AN AI APPROACH TO

AUTOMATIC NATURAL MUSIC TRANSCRIPTION

Karey Shi • Michael Bereket • {kareyshi, mbereket}@stanford.edu

BACKGROUND

AUTOMATIC MUSIC TRANSCRIPTION (AMT) is the task of generating a symbolic score-like representation of a polyphonic acoustical signal

- CHALLENGES: acoustical signal of concurrent notes can have complex interactions, there can be large variations in audio signals between instruments, and the combinatorial output space is very large

OUR GOAL: to implement an end-to-end pipeline that converts .wav piano audio files into a "natural" score-like representation

2 MAIN STEPS: *acoustic modeling* (which closely follows Sigtia et al.) and *score generation with smoothing.*

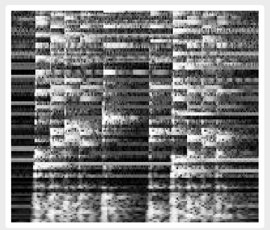
ACOUSTIC MODEL

DATASET: 138 MIDI files of expressive classical piano pieces

INPUT PREPROCESSING:

Generate .wav files from MIDIs → Downsample raw audio data to 16 kHz → Apply CQT → Normalize each feature across all frames

CONSTANT Q TRANSFORM (CQT): represents amplitude against a log frequency scale → geometrically spaced center frequencies and reduced number of frequency bins (less features)

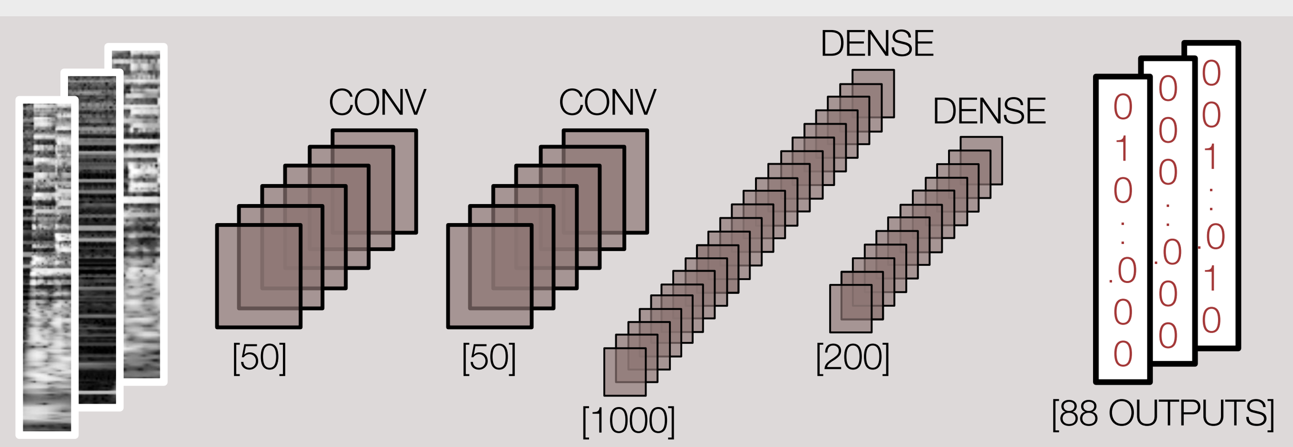


LABEL FORMAT:

MIDI files (of note-on, note-off events) → for each time slice: Binary vector of size 88 whether i^{th} note was present

CONVOLUTIONAL NEURAL NETWORK (CNN)

- Input = context window of frames (predicting for center frame)
- Pooling layers & weight sharing → reduce # of parameters
- Combined with CQT, CNN can learn pitch invariant features

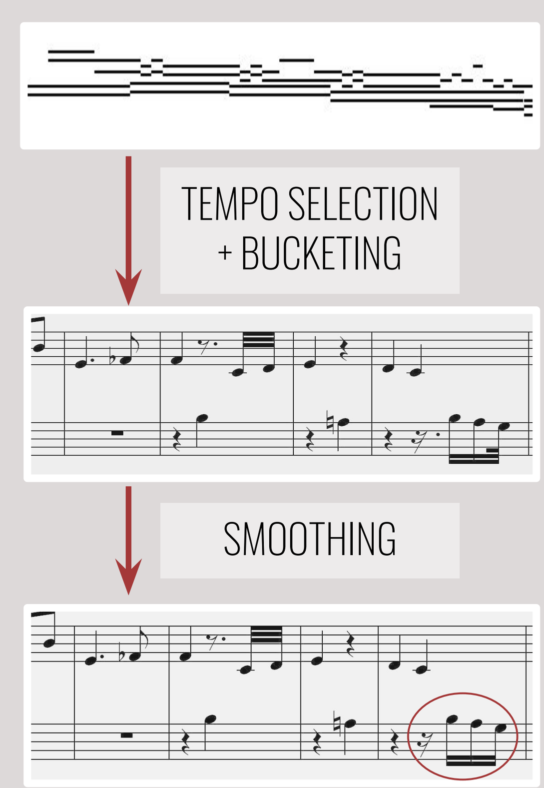


TRAINING: learning rate step decay, 0.3 dropout rate (after conv & dense layer), SGD with 0.9 momentum, loss = binary cross entropy

SCORE GENERATION

OBJECTIVE: derive a standard score from human performances

PIPELINE:



TEMPO SELECTION + BUCKETING:

- Given note-length observations, selects a constant tempo for the piece and places notes in buckets (e.g. ⅛ note, ¼ note) to produce score-like representation
- Difficult due to tempo irregularities and emotion in performances

SMOOTHING:

- Smooth rhythms and irregular sequences of note-events (HMM)

TEMPO SELECTION MODEL:

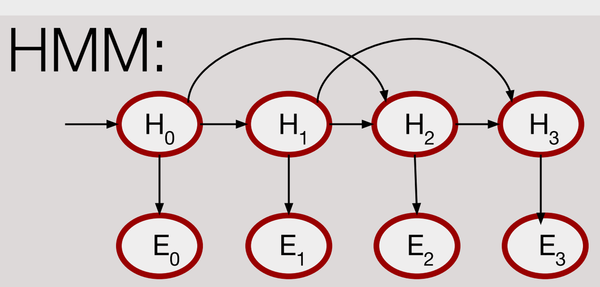
BUCKETS → LINEAR MODEL $\xrightarrow{\text{SGD}}$ CANDIDATE TEMPOS

$$Loss(x^{(i)}, \theta) = min_{b \in Buckets}(len(x^{(i)}, \theta) - len(b, \theta))^2$$

Where θ= tempo, and x^i represents the i^{th} note event

NOTE: multiple initial tempos θ for to explore local maxima

HMM:



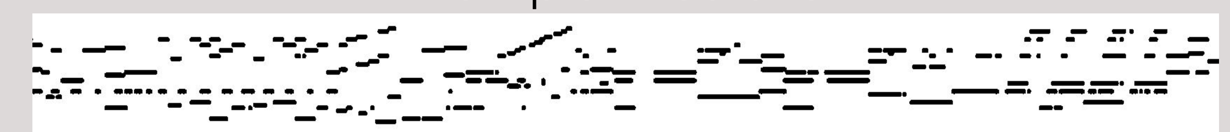
HIDDEN STATES: ground-truth rhythm buckets
EMISSION STATES: observed rhythm buckets
P_{TRANS}: n-gram probs over H_i's
P_{EMIT}: multinomial conditioned on H_i
INFERENCE: smoothing & sequence optimizations

RESULTS & DISCUSSION

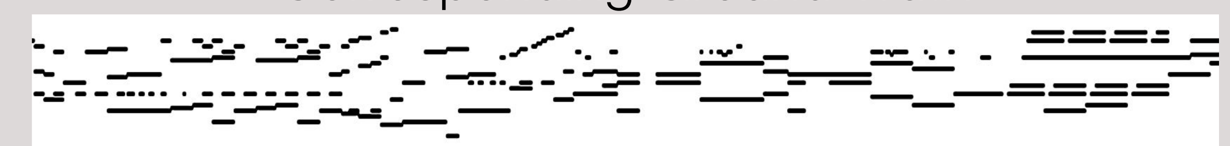
ACOUSTIC MODEL

TRAINING SET: 200680 frames, 110 songs | **TEST SET**: 50170 frames, 28 songs

Example Prediction



Corresponding Ground Truth



F1-SCORE		ACCURACY	
TRAIN: 74.09%	TEST: 54.45%	TRAIN: 99.81%	TEST: 98.72%

Our results are comparable with current state of the art transcription models; sparse nature of audio data and limited variation of sounds pose challenges

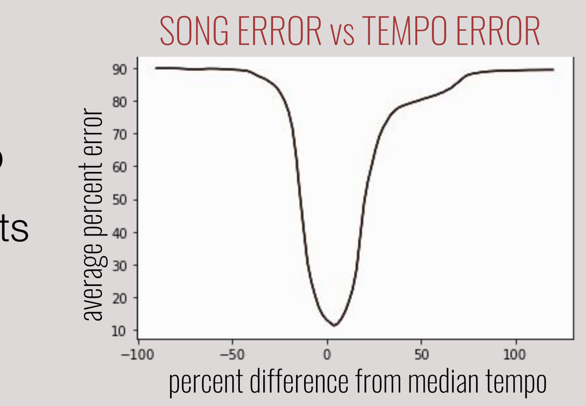
SCORE GENERATION

For **90.8%** of songs without multiple tempo, at least one out of top 4 predicted candidate tempos (ranked by loss) is within 9% of actual median tempo up (tolerant to doubling/halving)

TEMPO SELECTION ERROR ANALYSIS: majority of error can be alleviated in tempo selection and bucketing step

HMM ERROR ANALYSIS: HMM weights expected hidden sequences too heavily, leading to drastic change. Need to prioritize emissions

SONG ERROR vs TEMPO ERROR



NEXT STEPS

- Gather more data for acoustic model (generate synthetically composed audio, acquiring a wider range of sound fonts)
- Implement a weighted loss function for CNN
- Complete full pipeline (CNN→select tempo→smoothing)

REFERENCES:

[1] Sigtia S., Benetos E., & Dixon S. (2016). An End-to-End Neural Network for Polyphonic Piano Music Transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* 24(5), 927-939.
[2] Krueger, B. (2016). *Classical Piano: Midi Page.* Retrieved from http://www.piano-midi.de/
[3] Fugal, H. (2009). Optimizing the Constant-Q Transform in Octave. Paper presented at Linux Audio Conference, Parma, Italy, 2009.