Human Relative Pitch Identification and Audio Perception as Monte Carlo Markov Chains

Joshua Hellerstein
Massachusetts Institute of Technology
joshh@mit.edu

Christie hong
Massachusetts Institute of Technology
cshong@mit.edu

***Abstract*—"Perfect pitch" is a phenomenon and unique skill that some people possess which allows them to hear a pitch and, without any frame of reference, name the corresponding musical note. Most people, however, rely on some level of relative pitch to identify notes. A well-developed sense of *relative pitch* allows one to hear and recognize relationships between notes (intervals), even if he/she doesn't know the note names explicitly. Interval recognition is extremely helpful in the task of sight reading and the challenge of intonation, or playing in tune. With relative pitch training (often implicitly taught by studying music), a musician seeks to improve their ability to recognize notes. In this study we explore how musical pitch identification (across different levels of musical training) can be modeled as a Markov Chain Monte Carlo. We propose a novel way (not found in any prior literature) of using MCMC-Metropolis-Hastings to estimate the musical notes present in an arbitrary audio excerpt composed of piano notes and background noise. We then compare the model to how humans execute relative pitch identification, across different musical abilities. The comparison shows that our MCMC-MH model, tunable to different skill-levels, presents a surprisingly good model for how humans converge to the correct pitches present in a musical excerpt.**

`https://github.com/jhell96/music-perception-mcmc`

1. Introduction

For those who play or aspire to play instruments or sing, relative pitch is a useful tool to recognize relationships between notes (major, minor, augmented), and blend music with other instruments or voices. Relative pitch is defined as the ability of a human to identify and re-create a given musical note by comparing it to a reference note and identifying intervals between the reference and the true pitch. As one trains more and more in music, their precision with relative pitch increases.

But how is relative pitch applicable when one is performing in a noisy concert hall, or singing with other people in harmony? Interestingly enough, humans have an incredible ability to focus their auditory attention on a particular stimulus while filtering out a range of other stimuli – whether it be a party, a sports game, a bar, or a concert – humans can focus on a single part of an audio signal, and estimate its value. This phenomenon is known as the "cocktail party effect" [1]. Within the relative pitch problem we're trying to model, lies a simplified version of the cocktail party problem. In our exploration, we have mixed a set of true signals (notes and background noise), and attempt to model the human recovery of these notes.

We hypothesize that humans go through a measurement and trial process where they try playing (on a piano, for instance) certain note(s) and measure how similar their trials are to the audio. They then repeat this process until they converge on the ground truth pitches in the audio. This process of narrowing down the best possible choices is highly similar to the inference method of Markov Chain Monte Carlo (MCMC), on the likelihood of notes, given some audio observation $P(N|D)$ where N is the set of notes given some mixed audio signal data D. We believe that the MCMC proposal distribution, and likelihood function $P(D|N)$ corresponds to musical experience. As one trains their ear over time, they can better tell which notes N could have produced the audio signal D.

Formally, we want to approximate the probability

$$P(N|D) = \frac{P(D|N)P(N)}{P(D)}$$

where N are the musical notes in the mixed-audio D using MCMC-Metropolis-Hastings. We come up with a novel tunable model for the likelihood function $P(D|N)$ based on analyzing the similarity of the frequency spectrum of D, and proposal notes N. In doing so, we can estimate which notes are most likely to be in the audio signal. We tune our MCMC model to converge at a similar rate to humans, and approximate different skill levels.

2. Related Works

The previous work done in the domain of audio perception and MCMC is quite limited. Hence, we have taken a look at works done in audio perception, and separately in MCMC in the realm of cognitive science.

2.1. Past Works in Audio Perception

One particular work that is very similar to that of our exploration comes from *Simpson, et al.* and their research in extracting vocals from musical mixtures. More generally, their work looks into solving the cocktail party problem previously mentioned.

In this study, a database of songs that were available as a set of individual tracks that each contained a different instrument or voice, as well as the overlaid mixed version of the song were utilized as the data set. Each track was then split into 20 second segments to create a spectrogram, showing frequencies over time; this resulted in a unique fingerprint that identified a voice or instrument [7]. The key idea was that the spectrogram of the overlaid mixed track was essentially the same as all the component spectrograms put together. The database of individual tracks and the overlaid mixed track, as well as the spectrograms were our source of inspiration to have a similar database in our own exploration.

With all of these spectrograms, the team trained a convolutional deep neural network to pick out the voice from this mixed track, i.e. separate out the voice's unique spectrogram from the other spectrograms present. The results from *Simpson, et al.* are incredible, as they successfully managed to separate a singer's voice from the background music faster and better than a human could, and successfully solve the cocktail party problem.

2.2. MCMC in Cognitive Science

There have been a few studies of MCMC within cognitive science. One in particular that we took a look at was a study of MCMC and memory search. *Bourgin et al.* goal was to show that human thinking and human memory search does indeed mirror that of the MCMC process when taking the remote associates test (RAT). The RAT is a creativity test which consists of three common stimulus words (e.g. 'surprise,' 'line,' 'birthday') that appear unrelated. The test subject must think of a fourth word (in this case, 'party') that somehow relates to each of the first three words [2].

Bourgin et al. used human data as a base line for 25 problems, and ran a MCMC model with 100 simulations on the 25 RAT problems. They utilized the Metropolis-Hastings algorithm to calculate the transition state When running the MCMC model, they found that the model produces the same response clustering patterns, local dependencies, undirected search trajectories, and low associative hierarchies witnessed in human responses. The larger takeaway from this study is that humans do in fact think like an MCMC model – but can this be applied in auditory perception?

Intrigued by *Bourgin et al.*'s work in human memory, we then looked a bit deeper in human cognition in a different space. Specifically we took a look at *Yildirim et al.*'s study in MCMC models refining the generation of human faces in the visual perception space.

In this particular paper, the goal was to test a generative model, enhanced by MCMC, in the domain of face recognition. Our interest for this study lied specifically with the MCMC model. With each MCMC sweep, *Yildirim et al.* iterated a proposal-and-acceptance loop over several groups of random variables, until convergence. In just a few MCMC sweeps, each recognition-initialized chain converged much faster with results almost as good as the best randomly initialized inference chains after tens or hundreds of sweeps.

The results of the model and a successive behavioral experiment were clearly substantial. The model could not only reconstruct the shape and texture of a novel face from a single view, but also account for human behavior in "hard" recognition tasks, as well as qualitatively match neural responses in a network of face-selective brain areas [8]. This particular study served as the motivation behind our own exploration with MCMC and human audio perception within the problem of relative pitch.

3. Background

3.1. MCMC

Markov Chain Monte Carlo (MCMC) is a class of approximation algorithms. The variant of Metropolis-Hastings allows us to sample from a complex distribution (in our case $P(N|D)$) with only samples from a distribution proportional to our target distribution (in our case, $P(D|N)$ according to Bayes rule). The algorithm and techniques for implementing MCMC-MH can be found in various related works [4].

3.2. Music Perception

The perception of music is influenced by how the human auditory system encodes and retains acoustic and frequency information. Pitch is one of the main dimensions of sound, and is the perceptual correlate of periodicity in sounds [6]. Periodic harmonic sounds (including musical notes) are waveforms that repeat in time with a fundamental frequency F_0. Each musical note typically has an additional collection of frequencies, known as the harmonic spectra of a note. The harmonic spectra contains additional frequencies which are all weighted multiples of the common fundamental frequency $F0$, and change from instrument to instrument (while the fundamental frequency $F0$ e.g. musical pitch "A" remains the same). Both frequency and time information are present in the peripheral auditory system; frequency information is maintained throughout the auditory system up to and including the primary auditory cortex, whereas periodicity information in the time domain is maintained through the phase-locking of neurons in the auditory passage [5], [6].

However, when listening to a melody or a chord, humans perceive much more than just pitches. Listeners also encode how pitches of successive notes relate to each other - whether the previous note is higher or lower than the current note, and perhaps even by how much [6]. Relative pitch is intrinsic to how humans perceive music, and plays a key role is why humans can recognize a familiar melody even when all notes are shifted upwards or downwards in pitch.

One of the most important aspects of relative pitch is the direction of change from one note to the next, also known as a *contour*. Most people, regardless of their musical expertise, are good at encoding the contour of a novel series of notes; hence, humans ability to recognize simple melodies. However, in contrast to the majority competence with contours, humans tend to be severely less accurate at recognizing whether precise pitch intervals separating the notes of an unfamiliar melody are preserved in a transposition. For example, if listeners are played a novel random chord, followed by a second chord that is shifted to a different pitch range, majority of the time the listeners are unable to tell if the intervals between the notes have been changed, so long as the contour within the chords do not change. This is particularly true in listeners who do not have musical training [5].

Interestingly enough, when we remain in the same pitch domain, most instances of interval alterations that listeners can identify involve violations of tonal structure. Violation of tonal structure means that the interval alternation introduced a note that does not match the pitch set that the listener expects. Thus, a ”major interval” shifted to a ”diminished interval” is highly recognizable. But interval changes that substitute another note within the same musical scale, are often not noticed [6].

McDermott et al. concludes that relative pitch recognition for chords are not generally retained with much accuracy, but instead are readily incorporated into the tonal pitch structures that listeners learn via active exposure.

3.3. Chords and Patterns

For our research, it is important to be familiar with a few simple chords and patterns. A chord essentially consists of two or more notes played simultaneously that are certain interval apart with respect to the root pitch.

3.3.1. Triads. The most simple chord is a *major* triad. It consists of the root, the major third, and a perfect fifth. The most common, and most recognizable major triad is the C major chord. Next is the *minor* triad, which is similar to a major triad, but with the third lowered by a half step. Lesser known triads, but still important, are the *augmented* and *diminished* triads. Augmented triads are the same as a major triad, but with an augmented fifth, i.e a sharp five. Diminished triads are similar to minor triads, but with a diminished fifth, i.e. a flat five.

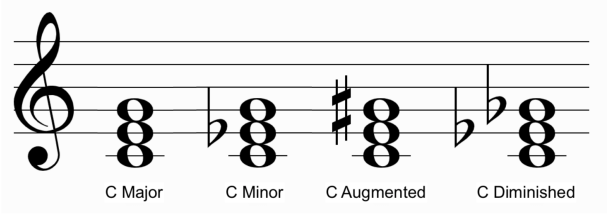

Figure 1. Four common triads in C

3.3.2. Sevenths. The next level of chords that we need to know are *seventh* chords. A seventh chord is a triad with one more note, which is either a *major seventh* above the root of a major chord, a *minor seventh* (flattened seventh) above the root of a minor chord , a *dominant seventh* (flattened seventh) above the root of a major chord, and lastly an *augmented seventh* (flattened seventh) above the root of an augmented third chord. There are several other seventh chords, however the listed four are the most common sevenths in musical theory.

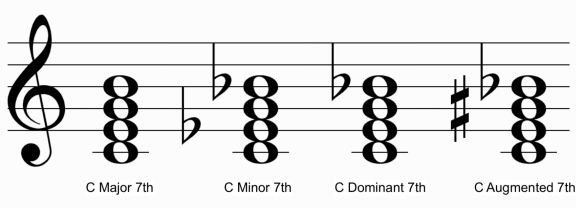

Figure 2. Four common sevenths in C

3.3.3. Inversions. Now that we have established some basic knowledge of chords, it is necessary to understand inversions. To invert a chord we simply rearrange its notes so that the original root note becomes and upper note. In the case of our triads, we can invert twice from its root position:

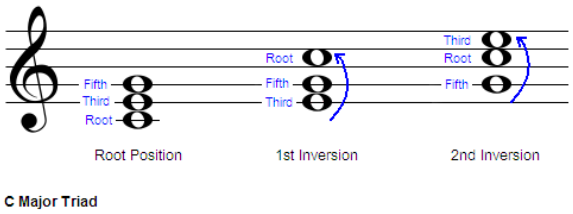

Figure 3. Major C triad inversion example

Knowledge of triads, sevenths, and inversions will be key in our tests.

3.4. Music as a Signal

This section seeks to give the necessary background information to understand how we work with audio data in our study.

3.4.1. Spectral Analysis. Audio and music are generally recorded as sound files such as ”.wav” files. The audio is represented as an array of floating point numbers, where each value represents a scaled voltage to be applied to a speaker, in order to vibrate the air a certain amount, and ”play” a song. This views audio in the time domain. It’s possible to view audio in the frequency domain, by using spectral analysis.

In general, the Fourier Transform $\mathscr{F}$ takes a time-domain signal and projects it onto an orthonormal basis of sine and cosine waves, making it possible to view the frequencies that comprise a signal as a ”spectrum.”

If we wish to see how the musical pitches in an audio piece change over time, we can perform a Short Time

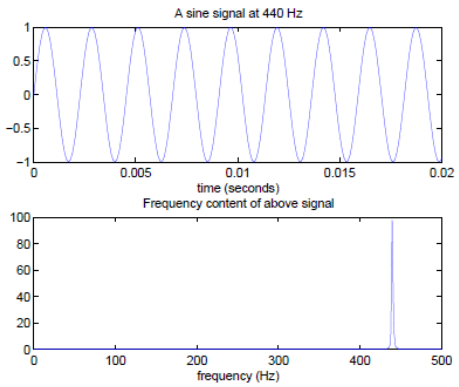

Figure 4. The Fourier transform of a sine signal, tuned to 440Hz, which is concert musical pitch "A"

Fourier Transform (STFT) to create a Spectrogram – a description of all the frequencies in the audio across time.

3.4.2. Chromagrams. Taking Spectrograms one step further, we can estimate not only the frequencies over time, but specifically *musical pitches over time*. This is known as a Chromagram. In a chromagram, we examine the spectrogram, and map every frequency to a "bin" – representing each of the 12 western musical notes. In this way, we are able to get a picture of how much energy there is in each of the musical note "bins" over time.

The first figure below plots the chromagram of one of our synthesized guesses of a C major chord. This synthesizes an audio signal from the pure piano notes, and creates a chromagram over time. The 3 yellow streaks across time represent the 3 pure piano notes in the chord. The other streaks which aren't as bright represent the harmonic frequencies of each of the fundamental pure notes, and are responsible for giving the notes a "piano sound," but also making it slightly more confusing to estimate which notes were the pure pitches actually played.

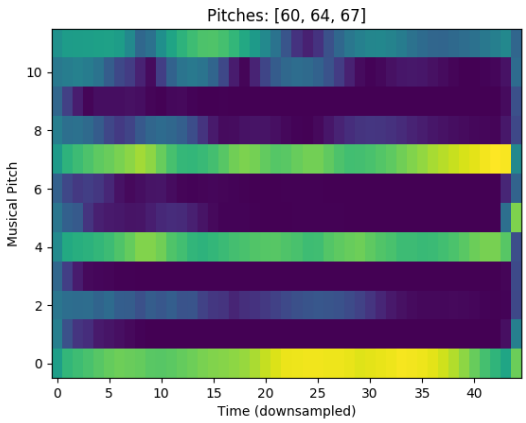

Figure 5. This is the normalized chromagram of a pure C major chord on the piano.

The second figure plots the chromagram of one of our tests we showed the participants. We can see that it is the same 3 piano notes forming a C major chord, but the background noise distorts the spectrum, and makes it harder to tell exactly which notes went into the audio. This explains why it may be difficult to separate out the individual sources from a discriminative approach (training a model to predict the notes being played). Instead, we can use a generative approach, as in MCMC, where we can synthesize audio and compare its similarity to the ground truth audio. We can do this using a raw energy comparison.

We define the energy spectra E of a chromagram to be the sum of the values (energy) in each of the frequency bins across time. In our case, this will yield a length 12 vector (for 12 musical notes), which we can later use as part of our similarity comparisons between chromagrams.

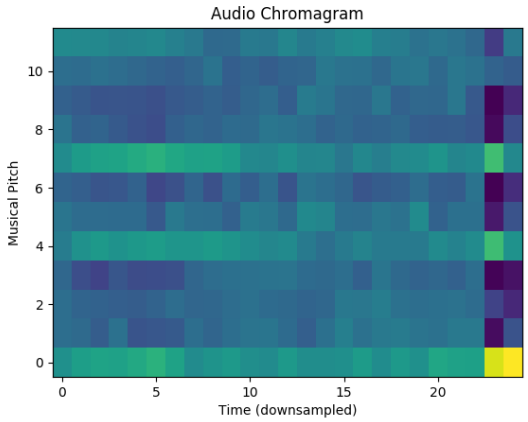

Figure 6. This is the normalized chromagram of a C major chord in one of our tests, which shows the background noise interfering with the true notes played.

4. Methodology

4.1. Data Collection

Much like *Bourgin et al.*'s work, we wanted to collect data of humans attempting to test their own relative pitch abilities. In order to stream line our data collection we decided to build an interactive UI, where test subjects would listen to pitches and chords that we have selected, and interact with a keyboard to identify the given pitch/chord.

`http://mitcompcogsci.pythonanywhere.com/`

4.1.1. UI. The UI was built on a Flask back-end, and runs on a simple JavaScript. There is a "Play Me" button, that plays the test pitch/chord that we have preselected. The test sound is a mixed track that is essentially singular tracks of notes and noise to simulate a simpler cocktail party problem [7]. Additionally, we built an interactive piano that plays back the notes that a user plays on her keyboard. The white piano notes map to keys 'A' to 'L,' and black piano notes map to keys 'Q' to 'P.' As the user plays keys and attempts to identify our given pitches/chords, we record all of their

attempts. Moreover, we also record each time the test subject presses the ”Play Me” button in their attempts.

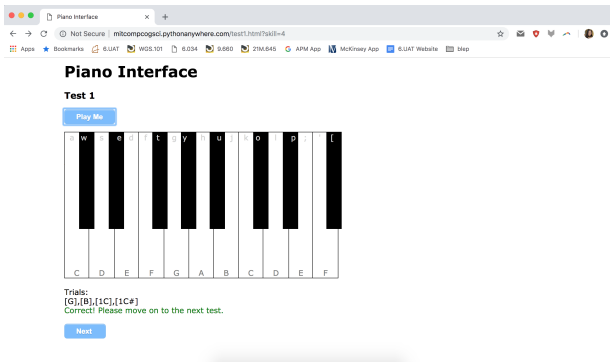

Figure 7. Piano UI with interactive keyboard

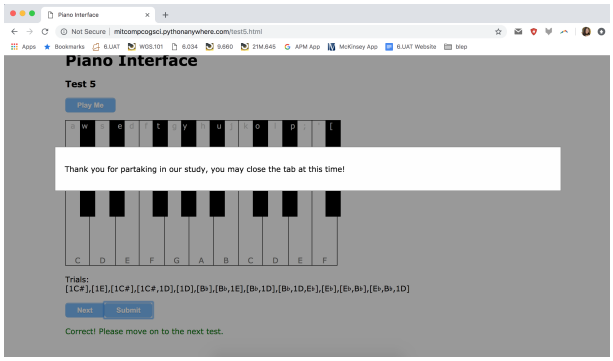

Figure 8. End of Piano UI that submits the test subject's trials

Our UI is broken down into six distinct pages. The first being a short instructions guide and an assessment of the number of years in musical training. The latter five pages are tests with various pitches and chords chosen in increasing difficulty.

4.1.2. Years of Musical Training. We split the years of musical training into four partitions, 0-1 years, 1-4 years, 4-7 years, and 7+ years. According to *McDermott et al.* and various other studies, strong relative pitch can be obtained with 7+ years of regular training. Of course, it must be acknowledged that strong relative pitch can be gained in less than 7 years with rigorous training, but given our pool of test subjects, we assume that the majority of our test subjects underwent regular musical training during their adolescence. Instrumental and vocal training in the first 1-4 years cover simple chords and interval recognition, and years afterwards it is constant training of the ear until one becomes incredibly strong with their relative pitch [6].

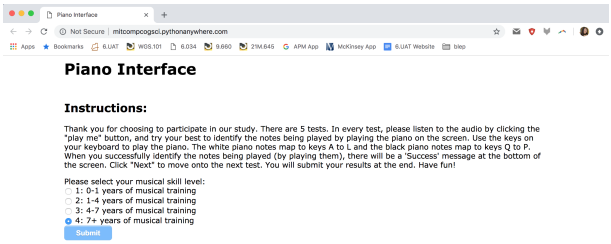

Figure 9. First page of our interactive UI assessing test subject's musical ability

4.1.3. Chords Chosen. We have five tests that increase in difficulty. We chose a simple pitch for the first test, and recognizable thirds for the second and third test - no inversions, or missing notes to confuse our test subjects. However, test four and five have a jump in difficulty to separate test subjects with strong relative pitch, and weak relative pitch. We chose seventh chords, that were missing a fifth or a third - notes that are key to identifying chords. Additionally, we inverted the fourth test to cause more confusion and truly differentiate relative pitch skills amongst our test subjects.

Test	Answer	Type
Test 1	1C#	Pitch
Test 2	F G#	F Minor Third
Test 3	C E G	C Major Triad
Test 4	E B♭ C	Inv. C Dominant 7th missing 5th
Test 5	E♭ B♭ 1D	E♭ Major 7th missing 3rd

4.2. MCMC Model

4.2.1. Problem Formulation. The following procedure is a detailed version of how we use MCMC to generate samples from the distribution of interest, $P(N|D)$. ”State” in this context represents a multi-hot-encoded vector of length 20 (20 notes) where the value is 1 if the note is on and 0 if the note is off.

1) Initialize all notes to be off
2) Propose a new state that toggles 1 note from the current state. Propose based on a proposal function with ”tunable confidence” that scores the similarity of the current state and proposed state.
3) Score the current and proposed states against the audio using a ”tunable similarity function” and accept the proposed state according to Metropolis-Hastings acceptance rule.
4) Iterate until number of iterations specified.

4.2.2. Similarity Function. The role of our similarity function is to score whether our proposal is a more likely ”generator” of the audio data we're trying to estimate, or our current state is the more likely ”generator”. We seek to find a function that is directly proportional to $P(D|N)$.

We approximate this probability by the following scoring method, for a given state S (of "on and off notes") against an audio signal A:

1) Render state S to an actual audio signal (play the notes that are toggled on).
2) Compute the Chromagrams of S and A
3) Build energy vectors E_S, $E_A \in \Re^{12}$ by summing the frequency values across time.
4) Set E_S, $E_A \leftarrow E_S$ - mean(E_S), E_A - mean(E_A)
5) Divide E_S, E_A by their corresponding L2-norms.
6) R $\leftarrow E_S \cdot E_A$ (Return their dot products)

Now that we have a raw score value between a given state S and audio A, we need to convert this into a probability of picking the current state, or the proposed state. To do this, we simply take the probabilities of selecting each as:

$$softmax(E_S, E_A; \beta)$$

with tunable "inverse temperature" β, which scales/shrinks their score. The softmax allows us to scale the scores into a probability $1 = P(D = A|N = S) + P(D = A|N = S')$ where S' is the proposed state, and S is the current state.

4.2.3. Proposal Function. We want to create a proposal function that suggests a new state to score, that moves to a place in "state-space" (which has 2^{20} possible options) that is similar to the current state. We cannot necessarily use a Gaussian distribution over close-by notes, because physical "similarity" doesn't necessarily correspond to musical similarity.

We take a hint from the Similarity function, and propose a new state by forming a distribution over all possible 1-note-changes and weighting them according to their similarity scores as mentioned above.

4.2.4. Tuning Parameters. We can thus view these tunable parameters as a way to adjust the "sensitivity" of such similarity and proposal measures. A higher value of β_{sim} implies we more strongly believe our similarity function, and the model is much more confident in its belief that one state is more similar to the audio than the other. In the same fashion, increasing $\beta_{proposal}$ indicates that the model is more likely to select similarly-sounding examples to try.

In this way, we have now created parameters which can mimic a variety of musical abilities. Higher values for both of these parameters correspond to, in different ways, a more-trained musical ability. Better musicians will know which notes to try next to match the pitch, and will also have a better idea if they're getting closer or further away from what the audio sounds like.

5. Analysis of Results

After tuning our model to fit the four different levels of musical experience, we looked deeper into how our MCMC model did individually against our test subjects.

5.1. Fitting MCMC to the Data

In an attempt to illustrate how we could actually fit the model to the data, we attempt to find tunable parameters β_{sim} and $\beta_{proposal}$ such that we fit different skill levels across all tests. For participants at a given musical level, we compute the histogram of notes played for each of the tests. We normalize it to turn it into a distribution, and then try to find parameters which minimize the KL-divergence of our MCMC model's steady-state distribution and the experimental note distribution over all the tests for a given level. We were able to fit parameters reasonably well, but due to the grid-search nature of our approach, and some of the noise in the data, we weren't able to extract as meaningful model parameters as we would like.

However, upon manual experimentation, we found that our results were consistent with our intuition: increasing both of the sensitivities did indeed increase the final accuracy of our MCMC model, and yield a faster convergence – illustrating the relationship between our tunable parameters, and musical ability.

5.2. Individual Analyses

At the individual level we compare our tuned MCMC model against a particular test subject and examine the choices made by both model and human test subject via chromagrams.

In particular, we examined Test 4, one of our more difficult tests, and observed how a human subject and our tuned MCMC model compared via chromagrams. Each of the human generated chromagrams tracks the subject's various trials until they discovered the right answer in yellow. Similarly, we graphed the tuned MCMC's trials as well in blue, and observed whether or not our model converged to the correct answer in yellow (at the very end of the chromagrams).

Below, we first observed a test subject who identified to have a musical experience level of 0-1 years, and the tuned MCMC model matched to that level of experience.

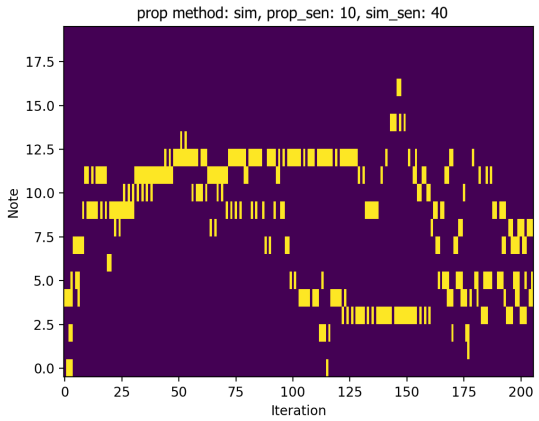

Figure 10. Test 4 Chromagram: Test subject with musical experience level of 0-1 years

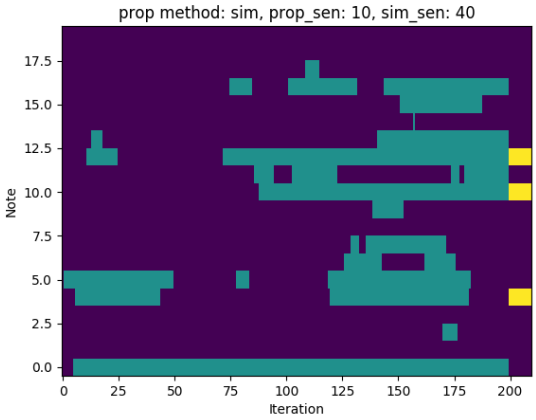

Figure 11. Test 4 Chromagram: MCMC Model tuned to musical experience level of 0-1 years

It is clear that the decisions between both the human and the MCMC model are sporadic, ranging across various notes on the keyboard. The sporadic changes in notes being tried from iteration to iteration is indicative of a lack of confidence, which in turn is a result of a lack of experience. For example, we see a bit of gradient ascent search happening in both chromagrams, however more so in the test subject's chromagram. This search is suggestive towards a lack of knowledge and understanding of chord make-up, and thus we observe a 'brute-force' like search in both graphs. Furthermore, despite differences in the iteration step, we see clustering around the same notes, particularly the higher ones such as *D, C* and $B\flat$. Higher notes are easily identifiable due to their frequencies [5]; hence both models show a liking towards those particular notes, but have severe difficulty locating the other two lower frequency notes.

Contrastingly, we observe Test 4 again but with a test subject that identified with a longer musical experience of 7+ years instead. Below, we display the chromagrams for both the human test subject, and the tuned MCMC model, respectively:

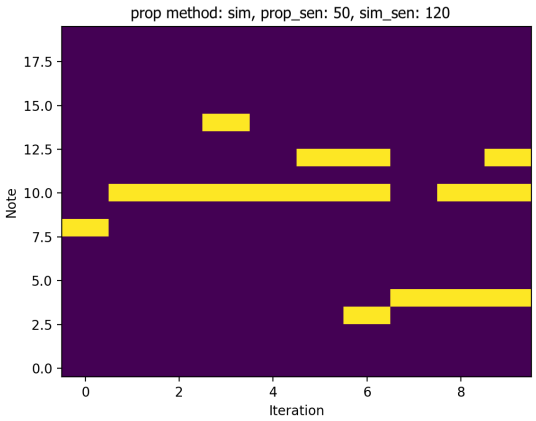

Figure 12. Test 4 Chromagram: Test subject with musical experience level of 7+ years

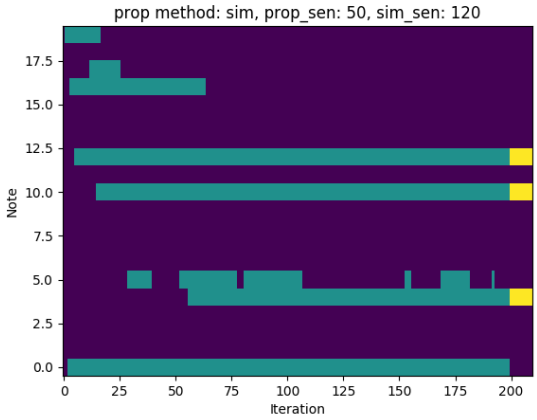

Figure 13. Test 4 Chromagram: MCMC Model tuned to musical experience level of 7+ years

Immediately, there is a stark contrast in simply the number of different notes tried. Instead of a hectic change of notes between iterations, we observe a continuity of notes over longer iterations. It is clear that there is a higher confidence in the selection of notes earlier on in the iterations for both our test subject, as well as our model, because they both realized that $B\flat$ was a part of the given chord and kept that constant for the majority of their iterations. This level of continuity, as well as quick recognition is highly representative of strong relative pitch.

In addition, not only is there stronger confidence in the selection of notes, but also there is knowledge of intervals present in these chromagrams as well. In the two previous chromagrams with 0-1 years of musical experience, we saw that both the model and the human were doing a type of gradient ascent/brute-force to search for the correct chord. However, we see that the test subject's trials as well as that of the tuned MCMC model's is far more directed – conveying a knowledge of intervals.

5.3. Probability Distribution

To ensure that our model correctly fits the human data, we plot the steady-state probability distribution of the MCMC model, alongside expert and beginner humans. The plot of the human data, is over all possible attempts, while the MCMC model is only after convergence. Thus, we expect some discrepancy between the two, as we cannot estimate "burn-in" time for human trials.

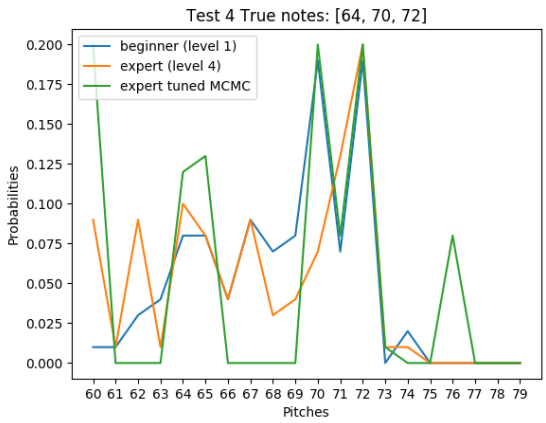

Figure 14. Probability distributions on a single test, with different skill levels, and a well-trained MCMC model

We see that the MCMC model fits the likelihood distributions relatively well, and agrees strongly with the expert, more than the beginner.

6. Limitations

While we have made some very interesting findings for human relative pitch identification and audio perception as Markov Chain Monte Carlo, there are few limitations in our work.

One of the largest limitations is in our choice of model. We decided to use Markov Chain Monte Carlo with Metropolis-Hastings, and Metropolis-Hastings requires burn-in. This is a limitation because we did not a define a prior initialization which people naturally have when it comes to choosing which note to press first after initially hearing a pitch/chord. Hence, in order to compensate for this lack of prior, we had to introduce burn-in.

Secondly, our collected data was quite noisy. After following up with a few of our test subjects, we realized there was a nuanced discontinuity that we did not take into consideration. Many of our test subjects had several years of experience in music *back in secondary school*, and have not touched an instrument regularly since coming to college. Thus, despite stating they had 4-7 or 7+ years of musical training, only a subset of these test subjects actually still play instruments and/or sing regularly. This is a severe limitation in our study as the confidence and level of understanding that our model generates may not correlate with our human subjects' data as much as we originally hypothesized to.

Lastly, some human test subjects exhibited systematic trial and error, i.e. enumeration of all possible keys. MCMC with Metropolis-Hastings is not well equipped to handle scenarios such as these, and would likely be better modeled by Hamiltonian Monte Carlo [4].

7. Conclusion

In conclusion, much was achieved in this exploration. First off, we successfully defined a model and tuned its parameters such that the MCMC model correlated with human musical skill talent, expressing weak confidence and lack of musical knowledge for humans with low musical knowledge exposure, and high confidence and strong musical knowledge for humans with many, many years of training. The ability to tune the "sensitivity" of the similarity and proposals measures, allowed us to take a deeper look into decision making in the realm of audio perception.

Subsequently, we successfully displayed that a tuned MCMC model converges very similarly to humans of various musical experience levels. Our observations of the chromograms between human test subject and model, at the same musical experience level, was not only similar, but also very insightful into the work that has already been done. We were able to draw connections and conclusions between human cognitive science, human audio perception, musical training, and computer science.

Last, but not least, we developed a novel way for music note(s) identification in mixed audio that has not been proposed before in the literature. We hope that our study inspires others to explore the computational cognitive science and audio perception.

8. Individual Contributions

8.1. Josh Hellerstein

I Implemented the MCMC algorithm, Keyboard synthesizer, audio manipulations and transforms (chromagrams etc...), as well as created the algorithms for similarity functions and proposal distribution functions. I also worked on getting the experimentation pipeline running and logging human trials on a server (back-end, fetching results, etc...). Further I worked on analysis and the writeup, including plotting distributions, visualizations, interpretation of results and formalization of math terms.

8.2. Christie Hong

I worked on several different components of the project. In the beginning, I analyzed several different research papers ranging across subjects of psychology and biology of audio perception, musical theory, applications of Markov Chain Monte Carlo in computational cognitive science, as well as the various computer science AI/ML works in audio perception. Josh and I worked together in firstly, understanding the psychology, biology, and music theory entwined in this problem, then we moved to piecing together the strengths of each computational paper into a new study for relative pitch. With a game plan in place, I worked on our UI: enabling user interaction with a live keyboard, efficiently collecting data real time, and storing our data for future analysis. We both worked on the analysis of data - I primarily focused on the analysis at the individual level. I compared and analyzed chromagrams for all test subjects across each of the musical skill levels, and drew insights from this analysis, specifically in-line with the research done previously. Lastly, we both worked together on writing up the final paper.

References

[1] Bee, Mark A. and Micheyl, Christophe. *The Cocktail Party Problem: What Is It? How Can It Be Solved? And Why Should Animal Behaviorists Study It?* J Comp Psychol. 2008 Aug; 122(3): 235251.

[2] Bourgin, David D. and Abbott, Joshua T. Abbott and Griffiths, Thomas L. and Smith, Kevin A. and Vul, Edward. *Empirical Evidence for Markov Chain Monte Carlo in Memory Search* 36th Annual Conference of the Cognitive Science Society. 2014.

[3] Chen, Tianqi and Fox, Emily and Guestrin, Carlos. *Stochastic gradient hamiltonian monte carlo*. International Conference on Machine Learning. 1683-1691. 2014

[4] Chib, Siddhartha and Greenberg, Edward. *Understanding the metropolis-hastings algorithm*. The american statistician.1995.

[5] Lotto, Andrew and Holt, Lori. *Psychology of auditory perception* John Wiley & Sons, Ltd. 2010.

[6] McDermott, Josh H. and Oxenham, Andrew J. *Music perception, pitch and the auditory system* Curr Open Neurobiol. 2009.

[7] Simpson, Andrew J.R. and Roma, Gerard and Plumbley, Mark D. . *Deep Karaoke: Extracting Vocals from Musical Mixtures Using a Convolutional Deep Neural Network*. 2015.

[8] Yildirim, Ilker and Kulkarni, Tejas D. and Freiwald, Winrich A. and Tenenbaum, Joshua B. *Efficient anylsis-by-synethsis in vision: A computational framework, behavioral tests, and comparison with neural representations* 2015.