Automated transcription of polyphonic piano music using structured SVM classification

Charles Hermann CIH5
Jonathan Shi JS2845

Abstract

We investigate the use of maximum-margin Markov classifiers in the automated transcription of polyphonic piano music. Existing work on this problem has involved separate and independent smoothing and classification steps. The use of a maximum-margin Markov classifier allows us to unify these two steps.

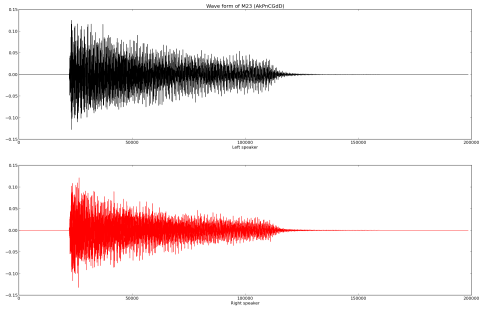

Figure 1. Wave of C1 (Low C)

1. Introduction

1.1. Motivation

One of the major areas of research in music information retrieval (MIR) is automated music transcription, the process of creating a musical score from an audio recording. Though this can be done manually by experienced musicians, this process is difficult and time consuming. As a result, the ability to automatically transcribe music has numerous positive ramifications in the musical community and the MIR community. In addition, transcription presents an interesting and difficult challenge even for seemingly easily versions of the task such as transcribing piano solos. Several factors make this a difficult area of study including but not limited to: the high-dimensionality of the datasets, the large margin for error (both false positives and false negatives), and the lack of a field-accepted feature vector/technique for extracting the data.

1.2. Feature analysis

An audio signal consists of a series of samples of wave displacement over time. As is often convenient, we process this audio input to obtain samples of signal intensities in various frequencies over time. This is called the short-term fourier transform (STFT).

Using the STFT, we obtain spectograms, which are plots of which particular frequencies are present at any particular time within a signal.

Features that we observe within a spectogram include:

Overtones. Each note, when played, consists of a fundamental frequency, which is the pitch of the note that we perceive, as well as a set of overtones at integer multiples of the original frequency.

Partials. A partial of a note is one of the theoretically pure sine wave constituents of that note. The fundamental frequency will be one of the partials, and each overtone will contribute another partial. Rather than being pure sine waves, which would show up in the spectograph as straight think horizontal lines, in reality each partial will be a bit fuzzy and show up as blurred horizontal bars in the spectograph.

It is also notable that some partials might fade faster than others, for a particular note, and that they sometimes disappear entirely for brief moments before reappearing within the same note.

Missing fundamentals. In certain cases, a human will hear a note being played at a certain pitch even if the fundamental frequency is omitted from the signal. This is because the human audio processing system can infer the fundamental frequency from the present overtones.

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

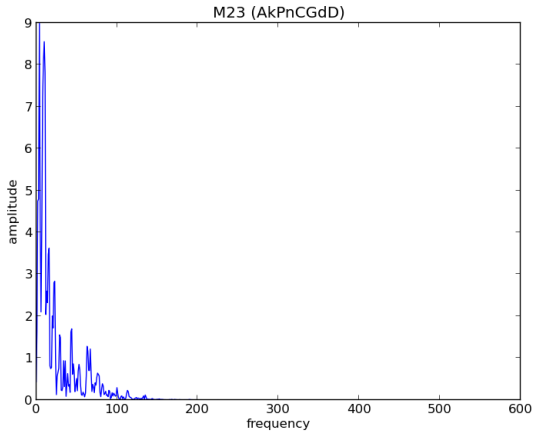

Figure 2. Frequency graph shortly after onset of C1 (Low C)

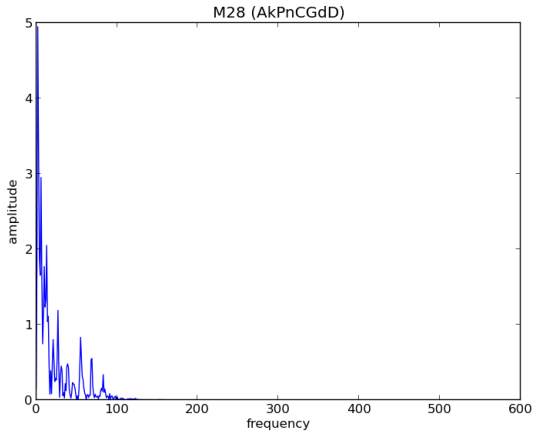

Figure 3. Frequency graph shortly after onset of E1 (Low E)

Beat frequencies. When two distinct frequencies are played over each other, they give rise to a beat frequency equal to the difference of the two original frequencies. Human ears hear these as subjective tones, and some previous attempts have incorrectly detected beat frequencies as new note onsets (**?**).

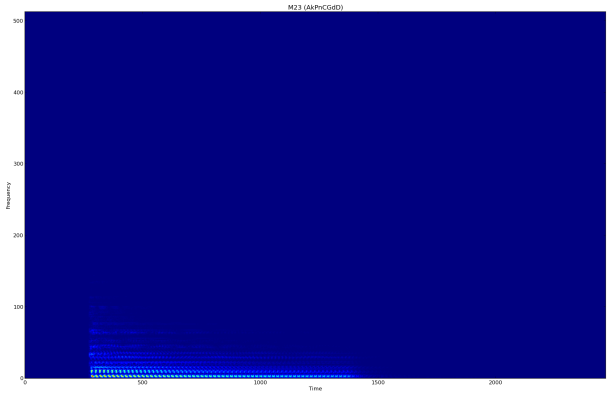

Figure 4. Frequency map of note C1 (Low C)

Previous work on this task have focused mostly around:

1. Discriminative SVMs and CRFs trained to detect individual notes or note events at the time windows when they occur (**?**) (**?**) (**?**).
2. Hidden Markov networks for particular chords (**?**).
3. Neural network approaches using recurrent networks (**?**) (**?**).
4. Estimation and then subtraction, or expectation-maximization, of sets of predetermined overtones corresponding to each possible fundamental frequency. (**?**).

Some previous work has incorporated feature sets from spectrograph data that have been specially tuned for the transcription task. These include a pitch salience function (**?**), deeply learned feature sets (**?**), features acquired via sparse nonnegative matrix factorization (**?**), and features generated by a model of human inner ear hair cells (**?**). Also feature sets have been obtained via deep learning (**?**) and sparse nonnegative matrix factorization (**?**), which have performed better in classification-based transcription than manually specified features.

Difficult points for previous classifiers have also included octave errors, since notes that are an octave apart from each other are very similar in their partials (**?**) (**?**). Previous methods also experienced difficulties in distinguishing which notes were onset and which were just held when some note onset event was detected (**?**).

There has been previous work (**?**) dealing with the use of max-margin methods to transcribe musical notes.

Our approach differs substantially from theirs because they apply a max-margin Markov network to classify sets of pre-segmented partials according to which instrument they belong to. Meanwhile we skip the analysis of partials and attempt to classify notes directly from the spectograms as features.

2. Methods

The core method being introduced is to leverage a maximum-margin Markov model (**?**) for classification on spectral data, which improves upon an HMM by using SVM techniques to find optimal separating hyperplanes that respect correlations between components of output labels.

2.1. Notation

Some notation to help in formalization:

We'll use $A \otimes B$ denotes the outer product of two vectors. More explicitly:

$$(A \otimes B)_{n_B i+j} = A_i B_j,$$

where n_B is the number of components in B. Informally, this is a new vector representing the set of all pairwise products of components of A with components of B.

Similarly, $A \oplus B$ will denote the direct sum, so that:

$$(A \oplus B)_i = \begin{cases} A_i & \text{if } i \leq n_A \\ B_{i-n_A} & \text{if } i > n_A. \end{cases}$$

Informally, this is simply concatenating the two vectors A and B.

2.2. Maximum-margin Markov network

We set up a Markov network such that correlations exist only between:

1. different pitches within the same time window, and
2. pitches that occur in adjacent time windows.

In the framework of Markov networks, we minimize an energy function $\psi(\mathbf{x}, \mathbf{y})$, which we will decompose as $\psi(\mathbf{x}, \mathbf{y}) = \sum_t \psi_t(\mathbf{x}, \mathbf{y})$, with one energy function per time window in the signal. A maximum-margin Markov network then sets:

$$\psi_t(\mathbf{x}, \mathbf{y}) = \exp[\mathbf{w}^T \mathbf{f}_t(\mathbf{x}, \mathbf{y})],$$

for some vector $\mathbf{w}$ that we train using SVM methods, and some vector of features $\mathbf{f}_t(\mathbf{x}, \mathbf{y})$.

2.3. Features

We use as our features essentially raw spectral data: A STFT of the audio data sampled at using the and with window size

We space the different windows

Since we are training a model that leverages correlations in note labels between adjacent time points, we also include the features from adjacent time points. This way the Markov network can leverage *changes* in intensity at each frequency in order to help predict changes in note labels.

Thus, if we let $\mathbf{x}_\omega(t)$ denote the signal intensity at frequency ω and time t, and similarly let $\mathbf{y}_\rho(t)$ denote an indicator variable for whether pitch ρ is on at time t, our expression for the feature vector is:

$$\begin{aligned} \mathbf{f}_t(\mathbf{x}, \mathbf{y}) = h_t(\mathbf{y}) \oplus \\ [\mathbf{x}(t+1) \oplus \mathbf{x}(t) \oplus \mathbf{x}(t-1)] \\ \otimes [\mathbf{y}(t+1) \oplus \mathbf{y}(t) \oplus \mathbf{y}(t-1)], \end{aligned}$$

where h_t encodes the Markov transition probabilities for each particular time step of the note labeling.

3. Data

Training and testing data were sourced from the MAPS dataset (**?**), a set of 31 GB of piano recordings in .wav format, recorded on varying pianos and in varying acoustic environments, with ground truth transcription labels provided.

We separate our data into training, validation, and test sets so that.

4. Evaluation

We evaluate according to the standard set in (**?**) and (**?**). In both of these papers, they worked with a large library of mp3 files accompanied by midi files. Their software produced output which could be directly compared with the midi files and measured for: notes correctly predicted, notes failed to be predicted, and notes incorrectly predicted.

5. Conclusion