IMPROVING GENERALIZATION FOR CLASSIFICATION-BASED 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

ABSTRACT

In this paper, we present methods to improve the generalization capabilities of a classification-based approach to polyphonic piano transcription. Support vector machines trained on spectral features are used to classify frame-level note instances, and the independent classifications are temporally constrained via hidden Markov model post-processing. Semi-supervised learning and multiconditioning are investigated, and transcription results are reported for a compiled set of piano recordings. A reduction in the frame-level transcription error score of 10% was achieved by combining multiconditioning and semi-supervised classification.

1. INTRODUCTION

Music transcription is the process of creating a score (i.e. a symbolic representation of the notes played) from a piece of audio. The ability to generate a list of the note times and pitches from a recording has numerous practical applications ranging from musicological analysis to content-based music retrieval tasks. Although expert musicians are capable of transcribing polyphonic pieces of music, the process is often arduous for complex recordings. As such, a number of systems have been developed that attempt to automatically generate music transcriptions. Unfortunately, the harmonic spectral structure at the core of musical consonance often results in constructive and destructive interference, making the transcription of polyphonic music a very challenging problem.

Moorer presented the first system for transcribing simultaneous notes in [1]. Since then, a number of models for polyphonic transcription have been presented in both the frequency domain [2] and the time domain [3]. Over time, the number of constraints required of the input audio by a given system has generally been reduced, and the overall transcription accuracy has gradually improved. More recently, systems such as [4] combined harmonic analysis with pattern recognition techniques in order to achieve relatively high transcription accuracy on complex, polyphonic piano and pop recordings.

In [5], a classification approach to music transcription was presented in which generic classifiers – rather than models specifically designed to exploit the structure of musical tones – were used to detect the presence or absence of a particular note. The classification-based system compared favorably to model-based systems when both the training and testing recordings were made with the same set of pianos; however, it did not generalize as well when presented with piano recordings from different environments, perhaps with slightly different tuning. Common to many supervised learning tasks, the classifier performance is limited by the amount and diversity of the labeled training data available; however, a great deal of relevant but unlabeled audio data exists. In this paper we seek to exploit this vast pool of unlabeled data, and to improve the value of the limited labeled data we have, to make the classification-based music transcription system generalize better to unseen recording conditions and instruments.

*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.

2. CLASSIFICATION-BASED TRANSCRIPTION

A supervised classification system infers the correct note transcriptions based only on training from labeled examples. The base audio representation is the short-time Fourier transform magnitude (spectrogram), upon which a set of support vector machines are trained to classify the presence or absence of each note in a given frame (which may contain other notes). The independent per-note classifications are smoothed in time with a hidden Markov model (HMM) post-processing stage.

2.1. 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 data and evaluate our system on validation and testing sets composed of piano recordings in different environments.

The labeled training data used in our experiments consisted of 92 randomly selected songs from the Classical Piano Midi Page, http://www.piano-midi.de/ . 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, 20 training recordings were made from a subset of the MIDI files using a Yamaha Disklavier playback grand piano. The MIDI performances were recorded as monaural audio files at a sampling

rate of 44.1 kHz, and 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. In addition to the labeled training audio, 54 unlabeled polyphonic piano files were collected from 20 different recording environments to be used in the semi-supervised learning experiments.

The validation set used to tune the parameters in our experiments was collected from the real world computing (RWC) database [6], and the ground-truth transcripts for the three validation files were aligned by Cont [7]. Our test set consisted of 19 piano recordings made from three different pianos including six pieces from the test set generated by Marolt [4], two pieces created by Scheirer [8], and 11 pieces recorded on a Roland HP 330e digital piano downloaded from the Classical Piano Midi Page. In some cases, limitations in the MIDI file parsing resulted in a constant time scale between labels and audio, so a compensating scaling constant was estimated to maximize the alignment between the time-compensated transcript and a noisy transcription of the audio made by the baseline SVM system.

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.

2.2. Frame-Level Note Classification

The support vector machine (SVM) is a supervised classification system that uses a hypothesis space of linear functions in a high dimensional feature space to learn separating hyperplanes that are maximally distant from all training points, or that minimizes the extent to which training patterns fall on the wrong side of the boundary. As such, SVM classification attempts to generalize an optimal decision boundary between classes of data.

Our classification system consists of 87 one-versus-all (OVA) 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. Different classifiers looked at different spectral regions, depending on where relevant energy was likely to occur: For the detectors of MIDI note numbers 21 to 83 (i.e. the first 63 piano keys), the input feature vector was composed of the 255 spectral coefficients corresponding to frequencies below 2 kHz. For MIDI note numbers 84 to 95, coefficients in the frequency range 1 kHz to 3 kHz were selected, and for MIDI note numbers 96 to 107, the range 2 kHz to 4 kHz was used as the feature vector. 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 frequency data, which is subjected to local mean and variance normalization [5]. At this stage, we further assume each frame to be independent of all other frames.

The SVMs were trained using Sequential Minimal Optimization [9], as implemented in the Weka toolkit [10]. A linear kernel was selected for the experiments, and the fixed penalty parameter, C, was optimized over a global grid search on the validation set using a subset of the training data. In previous experiments [5], more advanced kernels (e.g. radial basis functions) were used to perform classification; however, the higher-order kernels typically resulted in modest performance gains at the cost of a significant increase in computational complexity.

2.3. Hidden Markov Model Post-Processing

The obvious fault with classifying each frame independently is that the inherent temporal structure of the music is not exploited. We attempted 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 learned the temporal structure directly from the training data by modeling each note class independently with a two-state, on/off, HMM. The state dynamics, transition matrix and state priors were 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 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.

3. GENERALIZED LEARNING

Although the classification-based system performs well on different recordings made from the same set of pianos in the same environment, the success of the transcription system does not translate well to novel pianos and recording environments. In particular, slight differences in tuning have been identified as problematic. In this section, we propose methods for improving generalization by learning from unlabeled training data and by augmenting the value of the data for which we have labels.

3.1. Semi-supervised Learning

Millions of music recordings exist, yet only a very small fraction of them are labeled with corresponding transcriptions. Since the success of our classification-based transcription system is so heavily dependent on the quantity and diversity of the available training data, we have attempted to incorporate more of the data available to train new classification systems by applying different techniques to assign labels to unlabeled data.

Nearest neighbor clustering is a simple classification system in which a label is assigned to a particular point based on its proximity, using a given distance metric, to its k-nearest neighbors in the

feature space. For each frame-level feature vector calculated from the unlabeled data set, a set of 87 binary labels was generated by calculating the Euclidian distance to each point in the training data for a given note class and assigning the label of the (majority vote of the) k-nearest neighbors to the unlabeled point. For each note, an equal number of positive and negative training instances generated from the unlabeled data was added to the original training data set, and a new system of SVM classifiers was trained.

In our semi-supervised SVM approach, labels are assigned to unlabeled data by classifying the unlabeled points with our baseline SVM system. Frames were added to the training data provided their classified distance to the training boundary fell within a certain range. As an alternative to using the raw classifier output as a proxy for sampling selection, the HMM post-processing stage may be applied to the output of the unlabeled data classification. In some cases, the inclusion of the HMM stage results in class assignment updates due to temporal context, thus improving the insight of the trained classifier in ambiguous cases. Again, for each note, an equal number of positive and negative training instances generated from the unlabeled data was added to the original training set in order to create updated classifiers.

3.2. Multiconditioning

The quantity and diversity of the training data was extended by resampling the audio to effect a global pitch shift. Each recording from the training set was resampled at rates corresponding to frequency shifts of a fraction of a semitone in order to account for differences in piano tunings. The corresponding ground-truth labels were unaffected (since the target note class remained the same); however, the time axis was linearly interpolated in order to adjust for the time scaling. Symmetrically shifted frequency data was added to the original training set to make additional classifiers.

4. EXPERIMENTS

4.1. Evaluation Metrics

For each of the evaluated algorithms, a 10 ms frame-level comparison was made between the system output and the ground-truth transcript. We used the *frame-level transcription error score*, which is based on the "speaker diarization error score" defined by NIST for evaluations of 'who spoke when' in recorded meetings [11], to evaluate the proposed systems. We start with a binary 'piano-roll' matrix, which consists of one row for each note considered, and one column for each 10 ms time-frame. 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:

$$E_{tot} = \frac{\sum_{t=1}^{T} \max\big(N_{\text{ref}}(t), N_{\text{sys}}(t)\big) - N_{\text{corr}}(t)}{\sum_{t=1}^{T} N_{\text{ref}}(t)} \quad (3)$$

which is normalized by the total number of active note-frames in the ground-truth. Under this scheme, transcribing all notes as permanently silent 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:

$$E_{subs} = \frac{\sum_{t=1}^{T} \min\big(N_{\text{ref}}(t), N_{\text{sys}}(t)\big) - N_{\text{corr}}(t)}{\sum_{t=1}^{T} N_{\text{ref}}(t)} \quad (4)$$

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. The remaining components are "miss" and "false alarm" errors:

$$E_{miss} = \frac{\sum_{t=1}^{T} \max\big(0, N_{\text{ref}}(t) - N_{\text{sys}}(t)\big)}{\sum_{t=1}^{T} N_{\text{ref}}(t)} \quad (5)$$

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

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. We also note that the error measure is a score rather than a probability or proportion – i.e. it can exceed 100% if the number of insertions (false alarms) is very high. We favor this measure (which reflects common practice in speech recognition evaluation) because of the natural additive breakdown into the three types of error.

4.2. Experiments

In our first semi-supervised learning experiment, each frame of audio in the unlabeled data set was assigned the label of its k-nearest neighbors. From each song in the unlabeled set and for each note in the classification system 50 negative training instances and 50 positive training instances (when available) were added to the original set of training data This addition increased the quantity of training data by approximately 50%. The amount of training data used was held constant while the number of nearest neighbors, k, was varied from 1 to 7 in steps of 2 (odd values only). A classification system of SVMs was trained from each of the updated training sets; however, each resulted in a negligible change in transcription error on the validation set.

The baseline SVM system was then used to estimate transcriptions for each song in the unlabeled data set. Positive training instances were selected by varying the range of the distance to classifier boundary used for sampling selection. While holding the 50% increase in training data constant, we attempted sampling from a series of ranges by performing a grid search over the distance to classifier boundary, the best of which resulted in a 0.8% decrease in total error score on the validation set. In addition to sampling different distance to classifier boundary ranges to generate training instances, the HMM post-processing stage was applied to the raw classifier transcriptions on the unlabeled data set. From each song, 50 positive and negative instances were selected for each note class and additional classifiers were trained resulting in an 1.1% point reduction in the total error on the validation set. In order to demonstrate the variation in classifier performance due to the addition of semi-supervised training instances, the amount of estimated training data was varied as a fraction of percent increase in total data from 10-100% (in 10% increments) resulting in a monotonically decreasing reduction in the total error score on the validation set up to 1.9% for the training instances generated from the output of the SVM classifier with HMM smoothing.

Table 1: Transcription error results on the 19 song test set.

	Frame-level transcription			
System	E_{tot}	E_{subs}	E_{miss}	E_{fa}
SVM (baseline)	69.7	15.8	36.3	17.6
k-NN	70.5	15.1	37.3	18.1
SVM	68.9	10.2	49.7	9.0
SVM + HMM	68.5	15.6	33.9	19.0
MC	63.0	12.4	39.5	11.1
MC + SVM + HMM	59.1	8.6	38.6	12.3

We trained four additional classifiers in order to investigate the effects of generating training data from resampled audio. Each recording from the training set was resampled at symmetric rates corresponding to $\pm$ 0.5, 1.0, 1.5, 2.0% deviations from the original tone. In this experiment, the amount of resampled training data was held constant, while the range of resampled audio used to train the classifiers was varied. Incorporating the resampled audio resulted in 3.1%, 1.2%, 1.1%, and 0.9% respective reductions in the total frame-level error on the validation set. We suspect that the resampling rates closer to the original tone provide an advantage in performance because they are more likely to be in line with mild instrument detuning. The top performing resampled classifier was then used to generate labels for the unlabeled data set. The transcriptions were temporally smoothed via the HMM, and the estimated labels were sampled (50 positive and negative instances per class) to create additional training data for a final set of classifiers. The combination of the semi-supervised learning with the resampling technique resulted in an improvement of 4.7% on the validation set.

The parameters for each of the generalization techniques were optimized on the validation set based on frame-level transcription error, and the top performing classifier from each of the proposed systems was used to classify the 19 songs in the test set. The corresponding frame-level transcription and note onset detection results are displayed in Table 1. We note that the top performing system provided a 10% reduction in frame-level error on the test set. Finally both the baseline system and the system combining training data from multiconditioning and semi-supervised learning were used to classify 10 additional songs recorded on the piano used to create the training recordings. While the inclusion of the diversifying training data results in a mild performance reduction of 0.4% on the original instruments, the improvement in generalization seems to warrant the addition.

5. DISCUSSION

We have shown that a modest reduction in total transcription error may be achieved by combining multiconditioning and semi-supervised learning to generate additional training data for a classification based music transcription system. The proposed methods demonstrate that limited quantities of training data may be augmented to reduce classification error. We recognize that the semi-supervised method for selecting training instances is somewhat ad hoc. For instance, an optimal method for semi-supervised classification is presented [12] in which the misclassification error for each unlabeled feature vector is calculated for both the case where the point is a positive instance and the cases where the point is a negative instance of a given class. The label assigned to the each unlabeled point corresponds to the class that results in the smallest structural risk. We plan to further investigate the application of transductive learning; however, the proposed method in which we combine temporal smoothing to the classification outputs allows us to incorporate additional knowledge that is unavailable in the independent classification setting. As such, methods such as learning from the cases in which the SVM and the HMM disagree may allow us to more efficiently and effectively learn note classification boundaries, and we plan to pursue this insight in our future work.

6. REFERENCES

[1] J. Moorer, "On the transcription of musical sound by computer," *Computer Music Journal*, vol. 1, no. 4, pp. 32–38, 1977.

[2] A. Sterian, "Model-based segmentation of time-frequency images for musical transcription," Ph.D. dissertation, University of Michigan, 1999.

[3] J. 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.

[4] M. Marolt, "A connectionist approach to transcription of polyphonic piano music," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 439–449, 2004.

[5] G. Poliner and D. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007. [Online]. Available: http://www.hindawi.com/GetPDF.aspx?doi=10.1155/2007/48317

[6] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. International Conference on Music Information Retrieval*, Paris, October 2002, pp. 287–288.

[7] A. Cont, "Realtime multiple pitch observation using sparse non-negative constraints," in *International Conference on Music Information Retrieval*, Victoria, October 2006.

[8] E. Scheirer, "Music-listening system," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[9] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods – Support Vector Learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1998, pp. 185–208.

[10] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools with Java implementations*. San Francisco, CA, USA: Morgan Kaufmann, 2000.

[11] National Institute of Standards and Technology, "Spring 2004 (RT-04S) rich transcription meeting recognition evaluation plan," 2004. [Online]. Available: http://nist.gov/speech/tests/rt/rt2004/spring/documents/rt04s-meeting-eval-plan-v1.pdf

[12] K. Bennet and A. Demiriz, "Semi-supervised support vector machines," in *Proc. Advances in Neural Information Processing Systems*, Denver, December 1998, pp. 368–374.