Mach Learn (2006) 65:439–456
DOI 10.1007/s10994-006-8373-9

Classification-based melody transcription

Daniel P.W. Ellis · Graham E. Poliner

Received: 24 September 2005 / Revised: 16 February 2006 / Accepted: 20 March 2006 / Published online: 8 May 2006

Abstract The melody of a musical piece—informally, the part you would hum along with—is a useful and compact summary of a full audio recording. The extraction of melodic content has practical applications ranging from content-based audio retrieval to the analysis of musical structure. Whereas previous systems generate transcriptions 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 ADC 2004 Melody Competition evaluation set, and we show that a simple frame-level note classifier, temporally smoothed by post processing with a hidden Markov model, produces results comparable to state of the art model-based transcription systems.

Keywords Melody transcription · Audio · Music · Support vector machine · Hidden markov model · Imbalanced data sets · Multiway 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 (e.g., Birmingham et al., 2001). However, to deploy large-scale music organization and retrieval systems based on melodic content, we need mechanisms to automatically extract the melody from recorded audio. Such transcription would also be valuable in musicological analysis as well as numerous potential signal transformation applications.

Editors: Gerhard Widmer

D.P.W. Ellis (✉) · G.E. Poliner
LabROSA, Department of Electrical Engineering, Columbia University, New York NY 10027, USA
e-mail: dpwe@ee.columbia.edu
e-mail: graham@ee.columbia.edu

Although automatic transcription of polyphonic music recordings (multiple instruments playing together) has long been a research goal, it has remained elusive. In Goto and Hayamizu (1999), the authors proposed a reduced problem of transcribing polyphonic audio into a single melody line (along with a low-frequency bass line) as a useful but tractable analysis of audio. Since then, a significant amount of research has taken place in the area of predominant melody detection, including Goto (2004), Eggink and Brown (2004), Marolt (2004), Paiva et al. (2005), Li and Wang (2005).

These methods, however, all rely on a core analysis that assumes a specific audio structure, specifically that musical pitch is produced by periodicity at a particular fundamental frequency in the audio signal. For instance, the system of Goto and Hayamizu (1999) extracts instantaneous frequencies from spectral peaks in the short-time analysis frames of the music audio; Expectation-Maximization is used to find the most likely fundamental frequency value for a parametric model that can accommodate different spectra, but constrains all the frequency peaks to be close to integer multiples (harmonics) of the fundamental, which is then taken as the predominant pitch. This assumption that pitch arises from harmonic components is strongly grounded in musical acoustics, but it is not necessary for transcription. In many fields (such as automatic speech recognition) classifiers for particular events are built without any prior, explicit knowledge of how they are represented in the features.

In this paper, we pursue this insight by investigating a machine learning approach to automatic melody transcription. We propose a system that infers the correct melody label based only on training with labeled examples. Our algorithm identifies a single dominant pitch, assumed to be the melody note, for all time frames in which the melody is judged to be sounding. The note is identified via a Support Vector Machine classifier trained directly from audio feature data, and the overall melody sequence is smoothed via a hidden Markov model, to reflect the temporal consistency of actual melodies. This learning-based approach to extracting pitches stands in stark contrast to previous approaches that all incorporate prior assumptions of harmonic or periodic structure in the acoustic waveform. The main contribution of this paper is to demonstrate the feasibility of this approach, as well as describing our solutions to various consequent issues such as selecting and preparing appropriate training data.

Melody, or predominant pitch, extraction is an attractive task for the reasons outlined above, but melody does not have a rigorous definition. We are principally interested in popular music which usually involves a lead vocal part—the singing in a song. Generally, the lead vocal will carry a melody, and listeners will recognize a piece they know based only on that melody alone. Similarly in jazz and popular instrumental music, there is frequently a particular instrument playing the lead. By this definition, there is not always a melody present—there are likely to be gaps between melody phrases where only accompaniment will be playing, and where the appropriate output of a melody transcriber would be nothing (unvoiced); we will refer to this problem as voicing detection, to distinguish it from the pitch detection problem for the frames containing melody. There will always be ambiguous cases—music in which several instruments contend to be considered the 'lead', or notes which might be considered melody or perhaps are just accompaniment. Of course, a classification system simply generalizes from the training data, so to some extent we can crystallize our definition of melody as we create our training ground truth. On the whole, we have tended to stay away from borderline cases when selecting training data (and the standard test data we use has been chosen with similar criteria), but this ambiguity remains a lurking problem. For the purposes of this paper, however, we evaluate melody transcription as the ability to label music audio with either a pitch—a MIDI note number, or integer corresponding to one semitone (one key on the piano)—or an "unvoiced" label. Resolving pitch below the

level of musical semitones has limited musical relevance and would fit less cleanly into the classification framework.

The remainder of this paper is structured as follows: Training data is the single greatest influence on any classifier, but since this approach to transcription is unprecedented, we were obliged to prepare our own data, as described in the next section. Section 3 then describes the acoustic features and normalization we use for classification. In Section 4 we describe and compare different frame-level pitch classifiers we tried based on Support Vector Machines (SVMs), and Section 5 describes the separate problem of distinguishing voiced (melody) and unvoiced (accompaniment) frames. Section 6 describes the addition of temporal constraints with hidden Markov models (HMMs) by a couple of approaches. Finally, Section 7 discusses the results, and presents some ideas for future developments and applications of the approach.

2. Audio data

Supervised training of a classifier requires a corpus of labeled feature vectors. In general, greater quantities 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 availability of digital scores aligned to real recordings is very limited, there are some other possible sources for suitable data. We investigated using multi-track recordings and MIDI audio files for training; for evaluation, we were able to use some recently-developed standard test sets.

2.1. Multi-track recordings

Popular music recordings are typically created by layering a number of independently-recorded audio tracks. In some cases, artists (or their record companies) make available separate vocal and instrumental tracks as part of a CD or 12" vinyl single release. The 'acapella' vocal recordings can be used to create ground truth for the melody in the full ensemble music, since solo voice can usually be tracked at high accuracy by standard pitch tracking systems (Talkin, 1995; de Cheveigne & Kawahara, 2002). As long as we can identify the temporal alignment between the solo track and the full recording (melody plus accompaniment), we can construct 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 collection of 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 monaural 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 fundamental frequency estimator in WaveSurfer, which is derived from ESPS's get_f0 (Sjölander & Beskow, 2000). Fundamental frequency predictions were calculated at frame intervals of 10 ms and limited to the range 70–1500 Hz.

We used Dynamic Time Warping (DTW) to align the acapella recordings and the full ensemble recordings, 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. Example signals from the training data are illustrated in Fig. 1.

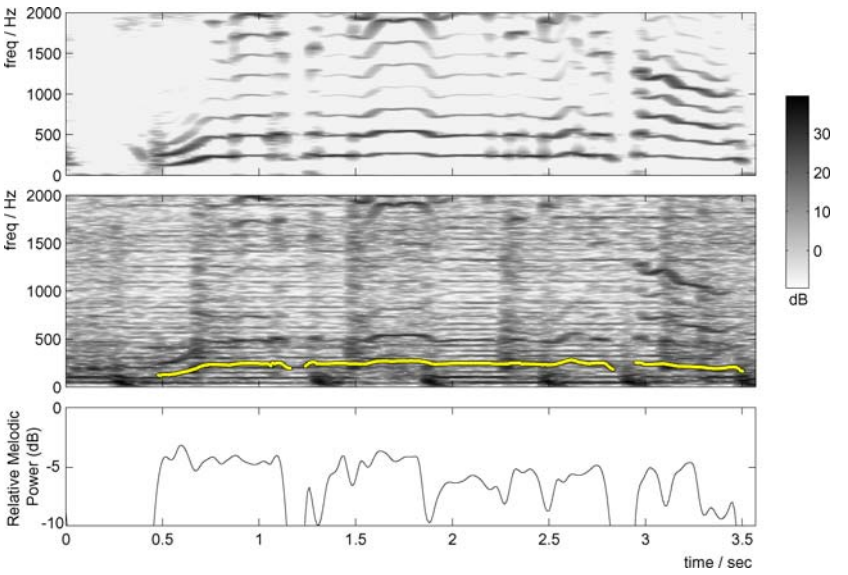

Fig. 1 Examples from training data generation. The fundamental frequency of the isolated melody track (top pane) is estimated and time-aligned to the complete audio mix (center). The fundamental frequency estimates, rounded to the nearest semitone are used as target class labels (overlaid on the spectrogram). The bottom panel shows the power of the melody voice relative to the total power of the mix (in dB); if the mix consisted only of the voice, this would be 0 dB

We ended up with 12 training excerpts of this kind, ranging in duration from 20 s to 48 s. Only the voiced portions were used for training (we did not attempt to include an 'unvoiced' class at this stage), resulting in 226 s of training audio, or 22,600 frames at a 10 ms frame rate.

2.2. MIDI audio

MIDI was created by the manufacturers of electronic musical instruments as a digital representation of the notes, times, and other control information required to synthesize a piece of music. As such, a MIDI file amounts to a digital music score that can easily be converted into an audio rendition. Extensive collections of MIDI files exist consisting of numerous transcriptions from eclectic genres. Our MIDI training data is composed of several frequently downloaded pop songs from www.findmidis.com. The training files were converted from the standard MIDI file format to monaural audio files (.WAV) with a sampling rate of 8 kHz using the MIDI synthesizer in Apple's iTunes. Although completely synthesized (with the lead vocal line often assigned to a wind or brass voice), the resulting audio is quite rich, with a broad range of instrument timbres, and including production effects such as reverberation.

In order to identify 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: Commonly, the lead voice in pop MIDI files is stored in 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 the analysis frames of the synthesized audio.

We used five MIDI excerpts for training, each around 30 s in length. After removing the unvoiced frames, this left 125 s of training audio (12,500 frames).

2.3. Resampled audio

In the case when the availability of a representative training set is limited, the quantity and diversity of musical 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 expanded training set consisted of all transpositions pooled together. The ground truth labels were shifted accordingly and linearly interpolated to maintain time alignment (because higher-pitched transpositions also acquire a faster tempo). Using this approach, we created a smoother distribution of the training labels and reduced bias toward the specific pitches present in the training set. Our classification approach relies on learning separate decision boundaries for each individual melody note with no direct mechanism to ensure consistency between similar note classes (e.g. C4 and C#4), or to improve the generalization of one note-class by analogy with its neighbors in pitch. Using a transposition-expanded training restores some of the advantages we might expect from a more complex scheme for tying the parameters of pitchwise-adjacent notes: although the parameters for each classifier are separate, classifiers for notes that are similar in pitch have been trained on transpositions of many of the same original data frames. Resampling expanded our total training pool by a factor of 13 to around 456,000 frames.

2.4. Validation and test sets

Research progress benefits when a community agrees a consistent definition of their problem of interest, then goes on to define and assemble standard tests and data sets. Recently, the Music Information Retrieval (MIR) community has begun formal evaluations, starting with the Audio Description Contest at the 2004 International Symposium on Music Information Retrieval (ISMIR/ADC 2004) (Gomez et al., 2004). Its successor is the Music Information Retrieval Evaluation eXchange (MIREX, 2005) (Downie et al., 2005). Each consisted of numerous MIR-related evaluations including predominant melody extraction. The ADC 2004 test set is composed of 20 excerpts, four from each of five styles, each lasting 10–25 s, for a total of 366 s of test audio. The MIREX 2005 melody evaluation created a new test set consisting of 25 excerpts ranging in length from 10–40 s, giving 536 s total. Both sets include excerpts where the dominant melody is played on a variety of musical instruments including the human voice; the MIREX set has more of a bias to pop music. In this paper, all of our experiments are conducted using the ADC 2004 data as a development set, since the MIREX set is reserved for annual evaluations

3. Acoustic features

Our acoustic representation is based on the ubiquitous and well-known spectrogram, which converts a sound waveform into a distribution of energy over time and frequency, very often displayed as a pseudocolor or grayscale image like the middle pane in Fig. 1; the base features for each time-frame can be considered as vertical slices through such an image. Specifically, the original music recordings (melody plus accompaniment) are combined into a single (mono) channel and downsampled to 8 kHz. We apply the short-time Fourier transform

(STFT), using $N = 1024$ point transforms (i.e. 128 ms), an N-point Hanning window, and a 944 point overlap of adjacent windows (for a 10 ms hop between successive frames). Only the coefficients corresponding to frequencies below 2 kHz (i.e. the first 256 bins) were used in the feature vector.

We compared different feature preprocessing schemes by measuring their influence on a baseline classifier. Our baseline pitch classifier is an all-versus-all (AVA) algorithm for multiclass classification using SVMs trained by Sequential Minimal Optimization (Platt, 1998), as implemented in the Weka toolkit (Witten & Frank, 2000). In this scheme, a majority vote is taken from the output of $(N^2 - N)/2$ discriminant functions, comparing every possible pair of classes. For computational reasons, we were restricted to a linear kernel. Each audio frame is represented by a 256-element input vector, with $N = 60$ classes corresponding to five octaves of semitones from G2 to F#7. In order to classify the dominant melodic pitch for each frame, we assume the melody note at a given instant to be solely dependent on the normalized frequency data below 2 kHz. For these results, we further assume each frame to be independent of all other frames. More details and experiments concerning the classifier will be presented in Section 4.

Separate classifiers were trained using six different feature normalizations. Of these, three use the STFT, and three are based on (pseudo)autocorrelation. 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 magnitudes of the bins are normalized by subtracting the mean and dividing by the standard deviation calculated in a 71-point sliding frequency window; there is no normalization along time. The goal is to remove some of the influence due to different instrument timbres and contexts in training 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 took the inverse Fourier transform (IFT) of the magnitude of the STFT, i.e. the autocorrelation of the original windowed waveform. 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 separated into 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 liftering (scaling the higher-order cepstra by an exponential weight). Scaling of individual feature dimensions can make a difference to SVM classifiers, depending on the kernel function used.

Table 1 compares the accuracy, on a held-out test set, 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

Table 1 Effect of normalization: Frame accuracy percentages on the ADC 2004 held-out test set for each of the normalization schemes considered, trained on either multi-track audio alone, MIDI syntheses alone, or both data sets combined. (Training sets are roughly balanced in size, so results are not directly comparable to others in the paper)

	Training data		
Normalization	Multi-track	MIDI	Both
STFT	56.4	50.5	62.5
71-pt norm	54.2	46.1	**62.7**
Cube root	53.3	51.2	62.4
Autocorr	55.8	45.2	62.4
Cepstrum	49.3	45.2	54.6
LiftCeps	55.8	45.3	62.3

Table 2 Impact of resampling the training data: Frame accuracy percentages on the ADC 2004 set for systems trained on the entire training set, either without any resampling transposition, or including transpositions out to ±6 semitones (500 frames per transposed excerpt, 17 excerpts, 1 or 13 transpositions)

Training set	# Training frames	Frame acc %
No resampling	8,500	60.2
With resampling	110,500	**67.7**

accuracy results are for the ADC 2004 melody evaluation set and correspond to melodic pitch transcription to the nearest semitone.

The most obvious result in Table 1 is that all the features, with the exception of Cepstrum, perform much the same, with a slight edge for the across-frequency local normalization. This is perhaps not surprising since all features contain largely equivalent information, but it also raises the question as to how effective our normalization (and hence the system generalization) has been. 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 forms 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 9% absolute improvement, seen for the best-performing 71-point normalized features. (We will discuss significance levels for these results in the next section.)

Table 2 shows the impact of including the replicas of the training set transposed through resampling over ±6 semitones. Resampling achieves a substantial improvement of 7.5% absolute in frame accuracy, underlining the value of broadening the range of data seen for each individual note.

4. Pitch classification

In the previous section, we showed that classification accuracy seems to depend more strongly on training data diversity than on feature normalization. It may be that the SVM classifier we used is better able to generalize than our explicit feature normalization. In this section, we examine the effects of different classifier types on classification accuracy, as well as looking at the influence of the total amount of training data used.

4.1. *N*-way all-versus-all SVM classification

Our baseline classifier is the AVA-SVM described in Section 3 above. Given the large amount of training data we were using (over 10^5 frames), we chose a linear kernel, which requires training time on the order of the number of feature dimensions cubed for each of the $O(N^2)$ discriminant functions. More complex kernels (such as Radial Basis Functions, which require training time on the order of the number of *instances* cubed) were computationally infeasible for our large training set. Recall that labels are assigned independently to each time frame at this stage of processing.

Our first classification experiment was to determine the number of training instances to include from each audio excerpt. The number of training instances selected from each song

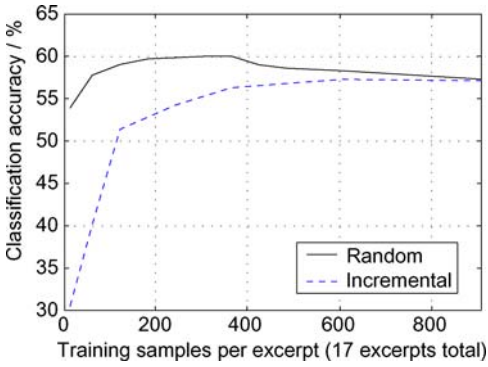

Fig. 2 Variation of classification accuracy with number of training frames per excerpt. Incremental sampling takes frames from the beginning of the excerpt; random sampling takes them from anywhere. Training set does not include resampled (transposed) data

was varied using both incremental sampling (taking a limited number of frames from the beginning of each excerpt) and random sampling (picking the frames from anywhere in the excerpt), as displayed in Fig. 2. Randomly sampling feature vectors to train on approaches an asymptote much more rapidly than adding the data in chronological order. Random sampling also appears to exhibit symptoms of overtraining.

The observation that random sampling achieves peak accuracy with only about 400 samples per excerpt (out of a total of around 3000 for a 30 s excerpt with 10 ms hops) can be explained by both signal processing and musicological considerations. Firstly, adjacent analysis frames are highly overlapped, sharing 118 ms out of a 128 ms window, and thus their feature values will be very highly correlated (10 ms is an unnecessarily fine time resolution to generate training frames, but is the standard used in the evaluation). From a musicological point of view, musical notes typically maintain approximately constant spectral structure over hundreds of milliseconds; a note should maintain a steady pitch for some significant fraction of a beat to be perceived as well-tuned. If we assume there are on average 2 notes per second (i.e. around 120 bpm) in our pop-based training data, then we expect to see approximately 60 melodic note events per 30 s excerpt. Each note may contribute a few usefully different frames to tuning variation such as vibrato and variations in accompaniment. Thus we expect many clusters of largely redundant frames in our training data, and random sampling down to 10% (or closer to one frame every 100 ms) seems reasonable.

This also gives us a perspective on how to judge the significance of differences in these results. The ADC 2004 test set consists of 366 s, or 36,600 frames using the standard 10 ms hop. A simple binomial significance test can compare classifiers by estimating the likelihood that random sets of *independent* trials could give the observed differences in empirical error rates from an equal underlying probability of error. Since the standard error of such an observation falls as $1/\sqrt{N}$ for N trials, the significance interval depends directly on the number of trials. However, the arguments and observations above show that the 10 ms frames are anything but independent; to obtain something closer to independent trials, we should test on frames no less than 100 ms apart, and 200 ms sampling (5 frames per second) would be a safer choice. This corresponds to only 1,830 independent trials in the test set; a one-sided binomial significance test suggests that differences in frame accuracies on this test of less than 2.5% are not statistically significant at the accuracies reported in this paper.

A second experiment examined the incremental gain from adding novel training excerpts. Figure 3 shows how classification accuracy increased as increasing numbers of excerpts, from 1 to 16, were used for training. In this case, adding an excerpt consisted of adding 500 randomly-selected frames from each of the 13 resampled transpositions described in

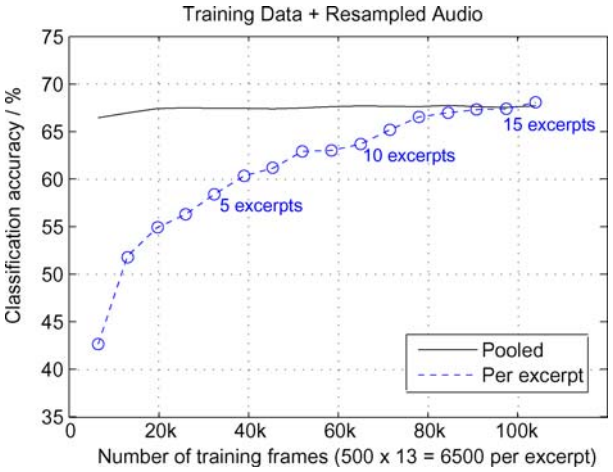

Fig. 3 Variation of classification accuracy with the total number of excerpts included, compared to sampling the same total number of frames from all excerpts pooled. This data set includes 13 resampled versions of each excerpt, with 500 frames randomly sampled from each transposition

Section 2, or 6,500 frames per excerpt. Thus, the largest classifier is trained on 104 k frames, compared to around 15 k frames for the largest classifier in Fig. 2. The solid curve shows the result of training the same number of frames randomly drawn from the pool of the entire training set; again, we notice that the system appears to reach an asymptotes by 20 k total frames, or fewer than 100 frames per transposed excerpt. We believe, however, that the level of this asymptote is determined by the total number of excerpts; if we had more novel training data to include, we believe that the "per excerpt" trace will continue to climb upwards. We return to this point in the discussion.

In Fig. 4, the pitched frame transcription success rates are displayed for the SVM classifier trained using the resampled audio. An important weakness of the classifier-based approach is

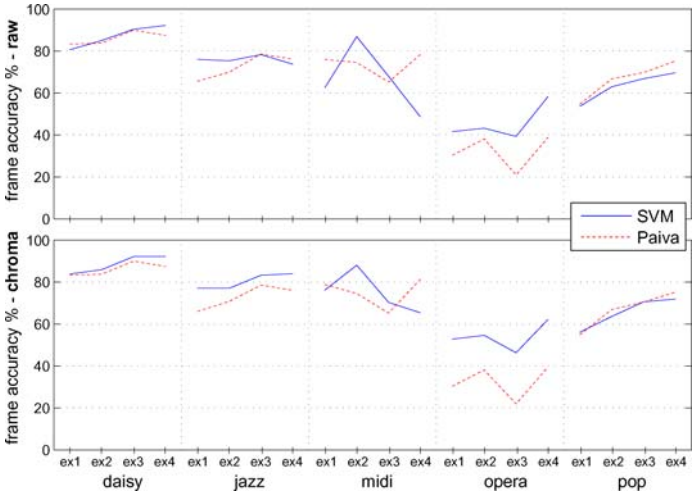

Fig. 4 Variation in voiced transcription frame accuracy across the 20 excerpts of the ADC 2004 evaluation set (4 examples from each of 5 genre categories, as shown). Solid line shows the classification-based transcriber; dashed line shows the results of the best-performing system from the 2004 evaluation. Top pane is raw pitch accuracy; bottom pane folds all results to a single octave of 12 chroma bins, to ignore octave errors

that any classifier will perform unpredictably on test data that does not resemble the training data. While model-based approaches have no problem in principle with transcribing rare, extreme pitches as long as they conform to the explicit model, by deliberately ignoring our expert knowledge of the relationship between spectra and notes our system is unable to generalize from the notes it has seen to different pitches. For example, the highest pitch values for the female opera samples in the ADC 2004 test set exceed the maximum pitch in all our training data. In addition, the ADC 2004 set contains stylistic genre differences (such as opera) that do not match our pop music corpora. That said, many of the pitch errors turn out to be the right pitch chroma class but at the wrong octave (e.g. F6 instead of F7). When scoring is done on chroma class alone—i.e. using only the 12 classes A, A#, ... G#, and ignoring the octave numbers—the overall frame accuracy on the ADC 2004 test set improves from 67.7% to 72.7%.

4.2. Multiple one-versus-all SVM classifiers

In addition to the N-way melody classification, we trained 52 binary, one-versus-all (OVA) SVM classifiers representing each of the notes present in the resampled training set. We took the distance-to-classifier-boundary hyperplane margins as a proxy for a log-posterior probability for each of these classes; pseudo-posteriors (up to an arbitrary scaling power) were obtained from the distances by fitting a logistic model. Transcription is achieved by choosing the most probable class at each time frame. While OVA approaches are seen as less sophisticated, Rifkin and Klautau (2004) present evidence that they can match the performance of more complex multiway classification schemes. Figure 5 shows an example 'posteriorgram' (time-versus-class image showing the posteriors of each class at each time step) for a pop excerpt, with the ground truth labels overlaid.

Since the number of classifiers required for this task is $O(N)$ (unlike the $O(N^2)$ classifiers required for the AVA approach) it becomes computationally feasible to experiment with additional feature kernels. Table 3 displays the best result classification rates for each of the SVM classifiers. Both OVA classifiers perform marginally better than the pairwise classifier, with the slight edge going to the OVA SVM using an RBF kernel.

Table 3 Frame Accuracy for multiway classifiers based on all-versus-all (AVA) and one-versus-all (OVA) structures. Because there are fewer classifiers in an OVA structure, it was practical to try an RBF kernel. Accuracies are given for both raw pitch transcription, and chroma transcription (which ignores octave errors)

Classifier	Kernel	Pitch	Chroma
AVA SVM	Linear	67.7	72.7
OVA SVM	Linear	69.5	74.0
OVA SVM	RBF	**70.7**	**74.9**

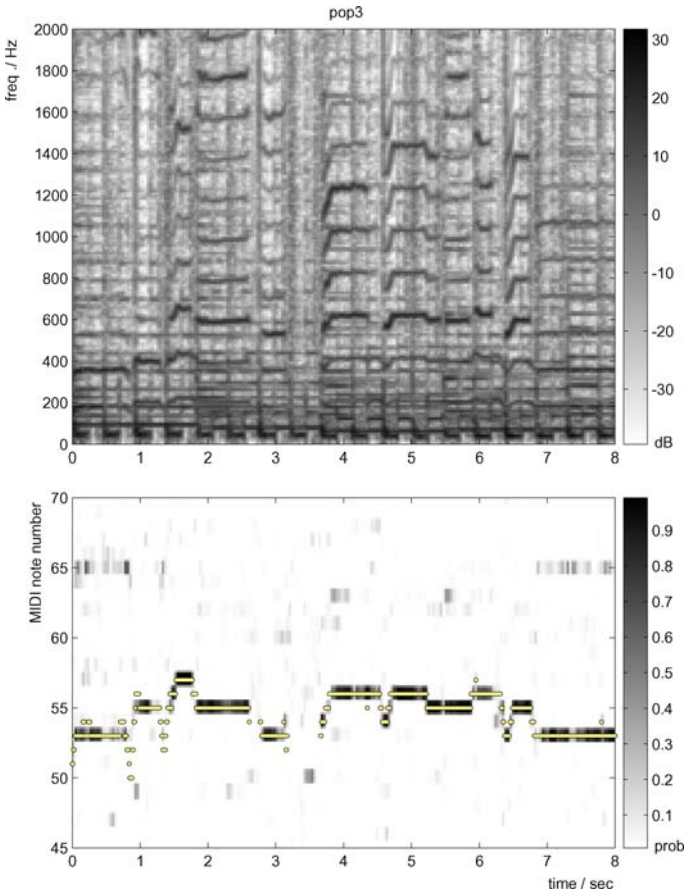

Fig. 5 Spectrogram and posteriorgram (pitch probabilities as s function of time) for the first 8 s of pop music excerpt "pop3" from the ADC 2004 test set. The ground-truth labels, plotted on top of the posteriorgram, closely track the mode of the posteriors for the most part. However, this memoryless classifier also regularly makes hare-brained errors that can be corrected through HMM smoothing

5. Voiced frame classification

Complete melody transcription involves not only deciding the note of frames where the main melody is active, but also discriminating between melody and non-melody (accompaniment) frames. In this section, we briefly describe two approaches for classifying instants as voiced (dominant melody present) or unvoiced (no melody present).

In the first approach, voicing detection is performed by simple energy thresholding. Spectral energy in the range $200 < f < 1800$ Hz is summed for every 10 ms frame. Each value is normalized by the median energy in that band for the given excerpt, and instants are classified as voiced or unvoiced via a global threshold, tuned on development data. Since the melody instrument is usually given a high level in the final musical mix, this approach is quite successful (particularly after we have filtered out the low- frequency energy of bass and drums). In keeping with our classifier-based approach, we also tried a second mechanism of a binary SVM classifier based on the normalized magnitude of the STFT, which could potentially learn

Table 4 Voicing detection performance. "Voicing Det" is the proportion of voiced frames correctly labeled; "False Alarms" is the proportion of unvoiced frames incorrectly labeled, so labeling all frames as voiced scores 100% on both counts, the first row. d' gives a threshold-independent measure of the separation between two Gaussian distributions that would exhibit this performance. "Vx Frame Acc" is the proportion of all frames given the correct voicing label

Classifier	Voicing Det	False Alarms	d'	Vx Frame Acc
All Voiced	100%	100%	0	85.6%
Energy Threshold	88.0%	32.3%	**1.63**	78.4%
Linear SVM	76.1%	46.4%	0.80	73.0%
RBF SVM	82.6%	48.1%	0.99	78.3%

particular spectral cues to melody presence. We tried both linear and RBF kernels for this SVM. The voiced melody classification statistics are displayed in Table 4, which shows that the simple energy threshold provides better results; however none of the classifiers achieves a higher frame accuracy than simply labeling all frames as voiced. Because the test data is more than 85% voiced, any classifier that attempts to identify unvoiced frames risks making more mislabelings than unvoiced frames correctly detected. We also report performance in terms of d', a statistic commonly used in evaluating detection systems in an effort to discount the influence of trading false alarms for false rejects. d' is the separation, in units of standard deviation, between the means of two unit-variance one-dimensional Gaussians that would give the same balance of false alarms to false rejects if they were the underlying distributions of the two classes; a d' of zero indicates indistinguishable distributions, and larger is better.

While our voicing detection scheme is simple and not particularly accurate, it is not the main focus of the current work. The energy threshold gave us a way to identify nearly 90% of the melody-containing frames, without resorting to the crude choice of simply treating every frame as voiced. However, more sophisticated approaches to learning classifiers for tasks with highly-skewed priors offer a promising future direction (Chawla et al., 2004).

6. Hidden Markov model post processing

The posteriorgram in Fig. 5 clearly illustrates both the strengths and weaknesses of the classification approach to melody transcription. The success of the approach in estimating the correct melody pitch from audio data is clear in the majority of frames. However, the result also displays the obvious fault of the approach of classifying each frame independently of its neighbors: the inherent temporal structure of music is not exploited. In this section, we attempt to incorporate the sequential structure that may be inferred from musical signals. We use hidden Markov models (HMMs) as one of the simplest yet most flexible approaches to capturing and using temporal constraints.

6.1. HMM state dynamics

Similarly to our data driven approach to classification, we learn temporal structure directly from the training data. Our HMM states correspond directly to the current melody pitch, thus the state dynamics (transition matrix and state priors) can be estimated from our 'directly observed' state sequences—the ground-truth transcriptions of the training set. The note class prior probabilities, generated by counting all frame-based instances from the resampled

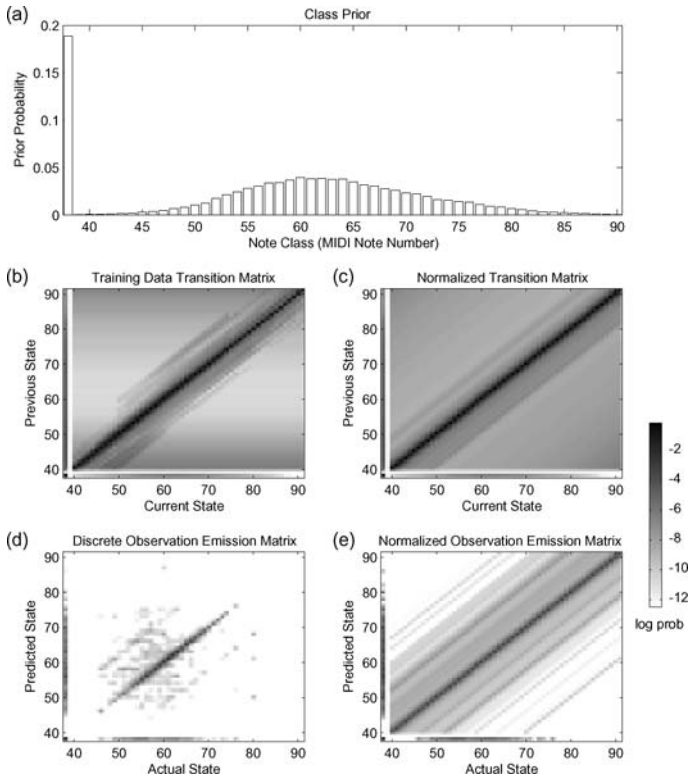

Fig. 6 Hidden Markov parameters learned from the ground truth and confusion matrix data. Top (pane (a)): class priors. Middle: state transition matrix, raw (pane (b)), and regularized across all notes (pane (c)). Bottom: Observation likelihood matrix for the discrete-label smoothing HMM, raw (pane (d)) and regularized across notes (pane (e))

training data, and the note class transition matrix, generated by observing all note-to-note transitions, are displayed in Fig. 6(a) and (b) respectively.

Note that although some bias has been removed in the note priors by symmetrically re-sampling the training data, the sparse nature of a transition matrix learned from a limited training set is likely to generalize poorly to novel data. In an attempt to mitigate this lack of generalization, each element of the transition matrix is replaced by the mean of the corresponding diagonal. This is equivalent to assuming that the probability of making a transition between two pitches depends only on the interval between them (in semitones), not on their absolute frequency. This normalized approach is close to implementing a relative transition vector, and the resulting normalized state transition matrix is displayed in Fig. 6(c).

6.2. Smoothing discrete classifier outputs

We can use an HMM to apply temporal smoothing even if we only consider the labels assigned by the frame-level classifier at each stage and entirely ignore any information on the relatively likelihood of other labels that might have been available prior to the final hard decision being made by the classifier. If the model state at time t is given by q_t, and the classifier output label is c_t, then the HMM will achieve temporal smoothing by finding the

most likely (Viterbi) state sequence i.e. maximizing

$$\prod_t p(c_t \mid q_t) p(q_t \mid q_{t-1}) \qquad (1)$$

$p(q_t \mid q_{t-1})$ is the transition matrix estimated from ground-truth melody labels in the previous subsection, but we still need $p(c_t \mid q_t)$, the probability of seeing a particular classifier label c_t given a true pitch state q_t. We estimate this from the confusion matrix of classifier frames—i.e. counts normalized to give $p(c_t, q_t)$—from some development corpus. In this case, we reused the training set, which might lead to an overoptimistic belief about how well c_t will reflect q_t, but was our only practical option. Figure 6(d) shows the raw confusion matrix normalized by columns (q_t) to give the required conditionals; pane (e) shows this data again regularized by setting all values in a diagonal to be equal, except for the zero (unvoiced) state, which is smoothed a little by a moving average, but otherwise retained. From the confusion (observation) matrix, we can see that the most frequently confused classifications are between members of the same chroma (i.e. separated by one or more octaves in pitch) and between notes with adjacent pitches (separated by one semitone).

For the total transcription problem (dominant melody transcription plus voicing detection), our baseline (memoryless) transcription is simply the pitch classifier output gated by the binary energy threshold. If at each instant we use the corresponding column of the observation matrix in our Viterbi decoder dynamic-programming local-cost matrix, we can derive a smoothed state sequence that removes short, spurious excursions of the raw pitch labels. Despite the paucity of information obtained from the classifier, Table 5 shows that this approach results in a small but robust improvement of 0.9% absolute in total frame accuracy.

Table 5 Melody transcription frame accuracy percentages for different systems with and without HMM smoothing. "Voicing Acc" is the proportion of frames whose voicing state is correctly labeled. "Pitch Acc" is the proportion of pitched frames assigned the correct pitch label. "Total Acc" is the proportion of all frames assigned both the correct voicing label, and, for voiced frames, the correct pitch label. All results are based on the all-versus-all SVM classifier using an RBF kernel. The "Memoryless" classifier simply takes the most likely label from the frame classifier after gating by the voicing detector; "MemlessAllVx" ignores the voicing detection and reports a pitch for all frames (to maximize pitch accuracy). "Discrete" applies HMM smoothing to this label sequence without additional information, "Posteriors" uses the pseudo-posteriors from the OVA SVM to generate observation likelihoods in the HMM, and "PostAllVx" is the same except the unvoiced state is excluded (by setting its frame posterior to zero)

Classifier	Voicing Acc	Pitch Acc	Total Acc
Memoryless	85.1	71.2	70.7
MemlessAllVx	86.1	76.8	66.1
Discrete	83.6	71.9	71.6
Posteriors	86.2	74.5	**73.2**
PostAllVx	86.1	**79.4**	68.3

6.3. Exploiting classifier posteriors

We constructed the OVA classifier to give us an approximation to log-posteriors for each pitch class, and we can use this detailed information to improve our HMM decoding. Rather than guessing the local likelihood of a particular note given the single best note, we can look directly at the likelihood of each note according to the classifiers. Thus, if the acoustic data at each time is x_t, we may regard our OVA classifier as giving us estimates of

$$p(q_t \mid x_t) \propto p(x_t \mid q_t) p(q_t) \qquad (2)$$

i.e. the posterior probabilities of each HMM state given the local acoustic features. Thus, by dividing each (pseudo)posterior by the prior of that note, we get scaled likelihoods that can be employed directly in the Viterbi search for the solution of Eq. 1. The unvoiced state needs special treatment, since it is not considered by the main classifier. We tried several approaches, including decoding the pitch HMM with the unvoiced state excluded (setting its observation likelihood to zero), then applying voicing decisions from a separate voicing HMM, or setting the observation posterior of the unvoiced state to $1/2 \pm \alpha$, where α was tuned on the development (training set), which gave significantly better results.

As shown in Table 5, using this information achieves another robust improvement of 1.6% absolute in total frame accuracy over using only the 1-best classification information. More impressively, the accuracy on the pitched frames jumps 3.3% absolute compared to the memoryless case, since knowing the per-note posteriors helps the HMM to avoid very unlikely notes when it decides to stray from the one-best label assigned by the classifier. If we focus only on pitch accuracy (i.e. exclude from scoring the frames whose ground truth is unvoiced), we can maximize pitch accuracy with a posterior-based HMM decode that excludes the unvoiced state, achieving a pitch accuracy of 79.4%, or 2.6% better than the comparable unsmoothed case.

7. Discussion and conclusions

We have described techniques for recovering a sequence of melodic pitch labels from a complex audio signal. Our approach consists of two steps: first, a discriminatively-trained local pitch classifier operating on single time frames, followed by a hidden Markov model trained to apply some of the temporal constraints observed in real melody label sequences to the resulting signal. This combination is ad-hoc: if the task is to exploit local features subject to sequential constraints, then we should look for an approach that optimizes the entire problem at once. Recently, Taskar et al. (2003) have suggested an approach to using the advantages of classifier-margin-maximization from SVMs in a Markov model, but we expect that solving the entire optimization problem would be impractical for the quantities of training data we are using. The important question is whether there are significant differences in how local features would be processed depending on the context in such a scheme, or whether the problem can in fact be separated into the two stages we have implemented without significant impact on performance. Looking at the intermediate posterior representation illustrated in Fig. 5, we suspect that this is an adequate representation of the acoustic information for the sequential modeling to be operating on. We feel that the most promising areas for improvement are improving the frame-level accuracy of this front end and more sophisticated post-processing (including improved voicing detection), and not a tighter integration between the two stages, which seem comfortably separate from our perspective.

There are several directions in which we plan to continue this work. Although this paper has considered the problem of melody extraction i.e. assigning at most one pitch to each time step, the OVA classifiers can be run and examined independently. With a different kind of decoding scheme running over the output posteriors, it may be possible to report more than one note for each time frame i.e. polyphonic transcription. Alternatively, there may be a direct classification path to polyphonic transcription i.e. training classifiers to recognize entire chords, although this raises serious combinatoric problems.

We believe the local harmonic structure carried by a chord could provide significant additional information and constraints for melody transcription. In conventional western music, there is an underlying harmonic progression of chords, and the melody notes are strongly dependent on the current chord. The chord sequence itself has a strong sequential structure, and is dependent on the overall key of the piece. This hierarchy of hidden states is a natural fit to hidden Markov modeling, based either on separate chord-related features or on the note observations alone, and a system that explicitly represents the current chord can have a more powerful, chord-dependent transition matrix for the pitch state.

We have shown that a pure classification approach to melody transcription is viable, and can be successful even when based on a modest amount of training data. In the MIREX 2005 evaluation, we entered the system denoted "AVA SVM" in this paper, without any HMM smoothing, which ranked third of ten entries; by switching to the OVA architecture, the RBF kernel, and the posterior-based HMM smoothing, we achieve a 5.5% absolute improvement in overall frame accuracy (from 67.7% to 73.2%), and believe this would carry across to the MIREX test. (The MIREX 2005 results are visible at `http://www.music-ir.org/evaluation/mirex-results/audio-melody/`.)

The natural question is: what is the upper limit on the performance achievable by this approach? One attraction of data-driven approaches is that further improvements can predictably be obtained by gathering additional training data. Figure 3 is the important example here: if we create ground truth for another 15 excerpts, how high can our accuracy grow? The slope of accuracy vs. excerpt count looks quite promising, but it may be reaching an asymptote. One of the most striking results of the recent MIREX evaluation is that our radically different approach to transcription scored much the same as the traditional model-based approaches; the performance spread across all systems on raw pitch accuracy was within 10%, from 58% to 68%.

This suggests that the limiting factors are not so much in the algorithms, but in the data, which might consist of 60–70% of relatively easy-to-transcribe frames with the pitch of the remaining frames very difficult to transcribe for classifier- or model-based approaches alike. If this were the case, we would not expect rapid improvements from our classifier with modest amounts of training data, although there must be some volume of training data that would allow us to learn these harder cases.

Given the expense of constructing additional training data—either in trying to beg multitrack recordings from studios and artists, or in manually correcting noisy first-pass transcripts, we are also curious about the tradeoff between quantity and quality in training data. Raw musical audio is available in essentially unlimited quantities: given an automatic way to extract and label ground-truth, we might be able to construct a positive-feedback training loop, a system that attempts to transcribe the melody (aided perhaps by weak annotation, such as related but unaligned MIDI files), then retrains itself on the most confidently-recognized frames, then repeats. Bootstrap training based on lightly-annotated data of this flavor has proven successful for building speech recognition systems from very large training sets (Lamel et al., 2002).

However, classification-based transcription has a number of other attractive features that make it interesting even if it cannot provide a significant performance improvement over conventional methods. One application that we find particularly appealing is to train the classifier on the accompaniment data *without* the lead melody mixed in, but still using the lead melody transcripts as the target labels. Then we will have a classifier trained to predict the 'appropriate' melody from accompaniment alone. With suitable temporal smoothness constraints, as well as perhaps some random perturbation to avoid boring, conservative melody choices, this could turn into a very interesting kind of automatic music generation or robotic improvisor.

Acknowledgments We thank Gerhard Widmer and two anonymous reviewers for their helpful comments on this paper. 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

Birmingham, W., Dannenberg, R., Wakefield, G., Bartsch, M., Bykowski, D., Mazzoni, D., Meek, C., Mellody, M., & Rand, W. (2001). MUSART: Music retrieval via aural queries. In *Proc. 2nd Annual International Symposium on Music Information Retrieval ISMIR-01* (pp. 73–82). Bloomington, IN.

Chawla, N., Japkowicz, N., & Kolcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explorations*, *6*(1), 1–6. URL `http://portal.acm.org/toc.cfm?id=1007730&type=issue`.

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

Downie, J., West K., Ehmann, A., & Vincent, E. (2005). The 2005 music information retrieval evaluation exchange (MIREX 2005): Preliminary overview. In *Proc. 6th International Symposium on Music Information Retrieval ISMIR* (pp. 320–323). London.

Eggink, J., & Brown, G. J. (2004). Extracting melody lines from complex audio. In *International Symposium on Music Information Retrieval* (pp. 84–91).

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

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

Goto, M., & Hayamizu, S. (1999). A real-time music scene description system: Detecting melody and bass lines in audio signals. In *Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis* (pp. 31–40). Stockholm.

Lamel, L., Gauvain, J.-L., & Adda, G. (2002). Lightly supervised and unsupervised acoustic model training. *Computer, Speech & Language*, *16*(1), 115–129. URL `citeseer.ist.psu.edu/lamel02lightly.html`.

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

Marolt, M. (2004). On finding melodic lines in audio recordings. In *Proc. 7th International Conference on Digital Audio Effects DAFx'04, Naples, Italy*. URL http://lgm.fri.uni-lj.si/∼matic/clanki/dafx04_marolt.pdf

Paiva, R. P., Mendes, T., & Cardoso, A. (2005) On the detection of melody notes in polyphonic audio. In *Proc. 6th International Symposium on Music Information Retrieval ISMIR* (pp. 175–182). London.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods—support vector learning* (pp. 185–208). Cambridge, MA, MIT Press.

Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classication. *Journal of Machine Learning Research*, *5*, 101–141, URL `http://jmlr.csail.mit.edu/papers/volume5/rifkin04a/rifkin04a.pdf`.

Sjölander, K., & Beskow, J. (2000). WaveSurfer—an open source speech tool. In *Proc. Int. Conf. on Spoken Language Processing*.

Talkin, D. (1995). A robust algorithm for pitch tracking (RAPT). In W. B. Kleijn & K. K. Paliwal, (Eds.), *Speech coding and synthesis*, chapter *14* (pp. 495–518). Elsevier, Amsterdam.

Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin markov networks. In *Proc. Neural Information Processing Systems NIPS*, Vancouver, URL `http://www.cs.berkeley.edu/~taskar/pubs/mmmn.ps`.

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

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