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

Melodies provide an important conceptual summarization of polyphonic audio. The extraction of melodic content has practical applications ranging from content-based audio retrieval to the analysis of musical structure. In contrast to previous transcription systems based on a model of the harmonic (or periodic) structure of musical pitches, we present a classification-based system for performing automatic melody transcription that makes no assumptions beyond what is learned from its training data. We evaluate the success of our algorithm by predicting the melody of the ISMIR 2004 Melody Competition evaluation set and on newly-generated test data. We show that a Support Vector Machine melodic classifier produces results comparable to state of the art model-based transcription systems.

Keywords: Melody Transcription, Classification

1 INTRODUCTION

Melody provides a concise and natural description of music. Even for complex, polyphonic signals, the perceived predominant melody is the most convenient and memorable description, and can be used as an intuitive basis for communication and retrieval e.g. through query-by-humming. However, to deploy large-scale music organization and retrieval systems based on melody, we need mechanisms to automatically extract this melody from recorded music audio. Such transcription also has value in musicological analysis and various potential signal transformation applications. As a result, a significant amount of research has recently taken place in the area of predominant melody detection (Goto, 2004; Eggink and Brown, 2004; Marolt, 2004; Paiva et al., 2004; Li and Wang, 2005).

Previous methods, however, all 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 many fields (such as automatic speech recognition) it is possible to build classifiers for particular events without any prior knowledge of how they are represented in the features.

In this paper, we pursue this insight by investigating a machine learning system to generate automatic melody transcriptions. We propose 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.

2 SYSTEM DESCRIPTION

The basic flow of our transcription system is as follows: First, the input audio waveform is transformed into a feature representation as some kind of normalized short-time magnitude spectrum. 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. Each of these steps is described in more detail below:

2.1 Acoustic Features

The original music recordings are combined to one channel (mono) and downsampled to 8 kHz. This waveform $x[n]$ is converted to the short-time Fourier transform (STFT),

$$X_{STFT}[k,n] = \sum_{m=0}^{N-1} x[n-m] * w[m] * e^{-j2\pi km/N} \quad (1)$$

using an $N = 1024$ point Discrete Fourier Transforms (i.e. 128 ms), an N-point Hanning window $w[n]$, and a 944 point overlap of adjacent windows (for a 10 ms grid). In most cases, only the bins corresponding to frequencies below 2 kHz (i.e. the first 256 bins) were used. To improve generalization across different instrument timbres

and contexts, a variety of normalizations were applied to the STFT, as described in section 3.

2.2 Support Vector Machines

Labeled audio feature vectors are used to train an SVM with a class label for each note distinguished by the system. The SVM is a supervised classification system that uses a hypothesis space of linear functions in a high dimensional feature space in order to learn separating hyperplanes that are maximally distant from all training patterns. As such, SVM classification attempts to generalize an optimal decision boundary between classes. Labeled training data in a given space are separated by a maximum margin hyperplane through SVM classification. In the case of N-way multi-class discrimination, a majority vote is taken from the output of $N(N-1)/2$ pairwise discriminant functions. In order to classify the dominant melodic note for each frame, we assume the melody note at a given instant to be solely dependent on the normalized frequency data below 2 kHz. We further assume each frame to be independent of all other frames.

2.3 Training Data

A supervised classifier requires a corpus of pairs of feature vectors along with their ground truth labels in order to be trained. In general, greater amounts and variety of training data will give rise to more accurate and successful classifiers. In the classification-based approach to transcription, then, the biggest problem becomes collecting suitable training data. Although the number of digital scores aligned to real audio is very limited, there are a few directions that facilitate the generation of labeled audio. In this experiment, we investigate multi-track recordings and MIDI audio files as sources of training data.

2.3.1 Multi-track Recordings

Popular music recordings are usually created by layering a number of independently-recorded audio tracks. In some cases, artists (or their record companies) may make available separate vocal and instrumental tracks as part of a CD or 12" vinyl single release. The 'acapella' vocal recordings can be used as a source for ground truth in the full ensemble music since they will generally be amenable to pitch tracking with standard tools. As long as we can keep track of what times within the vocal recording correspond to what times in the complete (vocal plus accompaniment) music, we can automatically provide the ground truth. Note that the acapella recordings are only used to generate ground truth; the classifier is not trained on isolated voices (since we do not expect to use it on such data).

A set of 30 multi-track recordings was obtained from genres such as jazz, pop, R&B, and rock. The digital recordings were read from CD, then downsampled into mono files at a sampling rate of 8 kHz. The 12" vinyl recordings were converted from analog to digital mono files at a sampling rate of 8 kHz.

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. YIN defines a periodicity measure,

$$Periodicity = \frac{P_{PERIODIC}}{P_{TOT}} \quad (2)$$

where $P_{PERIODIC}$ is the energy accounted for by the harmonics of the detected periodicity, and P_{TOT} is the total energy of a frame; Only frames with periodicity of at least 95% (corresponding to clearly-pitched voiced notes) 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). This time alignment was smoothed and linearly interpolated to achieve a frame-by-frame correspondence. The alignments were manually verified and corrected in order to ensure the integrity of the training data. Target labels were assigned by calculating the closest MIDI note number to the monophonic prediction at the times corresponding to the STFT frames.

2.3.2 MIDI Files

The MIDI medium enables users to synthesize audio and create a digital music score simultaneously. Extensive collections of MIDI files exist consisting of numerous renditions from eclectic genres. Our MIDI training data is composed of 30 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.

To find the corresponding ground truth, 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. Commonly, the lead voice in pop MIDI files is represented by a monophonic track on an isolated channel. In the case of multiple simultaneous notes in the lead track, the melody was assumed to be the highest note present. Target labels were determined by sampling the MIDI transcript at the precise times corresponding to each STFT frame in the analysis of the synthesized audio.

2.3.3 Resampled Audio

In the case when the availability of a representative training set is limited, the quantity and diversity of the training data may be extended by re-sampling the recordings to effect a global pitch shift. The multi-track and MIDI recordings were re-sampled at rates corresponding to symmetric, semitone frequency shifts over the chromatic scale (i.e. $\pm 1, 2, \ldots 6$ semitones). The ground truth labels were scaled accordingly and linearly interpolated in order to adjust for time alignment. This approach created a more Gaussian training distribution and reduced bias toward specific keys present in the training set.

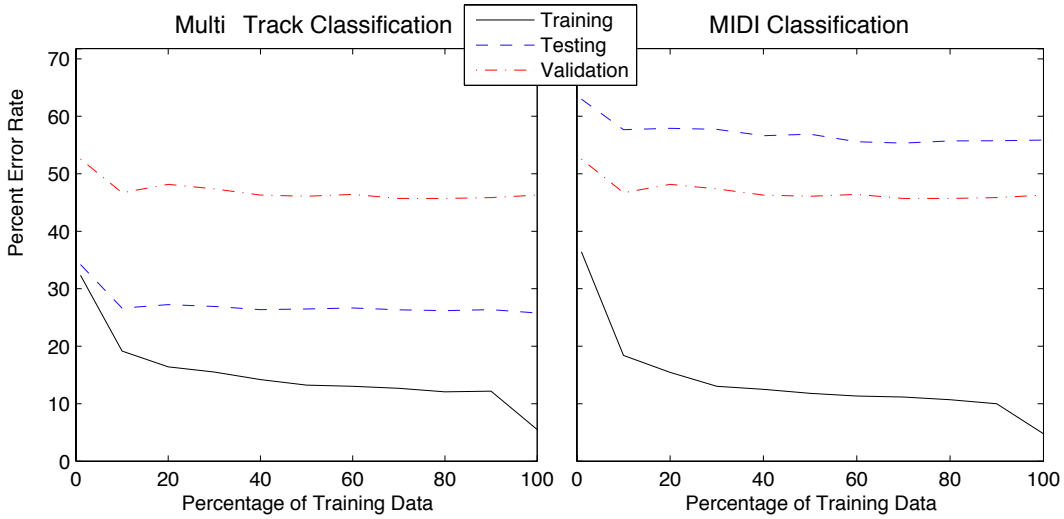

Figure 1: Variation of classifier frame error rate as a function of the amount of training data used, for training on real recordings (left) and MIDI syntheses (right). 100% of the training data corresponds to 30,000 frames or 300 s of audio. Curves show the accuracy on the training and test sets, as well as on the separate ISMIR 2004 set (see text).

3 EXPERIMENTS

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}) were used to train the classifiers. 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, and twelve potential output classes representing a one octave chroma scale for N-binary classification. Thirty multi-track recordings and thirty MIDI files with a clearly defined dominant melody were selected for our experiments; for each file, 1000 frames in which the dominant melody was present (10 s of audio data) were randomly selected to be used as training frames. Ten multi-track recordings and ten MIDI files were designated as the test set, and the ISMIR 2004 Melody Competition test set was used as a validation set (Gomez et al., 2004). This was an international evaluation for predominant melody extraction, the first of its kind, conducted in the summer of 2004. The evaluation data (which has now been released) consisted of 20 excerpts, four from each of 5 styles, covering a wide range of musical genres, and each consisting of about 30 s of audio. Following the conventions of that evaluation, to calculate accuracy we quantize the ground-truth frequencies for every pitched frame to the nearest semitone (i.e. to its MIDI number), and count an error for each frame where our classifier predicts a different note (or in some cases a different chroma i.e. forgiving octave errors). We do not, in this work, consider the problem of detecting frames that do not contain any 'foreground' melody and thus for which no note should be transcribed.

3.1 N-way Classification

We trained separate N-way SVM classifiers using seven different audio feature normalizations. Three normalizations use the STFT, and four normalizations use Mel-frequency cepstral coefficients (MFCCs). In the first case, we simply used the magnitude of the STFT normalized such that the maximum energy frame in each song had a value equal to one. For the second case, the magnitude of the STFT is normalized within each time frame to achieve zero mean and unit variance over a 51-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. The third normalization scheme applied cube-root compression to the STFT magnitude, to make larger spectral magnitudes appear more similar; cube-root compression is commonly used as an approximation to the loudness sensitivity of the ear.

A fourth feature configuration calculated the autocorrelation of the audio signal calculated by taking the inverse Fourier transform (IFT) of the magnitude of the STFT. Taking the IFT of the log-STFT-magnitude gives the cepstrum, which comprised our fifth feature type. Because overall gain and broad spectral shape are contained in the first few cepstral bins, whereas periodicity appears at higher indexes, this feature also performs a kind of timbral normalization. We also tried normalizing these autocorrelation-based features by local mean and variance equalization as applied to the spectra, and by liftering (scaling the higher-order cepstra by an exponential weight).

For all normalization schemes, we compared SVM classifiers trained on the multi-track training set, MIDI training set, and both sets combined. An example learning curve (based on the locally-normalized spectral data) is shown in figure 1. The classification error data was generated by training on randomly selected portions of the training set for cross validation, testing, and validation. The classification error for the testing and validation sets

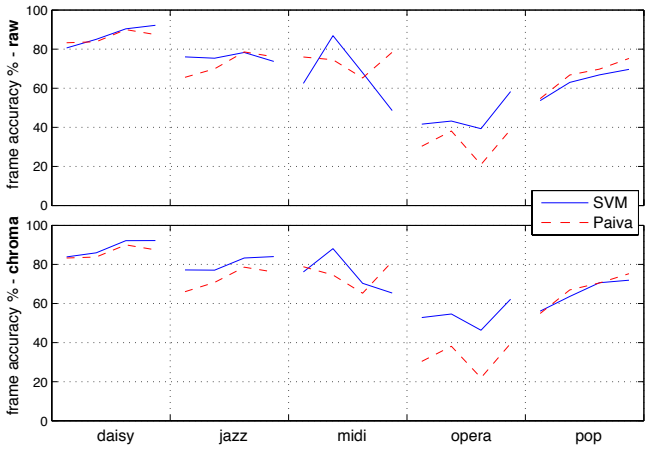

Figure 2: Variation in transcription frame accuracy across the 20 excerpts of the ISMIR 2004 evaluation set. Solid line shows the classification-based transcriber; dashed line shows the results of the best-performing system from the 2004 evaluation. Top pane is raw pitch accuracy; bottom pane folds all results to a single octave of 12 chroma bins, to ignore octave errors.

reaches an asymptote after approximately 100 seconds of randomly-selected training audio. Although the classifier trained on MIDI data alone generalizes well to the ISMIR validation set, the variance within the MIDI files is so great the classifier generalizes poorly to the MIDI test set.

Table 1 compares the accuracy of classifiers trained on each of the different normalization schemes. Here we show separate results for the classifiers trained on multi-track audio alone, MIDI syntheses alone, or both data sources combined. The frame accuracy results are for the ISMIR 2004 melody evaluation set and correspond to f0 transcription to the nearest semitone.

A weakness of any classification based approach is that the classifier will perform unpredictably on test data that does not resemble the training data, and a particular weakness of our approach of deliberately ignoring our prior knowledge of the relationship between spectra and notes is that our system cannot generalize from the notes it has seen to different pitches. For example, the highest f_0 values for the female opera samples in the ISMIR test set

Table 1: Frame accuracy percentages on the ISMIR 2004 set for each of the normalization schemes considered, trained on either multi-track audio alone, MIDI syntheses alone, or both data sets combined.

Normalization	**Multi-track**	**MIDI**	**ALL**
STFT	54.5	45.8	59.0
51-pt norm	52.7	51.3	**62.7**
Cube root	55.1	47.1	62.4
Autocorr	53.6	51.9	59.0
Cepstrum	48.5	44.7	52.1
NormAutoco	40.8	38.5	44.6
LiftCeps	53.4	48.6	60.3

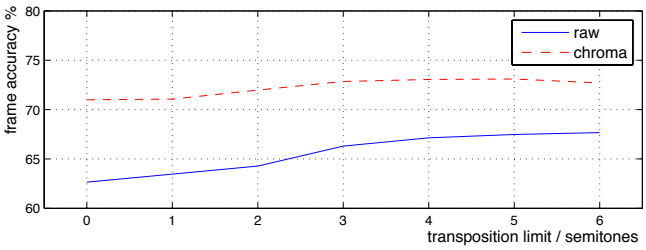

Figure 3: Effect of including transposed versions of the training data. As the training data is duplicated at all semitone transpositions out to ± 6 semitones, transposition frame accuracy improves by about 5% absolute for raw transcripts, and about 2% absolute for the chroma (octave-equivalent) transcription.

exceed the maximum pitch in all our training data. In addition, the ISMIR set contains stylistic genre differences (such as opera) that do not match our pop music corpora. However, if the desired output states are mapped into the range of one octave, a significant number of these errors are reduced. Neglecting octave errors yields an average pitched frame accuracy in excess of 70% on the ISMIR test set.

We trained six additional classifiers in order to display the effects of re-sampled audio on classification success rate. All of the multi-track and MIDI files were re-sampled to plus and minus one to six semitones, and additional classifiers trained on the resampled audio were tested on the ISMIR 2004 test set using the best performing normalization scheme. Figure 3 displays the classification success rate as the amount of re-sampled training data is varied from $\pm 1 \ldots 6$ semitones.

The inclusion of the re-sampled training data improves classification accuracy over 5%. In Figure 2, the pitched frame transcription success rates are displayed for the SVM classifier trained using the resampled audio compared with best-performing system from the 20 test samples from the 2004 evaluation, where the pitch estimates have been time shifted in order to maximize transcription accuracy (Paiva et al., 2004).

3.2 N Binary Classifiers

In addition to the N-way melody classification, we trained 12 binary SVM classifiers representing one octave of the notes of a western scale (the 'chroma' classes). The classifiers were trained on all occurrences of the given chroma and an equal number of randomly selected negative instances. We took the distance-to-classifier-boundary hyperplane margins as a rough equivalent to a log-posterior probability for each of these classes; Figure 4 shows an example 'posteriorgram', showing the variation in the activation of these 12 different classifiers as a function of time for two examples; the ground truth labels are overlaid on top. For the simple melody in the top pane, we can see that the system is performing well; for the female opera example in the lower pane, our system's unfamiliarity with the data is very apparent.

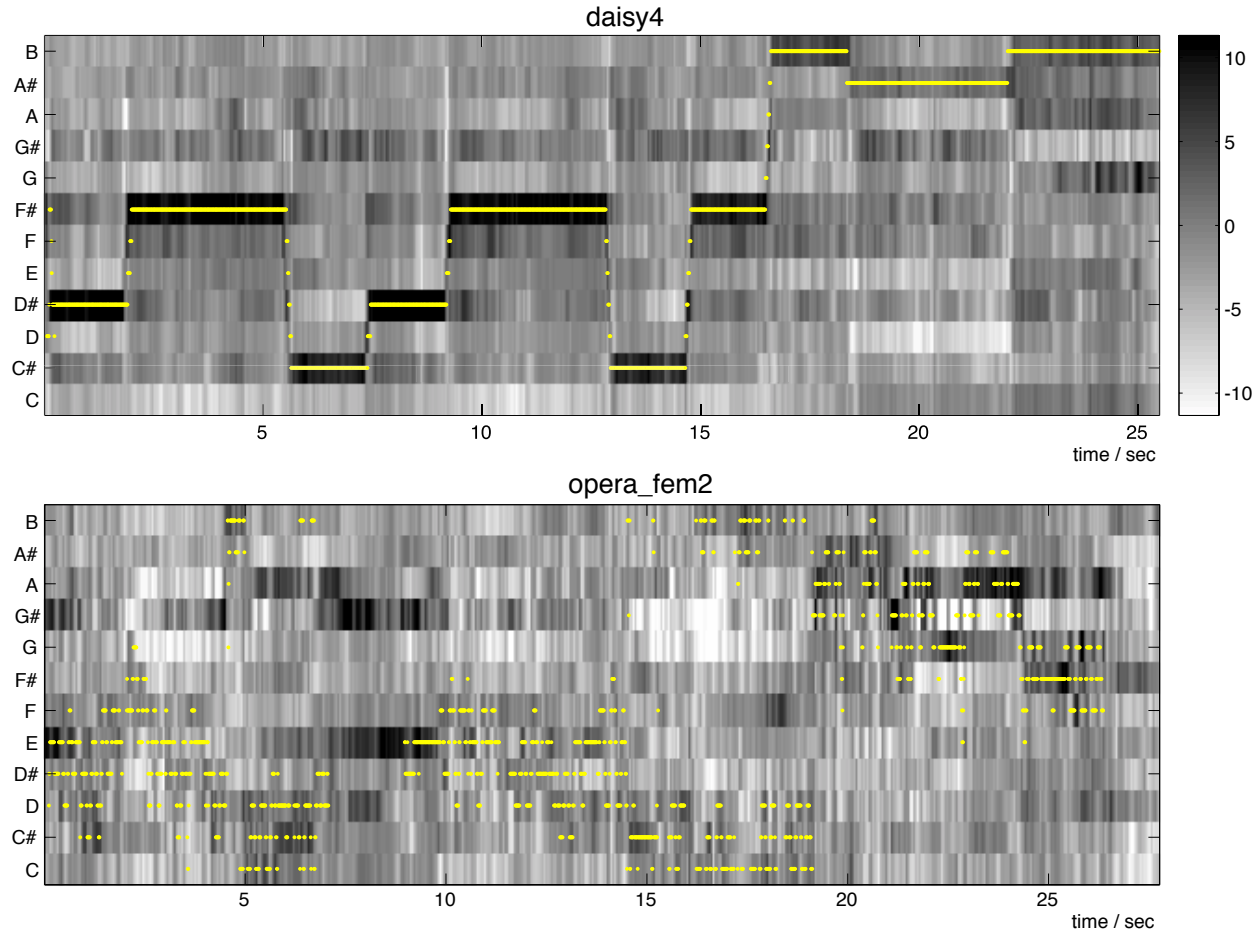

Figure 4: 'Posteriorgram' showing the temporal variation in distance-to-classifier boundary for 12 classifiers trained on the different notes of the octave. Ground-truth labels are plotted with dots. Top pane is a well-performing simple melody example. Bottom pane is a poorly-performing female opera excerpt.

4 DISCUSSION AND CONCLUSIONS

Looking first at table 1, the most obvious result is that all the features, with the exception of 'NormAutoco', perform much the same, with a slight edge for the 51-point across-frequency local-mean-and-variance normalization. In a sense this is not surprising since they all contain largely equivalent information, but it also raises the question as to how effective our normalization (and hence the system generalization) has been (although note that the biggest difference between 'Multi-Track' and 'MIDI' data, which is some measure of generalization failure, occurs for the first row, the STFT features normalized only by global maximum). It may be that a better normalization scheme remains to be discovered.

Looking across the columns in the table, we see that the more realistic multi-track data does form a better training set than the MIDI syntheses, which have much lower acoustic similarity to most of the evaluation excerpts. Using both, and hence a more diverse training set, always gives a significant accuracy boost – up to 10% absolute improvement, seen for the best-performing 51-point normalized features. We can assume that training on additional diverse data (particularly, say, opera) would further improve performance on this evaluation set.

As shown in figure 2, our classifier-based system is competitive with the best-performing system from the 2004 evaluation, and is a few percent better on average. This result must also be considered in light of the fact that there is no post-processing applied in this system. Instead, the performance represents scoring the raw, independent classification of each audio frame. Various smoothing, cleaning-up, and outlier removal techniques, ranging from simple median filtering through to sophisticated models of musical expectancy, are typically employed to improve upon raw pitch estimates from the underlying acoustic model.

This is the basis for our interest in the multiple parallel classifiers as illustrated in figure 4. By representing the outcome of the acoustic model as a probabilistic distribution across different notes, this front end can be efficiently integrated with a back-end based on probabilistic inference. In particular, we are investigating trained models of likely note sequences, starting from melodies extracted from the plentiful MIDI files mentioned above. We are further interested in hidden-mode models that can, for instance, learn and recognize the importance of latent

constraints such as the local key or 'mode' implied by the melody, and automatically incorporate these constraints into melody, just as is done explicitly in Ryynänen and Klapuri (2004).

We note that our worst performance was on the "opera" samples, particularly the female opera, where, as noted above, some of the notes were outside the range covered by our training set (and thus could never be reported by our classifier). While this highlights a strength of model-based transcription in comparison with our example-based classifier (since they directly generalize across pitch), there is a natural compromise possible: by resampling our training audio by factors corresponding to plus or minus a few semitones, and using these 'transposed' versions as additional training data (with the ground-truth labels suitably offset), we can 'teach' our classifier that a simple spectral shift of a single spectrum corresponds to a note change, just as is implicit in model-based systems.

By the same token, we may ask what the trained classifier might learn beyond what a model-based system already knows, as it were. By training on all examples of a particular note *in situ*, the classifier transcriber can observe not only the prominent harmonics in the spectrum (or autocorrelation) corresponding to the target pitch, but any statistical regularities in the accompaniment (such as the most likely accompanying notes). Looking at figure 4, for example at the final note of the top pane, we see that although the actual note was a B, the classifier is confusing it with a G – presumably because there were a number of training instances where a melody G included strong harmonics from an accompanying B, which could in some circumstances be a useful regularity to have learned. Indeed, noting that our current classifiers seem to saturate with only a few seconds of training material, we might consider a way to train a more complex classifier by including richer conditioning inputs; the inferred 'mode' hidden state suggested above is an obvious contender.

The full melody competition involved not only deciding the note of frames where the main melody was deemed to be active, but also discriminating between melody and non-melody (accompaniment) frames, on the face of it a very difficult problem. It is, however, a natural fit for a classifier: once we have our labeled ground truth, we can train a separate classifier (or a new output in our existing classifier) to indicate when background is detected and no melody note should be emitted; different features (including overall energy) and different normalization schemes are appropriate for this decision.

In summary, we have shown that the novel approach to melody transcription in which essentially everything is left to the learning algorithm and no substantial prior knowledge of the structure of musical pitch is hard-coded in, is feasible, competitive, and straightforward to implement. The biggest challenge is obtaining the training data, although in our configuration the amount of data required was not excessive. We stress that this is only the first stage of a more complete music transcription system, one that we aim to build at each level on the principle of learning from examples of music rather than through coded-in expert knowledge.

ACKNOWLEDGEMENTS

Many thanks to Emilia Gómez, Beesuan Ong, and Sebastian Streich for organizing the 2004 ISMIR Melody Contest, and for making the results available. 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. Eggink and G. J. Brown. Extracting melody lines from complex audio. In *Proc. Int. Conf. on Music Info. Retrieval ISMIR-03*, pages 84–91, 2004.

E. Gomez, B. Ong, and S. Streich. Ismir 2004 melody extraction competition contest definition page, 2004. `http://ismir2004.ismir.net/melody_contest/results.html`.

M. Goto. A predominant-f0 estimation method for polyphonic musical audio signals. In *18th International Congress on Acoustics*, pages 1085–1088, 2004.

Y. Li and D. Wang. Detecting pitch of singing voice in polyphonic audio. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages III.17–21, 2005.

M. Marolt. On finding melodic lines in audio recordings. In *DAFx*, 2004.

R. P. Paiva, T. Mendes, and A. Cardoso. A methodology for detection of melody in polyphonic music signals. In *116th AES Convention*, 2004.

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.

M. P. Ryynänen and A. P. Klapuri. Modelling of note events for singing transcription. In *Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, Jeju, Korea, October 2004. URL `http://www.cs.tut.fi/~mryynane/mryynane_final_sapa04.pdf`.

R. J. Turetsky and D. P. 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.