View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Goldsmiths Research Online

BeatBox: End-User Interactive Definition and Training of Recognizers for Percussive Vocalizations

Kyle Hipke[1], Michael Toomim[1], Rebecca Fiebrink[1,2], James Fogarty[1]

[1]Computer Science & Engineering
DUB Group, University of Washington
Seattle, WA 98195
{kwhipke1, toomim, jfogarty}@cs.washington.edu

[2]Goldsmiths University of London
Department of Computing
London, SE14 6NW, UK
r.fiebrink@gold.ac.uk

ABSTRACT

Interactive end-user training of machine learning systems has received significant attention as a tool for personalizing recognizers. However, most research limits end users to training a fixed set of application-defined concepts. This paper considers additional challenges that arise in end-user support for defining the number and nature of concepts that a system must learn to recognize. We develop BeatBox, a new system that enables end-user creation of custom beatbox recognizers and interactive adaptation of recognizers to an end user's technique, environment, and musical goals. BeatBox proposes rapid end-user exploration of variations in the number and nature of learned concepts, and provides end users with feedback on the reliability of recognizers learned for different potential combinations of percussive vocalizations. In a preliminary evaluation, we observed that end users were able to quickly create usable classifiers, that they explored different combinations of concepts to test alternative vocalizations and to refine classifiers for new musical contexts, and that learnability feedback was often helpful in alerting them to potential difficulties with a desired learning concept.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

Interactive machine learning, intelligent user interfaces, music.

1. INTRODUCTION

Machine learning offers powerful tools for interactive systems, especially systems that must understand and respond to complex input (e.g., gestures, images, sound). Traditional methods require designers write heuristic code specifying how to recognize input. Machine learning instead allows providing *examples* labeled with a desired *class*. This training data is provided to a supervised learning algorithm that learns a *classifier* to map input to class labels. The quality of training data provided to the learning algorithm directly impacts the reliability of the resulting classifier. System designers therefore carefully collect and curate large amounts of training data, creating one-size-fits-all recognizers that can used out-of-the-box by all end users of a system.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Request permissions from permissions@acm.org.
AVI' 14, May 27 - 29 2014, Como, Italy
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2775-6/14/05…$15.00.
http://dx.doi.org/10.1145/2598153.2598189

In contrast to designer creation of one-size-fits-all recognizers, recent research examines end-user control of the machine learning process. This allows interactive end-user creation of recognizers personalized or even tailored to the current task. For example, an end user might train a system to identify images related to a current concept of interest [5], apply customized qualitative codes in transcripts [10], automatically classify incoming email [16], or synthesize sound in response to a musician's gestures [4]. Due to the individualized nature of recognition in these types of applications, there is generally not pre-existing data available to train a classifier. Because high-quality training data is critical to obtaining a reliable classifier, prior research has explored end-user support for iteratively improving a classifier by adding to or editing provided training data [1,4,5].

Many compelling applications go beyond end-user *training* of a recognizer to also require end-user *definition* of the number and nature of classes to be learned. This paper considers interactive creation of beatbox recognizers, wherein an end user both defines a set of musical vocalizations and also trains a recognizer to identify those vocalizations. In this context, an end user may want to add a new class of vocalization, but only if it is unlikely to be confused with existing vocalizations. Alternatively, an end user may want to change the definition of a vocalization to make it more distinct from other vocalizations. End users may even want to temporarily disable recognition of an unneeded vocalization, perhaps to allow swapping in a vocalization that might otherwise be easily confused with the removed vocalization.

Beyond our current focus on beatbox recognition, many new dimensions of exploration and decision are introduced in any application where there is no pre-defined set of classes for which an end user can simply provide training examples. In such applications, end users must work to define a set of classes that are both: (1) suitable for the intended usage context, and (2) reliably differentiated by a learning algorithm trained with the end user's provided examples. These goals require mechanisms that enable not only experimentation with training data, but also with the number and nature of input classes to support efficient discovery of how changes in class definition are likely to impact the reliability and usefulness of a recognizer. We explore such mechanisms for flexible end-user definition and exploration of learning concepts in a new system for beatbox musical interaction.

The specific contributions of this paper include:

- Motivation and implementation of end-user interactive machine learning techniques that provide feedback on the learnability of a defined set of classes and support for dynamically exploring addition, removal, and redefinition of desired classes.
- Initial evidence of musical relevance of these techniques, demonstrated in a system for personalized beatbox recognition.
- Preliminary insight from a study of how end users employed learnability feedback, exploratory concept definition, and the beatbox recognition system as a whole.

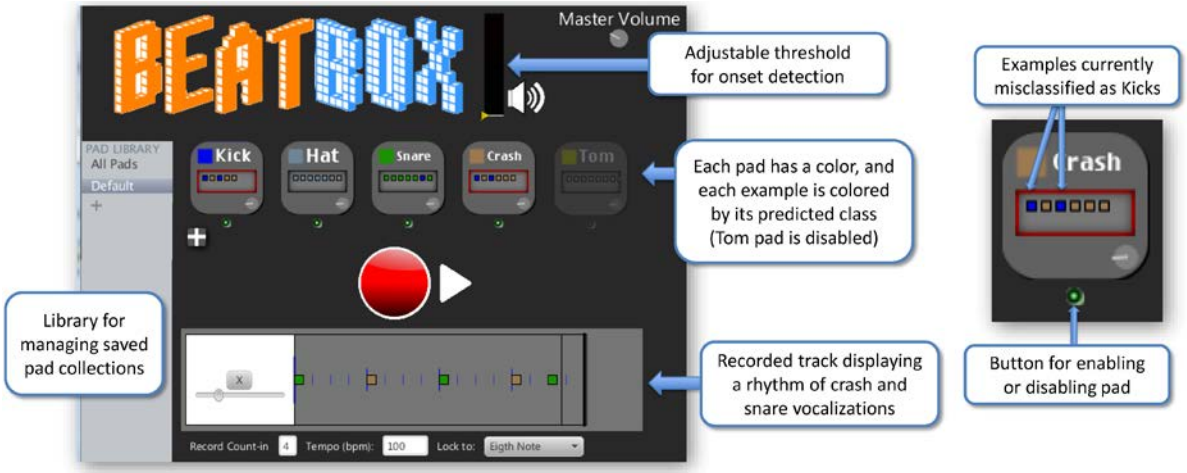

Figure 1. BeatBox enables end-user definition and training of custom beatbox recognizers. The top center workspace displays end-user defined pads, each containing training examples. Each example is colored to match to the pad to which it is most similar, making it easy to identify problematic examples or pads. A button under each pad enables or disables the pad, and pads can also be added, deleted, saved, or retrieved from the left Pad Library. The bottom of the workspace displays created tracks that can be edited or played back. New tracks can be recorded at any time, with each vocalization recognized as one of the enabled pads.

2. MOTIVATION OF BEATBOX

Musicians in many cultural traditions have long used percussive vocalization in performance and teaching, including musicians in hip hop, Indian classical percussion, and Australian "didjeridu talk" [3]. Professional musicians such as Michael Jackson have also used beatboxing as a way to "sketch" musical ideas that are later realized with other instruments [17]. Computer recognition of beatbox sounds has been used to query music databases [8], generate music notation [13], and control synthesizers in live performance [15].

Inspired by these practices, BeatBox is a new system we created to allow musicians to use beatboxing as a hardware-free input method that leverages their rhythmic and vocal expertise. The software automatically recognizes classes of percussive vocalization that have been defined and trained by the end user. Beatboxed input can be manipulated and layered in a track editor, and it can be played with the end user's vocalizations replaced with percussion samples or other sounds.

Prior work demonstrates feasibility of beatbox recognition for small numbers of fixed vocalization classes (e.g., two to five) [7,8,13,14,15]. Such work demonstrates reliable recognition using standard audio analysis features with supervised learning. For example, Sinyor et al. show 95.6% accuracy for a five-class vocabulary [14]. Prior work generally employs one-size-fits-all classifiers for a fixed set of classes trained on data from multiple people. To our knowledge, only Nakano et al. [13] adapt a trained classifier to an individual. Personalization is critical, as beatboxing technique varies [14,15] and vocalizations are impacted by many factors (e.g., gender, age, voice quality, audio environment).

BeatBox uses interactive machine learning to allow end users to create their own vocalization recognizers and dynamically adapt them to their technique, environment, and immediate musical goals (e.g., controlling a drum sample kit with either 3 or 15 instruments). Creating a beatbox recognizer that is effective for a given person and context requires the end user not only provide training examples, but also define the number and nature of input vocalization classes. Recognizers trained with varying numbers and types of vocalizations may present complex tradeoffs with regard to how classes overlap in a feature space, the types of mistakes a recognizer makes, and the person's proficiency in producing the defined vocalizations in the course of their musical work. Consequently, BeatBox aims to provide immediate and relevant feedback on the learnability of a set of defined classes together with lightweight mechanisms for adding, modifying, or removing entire classes of training examples.

3. ADDITIONAL RELATED WORK

We draw upon and contribute to research on end-user interactive machine learning and computational music, as discussed and cited in our previous two sections. Interactive machine learning has been applied in music by Fiebrink et al. [4], who observed a need to explore alternative learning concepts but did not examine mechanisms to make that exploration easier or more fruitful.

Prior work examines the need to help designers of gesture-based systems identify classes of gesture that are likely to be confused so they can work to improve recognizer reliability [2,9,11,12]. That work has focused on application developers designing general-purpose gesture sets that are to be deployed to end users with one-size-fits-all recognizers. The work also generally employs gesture-specific mechanisms for representing examples, identifying conflicts among classes, and presenting feedback on examples and classes. We build on this prior work in three ways: (1) we focus on end-user support for defining and training a recognizer, (2) our focus on music extends these ideas to new types of input, and (3) we show initial success with lightweight end-user exploration that should be explored in other domains.

4. BEATBOX IMPLEMENTATION

4.1 Defining Pads for Vocalization Classes

Figure 1 presents the BeatBox interface. A workspace provides a set of virtual *pads*, each associating a vocalization class (e.g., a mimicked bass drum) with a selected sound file (e.g., a recording of a real bass drum). These are analogous to hardware drum controller pads, physical buttons that trigger drum sounds. When the program is first run, no pads are present. An end user creates a new pad by clicking "+", choosing the sound file to be triggered, and then performing one or more examples of the vocalization that will trigger that pad. Each new example is represented as a colored square in the pad. Mousing over a square plays back the

recorded example, and clicking deletes it. At any time, a person can click within a pad to add more training examples or can define a new pad to add a new vocalization class. Entire pads can be deleted or temporarily disabled, as discussed below. The system automatically maintains a learned recognizer, trained on the examples currently associated with each active pad (i.e., ignoring any examples in disabled pads). This classifier is thus capable of matching any new vocalization to the most similar enabled pad.

4.2 Musical Use of Pads

After demonstrating one or more vocalizations for one or more pads, a person can immediately apply the recognizer to live input. BeatBox classifies each incoming vocalization, lighting up the most similar pad and playing its associated sound file. A person can therefore quickly experiment with the trained recognizer using free-form vocalizations. They can also record a rhythmic sequence of vocalizations into a *track*, using the track editor at the bottom. Each vocalization is immediately classified and colored according to the most similar pad. Tracks can be played back, manually edited, and layered to create more complex rhythms.

4.3 Classification Details

BeatBox analyzes vocalizations using a threshold-based onset detector, a feature extractor, and a classifier. One feature vector is extracted per onset. The spectral centroid and RMS (commonly used features related to timbre and volume) are computed for each of 80 100-sample analysis windows following the onset. The mean, standard deviation, minimum, and maximum are computed for normalized centroid and RMS over the 80 windows, yielding an 8-dimensional feature vector similar to those used in prior work (e.g., [14]). A k-nearest neighbor classifier using Euclidean distance [6] is trained on the examples associated with the active pads. The number of neighbors k is automatically selected to maximize cross-validation accuracy on the current training examples. In preliminary experiments on data from novice and experienced beatboxers, this classifier achieved accuracy comparable to or better than other standard learning algorithms.

4.4 Visualizing Classifier Reliability

BeatBox conveys both the expected reliability of a recognizer learned from example vocalizations and useful information about likely causes of error. We accomplish this by coloring the square for each training example according to the class predicted by the classifier trained from the enabled pads. A training example colored the same as its pad represents a correct classification, while an example colored different from its pad flags a problem. An isolated miscoloring may indicate a single example of poor quality (e.g., Figure 1's "Snare" pad). More widespread miscoloring may indicate two or more classes exhibit problematic overlap in the feature space and might not be reliably learnable (e.g., Figure 1's "Crash" and "Kick" pads). All examples are immediately re-colored with every change to the training data.

4.5 Experimenting with Vocalization Classes

Individual examples of poor quality can be examined, deleted, and replaced with new demonstrations. But this is insufficient when entire defined classes are problematic. BeatBox therefore allows interaction directly with entire pads to modify the learned recognizer. A pad may be disabled with a single click, removing its class and all of its training examples from the classifier. Re-enabling the pad returns the class and its associated examples to the training set. This allows rapid experimentation to determine what sets of vocalizations can be used together. Enabling or disabling a pad has the potential to impact the learnability of all other active pads, so these actions trigger re-coloring. Figure 2 illustrates an instance in which experimentation with vocalization concept definitions can lead to a more reliable recognizer, showing both the BeatBox interface as a person experiments and illustration of the overlap of desired classes in an abstract feature space. Finally, we support longer-term experimentation with vocalization classes by allowing pads to be named, saved to a pad library, and moved between the library and the active workspace.

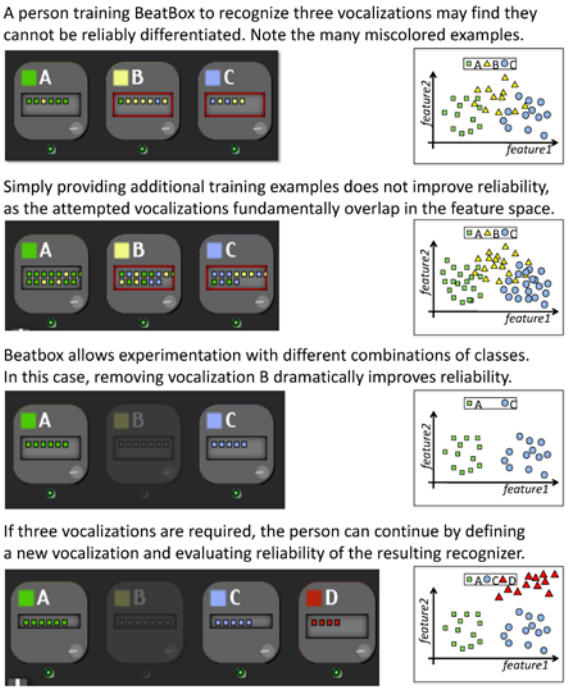

Figure 2. End-user experimentation with defining which classes to recognize can be critical to obtaining a reliable recognizer, as illustrated here in BeatBox and an abstract feature space.

5. OBSERVATIONAL STUDY

We conducted an observational study to gain initial insight into the effectiveness of the BeatBox system, whether and how end users interpret example color feedback, whether and how end users experiment with defining different classifiers using different combinations of pads, and to identify directions for future work. Of the seven participants (5 male, 2 female, ages 21-31), six had little or no prior beatboxing experience. A validation of the musical utility should include experienced beatboxers, but our goal in this initial study was a more narrow assessment of the mechanisms for exploring class definition and training.

Each participant session began with an experimenter demonstration of the features of BeatBox. Participants were then asked to use BeatBox to re-create a sequence of 6 simple audio prompt tracks, described in Table 1. They were asked to be as accurate as they could, though they could skip a prompt if they felt unable to complete it. The experimenter created a new pad for each sound file needed to re-create a prompt, but did not provide vocalization examples. Re-creation of a prompt required the participant populate each pad with at least one example, then record a new track in which they beatboxed the correct sequence of pads at the correct rhythm and tempo (which we kept simple to accommodate limited participant beatboxing experience). Participants could keep and reuse pads and examples from one prompt to the next. After all prompts, participants completed a short questionnaire.

Prompt Number	Pads Used	Hits in Prompt	Participants Completed	Time *mean (σ)*
1	Kick, Snare	4	7	2.89 (1.8)
2	Kick, Snare, Hi-hat	6	4	2.58 (2.1)
3	Kick, Snare, Hi-hat, Tom	8	2	2.73 (1.7)
4	Kick, Snare, Hi-hat	6	4	1.73 (1.1)
5	Orchestra Hit, Bass	4	7	1.67 (0.54)
6	2 Record-Scratches	4	7	3.47 (1.3)

Table 1. For each study prompt track: the pad types used, the number of vocalization hits in the track, the number of participants who completed the prompt, and the mean and standard deviation completion time, in minutes.

Our observations and logging indicate participants could successfully train BeatBox to recognize a vocalization vocabulary for most prompts, could employ this vocabulary to accurately create a track that mimicked the prompt, and could do so in under an average of 3 minutes. Participants highly agreed they understood how to use BeatBox (mean 1.71 on a 7-point Likert scale) and agreed they were ultimately able to train it to accurately recognize vocalizations (mean 2.85 on a 7-point scale).

Participants relied on both the example coloring feedback and disabling of pads to accomplish their tasks, thus lending preliminary validation to our design of these interactive machine learning elements. The initial vocalizations that participants chose were often not reliably learnable. Upon noticing miscolorings while giving example vocalizations, many participants played back examples for the confused classes, remarked that they sounded similar, and then attempted to change their vocalizations. In some cases, participants tried recording more examples for the confused classes. In others, they deleted examples that had been recorded accidentally or had not been performed consistently with their intended vocalization. Participants often disabled pads not used in the current prompt to improve classifier accuracy for the prompt. This allowed some participants to re-use prior vocalizations in later prompts, thus saving themselves the effort of defining new vocalizations that were reliably learned.

6. CONCLUSION

We have argued that support for quickly exploring and assessing the learnability of different concepts can allow end users to take fuller advantage of machine learning in building personalized and tailored recognizers, instead of being limited to classes provided by a pre-trained classifier. We have proposed techniques to provide end users feedback on the learnability of a set of desired concepts as well as support for dynamically changing the set of desired concepts, and we have implemented these techniques in a new system for end-user creation of custom beatbox recognizers. A preliminary study supports the utility of these techniques.

As a challenge for future work, our study revealed that example coloring feedback was clearly helpful but could not alert people to all relevant problems. In particular, participants often produced vocalizations more consistently during example recording (i.e., during training) than during track creation (i.e., during usage). As a result, some vocalization classes were reliably learnable during training (i.e., producing no example miscolorings), but were confused by the recognizer when participants created tracks. Future work might investigate mechanisms to obtain training examples that are more representative of vocalizations produced during track creation. For example, a system might have people supply training examples as rhythmic sequences of different vocalizations (i.e., making vocalizations as if creating a track). Alternatively, a system might allow vocalizations produced while making a track to also be added to pads (i.e., using correction of recognition in a track to gather additional training data for the recognizer). Systems could also give additional feedback to help identify and eliminate consistency problems.

Other future work might enhance the musical utility of BeatBox (e.g., by employing a larger set of audio analysis features to improve the range of sounds that can be recognized, by adding the ability to export tracks as MIDI for use with other composition software). We are also interested in extending BeatBox to allow end-user customized *continuous* vocal control over volume, pitch, and other musical dimensions of computer-generated sound.

7. ACKNOWLEDGEMENTS

This work was supported in part by Microsoft and the National Science Foundation under award IIS-0812590 and OAI-1028195.

8. REFERENCES

[1] Amershi, S., Fogarty, J., Kapoor, A., and Tan, D. Examining Multiple Potential Models in End-User Interactive Concept Learning. *CHI 2010*, 1357-1360.

[2] Ashbrook, D., and Starner, T. MAGIC: A Motion Gesture Design Tool. *CHI 2010*, 2159-2168.

[3] Atherton, M. Rhythm-Speak: Mnemonic, Language Play, or Song? *ICOMCS 2007*, 15-18.

[4] Fiebrink, R., Cook, P.R., and Trueman, D. Human Model Evaluation in Interactive Supervised Learning. *CHI 2011*, 147-156.

[5] Fogarty, J., Tan, D., Kapoor, A., and Winder, S. CueFlik: Interactive Concept Learning in Image Search. *CHI 2008*, 29-38.

[6] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11 (1), 2009, 10-18.

[7] Hazan, A. Performing Expressive Rhythms with BillaBoop Voice-Driven Drum Generator. *DAFX 2005*.

[8] Kapur, A., Benning, M., and Tzanetakis, G. Query-By-Beat-Boxing: Music Retrieval for the DJ. *ISMIR 2004*, 170-177.

[9] Kin, K., Hartmann, B., DeRose, T., and Agrawala, M. Proton: Multitouch Gestures as Regular Expressions. *CHI 2012*, 2885-2894.

[10] Kulesza, T., Stumpf, S., Burnett, M., Wong, W.-K., Riche, Y., Moore, T., Oberst, I., Shinsel, A., and McIntosh, K. Explanatory Debugging: Supporting End-User Debugging of Machine-Learned Programs. *VL/HCC 2010*, 41-48.

[11] Long, A. C. Quill: A Gesture Design Tool for Pen-Based User Interfaces. PhD Thesis, University of California, Berkeley, 2001.

[12] Lü, H., Fogarty, J. and Li, Y. Gesture Script: Recognizing Gestures and their Structure using Rendering Scripts and Interactively Trained Parts. *CHI 2014*, To Appear.

[13] Nakano, T., Goto, M., Ogata, J., and Hiraga, Y. Voice Drummer: A Music Notation Interface of Drum Sounds using Voice Percussion Input. *UIST 2005*, 49-50.

[14] Sinyor, E., McKay, C., Fiebrink, R., McEnnis, D., and Fujinaga, I. Beatbox Classification using ACE. *ISMIR 2005*.

[15] Stowell, D. Making Music through Real-Time Voice Timbre Analysis: Machine Learning and Timbral Control. PhD Thesis, Queen Mary University of London. 2010.

[16] Stumpf, S., Rajaram, V., Li, L., Wong, W.-K., Burnett, M., Dietterich, T., Sullivan, E., and Herlocker, J. Interacting Meaningfully with Machine Learning Systems: Three Experiments. *IJHCS*, 67 (8), 2009, 639-662.

[17] Michael Jackson Beatboxing. *ABC Primetime Live*, 1995. http://www.youtube.com/watch?v=K0MbIq4uwTc