A CLASSIFICATION APPROACH TO MELODY TRANSCRIPTION

Graham E. Poliner and Daniel P.W. Ellis
LabROSA, Dept. of Electrical Engineering
Columbia University, New York NY 10027 USA
{graham,dpwe}@ee.columbia.edu

ABSTRACT

Many melody transcription systems rely on a core of rule-based analysis that assumes a specific audio structure, namely that a musical pitch is realized as a set of harmonics of a particular fundamental. This assumption is strongly grounded in musical acoustics, but it is not strictly necessary. In this abstract, we present a system that learns to infer the correct melody label based only on training with labeled examples. Our algorithm performs dominant melodic note classification via a Support Vector Machine classifier trained directly from audio feature data. As a result, the proposed system may be easily generalized to learn many melodic structures or trained specifically for a given genre.

Keywords: Melody Transcription, Classification

1 SYSTEM DESCRIPTION

The basic flow of our transcription system is as follows: First, the input audio waveform is transformed into a normalized short-time magnitude spectrum feature representation. A Support Vector Machine (SVM) trained on real multi-instrument recordings and synthesized MIDI audio classifies each frame as having a particular dominant pitch, quantized to the semitone level. Finally, energy thresholding is used to perform voiced/unvoiced melody segmentation. Each of these steps is described in more detail below:

1.1 Acoustic Features

The input audio waveform is downsampled to 8 kHz and converted to the short-time Fourier transform (STFT) using an $N = 1024$ point Discrete Fourier Transforms (i.e. 128 ms), an N-point Hanning window, and a 944 point overlap of adjacent windows (for a 10 ms grid). Only the bins corresponding to frequencies below 2 kHz (i.e. the first 256 bins) are used. To improve generalization across different instrument timbres and contexts, the magnitude of the STFT is normalized within each time frame to achieve zero mean and unit variance over a 71-frame local frequency window, the idea being to remove some of the influence due to different instrument timbres and contexts in train and test data.

1.2 Support Vector Machine

Labeled audio feature vectors are used to train an SVM with a class label for each note distinguished by the system. The WEKA implementation of Platt's Polynomial Sequential Minimal Optimization (SMO) SVM algorithm was used to map the frequency domain audio features to the MIDI note-number classes (Witten and Frank, 2000; Platt, 1998). The default learning parameter values (C = 1, epsilon = 10^{-12}, tolerance parameter = 10^{3}) are used to train the classifier. Each audio frame was represented by a 256-element input vector, with sixty potential output classes spanning the five-octave range from G2 to F#7 for N-way classification.

1.3 Training Data

1.3.1 Multi-track Recordings

A set of multi-track recordings was obtained from genres such as jazz, pop, R&B, and rock. For each song, the fundamental frequency of the melody track was estimated using the YIN fundamental frequency estimator (de Cheveigne and Kawahara, 2002). Fundamental frequency predictions were calculated at 10 ms steps and limited to the range of 100 to 1000 Hz. Only frames with periodicity of at least 95% were used as training examples. To align the acapella recordings to the full ensemble recordings, we performed Dynamic Time Warp (DTW) alignment between STFT representations of each signal, along the lines of the procedure described in Turetsky and Ellis (2003). Target labels were assigned by calculating the closest MIDI note number to the monophonic prediction at the times corresponding to the STFT frames.

1.3.2 MIDI Files

Our MIDI training data is composed of frequently downloaded pop songs from www.findmidis.com. The training files were converted from the standard MIDI file format to mono audio files (.WAV) with a sampling rate of 8 kHz using the MIDI synthesizer in Apple's iTunes. The MIDI files were parsed into data structures containing the relevant audio information (i.e. tracks, channels numbers, note events, etc). The melody was isolated and extracted by exploiting MIDI conventions for representing the lead voice. Target labels were determined by sampling the MIDI transcript at the precise times corresponding to each

Table 1: Results of the formal MIREX 2005 Audio Melody Extraction evaluation from http://www.music-ir.org/evaluation/mirex-results/audio-melody/. Results marked with * are not directly comparable to the others because those systems did not perform voiced/unvoiced detection. Results marked † are artificially low due to an unresolved algorithmic issue.

Rank	Participant	Overall Accuracy	Voicing d'	Raw Pitch	Raw Chroma	Runtime / s
1	Dressler	**71.4%**	**1.85**	68.1%	71.4%	32
2	Ryynänen	64.3%	1.56	**68.6%**	**74.1%**	10970
3	Paiva 2	61.1%	1.22	58.5%	62.0%	45618
4	Poliner	61.1%	1.56	67.3%	73.4%	5471
5	Marolt	59.5%	1.06	60.1%	67.1%	12461
6	Paiva 1	57.8%	0.83	62.7%	66.7%	44312
7	Goto	49.9%*	0.59*	65.8%	71.8%	211
8	Vincent 1	47.9%*	0.23*	59.8%	67.6%	?
9	Vincent 2	46.4%*	0.86*	59.6%	71.1%	251
10	Brossier	3.2%* †	0.14 * †	3.9% †	8.1% †	41

STFT frame in the analysis of the synthesized audio.

1.4 Segmentation

Voiced/Unvoiced melody classification is performed by simple energy thresholding. The sum of the magnitude squared energy over the frequency range $200 < f < 1800\ Hz$ is calculated for each 10 ms frame. Each frame is normalized by the median energy value for the given song, and segments are classified as voiced or unvoiced with respect to a global threshold.

2 Results

The results of the formal MIREX 2005 Audio Melody Extraction evaluation are show in table 1. While "Raw Pitch" and "Raw Chroma" measure the accuracy of the dominant melody pitch extraction (measured only over the frames that were tagged as containing melody in the ground truth, and where the latter ignores octave errors), the "Overall Accuracy" combines pitch accuracy with correct detection of unvoiced frames; the "Voicing d'" figure indicates the accuracy of the detection of frames that do or do not contain melody (d' is the separation between two unit-variance Gaussians that would give the observed false alarm and false reject rates for some choice of threshold).

Calculating statistical significance for these results is tricky because the classification of individual 10 ms windows is highly non-independent – in most cases, two temporally-adjacent frames will correspond to virtually identical classification problems. Each individual melody note comes much closer to an independent trial: we estimate that there are about 2000 such trials in the test set, which consisted of 25 musical excerpts from a range of styles of between 10 s and 40 s in length. Given this many trials, and assuming the error rates remain the same at the note level, a one-tailed binomial significance test requires a difference in error rates of about 2.4% for significance at the 5% level for results in this range. Thus, roughly, for overall accuracy the performance differences between the rank 1 (Dressler) and 2 (Ryynänen) systems are significant, but the next three (including ours at rank 4) are not significantly different. Raw pitch and chroma, however, give another picture: For pitch, our system is in a three-way tie for top performance with the top two ranked systems, and when octave errors are ignored we are insignificantly worse than the best system (Ryynänen in this case), and almost significantly better than the top-ranked system of Dressler.

The fact that Dressler's system performed best overall even though it did not have the highest raw pitch accuracy is because it combined high pitch accuracy with the best voicing detection scheme, achieving the highest d'. Our voicing detection scheme, which consisted of a simple adaptive energy threshold, came in a joint second on this measure. Because voicing errors lead to false negatives (deletion of pitched frames) and false positives (insertion of pitch values during non-melody times), this aspect of the algorithm had a significant impact on overall performance. Naturally, the systems that did not include a mechanism to distinguish between melody and accompaniment (Goto, Vincent, and Brossier) scored much lower on overall accuracy despite, in some cases, raw pitch and chroma performance very similar to the higher-ranked systems.

We note with some regret that our system failed to score better overall than Paiva's 2nd submission despite exceeding it by a healthy margin on the other measures. This paradoxical result is explained in part by the fact that the voicing d' is calculated from all frames pooled together, whereas the other measures are averaged at the level of the individual excerpts, giving greater weight to the shorter excerpts. Paiva 2 did better than our system on voicing detection in the shorter excerpts (which tended to be the non-pop-music examples), thus compensating for the worse performance on raw pitch. Also, although not represented in the statistics of table 1, the voicing detection of Paiva 2 had an overall higher threshold (more false negatives and fewer false positives), which turned out to be a better strategy.

The final column in table 1 shows the execution time in seconds for each algorithm. We see an enormous variation of more than 1000:1 between fastest and slowest systems – with the top-ranked system of Dressler also the fastest! Our system is expensive, at almost 200 times slower, but not as expensive as several of the others. The evaluation, of course, did not place any emphasis on execution time, and what we are seeing is that some authors, ourselves included, paid the minimum of attention to this aspect.

3 Conclusions

While our system did not perform top in the evaluation, it was very comparable to the top systems (except, perhaps, in runtime), showing that pure classification is a very viable approach; all the other systems used explicit models of pitch notes as consisting of harmonic partials or periodic waveforms. Our system is less mature than those of some of the participants, who have been refining pitch extraction for many years; we are excited to see if enhancements such as larger and more diverse training sets, and improved normalization to reduce variability due to different instrument types, can improve our results still further. Since our approach is so radically different to those of the other systems, there is no reason to assume that we will 'plateau' at a similar level of performance – although the closely-bunched performance in this evaluation is quite striking, suggesting perhaps that the remaining 30% of frames may be much more difficult to recognize correctly.

ACKNOWLEDGEMENTS

Many thanks to Beau Silver and Nick Kruge of the University of Miami Music Engineering program for donating multi-track recording training data. This work was supported by the Columbia Academic Quality Fund, and by the National Science Foundation (NSF) under Grant No. IIS-0238301. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

A. de Cheveigne and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *Journal Acoustic Society of America*, 111(4):1917–1930, 2002.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, Cambridge, MA, 1998.

R. J. Turetsky and D. P. W. Ellis. Ground-truth transcriptions of real music from force-aligned midi syntheses. In *Proc. Int. Conf. on Music Info. Retrieval ISMIR-03*, 2003.

I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, CA, USA, 2000. ISBN 1-55860-552-5.