LEARNING TO CODE THROUGH MIR

Anna Xambó[1,2] **Alexander Lerch**[1] **Jason Freeman**[1]

[1] Center for Music Technology, Georgia Institute of Technology, Atlanta, GA, USA
[2] Digital Media Program, Georgia Institute of Technology, Atlanta, GA, USA
{anna.xambo,alexander.lerch,jason.freeman}@gatech.edu

ABSTRACT

An approach to teaching computer science (CS) in high-schools is using EarSketch, a free online tool for teaching CS concepts while making music. In this demonstration we present the potential of teaching music information retrieval (MIR) concepts using EarSketch. The aim is twofold: to discuss the benefits of introducing MIR concepts in the classroom and to shed light on how MIR concepts can be gently introduced in a CS curriculum. We conclude by identifying the advantages of teaching MIR in the classroom and pointing to future directions for research.

1. INTRODUCTION

Teaching how to code by combining CS with making music is becoming an established practice in schools, using technologies such as EarSketch [3], Sonic Pi [1], or Scratch [6]. Bringing the arts into science, technology, engineering, and math (STEM) education, also known as STEAM, embraces artistic thinking in real-world project-based learning [5]. This approach promotes broadening participation of under-represented populations into computing, such as women, African-Americans, and Latinos [2, 3]. Using music to teach CS has proven to be successful, and integrating MIR approaches in the curriculum could be the next step towards learning STEM concepts by combining music and CS.

The discipline of MIR is highly influential to the development of cutting-edge music technology. However, with a few exceptions [4], MIR concepts are hardly taught in the high school classroom because of their level of complexity and abstraction. In this paper, we propose a pedagogical approach to introducing basic machine learning (ML) concepts through MIR. We use the EarSketch platform as a proof of concept, yet other music and audio programming environments would be suitable as well.

2. EARSKETCH

EarSketch [3] is a browser-based Digital Audio Workstation (DAW) environment and provides a curriculum for creating music through code combining audio clips. The project started in 2011 with an offline version and it was ported to the browser in 2013. At present there are 4,000 audio clips available; furthermore, students can upload their own sounds. The diversity of audio clips, which are tagged by genre and instruments, allows for a range of music genres, such as *hip hop*, *dubstep*, and *techno*. The EarSketch curriculum aims to teach both CS and music concepts by encouraging students to write code in two modern programming languages, Python and JavaScript. EarSketch has been successfully deployed in a number of high schools, college courses, and summer camps [3].

3. EARSKETCH AND MIR

The current EarSketch curriculum includes a chapter on teaching computers to listen, in which the field of MIR is introduced and analysis features are explained. MIR is used both to introduce programming concepts and to illustrate connections between code and commercial applications such as music recommendation. The initial release supported 12 analysis features (e.g., flux, flatness, crest, kurtosis), however, they were hardly used. Therefore, the number of features was reduced to 2 for the current EarSketch version: *RMS* as a measure of the average volume, and *spectral centroid* as an estimate of the "center frequency." A set of exercises combine these features with algorithmic composition techniques, inspired by MEAPsoft [8]. For example, in "Sorting and Analysis", analysis features are used for sorting: list items are sorted based on their either spectral centroids or their RMS values. Students thus learn in parallel about sorting algorithms and concatenative synthesis by, e.g., making compositional decisions based on feature variation or similarity. In a different example, feature analysis is applied to a collection of audio clips. This is based on a certain threshold or a comparative conditional expression, thus creating (1) custom effects like a noise gate (RMS), or effects like a thresholded equalizer based on the spectral centroid, or (2) mappings from the feature values to effects, e.g., for an auto-volume function. Thus, students also learn concepts such as conditionals and mapping. These exercises are helpful to get a sense of MIR from the low-level perspective of feature extraction, as opposed

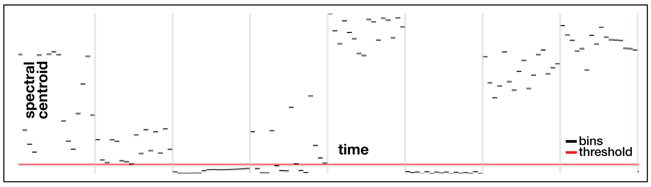

Figure 1. Timeline that shows spectral centroid values (bins of 1/16 of a measure) from a sequence of audio clips (electrical bass vs. electrical guitar) as well as a binary threshold value. Each gray vertical bar represents a measure.

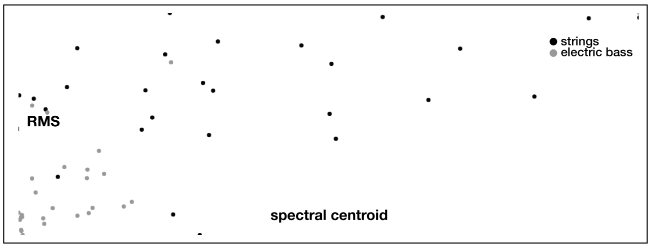

Figure 2. Scatter plot of tagged audio clips based on two analysis features.

to higher-level ML techniques. Yet, the content is rarely taught because it is above the high school level needs.

We argue that for a clearer understanding of important but complex topics at a high school level, interactivity and visualization are essential, as pointed out in [7]. Arguably, visualizing the process can provide students with contextual information of what is happening beyond the auditory real-time feedback of the above exercises. Next, we present two exercises based on visualizing MIR information, which can be helpful for teaching ML concepts.

4. A PEDAGOGICAL APPROACH TO MIR

We propose two examples of exercises for EarSketch that are complementary and contribute to the understanding of well-known MIR and ML concepts: (1) a one dimensional binary classifier implemented by simple thresholding, and (2) a two dimensional feature space visualization of different groups of sounds organized by the type of the sound source.

4.1 A binary thresholding classifier

The value of one feature is plotted over time and the student attempts — by adjusting a threshold — to differentiate between two signal types (e.g., speech vs. music, rhythm vs. soundscape, or two different instruments) that are randomly sequenced in a track. Alternatively, the student can use an automatically computed threshold. The learning outcomes include: (1) basic knowledge of signal properties and how they can be modeled by simple features extracted block-by-block, (2) understanding the concept of a simple binary classifier, and (3) familiarity with the concept of systematic evaluation by computing the accuracy. The timeline visualization and the threshold are shown in Fig. 1 and the code is available online. [1]

4.2 Feature space visualization

The audio clips of two different contrasting audio categories are represented by two-dimensional feature vectors and visualized in a scatter plot. The learning outcomes include: (1) understanding the concept of dimensionality reduction by feature aggregation, (2) understanding of scatter plots and general concepts of data visualization, and (3) basic understanding of feature space, distances, and multi-dimensionality. The implementation of the code is adapted from Matlab code available online. [2] An example of a scatter plot is shown in Fig. 2. The corresponding code is available online. [3]

5. CONCLUSION AND FUTURE WORK

This paper contributed a pedagogical approach to teaching CS and music concepts using MIR on the EarSketch platform. We believe that adding these and similar exercises to the existing EarSketch curriculum will help introduce higher level CS concepts to the students. Choosing MIR as a field of interest is, on the one hand, obvious, given the music-related curriculum EarSketch builds on. Yet, on the other hand, it is of genuine interest to students who use tools such as Spotify and Pandora on a daily basis to gain some insight into the basic concepts of music similarity and recommendation systems.

An important part of the exercises will be the discussion of the process and the concept of feature extraction, representation, and visualization. For example, the analysis features are simpler to explain by relating them to both the waveform and the spectrogram view, two plots that are already used in the EarSketch curriculum.

Future work includes exploring the pedagogical benefit of introducing different types of MIR tasks, e.g., a mood or emotion classifier or the identification of other musical properties such as keys or chord progressions. This could indicate the compatibility of two audio clips when mixed together. The investigation of real-time music analysis and how it can be applied to live coding practices in the classroom is another interesting direction. The combination of MIR and algorithmic composition techniques might also lead to a computational toolset for music composition and performance.

6. ACKNOWLEDGEMENTS

EarSketch receives funding from the National Science Foundation (CNS#1138469, DRL #1417835, DUE #1504293), the Scott Hudgens Family Foundation, the Arthur M. Blank Family Foundation, and the Google Inc. Fund of Tides Foundation.

[1] `http://bit.ly/2assHAZ` (accessed August 3, 2016).

[2] `https://github.com/alexanderlerch/ACA-Slides/blob/master/matlab/displayScatter.m` (accessed July 6, 2016).

[3] `http://bit.ly/2astBNW` (accessed August 3, 2016).

7. REFERENCES

[1] Samuel Aaron and Alan F. Blackwell. From Sonic Pi to Overtone: creative musical experiences with domain-specific and functional languages. In *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design*, pages 35–46, New York, USA, September 2013.

[2] Jamika D. Burge and Tiki L. Suarez. Preliminary analysis of factors affecting women and African Americans in the computing sciences. In Pamela J. Williams and Mark A. Friedman, editors, *Proceedings of the 2005 Conference on Diversity in Computing*, pages 53–56, New York, USA, October 2005. ACM.

[3] Jason Freeman, Brian Magerko, Tom McKlin, Mike Reilly, Justin Permar, Cameron Summers, and Eric Fruchter. Engaging underrepresented groups in high school introductory computing through computational remixing with EarSketch. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 85–90, 2014.

[4] Youngmoo E. Kim, Alyssa M. Batula, Raymond Migneco, Patrick Richardson, Brian Dolhansky, David Grunberg, Brandon Morton, Matthew Prockup, Erik M. Schmidt, and Jeffrey Scott. Teaching STEM concepts through music technology and DSP. In *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE*, pages 220–225, 2011.

[5] John Maeda. STEM+ Art= STEAM. *The STEAM Journal*, 1(1):34, 2013.

[6] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

[7] Bret Victor. Learnable programming: designing a programming system for understanding programs. Web resource, accessed August 4, 2016, from http://worrydream.com/LearnableProgramming, September, 2012.

[8] Ron Weiss, Douglas Repetto, Mike Mandel, Dan Ellis, Victor Adan, and Jeff Snyder. Meapsoft: A program for rearranging music audio recordings. Web resource, accessed August 4, 2016, from http://www.meapsoft.org, 2006.