Towards Speeding Audio EQ Interface Building with Transfer Learning

Bryan Pardo
EECS
Northwestern University
Evanston, IL
pardo@northwestern.edu

David Little
Comm. Sciences and Disorders
Northwestern University
Evanston, IL
d-little@u.northwestern.edu

Darren Gergle
Communication Studies
Northwestern University
Evanston, IL
dgergle@northwestern.edu

ABSTRACT
Potential users of audio production software, such as parametric audio equalizers, may be discouraged by the complexity of the interface. A new approach creates a personalized on-screen slider that lets the user manipulate the audio in terms of a descriptive term (e.g. "warm"), without the user needing to learn or use the interface of an equalizer. This system learns mappings by presenting a sequence of sounds to the user and correlating the gain in each frequency band with the user's preference rating. The system speeds learning through transfer learning. Results on a study of 35 participants show how an effective, personalized audio manipulation tool can be automatically built after only three ratings from the user.

Keywords
Human computer interaction, music, multimedia production, transfer learning

1. INTRODUCTION
We seek to simplify interfaces in software for media production and align them with the user's conceptual model. In this paper, we focus on audio equalizers. Our approach is to quickly and automatically personalize the controller through a guided learning interaction, where the user teaches the system a concept. The idea is to let the artist directly control the device in terms of the desired perceptual effect. For example, the tool would learn what "muffled" means to the artist, and then create a slider to let her make the recording more or less "muffled," bypassing the bottleneck of technical knowledge.

This approach has been adopted in the work of Sabin, Rafii and Pardo, which dynamically individualizes the mappings between human language descriptors and parameters for equalization and reverberation tools [1]. Their work has been commercialized in the equalization plug-in *iQ* [2] which has been downloaded by thousands of users and positively reviewed in the music audio popular press [3]. This indicates the interactive learning approach used in *iQ* (described in Section 3) is a useful new paradigm that complements existing tools.

While this approach has been successful in creating a new interface paradigm for equalization, the current method [1] requires a relatively large number of user ratings (on the order of 25) to achieve high-quality results. The work presented here this paper extends and improves on that approach by incorporating *transfer learning* [4]. The result is a personalized interface that is learned much faster.

NIME'12, May 21-23, 2012, University of Michigan, Ann Arbor.

2. PRIOR WORK
Prior work on learning a listener's preferences uses a case-by-case approach to setting the equalization curve of a small number of frequency bands in a hearing aid [5]. The most common procedure for doing this is known as the modified simplex procedure [e.g., 6, 7]. Another common approach has been to directly map equalizer parameters to commonly used descriptive words using a fixed mapping [e.g., 8, 9]. Recent work in the HCI community has sought to address these challenges by integrating algorithmic advances from the machine learning communities [10][11].

In terms of artistic creation for music, researchers have focused on two main streams of work related to music generation and production. The first uses new, often tactile, interaction techniques serving as new musical instruments or audio control surfaces [3] or for analyzing and representing musical dynamics such as chord changes [13]. Our work is related, but the use of transfer learning to use prior learned concepts is distinct.

The Wekinator [4] is an on-the-fly music performance control mapping tool that lets informed users interactively control machine learning algorithms of their choice by choosing inputs, features. This work is complementary to ours in that it is for more technically knowledgeable users, and does not use language as a paradigm.

We use transfer learning, as it is understood in the machine learning community. Transfer learning [4] makes use of data from previously learned tasks. While we are unaware of prior work applying transfer learning of audio concepts to create user-specific audio production tools, transfer learning has been applied to user interfaces outside the audio domain. Previous approaches range from customizing user interface controls and layout to best suit a given interaction environment [15,16,17] to those that interactively personalize results in content discovery or search and retrieval.

The most closely related work to ours outside of audio production tools is the CueFlik system [18], which uses machine learning to correlate natural-language concepts to *classify* digital objects. This contrasts with our work, where we learn concepts in order to *manipulate* the degree to which an object (an audio recording) conforms to a given concept.

3. THE BASELINE SYSTEM

Rather than use a single interface for all users, based on past hardware design, we utilize an approach to building a personalized interface for each user. The idea is to create a controller whose interface is conceptualized in descriptive terms defined by the user.

Before attempting to speed learning by applying transfer learning, we discuss the approach we adopt, based on an existing audio concept learner [1]. We give an overview of the process here, and refer the reader to the prior work for more detail on this process.

1. The user selects an audio file and a descriptor (e.g. "warm" or "tinny").
2. We process the audio file once with each of N probe 40-band equalization curves, making N examples.
3. The user rates how well each example sound exemplifies the descriptor (Figure 3).
4. We then build a model of the descriptor, estimating the effect of each frequency band on user response. This is done by correlating user ratings with the variation in gain of each band over the set of examples. The slope of the resulting regression line for each frequency band indicates the relative boost or cut for that frequency (Figure 2).
5. The system presents to the user a new controller that controls filtering of the audio based on the learned model of how to manipulate audio.

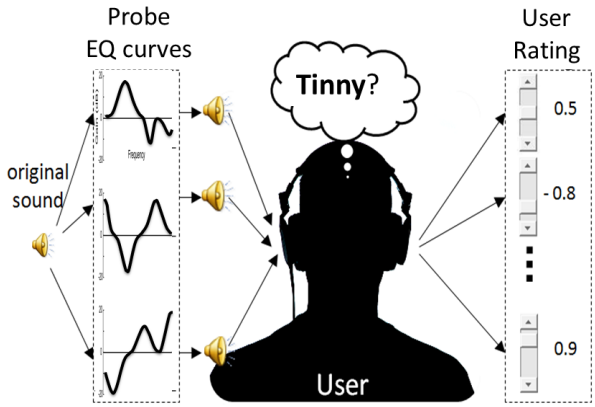

Figure 1. A user rates how well each equalization curve applied to an original sound embodies the word "tinny".

The existing approach requires the user to rate roughly 25 audio examples to generate an acceptable controller [1]. In this work, we speed learning so a good controller could be learned from roughly three user ratings of audio examples. We do this through reuse of data from prior users and concepts (transfer learning).

4. APPLYING TRANSFER LEARNING

Define a user-concept as sound adjective taught to the machine by a particular user (i.e. Bob's concept for "warm" sound). A user-concept is taught to the system by rating the example set ***M***, as described in Section 3. This typically takes on the order of 25 interactions to build a successful controller.

One effective way to speed concept learning is through the reuse of data from previously learned concepts. The machine learning community calls this transfer learning [4]. As more and more users train the system, we can increasingly use transfer learning to reduce the number of questions needed to build an acceptable controller for new users. When presented with a new user's concept, the learner may be able to achieve good results by asking only a few questions to locate the user's concept in a space defined by previous concepts taught by previous users. Once the concept is located in the space, previous training data can be used to inform the learning of the current concept.

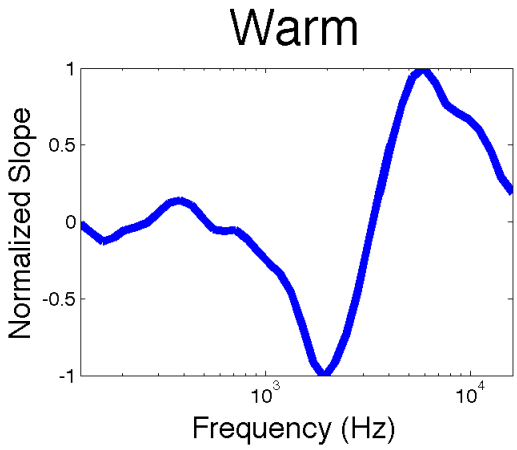

Figure 2. A learned equalization (EQ) curve for a single user's concept of "warm." The vertical axis indicates the relative boost or cut in the amplitude at the given frequency.

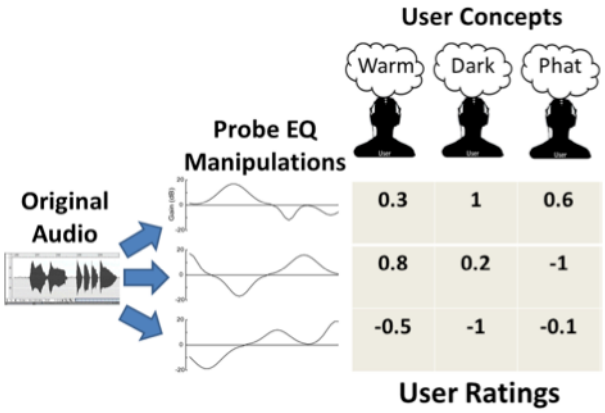

Figure 3. An audio file is manipulated with *m* equalization curves to create *m* examples. Each user rates all examples in terms of a particular adjective (e.g. "How 'warm' is this example). Ratings range from 1 to -1.

Figure 3 illustrates a pool of rated examples for three user-concepts: *warm*, *dark*, and *phat*. We can estimate the similarity between two user-concepts by determining how similar user responses were to the same set of examples when teaching the system those concepts. Presumably, the more similar the set of ratings, the more similar the concepts. Therefore, the more relevant the prior ratings are to learning the current concept.

To do transfer learning we create a fixed question set by manipulating a standard audio file (e.g. a 5 second passage from Delibes' *Flower Duet*) using a tool, such as an equalizer. Do this ***m*** times (on the order of 50), creating a set of examples ***M*** to be rated by users. For each of ***n*** users, have the user select some concept and rate all the examples in ***M*** on a continuous scale (-1 to 1, in our work) for how well each example conforms to that user's chosen concept. This creates a set of prior knowledge to use in transfer learning.

To do transfer learning, we first put an existing set of user-concepts into a vector space. Let ***Q*** be a subset drawn from the set ***M*** of examples rated by users. Each user–concept's location is determined by that user's ratings of the examples in ***Q*** when training the system on a concept.

When training the system on a new user-concept, rather than asking the user to rate the full set of ***M*** examples, we ask them to rate only the subset ***Q***, placing the new user-concept in the vector space. Then, we estimate the current user's ratings for the remaining ***M-Q*** examples by a weighted combination of user responses for past concepts. The weight given to the responses for a prior user-concept is determined by distance to the current user-concept in the vector space. We use these estimated ratings in the concept training procedure for the new user's concept. Properly done, this will greatly lessen the number of examples the typical user must rate before an effective controller can be learned.

While we could apply this approach even when two users' concepts have different labels (Bob's "warm" and Maria's "dark"), in this work we apply transfer learning only to data collected from other users training the system on the same concept word that the current user is teaching the system. For example, only the example ratings from prior users on the word "warm" would be included when learning a "warm" controller for a new user.

4.1 Distance and Weighting

We base the weight of user-concept *w(u)* by the distance to the new concept *v*.

$$w(u) = \frac{\exp(-2d(u,v)^2)}{\sum_{k \in U} \exp(-2d(k,v)^2)} \quad (1)$$

Equation (1) is the weight given to a user-concept, given a distance function $d(u,v)$. We considered a variety of mapping functions and p-norms. Space limits preclude describing our analysis, but experimental results showed us that Manhattan distance (L1 norm) performed best for our data. This is the distance used in the experiments reported in this work.

Given a set ***U*** of prior user-concepts that have been placed in a vector space as described earlier, we can then estimate what rating the new user will give to unrated example ***q*** using a weighted sum of prior user-concept ratings for that example.

$$\tilde{r}_v(q) = \sum_{u \in U} w(u) \cdot r_u(q) \quad (2)$$

5. EXPERIMENTAL DESIGN

We have argued that our approach lets us build personalized relevant controllers after only a few user interactions. How much does transfer learning speed (or improve) learning for our problem? To answer this question, we had a set of 35 users train the system on the meaning of five sound adjectives using our baseline learning method. All users taught the system the same 5 words: "muffled", "tinny", "broad", "bright" and "warm". In deployment of a system, we expect people to use a multitude of terms we cannot predict. However, to evaluate the effectiveness of transfer learning in a controlled environment we require participants use the same set of terms. These terms were selected in conjunction with a Ph.D in audio psychology.

All rated the same set of audio examples. The stimuli were always manipulations of a short (5 second) musical passage from Delibes' Flower Duet at the compact disc standard bit depth and rate (16 bits at 44.1 kHz). The Flower Duet was chosen for its broad spectral coverage and ease of repeated listening. We used a query set of 50 equalization curves found to be effective in previous work [1]. The excerpt from the Flower Duet was manipulated once by each of the 50 curves, creating 50 manipulated examples (the set ***M***). The same 50 examples were presented in randomized order for each word concept taught to the machine by each study participant.

Users were seated in a quiet room with a computer that controlled the experiment and recorded user responses. The stimuli were presented binaurally over headphones. Users could adjust the overall sound level. Each user took part in a single one-hour *session*. Each session was grouped into five *runs*. In a run, the user was presented with a single word (e.g. bright) and asked to teach the system their concept for that word by rating a set of example audio files on how well each example embodied the concept.

We kept a record of all user responses during baseline training. Given this, we tested each learning method as follows: For a particular user-concept (e.g. Maria's "warm") let the system see a subset of the user's ratings of examples (e.g. her first 5 example ratings). For transfer learning, allow access to a set of other users' responses for other concepts (e.g. all of Bob's and Dave's training data). Train the system on this data. Evaluate the system by how well it predicts the user's ratings on the rest of the training examples (Maria's other 45 ratings of examples for "warm").

5.1 Performance Measures

We evaluate our new learning methods through *machine-user correlation.* The *machine-user correlation* measures system performance by correlating user ratings of examples to machine predictions of user ratings. Given a set of rated examples, **M**, the machine-user correlation is the Pearson correlation coefficient of user ratings to machine-generated predictions of the human ratings for the entire set of rated examples. Since machine-user correlation can be calculated from our data without requiring further user testing, much of our evaluation uses this measure.

To generate a predicted rating, the system first learns a concept EQ curve using some method and condition (e.g. transfer learning of Maria's "warm," given a pool of 25 existing user-concepts and 10 examples rated by Maria). We then generate a prediction of the user's rating of a new example by comparing the learned concept EQ curve to the EQ curve applied to the new example to be rated by the user (the 11th example). The more similar the concept EQ and the EQ applied to the new example are, the higher the predicted user rating of that example.

6. RESULTS

For each of the user-concepts, we collected user ratings of the 50 audio examples. Since the same 50 examples were used for all users and all words, we could simulate the effectiveness of the baseline learner (without transfer learning) by selecting each user-concept and building a concept model from a randomly-selected subset of *n* rated examples. Using the learned concept model, we then predicted user responses on the remaining rated examples and measured the correlation between machine predictions and actual user ratings as described in a previous section. For a given value of *n*, this gave us on machine-correlation value per user-concept. We did this for each value of *n* from 1 (a single rated example from the current user) to 50 (all the rated examples for that user-concept), calculating the machine-user correlation at each step. This formed our baseline learning method.

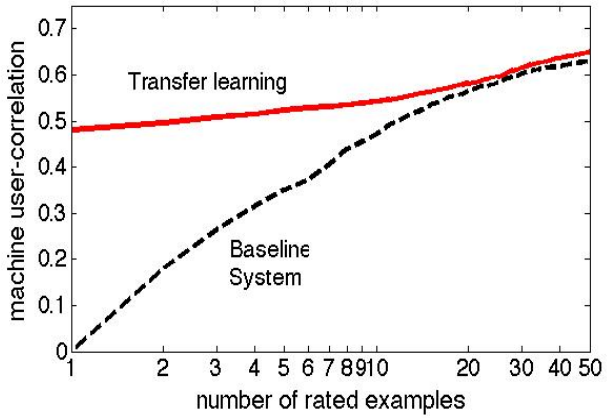

Figure 4. Mean machine-user correlation for each learning method, averaged over all words and all users. The relative performance of these methods was similar for each of the five descriptive words taught to the system.

We then repeated this process for each learning method. *Transfer learning* utilized data from n randomly selected ratings by the user, augmented by all of the example ratings from the other learned user-concepts. For this experiment, the only other user-concepts that were included in the weighted sum were those where the prior user taught the system the same word as the current user. The data from the remaining concepts was weighted using the n-dimensional Manhattan distance measure described earlier, where n is the number of rated examples for the current concept.

The learning curve as the number of rated examples increases is shown in Figure 8. Transfer learning speeds learning over the baseline in all cases. When prior user-concepts share a word with the current concept, this speed-up is dramatic, giving a usable controller with only a couple of user responses.

7. CONCLUSIONS

Using a simple approach to transfer learning, we have shown significant improvements in the number of user-ratings needed to learn a desired equalization controller from user feedback. A previous method required 25 rated examples to yield an effective controller. We reduce this by a factor of 10 in the case where there is a sufficiently large prior pool of users who have taught the system the same adjective.

This work promises to enable useful on-the-fly tool building in the recording studio or for home-studio use (e.g. an updated *iQ* plug-in for Apple's Garage Band). A user unfamiliar with existing equalizer interfaces could quickly (after answering just two or three questions) create tools to manipulate audio within the terms defined by the user. This algorithm could also be helpful for experienced users who would prefer to avoid directly adjusting equalizer parameters.

Future work includes applying transfer learning in the case where prior users have taught the system concepts but none of the concepts share a word label with the new concept to be learned, even though they may describe similar things. Another direction for future research is to adaptively select the order in which users are presented examples. Judicious selection of examples may speed learning even further.

8. ACKNOWLEDGEMENTS

This work is funded by National Science Foundation Grant numbers 1116384 and 1116384. We thank Alex Madjar and Andrew Sabin for their input and help.

9. REFERENCES

[1] Sabin, A., Rafii, Z. and Pardo, B. (2011). "Weighting function-based rapid mapping of descriptors to audio processing parameters," *Journal of the Audio Engineering Society*, pp. 419-430, vol. 59(6)

[2] http://www.ear-machine.com/iQ.html

[3] Burton, J. (2011), Ear Machine iQ Intelligent Equaliser Plug-in For Mac OS & Windows, *Sound on Sound*, June 2011, pp. 132-133.

[4] Pan, S. and Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22(10),* pp. 1345-1359.

[5] Neuman, A.C., et al. (1987). An evaluation of three adaptive hearing aid selection strategies. *J AcoustSoc Am, 82(6)*, pp. 1967-1976.

[6] Kuk, F.K. and Pape, N.M. (1992). The reliability of a modified simplex procedure in hearing aid frequency-response selection. *J Speech Hear Res, 35(2)*, pp. 418-429.

[7] Stelmachowicz, P.G., Lewis, D.E., and Carney, E. (1994). Preferred hearing-aid frequency responses in simulated listening environments. *J Speech Hear Res, 37(3)*, pp. 712-719.

[8] Mecklenburg, S. and Loviscach, J. (2006). subjEQt: Controlling an equalizer through subjective terms. In *Ext.Abs. of the Proc. of CHI 2006*, pp. 1109-1114. NY: ACM Press.

[9] Reed, D. (2001). Capturing perceptual expertise: a sound equalization expert system. *Knowledge-Based Systems, 14*, pp. 111-118.

[10] Morris, D. and Secretan, J. (2009). Computational creativity support: Using algorithms and machine learning to help people be more creative. In *Ext.Abs. of the Proc. of CHI 2009*, pp. 4733-4736. NY: ACM Press.

[11] Simon, I., Morris, D., and Basu, S. (2008). MySong: Automatic accompaniment generation for vocal melodies. In *Proc. of CHI 2008*, pp. 725-734. NY: ACM Press.

[12] Fiebrink, R., D. Morris, and M. R. Morris. (2009) Dynamic mapping of physical controls for tabletop groupware. *Proceedings of ACM CHI 2009*, Boston

[13] Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R., and Pignol, M. (2002). Generating remote control interfaces for complex appliances. In *Proc. of UIST 2002*, pp. 161-170. NY: ACM Press.

[14] Fiebrink, R., D. Trueman, and P. R. Cook. (2009) A meta-instrument for interactive, on-the-fly machine learning. *Proceedings of New Interfaces for Musical Expression (NIME*), Pittsburgh, PA

[15] Gajos, K. and Weld, D.S. (2004). SUPPLE: Automatically Generating User Interfaces. In *Proc. of IUI 2004*, pp. 93-100. NY: ACM Press.

[16] Lin, J. and Landay, J.A. (2002). Damask: A Tool for Early-stage Design and Prototyping of Multi-device User Interfaces. In *Proc. of the 8th International Workshop on Visual Computing*, pp. 573-580.

[17] Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R., and Pignol, M. (2002). Generating remote control interfaces for complex appliances. In *Proc. of UIST 2002*, pp. 161-170. NY: ACM Press.

[18] Fogarty, K., Tan, D., Kapoor, A., and Winder, S. (2008). CueFlik: Interactive Concept Learning in Image Search. In *Proc. of CHI 2008*, pp. 29-38. NY: ACM Press