Sound Texture Classification Using Statistics from an Auditory Model

Gabriele Carotti-Sha
Electrical Engineering
Stanford University
Email: gcarotti@stanford.edu

Evan Penn
Mangement Science & Engineering
Stanford University
Email: epenn@stanford.edu

Daniel Villamizar
Electrical Engineering
Stanford University
Email: danvilla@stanford.edu

***Abstract*—This project aims at applying machine learning techniques for the classification of acoustic textures and environments. Results are shown for different supervised learning methods, indicating that some of the proposed features are particularly useful for textural recognition.**

I. Introduction

Sound textures may be defined as a category of sounds produced by the superposition of many similar acoustic events. Falling rain, boiling water, chirping crickets, a moving train are some examples whose perceptual qualities can be captured to great extent by a small set of statistical measures, as shown by McDermott and Simoncelli [1]. As described in their paper, extensive work has been done to (1) analyze the features of sounds that are potentially used by the auditory system for textural recognition and (2) develop synthesis techniques to generate realistic-sounding textures based on these features. The notion of a texture is similar to that used in image processing, where new images are generated by first identifying characteristic distributions at each sample point of an exemplar and then extending those distributions to generate repeating patterns.

As a natural extension of this work, we propose the application of standard machine learning techniques to verify whether this same feature set can be used for classification.

Some audio texture examples are shown in Figures 1, 2, and 3. As can be seen, certain characteristics of the spectrum can be discerned to differentiate one class of sounds from another. However, it is in some cases difficult to know what class a particular texture belongs to. Note that for a human listener recognition is not a difficult task for the particular waveforms we chose; apart from steady state behavior, context and temporal pattern are extremely important psychoacoustic cues as well. The question is how small a feature space can we utilize for the purposes of classification.

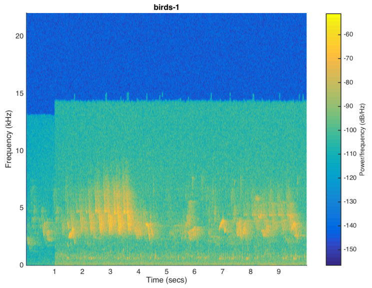

Fig. 1: birds chirping

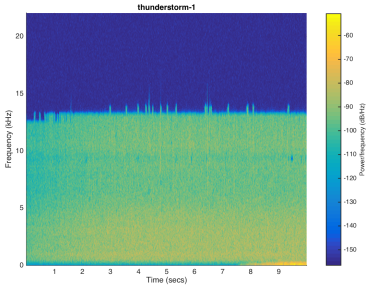

Fig. 2: thunderstorm

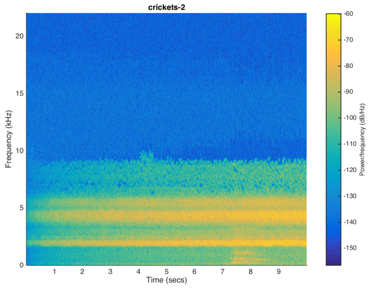

Fig. 3: crickets

II. Dataset and Features

A. Perceptual Model

We replicated the model of the auditory system developed in [1]. The input waveform, generally between 5 to 10 minutes long, is windowed into 7 second time frames with 50% overlap. Each frame serves as a training sample for measurement. Each window is then convolved with a bank of equivalent rectangular bandwidth (ERB) cosine filters whose center frequencies correspond to the masking sensitivity of human hearing across the spectrum. Qualitatively, acoustic stimuli with the same intensity are more easily discriminated at lower bands than at higher bands. The ERB bandwidths capture this phenomenon by providing finer resolution at the low end and increasing non-linearly up to Nyquist.

The envelope for each subband is then extracted by taking its Hilbert transform. A compression is applied in order to simulate the nonlinear sensitivity of the auditory system to intensity levels. Finally, a second filter bank is applied to each subband, further subdividing it into 23 modulation bands (see Figure 4). All filters are implemented as raised cosines so as not to introduce any power gain.

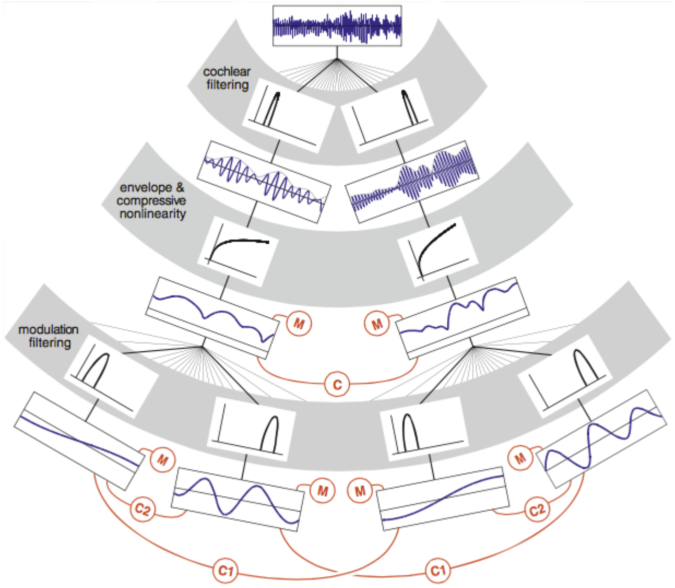

Fig. 4: Perceptual model implemented by McDermott and Simoncelli. Indicated as M are the moments computed for each subband and modulation band. C and C1 indicate correlations between subbands, whereas C2 are the correlations between modulation bands of a given subband.

B. Measured Features

Each input waveform is normalized (by signal RMS) so that the respective measures are on the same scale. The computed measures are: first through fourth moments and autocorrelation of each input subband, fourth through fourth moments of each subband envelope, power of each modulation band for each subband. This is a smaller set than that used by McDermott and Simoncelli, since they also included correlations between pre-modulated subbands and between modulated subbands. This was essential for synthesis, but not necessarily for classification.

C. Dataset

Train and test data was collected by taking live recordings of various acoustic environments (cafes, train and metro stations) and by accessing royalty free content online. We maintained input sampling rates at the limit of human hearing (44.1 kHz) with bit rate of either 16 or 24 bps. Train and test samples were taken from different recordings as an attempt to avoid overfitting.

III. Methods

We ported the public distribution of the Sound Synthesis Toolkit [2] to Python, implementing the previously described model. We then applied four different supervised learning methods: random forest, decision tree, regularized linear regression, and support vector classification using the scikit-learn distribution [3].

These features are numerous, leading to a need to avoid overfitting. We did this by keeping track of performance on meaningful feature subsets. We propose that there are two possible ways to use the recorded data when building the train/test set. One way is to take training and test samples

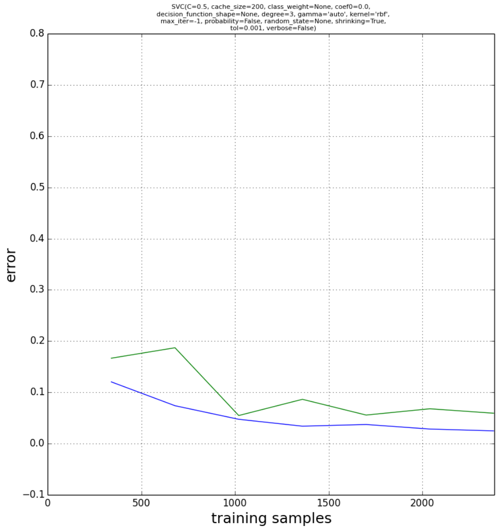
```
SVC(C=0.5, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
max_iter=-1, probability=False, random_state=None, shrinking=True,
tol=0.001, verbose=False)
```

Fig. 5: Train/test error for SVM with regularization of 0.5

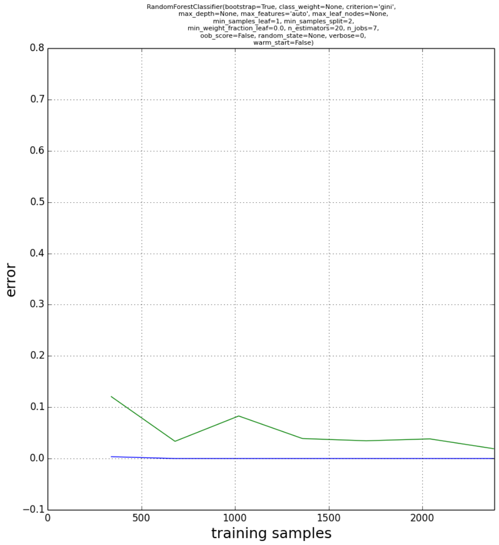
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
max_depth=None, max_features='auto', max_leaf_nodes=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=20, n_jobs=7,
oob_score=False, random_state=None, verbose=0,
warm_start=False)
```

Fig. 6: Train/test error for Random Forest Classifier using 20 estimators

from the same recording randomly. This would account for the scenario where we are trying to make predictions using the same equipment in a similar environment. Another way is to make the train/test data mutually exclusive with respect to the recordings such that no recording has windows in both the train and test split. This allows us to measure generalization performance over different instantiations of textures, rather than simply different time points in the recording of a single texture. Nonetheless, our models tended to fit the training data perfectly, and show only mediocre performance on test data. Further experiments showed that regularization strength did not make a large difference in either test or train performance. Clearly, more work is necessary.

We then performed feature selective and ablative analysis to determine the impact of the feature categories as shown in the tables below. The first set of features only includes correlations between subbands. The second set includes subband moments: mean, variance, skew, and kurtosis. The third set includes envelope features, while the last set shows all features combined.

We also attempted to implement a blended model that took into account all of the other models. For this, a validation set was exstracted from the test set. The validation set was not trained on. Instead, for each model the predicted class probabilities were kept for each validation row. This new matrix, with each row corresponding to a validation data point and k columns for each model where k is the number of classes. We then fit a logistic regression on the blended dataset. This did not yield an improvement. It may be because

IV. Results

A. Model Results

Initial results using train/test data from the same recordings showed a tendency to overfit as we discussed earlier (See Fig. 5 and 6). Our error plots show this trend. We show the confusion matrices, using all features, for Logistic Regression (one-vs-all) and Random Forest with 50 trees. The tables (last page) provide a breakdown of performance by model and feature subset on training and testing data, measured in simple accuracy. We see that logistic regression typically performs best, perhaps because its decision function is less complex and so resists overfitting. This data has 10 classes, so we are well above random guessing level.

V. Conclusion

A. Possible Applications

Given that each feature has an explicit physical meaning, the supervised classifiers thus generated can be implemented using specialized signal processing hardware. This offers the benefit of low-power consumption while maintaining high classification performance. These devices can be used to enable systems to become aware of the textural information of their environment. This could be useful, for example, in locations where visibility (or any non-acoustic sensing) is impaired. This technique could also be useful in voice texture recognition where the goal is to identify who is speaking rather than what they are saying.

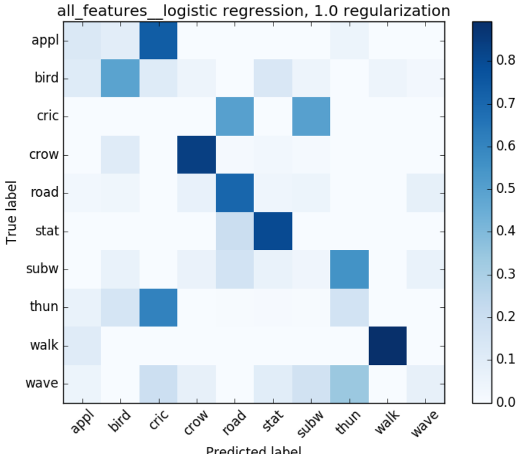

Fig. 7: logistic regression confusion matrix

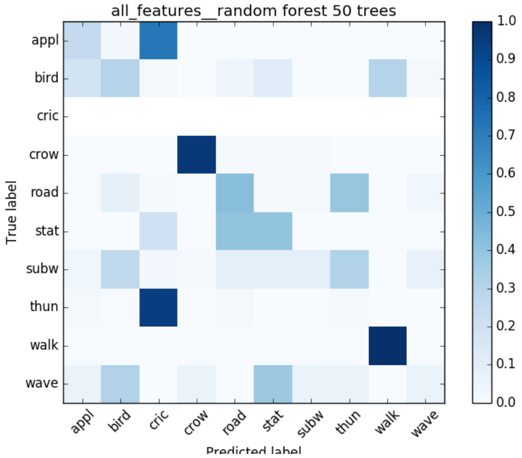

Fig. 8: random forest confusion matrix

B. Future Work

Our initial idea was to test whether this feature space could serve not only the purposes of classification of simple sounds, but also that of more complex signals (environments or speech). The first hurdle is that these features, as mentioned previously, are characteristic of the steady state signal. Transient information is essential to speech and to timbre information in general (a violin note, for example, is perceptually characterized by its onset as much as by its sustained acoustic behavior). The model would therefore have to incorporate some of the standard acoustic features (cepstral components, gaussian mixtures, etc.) used for more sophisticated applications. However, another direction to take would be to concentrate on the steady state behavior of an input source and study the limits of this representation. For example, though speech involves phonetic variation, the timbre quality of a person's voice, as in the sustained note performed by a professional singer, may very well be described by sets of time-averaged measures. Incorporating this information with time-varying structure could aid in realistic and personalized vocal synthesis or recognition.

Acknowledgment

The authors would like to thank Prof. Andrew Ng and the entire staff of CS229 at Stanford University for teaching an inspiring course. We would also like to thank Hyung-Suk Kim from the Dept. of Electrical Engineering, Stanford University, for pointing us to McDermott's research.

References

[1] J. H. McDermott, E. P. Simoncelli, *Sound Texture Perception via Statistics of the Auditory Periphery: Evidence from Sound Synthesis*, Neuron, 2011.

[2] http://mcdermottlab.mit.edu/bib2php/publications.php

[3] http://scikit-learn.org/stable/

TABLE I: Training Error Vs. Feature Subset for training data

Model	subband correlations	pre-modulation moments	modulated	all_features
random forest 4 trees	0.96	0.99	0.98	0.99
random forest 20 trees	1.00	1.00	1.00	1.00
random forest 50 trees	1.00	1.00	1.00	1.00
decision tree, max depth 3	0.50	0.60	0.58	0.59
decision tree, max depth 2	0.41	0.47	0.48	0.47
decision tree, no max depth	1.00	1.00	1.00	1.00
logistic regression, 0.1 regularization	0.80	0.83	1.00	1.00
logistic regression, 0.7 regularization	0.83	0.90	1.00	1.00
logistic regression, 1.0 regularization	0.83	0.91	1.00	1.00
SVM, rbf kernel, .5 regularization	0.71	0.79	0.95	0.95
SVM, rbf kernel, 1.0 regularization	0.90	0.86	0.98	0.99
GradientBoostingClassifier	0.94	0.91	0.99	1.00
ENSEMBLE	0.98	1.00	1.00	1.00
avg	0.84	0.87	0.92	0.92

TABLE II: Training Error Vs. Feature Subset for test data

	subband correlations	pre-modulation moments	modulated	all_features
random forest 4 trees	0.25	0.45	0.46	0.45
random forest 20 trees	0.30	0.52	0.49	0.47
random forest 50 trees	0.33	0.49	0.49	0.49
decision tree, max depth 3	0.17	0.27	0.35	0.32
decision tree, max depth 2	0.15	0.28	0.28	0.27
decision tree, no max depth	0.25	0.42	0.40	0.43
logistic regression, 0.1 regularization	0.25	0.41	0.51	0.51
logistic regression, 0.7 regularization	0.25	0.43	0.51	0.51
logistic regression, 1.0 regularization	0.24	0.43	0.51	0.51
SVM, rbf kernel, .5 regularization	0.28	0.43	0.48	0.47
SVM, rbf kernel, 1.0 regularization	0.28	0.45	0.49	0.50
GradientBoostingClassifier	0.26	0.38	0.41	0.42
ENSEMBLE	0.11	0.29	0.32	0.32
avg	0.24	0.40	0.44	0.44