set of note probabilities, the Naïve Bayes classifier was relatively accurate in accurately classifying notes with around 10% of notes merged together and roughly double of that split. This was encouraging because it reinforced that the main challenge in transcription was detecting note onset and offset, not converting those into notation. Further, this created interest in whether the Naïve Bayes classifier could be used to detect note onset and offset, i.e., whether a certain interval was likely to be two eighth notes, a quarter note, part of a longer note etc. There the results largely did not exceed random guessing, but with the intriguing exception that accuracy was higher if both the training set and the sample being analyzed where very similar in style.

Results and Discussion

Feature Selection	Algorithm	Onset Class Error	Non-Onset Class Error
DFT, PCA with k = 16	EM, #mixtures = 16	0.4911	0.1694
MFCC, 50 ms window, no PCA	EM, # mixtures = 16	0.3986	0.2083
MFCC, 100 ms window, no PCA	EM, #mixtures = 4	0.1355	0.4135
MFCC with high frequencies, 50 ms window, no PCA	EM, # mixtures = 16	0.4704	0.1066
MFCC with high frequencies, 100 ms window, no PCA	EM, # mixtures = 4	0.1592	0.4132
MFCC, 50 ms window, PCA with k = 32	EM, # mixtures = 16	0.4471	0.2745

These results indicate that there is a significant tradeoff between sensitivity and precision for the feature types and learning models that we employed. Although we were not able to train a model that simultaneously achieves high precision and recall, we noted that further filtering of the prior probability functions (when computed continuously, using a sliding window for signal samples) may

improve onset detection after further empirical observations. Indeed, many current algorithms use feature fusion and non-trivial detection functions as a computation layer after the GMM model to produce high-performing detection.

Conclusion:

Generally, the best results were obtained when we could make strong, accurate assumptions as in our Naïve Bayes classifier for certain audio samples, or when we had sophisticated feature vectors such as the mel-frequency cepstral coefficients. This points to the twofold path forward towards more accurate transcription. For the first and more uncertain approach, it may be possible to combine machine learning algorithms of the type used in this project with other machine learning algorithms that endeavor to take the entire sample and classify the style of music of piece in the melody of the whole. Such algorithms would need to have considerable power to break down music into much finer pieces than commonly recognized genres the and tell when a piece has influence of multiple styles before it is possible to make confident assumptions about the distribution of likelihood of various note lengths, but the possibility cannot be ruled out. It is not unlikely that when human listeners transcribe music, their expectations are affected by knowledge of the style of music.

The second path is what we would put the bulk of our efforts into if we had additional resources in the form of team members, time, or computational resources: refinement of the feature vector. From our experience in this project, a feature vector that can reliably distinguish between a frame in which a note onset occurs and a frame in which a transition did not occur is unlikely to simple, but we have demonstrated that there are feature vectors that contain information about the likelihood of onsets and that some feature vectors produce more accurate results than others. The challenge that remains in order to achieve accurate transcription and commercialization is one of optimization.

Citations

[1] C. C. Toh , B. Zhang and Y. Wang "Multiple-feature fusion based onset detection for solo singing voice", *Proc. Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, 2009

[2] Chih-Chung Chang , Chih-Jen Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST), v.2 n.3, p.1-27, April 2011

[3] Viitaniemi, Timo, Anssi Klapuri, and Antti Eronen. "A probabilistic model for the transcription of single-voice melodies." *Proceedings of the 2003 Finnish Signal Processing Symposium, FINSIG'03*. 2003.