A Discriminative Model for Polyphonic Piano 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

June 17, 2006

Abstract

In this paper we present a discriminative model for polyphonic piano transcription. Support Vector Machines trained on spectral features are used to classify frame-level note instances. The classifier outputs are temporally constrained via hidden Markov models, and the proposed system is used to transcribe both synthesized and real piano recordings. A frame-level transcription accuracy of 68% was achieved on a newly generated test set, and direct comparisons to previous approaches are provided.

1 Introduction

Music transcription is the process of creating a musical score (i.e. a symbolic representation) from an audio recording. Although expert musicians are capable of transcribing polyphonic pieces of music, the process is often arduous for complex recordings. As such, the ability to automatically generate transcriptions has numerous practical implications in musicological analysis and may potentially aid in content-based music retrieval tasks.

The transcription problem may be viewed as identifying the notes that have been played in a given time period (i.e. detecting the onsets of each note). Unfortunately, the harmonic series interaction that occurs in polyphonic music significantly obfuscates automated transcription. Moorer [1977] first presented a limited system for duet transcription. Since then, a number of acoustical models for polyphonic transcription have been presented in both the frequency domain [Rossi et al., 1997, Sterian, 1999, Dixon, 2000] and the time domain [Bello et al., 2002].

These methods, however, rely on a core analysis that assumes a specific audio structure; namely that musical pitch is produced by periodicity at a particular fundamental frequency in the audio signal. For instance, the system of Klapuri [2005] estimates multiple fundamental frequencies from spectral peaks using a computational model of the human auditory periphery. Then, discrete hidden Markov models (HMMs) are iteratively applied to extract melody lines from the fundamental frequency estimations [Ryynänen and Klapuri, 2005].

The 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 using the minimum of prior knowledge of how they are represented in the features. Marolt [2004] presented such a classification-based approach to transcription using neural networks, but a filterbank of adaptive oscillators was required in order to reduce erroneous note insertions. Bayesian models have also been proposed for music transcription [Godsill and Davy, 2002, Cemgil et al., 2005, Kashino and Godsill, 2004]; however, these inferential treatments, too, rely on physical prior models of musical sound generation.

In this paper, we pursue the insight that prior knowledge is not strictly necessary for transcription by examining a discriminative model for automatic music transcription. We propose a supervised classification system that infers the correct note labels based only on training with labeled examples. Our algorithm performs polyphonic transcription via a system of Support Vector Machine (SVM) classifiers trained from spectral features. The independent classifications are then temporally smoothed in an HMM post-processing stage. We show that a classification-based system provides significant advantages in both performance and simplicity over acoustic model approaches.

The remainder of this paper is structured as follows: We describe the generation of our training data and acoustic features in Section 2. In Section 3, we present a frame-level SVM system for polyphonic pitch classification. The classifier outputs are temporally smoothed by a note-level HMM as described in Section 4. The proposed system is used to transcribe both synthesized piano and recordings of a real piano, and the results, as well as a comparison to previous approaches, are presented in Section 5. Finally, we provide a discussion of the results and present ideas for future developments in Section 6.

2 Audio Data and Features

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. In this paper, we investigate using synthesized MIDI audio and live piano recordings to generate training, testing, and validation sets.

2.1 Audio Data

MIDI was created by the manufacturers of electronic musical instruments as a digital representation of the notes, timing, and other control information required to synthesize a piece of music. As such, a MIDI file amounts to a

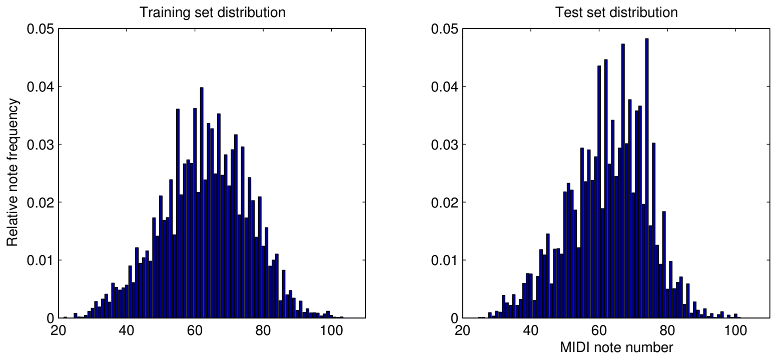

Figure 1: Note distributions for the training and test sets.

digital music score that can be converted into an audio rendition. The MIDI data used in our experiments was collected from the Classical Piano Midi Page, `http://www.piano-midi.de/` . The 130 piece data set was randomly split into 92 training, 25 testing, and 13 validation pieces. Table 5 gives a complete list of the composers and pieces used in the experiments.

The MIDI files were converted from the standard MIDI file format to monaural audio files with a sampling rate of 8 kHz using the synthesizer in Apple's iTunes. In order to identify the corresponding ground truth transcriptions, the MIDI files were parsed into data structures containing the relevant audio information (i.e. tracks, channels numbers, note events, etc). Target labels were determined by sampling the MIDI transcript at the precise times corresponding to the analysis frames of the synthesized audio.

In addition to the synthesized audio, piano recordings were made from a subset of the MIDI files using a Yamaha Disklavier playback grand piano. 20 training files and 10 testing files were randomly selected for recording. The MIDI file performances were recorded as monaural audio files at a sampling rate of 44.1 kHz. Finally, the piano recordings were time-aligned to the MIDI score by identifying the maximum cross-correlation between the recorded audio and the synthesized MIDI audio.

The first minute from each song in the data set was selected for experimentation which provided us with a total of 112 minutes of training audio, 35 minutes of testing audio, and 13 minutes of audio for parameter tuning on the validation set. This amounted to 56497, 16807, and 7058 note instances in the training, testing, and validation sets respectively. The note distributions for the training and test sets are displayed in Figure 1.

2.2 Spectral Features

We applied the short-time Fourier transform to the audio files using $N = 1024$ point Discrete Fourier Transforms (i.e. 128 ms), an N-point Hanning window, and an 80 point advance between adjacent windows (for a 10 ms hop between successive frames). In an attempt to remove some of the influence due to timbral and contextual variation, the magnitudes of the spectral bins were normalized by subtracting the mean and dividing by the standard deviation calculated in a 71-point sliding frequency window. Note that the live piano recordings were down-sampled to 8 kHz using an anti-aliasing filter prior to feature calculation in order to reduce the spectral dimensionality.

Separate one-versus-all (OVA) SVM classifiers were trained on the spectral features for each of the 88 piano keys with the exception of the highest note, MIDI note number 108. For MIDI note numbers 21 to 83 (i.e. the first 63 piano keys), the input feature vector was composed of the 255 coefficients corresponding to frequencies below 2 kHz. For MIDI note numbers 84 to 95, the coefficients in the frequency range 1 kHz to 3 kHz were selected, and for MIDI note numbers 95 to 107, the frequency coefficients from the range 2 kHz to 4 kHz were used as the feature vector. In [Ellis and Poliner, 2006], a number of spectral feature normalizations were attempted for melody classification; however, none of the normalizations provided a significant advantage in classification accuracy. We have selected the best performing normalization from that experiment, but as we will show in the following section, the greatest gain in classification accuracy is obtained from a larger and more diverse training set.

3 Frame Level Note Classification

The Support Vector Machine 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 of data. Subsequently, labeled training data in a given space are separated by a maximum-margin hyperplane through SVM classification.

Our classification system is composed of 87 OVA binary note classifiers that detect the presence of a given note in a frame of audio, where each frame is represented by a 255-element feature vector as described in Section 2. We took the distance-to-classifier-boundary hyperplane margins as a proxy for a note-class log-posterior probability. In order to classify the presence of a note within a frame, we assume the state to be solely dependent on the normalized frequency data. At this stage, we further assume each frame to be independent of all other frames.

The SVMs were trained using Sequential Minimal Optimization [Platt, 1998], as implemented in the Weka toolkit [Witten and Frank, 2000]. A Radial Basis Function (RBF) kernel was selected for the experiments, and the γ and C parameters were optimized over a global grid search on the validation set using a subset of the training set. In this section, all classifiers were trained using the 92 MIDI training files and classification accuracy is reported on the validation set.

Our first classification experiment was to determine the number of training

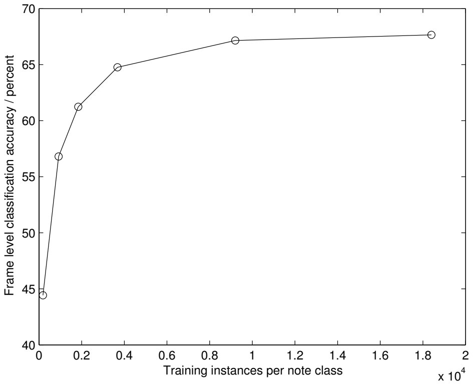

Figure 2: Variation of classification accuracy with number of randomly selected training frames per note, per excerpt.

instances to include from each audio excerpt. The number of training excerpts was held constant, and the number of training instances selected from each piece was varied by randomly sampling an equal number of positive and negative instances for each note. As displayed in Figure 2, the classification accuracy begins to approach an asymptote within a small fraction of the potential training data. Since the RBF kernel requires training time on the order of the number of training instances cubed, 100 samples per note class, per excerpt was selected as a compromise between training time and performance for the remainder of the experiments. A more detailed description of the classification metrics is given in Section 5.

The observation that random sampling approaches an asymptote within a couple of hundred samples per excerpt (out of a total of 6000 for a 60 s excerpt with 10 ms hops) can be explained by both signal processing and acoustic 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 it is the standard used in evaluation). Furthermore, 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. As we noted in Section 2, there are on average 8 note events per second in the training data. Each note may contribute a few usefully different frames due to variations in accompanying notes. Thus we expect many clusters of largely redundant frames in our training data, and random sampling down to 2% (roughly equal to the median prior probability of a specific note occurrence) is a reasonable approximation.

A second experiment examined the incremental gain from adding novel training excerpts. In this case, the number of training excerpts was varied while holding constant the number of training instances per excerpt. The dashed line

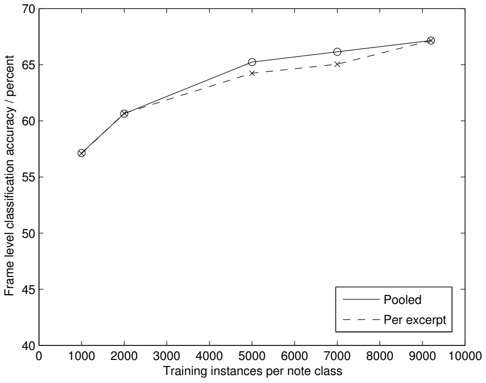

Figure 3: Variation of classification accuracy with the total number of excerpts included, compared to sampling the same total number of frames from all excerpts pooled.

in Figure 3 shows the variation in classification accuracy with the addition of novel training excerpts. In this case, adding an excerpt consisted of adding 100 randomly-selected frames per note class (50 each positive and negative instances). Thus, the largest note classifiers are trained on 9200 frames. The solid curve displays the result of training on the same number of frames randomly drawn from the pool of the entire training set. The limited timbral variation is exhibited in the close association of the two curves.

4 Hidden Markov Model Post Processing

An example ‘posteriorgram’ (time-versus-class image showing the pseudo-posteriors of each class at each time step) for an excerpt of Für Elise is displayed in the top pane of Figure 4. The posteriorgram clearly illustrates both the strengths and weaknesses of the discriminative approach to music transcription. The success of the approach in estimating the 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 by using hidden Markov models to capture temporal constraints.

Similarly to our data-driven approach to classification, we learn temporal structure directly from the training data. We model each note class independently with a two-state, on/off, HMM. The state dynamics, transition matrix and state priors are estimated from our ‘directly observed’ state sequences – the ground-truth transcriptions of the training set.

If the model state at time t is given by q_t, and the classifier output label is

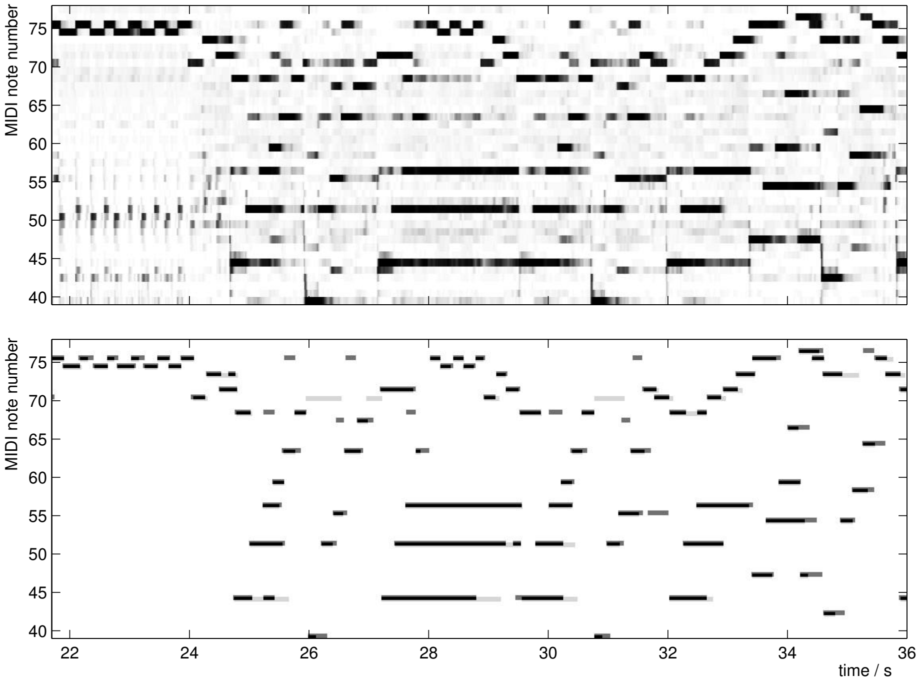

Figure 4: *Top*: Posteriorgram (pitch probabilities as a function of time) for an excerpt of Beethoven's Für Elise. *Bottom*: The HMM smoothed estimation (dark gray) plotted on top of the ground truth labels (light gray; overlaps are black).

c_t, then the HMM will achieve temporal smoothing by finding the most likely (Viterbi) state sequence i.e. maximizing

$$\prod_t p(c_t|q_t)p(q_t|q_{t-1}) \tag{1}$$

where $p(q_t|q_{t-1})$ is the transition matrix estimated from ground-truth transcriptions. We estimate $p(c_t|q_t)$, the probability of seeing a particular classifier label c_t given a true pitch state q_t, with the likelihood of each note being 'on' according to the output of the classifiers. Thus, if the acoustic data at each time is x_t, we may regard our OVA classifier as giving us estimates of

$$p(q_t|x_t) \propto p(x_t|q_t)p(q_t) \tag{2}$$

i.e. the posterior probabilities of each HMM state given the local acoustic features. 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 equation 1.

HMM post-processing results in an absolute improvement of 2.8% yielding a frame-level classification accuracy of 70% on the validation set. Although the improvement in frame-level classification accuracy is relatively modest, the HMM post-processing stage reduces the total onset transcription error by over 7%, primarily by alleviating spurious onsets. A representative result of the improvement due to HMM post processing is displayed in the bottom pane of Figure 4.

5 Transcription Results

In this section, we present a number of metrics to evaluate the success of our approach. In addition, we provide empirical comparisons to the transcription systems proposed by Marolt [2004] and Ryynänen and Klapuri [2005]. It should be noted that the Ryynänen and Klapuri system was developed for general music transcription, and the parameters have not been tuned specifically for piano music.

5.1 Frame-Level Transcription

For each of the evaluated algorithms, a 10 ms frame-level comparison was made between the algorithm (system) output and the ground-truth (reference) MIDI transcript. We start with a binary 'piano-roll' matrix, with one row for each note considered, and one column for each 10 ms time-frame. There is, however, no standard metric that has been used to evaluate work of this kind: we report two, one based on previous piano transcription work, and one based on analogous work in multi-party speech activity detection. The results of the frame-level evaluation are displayed in Table 1.

The first measure is a frame-level version of the metric proposed by Dixon [2000], defined as Overall Accuracy:

$$Acc = \frac{TP}{(FP + FN + TP)} \tag{3}$$

where TP ("true positives") is the number of correctly transcribed voiced frames (over all notes) FP ("false positives") is the number of unvoiced note-frames transcribed as voiced, and FN ("false negatives") is the number of voiced note-frames transcribed as unvoiced. This measure is bounded by 0 and 1, with 1 corresponding to perfect transcription. It does not, however, facilitate an insight into the trade-off between notes that are missed and notes that are inserted.

The second measure, *frame-level transcription error score*, is based on the "speaker diarization error score" defined by NIST for evaluations of 'who spoke when' in recorded meetings [National Institute of Standards and Technology, 2004]. A meeting may involve many people, who, like notes on a piano, are often silent but sometimes simultaneously active (i.e. speaking). NIST developed a metric that consists of a single error score which further breaks down into substitution errors (mislabeling an active voice), "miss" errors (when a voice is truly active but results in no transcript), and "false alarm" errors (when an active voice is reported without any underlying source). This three-way decomposition avoids the problem of ‘double-counting’ errors where a note is transcribed at the right time but with the wrong pitch; a simple error metric as used in earlier work, and implicit in *Acc*, biases systems towards not reporting notes, since not detecting a note counts as a single error (a “miss”), but reporting an incorrect pitch counts as two errors (a “miss” *plus* a “false alarm”). Instead, at every time frame, the intersection of N_{sys} reported pitches and N_{ref} ground-truth pitches counts as the number of correct pitches N_{corr}; the total error score, integrated across all time frames t is then:

$$E_{tot} = \frac{\sum_{t=1}^{T} max\big(N_{ref}(t), N_{sys}(t)\big) - N_{corr}(t)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{4}$$

which is normalized by the total number of active note-frames in the ground-truth, so that reporting no output will entail an error score of 1.0.

Frame-level transcription error is the sum of three components. The first is substitution error, defined as:

$$E_{subs} = \frac{\sum_{t=1}^{T} min\big(N_{ref}(t), N_{sys}(t)\big) - N_{corr}(t)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{5}$$

which counts, at each time frame, the number of ground-truth notes for which the correct transcription was not reported, yet *some* note was reported – which can thus be considered a substitution. It is not necessary to designate *which* incorrect notes are substitutions, merely to count how many there are. The remaining components are “miss” and “false alarm” errors:

$$E_{miss} = \frac{\sum_{t=1}^{T} max\big(0, N_{ref}(t) - N_{sys}(t)\big)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{6}$$

$$E_{fa} = \frac{\sum_{t=1}^{T} max\big(0, N_{sys}(t) - N_{ref}(t)\big)}{\sum_{t=1}^{T} N_{ref}(t)} \tag{7}$$

These equations sum, at the frame level, the number of ground-truth reference notes that could not be matched with any system outputs (i.e. misses after substitutions are accounted for) or system outputs that cannot be paired with any ground truth (false alarms beyond substitutions) respectively. Note that a conventional false alarm *rate* (false alarms per non-target trial) would be both misleadingly small and ill-defined here, since the total number of non-target instances (note-frames in which that particular note did not sound) is very large, and can be made arbitrarily larger by including extra notes that are never used in a particular piece.

The error measure is a score rather than some probability or proportion – i.e. it can exceed 100% if the number of insertions (false alarms) is very high. In line with the universal practice in the speech recognition community we feel this is the most useful measure, since it gives a direct feel for the quantity of errors that will occur as a proportion of the total quantity of notes present.

Table 1: Frame level transcription results on our full synthesized-plus-recorded test set.

Algorithm	Acc	E_{tot}	E_{subs}	E_{miss}	E_{fa}
SVM	67.7%	34.2%	5.3%	12.1%	16.8%
Ryynänen and Klapuri	46.6%	52.3%	15.0%	26.2%	11.1%
Marolt	36.9%	65.7%	19.3%	30.9%	15.4%

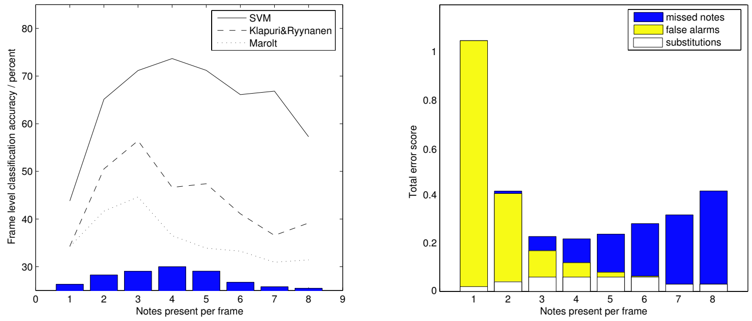

Figure 5: *Left*:Variation of classification accuracy with number of notes present in a given frame and relative note frequency. *Right*: Error score composition as a function of the number of notes present.

It aids intuition to have the errors break down into separate, commensurate components that add up to the total error, expressing the proportion of errors falling into the distinct categories of substitutions, misses, and false alarms.

As displayed in Table 1, our discriminative model provides a significant performance advantage on the test set with respect to frame-level accuracy and error measures – outperforming the other two systems on 33 out of the 35 test pieces. This result highlights the merit of a discriminative model for note identification. Since the transcription problem becomes more complex with the number of simultaneous notes, we have also plotted the frame-level classification accuracy versus the number of notes present for each of the algorithms in the left panel of Figure 5; the total error score (broken down into the three components) with the number of simultaneously occurring notes for the proposed algorithm is displayed in right panel. As expected, there is an inverse relationship between the number of notes present and the proportional contribution of false alarm errors to the total error score. However, the performance degradation is not as severe for the proposed method as it is for the harmonic-based models.

In Table 2, a breakdown of the transcription results is reported between the synthesized audio and piano recordings. The proposed system exhibits the most significant disparity in performance between the synthesized audio and piano recordings; however, we suspect this is because the greatest portion of

Table 2: Classification accuracy comparison for the MIDI test files and live recordings. The MIDI SVM classifier was trained on the 92 MIDI training excerpts, and the Piano SVM classifier was trained on the 20 piano recordings. Numbers in parentheses indicate the number of test excerpts in each case.

Algorithm	**Piano** (10)	**MIDI** (25)	**Both** (35)
SVM (Piano only)	59.2%	23.2%	33.5%
SVM (MIDI only)	33.0%	74.6%	62.7%
SVM (both)	56.5%	72.1%	67.7%
Ryynänen and Klapuri	41.2%	48.3%	46.3%
Marolt	38.4%	40.0%	39.6%

Table 3: Frame level transcription results on recorded piano only (ours and Marolt test sets).

Algorithm / test set	Acc	E_{tot}	E_{subs}	E_{miss}	E_{fa}
SVM / our piano	56.5%	46.7%	10.2%	15.9%	20.5%
SVM / Marolt piano	44.6%	60.1%	14.4%	25.5%	20.1%
Marolt / Marolt piano	46.4%	66.1%	15.8%	13.2%	37.1%
Ryynänen and Klapuri / Marolt piano	50.4%	52.2%	12.8%	21.1%	18.3%

Table 4: Note onset transcription results.

Algorithm	Acc	E_{tot}	E_{subs}	E_{miss}	E_{fa}
SVM	62.3%	43.2%	4.5%	16.4%	22.4%
Ryynänen and Klapuri	56.8%	46.0%	6.2%	25.3%	14.4%
Marolt	30.4%	87.5%	13.9%	41.9%	31.7%

the training data was generated using synthesized audio. In addition, we show the classification accuracy results for SVMs trained on MIDI data and piano recordings alone. The specific data distributions perform well on more similar data, but generalize poorly to unfamiliar audio. This clearly indicates that the implementations based only on one type of training data are overtrained to the specific timbral characteristics of that data and may provide an explanation for the poor performance of neural network-based system. However, the inclusion of both types of training data does not come at a significant cost to classification accuracy for either type. As such, it is likely that the proposed system will generalize to different types of piano recordings when trained on a diverse set of training instances.

In order to further investigate generalization, the proposed system was used to transcribe the test set prepared by Marolt [2004]. This set consists of six recordings from the same piano and recording conditions used to train his neural net and is different from any of the data in our training set. The results of this test are displayed in Table 3. The SVM system commits a greater number of substitution and miss errors compared to its performance on the relevant portion of our test set, reinforcing the possibility of improving the stability and robustness of the SVM with a broader training set. Marolt's classifier, trained on data closer to his test set than to ours, outperforms the SVM here on the overall accuracy metric, although interestingly with a much greater number of false alarms than the SVM (compensated for by many fewer misses). The system proposed by Ryynänen and Klapuri outperforms the classifiaction-based approaches on the Marolt test set; a result that underscores the need for a diverse set of training recordings for a practical implementation of a discriminative approach.

5.2 Note Onset Detection

Frame-level accuracy is a particularly exacting metric. Although offset estimation is essential in generating accurate transcriptions, it is likely of lesser perceptual importance than accurate onset detection. In addition, the problem of offset detection is obscured by relative energy decay and pedaling effects. In order to account for this and to reduce the influence of note duration on the performance results, we report an evaluation of note onset detection.

To be counted as correct, the system must "switch on" a note of the correct pitch within 100 ms of the ground-truth onset. We include a search to associate any unexplained ground-truth note with any available system output note within the time range in order to count substitutions before scoring misses and false alarms. We use all the metrics described in Section 5.1, but the statistics are reported with respect to onset detection accuracy rather than frame-level transcription accuracy. The note onset transcription statistics are given in Table 4. We note that even without a formal onset detection stage, the proposed algorithm provides a slight advantage over the comparison systems on our test set.

6 Discussion

We have shown that a discriminative model for music transcription is viable and can be successful even when based on a modest amount of training data. The proposed system of classifying frames of audio with SVMs and temporally smoothing the output with HMMs provides advantages in both performance and simplicity when compared to previous approaches. Additionally, the system may be easily generalized to learn many musical structures or trained specifically for a given genre or composer. A classification-based system for dominant melody transcription was recently shown to be successful in [Ellis and Poliner, 2006]. As a result, we believe that the discriminative model approach may be extended to perform multiple instrument polyphonic transcription in a data association framework.

We recognize that separating the classification and temporal constraints is somewhat *ad hoc*. Recently, Taskar et al. [2003] suggested an approach to apply maximum margin classification in a Markov framework, but we expect that solving the entire optimization problem would be impractical for the scope of our classification task. Furthermore, as shown in Section 3, treating each frame independently does not come at a significant cost to classification accuracy. Perhaps the existing SVM framework may be improved by optimizing the discriminant function for detection, rather than maximum-margin classification as proposed in [Schlkopf et al., 2001].

A close examination of Figure 4 reveals that many of the note-level classification errors are octave transpositions. Although these incorrectly transcribed notes may have less of a perceptual effect on resynthesis, there may be steps we could take to reduce these errors. Perhaps more advanced training sample selection such as selecting members of the same chroma class or frequently occurring harmonically related notes (i.e. classes with the highest probability of error) would be more valuable counter-examples on which to train the classifier. In addition, rather than treating note state transitions independently, a more advanced HMM observation could also reduce common octave errors.

A potential solution to resolve the complex issue of offset estimation may be to include a hierarchical HMM structure that treats the piano pedals as hidden states. A similar hierarchical structure could also be used to include contextual clues such as local estimations of key or tempo. The HMM system described in this paper is admittedly naive; however, it provides a significant improvement in temporal smoothing and greatly reduces onset detection errors. The inclusion of a formal onset detection stage could further reduce note detection errors occurring at rearticulations.

Although the discriminative model provides advantages in performance and simplicity, perhaps the most important result of this paper is that no formal acoustical prior knowledge is required in order to perform transcription. At the very least, the proposed system appears to provide a front-end advantage over spectral-tracking approaches, and may fit nicely into previously-presented temporal or inferential frameworks. In order to facilitate future research using classification-based approaches to transcription, we have made the training and evaluation data available at `http://labrosa.ee.columbia.edu/projects/piano/` .

Acknowledgments

We would like to thank Dr. Matija Marolt, Dr. Anssi Klapuri, and Matti Ryynänen for their valuable contributions to the empirical evaluations. We would also like to thank Prof. Tony Jebara for his insightful discussions. 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

J.P. Bello, L. Daudet, and M. Sandler. Time-domain polyphonic transcription using self-generating databases. In *Proc. 112th Convention of the Audio Engineering Society*, Munich, May 2002.

A.T. Cemgil, H.J. Kappen, and D. Barber. A generative model for music transcription. *To Appear in IEEE Transactions on Speech and Audio Processing*, 2005.

S. Dixon. On the computer recognition of solo piano music. In *Proc. Australasian Computer Music Conference*, Brisbane, 2000.

D. Ellis and G. Poliner. Classification-based melody transcription. *Machine Learning*, 2006. URL `http://dx.doi.org/10.1007/s10994-006-8373-9`.

S. Godsill and M. Davy. Bayesian harmonic models for musical pitch estimation and analysis. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando, 2002.

K. Kashino and S. Godsill. Bayesian estimation of simultaneous musical notes based on frequency domain modelling. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, 2004.

A. Klapuri. A perceptually motivated multiple-f0 estimation method. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, October 2005.

M. Marolt. A connectionist approach to transcription of polyphonic piano music. *IEEE Transactions on Multimedia*, 6(3):439–449, 2004.

J.A. Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, 1(4):32–38, 1977.

National Institute of Standards and Technology. Spring 2004 (RT-04S) rich transcription meeting recognition evaluation plan, 2004. URL `http://nist.gov/speech/tests/rt/rt2004/spring/documents/rt04s-meeting-e%val-plan-v1.pdf`.

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.

L. Rossi, G. Girolami, and M. Leca. Identification of polyphonic piano signals. *Acustica*, 83:1077–1084, 1997.

M. Ryynänen and A. Klapuri. Polyphonic music transcription using note event modeling. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, October 2005.

B. Schlkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

A.D. Sterian. *Model-based Segmentation of Time-Frequency Images for Musical Transcription.* PhD thesis, University of Michigan, 1999.

B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Proc. Neural Information Processing Systems Conference (NIPS)*, Vancouver, 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.

Table 5: MIDI compositions from http://www.piano-midi.de/ . † denotes songs for which piano recordings were made.

Composer	Training	Testing	Validation
Albéniz	España (Prélude†, Malagueña, Sereneta, Zortzico) Suite Española (Granada, Cataluña, Sevilla, Cádiz, Aragon, Castilla)	España (Tango), Suite Española (Cuba)	España (Capricho Catalan)
Bach	BWV 850†	BWV 847†	BWV 846
Balakirew	Islamej†		
Beethoven	Appassionata 1-3, Moonlight (1, 3), Pathetique (1)†, Waldstein (1-3)	Für Elise† Moonlight (2), Pathetique (3)†	Pathetique(2)
Borodin	Petite Suite (In the monastery†, Intermezzo, Mazurka, Serenade, Nocturne)	Petite Suite (Mazurka)	Réverie
Brahms	Fantasia (2†, 5), Rhapsodie	Fantasia (6)†	
Burgmueller	The pearls†, Thunderstorm	The Fountain	
Chopin	Opus 7 (1†, 2), Opus 25 (4), Opus 28 (2, 6, 10, 22), Opus 33 (2, 4)	Opus 10 (1)†, Opus 28 (13)	Opus 28 (3)
Debussy	Suite bergamasque (Passepied†, Prélude)	Menuet	Clair de Lune
Granados	Danzas Españolas (Oriental†, Zarabanda)	Danzas Españolas (Villanesca)	
Grieg	Opus 12 (3), Opus 43 (4), Opus 71 (3)†	Opus 65 (Wedding)	Opus 54 (3)
Haydn	Piano Sonata in G major 1†	Piano Sonata in G major 2 †	
Liszt	Grandes Etudes de Paganini (1†-5)	Love Dreams (3)	Grandes Etudes de Paganini (6)
Mendelssohn	Opus 30 (1)†, Opus 62 (3,4)	Opus 62 (5)	Opus 53 (5)
Mozart	KV 330 (1†-3), KV 333 (3)	KV 333 (1)†	KV 333 (2)
Mussorgsky	Pictures at an Exhibition (1†,3,5-8)	Pictures at an Exhibition (2,4)	
Schubert	D 784 (1†,2), D 760 (1-3), D 960 (1,3)	D 760 (4)†	D 960 (2)
Schumann	Scenes from Childhood (1-3, 5, 6†)	Scenes from Childhood (4) †	Opus 1 (1)
Tchaikovsky	The Seasons (February, March, April†, May, August September, October, November, December)	The Seasons (January†, June)	The Seasons (July)