Journal of the Audio Engineering Society
Vol. 64, No. 7, July 2016 (© 2016)
DOI: http://dx.doi.org/10.17743/jaes.2016.0013

Supervised and Unsupervised Sound Retrieval by Vocal Imitation

YICHI ZHANG AND ZHIYAO DUAN, *AES Member*
(yichi.zhang@rochester.edu) (zhiyao.duan@rochester.edu)

Dept. of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, USA

Searching sounds with text labels is often problematic and time consuming as text labels do not often describe the detailed audio content. Query by example is a way to improve the effectiveness and efficiency of sound retrieval. In this paper we propose a novel approach for sound query by example: query by vocal imitation. Vocal imitation is commonly used in human communication and can be employed for novel human-computer interaction. We propose two related systems. The supervised system addresses the retrieval problem by vocal imitation recognition. It trains a multi-class classifier using training vocal imitations of different sound classes in the library and classifies a new imitation query into one of the classes. This system thus cannot retrieve sounds that are not trained. The unsupervised system is more flexible in that it measures the feature distance between the imitation query and each sound in the library and returns sounds the most similar to the query. One challenge of designing these systems is finding an effective feature representation of imitation queries and sounds in the library. Existing handcrafted audio features may not work well given the variety of vocal imitations and the mismatch between vocal imitations and actual sounds. We propose to learn feature representations from training vocal imitations automatically using a Stacked Auto-Encoder (SAE). Experiments show that sound retrieving performance by automatically learned features outperform those carefully handcrafted ones that were used in existing systems in both supervised and unsupervised settings.

0 INTRODUCTION

Designing ways to efficiently access and manage multimedia documents such as audio recordings is an important information retrieval task as these documents proliferate and grow. Existing ways to index and search audio documents are based on text metadata and text-based search engines. This, however, is not efficient or effective in many scenarios. First, much of the audio in user-contributed online repositories (e.g., SoundCloud, Freesound) has metadata that does not describe the details of the audio content, making the content undiscoverable through a text-based search. Second, files labeled with content-relevant tags do not often have specific enough tags based on which searches can return hundreds or thousands of examples. Third, even for audio libraries that are carefully designed with a hierarchical taxonomy and detailed text labels (e.g., sound effect libraries), searching a specific sound is not easy. It requires users to be familiar with the taxonomy and remember the detailed descriptors of the sound, which is the ability that only experienced sound production engineers have. Fourth, even for these experts, difficulties exist. Many sounds, especially computer-synthesized sounds, do not have semantic meanings and are often labeled with the parameters of the synthesizers, and text-based search becomes very non-intuitive.

A query-by-example (QBE) [1] sound retrieval system addresses these issues. Presented with an audio recording as a query, the system compares the query with sound files in the library and returns files similar to the query. It can be combined with a text-based search to make the search more efficient, effective, and intuitive.

QBE for music files has been addressed in several scenarios. Query-by-beat-boxing allows users to find drum loops with similar rhythmic patterns to their vocal percussion [2]. Query-by-humming allows users to find songs with a similar melody to their humming or singing [3], [4]. This technique, however, does not generalize to sounds that do not have a pitch. Cover song identification retrieves songs that are the same as the query song but of a different version (e.g., by a different band, in different environments) [5], [6]. However, this technique does not generalize to non-music audio.

In this paper we propose a novel QBE system for general audio. More specifically, the system takes a user's vocal imitation as a query and searches for sounds in the library that are similar to the query. Vocal imitation is a human behavior where the user utilizes his/her voice as well as

lips, tongues, cheeks, etc., to mimic a specific sound. The reason that we use vocal imitations as queries are twofold. First, vocal imitation is widely used in human communication. It improves the vividness of a presentation; helps to convey ideas that are difficult to describe in language; and is an effective way to communicate with people not speaking the same language. Second, vocal imitation is a natural extension to speech and singing and it broadens existing ways of audio-based human-computer interaction. If we proceed one step further, it can enable novel ways for animal-computer interaction. In fact, vocalization is arguably the most natural way for animals to interact with computers.

Modeling vocal imitations and using them as queries, however, is challenging. The challenges mainly reside on the feature representation, i.e., what features are most effective to represent vocal imitations as queries for sound search. The reasons are twofold. First, for different kinds of sounds, the key aspects that people tend to imitate are different. For example, for a car horn (du-du) sound, the key characterizing aspects are likely to be the constant pitch contour and the rhythm (relative lengths of horns versus silences), while the absolute pitch (e.g., 200 Hz vs. 300 Hz) and timbre (e.g., du versus beep, ba) of different imitations can be quite different. For a cat-meowing sound, however, the key aspects are likely to be both the pitch and timbre evolution, while the sound, hence the imitations as well, may lack a clear rhythm. Second, vocal imitations are subject to the physical constraints imposed by the human voice system. For example, the human voice cannot match the variety of pitch, timbre, and dynamics of many target sounds; cannot make as fast amplitude or frequency modulations as motors or synthesizers; and cannot produce polyphonic sounds. Therefore, surface-level features of a correct query-sound pair can lie in very different spaces, although some deeper-level representations of them must be similar.

In this paper we present our work on designing sound retrieval systems using vocal imitation queries. Two related systems are presented: the supervised system [7] addresses the retrieval problem through vocal imitation recognition. A multi-class classifier is first trained for all sounds in the database using their training vocal imitations. When a new imitation query is presented, the system classifies it into one of the sounds in the database and retrieves the sound. While this system achieves good performance, it only works in the closed-set scenario, i.e., it cannot retrieve sounds that are not trained.

The unsupervised system [8], however, is much more flexible. It measures the distance between the imitation query and each sound concept in the database and retrieves the closest ones to the user. We explore different kinds of distance measures at two different levels: patch-level and recording-level. As for the former, each imitation query and sound is represented by a sequence of features extracted in each overlapping 525 ms long segment of the audio. We propose to calculate the combination of the Kullback-Leibler (K-L) divergence [9] and the Dynamic Time Warping (DTW) distance [10] between these two feature vector sequences of each query-sound pair. For the latter, we represent each imitation query and sound by a long feature vector that summarizes the whole recording statistics over the patch-level features. We then calculate the cosine distance between the two recording-level feature vectors for each imitation-sound pair.

To address the feature representation challenges, we propose to learn a feature representation from a collection of vocal imitations using a Stacked Auto-Encoder (SAE), one type of deep neural network. Compared to handcrafted features such as Mel-frequency Cepstral Coefficients (MFCC) [11], and spectral features automatically learned features have the benefit of tailoring the feature representation to the specific type of input data and modeling the complex non-linear relationships between them.

We conduct systematic experiments using the VocalSketch Data Set v1.0.4 [12]. Experiments show that the automatically learned features by the SAE outperform the carefully handcrafted features in both our supervised and unsupervised systems. Experiments also show that the more flexible unsupervised system achieves comparable performance with its supervised counterpart in several categories. Detailed comparison of different distance measures of the unsupervised system is also provided.

The main contributions of this work are threefold. First, this is the first systematic investigation of sound retrieval by vocal imitation and we propose both a supervised and unsupervised system. To our best knowledge, there exists little work about this topic. Second, this is the first work that employs automatic feature learning techniques and demonstrates their superiority over carefully handcrafted features for sound retrieval. Third, we conduct thorough experiments on a large dataset while existing works used a much smaller dataset or lacked experiments.

Preliminary versions of the proposed systems have been published in [7] and [8]. In this paper we improve the automatic feature learning module by designing a better SAE, improve the distance measure module of the unsupervised system, construct more competitive baselines, and conduct more systematic experiments. The rest of the paper is organized as follows:

We first review related work in Sec. 1, then introduce feature representations by neural networks in Sec. 2. Our proposed supervised and unsupervised vocal imitation recognition systems are described in Sec. 3 and Sec. 4 respectively. Experimental results are shown in Sec. 5 and finally we conclude the paper in Sec. 6.

1 RELATED WORK

To our best knowledge, there are few systems designed for sound retrieval by vocal imitation.

Roma and Serra [13] proposed an online system that allows the user to query sounds on Freesound.org by recording audio with a microphone. The statistics of MFCC and their derivatives were used as descriptors to represent a given audio clip but no formal evaluation was reported.

Blancas et al. [14] built a supervised system using carefully designed features extracted by the Timbre

Toolbox [15] and an SVM classifier. A vocal imitation query was classified to a class within each sound category and sounds in that class were retrieved. This system was further combined with text-based search to retrieve sounds that have similar text labels to the sounds in the class. This system, however, was only evaluated on four categories (cat, dog, car, and drums) and each category only had three or four classes. In fact, when scaled to a larger database with more classes, the hand-crafted features may have difficulties in representing the complex acoustic aspects of vocal imitations. In addition, this supervised system cannot retrieve sounds that do not have training imitations.

Helén and Virtanen [16] designed a query by example system for generic audio. Other than query by vocal imitation, one sample drawn from the database served as a query example and the rest were considered as the database. Both query and database sound samples were divided into short frames and a feature vector is extracted in each frame. The query-sample pairwise similarity is measured by the difference between the probability density functions (pdfs) of their frame-wise features by Gaussian mixture models (GMM). The pdf difference was represented by various similarity measures like Mahalanobis distance, K-L divergence and its variations, cross-likelihood ratio test, etc. However, traditional handcrafted features such as MFCCs, spectral spread, harmonic ratio, total energy, etc., were extracted.

2 FEATURE REPRESENTATIONS

One of the main challenges of sound retrieval by vocal imitation is to find appropriate feature representations of vocal imitations. A good representation should capture the essential sound aspects that make the imitations resemble the corresponding sounds. Conventional audio features are designed by clever engineering to cover different aspects of sounds, including pitch (by fundamental-frequency related features), loudness (by energy features), timbre (by spectral and cepstral features), and their temporal modulations (by temporal differences of the features). These features, however, may have difficulties in modeling the relationships between vocal imitations and the corresponding sounds. There are two reasons.

First, the key aspects along which imitations are similar to their corresponding sounds may not be a simple enumeration of the surface-level aspects that the conventional features cover. Some deeper interactions between the surface-level aspects might be important in modeling vocal imitations. Second, surface-level features of vocal imitations are often quite different from those of the corresponding sounds, due to the physical constraints of the human voice system. However, there must be some deeper-level representation by which the imitation and the corresponding sound are similar.

In recent years, automatic feature learning [17] has shown its significant advantages over handcrafted features in many tasks in computer vision [18], speech recognition [19], and music information retrieval [20]. The basic idea is to use Deep Neural Networks (DNN) to fit the training data in an unsupervised way. Thanks to the highly nonlinear relationship between input and output of the DNNs, automatically learned feature representations are often a highly non-linear transformation of the input raw data and are often able to capture the underlying structures. For example, feature representations learned by a convolutional neural network on human face images show local organs such as nose and eye in shallow layers and holistic representations of the face in deeper layers [18].

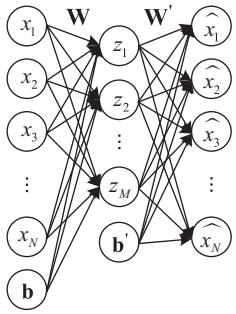


Fig. 1. Illustration of an auto-encoder. **W** denotes the weights between the input layer (x) and the hidden layer (z), and $\mathbf{W}'$ denotes the weights between the hidden layer (z) and the output layer (y). **b** and $\mathbf{b}'$ are the biases. Auto-encoders are trained to reconstruct the inputs at the outputs.

In this paper we choose to adopt Stacked Auto-Encoder (SAE) [21] for automatic feature learning because it is simple, easy-to-train, and achieves feature dimensionality reduction. As shown in Fig. 1, an auto-encoder is a neural network with one hidden layer and the same amount of inputs and outputs. The input vector is normalized to the range from 0 to 1. The transfer function of each hidden neuron and output neuron is a sigmoid function that squashes the normalized input into a bounded output ranging from 0 to 1. This model uses the backpropagation algorithm to learn the parameters **W**, **b**, $\mathbf{W}'$ and $\mathbf{b}'$, so that the output layer $\widehat{x}$ approximates the input layer x. As the hidden layer often has a smaller size than the input/output layers, the hidden layer output is forced to learn a compressive representation of the input, which achieves dimension reduction.

An SAE is constructed by stacking multiple auto-encoders together. Fig. 2 shows the process of building an SAE with two hidden layers. In this model we utilize a greedy layer-wise training process to learn the weights and bias. Specifically, by feeding the raw data as input, we first learn the parameters $\mathbf{W_1}$, $\mathbf{b_1}$, $\mathbf{W_1}'$, and $\mathbf{b_1}'$ of the first auto-encoder with the backpropagation algorithm. We then discard the output layer and feed the hidden layer output to the second auto-encoder to learn its parameters $\mathbf{W_2}$, $\mathbf{b_2}$, $\mathbf{W_2}'$, and $\mathbf{b_2}'$. The second auto-encoder is thus stacked onto the hidden layer of the first auto-encoder and the resulted SAE has two hidden layers. Following the same rule, we can stack more auto-encoders and build SAEs with more hidden layers. A sparsity constraint is added to the objective function of training by setting the average activation of each hidden neuron to be close to 0. This means in most time the hidden neurons are inactive. The sparsity constraint

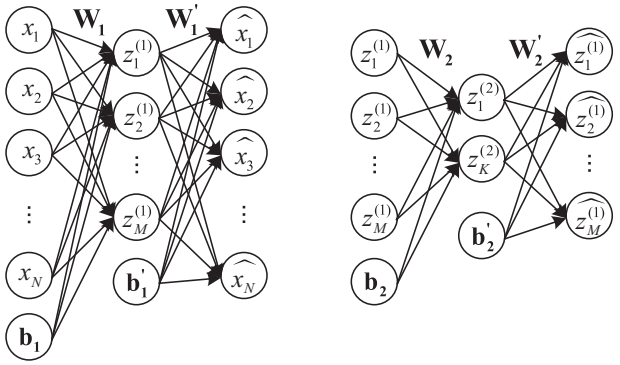


Fig. 2. Illustration of a stacked auto-encoder (SAE) with two hidden layers as two auto-encoders. Raw data is fed to the first auto-encoder to learn $\mathbf{W_1}$ and $\mathbf{b_1}$. The first auto-encoders hidden layer output is fed to the second auto-encoder to learn $\mathbf{W_2}$ and $\mathbf{b_2}$.

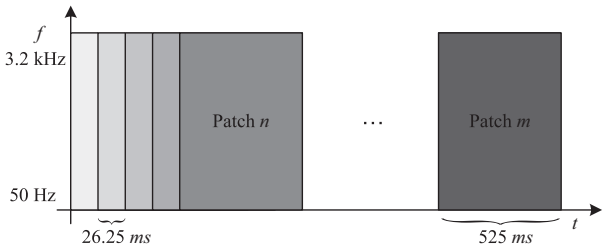
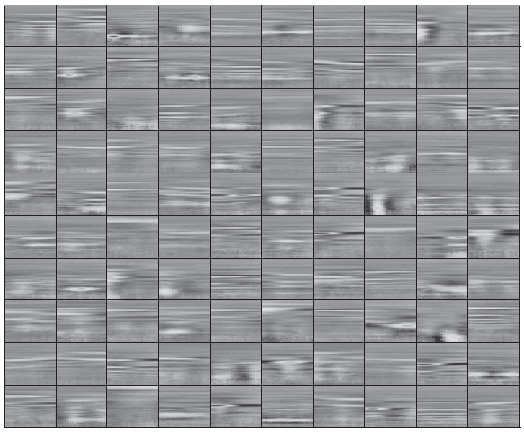


Fig. 3. Patch segmentation of a CQT spectrogram.

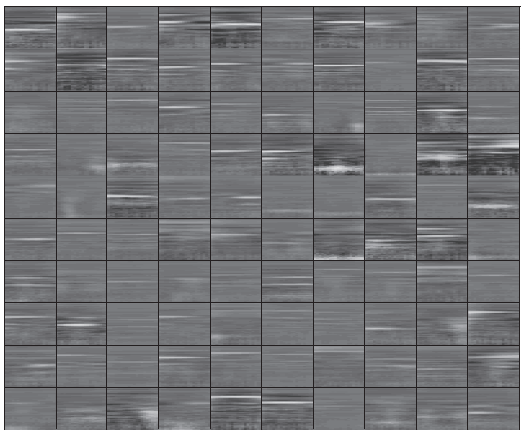
simulates the mechanism of actual nerve cells and helps to discover interesting feature structure [22].

We adopt the SAE with two hidden layers to learn a feature representation from vocal imitations. Instead of using raw audio waveforms as inputs to the SAE, we use a time-frequency representation as inputs. To do so, we first resample each training vocal imitation file with a sampling rate of 16 kHz. We then perform a 6-octave (50 to 3200 Hz) Constant-Q Transform (using CQT toolbox in the MATLAB environment) [23] to calculate a magnitude spectrogram with a logarithmic frequency scale. This logarithmic scale is in correspondence to the human auditory perception and greatly reduces the frequency dimensionality thanks to its lower frequency resolution at high frequencies. We use 12 frequency bins in each octave and in total 72 frequency bins for the entire spectrogram. The CQT hop size is 26.25 ms. We then segment the spectrogram into overlapping patches with a patch size of 525 ms (20 frames) and a patch hop size of 26.25 ms (1 frame). The patch size is chosen based on the fact that one syllable, which is the smallest unit to carry semantic meanings, is roughly 250 ms long on average in normal English speeches [24]. By choosing the patch size of 525 ms, we hope that patches capture some important temporal dependencies used in vocal imitations. Therefore, the spectrogram of each patch is a matrix with dimensions of 72*20. We then vectorize the matrix into a 1440-d vector and feed it to the SAE. Fig. 3 shows the patch segmentation of a CQT spectrogram of a vocal imitation query.

The SAE is thus designed to have 1440 input neurons. The number of neurons of the first and second hidden layers



(a) Visualization of the first hidden layer features.



(b) Visualization of the second hidden layer features.

Fig. 4. Feature visualization. Each subfigure is composed of 100 (10×10) spectrograms showing the first 100 features captured by the network weights in that layer. For each spectrogram the horizontal axis represents time and the vertical axis represents frequency. Lighter color represents higher energy.

are empirically set to 1000 and 600, respectively. Each neuron is fully connected to the neurons in the previous layer by a set of weights. After training, these weights represent a feature formula (or filter) captured by the neuron. In other words, the neuron is more activated when the previous layer outputs values similar to these weights. For the new vocal imitation inputs, the hidden layer activations are used as the features. Therefore, there are in total 1000 and 600 features in the first and second hidden layers, respectively, each of which captures a different pattern in the data. These patterns can be visualized using the method described in [18]. Fig. 4 shows the first 100 features in the first and second hidden layer. We can see that the first hidden layer extracts features that act as preliminary building blocks of the CQT spectrogram. The feature for each neuron in the second hidden layer is obtained by a weighted linear combination of features of the first hidden layer neurons to which it is strongly connected. These features are more abstract.

The second-hidden layer output is used to represent each patch, which is a 600-d vector. We further calculate the first-order derivative (delta) of the vector w.r.t. time to capture

longer-term temporal evolution, resulting a 1200-d vector for each patch. Therefore, each vocal imitation is represented by a sequence of 1200-d feature vectors.

3 PROPOSED SUPERVISED SYSTEM

Based on the 1200-d vector sequence representation of vocal imitations, we first design a supervised system for sound retrieval. We view each sound in the library as a class and use a multi-class SVM to classify a vocal imitation query to these classes. Sounds in classes with the highest classification confidence realized in [25] are retrieved for the imitation query. This system requires us to design recording-level features from the representations for classification, train the classifier, and retrieve sounds according to the classification results. We describe these modules in detail in the following subsections.

3.1 Feature Extraction

To convert the 1200-d patch-level vector sequence representation into recording-level features, we calculate six statistics in each dimension: maximum, minimum, mean, median, standard deviation, and interquartile range. Therefore, each vocal imitation query is represented as a 7200-d feature vector to achieve feature early fusion. Note that these statistics provide a simple summarization of the patch-level representation; however, it does not capture the temporal evolution beyond single patches. Given that temporal evolution within single patches is already captured by the SAE representation and the dimensionality is already quite high, we do not model longer-term temporal evolution at the recording-level features, to avoid overfitting in the classification.

3.2 Classification

We view each sound in the library as a potential class that a vocal imitation query falls into. We collect around 10 vocal imitations for each class and use them to train a multi-class SVM with the LIBSVM package [26]. We use the C-SVC classifier with a Radial Basis Function (RBF) kernel, and tune the cost of constraints violation $C = 1000$ empirically. Before training, we normalize each dimension of the 7200-d vector into the range of –1 and 1. For a new vocal imitation query, we perform the same normalization. We then classify it using the multi-class SVM, under the assumption that the query is within these classes.

3.3 Sound Retrieval

Given the classification result, sound(s) in the returned class can be retrieved. However, the returned class may not always be correct. Therefore, in addition to the binary classification output, we also obtain a probabilistic classification output, showing the probability (confidence) that the vocal imitation belongs to each of the classes. In LIBSVM [26], the one-against-one class probabilities are first calculated, then the posterior probability of a specific class can be obtained by solving the optimization problem described in [25]. We then sort these classes according to their classification probabilities from high to low and return sounds in the highly-ranked classes.

3.4 Discussions

In our preliminary work [7], we performed classification on the patch-level features and then obtained recording-level classification results through majority vote. Later, we observed that summarizing patch-level features with simple statistics and performing classification at the recording-level directly gives better results and saves computation. Therefore, we only describe this new setting in this paper.

It is noted that this supervised system does not compare the vocal imitation query with the sounds in the library directly. Instead, the link between them is established through classification, which is based on the assumption that training vocal imitations and the new imitation query of a sound are similar. This assumption is valid for many kinds of sounds, however; it can fail in some cases when the sound is complex and hard to imitate and when the training imitations are produced by people with very different cultural backgrounds from the user. In addition, this system does not work in an open set scenario, i.e., it cannot retrieve sounds/classes that are not trained. This is a significant limitation of the system to be deployed by itself to a large sound library where many sounds lack training imitations. However, the supervised idea is useful in sound retrieval by vocal imitation systems as it mines the collaborative information across users when more and more vocal imitation queries are contributed and collected.

4 PROPOSED UNSUPERVISED SYSTEM

To make sound retrieval more flexible and independent of the existence of training vocal imitations of sounds, we design an unsupervised system named IMISOUND. Again, this system uses the automatically learned representations for vocal imitations, which are described in Sec. 2. This system also represents sounds in the library with this representation, mapping sounds to the same feature space as vocal imitations. It then calculates the distances between the imitation query and each sound in the library and retrieves the closest sounds. In the following we describe the details of the distance calculation.

4.1 Distance Calculation

As described in Sec. 2, each imitation query is represented by a sequence of vectors, each of which corresponds to the second-hidden-layer output of the SAE taking a patch as input. We further represent each sound in the library in the same way by performing CQT, segmenting it into patches, and feeding them to the SAE. We can calculate the distance between the two vector sequences at two different levels: the patch-level that considers the temporal evolution or distribution of the vectors, and the recording-level that considers simple statistics of the vectors.

4.1.1 Patch-Level Distance

Here in the patch-level setting, we only use the original 600-d feature vectors but not the deltas for two reasons: (1) dimensionality reduction and ease of computation, (2) the delta components are designed to measure temporal evolution, but the distance calculation described below will cover this purpose.

So now both the imitation query and the target sound are represented by a sequence of 600-d vectors, although their sequence lengths can be quite different. If the imitation and the sound are similar, their feature sequences tend to resemble each other in terms of the temporal evolution or the probability distribution.

To consider the temporal evolution in distance calculation, we align the two sequences using Dynamic Time Warping (DTW) and use the alignment cost as a distance. This distance considers how the 600 dimensions evolve collectively over time and may capture pitch and timbre evolution. To perform DTW on the two feature sequences, we use cosine distance in Eq. (3) for the local cost measure that ignores the absolute energy difference. We align the first vectors and the last vectors of the two sequences and find the warping path that gives the lowest overall cost. This cost is the DTW distance we want, denoted by D_{DTW}.

To consider the probability distribution of the vectors, we calculate the symmetric K-L divergence along each dimension (e.g., the i-th dimension) as

$$D_{K-L_{(i)}}(P||Q) = \frac{1}{2}(D_{kl}(P||Q) + D_{kl}(Q||P)) = \frac{1}{2}\left(\sum_j P(j) ln \frac{P(j)}{Q(j)} + \sum_j Q(j) ln \frac{Q(j)}{P(j)}\right), \quad (1)$$

where P and Q represents the distribution along one dimension of the query and the sound candidate respectively, and j indexes the histogram bins for that dimension. Due to the sparsity of hidden neuron activation, many elements of the feature vectors are close to zero. We convert the feature values to a logarithmic scale first, and then approximate the distribution along each dimension with a histogram of 44 bins. Therefore, P and Q are actually the distributions of the logarithm of the feature values. As a 3-s recording contains about 114 vectors, we find the histogram can approximate the distribution well. Finally, the symmetric K-L divergence in all dimensions are summed together to obtain the overall symmetric K-L divergence D_{K-L}, which is used as the distance between the query and the sound. Fig. 5 illustrates the calculation process.

It is noted that this K-L divergence calculation assumes that different dimensions are independent, which misses the covariance between different dimensions. This design is to avoid the curse of dimensionality given the much fewer vectors than dimensions. In addition, the K-L divergence does not model the temporal evolution that can be very important in describing the similarity between sounds. However, compared to the DTW distance, the K-L divergence is easier to compute and may work better when the temporal evolution of the sound is not imitated well.

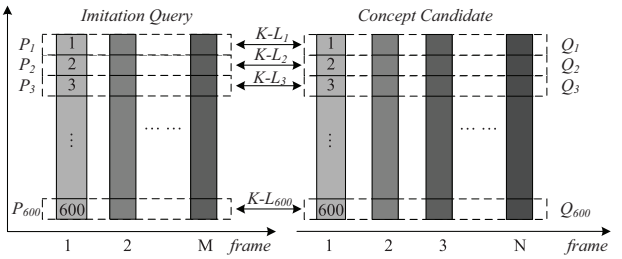


Fig. 5. K-L divergence calculation.

Given their complementary nature, we also propose to combine D_{K-L} and D_{DTW} in an L-1 space, i.e., summing them as another patch-level distance. To make sure that they are of the same scale, we normalize them by their maximal values before the summation. The final distance is thus calculated as

$$D = \frac{D_{K-L}}{max(D_{K-L})} + \frac{D_{DTW}}{max(D_{DTW})}. \quad (2)$$

4.1.2 Recording-Level Distance

In building the supervised system in Sec. 3.1, we adopted an early fusion technique to summarize the sequence of feature vectors of a vocal imitation with one single feature vector of their six statistics. This greatly reduces size of the representation and computation. Here we adopt the same idea to calculate a recording-level distance. To do so, both the imitation query and the sound candidate is represented by a 7200-d feature vector that includes the six statistics of each of the 1200 dimensions. Here in the recording-level setting delta components of the feature vectors are included because the distance calculation does not cover temporal evolution. Then the cosine distance between the two feature vectors is calculated by

$$d_{cos} = 1 - \frac{< x_s, x_t >}{||x_s|| \cdot ||x_t||}, \quad (3)$$

where x_s and x_t represent the feature vector for vocal imitation and sound candidate, respectively.

The cosine distance compares the angle between two vectors. It increases when the angle increases. One advantage of cosine distance over Euclidean distance is that it discards the magnitude, hence making the distance less affected by the volume mismatch between the imitation and the sound candidate.

The cosine distance is similar to the K-L divergence presented before in the sense that both calculate the distribution mismatch between the query and candidate and ignores the temporal information. Compared to the K-L divergence, the cosine distance is based on a more compact representation, is easier to compute, and considers the relation across dimensions. If the retrieval performances are similar, cosine distance is preferred as sounds in the library can be represented by a single vector.

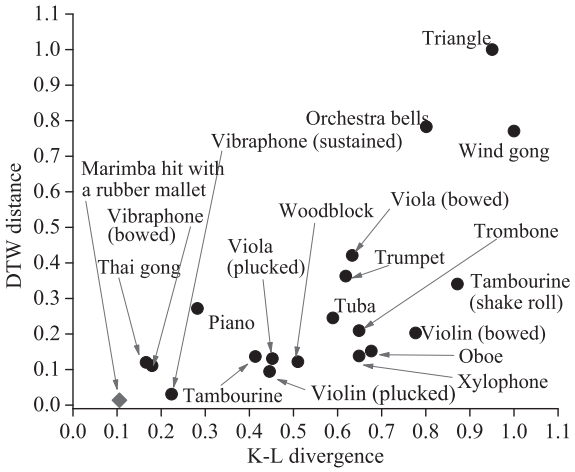


Fig. 6. Distances between a vocal imitation and sound candidates of different acoustic instruments.

4.2 Sound Retrieval

Distances between the imitation query and all sound candidates are then ranked, and candidates with the shortest distances are returned to the user.

Fig. 6 shows an example of sound retrieval after patch-level distance calculation between a vocal imitation query for "Marimba hit with a rubber mallet" and 20 sound candidates within the category of acoustic instruments. Most pitched sounds are of the same pitch. We see that the target sound "Marimba hit with a rubber mallet" is indeed the closest to the origin (the vocal imitation) in this 2-d space. After listening to the sound candidates, we find some interesting aspects. The closest candidates (e.g., "Vibraphone (sustained)," "Thai gong," "Violin (plucked)," "Tambourine," and "Piano"), including the target sound, are all percussive sounds except "Vibraphone (bowed)." Their K-L divergences are smaller than other candidates. We argue that this is because percussive sounds have a wider dynamic range than non-percussive sounds in each dimension, and this is captured by the K-L divergence. In addition, the several furthest candidates (e.g., "Triangle," "Orchestra bells," and "Wind gong") have very different frequency distributions in the CQT spectrogram from the vocal imitation, even though they are also percussive. Therefore, their 1200-d feature vectors obtained by passing the spectrogram through the SAE are very different from those of the imitation as well. This makes both their K-L divergences and the DTW distances large.

5 EXPERIMENTS

We conduct experiments to answer the following questions: (1) how do the automatically learned features compare with handcrafted features used in existing systems in both supervised and unsupervised settings in terms of retrieval performance? (2) How does the unsupervised system compare with the supervised system? (3) How do different distance calculations compare with each other?

5.1 Dataset

VocalSketch Data Set v1.0.4 [12] is adopted in our experiments. This dataset includes recordings of real life sound concepts in four categories, i.e., Acoustic Instruments, Commercial Synthesizers, Everyday, and Single Synthesizer. In each above category, there are 40, 40, 120, and 40 sound concepts respectively. In addition, each sound concept (real-world sound) in the dataset has around 10 to 20 vocal imitations obtained by people with different gender, various age range, nationalities, language skills, music backgrounds, etc., through Amazon's Mechanical Turk. There are two types of vocal imitations. One is to imitate in response to a reference sound recording. For example, the Amazon Turker is asked to listen to a recording of car horn first and then imitate the sound concept. The other one is to imitate in response to descriptive text labels, i.e., the Amazon Turker only has access to the word car horn and then imitate the sound concept based on his/her understanding. We only use the first type of vocal imitations in our experiments for evaluation purposes. Although the second type provides users more freedom and are closer to real-world situations, the similarity between the imitation and the target recording can be questionable in some cases, making the evaluation difficult, especially in the unsupervised setting. A detailed description of the sound concepts across all the categories is shown in Table 1.

5.2 Evaluation Measures

We use two measures to evaluate the system performance: (1) classification accuracy for the supervised system, as it is highly related to the sound retrieval performance. It is defined as the percentage of correctly classified imitations among all imitations within one of the four categories of the dataset. The rationale is that the user is clear about which category the target sound lies in when the imitation is made. (2) Mean Reciprocal Rank (MRR), for both the supervised and unsupervised systems. It is calculated as

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i}, \tag{4}$$

where $rank_i$ is the rank of the correct sound concept in retrieved sound list for the textiti-th vocal imitation; Q is the total number of testing vocal imitations. *MRR* ranges from 0 to 1 with a higher value for a better retrieving performance. A value of 1 means that the correct concept is always the top retrieving candidate. While a value of 0.5 suggests that the correct concept is ranked the 2nd among all concepts, on average. Again, *MRR* is calculated within one of the four categories.

5.3 Results of the Supervised System

5.3.1 Experimental Setup

For convenience, here we interchangeably use the term sound concepts with classes. In the four categories, we use vocal imitations from half of all the sound concepts to train the stacked auto-encoder for automatic feature learning.

Table 1. Description of the VocalSketch v1.0.4 dataset [12].

Category	Sound Concept	No. Concept
Acoustic Instruments	Orchestral instruments playing a single note with the pitch C in an appropriate octave	40
Commercial Synthesizers	Various recordings from Apples Logic Pro music production suite	40
Everyday	A wide variety of acoustic events in everyday life	120
Single Synthesizer	A single 15-parameter subtractive synthesizer playing a note with the pitch C	40

Then we use the rest half of the sound concepts to train and test the multi-class classifier within each category. This partition process helps us to prevent the multi-class classifier over-fitting sound concepts whose vocal imitations have been used in learning the feature representations. For the multi-class training and testing, there are 10 vocal imitations for each sound concept. Ten-fold cross validation is used to calculate the results.

5.3.2 Baseline Method

We compare the proposed system to a baseline system described in [14]. In that paper the authors first extract 472-d features including global descriptors and time-varying descriptors from each vocal imitation by the Timbre Toolbox [15]. Then features are fed to Weka [27] for SVM classification. A C-SVC classifier with an RBF kernel is again used, and the cost of constraints violation parameter C is set to 1000 as well to obtain the highest classification accuracy. Therefore, the baseline system only differs from the proposed system at the feature extraction stage.

5.3.3 Results

Table 2 shows performance comparisons between the proposed supervised system and the baseline system. We adopt 10-fold cross validation to avoid over-fitting. Several interesting results can be observed as the following.

First, both the proposed and baseline systems achieve significantly higher classification performance than random guesses. Note that random guess classification accuracies of the four categories would be 5%, 5%, 1.67%, and 5%, respectively. In addition, as shown in Table 2, the highest MRR (0.5822) of the proposed system is obtained by Single Synthesizer. This indicates that the correct sound concept is ranked between the 1st and the 2nd among the 20 concepts in that category, on average. The lowest MRR (0.3881) is obtained by Commercial Synthesizers. This value still tells that the correct sound concept is ranked between the 2nd and the 3rd among the 20 concepts in the category, on average. This indicates that the proposed supervised learning framework for vocal imitation recognition and retrieval is feasible and promising.

Second, the average classification and sound retrieval performance of the proposed system outperforms that of the baseline in all categories except Commercial Synthesizers. Higher values are shown in bold. This supports our claim that features learned automatically are more suitable than handcrafted features for vocal imitation recognition. This improvement is quite significant in the Single Synthesizer category whose semantic meanings are ambiguous. One possible reason for this improvement is that the Timbre Toolbox only extracts surface-level features. While the automatically learned features are able to reveal deeper connections between the vocal imitation and target sound concept.

Finally, we compare performances in different categories. We can see that both systems achieve much better results in the other three categories than the Commercial Synthesizer category, even including the Everyday category that has much more (60) classes than the other categories (20). After listening through all sounds and their imitations, we think that this is mainly because sounds in the other three categories are easier to imitate. Sounds in the Commercial synthesizer category, however, are more complex. Most of them contain multiple acoustic aspects such as transients, noise, and modulations on pitch and timbre. Therefore, they are more difficult to imitate and less consistency is expected among different people's imitations.

5.4 Results for the Unsupervised System

5.4.1 Experimental Setup

Similar to the supervised system, we use vocal imitations from half of all the sound concepts to train the stacked auto-encoder for feature learning, and the second half of the sound concepts for distance calculation and retrieval performance evaluation. Eventually, the unsupervised system is comparing vocal queries with real-world sounds, relying that the feature representations are similar enough in the two domains.

5.4.2 Comparison Methods

We compare four versions of the proposed system and a baseline system. Three out of the four versions use patch-level distances: DTW distance, K-L divergence, and their combination. The fourth version uses cosine distance in the recording level. For the baseline system, we designed it based on [14], by adopting the Timbre Toolbox to extract recording-level features for each vocal imitation and sound concept recording, and calculating the cosine distance. This baseline system is to validate the advantage of automatic feature learning over handcrafted features for our unsupervised sound retrieving task.

5.4.3 Results

Fig. 7 shows the performance comparisons. We describe several interesting observations in the following.

First, all four versions of the proposed system (the first four boxes in each category panel) outperform the baseline system in all categories. This indicates that the automatically learned features are more suitable than the carefully

Table 2. Classification and retrieving performance between the proposed and baseline system.

Category	Accuracy (Proposed)	MRR (Proposed)	Accuracy (Baseline)	MRR (Baseline)
Acoustic Instruments	**35.50%**	**0.5437**	27.00%	0.5114
Commercial Synthesizers	23.50%	0.3881	**29.00%**	**0.4547**
Everyday	**27.50%**	**0.4197**	26.33%	0.4168
Single Synthesizer	**43.00%**	**0.5822**	30.50%	0.4832

handcrafted features in the unsupervised setting as well. The highest MRR in our proposed system achieves 0.437 MRR in the Acoustic Instruments category using cosine distance. This means that on average, the target sound is ranked around between the 1st and 2nd among the 20 recordings in that category. For the Everyday category, there is a big gap between the unsupervised and its supervised counterpart. This may be due to the larger amount and diversity of sounds in this category. Nevertheless, the 0.142 MRR value achieved by DTW + K-L suggests that the target sound is ranked around the 7th among the 60 recordings in the category. It is noted that the MRR measure is very conservative in describing the system's performance in practice, since a user does not necessarily know precisely which sound he/she wants to retrieve. Sounds that are similar enough to the query should be all of some interest.

Second, the four versions of the proposed system do not show much difference. The combined patch-level distance (DTW + K-L) is slightly better than both DTW and K-L. It means that the MRR's obtained by combining of K-L divergence and DTW distance is better than those using either K-L divergence or DTW distance individually. This is because K-L divergence only measures the distribution difference of features, while DTW distance compares the difference of temporal evolution.

Finally, the recording-level distance (Cosine) achieves very similar performance with DTW + K-L. The lowest MRR value is 0.123 in the Everyday category. It means the target sound is still ranked between the 8th and 9th among the 60 recordings within the category. As analyzed in Sec. 4.1, the cosine-distance version is preferred compared with DTW + K-L distance, because its computation is much simpler and each sound in the library can be represented by a single feature vector instead of a sequence of feature vectors.

6 CONCLUSIONS

In this paper we proposed approaches to sound retrieval by vocal imitations. To address the feature representation challenge of vocal imitations, we employed a two-hidden-layer Stacked Auto-Encoder (SAE) to learn features automatically from a large variety of vocal imitations. We then designed two systems based on the feature representation. The supervised system views the retrieval problem as a classification problem of vocal imitations. We used a multi-class SVM to classify the imitation query to a sound class in the library. The unsupervised system calculates distances between the vocal imitation query and each of the sound candidates in the library and retrieves the closest ones. We explored different distances at both the patch-level and recording level. We conducted experiments using a large vocal imitation dataset. Experiments showed that the automatically learned features significantly outperformed handcrafted features used in existing systems in both supervised and unsupervised settings. In addition, the retrieving performance of the unsupervised system is promising as it does not require training imitation data for the sounds to be retrieved and can be scaled to larger libraries.

7 FUTURE WORK

For future work, we would like to further evaluate both systems' retrieving performance in larger-scale sound libraries. We also would like to implement a real, practical system for users to use and conduct user studies for this

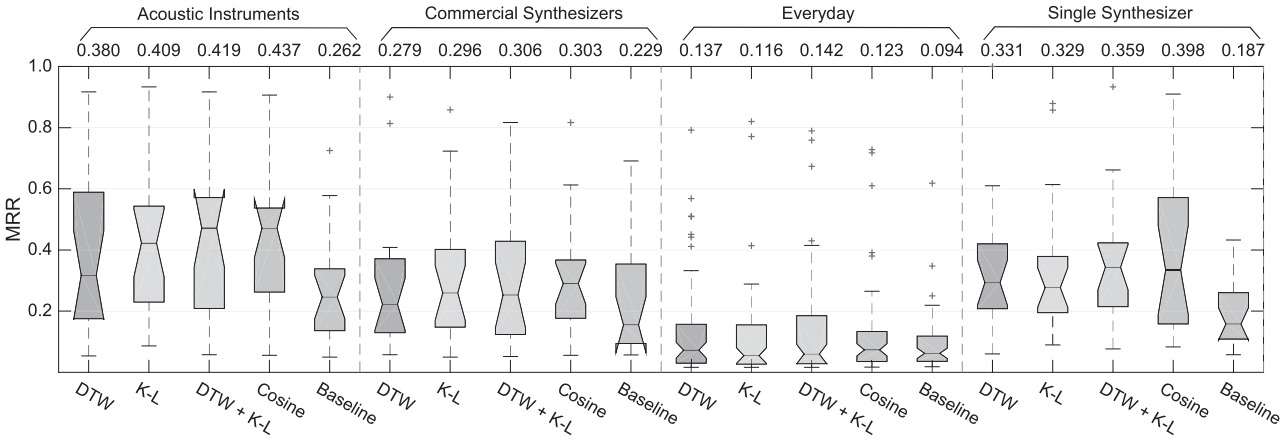


Fig. 7. Sound retrieving performance comparison between the proposed and baseline systems.

system. On the technical side, we would like to further improve the automatic feature learning module by exploring other deep neural networks such as Recurrent Neural Networks to model the temporal evolution of vocal imitations, or Convolutional Neural Networks to model the local correlations of the CQT spectrogram of vocal imitations similar to image processing.

8 ACKNOWLEDGEMENT

Special thanks go to Mark Cartwright and Bryan Pardo for generously providing us with the VocalSketch Data Set v1.0.4.

9 REFERENCES

[1] M. M. Zloof, "Query-by-Example: A Data Base Language," *IBM Systems J.*, vol. 16, no. 4, pp. 324–343 (1977 Dec.). http://dx.doi.org/10.1147/sj.164.0324

[2] A. Kapur, M. Benning, and G. Tzanekakis, "Query-by-Beating-Boxing: Music Retrieval for the DJ," *Proc. International Conference on Music Information Retrieval* (Barcelona, Spain, 2004), pp. 170–177.

[3] V. Kharat, K. Thakare, and K. Sadafale, "A Survey on Query by Singing/Humming," *Int. J. Computer Applications*, vol. 111, no. 14, pp. 39–42 (2015 Feb.). http://dx.doi.org/10.5120/19608-1484

[4] A. Ghias, J. Logan, D. Chamberlin et al., "Query by Humming: Musical Information Retrieval in an Audio Database," *Proc. the Third ACM International Conference on Multimedia* (New York, NY, 1995), pp. 231–236. http://dx.doi.org/10.1145/217279.215273

[5] T. Bertin-Mahieux and D. P. Ellis, "Large-Scale Cover Song Recognition Using Hashed Chroma Landmarks," *Proc. Applications of Signal Processing to Audio and Acoustics, 2011 IEEE Workshop on (WASPAA)* (New Paltz, NY, 2011), pp. 231–236. http://dx.doi.org/10.1109/ASPAA.2011.6082307

[6] J. Serra, H. Kantz, X. Serra et al., "Predictability of Music Descriptor Time Series and its Application to Cover Song Detection," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 514–525 (2012 Feb.). http://dx.doi.org/10.1109/TASL.2011.2162321

[7] Y. Zhang and Z. Duan, "Retrieving Sounds by Vocal Imitation Recognition," *Proc. Machine Learning for Signal Processing, 2015 IEEE 25th International Workshop on (MLSP)* (Boston, MA, 2015), pp. 1–6. http://dx.doi.org/10.1109/mlsp.2015.7324316

[8] Y. Zhang and Z. Duan, "IMISOUND: An Unsupervised System for Sound Query by Vocal Imitation," *Proc. Acoustics, Speech and Signal Processing, the 41st IEEE International Conference on (ICASSP)* (Shanghai, China, 2016), pp. 1–5.

[9] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, pp. 79–86 (1951 Mar.). http://doi.org/bm59cw

[10] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *Acoustics, Speech and Signal Processing, IEEE Transaction on*, vol. 26, no. 1, pp. 43–49 (1978 Feb.). http://dx.doi.org/10.1109/TASSP.1978.1163055

[11] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *Acoustics, Speech and Signal Processing, IEEE Transaction on*, vol. 28, no. 4, pp. 357–366 (1980 Aug.). http://dx.doi.org/10.1109/TASSP.1980.1163420

[12] M. Cartwright and B. Pardo, "VocalSketch: Vocally Imitating Audio Concepts," *Proc. the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, South Korea, 2015), pp. 43–46. http://dx.doi.org/10.1145/2702123.2702387

[13] G. Roma and X. Serra, "Querying Freesound with a Microphone," *Proc. the First Web Audio Conference* (Paris, France, 2015).

[14] D. S. Blancas and J. Janer, "Sound Retrieval from Voice Imitation Queries in Collaborative Databases," presented at the *AES 53rd International Conference: Sematic Audio* (2014 Jan.), conference paper P2-8.

[15] G. Peeters, B. L. Giordano, P. Susini et al., "The Timbre Toolbox: Extracting Audio Descriptors from Musical Signals," *J Acous. Soc. Amer.*, vol. 130, no. 5, pp. 2902–2916 (2011 Nov.). http://dx.doi.org/10.1121/1.3642604

[16] M. Helén and T. Virtanen, "Audio Query by Example Using Similarity Measures between Probability Density Functions of Features," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, pp. 1–12 (2010 Jan.). http://dx.doi.org/10.1155/2010/179303

[17] G. E. Hinton, S. Osindero, and Y. W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554 (2006 Jul.). http://dx.doi.org/10.1162/neco.2006.18.7.1527

[18] H. Lee, R. Grosse, R. Ranganath, et al., "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical representations," *Proc. the 26th Annual International Conference on Machine Learning (ICML)* (Montreal, Canada, 2009), pp. 609–616. http://dx.doi.org/10.1145/1553374.1553453

[19] G. E. Hinton, L. Deng, D. Yu, et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97 (2012 Nov.). http://dx.doi.org/10.1109/MSP.2012.2205597

[20] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics," *Proc. the 13th International Society for Music Information Retrieval Conference (ISMIR)* (Porto, Portugal, 2012), pp. 403–408.

[21] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507 (2006 Jul.). http://dx.doi.org/10.1126/science.1127647

[22] H. Lee, A. Battle, R. Raina, et al., "Efficient Sparse Coding Algorithms," *Proc. Advances in Neural Information Processing Systems* (Vancouver, Canada, 2006), pp. 801–808.

[23] C. Schörkhuber and A. Klapuri, "Constant-Q Transform Toolbox for Music Processing," *Proc. the 7th Sound*

and Music Computing Conference (Barcelona, Spain, 2010).

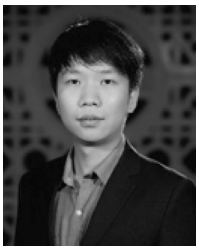
[24] E. M. Mugler, J. L. Patton, R. D. Flint, et al., "Direct Classification of All American English Phonemes Using Signals from Functional Speech Motor Cortex," *J. Neural Engineering*, vol. 11, no. 3, pp. 035015 (2014 May). http://dx.doi.org/10.1088/1741-2560/11/3/035015

[25] T. F. Wu, C. J. Lin, and R. C. Weng, " Probability Estimates for Multi-Class Classification by Pairwise Coupling," *J. Machine Learning Res.*, vol. 5, pp. 975–1005 (2004 Dec.).

[26] C. C. Chang and C. J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1–27 (2011 Apr.). Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm. http://dx.doi.org/10.1145/1961189.1961199

[27] M. Hall, E. Frank, G. Holmes et al., "The WEKA Data Mining Software: An Update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18 (2009 Nov.). http://dx.doi.org/10.1145/1656274.1656278

THE AUTHORS



Yichi Zhang

Yichi Zhang is a second-year Ph.D. candidate in the Department of Electrical and Computer Engineering at University of Rochester, in the AIR Lab under the supervision of Prof. Zhiyao Duan. He recived his M.S. degree in optical engineering focusing on optical fiber communications and DSP algorithms from Huazhong University of Science and Technology in 2014, under the supervision of Prof. Changjian Ke. He received his bachelors degree in electrical and information engineering from Wuhan Univeristy of Technology in 2011. His research interests include machine learning, deep neural networks, and computer audition.

Zhiyao Duan

•

Zhiyao Duan is an assistant professor in the Electrical and Computer Engineering Department at University of Rochester. He received his B.S. and M.S. in automation from Tsinghua University, China, in 2004 and 2008, respectively, and received his Ph.D. in computer science from Northwestern University in 2013. His research interest is in the broad area of computer audition, i.e., designing computational systems that are capable of analyzing and processing sounds, including music, speech, and environmental sounds. Specific problems that he has been working on include automatic music transcription, multi-pitch analysis, music audio-score alignment, sound source separation, speech enhancement, and sound retrieval.